Towards a unifying theory of generalization

by

Eric Schulz

First Supervisor:
Dr. Maarten Speekenbrink

Second Supervisor:
Prof. David R. Shanks

Internal Examiner:
Prof. Peter Dayan

External Examiner:
Prof. David S. Leslie

A DISSERTATION SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF

Doctor of Philosophy
Experimental Psychology
University College London

July, 2017

Declaration

I, Eric Schulz, declare that this thesis, titled "Towards a unifying theory of generalization", and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at University College London.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this university or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Eric Schulz University College London

To my grandparents.

Acknowledgments

First and foremost I would like to thank my supervisor Maarten Speekenbrink, who is likely the most encouraging and intelligent person I know. Thank you for literally leaving your door always open, for meeting me every day, for always being up for a chat about science in general and psychology in particular. You are not only a great supervisor but also a good friend.

I would also like to thank my second supervisor David Shanks for being someone I have always looked up to, someone who has helped me to appreciate the beauty of psychology.

Thanks to Philip Treleaven, Yonita Carter, and the UK Centre for Doctoral Training in Financial Computing and Analytics for funding this research.

I have also benefited greatly from 4 "unofficial supervisors". I would like to thank Samuel Gershman for hosting me during my time at Harvard and for having been an incredible collaborator throughout. Thanks also to Joshua Tenenbaum for hosting me at MIT and for all his excitement about the projects presented here. Andreas Krause also helped me greatly during my time at ETH Zürich and beyond by teaching me about the mathematical underpinnings of this thesis. Björn Meder has hosted me on countless occassions at the Max-Planck Institute in Berlin and always seems to ask the right "Wait but why"-question at the time.

I have been happy to meet many inspiring and smart people throughout my PhD. This thesis is the product of having collaborated with Dominik Bach, David Duvenaud, Zoubin Ghahramani, Sam Gershman, Jose Miguel Hernández Lobato, Quentin Huys, Emmanouil Konstantinidis, Andreas Krause, Björn Meder, Jonathan Nelson, David Reshef, Joshua Tenenbaum, and Charley Wu.

I would also like to thank my parents, Rainer and Carola Schulz, for their unconditional support and encouragement throughout my journey from a helpless undergraduate to a slightly less helpless postgraduate student, my brother for being the coolest teenager I know, and the rest of my family for always believing in me.

This journey has been considerably easier due to many great friends in London and abroad. Thank you Tarek Al-Zand and Stefan Irmscher for being my oldest friends, Neil Bramley for sharing a lot of this path with me, Chloe Bukata for life and board game advice, Leonardo Cohen for letting me eat his rice crackers, Saoirse Connor Desai for buying me

a tooth brush in Mexico, Will Crichton for literature conversations on St. Patrick's day, Ishita Dasgupta for collaborating on projects not included in this thesis, Pedro Miguel Esperanca Grillo for appreciating Bayesian stats, Tobias Gerstenberg for always making me feel welcome, Tom Hardwicke for promoting open science, Adam Harris for many helpful comments at the Lord John Russell, Abby Ipser for being small, Lara Kirfel for taking the Victoria line once, Dave Lagnado for reminding everyone that science is fun, Brad Love for shooting ideas at me, Philip Newall for hosting workshops, Paula Parpart for having 4 careers and still attending board game nights, Sarah Potratz and Christian Schnürch for buying a new couch, Benjamin Riedel for not taking himself too seriously, Kazys Simutis for keeping his calms, Deepesh Vadher for yelling "Sun" that one time, Sebastian Weichwald for being a walking mandala, and everyone else I have forgotten.

Thanks also to four eras of 2 Finsbury Park Road housemates who have helped to make London feel like home.

Finally, I would like to thank Cora Schefft for being there.

Abstract

How do humans generalize from observed to unobserved data? How does generalization support inference, prediction, and decision making? I propose that a big part of human generalization can be explained by a powerful mechanism of function learning. I put forward and assess Gaussian Process regression as a model of human function learning that can unify several psychological theories of generalization.

Across 14 experiments and using extensive computational modeling, I show that this model generates testable predictions about human preferences over different levels of complexity, provides a window into compositional inductive biases, and –combined with an optimistic yet efficient sampling strategy– guides human decision making through complex spaces.

Chapters 1 and 2 propose that, from a psychological and mathematical perspective, function learning and generalization are close kin. Chapter 3 derives and tests theoretical predictions of participants' preferences over differently complex functions. Chapter 4 develops a compositional theory of generalization and extensively probes this theory using 8 experimental paradigms.

During the second half of the thesis, I investigate how function learning guides decision making in complex decision making tasks. In particular, Chapter 5 will look at how people search for rewards in various grid worlds where a spatial correlation of rewards provides a context supporting generalization and decision making. Chapter 6 gauges human behavior in contextual multi-armed bandit problems where a function maps features onto expected rewards. In both Chapter 5 and Chapter 6, I find that the vast majority of subjects are best predicted by a Gaussian Process function learning model combined with an upper confidence bound sampling strategy. Chapter 7 will formally assess the adaptiveness of human generalization in complex decision making tasks using mismatched Bayesian optimization simulations and finds that the empirically observed phenomenon of undergeneralization might rather be a feature than a bug of human behavior.

Finally, I summarize the empirical and theoretical lessons learned and lay out a road-map for future research on generalization in Chapter 8.

Published and submitted work

The research presented in this thesis is based on a selection of papers that I have worked on during my doctoral studies.

The examples used in Chapter 3 appear in:

Schulz, E., Speekenbrink, M., & Krause, A. (2017b). A tutorial on Gaussian process regression with a focus on exploration-exploitation scenarios. *bioRxiv*, (pp. 095190)

Experiment 1 presented in Chapter 4 appears in:

Schulz, E., Tenenbaum, J. B., Reshef, D. N., Speekenbrink, M., & Gershman, S. (2015c). Assessing the perceived predictability of functions. In *Proceedings of the 37th Annual Meeting of the Cognitive Science Society* (pp. 2116–2121). Cognitive Science Society

Experiments 2-9 presented in Chapter 5 appear in:

Schulz, E., Tenenbaum, J. B., Duvenaud, D., Speekenbrink, M., & Gershman, S. J. (2016e). Compositional inductive biases in function learning. *bioRxiv*

Schulz, E., Tenenbaum, J., Duvenaud, D. K., Speekenbrink, M., & Gershman, S. J. (2016d). Probing the compositionality of intuitive functions. In *Advances in Neural Information Processing Systems* (pp. 3729–3737)

The experiments presented in Chapter 6 were originally inspired by:

Schulz, E., Huys, Q. J., Bach, D. R., Speekenbrink, M., & Krause, A. (2016a). Better safe than sorry: Risky function exploitation through safe optimization. In *Proceedings of the Thirty-Eighth Annual Conference of the Cognitive Science Society* (pp. 1140–1145).: Cognitive Science Society

Experiments 10-11 presented in Chapter 6 appear in:

Wu, C. M., Schulz, E., Speekenbrink, M., Nelson, J. D., & Meder, B. (2017). Mapping the unknown: The spatially correlated multi-armed bandit. In *Proceedings of the 39th Annual Conference of the Cognitive Science Society*: Cognitive Science Society

Wu, C. M., Schulz, E., Speekenbrink, M., Nelson, J. D., & Meder, B. (submitted). Exploration and generalization in vast spaces

Experiments 12-14 presented in Chapter 7 appear in:

Schulz, E., Konstantinidis, E., & Speekenbrink, M. (in press). Putting bandits into context: How function learning supports decision making. *Journal of Experimental Psychology: Learning, Memory, and Cognition*

Schulz, E., Konstantinidis, E., & Speekenbrink, M. (2015b). Learning and decisions in contextual multi-armed bandit tasks. In *Proceedings of the Thirty-Seventh Annual Conference of the Cognitive Science Society* (pp. 2122–2127).: Cognitive Science Society

Schulz, E., Konstantinidis, E., & Speekenbrink, M. (2015a). Exploration-exploitation in a contextual multi-armed bandit task. In *International Conference on Cognitive Modeling* (pp. 118–123)

Simulations 1-3 presented in Chapter 8 appear in:

Schulz, E., Speekenbrink, M., Hernández-Lobato, J. M., Ghahramani, Z., & Gershman, S. J. (2016b). Quantifying mismatch in Bayesian optimization. In *NIPS Workshop on Bayesian Optimization: Black-box Optimization and beyond*

Contents

Listing of figures

Listing of tables

"Because any object or situation experienced by an individual is unlikely to recur in exactly the same form and context, psychology's first general law should, I suggest, be a law of generalization."

Roger N. Shepard, 1987

1 Introduction

This chapter motivates the study of function learning to construct a theory of human generalization, introduces Gaussian Process regression as the main algorithm of this thesis, and lays out a roadmap of the chapters ahead.

1.1 The quest for a universal law

"Has psychology any hope for achieving a law that is comparable in generality (if not predictive accuracy) to Newton's universal law of gravitation?" Roger Shepard asked in 1987 in the light of the tercentenary of Newton's *Principia*. Indeed, the quest for universal psychological laws has been as old (and perhaps even older) as psychology itself. Willhelm Wundt tried to find the "elements of thoughts and what rules they adhere to" (Wundt, 1896). Herman Ebbinghaus lived a life of solitude collecting data only on himself in the attempt to find the "elements of a general theory of learning" (Ebbinghaus, 1885). And even today, the quest for generalized theories seems to be as timely and appropriate as ever with researchers assessing different unifying principles of cognition such as simplicity (Chater & Vitányi, 2003), free energy (Friston, 2010), or Bayesian sampling (Sanborn & Chater, 2017), others proposing ways of "how to grow a mind" (Tenenbaum et al., 2011), or yet even more ambitious goals such as Google DeepMind's agenda to "solve intelligence and then use it to solve everything else" (Simonite, 2016).

Back in 1987, Shepard argued that psychology's first general law should be a law of generalization as any object or situation experienced by an individual is unlikely to recur in exactly the same form and context. Today, 30 years later, Gershman & Daw (2017) point out that current theories of reinforcement learning cannot account for the fact that the "data are relatively sparse and, indeed, precisely the same situation may never be encountered twice". Whereas Shepard proposed an exponential similarity function in psychological space, Gershman & Daw (2017) propose a non-parametric representational mechanism guiding reinforcement learning. Yet, the overarching question remains the same: "How do humans and other animals know what to do or think in novel situations?". This question in turn immediately leads to other related questions such as "How are people able to learn and search for rewards in scenarios where not all options can be exhaustively explored?" or "What allows us to perceive and a draw inferences from patterns as efficiently as we do?". Shepard's assertion seems to be as valid as ever: the core principle guiding decision making and cognitive inferences is the ability to generalize beyond encountered observations (see also Tenenbaum & Griffiths, 2001, for a similar argument).

The aim of this thesis is to develop an empirically grounded and mathematically refined theory of human generalization. The main argument of this thesis is that if we want to study generalization, then –really– what we have to study is function learning. Put differently, the main premise of this thesis is that function learning, i.e. the ability to learn a mapping from inputs to outputs that generalizes to novel input-output pairs, can explain a big chunk of human generalization. If this is indeed true, then finding an expressive model of human function learning will shed light on some of our most fundamental psychological questions and would be applicable across many domains of cognition.

1.2 Generalization as function learning

I will define generalization as the ability to generate and act according to predictions for new observations based on underlying commonalities, i.e. as the ability to make predictive inferences about unobserved outcomes. At the core of this definition is the assumption that generalization can be achieved by learning about an underlying function that allows for generating predictions for both observed and novel inputs. This is different to other forms of induction that, given a few examples, extract a global rule (for example, Holyoak, 1985), as the rule does not necessarily have to be extracted all at once (functions can be learned piece-wisely) and predictions also come with a quantification of uncertainty (it is a case of Bayesian induction, so to say). It also goes beyond generalization in the "Shepardian" sense as novel observations do not only produce the same response as the most similar previously observed exemplar, but rather map onto a rich behavioral repertoire based on functional inferences. Function learning allows for global inferences which generalize to novel instances.

The presented results will be heavily based on the theory and application of Gaussian Process regression (Rasmussen & Williams, 2006). Gaussian Process regression is a non-parametric Bayesian approach towards regression problems that –instead of assuming a particular parametric form *a priori*– acts as a universal function approximator which can capture human behavior with mathematical elegance and computational ease (Williams & Rasmussen, 1996). It is also based on prior research on human function learning (Lucas et al., 2015) and can be extended to different psychological domains.

The focus of this thesis will be on simple function learning tasks first, that is the question of how people learn, complete, make judgments about, and predict functional patterns. Afterwards, Gaussian Process-based function learning will be extended to self-directed function learning scenarios in which participants have to both learn the mapping between inputs and outputs and make choices based on expectations generated by that mapping in order to gain rewards.

The main questions of this thesis are threefold:

1. What does it mean for a theory of generalization to be grounded in function learning?
2. How do people utilize functional generalization to search for rewards?
3. How do insights gained from human function learning behavior generalize to other domains?

Gaussian Process regression provides a useful tool to address all of these questions. First, as different forms of generalization can be encoded directly into Gaussian Process regression models, and as these forms can also be combined explicitly (Duvenaud et al., 2013), it can be used to derive a compositional theory of functions in which smaller parts are combined to explain complex structural patterns. Second, as Gaussian Process regression is also a common tool for problems involving the optimization of unknown functions (Snoek et al., 2012), it is an informative method to assess how humans approach tasks incorporating both function learning and the search for rewards. Third, as a mechanism of generalization can theoretically be applied to many other psychological domains, from learning causal relata from observations (Hoyer et al., 2009) to inferring people's intentions from their actions (Wang et al., 2012), a final question will be how much a theory of generalization grounded in function learning itself generalizes to other tasks, i.e. in what sense the proposed model can indeed be seen as a universal law, a psychological process model, or just a window into an empirically observed phenomenon.

1.3 Overview

This thesis consists of 9 chapters. Chapter 2 introduces the historical background of theories concerning human generalization and function learning. Therein, I argue that an understanding of function learning is cardinal to understanding human generalization. The mathematical underpinnings of Gaussian Process regression, kernel-encoded functional characteristics, structure search by kernel composition, as well as Bayesian optimization are laid out in Chapter 3. Chapter 4 demonstrates how precise predictions can be derived from Gaussian Process learning curves and demonstrate that factors theoretically influencing a function's generalizability equally influence participants' judgments. Chapter 5 introduces and extensively probes a compositional theory of function learning and generalization. Chapter 6 assesses how people explore and exploit options on a grid containing spatially correlated reward allowing for functional generalization. The contextual multi-armed bandit, a paradigm in which participants have to both learn about a functions mapping features to outcomes and generate rewards, is assessed across different degrees of difficulty in Chapter 7. Chapter 8 tries to shed more light onto the adaptiveness of human behavior by assessing what happens to different Bayesian optimization routines in the case of mismatched prior assumptions about the generalizability of observations. Chapter 9 concludes this thesis by summarizing the main lessons learned while laying out a road-map of how a Gaussian Process-based theory of generalization could further unify different psychological phenomena.

"It is not sufficient to think in terms of the discrete response to the discrete stimulus; what is being learned by these subjects is a mediating function, a set of rules or programs for transmuting an entire range of stimuli into a range of responses."

Douglas Carroll, 1963

2

From generalization to function learning

This chapter argues for function learning as a mechanism of generalization and summarizes psychological studies on function learning. Important effects and theories are explained.

2.1 Shepard's rule of generalization

Roger Shepard was among the first psychologists to study human generalization (Shepard, 1957, 1958). During the 1950s, operant conditioning was among the most popular theories of human and animal behavior, such that many scientists believed that a universal law of psychology should start with a law of how stimuli become associated with different responses (Watson et al., 1958; Skinner, 1945). However, Pavlov (1941) already observed that dogs would not only salivate at the sound of a bell that had preceded feeding on multiple trials before, but also at other stimuli, in particular sounds that were similar in pitch to the original sound. Afterwards, many behaviorists were interested in the so-called "gradient of stimulus generalization", i.e. how similar two different stimuli had to be in order to produce the same response. For example, Guttman & Kalish (1956) found that pigeons' ability to discriminate between different spectra of color along the wave-length continuum was predictive of the animals' response rate during extinction, showing a continuous gradient of generalization. This study led to a line of research investigating the factors that influence the gradient curve of generalization (see Honig & Urcuioli, 1981, for a thorough review).

Shepard (1957) proposed a theory of generalization to explain how similar stimuli can lead to the same response. The main idea behind his model was that the probability that a response to one stimulus will be generalized to another stimulus is a function of the distance between the two. "Generalization" in Shepard's theory is measured by means of confusion error (i.e., how often a particular stimulus leads to different responses), while "distance" is defined as a comparison between stimuli in a pre-defined psychological space.

For example, in one of his first experiments, Shepard (1958) used colored circles of uniform size and a constant red hue but with varying brightness and saturation and randomly assigned numerical responses from “one” through “nine” to these stimuli. When participants had to learn the labels of the different stimuli by being presented with a stimuli, guessing the label, and then observing the actual label, Shepard found that the probability of misclassification decreased with the distance between a stimulus’ brightness and saturation. Later on, using further experimental evidence from both human and non-human subjects, Shepard hypothesized that the probability of generalization $g(d)$ will fall off exponentially with the distance d between two stimuli scaled by a constant μ

$$g(d) = \exp\left(-\frac{2d}{\mu}\right). \tag{2.1}$$

An example of Shepard’s rule of generalization is shown in Figure 2.1. This plot shows that the probability of generalizing a response to a new stimulus from an old one (the ones shown left and right) decreases exponentially with their psychological distance.

2.2 Carroll’s idea to study function learning

Douglas Carroll, a PhD-student at Harvard at the time when Shepard had just taken up a faculty position there, wondered whether Shepard’s theory of generalization really explained all there was to human generalization. Had a universal law of generalization finally been discovered? And “what would happen when a large number of stimuli are associated with a large number of responses in complex learning situations”?

Carroll’s idea was that only determining a “gradient of stimulus generalization” cannot explain the richness of behavior we observe in both humans and other animals. In particular, Shepard’s idea was that a novel stimulus would lead to the same response as the closest seen stimulus so far. However, often times people (and other animals too) can provide a completely novel responses to a previously unobserved stimulus indicating a much

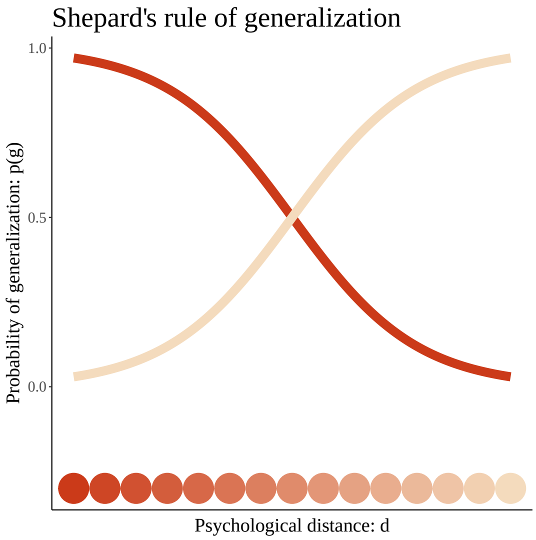


Figure 2.1: Example of Shepard's rule of generalization. As the distance between two stimuli grows, the probability to generalize one stimuli onto the same response as the other decreases exponentially. The dark red circle on the bottom left and the light red circle on the bottom right are the observed stimuli. The circles between the two stimuli are new observations. The dark red line marks the probability of a same responses as to the dark red circle. The light red line marks the probability of a same responses as to the light red circle.

richer behavioral repertoire. Indeed, looking closer at the results presented in Shepard (1958), there was some variance unexplained within the "colored dots"-task of which Carroll thought that "one might suppose that, lurking within this large individual variability, there could be something over and above generalization in the Hullian sense" (Carroll, 1963).

The studies described in Carroll's doctoral thesis entitled "Functional Learning: The Learning of continuous functional mappings relating stimulus and response continua" (Carroll, 1963) mark the first psychological experiments directly probing human function learning. They originated precisely from the intention to address questions about human generalization that Shepard had seemingly left unanswered.

In his experimental paradigm, Carroll assessed how participants associate continuous inputs with continuous outputs in a task that he referred to as "subjective curve fitting". His

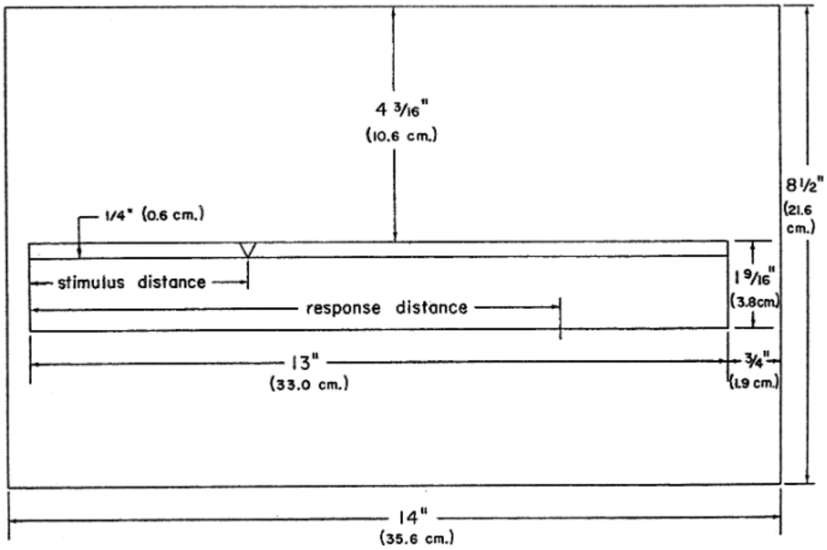


Figure 2.2: Carroll's original paradigm (taken from the original publication). Stimulus distance is marked by a "V". Response distance is marked by a vertical slash.

experiments asked people to first associate an input, marked as the horizontal distance from the left side of a paper and a letter "V", with a response marked as the horizontal distance from the left side of a paper and a vertical slash (see Figure 2.2).

For the underlying functions connecting stimuli to responses, Carroll used –unknown to participants– either a positive linear, a quadratic, or a random function. Moreover, after the initial learning stage in which participants only had to learn the association between inputs and outputs, Carroll asked participants to produce responses (i.e., to draw a new vertical dash onto the paper) given both previously seen but also novel input dimensions.

Carroll's model of human generalization was inspired by an idea put forward by von Neumann et al. (1941) that treats curve fitting as an attempt to explain as much variance as possible (measured by an "index of continuity" which is mathematically equivalent to minimizing least squares). By this definition, participants learn an association between a stimulus S and a response R by fitting a function f such that as much of the variance between S

and R is explained as possible by using as few parameters p_i as necessary.

$$R = f(p_0, p_1, \cdots, p_k, S) \tag{2.2}$$

The function f then can be used to generate responses that can be very different from the seen responses so far.

As an infinite number of possible functions are possible, Carroll focused on simple polynomials*. His experiments showed that participants were indeed able to learn *S-R* combinations better if these were governed by an underlying function instead of just a random mapping such as the ones originally applied by Shepard. Moreover, linear functions were easier to learn than non-linear functions. Most importantly, however, instead of simply mapping the response to a novel stimulus to the response of the closest stimulus experienced so far, participants were indeed able to correctly extrapolate to stimuli they had not previously encountered, an empirical indicator of a facet of human behavior that goes beyond generalization as defined by operant conditioning.

Thus, Carroll's was the first ever published theory on function learning and his model of human generalization went beyond simple stimulus-response-matching.

Carroll's theory belongs to the category of so-called *rule-based accounts of function learning*. These accounts propose that participants approach generalization by learning a set of fixed functional rules (for example, polynomial regression weights of a fixed order) and then try to find the best rule (for example, by optimizing the weights) in order to generalize from observed to unobserved data.

*Carroll mentions that other functions, for example sinusoidal combinations, could also be incorporated into his framework.

2.3 Rule-based accounts of function learning and generalization

During the 1970s, Brehmer (1971) tried to further assess what kind of functional rules might be harder to learn than others. His experiments utilized a similar design to Carroll's with the only difference that participants had to associate the length of a vertical line with a numerical output between 0 and 40. Moreover, Brehmer assessed the ease of learning 4 types of functions: a linear function with a positive trend, a linear function with a negative trend, a U-shaped (quadratic) function, and a inverse U-shaped function. Interestingly, Brehmer (1971) never mentions the discussion about psychological rules of generalization, but rather only talks about how participants learn different functional rules. Thus, his work seems to mark the time when psychologists began to study function learning for its own sake without direct implications for rules of human generalization[†].

The results of Brehmer's experiments showed again that participants were able to learn linear functions better than non-linear ones. Furthermore, linear functions with a positive trend were easier to learn than linear functions with a negative trend. Surprisingly, there was no reliable difference between the U-shaped and inverse U-shaped functions. In a later study (Brehmer et al., 1974), participants had to learn different functions while also providing verbal reports. The results led Brehmer (1976) to postulate a hierarchical model of functional hypothesis testing, according to which participants start out expecting a linear function with a positive trend and adjust their hypothesis if the encountered evidence is strong enough.

One of the first studies that assessed how well participants can control (instead of only

[†] Historically, this could also be due to the fact that generalization was then broadly discussed among psychologists studying categorization, where Shepard's ideas ultimately led to Nosofsky's Generalized Context Model (Nosofsky, 1986).

learn) the output of different functions was conducted by Berry & Broadbent (1984). Within a paradigm called the "sugar factory" participants had to learn how a continuous value called "work force " relates to the amount of sugar a factory can produce. Moreover, participants had to later produce a given value of sugar per trial, thereby having to control the output of sugar over time. These experiments also assessed whether participants are better at controlling linear than exponential functions and found significant task improvements when a non-linear function was "linearized" before the task started. This led to the belief that participants generally have problems controlling non-linear functions such that Berry & Broadbent (1984) suggested to present non-linear functions in a linear way to make control easier in applied settings. This led to a line of research investigating how participants control complex systems (see Osman, 2010, for an extensive review).

Koh & Meyer (1991) let participants learn associations between a stimulus' length and the duration of a response where the underlying function was either a power law function, a logarithmic function, or a linear function with a positive slope. They found a systematic response bias during the early stages of learning for the logarithmic and linear functions but not the power-law function. However, this bias gradually disappeared as training progressed. Koh & Meyer (1991) therefore proposed an adaptive regression model similar to that of Brehmer (1976) but with the additional constraint that stimulus-response pairs are treated as a power function. This idea was explicitly conceived by the way participants seem to generalize cross-modally in psycho-physics experiment (Shepard, 1981).

A general problem for rule-based accounts of function learning is that the class of possible models a participant can represent might be extremely large or –possibly– infinite. Thus, it is not *a priori* clear why only a particular set of parametric shapes should be considered. This means that it is not at all obvious how participants acquire knowledge of underlying rules, nor how they manage to assess all possible rules given some functional data. Therefore, psychologists during the 1990s tried to overcome the problem of having to predefine a parametric set of functions by using models that only learn associatively, i.e. based on simple input-output similarities. Interestingly, the resulting *similarity-based accounts of function learning* are again very close to Shepard's original theory of generalization.

2.4 Similarity-based accounts of function learning

Similarity-based accounts of generalization in function learning propose that people learn functions by associating inputs with outputs: if input x is paired with output y, then inputs similar to x should produce outputs that are similar to y. Busemeyer et al. (1997) formalized this intuition using a connectionist model (Associative-Learning Model; ALM) in which inputs activate an array of hidden units representing a range of possible input values; each hidden unit is activated in proportion to its similarity to the current input. Learned associations between the hidden units and the response units map the similarity-based activation pattern to output predictions. More specifically, when a cue value, $x(t)$, is presented, it activates input node X_i from the set of n input nodes according to a Gaussian similarity function:

$$a_i[x(t)] = \frac{1}{\exp\left(X_i - \psi_x(x(t))/\sigma\right)}. \tag{2.3}$$

where ψ is a transformation function and σ scales the distance between the two values.

This means that inputs are related to each other by how far apart they are and then produce similar outputs as compared to the seen output for previously observed nearby inputs. The output of the response node is governed by a weighted sum of associations for which the weights are estimated by a simple delta-rule of learning (Stone, 1986). The Associative

Learning Model contains 3 free parameters. The first one is simply the learning rate of the delta-rule α and therefore is not so important for the discussion about generalization here. However, Busemeyer et al. (1997) called the other two scaling parameters, σ_1 and σ_2, "the generalization gradients for the physical continuum used to display the stimuli and the generalization criterion for the output". Thus, this part of the Associative Learning Model explicitly ties back to Shepard's original law but within a function learning context. However, this also means that such a model, even though it is able to learn how participants respond to stimuli they have previously encountered, naturally has problems accounting for the way participants generalize to novel inputs. If new inputs are assessed based on how close they are to previously observed inputs, then what is going to happen if a new input is far away from the experienced input space?

DeLosh et al. (1997) conducted several studies focusing on extrapolation. As the title "Extrapolation: the sine qua non for abstraction in function learning" suggests, DeLosh et al. (1997) propose that extrapolation, i.e. generating predictions for novel inputs, should be the yardstick to measure the performance of different function learning models. What is interesting about this proposal, is that generalization to new observed points is now used to compare different models of function learning, whereas, previously, function learning was conceived specifically as a mechanism of generalization. To put this differently, one could argue that learning step-wisely within an interpolation set already involves generalization as points have to be observed for the first time at some point during training. Nonetheless, DeLosh et al. (1997) brought the discussion about generalization back into research on human function learning. In their experiments, participants had to learn the relation between scales of possible stimulus magnitudes which ranged from from 0 to 100 as indicated by an unfilled horizontal bar labeled numerically in 10-point increments. However, to allow for extrapolation trials during the transfer phase of the experiment, the magnitudes presented during training were limited to values between 30 and 70, and the test trials contained both stimuli from within that range but also extrapolation trials that were either smaller than 30 or larger than 70. The underlying functions were either linear, exponential, or quadratic.

DeLosh et al. (1997)'s results showed again that linear functions are easier to learn than exponential functions, which in turn are easier to learn than quadratic functions. More importantly however, something unexpected occurred during extrapolation trials. Even though participants were generally able to extrapolate all of the functions, extrapolation was worse than interpolation and turned out to be more linear than expected. This "linearity bias" even affected cases for which the underlying function had been non-linear; for example, when trained on a quadratic function, participants' average predictions fell between the true function and straight lines fitted to the closest training points as seen in Figure 2.3.

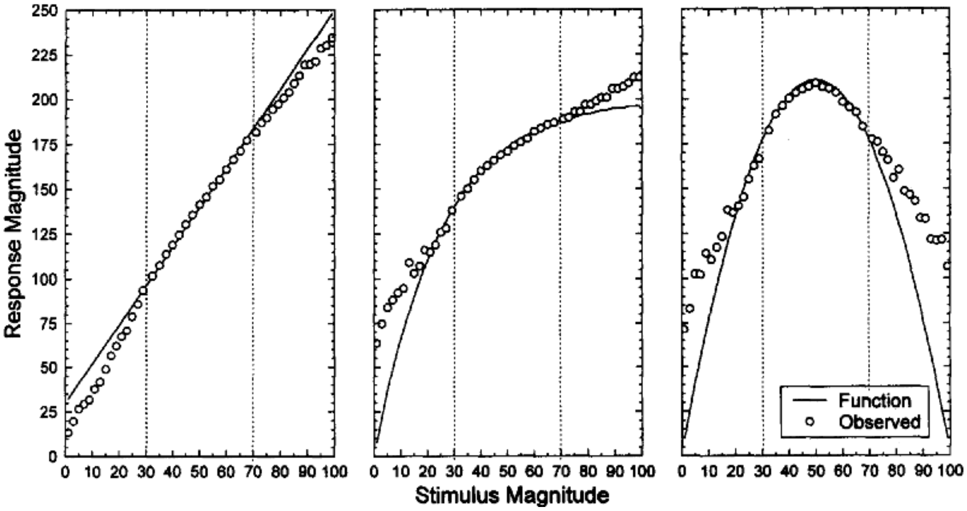


Figure 2.3: Results of extrapolation experiment conducted by DeLosh et al. (1997). When learning a quadratic function, participants' extrapolation lines fall between the true function and straight lines fitted to the closest training points.

That participants tend to extrapolate functions linearly is nowadays recognized as an established phenomenon (Kalish et al., 2004; Kwantes & Neal, 2006). Thus, although similarity-based models are more flexible than rule-based parametric accounts, they face the problem of having to explain how participants generalize beyond the encountered input space. One possible solution to this problem is to combine the more powerful similarity-based function approximation techniques with the extrapolative abilities of simple rules. This leads to so-called *hybrid accounts of function learning*.

2.5 Hybrid accounts of function learning

Hybrid accounts of function learning normally contain an associative learning process that acts on explicitly-represented rules. One such hybrid model, the *Extrapolation-Association Model* (EXAM) put forward by Busemeyer et al. (1997), assumes similarity-based interpolation, but extrapolates using simple linear rules. The model does this by falling back onto a parametric representation of the underlying function whenever generalization to points outside of the input space is required. EXAM effectively captures the human bias towards linearity, and predicts human extrapolations for a variety of functions. However, it does not account for participants' ability to extrapolate non-linear functions (Bott & Heit, 2004) and also has problems explaining other effect, such as the *knowledge partitioning effect*. This effect is based on the observation that participants' functional knowledge can be partitioned into distinct subsets based on a provided context. For example, Lewandowsky et al. (2002) showed that fire fighters learn two distinct relationships between wind speed, ground slope, and the rate at which a fire spreads, depending on whether the fire was labeled as a "standard forest fire", or a "back burn" fire set to mitigate damage from future fires. As EXAM generates uni-modal predictions for any given input, it cannot account for this partitioning effect.

A hybrid model that can explain this knowledge partitioning effect is the *population of linear experts model* (POLE; Kalish et al., 2004). This model approximates functions using piece-wise linear representations and captures the knowledge partitioning effect by approximating partitioned functions with different locally linearized models[‡]. Moreover, it also explains the order-of-difficulty effects collected and put forward by McDaniel et al. (2009), i.e. that some functions are easier to learn than others. However, it is unclear if a combination of linear functions is able to explain human function learning in more complex and naturalistic settings, especially given that participants are able to learn in non-linear settings as well (see Bott & Heit, 2004). Additionally, Byun (1995) assessed whether participants can learn more complex functions using either a systematically-increasing sequence of stimulus magnitudes during training, or a randomly-organized sequence of the same magnitudes. These training sequences facilitated learning of non-monotonic quadratic and cyclic functions, with systematic sequences producing better performance later on. Thus, participants seem to be able to learn more complex than linear functions if the learning procedure is simplified.

2.6 Function learning everywhere?

The studies described so far have shown how the study of human generalization has inspired the study of function learning. However, generalization more broadly and function learning in particular are also important in many other psychological domains. Even though many of these studies do not mention the terms "function learning" or "generalization" directly, some of them have nonetheless come up with related theories and discovered related phenomena.

[‡] Any differentiable function can be represented in locally linear terms

Another area explaining how people make sense of and generalize from functional structure are investigations of human forecasting. Within studies about forecasting, participants are normally confronted with multiple data points at once and then have to predict future or otherwise left-out points. At least 4 different factors that can influence the way in which participants forecast visually presented data have been established (see Bolger & Harvey, 1998; Goodwin & Wright, 1993, for detailed reviews). First, participants have a tendency to dampen trends when generating forecasts from noisy data. This means that their forecasts lie below upward trend lines but above downward ones. Therefore, it appears that forecasters tend to underestimate the steepness of functions (Andreassen & Kraus, 1990; Keren, 1983). Trend damping is greater for downward than for upward trended data, especially when the data representation format is visual rather than in a data table (Harvey & Bolger, 1996). Second, forecasts tend to overestimate functions lacking a trend (Eggleton, 1982), i.e. sometimes people perceive a linear trend where there is none. Thirdly, participants seem to deliberately attach random noise to their forecasts, and add more noise to forecasts from noisier data series. This means that their forecasts appear to represent the way the series will appear once the outcome has occurred instead of a smooth prediction line (Harvey Teresa Ewart Robert West, 1997). Finally, forecasts for independent series should lie on the series mean, but instead they have been found to lie between the mean and the last revealed data point (Eggleton, 1982), a finding very similar to the results of the more traditional function learning paradigms.

A related paradigm which is also heavily based on function learning is multiple cue probability learning (MCPL, Kruschke & Johansen, 1999; Speekenbrink & Shanks, 2008). In MCPL tasks, participants are shown an array of cues that are probabilistically related to an outcome and have to learn the underlying function mapping the cues' features to expected outcomes. One well-known paradigm of such studies is the "Weather Prediction Task" (Speekenbrink et al., 2008). In this task, participants predict the state of the weather ("rainy" or "fine") based on a set of "tarot cards". Gluck et al. (2002) found that participants broadly seem to use three different strategies to learn within this task. The first is an optimal multi-cue strategy, in which they respond to each pattern on the basis of associations of all cues with each outcome, thus learning the function reasonably well. The second is a one-cue strategy, in which they respond on the basis of presence or absence of a single cue, disregarding all other cues. Finally, the third strategy is a singleton strategy, in which they learn only about the four patterns that have only one cue present and all others absent.

Function learning has also been discussed in the reinforcement learning literature. Already in Watkins (1989), different methods on how to generalize expected rewards over continuous or large state spaces are discussed and possible function approximators such as neural networks or the so-called "Cerebellar Model Articulation Computer", a hash function to retrieve values over spaces, are introduced. Sutton (1996) proposed generalization by function learning as a method to further improve reinforcement learning algorithms. Nowadays, deep neural networks, a powerful methods of function approximation and generalization, aid reinforcement learning models to reap rewards in complex tasks such as playing Atari games (Mnih et al., 2015) or Go (Silver et al., 2016).

Psychologically, Ludvig et al. (2008) recently proposed linear value function approximations for generalization across time in order to predict future rewards. Finally, Gershman & Daw (2017) argue for a kernel-based approximation of the value function that maps past episodes onto future expected values. Again, all of these proposals essentially propose function learning as a tool for generalization.

2.7 Discussion

Studies involving participants' understanding and representation of functions were originally inspired by fundamental questions about laws of human generalization and are nowadays scattered around different psychological sub-fields. Whereas original theories of human function learning focused on generalization explicitly and postulated rule-based mechanisms, later studies promoted similarity-based learning algorithms and frequently studied function learning without explicit claims about generalization. The current state of the science are hybrid accounts that were introduced to explain how people generalize to unobserved data. Within the field of subjective forecasting, a list of possible "forecasting biases" has been put forward, some of which seem to mirror established findings within the more traditional function learning literature. Multiple cue probability learning paradigms ask participants to make predictions based on multi-variate functions. Here too it has been found that participants tended to learn in a linear way but can adapt to non-linear functions if necessary. Within reinforcement learning approaches, function learning can be seen as a mechanism of value function approximation and generalization and has been put forward as a proposal within the framework of episodic reinforcement learning.

These studies show that function learning is important for various psychological domains. Moreover, the topics described here are only the ones that directly mention the term "function". Beyond that, there are also plenty of other branches of psychological research that –in one way or another– can be conceptualized as function learning. For example, multi-attribute decision making (Bröder & Schiffer, 2003) requires participants to learn to associate different features with expected outcomes, pattern completion paradigms (Kanizsa et al., 1976) ask participants to extrapolate a pattern from seen points, and generalization across environments (Gershman & Niv, 2015) requires participants to extrapolate from features of the current environment to novel ones. That function learning is seen a separate subfield of psychology is mostly due to historical developments and practical conventions.

“It is a friendship made in heaven and executed on earth.”

Eric Kandel about Aplysia (Eric Schulz about GPs)

3

From function learning to generalization

This chapter introduces Gaussian Processes regression as a tool to investigate function learning and shows how assumptions about the kernel correspond to assumptions about human generalization.

3.1 Motivation

Let me first define the set-up of learning a function more formally before then discussing how it relates to generalization. If f denotes an (unknown) function which maps inputs x to outputs y, $f : X \rightarrow Y$, then function learning is the task to learn about f as well as possible provided the current input-output observations $\mathcal{D} = \{X, y\}$. Learning about f can progress by the 3 following ways.

In a *single function learning* scenario, the task is to find the best model of f given the current data $\mathcal{D}$. This is normally required in tasks that focus on either forecasting or pattern completion (as all of the data is revealed at once) and the representation of f is used to generate summaries of the presented patterns or to generalize the underlying function to novel input points. Somewhat misleadingly, psychology normally defines interpolation as generating predictions for input points within a provided set during the learning stage and extrapolation as generating predictions for input points from outside the learning set. However, generalization can technically already happen when generating predictions for input points within a learning set which have not been observed previously, i.e. there is a difference between interpolative judgments that require functional knowledge as compared to a more simplistic cued recall. I will nonetheless adhere to the traditional distinction between interpolation and extrapolation in this thesis.

A *sequential function learning* task is made of a sequence of multiple *single function learning* tasks such that a participant is confronted with input-output pairs, i.e. the data $\mathcal{D}$, step-wisely and –conventionally– has to generate predictions at every step. After making a prediction, the participant is shown the next data points, supposedly updates her underlying model, and so forth. This process makes up the key assumption of models used in the more traditional function learning tasks such as the ones described in Chapter 2[*].

The other variant of function learning differs from the previously described two variants in the way data points are selected. Whereas in single and sequential function learning tasks, the participant receives data points either randomly or in a way pre-determined by the experimentalist (for example, presented from left to right, etc.), in *self-directed function learning tasks* participants have to actively select input points (and then observe the output for the selected input points) in order to either learn about or optimize the underlying function f as well as possible (Gureckis & Markant, 2012). This requires participants to form expectations about the usefulness of different possible observations given their current representation of f, then to select the input they think promises to be be maximally useful, observe the output, update their model, and so forth. This thesis will particularly focus on *function exploration-exploitation* tasks in which participants are asked to produce the highest sum of outputs over trials. This in turn asks for a delicate trade off between exploration, that is to sample unknown points in order to learn more about the underlying function, and exploitation, that is to sample points that are expected to produce high outputs.

To summarize, the following two ingredients are needed to characterize all function learning scenarios:

1. A model of f that is used to learn and generalize about f's shape.
2. A sampling strategy to select inputs based on the current knowledge of f.

The difference between 1 and 2 also maps onto the distinction between belief and sampling models, central to theories in statistics (Lindley, 1956), active learning (Nelson, 2005), and philosophy of science (Crupi & Tentori, 2014).

[*] Note that one could equally differentiate between visually presented points (sometimes called "curve fitting") and sequentially presented outputs of different magnitudes. Function learning really comes in many forms.

Whereas all variants of function learning need to address what kind of model participants utilize to represent the underlying function, assessing which sampling strategy describes participants best will only be prevalent in studies concerning *function exploration-exploitation*. The next sections will first describe how to find a useful mathematical model of how participants learn about *f* before one particular sampling strategy to optimize unknown functions is introduced.

3.2 Linear regression: the weight space view

I begin by considering the most frequent approach to model unknown functions: linear regression (here defined as Bayesian linear regression). This can also be seen as a Bayesian variant of the rule-based approaches described in Chapter 2 and seems to capture the way in which participants generalize to unseen input spaces reasonably well.

As an example, I will use the set of stimulus-response pairs presented in Table 3.1. These contain observations of stimuli (inputs) x_t and responses (outputs) y_t at different time points t. The running example will also include having to make a new prediction of the value of y given a new stimuli $x_\star = 3$. In linear regression, it is assumed that the outputs are

Table 3.1: Observations for the function learning example. Inputs x_t and corresponding outputs y_t observed at times $t = 1, \ldots, 6$.

t	x_t	y_t
1	0.9	0.1
2	3.8	1.2
3	5.2	2.1
4	6.1	1.1
5	7.5	1.5
6	9.6	1.2

a linear function of the inputs with additional noise:

$$\begin{aligned} y_t &= f(x_t) + \varepsilon_t & (3.1) \\ &= \beta_0 + \beta_1 x_t + \varepsilon_t, & (3.2) \end{aligned}$$

where the noise term ε_t follows a normal distribution

$$\varepsilon_t \sim \mathcal{N}(0, \sigma_\varepsilon^2) \tag{3.3}$$

with mean 0 and variance σ_ε^2. This can also be written in matrix algebra as

$$y_t = \mathrm{x}_t^\top \mathrm{w} + \varepsilon_i \tag{3.4}$$

defining the vectors

$$\mathrm{x}_t = \begin{bmatrix} 1 \\ x_t \end{bmatrix}, \qquad \mathrm{w} = \begin{bmatrix} \beta_0 \\ \beta_1 \end{bmatrix}. \tag{3.5}$$

To predict the output for $x^\star$, one needs to estimate the weights from the previous observations

$$\mathrm{X}_t = \begin{bmatrix} 1 & 0.9 \\ 1 & 3.8 \\ \vdots & \vdots \\ 1 & 9.6 \end{bmatrix}, \qquad \mathrm{y}_t = \begin{bmatrix} 0.1 \\ 1.2 \\ \vdots \\ 1.2 \end{bmatrix}. \tag{3.6}$$

In a Bayesian set-up, this is done through the posterior distribution over the weights. If a Gaussian prior over the weights $p(\mathrm{w}) = \mathcal{N}(0, \Sigma)$ is used and likelihood is also Gaussian $p(\mathrm{y}_t|\mathrm{X}_t, \mathrm{w}) = \mathcal{N}(\mathrm{X}_t^\top \mathrm{w}, \sigma_\varepsilon^2 \mathrm{I})$, then the posterior distribution is

$$\begin{aligned} p(\mathrm{w}|\mathrm{y}_t, \mathrm{X}_t) &\propto p(\mathrm{y}_t|\mathrm{X}_t, \mathrm{w})p(\mathrm{w}) \\ &= \mathcal{N}\left(\frac{1}{\sigma^2}\mathrm{A}_t^{-1}\mathrm{X}_t\mathrm{y}_t, \mathrm{A}_t^{-1}\right) \end{aligned} \tag{3.7}$$

where $\mathrm{A}_t = \Sigma^{-1} + \sigma^{-2}\mathrm{X}_t\mathrm{X}_t^\top$. As inference here is performed over the weights (i.e., one tries to find the best estimate for the β-weights), this is also sometimes referred to as "the weight space view of regression". To predict the output $y_\star$ at a new test point $\mathrm{x}_\star$, the error term can be ignored and the focus is on the expected value which is provided by the function f, predicting $f_\star = y_\star - \varepsilon_\star = f(\mathrm{x}_\star)$. In the predictive distribution of $f_\star$, the uncertainty regarding the weights is averaged out

$$\begin{aligned} p(f_\star|\mathrm{x}_\star, \mathrm{X}_t, \mathrm{y}_t) &= \int p(f_\star|\mathrm{x}_\star, \mathrm{w})p(\mathrm{w}|\mathrm{X}_t, \mathrm{y}_t)d\mathrm{w} \\ &= \mathcal{N}\left(\frac{1}{\sigma^2}\mathrm{x}_\star^\top\mathrm{A}_t^{-1}\mathrm{X}_t\mathrm{y}_t, \mathrm{x}_\star^\top\mathrm{A}_t^{-1}\mathrm{x}_\star\right) \end{aligned} \tag{3.8}$$

A good prediction is the mean of this predictive distribution and comparing the mean to that in (3.7), one can simply multiply the posterior mean of w with the new input $\mathrm{x}_\star$, resulting in the prediction $0.56 + 3 \times 0.12 = 0.92$.

Bayesian linear regression suffers from the same drawbacks as the rule-based accounts of function learning introduced before: it assumes that encountered functions have indeed a linear shape. However, adaptive generalization makes it necessary to learn about non-linear dependencies as well. One straight-forward adjustment is to use a projection of the inputs x onto a "feature space" by using a function $\varphi(\mathrm{x})$. A common projection is to use polynomials, resulting in polynomial regression. For example, in cubic regression, which assumes a function $f(x) = \beta_0 + \beta_1 x + \beta_2 x^2 + \beta_3 x^3$, this results in a Bayesian equivalent of the most complex model put forward by Carroll (1963). Deriving the posterior for this model is similar to the linear regression described before, with the only difference that the input matrix X_t is replaced by its projection

$$\Phi_t = \varphi(\mathrm{X}_t) = \begin{bmatrix} 1 & 0.9 & 0.81 & 0.729 \\ 1 & 3.8 & 14.44 & 54.872 \\ \vdots & \vdots & \vdots & \vdots \\ 1 & 9.6 & 92.16 & 884.736 \end{bmatrix}. \tag{3.9}$$

In the aforementioned example, this would result in the prediction $f_\star = -0.67 + 0.98 \times 3 - 0.13 \times 3^2 + 0.01 \times 3^3 = 1.37$.

Projecting input variables onto a feature space offers considerably more flexibility and allows one to encode functions of many different shapes. Therefore, it is not a surprise that this method was put forward by earlier theories of human function learning (Carroll, 1963; Brehmer et al., 1974)[†].

However, this flexibility is also a drawback. There are infinitely many projections possible and one has to choose one either a priori or by model comparison within a set of possible projections. Especially if the task is to learn about a completely unknown function, which is a problem people frequently face in the real world, this approach will not be feasible as there is little guidance as to which projections should be included. From a psychologi-

[†] Of course the frequentist proposals of function learning models at that time also went hand in hand with statistical advances in computing regression weights (Zellner, 1962; Hoerl & Kennard, 1970), an interdependency that is frequently found in the history of psychology (Gigerenzer, 1991)

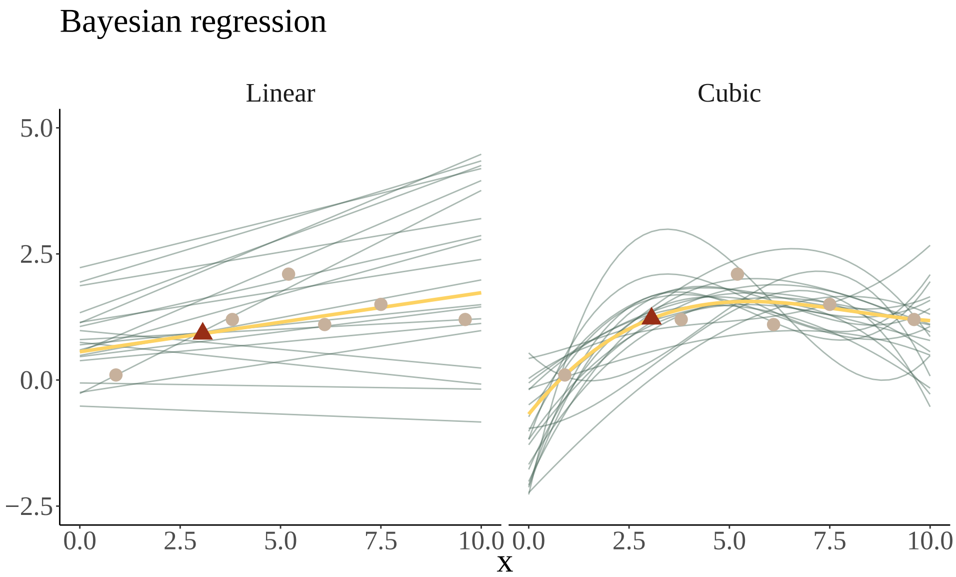


Figure 3.1: Example of performing Bayesian linear and cubic regression. Green lines indicate predictions for different sampled posterior weights. Dots mark empirical observations. Yellow lines mark the current mean posterior predictions. The red triangle shows the prediction for the newly presented stimuli $x_\star = 3$.

cal perspective, if the aim is to find universal rules of generalization based on the parametric Bayesian regression approach, then one would have to test out many different parameterizations and how they combine, distinguishing between idiosyncratic and nomothetic parametric rules. Gaussian Process regression, which I will introduce next, offers a principled solution to the problem of choosing between many functional forms as the projections are chosen implicitly, effectively letting the observed data directly contribute to the decision about model complexity. Moreover, the properties of the chosen kernel function directly correspond to a gradient of generalization in function space. Nonetheless, I will return to the problem of assessing how different subparts are combined in Chapter 5.

3.3 The function space view

In the weight space view of the previous section, the focus has been on distributions over weights. As each set of weights implies a particular function, a distribution over weights also implies a distribution over functions. Gaussian Process regression focuses on such distributions over functions directly. A Gaussian Process (GP) defines a distribution over functions such that, if any two points are picked, observations of the outputs of the function at these two points follow a joint Gaussian distribution. More formally, a Gaussian Process is defined as a collection of random variables, any finite number of which have a multivariate Gaussian distribution. In Gaussian Process regression, the output y of a function f at input x can be written as

$$y = f(\mathrm{x}) + \varepsilon \tag{3.10}$$

with $\varepsilon \sim \mathcal{N}(0, \sigma_\varepsilon^2)$. Note that this is similar to the assumption made in linear regression, i.e. that an observation consists of an independent "signal" term $f(\mathrm{x})$ and a "noise" term ε. Different in Gaussian Process regression, however, is that the assumption that the signal term is also a random variable which follows a particular distribution. This distribution is subjective in the sense that the distribution reflects the uncertainty regarding the function. The uncertainty regarding f can be reduced by observing the output of the function at different input points. The noise term ε reflects the inherent randomness in the observations. In Gaussian Process regression, the function $f(\mathrm{x})$ is taken to be distributed as a Gaussian Process:

$$f(\mathrm{x}) \sim \mathcal{GP}\left(m(\mathrm{x}), k(\mathrm{x}, \mathrm{x}')\right). \tag{3.11}$$

A Gaussian Process $\mathcal{GP}$ is defined by a *mean* and a *covariance* function. The mean function $m(\mathrm{x})$ reflects the expected function value at input x:

$$m(\mathrm{x}) = \mathbb{E}[f(\mathrm{x})], \tag{3.12}$$

i.e. the average of all functions in the distribution evaluated at input x. The prior mean function is frequently set to $m(\mathrm{x}) = 0$ in order to avoid expensive posterior computations and only perform inference via the covariance directly. The covariance function $k(\mathrm{x}, \mathrm{x}')$ models the dependence between the function values at different input points x and x′:

$$k(\mathrm{x}, \mathrm{x}') = \mathbb{E}\left[(f(\mathrm{x}) - m(\mathrm{x}))(f(\mathrm{x}') - m(\mathrm{x}'))\right] \tag{3.13}$$

The function k is normally called the *kernel* of the Gaussian Process (Jäkel et al., 2007). Commonly, the choice of an appropriate kernel is based on assumptions such as smoothness and likely patterns to be expected in the data. One very popular choice of a kernel function is the radial basis function kernel, which is defined as

$$k(\mathrm{x}, \mathrm{x}') = \sigma_f^2 \exp\left(-\frac{\|\mathrm{x} - \mathrm{x}'\|^2}{2\lambda^2}\right). \tag{3.14}$$

The radial basis function provides an expressive kernel to model smooth functions. It models the correlation between points in dependency of their distance.

As in Shepard's universal law of generalization, the radial basis function kernel models this dependency as a exponential function. The two hyper-parameters λ (called the length-scale) and σ_f^2 (the signal variance) can be varied to increase or reduce the correlation between points and consequentially the smoothness of the resulting function from a mathematical perspective. However, from a psychological perspective, λ could also be called the "gradient of functional generalization" as it determines how quickly the correlation between points decays as their distance grows longer and thereby how similar a function will be at two different points given their mutual distance. Figure 3.2 shows differently broad gradients of functional generalization resulting from setting the length-scale parameter λ to different values.

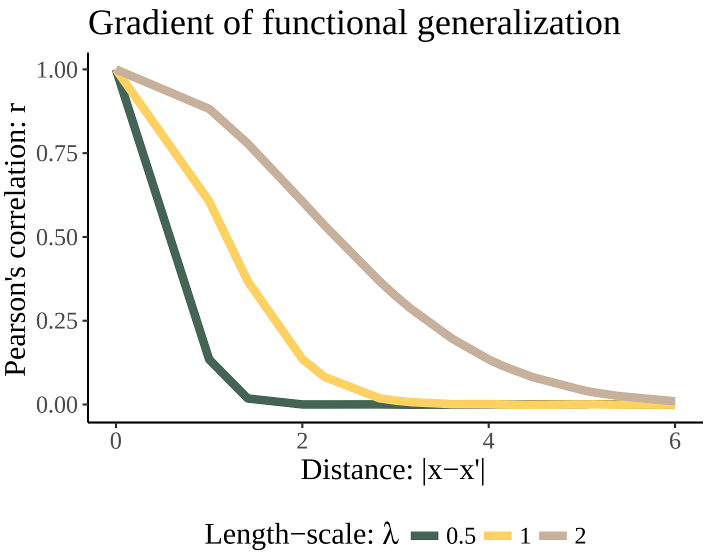


Figure 3.2: Different gradients of functional generalization as induced by setting the length-scale $\lambda = \{0.5, 1, 2\}$. The correlation is assessed as the mean correlation between two points in dependency of their mutual distance. For bigger λ-parameters, the correlation decays more slowly leading to broader generalization.

This also means that, theoretically, observations are expected to look more similar to the overall mean as their distance to the observed points grows longer. A radial basis function kernel therefore operationalizes Shepard's exponential rule of generalization in function space.

3.3.1 Sampling functions from a GP

Although Gaussian Processes are continuous, sampling a function from a Gaussian Process is generally done by selecting a set of input points. Theoretically, a function can be represented as a vector of infinite size; however, as one only has to make predictions for finitely many points in practice, it is feasible to simply draw outputs for these points by using a multivariate normal distribution with a covariance matrix generated by the kernel. Let $\mathrm{X}_\star$ be a matrix with on each row i a new input point $\mathrm{x}_i^\star$, $i = 1, \ldots, n$. To sample a function, the covariances between all inputs in $\mathrm{X}_\star$ is first computed between all points and collected in a $n \times n$ matrix:

$$K(\mathrm{X}_\star, \mathrm{X}_\star) = \begin{bmatrix} k(\mathrm{x}_1^\star, \mathrm{x}_1^\star) & k(\mathrm{x}_1^\star, \mathrm{x}_2^\star) & \ldots & k(\mathrm{x}_1^\star, \mathrm{x}_n^\star) \\ k(\mathrm{x}_2^\star, \mathrm{x}_1^\star) & k(\mathrm{x}_2^\star, \mathrm{x}_2^\star) & \ldots & k(\mathrm{x}_2^\star, \mathrm{x}_n^\star) \\ \vdots & \vdots & & \vdots \\ k(\mathrm{x}_n^\star, \mathrm{x}_1^\star) & k(\mathrm{x}_n^\star, \mathrm{x}_2^\star) & \ldots & k(\mathrm{x}_n^\star, \mathrm{x}_n^\star). \end{bmatrix} \tag{3.15}$$

Choosing the usual prior mean function $m(\mathrm{x}) = 0$, sample values of f at inputs $\mathrm{X}_\star$ from the $\mathcal{GP}$ are generated by sampling from a multivariate normal distribution

$$\mathrm{f}_\star \sim \mathcal{N}(0, K(\mathrm{X}_\star, \mathrm{X}_\star)) \tag{3.16}$$

where the notation $\mathrm{f}_\star = [f(\mathrm{x}_1^\star), \ldots, f(\mathrm{x}_n^\star)]^\top$ is used. Note that $\mathrm{f}_\star$ is a sample of the function values. To sample observations $\mathrm{y}_\star$, an additional and independent sample of the noise term ε has to be added.

3.3.2 Posterior predictions and generalization with GPs

Suppose observations $\mathcal{D}_t = \{\mathrm{X}_t, \mathrm{y}_t\}$ have been collected and the goal is to make predictions for new inputs $\mathrm{X}_\star$ by drawing $\mathrm{f}_\star$ from the posterior distribution $p(f|\mathcal{D}_t)$. By definition, previous observations y_t and function values $\mathrm{f}_\star$ follow a multivariate normal distribution. This distribution can be written as

$$\begin{bmatrix} \mathrm{y}_t \\ \mathrm{f}_\star \end{bmatrix} \sim \mathcal{N}\left(0, \begin{bmatrix} K(\mathrm{X}_t, \mathrm{X}_t) + \sigma_\varepsilon^2 \mathrm{I} & K(\mathrm{X}_t, \mathrm{X}_\star) \\ K(\mathrm{X}_\star, \mathrm{X}_t) & K(\mathrm{X}_\star, \mathrm{X}_\star) \end{bmatrix}\right) \tag{3.17}$$

where I is an identity matrix (with 1's on the diagonal, and 0 elsewhere) and σ_ε^2 is the assumed noise level of observations (i.e. the variance of ε). Using standard results, the conditional distribution $p(\mathrm{f}_\star|\mathrm{X}_t, \mathrm{y}_t, \mathrm{X}_\star)$ is then a multivariate normal distribution with mean

$$K(\mathrm{X}_\star, \mathrm{X}_t)[K(\mathrm{X}_t, \mathrm{X}_t) + \sigma_\varepsilon^2 \mathrm{I}]^{-1} \mathrm{y}_t \tag{3.18}$$

and covariance matrix

$$K(\mathrm{X}_\star, \mathrm{X}_\star) - K(\mathrm{X}_\star, \mathrm{X}_t)[K(\mathrm{X}_t, \mathrm{X}_t) + \sigma_\varepsilon^2 \mathrm{I}]^{-1} K(\mathrm{X}_t, \mathrm{X}_\star) \tag{3.19}$$

Note that this posterior is also a GP with mean function

$$m_t(\mathrm{x}) = K(\mathrm{x}, \mathrm{X}_t)[K(\mathrm{X}_t, \mathrm{X}_t) + \sigma_\varepsilon^2 \mathrm{I}]^{-1} \mathrm{y}_t \tag{3.20}$$

and kernel

$$k_t(\mathrm{x}, \mathrm{x}') = k(\mathrm{x}, \mathrm{x}') - K(\mathrm{x}, \mathrm{X}_t)[K(\mathrm{X}_t, \mathrm{X}_t) + \sigma_\varepsilon^2 \mathrm{I}]^{-1} K(\mathrm{X}_t, \mathrm{x}') \tag{3.21}$$

To predict $\mathrm{f}_\star$, the mean function in (3.20) can be used, or functions from the GP with this mean function and kernel (3.21) can be sampled as described in the previous section.

Figure 3.3 shows samples from a GP prior parameterized by a radial basis function kernel and the posterior mean functions after the data in Table 3.1 has been observed.

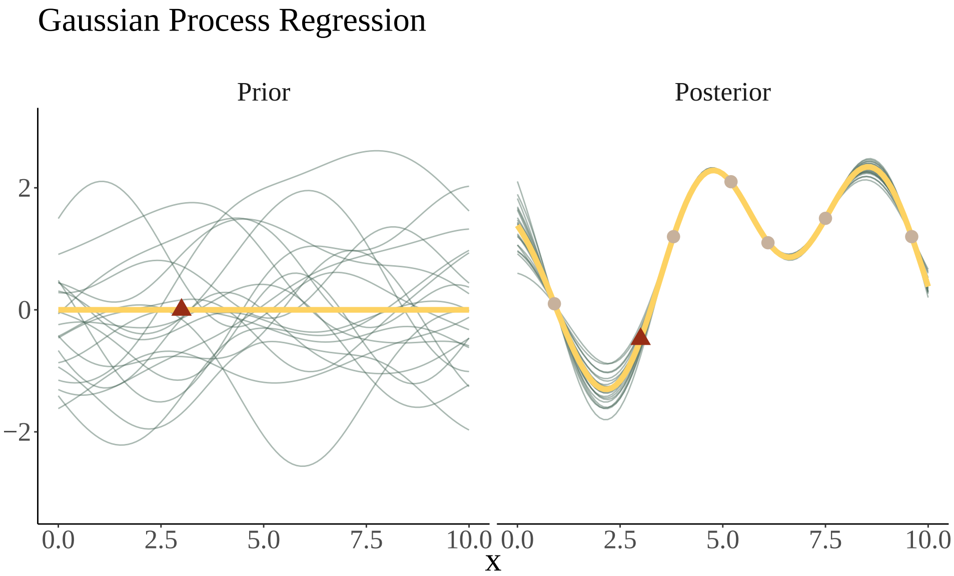


Figure 3.3: Green lines indicate predictions for different sampled posterior weights. Dots mark empirical observations. Yellow lines mark the current mean of the GP (λ was set to 0.5). The red triangle shows the prediction for the newly presented stimuli $x_\star = 3$.

Notice how Gaussian Process regression does not predict the same response for $x_\star = 3$ as for the closest other observed value. Instead, it generalizes to this point by using the underlying kernel mapping the distance of the newly observed input points and all other points to an expectation of an underlying functional value.

3.3.3 Switching back to the weight view

The mean function (3.20) can also be rewritten as

$$m_t(\mathrm{x}) = \sum_{i=1}^{t} w_i k(\mathrm{x}_i, \mathrm{x}) \tag{3.22}$$

where each x_i is a previously observed input value in X_t and the weights are collected in the vector $\mathrm{w} = (K(\mathrm{X}_t, \mathrm{X}_t) + \sigma^2 \ \mathrm{I})^{-1} \mathrm{y}_t$. What this equation tells us is that Gaussian Process regression is equivalent to a rule-based regression model using basis functions k to project the inputs into a feature space. To make new predictions, every output y_t is weighted by how similar the corresponding input x_t was to the to be predicted point x by a similarity measure induced by the kernel. This results in a simple weighted sum to make predictions for new points. The posterior predictive mean then is a linear combination of the features. This means that a conceptually infinite parameter space boils down to a finite sum when making predictions. This sum only depends on the chosen kernel k and the data $\mathcal{D}_t$ observed thus far (Kac & Siegert, 1947). More importantly, this means that a specific gradient of functional generalization can be induced by an underlying kernel to define a regression that is both a rule-based and a similarity-based model of function learning. The shape and form of the kernel then directly implies a particular gradient of generalization. Thus, Gaussian Process regression offers a unifying bridge between the two psychological accounts of function learning. Moreover, this means that assessing which kernel (or process of kernel construction) describes human generalization will provide insights about psychological rules of generalization more broadly.

Details for generating a prediction for $x_\star = 3$ given a radial basis function kernel with length scale $\lambda = 1$, and observation variance $\sigma_\varepsilon^2 = 0.01$ are provided in Table 3.2.

Table 3.2: Example of generating a prediction using a Gaussian Process with a radial basis function kernel. $w_i = (K(X,X) + \sigma_\varepsilon^2 I)^{-1} y_i$; $x_\star$=3;

t	x_t	y_t	w_t	$k(x_t, x_\star)$	$w_t k(x_t, x_\star)$
1	0.9	0.1	0.51	0.38	0.19
2	3.8	1.2	-3.88	0.87	-3.37
3	5.2	2.1	13.3	0.34	4.53
4	6.1	1.1	-12.55	0.12	-1.48
5	7.5	1.5	5.83	0.01	0.06
6	9.6	1.2	-0.34	0.00	0.00
				$\sum_{t=1}^{6} w_t k(x_t, x_\star)$:	-0.06

3.3.4 Optimizing hyper-parameters

As the kernel normally contains hyper-parameters such as the length-scale λ and the signal variance σ_f^2, a point estimate of the hyper-parameters is frequently computed by maximizing the marginal (log) likelihood. This is similar to parameter estimation by maximum likelihood and is also referred to as type-II maximum likelihood (ML-II, cf Williams & Rasmussen, 2006).

Given the data $\mathcal{D} = \{\mathrm{X}, \mathrm{y}\}$ and hyper-parameters θ, the log marginal likelihood is

$$\log p(\mathrm{y}|\mathrm{X}, \theta) = -\frac{1}{2}\mathrm{y}^\top \mathrm{K}_y^{-1}\mathrm{y} - \frac{1}{2}\log|\mathrm{K}_y| - \frac{n}{2}\log 2\pi \tag{3.23}$$

where $\mathrm{K}_y = K(\mathrm{X},\mathrm{X}) + \sigma_\varepsilon^2 \mathrm{I}$ is the covariance matrix of the noisy output values y. The marginal log likelihood can be viewed as a penalized fit measure, where the term $\frac{1}{2}\mathrm{y}^\top \mathrm{K}_y^{-1}\mathrm{y}$ measures the data fit, that is how well the current kernel parametrization explains the dependent variable, and $\frac{1}{2}\log|\mathrm{K}_y|$ is a complexity penalization term. The final term $\frac{n}{2}\log 2\pi$ is a normalization constant. The marginal likelihood is commonly maximized through a gradient-ascent based optimization using the partial derivatives w.r.t. ϑ :

$$\frac{\partial}{\partial \vartheta_j} \log p(\mathrm{y}|X, \theta) = \frac{1}{2}\mathrm{y}^\top K^{-1}\mathrm{y} - \frac{1}{2}\mathrm{tr}\left(K^{-1}\frac{\partial K}{\partial \vartheta_j}\right) \tag{3.24}$$

$$= \frac{1}{2}\mathrm{tr}\left((\alpha\alpha^\top - K^{-1})\frac{\partial K}{\partial \vartheta_j}\right) \tag{3.25}$$

with $\alpha = K^{-1}\mathrm{y}$.

This method optimizes the gradient of generalization given some empirical observations. However, the goal of this thesis will sometimes be different from that. In particular, I will try to find a particular gradient, shape, or construction of a kernel to explain human behavior.

3.4 Encoding smoothness

The radial basis function kernel is a special case of a general class of kernel functions called the Matérn kernel. The Matérn covariance between two points with distance $\tau = |\mathrm{x} - \mathrm{x}'|$ is

$$k \quad (\tau) = \sigma^2 \frac{2^{1-\nu}}{\Gamma(\nu)} \left(\sqrt{2\nu}\frac{\tau}{\lambda}\right)^\nu \mathcal{K}_\nu \left(\sqrt{2\nu}\frac{\tau}{\lambda}\right) \tag{3.26}$$

where Γ is the gamma function, $\mathcal{K}_\nu$ is the modified Bessel function of the second kind, and λ and ν are non-negative covariance parameters with λ again encoding the gradient of smoothness as before. A $\mathcal{GP}$ with a Matérn covariance has sample paths that are $\nu - 1$ times differentiable. When $\nu = p + 0.5$, the Matérn kernel can be written as a product of an exponential and a polynomial of order p.

$$k_{p+0.5}(\tau) = \sigma_f^2 \exp\left(-\frac{\sqrt{2\nu}\tau}{\lambda}\right) \frac{\Gamma(p+1)}{(2p+1)} \times \sum_{i=0}^{p} \frac{(p+i)!}{i!(p-i)!} \left(\frac{\sqrt{8\nu}\tau}{\lambda}\right)^{p-i} \tag{3.27}$$

Here, p directly determines how quickly the covariance between two points thins out in dependency of how far the two points are apart. If $p = 0$, then this leads to the Ornstein-Uhlenbeck kernel

$$k_{0.5}(\tau) = \sigma_f^2 \exp\left(-\frac{\tau}{\lambda}\right), \tag{3.28}$$

which encodes the prior assumption that the function is extremely unsmooth (rough) and that observations do not provide much information about points that are close to the points observed so far. As $p \to \infty$ in the limit, the Matérn kernel becomes the radial basis function kernel introduced before. Figure 3.4 shows the gradient of functional generalization for both an Ornstein-Uhlenbeck (Matérn 1/2) and a radial basis function kernel. Whereas the correlation for the Ornstein-Uhlenbeck kernel thins out very quickly but has fat tails, the correlation for the radial basis function kernel does not thin out that quickly but has thinner tails.[‡].

Figure 3.5 shows prior and posterior samples for both the Ornstein-Uhlenbeck and the radial basis function kernel. Notice how the prior samples are a lot more "rugged" for the former and much smoother for the latter. How encoding different prior smoothness assumptions leads to different posterior samples after having observed the same set of points as before can also be observed. In particular, expecting very rough functions a priori leads

[‡] In fact, it can be shown that the radial basis function kernel's spectral density is normally distributed, whereas the density for the Ornstein-Uhlenbeck kernel is a Student's t-distribution, see also Rasmussen & Williams (2006).

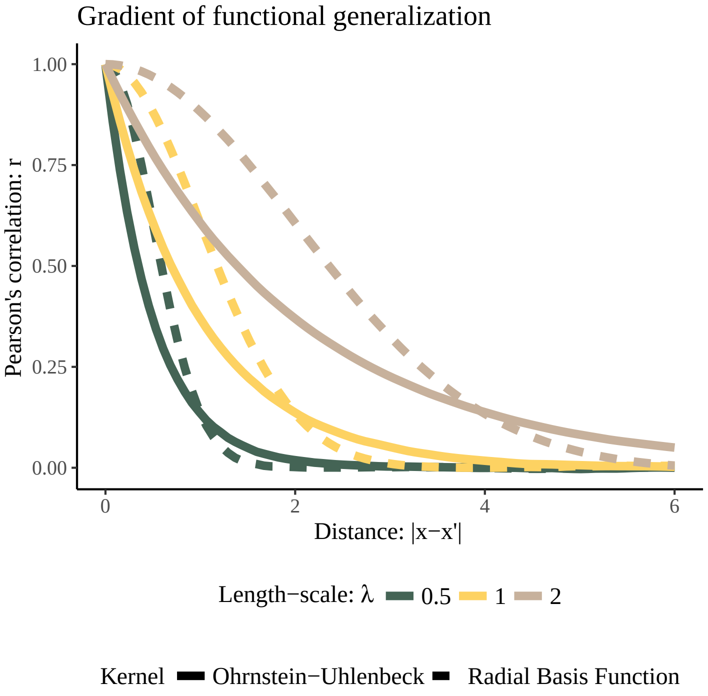


Figure 3.4: Different gradients of functional generalization as induced by different Matérn kernels. The correlation is assessed as the mean correlation between two points in dependency of their mutual distance. For bigger λ-parameters, the correlation decays more slowly leading to broader generalization.

to posteriors that do not generalize far beyond the encountered observations, whereas expecting smooth functions leads to posterior samples that generalize more broadly beyond the encountered points.

3.5 Composing kernels

So far, I have only talked about Gaussian Process regression models that generate predictions which reverse back to the overall mean as the distance to the observed set grows longer. Comparing these kind of kernels is similar to assessing different (more traditional) gradients of functional generalization. However, the assumption of decaying gradients might not

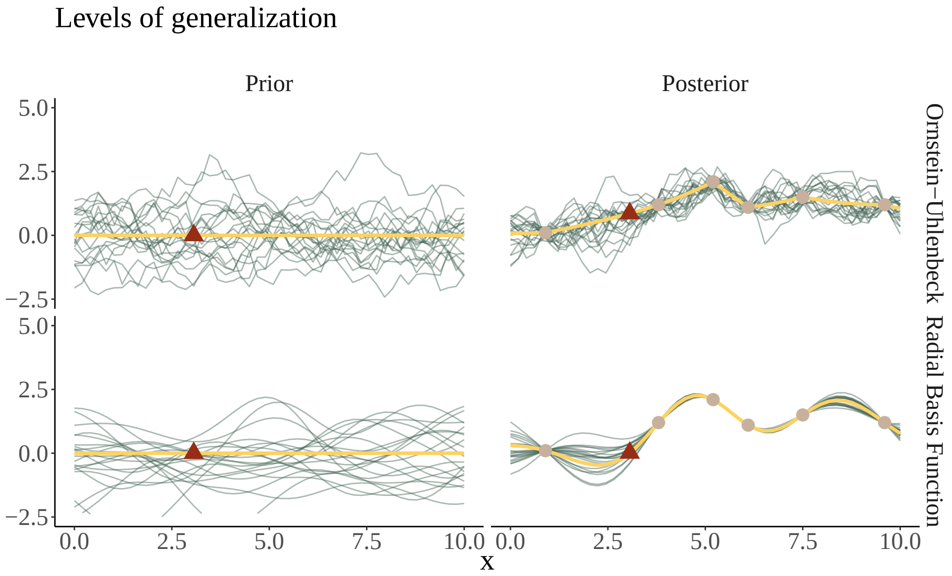


Figure 3.5: Example of samples from differently smooth Gaussian Process priors and their posteriors after having observed the same set of points. Grey lines indicate samples from the GP. Black dots mark empirical observations. The dark gray line marks the current mean of the GP. The red triangle shows the prediction for the new data point.

capture all phenomena of human function learning and generalization. First of all, as I have described before, subjects tend to extrapolate in a linear fashion which does not correspond to correlations between points that thin out over longer distances. Secondly, another observation comes from the previously mentioned study by DeLosh et al. (1997). Therein, not all participants extrapolated an underlying quadratic function as shown in Figure 2.3. Some participants also extrapolated this function as shown in Figure 3.6.

Even though these participants were classified as not extrapolating the function correctly, they nonetheless seem to have picked up a pattern, i.e. periodic extrapolation, a result similar to what was later found by Kalish (2013). Thus, a truly unifying model of generalization should also be able to capture these more elaborate patterns of generalization.

An advantage of Gaussian Process regression is that different kernels can be combined, thereby creating a rich set of interpretable and reusable building blocks (Duvenaud et al.,

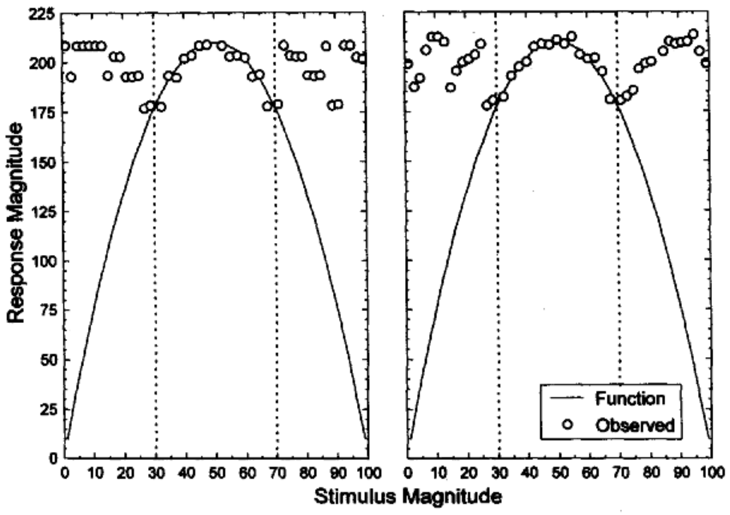


Figure 6. The transfer predictions of individual participants in the quadratic function condition who did not extrapolate in either extrapolation region.

Figure 3.6: Results of extrapolation experiment conducted by DeLosh et al. (1997), possibly showing periodic extrapolation.

2013). For example, adding two kernels together models the data as a superposition of independent functions, while multiplying a kernel with the radial basis function kernel, locally smooths the effect of the first kernel.

Figure 3.7 shows a data set of atmospheric concentration of carbon dioxide over a forty year horizon. This is a frequently use data set for structure discovery algorithms and it is immediately clear that there is a pattern within this data. The CO2-concentration seems to increase over the years, there seems to be some periodicity by which at some times within each year the CO2 emission is higher, and this period may not be perfectly replicated every year. Using a Gaussian Process regression framework, different kernels can be combined as building blocks in the attempt to explain these patterns. Figure 3.7 shows the posterior predictive mean of different kernel compositions. The first one shows a radial basis function kernel alone, the second a sum of a radial basis function kernel and a linear kernel, $k(x, x') =$

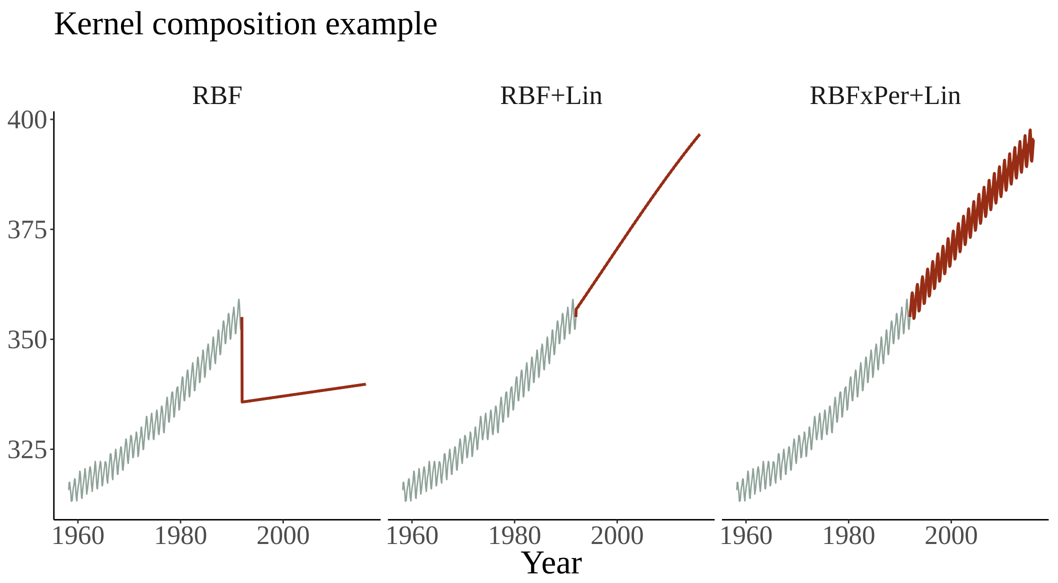


Figure 3.7: Example of composing kernels by combining simpler kernels in order to explain a complex function. Data were mean-centered before fitting the Gaussian Process and predictions were transformed back afterwards. Grey lines show observed CO2 emissions. Red lines show posterior predictions of Gaussian Process regressions with different kernels: RBF is a radial basis function kernel, RBF+Lin is a kernel composed by adding a RBF and a linear kernel, RBF$\times$Per + Lin is a kernel composed by multiplying a radial basis function and periodic kernel and adding a linear kernel.

$(x-c)(x'-c)$, and the third one the sum between a linear kernel and the product between a radial basis function kernel and a periodic kernel, $k(x,x') = \sigma_f^2 \exp\left(-\frac{2\sin^2(\pi|x-x'|)}{\lambda^2}\right)$. As the radial basis function kernel tends to reverse back to the mean over time, it does not capture the linear trend of the data. Therefore, adding a linear kernel to the radial basis function kernel already improves predictions. Finally, multiplying the radial basis function kernel with a periodic kernel to create a locally smoothed periodic kernel, which is then combined with an increasing trend by adding a linear kernel seems to predict the data best. This shows that the kernel can also be used to encode structural assumptions about the underlying function more explicitly, especially when one wants to explain more complex patterns than just generalizing over smooth functions (see Lloyd et al., 2014).

3.6 Exploration-Exploitation

I have discussed Gaussian Process regression as a tool to be applied in *single function learning* or *sequential function learning* settings. However, as mentioned before, the third variant of function learning, *self-directed function learning*, requires participants to direct their own sampling behavior (Gureckis & Markant, 2012). In particular, in an *function exploration-exploitation* scenario the goal is to find the input to a function that produces the maximum output

$$\mathrm{x}_{\max} = \arg\max_{\mathrm{x} \in D} f(\mathrm{x}) \tag{3.29}$$

as quickly as possible. where $\mathrm{x}_{\max}$ is the input that produces the highest output. Exploration here means to sample from more uncertain parts of the input space in order to gain more knowledge about the function that can be useful later on, whereas exploitation means choosing inputs that are likely to produce high outputs. One way to measure the quality of learning in a function exploration-exploitation scenario is to quantify regret (Bubeck et al., 2012). Regret is the difference between the output of the currently chosen argument and the best output possible, i.e. how much on every trial you could have chosen better

$$r(\mathrm{x}) = f(\mathrm{x}_{\max}) - f(\mathrm{x}). \tag{3.30}$$

The total regret is the sum of the regret over all trials, and the goal in an exploration-exploitation scenario is to minimize the cumulative regret:

$$R_t = \sum_{u=1}^{t} r(x_u) \tag{3.31}$$

It can be shown that finding a strategy that chooses the inputs to minimize the expected cumulative regret is NP-hard (Krause & Guestrin, 2005). That is, determining the sequence of queries (i.e. input choices) that leads to the lowest total regret is impossible for all but the most trivial cases. However, there is a greedy trick one can apply in this scenario, which starts by reinterpreting the function maximization – or regret minimization – problem as a multi-armed bandit task (see Steyvers et al., 2009). In a bandit task, there are multiple options (arms) with unknown probability of producing a reward and the goal is to choose the best arm in order to maximise the overall reward (the name stems from the one armed bandits that can be found in casinos). In the current situation, we can view the discretized input points as the arms of a multi-armed bandit, and the output of the function at those points as the unknown rewards that are associated to each arm. What distinguishes this situation from traditional bandit tasks is that the rewards of the arms are correlated in dependency of the underlying covariance kernel. Nevertheless, viewing the task as a multi-armed bandit allows to use sampling strategies that have been devised for traditional bandit tasks. One popular sampling strategy is called the upper confidence bound (UCB) algorithm. This algorithm relies on the following sampling strategy:

$$UCB_t(\mathrm{x}) = \mu_{t-1}(\mathrm{x}) + \beta\sigma_{t-1}(\mathrm{x}), \tag{3.32}$$

where β is a free parameter that determines the width of the confidence interval, $\mu_{t-1}(\mathrm{x})$ and $\sigma_{t-1}(\mathrm{x})$ are the predictive mean and standard deviation at a point x. For example, setting $\beta = 1.96$ results in a 95% confidence interval for a single value x given a Gaussian distribution. The UCB algorithm chooses the arm for which the upper confidence bound is currently the highest. The upper confidence bound is determined by two factors: the current estimate of the mean of *f* at a particular point (the higher the estimate, the higher the bound) and the uncertainty attached to that estimate (the higher the uncertainty, the higher the bound). Therefore, the UCB algorithm trades off naturally between expectation and uncertainty, that is exploitation and exploration. An example of how the UCB algorithm works, using the same data as before, is shown in Figure 3.8.

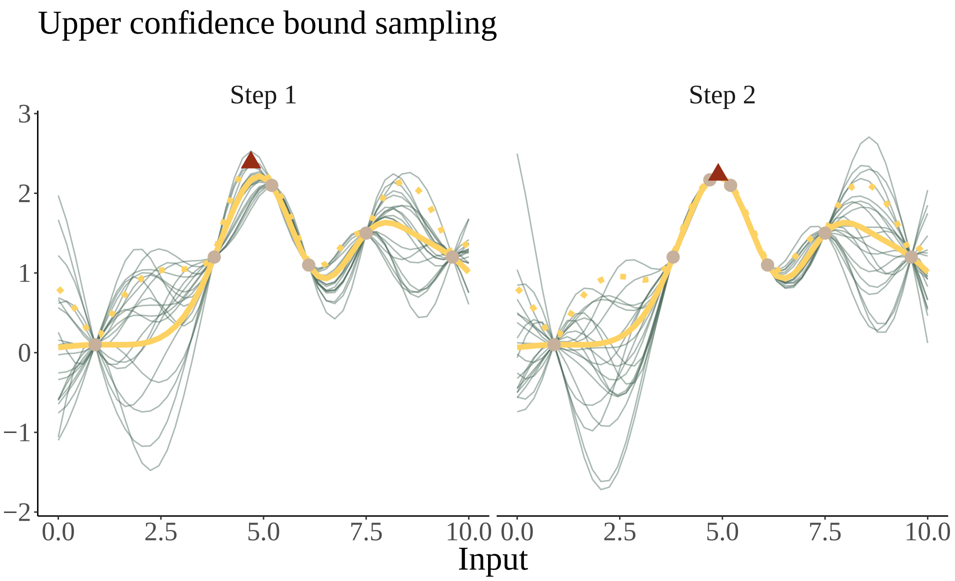


Figure 3.8: GP-UCB example. The yellow line marks the current mean of the GP. The dashed line marks the GP's upper confidence bound. The light green lines are samples from the GP. The red triangle marks the point that currently produces the highest UCB.

Even though the greedy UCB strategy is optimistically naïve (i.e., it inflates expectations by their uncertainties), it can be shown that its regret is sublinear, using an argument that relies on the submodularity and monotonicity of the overall information gain (Srinivas et al., 2010). Sublinear regret means that the regret per round goes down in expectation, thereby guaranteeing that the algorithm picks better points over time.

The Gaussian Process upper confidence bound sampling strategy is only one out of many Bayesian optimization sampling strategies (Snoek et al., 2012; Brochu et al., 2010). All of these optimization routines essentially use Gaussian Process regression as a model to represent and generalize the underlying function but apply different sampling strategies, that is different ways to map current expectations and their attached uncertainties, onto choices. Therefore, functional generalization is also important when searching for rewards that can be predicted by context or other features.

3.7 Past and present research utilizing Gaussian Process regression

Gaussian Process regression provides a concrete language for constructing and comparing priors over functions. As priors over functions can be explicitly encoded by the kernel, this means that the kernel can be used to describe expectations about functional regularities (e.g., linear trends, periodic patterns, etc.), thereby creating inductive biases that make generalization easier.

Previous work has begun to study what kind of kernels might describe human function learning and generalization. One experimental technique, known as *iterated learning*, estimates priors by simulating a Markov chain across multiple people—essentially a "game of telephone". Under certain assumptions, the Markov chain is guaranteed to converge to a stationary distribution, such that –asymptotically– it will generate samples from the prior (Griffiths & Kalish, 2007). In a function learning version of this task, participants see a set of points, which they then have to remember and re-draw. The resulting points are presented to another participant, who is asked to repeat the same procedure. Iterating this procedure multiple times will reveal participants' priors over functions. Kalish et al. (2007) showed that participants consistently converged to linear functions with a positive slope, even when the starting points came from a linear function with a negative slope, quadratic functions, or when they were generated at random.

Lucas et al. (2015) were the first to propose Gaussian Process regression as a model of human function learning. In their seminal work on Gaussian Process-based function learning, they explained the iterated learning results by postulating a GP with a mixture of experts kernel that is mostly dominated by a positively linear kernel generating functions with a linear upwards trend, but can also generate smooth non-linear extrapolations by utilizing a non-linear radial basis function kernel. Additionally, their proposed kernel contained negatively linear and quadratic components. Samples from the GP parameterized in this way tend to be linear lines with a positive slope. This parametrization was able to explain most of the findings explained in Chapter 2, such as the partitioning of the learning space, the mixture-like patterns of extrapolations, and the difficulty of learning some functions (e.g., exponential patterns) relative to others (e.g., linear patterns).

Going further, Wilson et al. (2015) attempted to infer the “human kernel” by having participants generate extrapolations for different functions sampled from a radial basis kernel and fitting a non-parametric kernel (the spectral mixture defined by Wilson & Adams, 2013, which I will describe in detail in Chapter 5) to their extrapolations. This approach is actually quite close to the attempt of finding a functional gradient of generalization by reverse engineering what a gradient explaining human behavior looks like. The results showed that participants expected long-distance correlations between points. These correlations can be viewed as arising from a mixture between linear and radial basis components. In a second experiment, they showed that participants can effectively learn functions sampled from a mixture of a product of linear and spectral kernels. These results can be interpreted in two different ways. Either participants’ functional gradient of generalization is adaptive to the situation at hand or there is something beyond just a gradient guiding participants’ extrapolation judgments. I will further probe this difference in Chapter 5.

Borji & Itti (2013) were the first to compare human behavior to that of Gaussian Process-based function optimization models within a uni-variate search task in which participants had to optimize unknown functions. Their results showed that participants sometimes outperform Bayesian optimization algorithms but overall generate similar queries and show similar regret as a Gaussian Process upper confidence bound sampling algorithm.

3.8 The current proposal

To summarize, Gaussian Process regression as a model of generalization unites both rule-based and similarity-based accounts of function learning. The form and shape of the kernel of a Gaussian Process regression directly corresponds to assumptions about laws of generalization which might describe human behavior. My previous definition of generalization as the ability to generate and act according to predictions for new observations based on underlying commonalities directly leads (as it has historically too) to studying the psychology of function learning as a framework to discover human laws of generalization. Provided that Gaussian Process regression is a powerful method to assess different kinds of function learning, I propose to use it to study generalization. What this will mean concretely, is that I will use Gaussian Process regression to model human function learning, assess what kind of kernels and kernel parameterizations describe and predict behavior best, and then use the empirical results as indicators of psychological rules. For this, Gaussian Process is a vessel providing a window into human generalization. Thus, my proposal is similar to common approaches of Bayesian cognitive science (Chater & Oaksford, 2008; Chater et al., 2006) in that I do not restrict myself to one particular proposal (for example, one particular rule of generalization) and then assess the resulting restricted set to produce theoretical claims. Instead, I used Gaussian Process regression as a unifying model of generalization that incorporates as many different candidate rules as possible and then infer which one of those best explains human generalization.

With the framework of Gaussian Process regression in hand, I will now start to assess generalization across various paradigms.

"Theorems deduced which concern phenomena not already known must be submitted to carefully controlled experiments."

Clark Hull, 1935

4

From theory to behavior

This chapter derives characteristics influencing GP learning curves and assesses if participants' perceived predictability of intuitive functions are influenced by the same factors.

4.1 Better noisy than complex?

Previous research on human function learning has mostly focused on either *single* or *sequential function learning* tasks. Yet, there might be other aspects of function learning that can be derived from a formal theory of generalization.

Imagine having to pick between two different learning situations. The first one is relatively complex but not so noisy and the latter is relatively simple but noisy. Which one would you prefer Put differently, do you think it is easier to adapt to a system that is complex but not very noisy or to a system that is more noisy but in which observations generalize more broadly?

As it turns out, the theory of Gaussian Process regression can provide some guidance on what a sensible decision maker would prefer. Even though this guidance is not necessarily the only possible answer to this problem, it can nonetheless be informative about whether it is easier to learn in a simple yet noisy or a complex but deterministic system. Moreover, predictions that are directly derived from statistical learning theory can in this case be compared to participants' judgments about how well they think they can predict presented functions.

In order to asses this, I will first introduce the concept of Gaussian Process learning curves, which theoretically relate the expected generalization error of newly sampled input points to a function's smoothness (which in this case is just another measure of complexity), its attached noise, as well as the number of sample points previously observed. I will show that these theoretically derived learning curves predict that a rational learner should prefer to reduce complexity over noise, or taking the example from above, would rather choose to work for a noisy but currently simple start-up than for a deterministic but complex bank.

Afterwards, I will assess these predictions empirically by eliciting participants' predictability judgments, that is asking them how well they think they could generalize from given points sampled from different functions, to predict newly sampled points. I find that their judgments align well with the theoretical predictions derived from Gaussian Process learning curves.

4.2 Gaussian Process learning curves

Theoretically, learning curves relate the expected generalization error of a model to the amount of training data (Opper & Vivarelli, 1999; Williams & Vivarelli, 2000; Sollich & Halees, 2002). They can be seen as a mathematical expression of a function's predictability, given assumptions about Gaussian Process-based priors over functions, the noise process, and the distribution of inputs. Intuitively, a function exhibits a higher predictability if it is easier to predict new input points that are randomly and uniformly chosen from the input space. If points are easier to predict, then the generalization error is lower as predictions will be closer to the underlying truth of the the actual function. Therefore, these two things, the predictability and the generalization error of a function, are two sides of the same coin (see also Goerg, 2013).

The learning curves for Gaussian Process regression can be used to derive *a priori* predictions about how different properties of functions such as smoothness, variance, and sample size influence their perceived predictability. In particular, factors that increase the generalization error should also lead to lower predictability judgments.

Given a dataset $\mathcal{D} = \{\mathrm{x}, \mathrm{y}\}$ and an error function $\mathcal{L}(\cdot, \cdot)$ which measures the difference between the predicted and true function values, the data-dependent generalization error is defined as the expected error on a test input $x_\star$ sampled from a uniform distribution, marginalizing over the latent function:

$$E(\mathrm{x}) = \int_f p(f) \int_{x_\star} \mathcal{L}(f(x_\star), \bar{f}(x_\star))\, dx_\star\, df, \tag{4.1}$$

where

$$\bar{f}(x) = \mathbb{E}[f(x_\star)|\mathrm{x}, \mathrm{y}]. \tag{4.2}$$

It is called the *data-dependent error* as it still depends on the position of the observed input points. The data-independent generalization error is defined as the expectation of $E(\mathrm{x})$ with respect to a density $p(\mathrm{x})$ on inputs with sample size n:

$$\mathcal{E}(n) = \int_{\mathrm{x}} p(\mathrm{x}) E(\mathrm{x})\, d\mathrm{x}. \tag{4.3}$$

It is called the *data-independent error* as it does not depend on the observations directly, but rather provides an *a priori* expectation of the error after n sample points have been observed.

A learning curve is constructed by calculating the data-independent generalization error as a function of the sample size. While the learning curve is not analytically tractable (except for a few special cases), it is possible to derive a lower bound using the eigenfunction expansion of the covariance function:

$$k(x, x') = \sum_i \lambda_i \varphi_i(x) \varphi_i(x), \tag{4.4}$$

where $\{\lambda_i\}$ is the spectrum of eigenvalues (decreasing as a function of i) and $\{\varphi_i(x)\}$ are the eigenfunctions.

As shown by Opper & Vivarelli (1999), the generalization error for the squared-loss error function,

$$\mathcal{L}(y, \hat{y}) = |y - \hat{y}|^2, \tag{4.5}$$

can be lower-bounded by:

$$\mathcal{E}(n) \geq \sigma_\varepsilon^2 \sum_{i=1}^{N} \frac{\lambda_i}{\sigma_\varepsilon^2 + n\lambda_i}, \tag{4.6}$$

where N is the number of non-zero eigenvalues. Loosely speaking, the eigenvalue spectrum summarizes how the correlation between the function values of two points changes as a function of their input distance, a mathematical translation of the previously shown functional gradient of generalization.

Smoother functions have eigenvalues that decay more slowly across the spectrum. Smooth functions have long-distance correlations, which makes it easier to learn and therefore leads to smaller generalization errors (as also shown in Figure 3.5).

The theoretical predictions of learning curves can be seen even more clearly when looking at covariance functions with power-law spectral decay[*] (Sollich & Halees, 2002):

$$\lambda_i \propto i^{-r}. \tag{4.7}$$

Asymptotically ($\mathcal{E}(n) \ll \sigma_\varepsilon^2$), the learning curve scales as

$$\mathcal{E}(n) \propto \left(\frac{\sigma_\varepsilon^2}{n}\right)^{-\frac{r-1}{r}}. \tag{4.8}$$

This analysis shows that the most important factor influencing a Gaussian Process' learning curve is the smoothness of the covariance function (parametrized in terms of the spectral decay rate *r*). The generalization error depends polynomially on the variance but exponentially on smoothness. The implication is that noisy but smooth functions are easier to generalize and therefore to predict than deterministic but complex functions. This is intuitive, because smooth functions allow data to be more strongly aggregated across different input points (observing the function at one point provides more information about other points as seen in Chapter 3), whereas anything one can learn about a complex function is very local. Thus, the approximated learning curves predict that participants should choose simple but noisy over complex but less noisy function learning scenarios.

To create functions with different levels of smoothness, a flexible class of stationary covariance functions constructed from the modified Bessel function (Rasmussen & Williams, 2006) can be employed: the Matérn class introduced in the previous chapter. Let me briefly restate the Matérn class of covariance function for $\tau = |x - x'|$ here:

[*] The one-dimensional Ornstein-Uhlenbeck covariance function has this property, with $r = 2$.

$$k_\nu(\tau) = \sigma_f^2 \frac{2^{1-\nu}}{\Gamma(\nu)} \left(\sqrt{2\nu}\frac{\tau}{\lambda}\right)^\nu \mathcal{K}_\nu \left(\sqrt{2\nu}\frac{\tau}{\lambda}\right) \tag{4.9}$$

where $\lambda > 0$ is a length-scale parameter, $\mathcal{K}_\nu(\cdot)$ is the modified Bessel function of order $\nu = p - 0.5$ for integral p, and $\Gamma(\cdot)$ is the gamma function. I will refer to p as the order of the Gaussian Process.

When $p = 1$, Eq. 4.9 corresponds to the Ornstein-Uhlenbeck covariance function, which generates a process that is not mean square differentiable (see Rasmussen & Williams, 2006). This means that sampled functions will produce very rough outputs. When $p > 1$, the process is $p - 1$ times mean square differentiable, becoming smoother with increasing p. In the limit $p \to \infty$, it is equivalent to a radial basis function kernel. I will use Matérn kernel functions of order up to 3 here as it is empirically hard to distinguish between functions of higher order (see Cornford et al., 2002, for further discussion).

With these theoretical predictions in hand, I will now turn to an empirical exploration of the formally derived factors influencing a function's perceived predictability.

4.3 Experiment 1: Perceived predictability of functions

Participants will be asked to judge the predictability of provided functions (displayed as scatter plots), while the underlying smoothness, noisiness, and sample size are manipulated. This allows us to quantitatively measure the influence of these different factors on perceived predictability. Based on the analysis of learning curves described before, I derive the following 3 hypotheses:

1. Sample size and smoothness will correlate positively with perceived predictability, whereas noise variance will correlate negatively.
2. The effect of smoothness will be stronger than the effects of noise and sample size.

3. The approximate learning curve given a sample will be the most important factor influencing participants' predictability judgments.

4.3.1 Participants

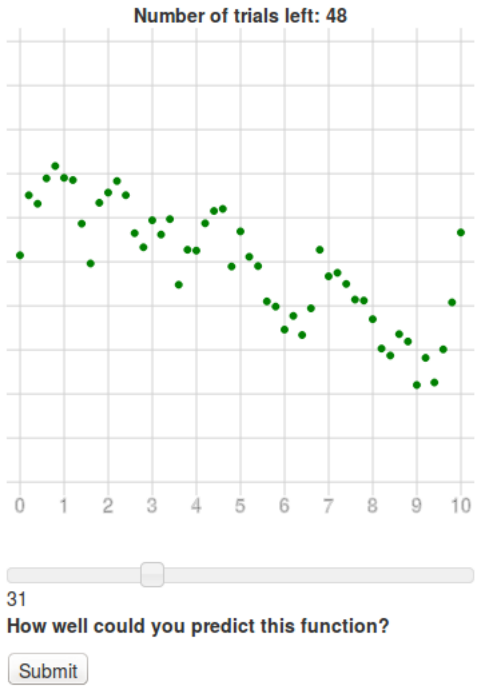
47 participants were recruited via prolificacademic.co.uk and received £1 for their participation. 27 participants were female and the overall age had a mean of 24 with a standard deviation of 5. The experiment took 8 minutes on average to be completed.

4.3.2 Task

Participants were told they had to assess how well they could potentially predict different functions on a scale from 0 (not at all) to 100 (certainly). It was explained in detail to them that prediction means to assess a new input point uniformly sampled from the input range. After passing three comprehension check questions, participants were sequentially shown 50 different samples of functions and had to indicate how well they thought they could predict a newly sampled point of that function. A screenshot of the experiment is shown in Figure 4.1.

4.3.3 Design

Participants went through 50 sequential trials in total where points were sampled equidistantly from different GPs with the Matérn covariance function. The parameters for the smoothness $p = [1, 2, 3]$, the variance $\sigma^2 = [0, 0.05, 0.1, 0.15, 0.2]$, and sample size $n = [10, 20, 30, 40, 50]$ were randomly selected on each trial. As Gaussian Process samples were



Number of trials left: 48

How well could you predict this function?

Figure 4.1: Screenshot of the predictability experiment. Dots were sampled uniformly from a function whose order, variance, and sample size was sampled prior and unknown to participants. Participants had to indicate how well they thought they could predict this function on a scale from 0 (not at all) to 100 (certainly).

created on the spot in JavaScript[†], so no participant saw exactly the same function; only the different characteristics governing the generating process were manipulated. The length-scale of the provided covariance function was fixed to $\lambda = 1$.

4.3.4 Results

Figure 4.2 shows the relationship between different Gaussian Process parameters and participants' perceived predictability.

It can be seen there that increasing smoothness results in higher perceived predictabil-

[†] I created a JavaScript library to sample from different Gaussian Process priors at the client's site, available at github.com/ericschulz/gpsmooth.

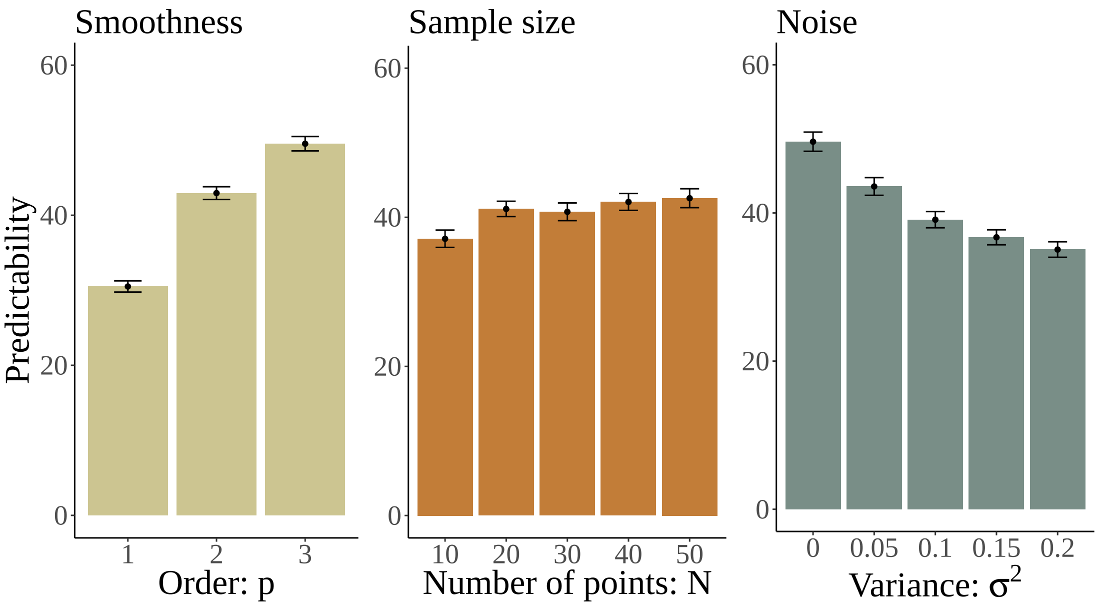


Figure 4.2: Participants' perceived predictability in dependency of different levels of smoothness, noise, and sample size. Error bars represent the standard error of the mean.

ity, and increasing noise variance reduces perceived predictability. The overall effect of an increasing sample size on the perceived predictability seems relatively small. This finding has at least two potential explanations. First, it might be the case that participants overestimate the predictability with small sample sizes as they tend to infer smoother functions than the ones generating the data. Secondly, the equidistantly spaced inputs I presented to participants might permit easier prediction, since they cover more space overall. In order to quantitatively assess the influence of the different factors on participants' perceived predictability, I performed a mixed-effects regression with judgments nested within participants. The parameter estimates of the fixed effects are summarized in Table 4.1.

In agreement with the above qualitative characterization, both smoothness and noise variance have a significant effect on predictability. The effect of sample size, on the other

Table 4.1: Parameter estimates of factors influencing predictability judgments extracted from a mixed-effects regression analysis. Conditional pseudo-$r^2 = 0.30$.

	Estimate	s.e.	t-value	Pr(>|t|)
Intercept	41.70	1.82	22.93	0.00
n	0.79	1.02	0.77	0.44
p	8.00	0.72	11.11	0.00
σ^2	-5.21	0.66	-7.90	0.00
$n \times p$	1.48	0.38	3.90	0.00

hand, is not significant. Nonetheless, sample size interacts significantly with smoothness: increasing sample size reduces the effects of smoothness on predictability judgments. Thus, sample size does appear to be a modulator of perceived predictability. Therefore, hypothesis 1 can only be seen as partially confirmed.

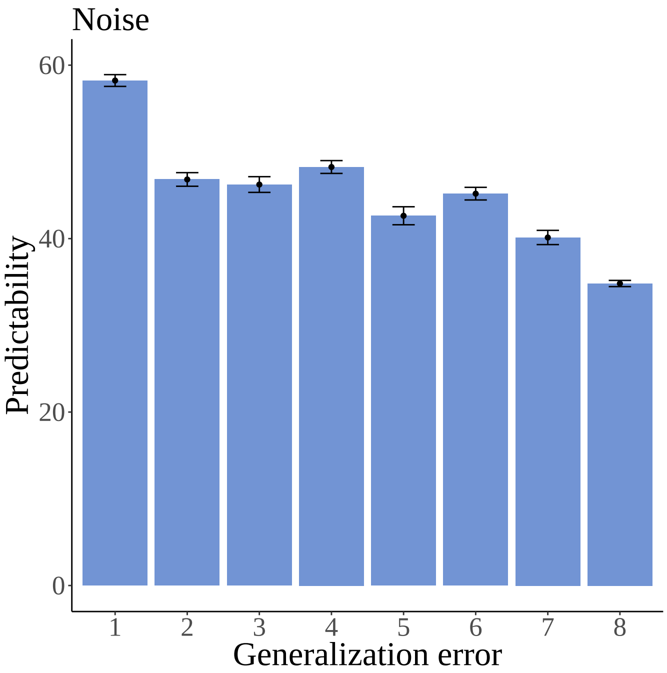


Figure 4.3: Perceived predictability as a function of theoretical generalization error. Derived generalization error has been distributed across 8 equally-sized bins.

Next, I calculated the correlations between each of the different factors and the predictability judgments for each participant individually and found that the averaged correlation between smoothness and perceived predictability ($r = 0.36, p < 0.01$) was indeed greater than the correlation between noise and perceived predictability ($r = -0.24$, $p < 0.01$) and between sample size and perceived predictability ($r = 0.06, p > 0.05$). Therefore, the second hypothesis was confirmed.

Finally, I examined whether the theoretical learning curve provides an accurate quantitative model of perceived predictability. As shown in Figure 4.3, the theoretical generalization error (binned for the plot, but not for the analysis) is a modest but significant predictor of perceived predictability ($r = -34, p < 0.01$). On the one hand, this suggests that the model does not capture all of the essential determinants of predictability perception. On the other hand, however, a correlation coefficient essentially assumes a linear dependency between the two factors, whereas –theoretically– the effect of learning curves on judgments can be expected to be nonlinear. Therefore, the analysis above was repeated, but this time using an importance measure derived by a random forest random permutation method (Archer & Kimes, 2008). Although this method is not often applied in psychology, it is frequently found in other areas such as genetics and biostatistics (Boulesteix et al., 2012). The random forest importance measure assesses for each entered variable separately by how much the predictive performance of a random forest regression decreases if this factor is randomly permuted and thereby rendered uninformative. The results of this procedure are also displayed in Table 4.2 and show that the importance of the theoretically derived prediction error is indeed higher than that of the other three variables. Thus, the last hypothesis can be confirmed within limited constraints.

Table 4.2: Average correlation between factors and participants' predictability judgments r and importance as assessed by a random forest permutation algorithm.

Variable	Average r	Importance
n	0.06	8.04
σ^2	-0.24	8.25
s	0.36	10.02
$\mathcal{E}(n)$	-0.34	11.58

4.4 Discussion

This investigation of predictions directly derived from Gaussian Process learning theory shows that theoretical predictions about a function's generalizability provide a framework for understanding participants' perception of a function's predictability. This model captures qualitative effects of a function's smoothness, noise variance, and sample size on the generalization error, which is essentially the the inverse of predictability. The smoothness of a function exerts a stronger influence on predictability than noise or sample size, consistent with both theoretically derived learning curves and the experimental data. This means that a smooth but noisy function is perceived as more predictable than a complex but near-deterministic function. Thus, it is –at least according to this theory– easier to learn in a noisy yet simple than a deterministic but complex system; generalizability is preferred over determinism.

Furthermore, I have shown that the model can quantitatively capture participants' predictability judgments, although it still leaves a fair amount of variance unexplained. One reason for the relatively low correlation between generalization error and predictability might be that the analysis assumed the covariance function is known. If participants use a different covariance function, this will change the form of the learning curves (although the qualitative predictions of the results reported here would remain the same; see Sollich, 2005). In future work, it will be worthwhile to explore models that learn the covariance function parameters directly.

The answer to the question of how people perceive the predictability of a function is probably more nuanced than the account I have put forward here. Statistically speaking, the ability to learn a function of a given complexity improves with increasing sample size, and thus for a given level of complexity there is a sample size at which –intuitively– one would judge a function to be highly predictable. This means that, for example, the effect of smoothness on predictability might be relatively weaker than what I have found when tested at much larger samples sizes. At different levels of complexity and sample size, the human perception of the predictability of functions, as well as which factors drive that predictability, may vary.

Another drawback of the experiment presented here is that participants were never actually asked to generate predictions for a presented function, but only had to judge their perceived predictability. As elicited judgments about one's skills do not always fully correspond to one's actual abilities (Paulhus et al., 1998), the proposed measure of predictability will have to be validated further by letting participants choose between different functions and then asking them to generate predictions for newly observed points directly. This will bring predictability even closer to traditional approaches of experiments on human function learning (DeLosh et al., 1997). Unlike previous work on function learning, which has focused on interpolation and extrapolation performance, the work explored here represents a relatively novel facet of participants' perception of functions, the way in which they think a given function is predictable. I believe that this assay provides another rich source of information about function knowledge.

A different direction for future research is manipulating the way in which input points are sampled. One major difference between my current implementation of assessing participants' predictability and the theoretically-derived generalization error, is that I have sampled points equidistantly, whereas theoretically input points are assumed to either be sampled uniformly or normally at random. That points were displayed equidistantly could have caused the relatively small effect of sample size on predictability judgments as participants might have been more able to generalize functions between even a small number of equidistantly presented points. Thus, future experiments could try to assess how judgments differ if points are sampled from other distributions. Also, learning curves change as a function of whether inputs are sampled randomly or using self-directed exploration (Ritter, 2007).

While I have only considered a fairly simple family of covariance functions, evidence suggests that people can generalize over richer representations; for example, a single function may be partitioned into several different sub-functions (Kalish et al., 2004). It is possible to take this one step further and ask whether a functional gradient of generalization is indeed all there is to function learning or if perhaps it is made of simpler building blocks using a "function grammar" (Duvenaud et al., 2013). I will address this question in the next chapter.

“It is the task of our science to find the elements of thought and the rules they adhere to, how they connect.”

Wilhelm Wundt, 1896

5

A compositional theory of generalization

This chapter develops a compositional theory of function learning and tests this theory within pattern completion, predictability, numerosity, change detection, and working memory tasks.

5.1 Introduction

As machine learning algorithms get better across many tasks such as visual pattern classification (Krizhevsky et al., 2012), game playing (Silver et al., 2016), natural language understanding (Collobert et al., 2011), and image generation (Goodfellow et al., 2014), a main question becomes if there are still psychological findings and theories from which current algorithms could potentially benefit (Lake et al., 2016).

One frequently mentioned and almost indisputable area is people's ability to generalize broadly beyond encountered examples (Schmidt, 2009). For example, when having to judge what kind of tower blocks are stable, a neural network trained on 4 blocks can achieve a level of performance better than most human subjects, but again performs at chance level as soon as 5 instead of 4 blocks are used (Lerer et al., 2016). Whereas deep reinforcement learning algorithms can learn to play many Atari games better than even professional human players, they need more than a billion training epochs, whereas the biggest chunk of human learning in such games (for experienced players) happens within only the first few games (Tsividis et al., 2017). Whereas generative adversarial networks can produce coherent sentences and text (Rajeswar et al., 2017), human language production capacities go far beyond this, making it possible to write poems, novels, or even a PhD-thesis. What enables us to generalize so quickly and broadly beyond the encountered data?

One promising answer to this question is *compositionality* (Goodman et al., 2008b), the observation that often times more complex relations can be broken up into simpler elements and –even more importantly– that simple elements and rules about how they can be combined generate a grammar that can potentially be infinitely large (Fodor, 2001). As an example of a successful application of the human ability of compositional generalization to a machine learning problem, consider Lake et al. (2015)'s research on the generation of different alphabets of characters. Their proposal is a Bayesian Program Induction framework (Lake et al., 2013) that learns simple stochastic programs to represent concepts, building them compositionally from parts, subparts, and spatial relations. This model defines a generative model that can sample new types of concepts by combining parts and subparts in new ways, essentially a compositional approach to character learning. Within a simple Turing-like task participants could not distinguish new characters generated by this model from characters generated by other humans, thereby demonstrating the full expressiveness of this approach. Yet, how can compositionality support human function learning and generalization? In what way, if any, is human function learning compositional?

This chapter probes compositionality as a mechanism for generalization in human function learning. Doing so, I will define a compositional grammar of Gaussian Process kernels and rules of how they can be combined. This approach is then tested against two other models of Gaussian Process structure learning, first within relatively simple pattern completion tasks, but later also in more elaborate scenarios such as change detection and working memory paradigms. Results show that humans do indeed seem to possess compositional inductive biases and can generalize, perceive, and memorize compositional patterns better than non-compositional ones.

5.2 Compositionality as a core principle of generalization

Humans are seemingly able to adapt to a wide variety of structural forms (Kemp & Tenenbaum, 2009; Gershman & Niv, 2010); the space of such forms is essentially unbounded, raising the question of how an infinite variety of forms can be represented. A tractable approach to accomplish this is to define a compositional system that can build complex structures through the composition of simpler elements. For example, in functional programming languages, functional primitives can be combined to create more complex functions which can then be re-combined to create even more complex functions (Peyton Jones, 1987). Via re-combinations, compositionality leads to a large increase of productivity—an infinite number of representations can be constructed from a finite set of primitives (Fodor, 1975; Fodor & Pylyshyn, 1988).

One source of evidence for compositionality in cognition comes from studies of rule-based concept learning. In these studies, the rules can be expressed as functions of logical primitives (e.g. Bruner et al., 1956; Shepard et al., 1961). By studying participants' mistakes and the relative learning difficulty of different concepts, researchers tried to unravel the primitives of symbolic thought and how these primitives are combined. This work has led to a rich set of theoretical ideas and empirical constraints on compositionality in concept learning (Piantadosi et al., 2016; Goodman et al., 2008a; Lake et al., 2015; Nosofsky et al., 1994).

Kemp (2012) provided an exhaustive characterization of compositionality in logical domains, showing how a "conceptual universe" can be formed by a rule inference scheme based on minimal description length that explains logical reasoning data across a wide set of domains. Recently, Piantadosi et al. (2016) showed how different sets of structural primitives, embedded in an approximate Bayesian inference framework, can predict distinct learning curves in rule-based concept learning experiments. Based on these and other studies, Lake et al. (2016) argued that compositionality should be a core principle of algorithms "that learn and think like people."

Given the widespread theoretical and empirical support for compositionality in cognitive science, it is natural to ask whether humans make use of compositionality in their ability of generalization. With a few exceptions (Gershman et al., 2017, 2015b), prior work on function learning with GPs has assumed a fixed, non-compositional kernel[*]. Thus, the goal of this chapter is to formalize a compositional approach to function learning and compare it to alternative non-compositional approaches. This comparison is especially interesting when considering different variants of generalization as on the one hand generalization could just be driven by an exponentially decaying gradient of functional generalization or, on the other hand, by a compositional grammar of smaller parts.

I will probe these differences by positing two candidate kernel parameterizations that express conceptually different inductive biases. The first two learn about and generalize pattern by approximating the gradient directly. The latter accomplishes generalization by composing parts of a functional grammar. I will then present a series of experimental tests that pit these kernels against each other. The aim is to see if explaining functional generalization requires more than functional gradients, and –if so– whether and how a compositional theory of generalization can fill this theoretical gap.

5.3 Generalization and structure learning with Gaussian processes

Broadly speaking, there are two approaches to parameterizing the kernel space: a fixed functional form with continuous parameters, or a combinatorial space of functional forms. These approaches are not mutually exclusive; indeed, the success of the combinatorial approach depends on optimizing the continuous parameters for each form. Nonetheless, this distinction is useful because it allows to separate different forms of functional generalization. A function might have internal structure such that when this structure is revealed, the apparent functional complexity is significantly reduced. For example, a function com-

[*] This is also true for approaches utilizing a mixture of expert kernel, which is fixed *a priori*, but see Kalish (2013) for further discussion.

posed of many piece-wise linear segments might have a long description length under a typical continuous parametrization (e.g., a radial basis function kernel), because it violates the smoothness assumptions of the prior. However, conditional on the change-points between segments, the function can be decomposed into independent parts each of which is well-described by a simple continuous parametrization (see Lee & Yuille, 2006, for a discussion of how this strategy is used by the brain in early vision). If internally structured functions are "natural kinds", then the combinatorial approach may be a good model of human generalization.

In the rest of this section, I describe three kernel parameterizations and their implications for human generalization. The first two are continuous, differing in their expressiveness. The third one is combinatorial, allowing it to capture complex patterns by composing simpler kernels.

5.3.1 Radial basis function kernel

As explained in Chapter 3, the radial basis function kernel is a commonly used kernel in many machine learning applications, embodying the assumption that the covariance between function values decays exponentially with input distance:

$$k(x, x') = \sigma_f^2 \exp\left(-\frac{|x - x'|^2}{2\lambda^2}\right), \tag{5.1}$$

where σ_f^2 is a scaling parameter and λ is a length-scale parameter determining the speed of the decay over the distance between inputs. The radial basis function kernel can be seen as a direct operationalization of (Shepard, 1987)'s universal rule of generalization within the function learning domain. It assumes that the same smoothness properties apply globally for all inputs and will provide a baseline to compare with more expressive kernels.

5.3.2 Spectral mixture kernel

The second approach is based on the fact that any stationary kernel can be expressed as an integral using Bochner's theorem. Letting $\tau = |x - x'| \in \mathbb{R}^P$, then

$$k(\tau) = \int_{\mathbb{R}^P} e^{2\pi i s^\top \tau} \psi(\mathrm{d}s). \tag{5.2}$$

where ψ is a positive finite measure. The density $S(s)$ of ψ is called the spectral density of k; k and S are Fourier duals:

$$k(\tau) = \int S(s) e^{2\pi i s^\top} \quad ds \tag{5.3}$$

$$S(s) = \int k(\tau) e^{-2\pi i s^\top} \quad d\tau \tag{5.4}$$

Therefore, every kernel can also be represented by a distribution over the spectral density space. Wilson & Adams (2013) showed that the spectral density can be approximated by a mixture of Gaussians. The spectral density modeled with a single Gaussian is

$$\varphi(s, \mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\{-\frac{1}{2\sigma^2}(s - \mu)^2\} \tag{5.5}$$

$$S(s) = \frac{1}{2}[\varphi(s) - \varphi(-s)] \tag{5.6}$$

The resulting kernel is given by

$$k(\tau) = \exp\{-2\pi^2\tau^2\sigma^2\} \cos(2\pi\tau\mu) \tag{5.7}$$

Extending this result to a mixture of Q Gaussians results in:

$$k(\tau) = \sum_{q=1}^{Q} w_q \prod_{p=1}^{P} \exp\{-2\pi^2 \tau_p^2 v_q^{(p)}\} \cos(2\pi \tau_p \mu_q^{(p)}) \tag{5.8}$$

where the qth component has the mean $\mu_q = (\mu_q^{(1)}, \ldots, \mu_q^{(p)})$ and covariance matrix $M_q = \text{diag}(v_q^{(1)}, \ldots . v_q^{(p)})$, where the inverse mean represents the component periods and the inverse standard deviation the length scales. The result is a non-parametric approach to Gaussian Process regression, in which complex kernels are approximated by mixtures of simpler ones. This approach is similar to assuming that participants learn about the gradient of functional generalization directly, by approximating the underlying correlational dependencies that have generated a given function. It is appealing when simpler kernels (e.g., the radial basis function) fail to capture functional structure. Its main drawback is that because structure is captured implicitly via the spectral density, the building blocks are psychologically less intuitive: humans appear to have preferences for linear (Kalish et al., 2007) and periodic (Bott & Heit, 2004) functions, which are not straightforwardly encoded in the spectral mixture (though of course a mixture can approximate these functions). Since the spectral kernel has been successfully applied to reverse engineer human kernels (Wilson et al., 2015), it is a useful reference of comparison to more structured compositional approaches.

5.3.3 Compositional kernel

As positive semi-definite kernels are closed under addition and multiplication, it is possible to create richly structured and interpretable kernels from well-understood base components. For example, by summing kernels, one can model the data as a sum of independent functions. Imagine a function that is linearly increasing over time but also shows some seasonal periodicity; then a combination of a linear and a periodic kernel added together might be a good description of that function (as described in Chapter 3).

Figure 5.1 shows an example of how different kernels (radial basis function, linear, periodic) can be combined. The approach put forward here, following Duvenaud et al. (2013), is to define a grammar over kernels that generates new kernels through summation or multiplication of simpler base kernels. Table 5.1 summarizes the kernels used in this grammar. Given a set of input-output pairs, the task facing the learner is to identify both the function and an underlying parse tree. As with the other kernel parameterizations, the parse tree can be chosen to maximize the marginal likelihood.

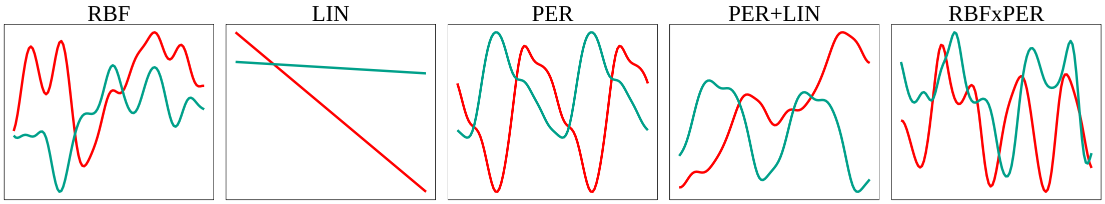


Figure 5.1: Examples of base and compositional kernels. The base kernels are radial basis function (**RBF**), linear (**LIN**), and periodic (**PER**); the composition operators are addition and multiplication. Adding a periodic and a linear kernel creates functions with trends and seasonality. Multiplying a periodic kernel with a radial basis function results in more localized periods than a standard periodic kernel would be able to capture.

Table 5.1: Base kernels in the compositional grammar.

Name	Definition
Linear	$k(x, x') = (x - \vartheta_1)(x' - \vartheta_1)$
Radial basis function	$k(x, x') = \sigma_f^2 \exp\left(-\frac{(x-x')^2}{\lambda^2}\right)$
Periodic	$k(x, x') = \sigma_f^2 \exp\left(-\frac{2\sin^2(\pi\lvert x-x'\rvert\theta)}{\lambda^2}\right)$

While other compositional grammars are possible –e.g., including a more diverse set of base kernels or other composition operators such as convolution or scaling– the proposed grammar is a useful starting point, since the components are intuitive and likely to be psychologically plausible. For tractability, the maximum number of combined kernels is fixed to be 3 and repetition of kernels are not allowed in order to restrict the complexity of the inference. The complete set of resulting kernels is shown in Table 5.2.

Table 5.2: Kernel combinations in the compositional grammar and their interpretations.

Combination	Interpretation
Linear	Linear function
Radial basis	Locally smooth function
Periodic	Repeated pattern
Linear + Periodic	Linear trend plus repeated pattern
Linear + RBF	Linear trend plus local deviations
RBF + Periodic	Repeated pattern plus local deviations
Linear × Periodic	Repeated pattern with increasing amplitude
Linear × RBF	Local deviations with increasing amplitude
RBF × Periodic	Slowly changing repeated pattern
Linear + RBF + Periodic	Linear trend plus local deviations plus repeated pattern
Linear + Periodic × RBF	Linear trend plus slowly changing repeated pattern
Periodic + Linear × RBF	Repeated pattern plus local deviations with increasing amplitude
Linear × RBF × Periodic	Slowly changing repeated pattern with increasing amplitude

I will use this set of kernels as the proposed compositional function class. Given a particular task, the specific composition from the set is chosen that maximizes the marginal likelihood.

Comparing these different kernels with each other will provide insights into the way people extrapolate from experience in general and learn about functions in particular. If a radial basis function kernel describes participants' function learning well, then research on generalization should focus on finding the exact gradient of generalization, i.e. estimating λ in different tasks. If the spectral density kernel describes function learning well, this could imply that participants learn about structure implicitly and that one would have to extract the human kernel over different function learning tasks, comparing how general the distribution of the spectral density is[†]. If the compositional kernel explains human function learning best, then this means that participants are able to generalize more broadly and efficiently than what was previously thought.

5.4 Experiment 2a: Pattern completions of compositional functions

The first experiment examines whether participants prefer compositional over non-compositional predictions of functions if the ground truth is indeed compositional. Even though this mainly reveals whether people make predictions in accordance with the underlying structure (i.e., it is not informative about inductive biases *per se*), Experiment 2b will examine the case where the ground truth is non-compositional. If people prefer compositions in both experiments, then that means that they have indeed strong compositional inductive biases for pattern completions.

I used a "pattern completion" paradigm, motivated by prior research on pattern perception as a window into cognitive representations (e.g., Buffart et al., 1981; Kanizsa, 1979). Participants chose among 3 different completions of a partial one-dimensional function.[‡] The candidate completions were generated by the different structure learning approaches described before. The main hypothesis was that if participants have structured, compositional representations of functions, then they should prefer pattern completions generated from the compositional kernel.

[†] This would also be an interesting approach towards assessing human generalization, for example by assessing the density for self-directed vs. simple function learning tasks.

[‡] This is essentially a form of extrapolation judgment, but unlike typical extrapolation paradigms that test input-output pairs one at a time, pattern completion asks participants to consider a set of input-output pairs.

5.4.1 Methods

Participants

52 participants (29 women) with an average age of 36.15 (SD = 9.11) were recruited via Amazon Mechanical Turk and received $1 for their participation. The experiment took 5 minutes on average to complete.

Design

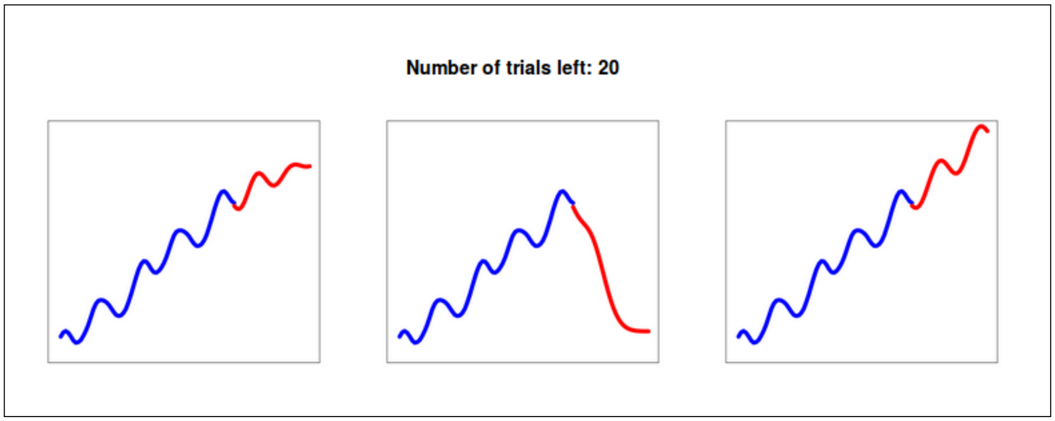
20 different functions were pre-selected[§] sampled from a Gaussian Process parametrized by various compositional kernels within an input space of $x = [0, 0.1, 0.2, \cdots, 10]$. Afterwards, the functional outputs for $x_{\text{learn}} = [0, 0.1, 0.2, \cdots, 7]$ were used as a training set to which all three approaches were fitted and then used to generate predictions for a test set $x_{\text{test}} = [7.1, 7.2, \cdots, 10]$ for generalization. The hyper-parameters of the radial basis kernel as well as the number of components and the hyper-parameters of the spectral mixture kernel were fitted by optimizing the marginal likelihood of the training set. The best prediction of the compositional kernel was found by choosing the composition from the grammar (see Table 5.2) that produced the best marginal likelihood (following Duvenaud et al., 2013).

The different mean predictions were then used to generate 3 plots (one for each kernel approach) that showed the given input as a blue curve and the new predictions (the extrapolation pattern) as a red curve. The procedure was repeated for 20 different compositions, each corresponding to a separate trial.

[§] Most of these functions were generated from two composed kernels and contained either a linear or a periodic component.

Procedure

Participants were asked to select one of 3 extrapolations (pattern completions) they thought best completed a given function (Figure 5.2). The position at which a kernel's predictions appeared was randomized on every trial.



Number of trials left: 20

Figure 5.2: Screenshot of Experiment 2a. Pattern completions (shown in red) were generated by a spectral mixture (left), a radial basis (middle), and a compositional kernel (right). The actual position was determined at random.

5.4.2 Results and discussion

Participants chose the compositional completions on 69% of the trials, the spectral mixture completions on 17%, and the radial basis completions on 14% (Figure 5.3).

Overall, the compositional completions were chosen significantly more often than the other two completions ($\chi^2(N = 1040, df = 2) = 591.2, p < 0.01$). Moreover, assessing for each participant individually whether or not compositional completions were chosen more frequently than the other two based on a $\chi^2(df = 2)$-test, showed that 44 out 52 participants significantly preferred the compositional pattern completions ($\alpha = 0.05$ significance level). The results thus support the hypothesis that participants prefer compositional pattern completions when the ground truth is compositional.

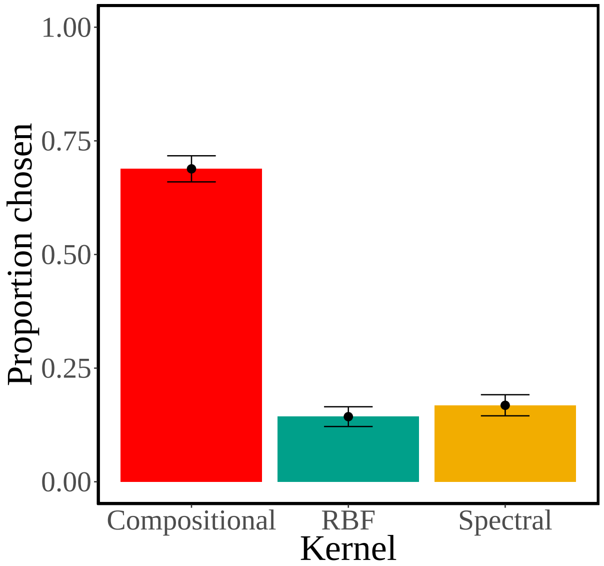


Figure 5.3: Results of Experiment 2a. Proportion of pattern completion choices for three kernels. Error bars represent the standard error of the mean.

5.5 Experiment 2b: Pattern completions of non-compositional functions

While the results of Experiment 2a suggest a preference for compositionally structured functions, they do not indicate whether humans have an inductive bias for such functions, since the results are perfectly compatible with the possibility that participants adapted to the ground truth structure without a compositional inductive bias. In Experiment 2b, I subject the compositional theory to a stronger test, measuring pattern completion preferences when the ground truth is non-compositional (specifically, functions drawn from a GP with the spectral mixture kernel). If participants prefer compositional completions in this experiment, I can be more confident that the preference for such functions arises from an inductive bias.

5.5.1 Methods

Particpants

65 participants (mean age=30, SD = 9.84, 31 male) were recruited from Amazon's Mechanical Turk web service and received $0.5 for their participation. The experiment took 4 minutes on average to complete.

Design

The design was identical to the one used in Experiment 2a, apart from the fact that in this experiment the underlying functions were sampled (without any further selection) from a GP with the spectral mixture kernel parametrized with a randomly assigned number of components (sampled uniformly between 2 and 5). This time, only completions for the compositional and the spectral kernel were generated. I did not generate completions for the radial basis function kernel because in 35% of the cases on average these completions corresponded closely to predictions of the compositional kernel. This correspondence arose due to the fact that there was not much compositional structure for the compositional kernel to capture in samples from the spectral mixture kernel. Additionally, the RBF kernel can be seen as a special case of both the spectral mixture and the compositional kernel.

Procedure

The procedure was as described in Experiment 2a, with the one difference that participants made choices between two (rather than three) completions.

5.5.2 Results and discussion

As in Experiment 2a, participants chose compositional completions more frequently than non-compositional (spectral mixture) completions (68% vs. 32%, $\chi^2(N = 1300, df = 2) = 172.8, p < 0.01$; Figure 5.4). Moreover, 41 of 65 participants significantly preferred the compositional over spectral pattern completions at the $\alpha = 0.05$-level. This finding is consistent with the claim that human inductive biases for functions are sufficiently strong and compositional to induce preference for compositional completions even when the ground truth is non-compositional.

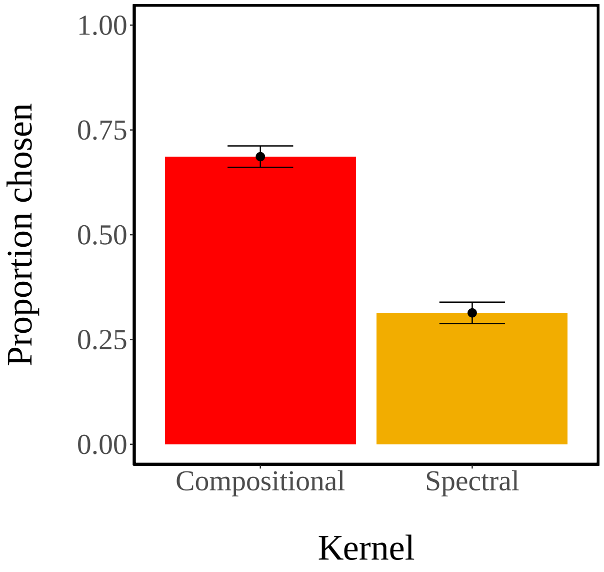


Figure 5.4: Results of Experiment 2b. Proportions of pattern completion choices. Error bars represent the standard error of the mean.

5.6 Markov chain Monte Carlo with people

The next set of experiments seeks to elicit samples from a compositional posterior predictive distribution over functions in order to gain finer-grained insight into the subjective representation of functions. This is accomplished by using a technique called *Markov chain Monte Carlo with People* (MCMCP; Sanborn et al., 2010). This technique asks participants to accept or reject proposed hypotheses, effectively simulating a Markov chain whose stationary distribution is the posterior over hypotheses given the presented data. In the following experiments, I condition on a training set and ask participants to choose between completions (as in Experiments 2a and 2b); thus the stationary distribution is the posterior predictive distribution over pattern completions.

If the proposal distribution is symmetric (the probability of proposing a new state x^* from the current state x is the same as the probability of proposing x from x^*), then one psychologically plausible acceptance function is Barker's acceptance function (Barker, 1965):

$$A(x^*, x) = \frac{\pi(x^*)}{\pi(x^*) + \pi(x)}, \tag{5.9}$$

where $\pi(x) \propto P(x)$. Letting $\mathcal{D}$ indicate the training set, this leads to the following expression for the acceptance function:

$$A(k^*, k) = \frac{p(k^*|\mathcal{D})}{p(k^*|\mathcal{D}) + p(k|\mathcal{D})} \tag{5.10}$$

which corresponds to Luce's choice rule, the most common model for discrete choice distributions in psychology. Thus, under fairly standard assumptions about the choice process, it is possible to elicit samples from the desired distribution, $p(k|\mathcal{D})$, i.e. the posterior predictive distribution over kernels given the data. This means that MCMCP can be used to gain insights into a restricted distribution (under the assumption that predictions are based on the grammar) over compositional parts by assessing how frequently a given compositional prediction has been preferred over another one. I will therefore use empirical data to assess if participants' posterior distributions over different kernels approximates sensible compositions.

5.7 Experiment 3a: Compositional ground truth

In the first MCMCP experiment, the underlying functions were sampled from compositional kernels in order to see if the posterior over compositional completions converges to patterns that match the true underlying kernel. Using a MCMCP-like design provides the opportunity to check if the found restricted posterior predictive distribution corresponds well with the predictions generated by the compositional kernel.

5.7.1 Methods

Design

Eight different functions were sampled from various compositional kernels. The input space was split into training and test sets, and then all kernel combinations were used to generate completions for the test set. I generated completions from all kernels shown in Table 5.2, optimizing each kernel's hyper-parameters on the training set and then generating the completions for the extrapolation set. On each trial, participants chose between their most recently accepted extrapolation and a new proposal. Proposals were sampled uniformly from this set. I mainly focused on combinations containing linear and periodic components, as these provide more interesting structure than the smoothness induced by samples from the radial basis function kernel.

Participants

51 participants (27 male) with an average age of 32.55 (SD = 8.21) were recruited via Amazon's Mechanical Turk web service and paid $1. The experiment took 8 minutes on average to complete.

Procedure

There were 8 blocks of 30 trials, where each block corresponded to a single extrapolation set. The order of the blocks was randomized for each participant.

5.7.2 Results and discussion

The average proportion of accepted kernels over the last 5 trials was calculated, as shown in Figure 5.5. The first 25 trials were excluded to avoid trials on which participants might not have reliably moved towards their subjective posterior extrapolations, a process commonly called "burn-in".

In all cases, participants' subjective probability distribution over completions placed the greatest mass on the data-generating kernel (marked in red). Furthermore, I observed a strong rank correlation between an approximated posterior probability over completions[¶] and participants' subjective distribution ($\varrho = 0.91, p < 0.01$). Thus, participants approximately moved toward the true underlying posterior when the functions were generated from compositional kernels.

[¶] As there is currently no reliable way to generate an exact posterior over compositional parts, the normalized marginal likelihood of each composition was used as a rough approximation.

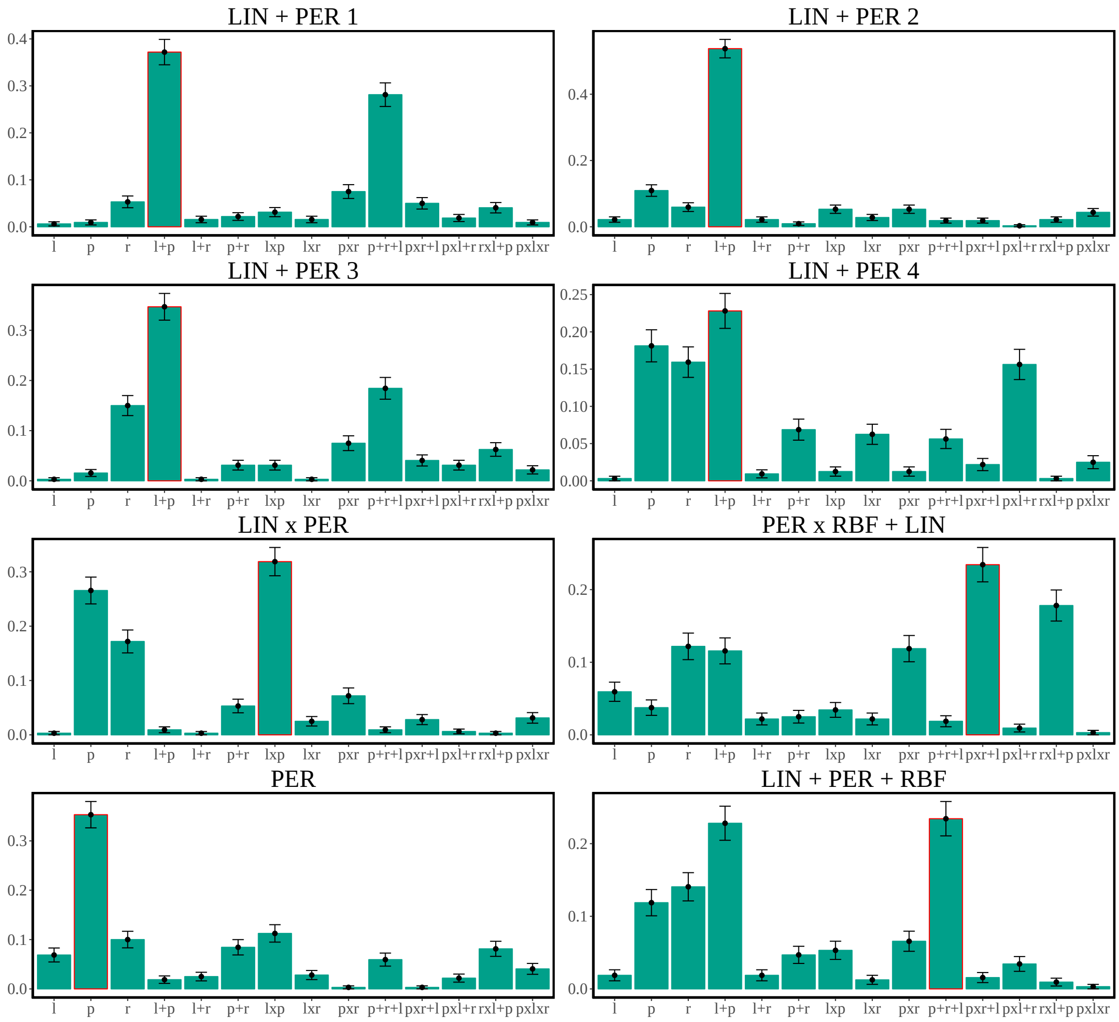


Figure 5.5: Results of Experiment 3a. Proportions of chosen completions over the last 5 trials. Error bars represent the standard error of the mean. Generating kernel (ground truth) marked in red. l: linear kernel, p: periodic kernel, r: radial basis function kernel.

5.8 Experiment 3b: Real-world functions

The second MCMCP experiment assessed what structures people converged to when faced with real-world data, where the ground truth is unknown. The expectation was that real-world data is often intrinsically compositional (see Duvenaud et al., 2013; Grosse et al., 2012), and hence human inductive biases may be adapted to such functions.

Participants

51 participants (31 male) with an average age of 32.55 (SD = 12.14) were recruited via Amazon Mechanical Turk and received $1 for their participation. The experiment took 7 minutes on average to complete.

Procedure

I used four real-world time series (Figure 5.6): airline passenger data, volcano CO2 emission data[‖], the number of gym memberships over 5 years, and the number of times people googled the band "Wham!" over the last 8 years. Some of these functions have been extensively analyzed in the time series modeling literature and all of them showed interesting patterns a priori. Participants were not told any information about the data sets (including input and output descriptions); they were simply shown the unlabeled input-output pairs in the experiment.

The input space was split into training (75% of the data) and test sets (25% of the data), and then all kernel combinations were fitted to the training set and used to generate completions for the test set. Proposals were sampled uniformly from this set. As periodicity in the real world is rarely ever purely periodic, I adapted the periodic component of the grammar by multiplying a periodic kernel with a radial basis function kernel, thereby locally smoothing the periodic part of the function.[**] Apart from the different training sets, the procedure was identical to Experiment 3a.

[‖] As shown in Chapter 3

[**] See the following page for an example: `http://learning.eng.cam.ac.uk/carl/mauna`.

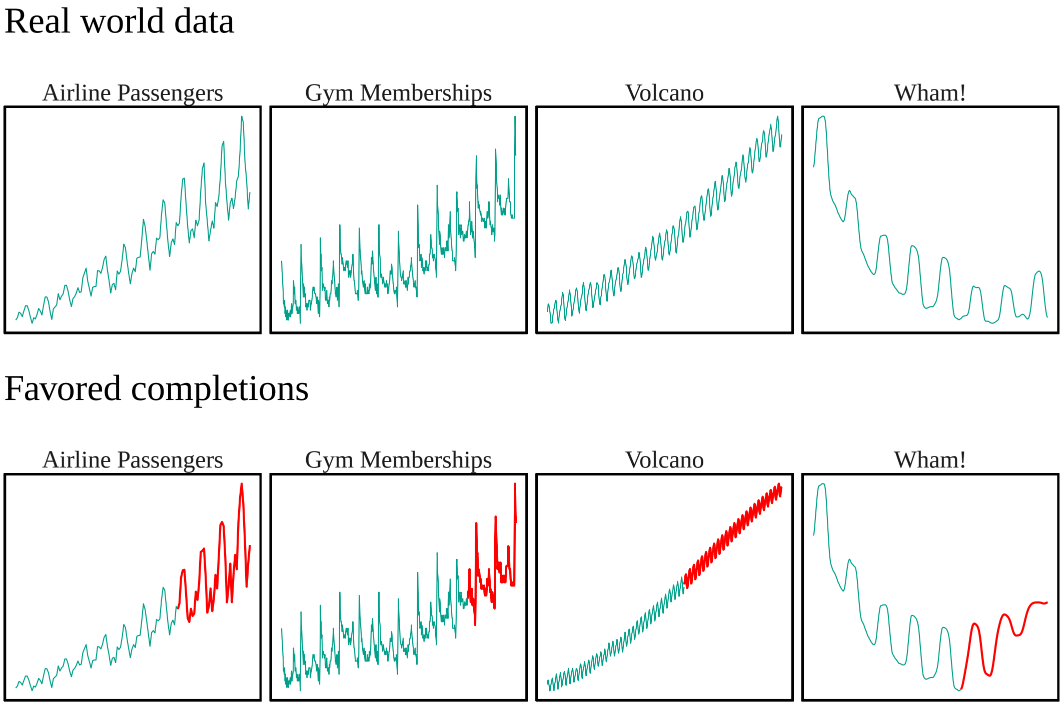


Figure 5.6: (Top) Real-world data sets used in Experiment 3b. Descriptions and origin of the data were unknown to participants. (Bottom) Participants were shown the region in blue; most frequently selected completions are shown in red. Note that the periodic composition has been adapted by multiplying it with a radial basis kernel.

Results and discussion

Results (again taken from the last 5 trials) are shown in Figure 5.7, demonstrating that participants moved to intuitively plausible patterns. In particular, for both the volcano and the airline passenger data, participants moved to compositions resembling those found in previous analyses Duvenaud et al. (2013). The most frequently chosen completions for each data set are shown in Figure 5.7. The rank correlation between the subjective distributions and the approximated posterior over completions was significantly positive ($\varrho = 0.83$, $p < 0.01$), supporting the hypothesis that the compositional pattern completions capture human inferences about functions.

5.9 Experiment 4: Manual pattern completion

As so far participants were only asked to make choices between a discrete set of pattern completions, the next experiment will measure pattern completion in a less constrained task.

In Experiment 4, participants had to draw the pattern completions manually (see Cox et al., 2012, for related work), instead of choosing completions out of a provided set.

5.9.1 Methods

Design

On each round of the experiment, functions were sampled from the compositional grammar at random, the number of points to be presented on each trial was sampled uniformly between 100 and 200, and the noise variance was sampled uniformly between 0 and 25. Finally, the size of an unobserved region of the function (for completion) was sampled to be of a size between 5 and 50. Participants were asked to manually draw the function best

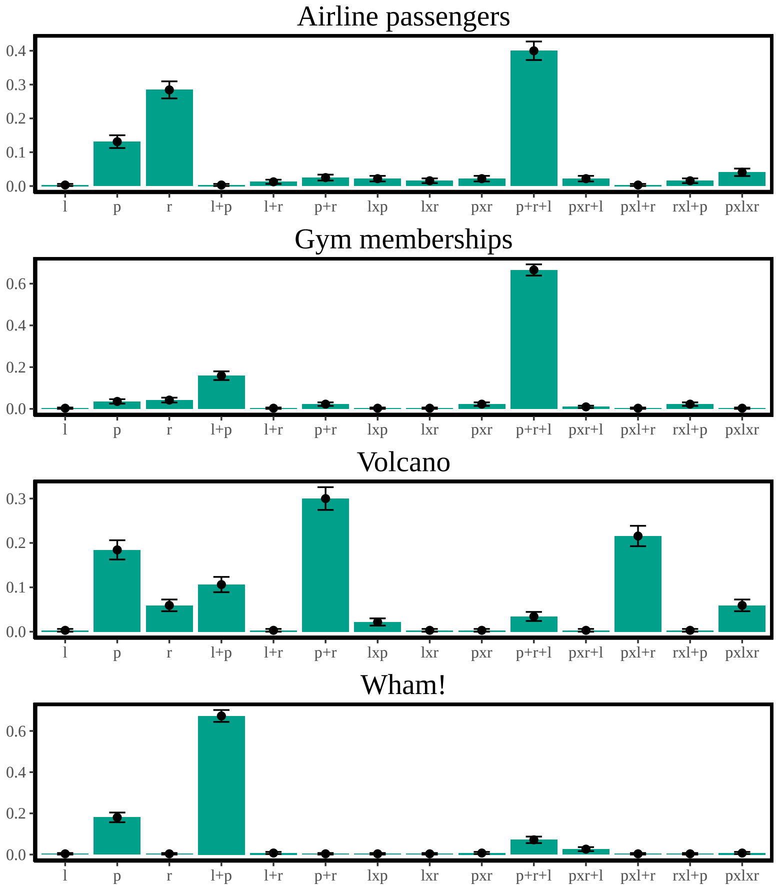


Figure 5.7: Results of Experiment 3b. Proportions of chosen predictions over last 5 trials. l: linear kernel, p: periodic kernel, r: radial basis function kernel.

describing the observed data and to complete this function within the observed and unobserved regions. A screenshot of the experiment is shown in Figure 5.8.

Participants

36 participants (23 males) with a mean age of 30.5 (SD = 7.15) were recruited from Amazon Mechanical Turk and received $2 for their participation. The experiment took 12 minutes on average.

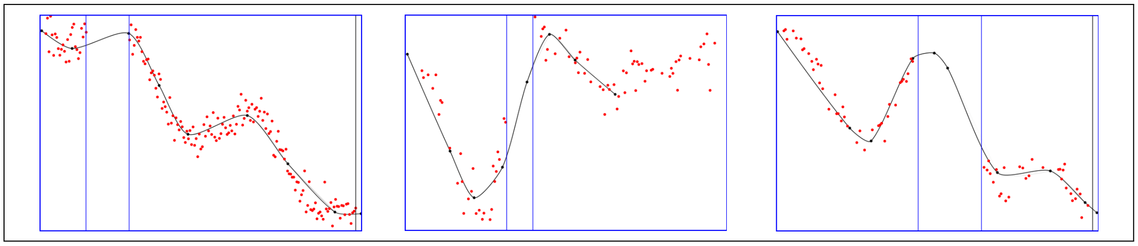


Figure 5.8: Screenshots of manual pattern completions (Experiment 4). The unobserved region (for completion) is delimited by vertical lines.

Procedure

Participants were asked to draw lines into a cloud of dots that they thought best described the given data. To facilitate this process, participants placed black points into the cloud, which were then automatically connected by a black line based on a cubic Bezier smoothing curve (see Forrest, 1972). The smoothness of the Bezier curve was adapted online to fit the points provided by the participants, thereby making it possible to draw less smooth lines by placing points closer to each other[††]. Participants were asked to place the first point on the left boundary and the final point on the right boundary of the graph. In between, participants were allowed to place as many points as they liked (from left to right) and could remove previously placed points. There were 50 trials in total. The dots were sampled from a function that was in turn sampled from a kernel chosen at random out of all possible combinations in the compositional grammar (as shown in Table 5.2).

5.9.2 Results and discussion

The average root mean squared distance between participants' predictions (the line they drew) and the mean predictions of each kernel given the data was assessed, for both inter-

[††] Participants found this set-up to be very intuitive. The source code for this experiment can be found online at https://github.com/ericschulz/drawfunctions

and extrapolation areas. No parameters were optimized in order to achieve better distance measures. As before, the parameters for the radial basis kernel and the number of components as well as the parameters for the spectral mixture kernel were optimized via the log marginal likelihood. I also generated predictions of the compositional kernel by choosing the composition that produced the best marginal log-likelihood given the presented dots. Results are shown in Figure 5.9.

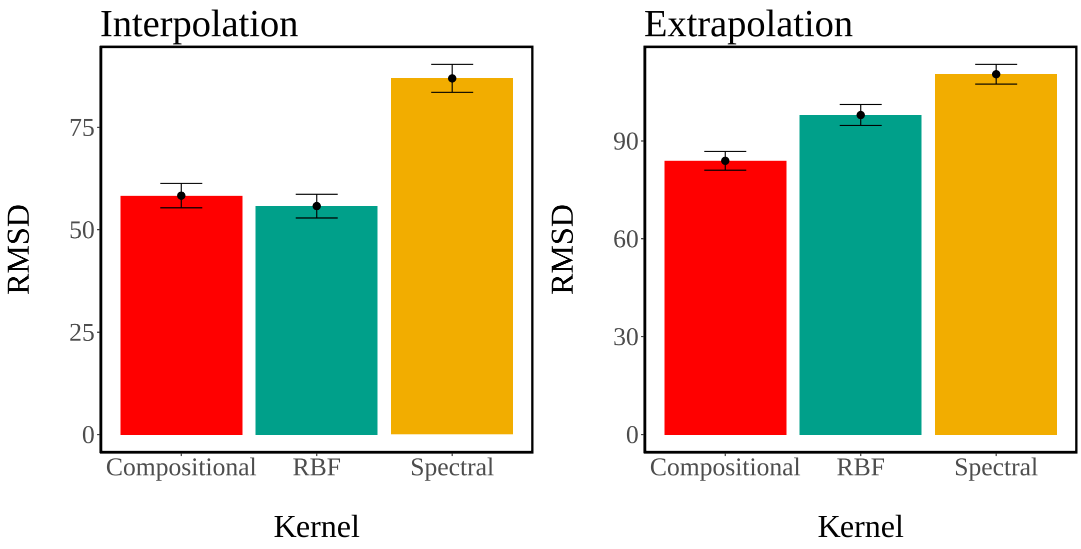


Figure 5.9: Results of Experiment 4. Average root mean squared distance for interpolation (left) and extrapolation (right) drawings. Error bars show standard error of the mean.

The mean distance between the model predictions and participants' drawings were compared by performing a hierarchical t-test between the distances produced by the different models while accounting for the nestedness of distance measures within participants and trials; this is essentially the same as performing a mixed-effects regression. The mean distance between predictions and participants' drawings was significantly higher for the spectral mixture kernel than for the compositional kernel in both interpolation (86.96 vs. 58.33, hierarchical t-test: $t(676) = -2.28, p < 0.05, d = 0.35$) and extrapolation areas (110.45 vs

83.91, hierarchical t-test: $t(434) = 2.2, p < 0.05, d = 0.33$). The radial basis function kernel produced similar distances as the compositional kernel in interpolation (55.8, hierarchical t-test: $t(649) = -0.55, p > 0.05, d = 0.03$), but predicted participants' drawings significantly worse in extrapolation areas (97.9, $t(508) = 1.8, p < .05, d = 0.17$). As extrapolation is normally seen as the best criterion to measure generalization and thus the best measure to compare models of function learning (DeLosh et al., 1997), the difference in extrapolation judgments provides further evidence for the compositional kernel model.

5.10 Generating comparison functions

One potential concern with the previous experiments is that compositional and non-compositional functions may differ in terms of low-level perceptual characteristics, and these differences (rather than the hypothesized high-level structural differences) are driving the behavioral effects. To address this concern, a set of non-compositional functions whose low-level perceptual characteristics are matched with a set of compositional functions is required.

Thus, I generated 500 functions from both compositional and non-compositional kernels. The compositional functions were sampled randomly from the compositional grammar by first randomly sampling a kernel composition and then sampling a function from that kernel, whereas the non-compositional functions were sampled from the spectral mixture kernel, where the number of components was varied between 2 and 6 uniformly. Following Goerg (2013), I then calculated the standardized spectral entropy (or "forecastability") for each function:

$$\Omega(f) = 1 - \frac{H(f)}{\ln(2\pi)} \tag{5.11}$$

where $H(\cdot)$ is the Shannon entropy (see Appendix A.1) of the function's normalized spectral density S_f.

Forecastability can be seen as a measure of how well future output values of a function can be predicted. It takes values between 0% and 100%, representing the proportional reduction in entropy a given function achieves relative to white noise. Although forecastability technically assumes stationary functions, it has also been shown to produce reliable measures for non-stationary functions in practice (Hyndman et al., 2015). Of the 500 non-compositional functions generated, I selected those that had a forecastability of higher than 20%, but set the maximum predictability for the compositional kernel functions to be less than 40% and the minimum predictability for the spectral mixture kernel functions to be higher than 40%. This means that theoretically the functions generated from the spectral mixture kernel are more forecastable on average (i.e., contain lower entropy in their spectral density) than the functions sampled from the compositional kernel. Because of this forecastability advantage for non-compositional functions, behavior consistent with the compositional kernel predictions would then provide especially strong evidence for the compositional framework.

Afterwards, I verified that all of the functions were matched in terms of low level visual properties, as measured by a similarity measure based on the discrete wavelet Haar transform Montero & Vilar (2014). The basic idea of this similarity measure is to replace the original series by its wavelet approximation coefficients at an appropriate scale, and then to measure similarity between the wavelet coefficients (see Appendix). This measure not only provides a state-of-the art similarity metric between two functions, but also has been used to model human behavior in visual integration tasks (Field et al., 1993). Each function was ranked based on its wavelet similarity to other functions and the top functions (20 compositional and 20 spectral) were selected from this set. These functions, which are used in the experiments reported below, are shown in Figures 5.10 and 5.11. Subsequently, I will refer to these functions as the "matched set."

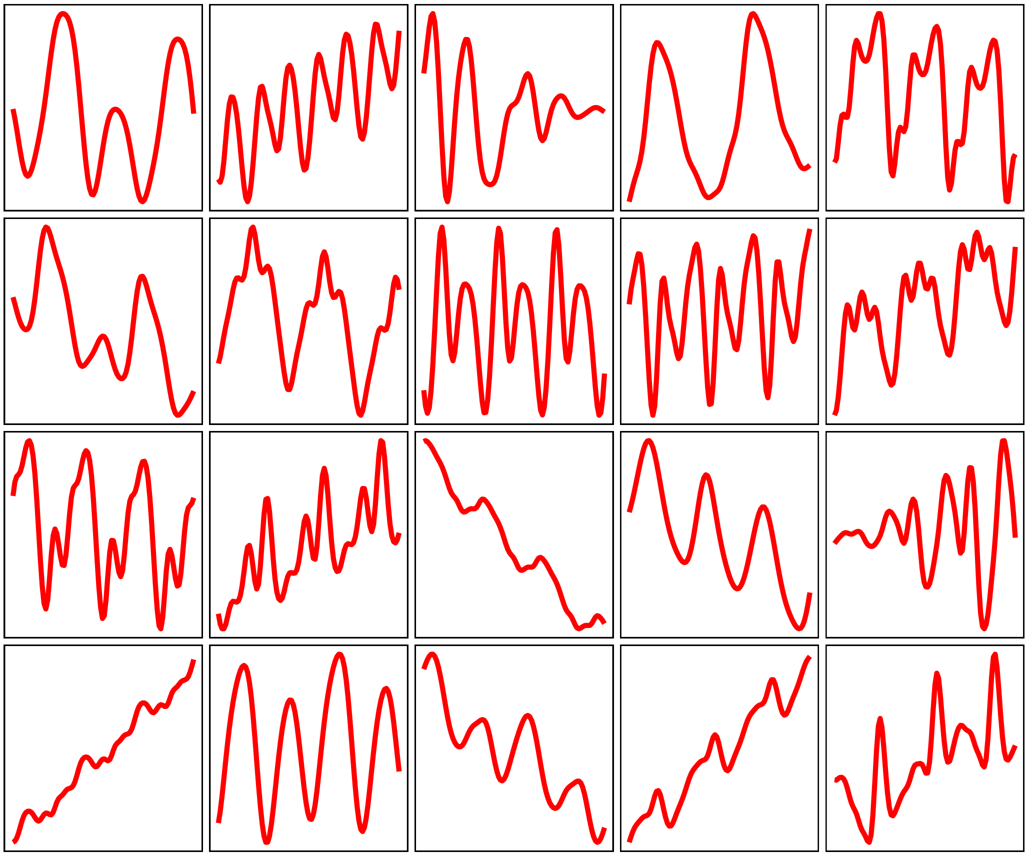


Figure 5.10: Sampled compositional functions.

5.11 Experiment 5: Compositional predictability

If human inductive biases for functions are inherently compositional, then compositional functions should be perceived as more *predictable*.[‡‡] In Chapter 4, I operationalized predictability in terms of generalization error. As predictability is the expected prediction

[‡‡] Predictability is closely related to forecastability, insofar as both measures can be characterized in terms of the eigenspectrum (Sollich, 2002, see also Chapter 4). Here, I focus on predictability since it has a more intuitive interpretation.

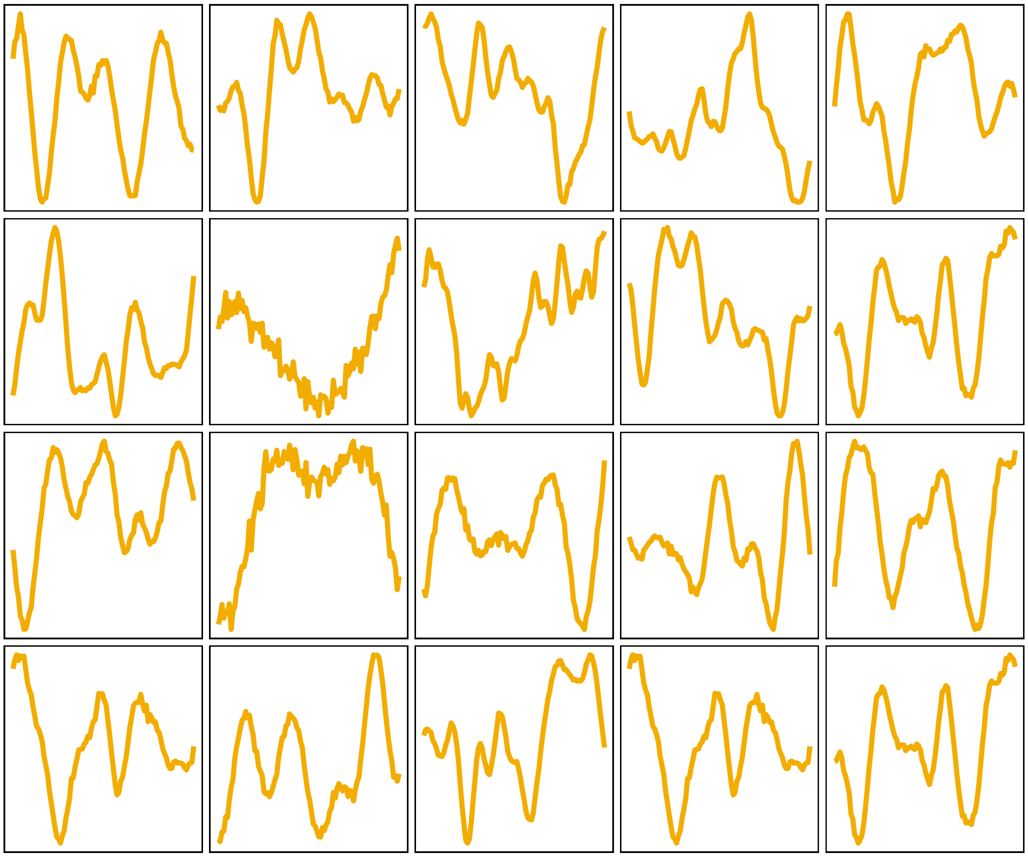


Figure 5.11: Sampled spectral mixture functions.

error for a newly-sampled input point from within a (interpolation) set of points, more predictable functions should lead to lower generalization error for new inputs. This analysis identified several key factors determining the predictability of a function; specifically, predictability increases with sample size and smoothness, and decreases with noise.

In this section, I will assume that compositionality is another key factor influencing predictability. If inductive biases for functions are compositional, then I expect that such functions are perceived as more predictable. This will be now assessed by collecting both absolute and relative measures of subjective predictability.

5.11.1 Methods

Participants

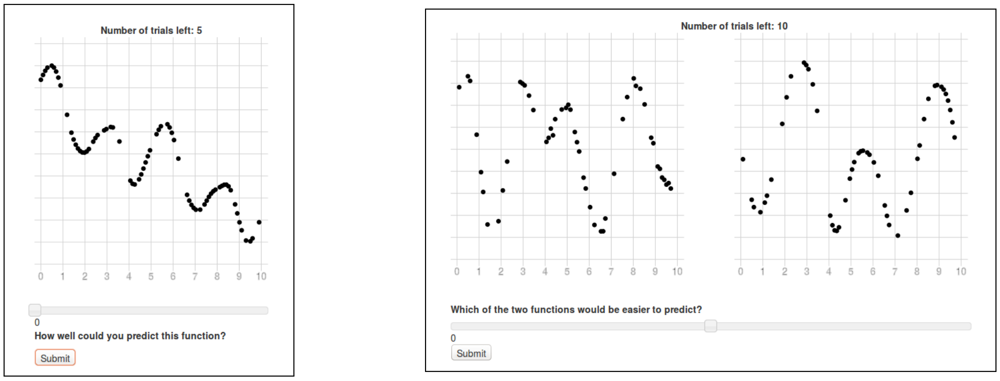
50 participants (mean age=32, SD=7.2; 32 males) were recruited via Amazon Mechanical Turk and received $0.5 for their participation. The experiment took 9 minutes on average.

Procedure

On each trial, one of the "matched" functions defined above was randomly sub-sampled with a sample size n_j drawn uniformly from $\{50, 60, \ldots, 100\}$. Participants were asked to judge how well they thought they could predict a newly sampled input point for the function on a scale ranging from 0 (not at all) to 100 (very well). After judging the subjective predictability for all 40 of the matched functions in a randomized order, participants then had to make pairwise comparisons between compositional and non-compositional functions from -100 (function presented on the left is definitely easier to predict) to 100 (function presented on the right is definitely easier to predict). As with the absolute predictability judgments, the sample size n_j was varied randomly, with the constraint that both functions had the same sample size and the position of the functions were counter-balanced over trials. Screenshots of the two tasks are shown in Figure 5.12.

5.11.2 Results and discussion

As shown in Figure 5.13, compositional functions were perceived as more predictable (mean predictability judgment=55.92) than non-compositional functions (mean predictability judgment=37.88). This was confirmed by a hierarchical t-test with random effects for participants and items, $t(31) = 8.35, p < 0.001, d = 0.73$. In, the predictability judgments for compositional judgments were more strongly correlated with sample size ($r = 0.28, p <$


Number of trials left: 5

0

How well could you predict this function?

Number of trials left: 10

Which of the two functions would be easier to predict?

0

Figure 5.12: Screenshots of the two predictability judgment tasks (Experiment 5). (Left) Absolute predictability judgments. (Right) Relative predictability judgments.

0.001) compared to predictability judgments for non-compositional functions ($r = 0.18$, $p < 0.001$; difference test: $z = 2.36$, $p < 0.05$).

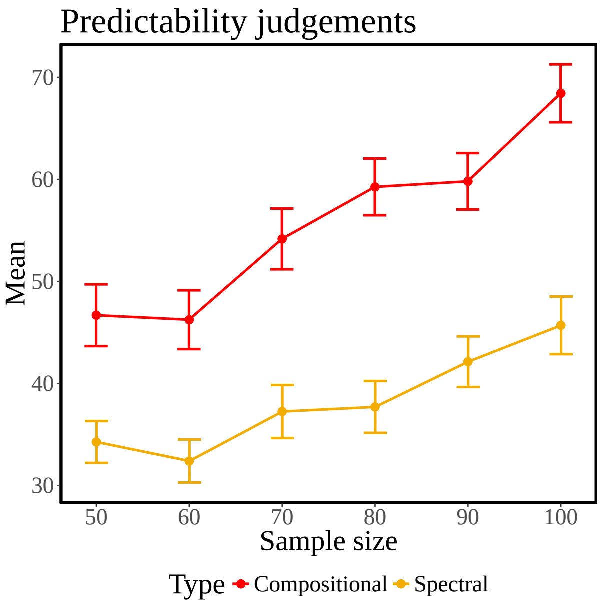


Figure 5.13: Mean predictability judgments. Error bars represent the standard error of the mean.

To analyze these results further, I fit a hierarchical regression model of subjective predictability (nested within participants) with two predictors: samples size and a dummy variable indicating whether or not a presented function was compositional. The fixed effect results of this analysis, summarized in Table 5.3, demonstrate main effects of both compositionality and sample size.

Table 5.3: Regression model of predictability judgments. Overall model fit: Conditional pseudo-$R^2 = 0.17$.

	Estimate	Std. Error	t-value	Pr(>|t|)
Intercept	13.06	3.50	3.72	0.0002
Compositional	17.44	1.53	11.39	0.0000
Sample size	0.34	0.04	7.42	0.0000

The relative predictability judgments tell a similar story (Figure 5.14). As with the absolute judgments, compositional functions were perceived to be more predictable relative to non-compositional functions ($t(499) = 13.502, p < 0.001, d = 0.63$), and this difference increased with sample size ($r = 0.14, p < 0.001$).

To understand how subjective predictability judgments match theoretical predictability, I approximated the generalization error using the average squared error for a randomly sampled new input point. This approximation is computed for every sample that a participant saw under different kernel choices: linear, radial basis function, spectral mixture, or a compositional kernel. To generate predictions from each kernel, the hyper-parameters (including the composition for the compositional kernel) that maximized the log marginal likelihood were used and the mean generalization error was assessed empirically. This approach tests if the ease of generalizations for different functions as assessed by different approaches towards generalizations tracks participants' predictability judgments.

Figure 5.15 shows the average correlation between each kernel's generalization error and participants' predictability judgments, averaged over all trials and participants.

The spectral mixture kernel (t-test against a correlation of 0: $t(49) = -2.351, p < 0.05$, $d = 0.33$), the compositional kernel ($t(49) = -19.73, p < 0.001, d = 2.79$) as well

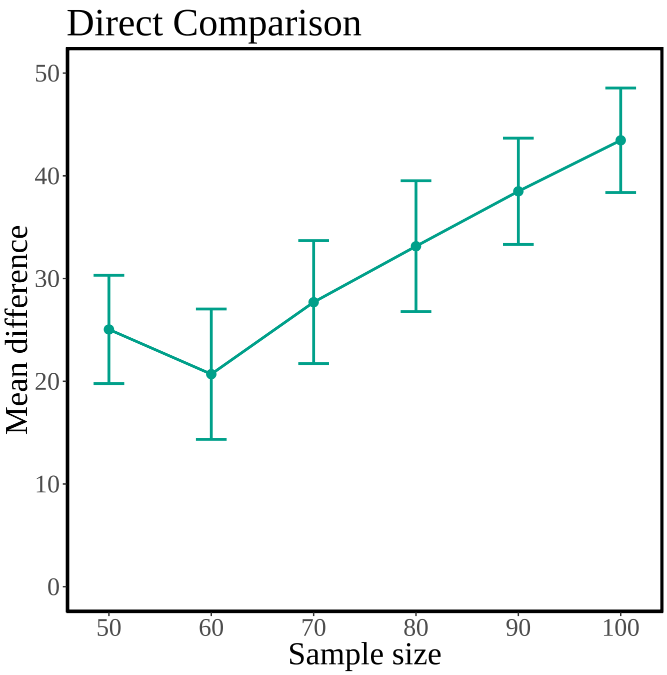


Figure 5.14: Relative predictability judgments. Positive values indicate that the compositional function was judged to be more predictable than the non-compositional function. Error bars represent the standard error of the mean.

as the radial basis function kernel ($t(49) = -4.3, p < 0.01, d = 0.61$) all correlated significantly with participants' judgments. The linear ($t(49) = 0.846, p > 0.1, d = 0.12$) kernel did not correlate with participants' judgments better than chance. Crucially, the compositional kernel showed a significantly higher correlation than either the spectral mixture kernel ($t(49) = -7.86, p < 0.001, d = 1.57$) or the radial basis function kernel ($t(49) = -9.92, p < 0.001, d = 1.49$).

In summary, compositional functions were perceived as more predictable, even though they were generated to be theoretically less "forecastable" than the non-compositional functions sampled from the spectral mixture kernel. The generalization error produced by the compositional structure learning approach matched participants' predictability judgments more closely than any of the alternative models.

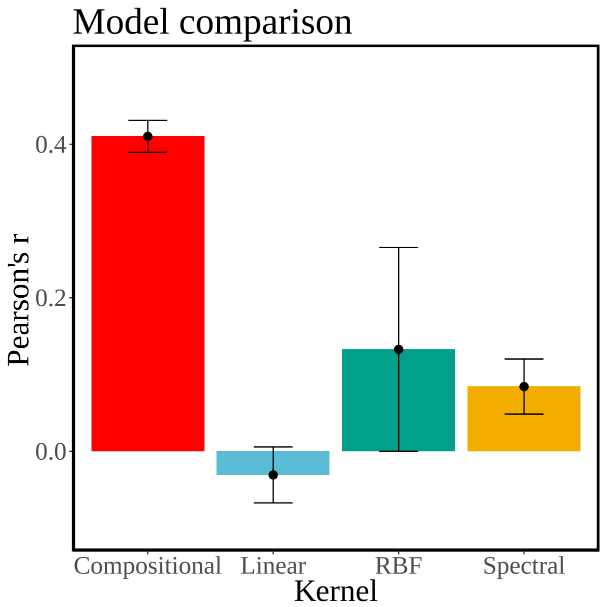


Figure 5.15: Average correlation between model predictions and participants' predictability judgments. Predictions were derived from models' average generalization error. As generalization error and predictability are inversely related, the inverse of the generalization error was used to produce the correlations. Error bars represent the standard error of the mean.

5.12 Experiment 6: Traditional function learning paradigm

In the experiments presented so far, I have only used *single function learning* paradigms with a particular focus on pattern completions, either by letting participants choose completions in forced choice or MCMCP tasks or by asking them to draw manual pattern completions. Psychological experiments on function learning have traditionally focused on *sequential function learning* (as described in Chapter 3). For example, Carroll (1963) asked participants to predict the width of one bar given another bar's width. After every prediction, participants received feedback about the actual width of the bar. Performing this task required participants to build up a representation of a function sequentially, rather than a more perceptual generalization that can be used in a single function learning task.

Thus, in order to compare the compositional and non-compositional representations in terms of how they describe participants' learning and generalization within a historically more accurate setting, the next experiment will use a similar (yet slightly modified) paradigm as the one used by Carroll (1963).

5.12.1 Methods

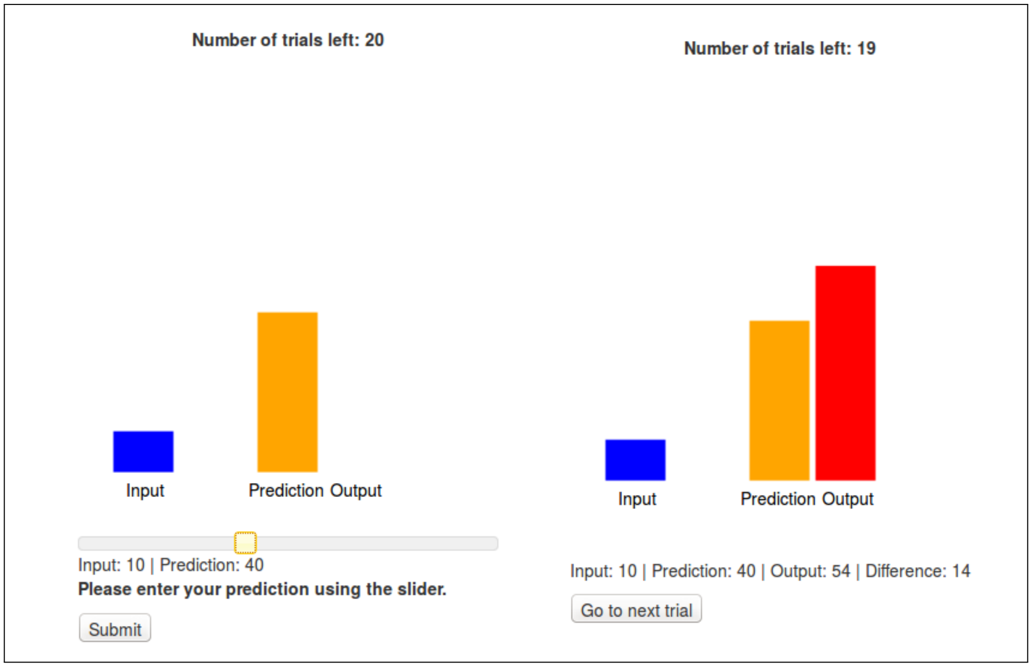
Participants

46 participants (mean age=31, SD=11; 30 males) were recruited via Amazon Mechanical Turk and received $0.5 for their participation. The experiment took 12 minutes on average to complete.

Procedure

Participants were asked to predict an output indicated by the height of a red bar given the current input indicated by the height of a blue bar (Figure 5.16). On each trial, they saw the current input (a blue bar) and had to indicate their predictions by adjusting the height of an orange bar using a slider. After submitting their prediction, the actual output appeared marked as a red bar directly next to the orange bar. Additionally, participants saw the numerical value of the current input, their prediction and –after they submitted their predictions– the output, as well as the difference between their prediction and the actual output. It was explicitly stated to participants that the input is related to the output by an underlying function and that they had to learn that function in order to produce the smallest possible difference between their predictions and the outputs.

Participants learned 4 different functions over 4 blocks. Out of the 4 different functions participants had to learn, 2 were sampled from the set of matched compositional functions


Number of trials left: 20

Input: 10 | Prediction: 40

Please enter your prediction using the slider.

Number of trials left: 19

Input: 10 | Prediction: 40 | Output: 54 | Difference: 14

Figure 5.16: Screenshots of Experiment 6. The height of the blue bar indicates the input. The height of the orange bar marks the prediction and can be adjusted by using the slider. After the prediction was submitted, the actual output, marked by the height of the red bar, appeared and participants were told the absolute difference between their prediction and the actual outcome.

and 2 were sampled from the set of matched non-compositional functions. Each block consisted of 20 trials (input-output pairs). The input was uniformly sampled between 0 and 100, and the output was also transformed to range between 0 and 100.

5.12.2 Results and discussion

Using a similar hierarchical t-test as before, participants' predictions were only marginally more accurate for the compositional functions than for the non-compositional functions ($t(34) = 1.6, p < 0.1, d = 0.18$, see Figure 5.17). Performing a linear regression with the absolute difference between participants' predictions and the actual outputs as the dependent variable and a factor variable encoding whether or not the underlying function was compositional as well as the trial number (how many predictions participants had made before) resulted in the model summarized in Table 5.4.

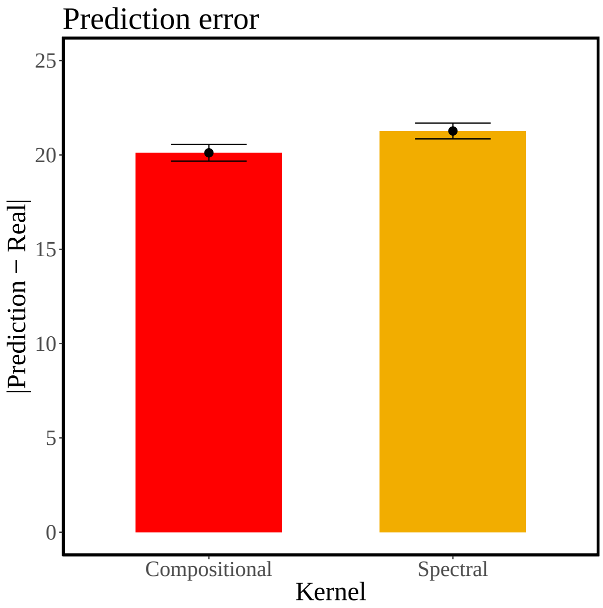


Figure 5.17: Results of Experiment 6. Mean difference between predictions and outcomes for compositional and spectral kernel. Error bars represent the standard error of the mean.

Table 5.4: Regression model of the traditional function learning paradigm. Overall model fit: $R^2 = 0.14$.

	Estimate	Std. Error	t value	Pr(>|t|)
Intercept	23.01	0.72	32.02	0.0000
Compositional	-1.56	0.62	-2.52	0.0118
Trial	-0.30	0.0535	-5.65	0.0000

Participants improved over the course of each block, as evidenced by the main effect of trial. While compositional functions are on average predicted better ($\beta = -1.56$), this effect turned out to be relatively small; perhaps this is because both compositional and non-compositional functions were non-linear, and these are notoriously hard to learn in the traditional function learning paradigm (Byun, 1995). Alternatively, this could also be explained by the fact that the matched set was created to contain relatively similar functions.

To further disentangle the different models, sequential (i.e., trial-by-trial) predictions for the linear, compositional, spectral mixture and radial basis function kernels, were created. This allows to further assess how well each model's predictions for the $n+1$ trial after having seen n outputs matched participants' predictions. As shown in Figure 5.18, the compositional kernel described participants' predictions best ($r = 0.54, p < 0.01$) followed by the linear kernel ($r = 0.21, p < 0.01$). The correlation for both the RBF kernel ($r = 0.03$) and the spectral mixture kernel did not differ significantly from zero (mean correlation: $r = 0.04, p > 0.5$). Importantly, the compositional kernel predicted participants' predictions significantly better than the linear kernel ($t(45) = 9.38, p < 0.001, d = 1.14$). Additionally, more heuristic strategies such as matching the height of the input bar ($r = 0.14$, $p < 0.01$) or entering as the next input the output of the previous trial ($r = 0.15, p < 0.01$) did not predict participants' trial-by-trial behavior better than the compositional model. Thus, the compositional kernel appears to provide the best account of trial-by-trial learning in this more traditional function learning paradigm.

5.13 Experiment 7: Assessing numerosity

In the next series of experiments, I will explore the broader implications of a compositional function representation for 3 domains of cognition: numerosity perception, change detection, and short-term memory. I will begin with a standard numerosity judgment task, in which participants estimate the number of dots appearing briefly on a screen. Zhao & Yu (2016) showed that, in the absence of explicit grouping cues, structured configurations of dots led to lower estimates compared to random configurations. This effect of structure on perceived numerosity appears to arise from the fact that dots belonging to structured con-

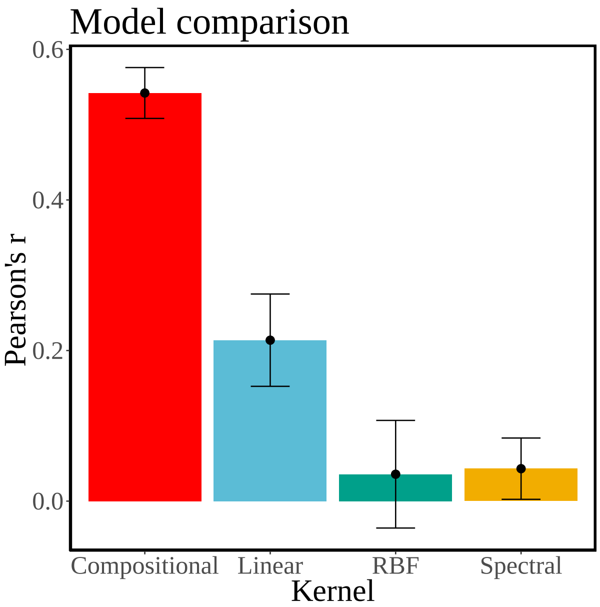


Figure 5.18: Model fit (Pearson correlation coefficient) for predicting participants' next predictions given the current input over all trials in Experiment 6. Error bars represent the standard error of the mean.

figurations are more likely to be perceptually grouped together; these groups then become the units which are enumerated. In a related study, Zhao et al. (2011) showed that the accuracy of numerosity judgments improves when statistical regularities are removed, indicating a direct relationship between statistical learning and numerosity perception.

If compositional functions are perceived as more "structured" than non-compositional functions (as suggested indirectly by the predictability experiment), then a natural hypothesis is that dot configurations generated from compositional functions will be perceived as less numerous.

5.13.1 Methods

Participants

91 participants (mean age=37.84, SD=7.87, 48 female) were recruited via Amazon Mechanical Turk and received $0.5 for their participation. The experiment took 7 minutes on average to complete.

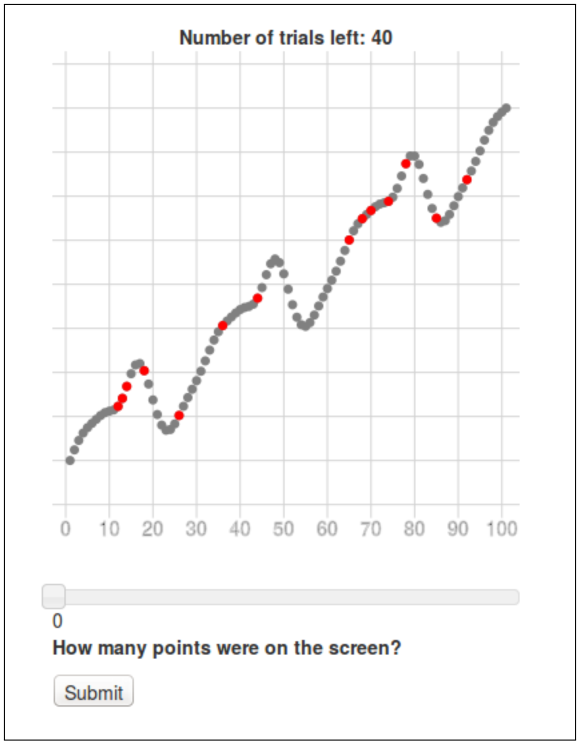
Procedure

Each participant completed a total of 40 trials. On each trial, participants were presented with a function drawn from the set of matched functions, represented by 100 equidistant gray dots on the screen. This configuration was displayed for one second, after which the dots vanished. Out of the 100 dots, $n = [5, 6, 7, \ldots, 15]$ randomly selected dots were marked as red. Participants were asked to estimate the number of red dots (between 0 and 20) after all the dots had vanished. A screen shot of the experiment is shown in Figure 5.19.

5.13.2 Results and discussion

Figure 5.20 shows the effect of the number of dots on participants' numerosity judgments, demonstrating that increasing the number of red dots led to greater underestimation of the actual number, consistent with the idea that perceived numerosity diminishes as structure becomes more discernible. Moreover, it can be seen that there is a difference between compositional and non-compositional functions, even though this difference was rather small.

As I hypothesized that participants would underestimate the red dots if they were superimposed on more structured functions (functions sampled from a compositional kernel) as compared to relatively unstructured functions (sampled from the spectral mixture kernel) overall, I performed an aggregated comparison next. Figure 5.21 shows the direct



Number of trials left: 40

How many points were on the screen?

Figure 5.19: Screenshot of Experiment 7. Dots stayed on the screen for 1000ms and then vanished before the slider could be used.

comparison of numerosity estimates for compositional and non-compositional patterns. Overall, participants underestimated the number of dots more for compositional than for non-compositional patterns (-0.74 vs. -0.53, hierarchical t-test: $t(38) = -2.2, p < 0.05$, $d = 0.43$).

A regression analysis, with the difference between participants' estimates and the actual number of dots as the dependent variable, was performed to account for both the number of dots and the function type. The results (presented in Table 5.5) showed that with an increasing number of red dots, participants showed a stronger tendency to underestimate numerosity ($\beta = -0.39$). Additionally, there was a main effect of function type ($\beta = -0.17$).

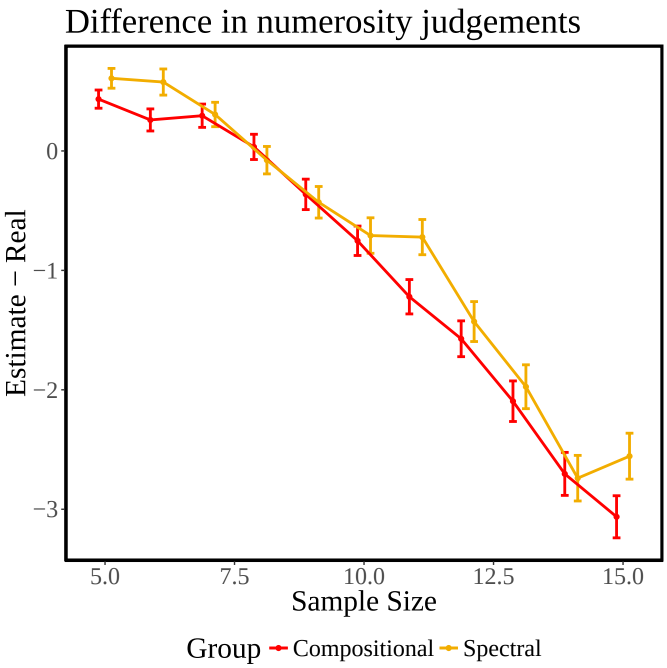


Figure 5.20: Mean difference between presented and estimated number of red dots over number of shown dots. Error bars represent the standard error of the mean. As more red dots are presented, participants tend to underestimate the actual number more.

Taken together, these results indicate that perceived numerosity is reduced to a greater extent by compositional functions compared to non-compositional functions, a finding that agrees with the results of Zhao & Yu (2016): structural regularity distorts the units of perception, making them appear less numerous. Even though the non-compositional functions used here were in fact structured (as measured by forecastability), they produced a weaker effect on numerosity relative to compositional functions, arguably because their structure is less "intuitive" (i.e., they lack the inductive bias needed to easily perceive this structure).

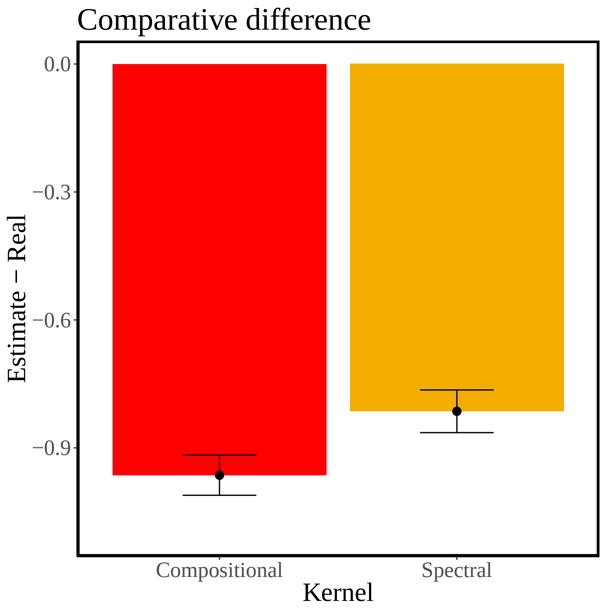


Figure 5.21: Results of Experiment 7. Comparative judgments. Lower values mean that participants underestimated the number of red dots. Error bars represent the standard error of the mean.

Table 5.5: Parameter estimates from numerosity judgments regression analysis.

	Estimate	Std. Error	t-value	Pr(>|t|)
Intercept	3.23	0.12	27.631	<2e-16
Number of dots	-0.39	0.01	-35.845	<2e-16
Compositional	-0.17	0.07	-2.43	0.015

5.14 Experiment 8: Change detection with functions

Next, I assessed the effect of compositional structure on change detection performance (Pashler, 1988; Rouder et al., 2011). In a typical change detection paradigm, participants judge whether two stimuli presented rapidly in sequence are the same or different. It has been suggested that structured representations facilitate change detection by allowing a summary representation of the stimulus to be stored in short-term memory (Brady & Tenenbaum, 2013). When the stimulus consists of multiple items, those items that are not assimilated into the summary representation are encoded as "outliers". However, because memory is capacity-limited, only a small number of outliers can be encoded. Thus, the summary representation frees up encoding resources for specific items. The more structure in the display that can be encoded, the more resources will be available for encoding specific items.

The critical question concerns the nature of this structure—what are the inductive biases that constrain short-term memory representations? Brady & Tenenbaum (2013) used Markov random fields to encode information about object features; several other structural assumptions have been explored in the literature (Orhan & Jacobs, 2013; Mathy & Feldman, 2012; Lew & Vul, 2015). Here I explore the possibility that the representation of functions in short-term memory is compositional, leading to the prediction that change detection will be more accurate with compositional functions compared to with non-compositional functions. This prediction links the compressibility of functions directly to their generalizability. In fact, recent attempts in Bayesian machine learning have shown that models providing a good description of data can also be used for lossy compressions of the same data (see Ghahramani, 2015).

5.14.1 Participants

66 participants (mean age=30, SD=8.2, 42 males) were recruited from Amazon Mechanical Turk and received $0.5 for their participation. The experiment took 10 minutes on average.

5.14.2 Procedure

Participants judged whether or not two consecutively displayed patterns were the same (by pressing "J") or different (by pressing "L"; see Figure 5.22). As in the numerosity experiment, the presented functions came from the matched set (20 compositional, 20 non-compositional, sampled without noise) and were displayed on the screen as 100 horizontally equidistant gray dots. On each trial, participants saw the original structure for 1000ms, followed by an inter-stimulus interval (500ms) and then a test probe (1000ms). Unbeknownst to participants, the probe had a 50% chance of being the same or different. Change probes were constructed by randomly selecting $n \in \{2, 3, \cdots, 5\}$ dots and permuting them (under the constraint that no point ends up at the same position as before).

5.14.3 Results and discussion

Figure 5.23 shows the mean proportions of correct responses for compositional and non-compositional trials across different levels of change: "no change", "small change" (1-2 dots permuted) and "large change" (3-5 dots permuted). This analysis demonstrated that performance was superior for change of compositional structure.

Participants responded correctly on 82% of no-change trials for the compositional functions and on 81% of the no-change trials for the non-compositional functions. Thus, there was no significant difference in correct rejection rate for the two types of functions ($\chi^2(N = 1185, df = 1) = 0.21, p > 0.05$). However, 77% of the changed compositional functions were correctly identified as having changed, whereas that proportion was only 67% for the changed non-compositional functions. Therefore, change is more easily detected for compositional functions ($\chi^2(N = 1188, df = 1) = 12.13, p < 0.001$). In total, 49 of 66 participants correctly identified change of compositional functions more frequently than change of non-compositional functions ($\chi^2(N = 66, df = 1) = 9.67, p < 0.01$).

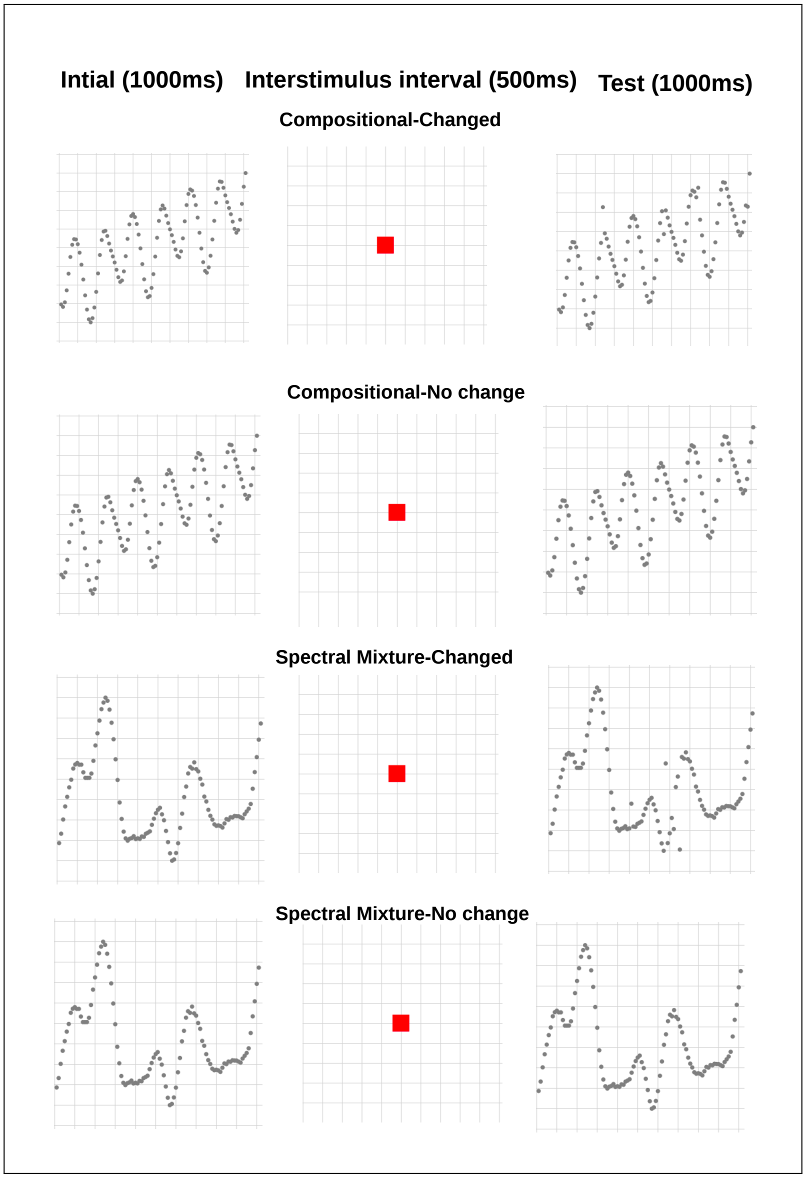


Figure 5.22: Schematic of change detection task (Experiment 8). Initial stimulus was presented for 1000ms, drawn either from the compositional or from the non-compositional (spectral mixture) set. After an inter-stimulus interval (500ms), a test probe was presented and participants had to make a same/different judgment.

In order to quantitatively capture these results, I put forward a Bayesian theory of functional change detection. Let $\mathcal{D}_1$ and $\mathcal{D}_2$ denote the first and second stimuli (input-output

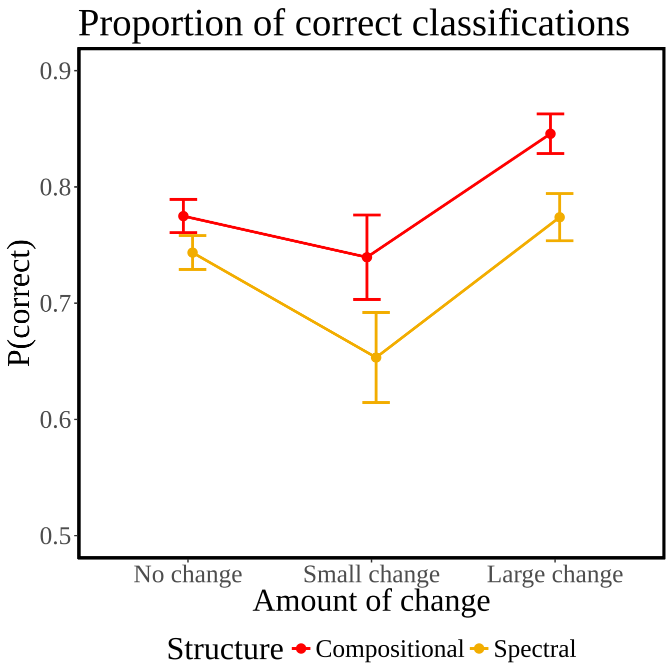


Figure 5.23: Results of Experiment 8. Proportion of correctly detected changes. Participants were more likely to detect changes in compositional functions. Error bars represent the standard error of the mean.

pairs), respectively, and let f_1 and f_2 denote the corresponding functions. The task facing participants is to determine the posterior probability that the two functions are different:

$$P(f_1 \neq f_2 | \mathcal{D}_1, \mathcal{D}_2) = \frac{P(\mathcal{D}_1, \mathcal{D}_2 | f_1 \neq f_2)}{P(\mathcal{D}_1, \mathcal{D}_2 | f_1 \neq f_2) + P(\mathcal{D}_1, \mathcal{D}_2 | f_1 = f_2)}, \tag{5.12}$$

where for simplicity we have assumed that the prior probability of change is 0.5. The change ($f_1 \neq f_2$) likelihood is given by:

$$P(\mathcal{D}_1, \mathcal{D}_2 | f_1 \neq f_2) = \int_{f_1} P(\mathcal{D}_1 | f_1) P(f_1) df_1 \int_{f_2} P(\mathcal{D}_2 | f_2) P(f_2) df_2, \tag{5.13}$$

and the no-change ($f_1 = f_2$) likelihood is given by:

$$P(\mathcal{D}_1, \mathcal{D}_2 | f_1 = f_2) = \int_f P(\mathcal{D}_1 | f) P(\mathcal{D}_2 | f) P(f) df. \tag{5.14}$$

This probabilistic model assumes that functions are encoded by a given kernel in short term memory and can make trial-by-trial predictions of performance on the change detection task. As in the other analyses, hyper-parameters were optimized with respect to the marginal likelihood for a given kernel. For the spectral kernel, the number of mixture components was treated as a free hyper-parameter (ranging from 1 to 6).

To allow for noise in the decision process, the binary response was modeled using a logistic function of the model predictions, with separate predictors for compositional and non-compositional predictions. The results of this logistic regression are summarized in Table 5.6. The compositional model was a significant predictor of human change detection performance ($\beta = 0.0122, p < 0.001$), whereas the spectral mixture model was only marginally predictive ($p = 0.05$).

Table 5.6: Result of change detection logistic regression.

	Estimate	Std. Error	z value	Pr(>|z|)
Intercept	1.3018	0.0918	14.19	0.0000
Compositional prediction	0.0122	0.0006	19.61	0.0000
Spectral mixture prediction	0.0012	0.0006	1.95	0.0517

To show this result in a different way, I computed the point biserial correlation between the model predictions and human responses (Figure 5.24). The average correlation per participant for the compositional model was significantly higher than the average correlation for the spectral mixture model ($r = 0.68$ vs. $r = 0.15$, $t(132) = 20.5, p < 0.001, d = 3.54$).

In summary, change is more easily detected in compositional functions compared to

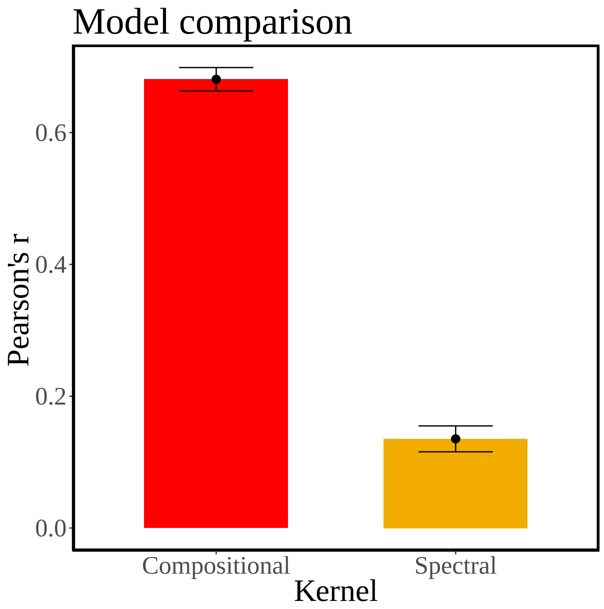


Figure 5.24: Mean point biserial correlation between model predictions and participants' responses in the change detection task. Error bars represent the standard error of the mean.

non-compositional functions, consistent with the idea that compositional functions are more efficiently encoded into a summary representation. Moreover, a GP theory of change detection, which equates compressing a function with storing it in short-term memory and thereby builds up a link between compressibility and generalization, showed that a compositional kernel allows to quantitatively predict human change detection performance.

5.15 Experiment 9: Compositional chunking in short-term memory

The results of Experiment 8 suggested that change detection is facilitated by compositional summary representations. This has introduced the idea is that structural regularities lead to higher generalizability which in turn increases memory capacity due to making functions more compressible (Brady et al., 2011; Mathy & Feldman, 2012). This also means that if multiple items can be chunked together, then a greater number of items can be stored in memory (Miller, 1956). Chunking has been posited as the basis of exceptional expert memory (Chase & Simon, 1973; Gobet, 1998) and story comprehension (Thorndyke, 1977).

The next experiment pursues this idea further, asking whether compositional functions are more compressible (and hence more memorable) than non-compositional functions. In order to do so, a standard short-term memory task is applied (the Sternberg paradigm; Sternberg et al., 1966), in which participants are shown a rapid sequence of items (functions) followed by an old/new judgment of a probe item. Additionally, this task allows to measure the interplay between compositionality and set size.

5.15.1 Methods

Participants

133 participants (mean age=31.05, SD=8.19, 71 male) were recruited via Amazon Mechanical Turk and received $0.5 for their participation. The experiment took 9 minutes on average.

Procedure

Participants were shown between 2 and 6 functions sampled randomly from the matched set. Each function appeared on the screen for 1000ms. A 500ms inter-trial interval succeeded the final item, followed by a probe item. Participants were asked to judge as quickly as possible whether the probe item was old (i.e., appeared in the preceding set) or new. There were 15 trials in total (3 trials for each set size). The probe was randomly selected to be either compositional or non-compositional, and probes were old on half of the trials. A schematic of the experiment is shown in Figure 5.25.

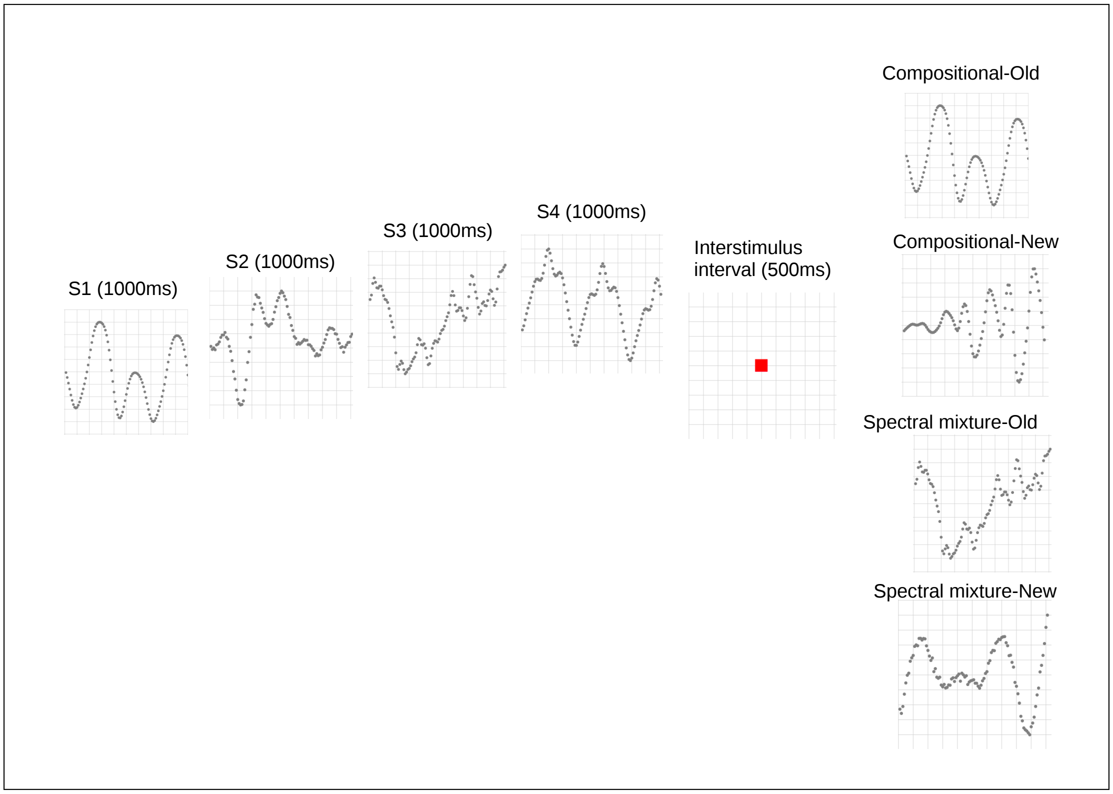


Figure 5.25: Schematic of Experiment 9. A sequence of stimuli was sampled from the matched set of functions. Every stimulus was presented for 1000ms, followed by an inter-stimulus interval (500ms), and then a probe (speeded old/new judgment).

5.15.2 Results and discussion

23 failed to respond correctly for more than half of the trials and were therefore excluded from further analysis. All trials that took longer than 5 seconds or less than 500ms were removed prior to analyzing the data.

Participants responded correctly on 77.8% of the compositional probes and on 66.7% of the non-compositional probes—a significant difference in accuracy ($\chi^2(N = 1456, df = 1) = 14.252, p < 0.001$). 71 of 110 participants responded correctly more frequently to the compositional probes than to the non-compositional probes ($\chi^2(N = 110, df = 1) = 9.3$,

$p < 0.01$).

This result can be decomposed further into a 75% hit rate for compositional probes, compared to a 67% hit rate for the non-compositional probes (a significant difference between hit rates: $\chi^2(N = 709, df = 1) = 5.19, p < 0.05$). 78 of 110 participants showed a higher hit rate for compositional probes as compared to non-compositional probes ($\chi^2(N = 110, df = 1) = 19.2, p < 0.01$).

Finally, there was also a significant difference between the correct rejection rate (78% for compositional probes vs. 65% for non-compositional probes; $\chi^2(N = 747, df = 1) = 8.87$, $p < 0.01$). 76 of 110 participants showed a higher correct rejection rate for compositional probes as compared to non-compositional probes ($\chi^2(N = 110, df = 1) = 17.92, p < 0.01$).

To disentangle compositionality and set size, a logistic regression was run to predict the probability of a correct response from compositionality (i.e., compositional vs. non-compositional probe) and set size factors. As shown in Table 5.7, the probability of responding correctly decreases with set size ($\beta = -0.22$), and compositional probes are more likely to be correctly identified ($\beta = 1.24$). Moreover, there was a significant interaction effect, whereby the compositional advantage decreases with set size ($\beta = -0.18$; Figure 5.26). This might be due to an increased guessing rate at a higher level of difficulty that potentially obscures the compositional advantage.

Table 5.7: Results of logistic regression analysis of the memory experiment.

	Estimate	Std. Error	z-value	Pr(>|z|)
Intercept	2.04	0.29	7	<2.57e-12
Set size	-0.22	0.06	-3.33	<0.001
Compositional	1.24	0.37	3.33	<0.001
Set size × Compositional	- 0.18	0.08	2.2	0.02

I next developed a Bayesian model of performance in the task, adapting the same basic

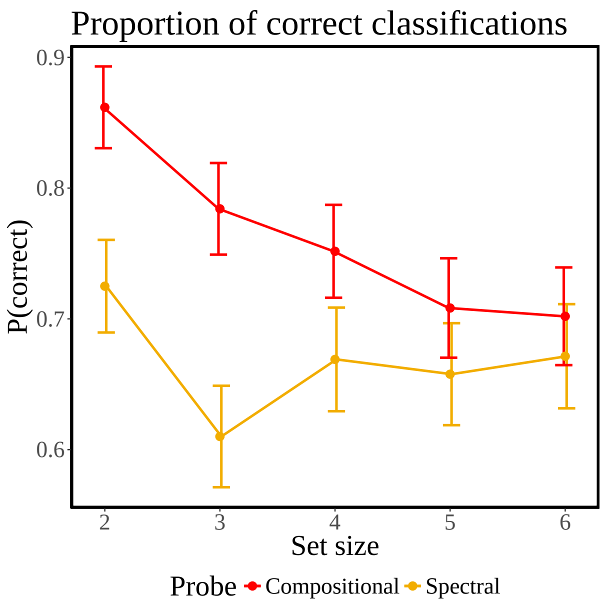


Figure 5.26: Proportion of correctly identified probes as a function of set size. Error bars represent the standard error of the mean.

framework that was applied to the change detection task. A participant is exposed to a study list (sequence of independent input-output datasets), denoted by $\mathcal{D}_{1:N}$, generated by underlying functions $f_{1:N}$. The task is to compute the posterior probability that a new dataset (the test probe), denoted by $\mathcal{D}'$, was generated by one of the functions in the study list:

$$P(f^* \in f_{1:N}|Y) = \sum_{n=1}^{N} P(f^* = f_n|\mathcal{D}_n, \mathcal{D}') \tag{5.15}$$

$$\propto \sum_{n=1}^{N} P(f^* = f_n)P(\mathcal{D}_n, \mathcal{D}'|f^* = f_n). \tag{5.16}$$

Following the structure of the experiment, I will assume that the probability of an “old” trial, $\sum_{n=1}^{N} P(f = f_n)$, is 0.5. The marginal likelihood is given by:

$$P(\mathcal{D}_n, \mathcal{D}'|f = f_n) = \int_f P(\mathcal{D}_n|f_n = f)P(\mathcal{D}'|f = f)P(f)df. \tag{5.17}$$

The general optimization procedure is the same as described in the change detection model.

The models' predictions for both the compositional and spectral mixture models were entered into a logistic regression to predict participants' old/new responses. Results of the logistic regression analysis are summarized in Table 5.8. Only the compositional model was a significant predictor of responses ($\beta = 0.054, p < 0.001$), whereas the spectral mixture model was not a significant predictor ($\beta = 0.01, p = 0.07$).

Table 5.8: Results of logistic regression analysis of the memory experiment.

	Estimate	Std. Error	z value	Pr(>|z|)
Intercept	1.4314	0.2147	6.67	0.0000
Compositional prediction	0.0543	0.0098	5.53	0.0000
Spectral mixture prediction	0.0176	0.0096	1.83	0.0676

Finally, I calculated the correlation between each model's predictions and participants' responses. As shown in Figure 5.27, the compositional model produced a superior correlation compared to the non-compositional model ($r = 0.35$ vs. $r = 0.09$; $t(262) = 9.97$, $p < 0.001$, $d = 0.34$). Moreover, assessing two more heuristic models, neither the mean squared distance between the probe and all functions in the set ($r = 0.094$), nor counting the number of types within the set and then stating new if the probe is of the same type as most functions within the set and old otherwise ($r = 0.04$) lead to higher correlations with participants' responses.

In conclusion, compositional functions are more easily remembered than non-compositional functions in a version of the Sternberg task. Participants' old/new judgments were well-

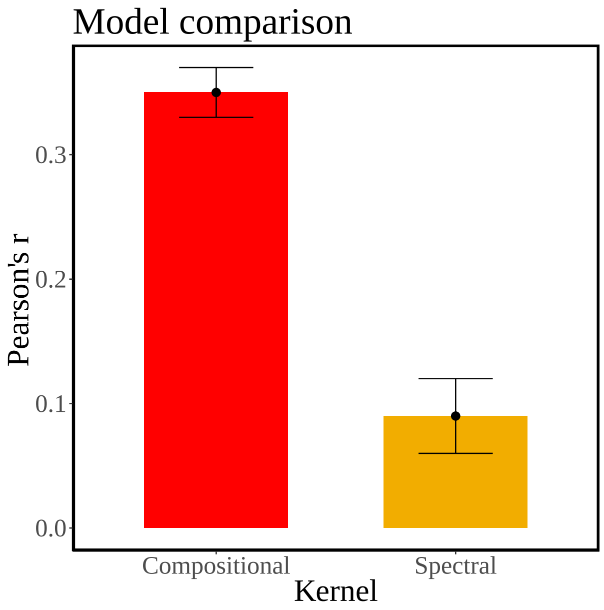


Figure 5.27: Mean point biserial correlation for the compositional and the spectral mixture models. Error bars represent the standard error of the mean.

captured by a Bayesian model of short-term memory that used a compositional kernel. This Bayesian model was built under the assumption that a model that allows that assumes a particular function generalizes better than another, also manages to store this function more efficiently in short-term memory.

5.16 Discussion

In this chapter, I pursued the hypothesis that humans have functional inductive biases that are compositionally structured—that is, humans prefer to represent complex functions as compositions of simpler building blocks. I formalized this idea using a compositional kernel within a Gaussian Process regression framework. Human inductive biases were assessed across a diverse range of experiments. The first set (Experiments 2, 3, 4 and 6) attempted to directly measure inductive biases using extrapolation and interpolation judgments, finding that participants preferred compositional over non-compositional pattern completions. The second set of experiments examined the broader implications of compositionality, finding that compositional functions are perceived as more predictable (Experiment 5) and memorable (Experiment 9) compared to non-compositional functions. Furthermore, discrete displays of items are perceived as less numerous (a signature of statistical regularities; Experiment 7), and changes in such displays are more easily detected (Experiment 8). Taken together, these experimental findings provide strong support for the compositional hypothesis.

5.16.1 Related work

The work presented here is connected to several lines of previous work. Most relevant are the seminal work of Griffiths et al. (2009) and Lucas et al. (2015) on Gaussian processes for function learning in general, and Wilson et al. (2015) more recent attempts to reverse-engineer the human kernel using a non-parametric kernel in particular. The work presented in this chapter is complementary to this research. If one wants to find out how people perceive and reason about functional structure, then one needs both a bottom-up theory to describe how people make sense of structure as well as a top-down method to indicate what the final structure might look like when represented as a kernel. Additionally, implementing a structure search algorithm as a parse tree, as I have done here, has recently been shown to be statistically efficient (Xu et al., 2017).

The approach described here also fits snugly with past attempts to model compositional structure in other cognitive domains. Of course, language (e.g., Chomsky, 1957) and object perception (e.g., Biederman, 1987) have long traditions of emphasizing compositionality. More recently, these ideas have been extended to other domains. For example, Gershman et al. (2015b) showed how hierarchical motion perception could be understood as a kind of vector analysis, using compositional Gaussian processes to model the combination of motion flow fields. This approach has also been applied to decision making; Gershman et al. (2017) used GPs to model utility functions over tree-structured objects (e.g., meals in a restaurant). In both motion perception and decision making, simpler non-compositional models failed to explain human performance.

5.16.2 Compositional generalization

The experiments described in this chapter establish the importance of compositional representations for functional pattern recognition and generalization. As the framework was restricted to a small number of components primarily for practical purposes, an important direction for future work will be to systematically investigate the boundaries of function composition. Fortunately, the GP formalism can accommodate a wide variety of compositional structures (Duvenaud et al., 2013), so I expect that this task can be accomplished without deviating too far from the analytical framework I have laid out here.

This work has probed in what sense (if any) human function learning can be seen as compositional within a single and sequential function learning tasks. However, composing structures learned across different tasks could provide even more powerful learning strategies across tasks. Future work could investigate the composition of structure across tasks (as for example shown by Hwang et al., 2016), for example in one-shot learning or trial-by-trial function learning experiments.

There are many other potential implications of compositional inductive biases for functions. For example, these inductive biases could shape active learning (i.e., tasks in which participants can choose the next data point). We believe that active learning model comparisons constitute an interesting framework to pit the structured and unstructured approaches against each other (Parpart et al., 2015).

Related issues arise in reinforcement learning tasks, where agents must balance exploration and exploitation. Here, the compositional approach could be used to design nonparametric value function approximators, which have been proposed as cognitively and neurally plausible solutions to reinforcement learning problems (Gershman & Daw, 2017). New experiments will be required to discern whether reinforcement learning exploits compositional inductive biases. However, in order to assess the plausibility of a compositional function learning approach to reinforcement problems, I first have to assess whether the Gaussian Process regression approach towards human function learning is amenable to self-directed function learning problems at all. This will be the focus of the next chapter.

“I tried to develop theories that took account of the uncertainty and the complexity in the world.”

Herbert Simon, 1978

6

Exploration and generalization in vast spaces

This chapter combines Gaussian Process regression with upper confidence bound sampling to model how participants explore and exploit unknown functions in spatially correlated multi-armed bandit tasks.

6.1 Introduction

The experiments so far have investigated how participants perceive, judge about, or predict unknown functions in *single or sequential function learning tasks*. However, function learning and generalization are also important for another domain, which is when the goal is to generate rewards, i.e. in *function exploration-exploitation tasks*.

Consider having just arrived at a new place and having to decide where to dine tonight. If it is a big city, then there are likely many different restaurants available and quite possibly more than you could ever try during your stay or even –if you are in a place like London, Berlin, or New York– during a life time. How do you decide where to dine tonight?

Fortunately, this might not be the first time you have to make this kind of decision. Instead, you have likely tried out different restaurants in the past and you know what you like and are confident about what kind of experience you want. Additionally, different restaurants also come with different features such as the cuisine on offer, their location, their price, their rating on different travel websites, and so forth, and you can use these features to make a prediction about a restaurant's culinary quality and the kind of experience you expect when dining there. This not only means that you can use this information to make your decision, it also means that a situation in which you would otherwise have to explore a vast number of different restaurants in order to learn about their underlying qualities and cuisine suddenly becomes much more tractable as you utilize a function that maps features onto expected rewards. Learning a mapping between features and expected outcomes whilst making decisions that both explore and exploit the underlying function is the topic of this chapter.

I investigate how people search for rewards in complex grid worlds. These grid worlds can be seen as a multi-armed bandit problem where every tile of the grid is an arm that can be sampled in order to produce and accrue an underlying reward. It will generally be participants' task to find rewards over multiple trials by sampling inputs that produce high outcomes. Within these tasks, inputs that are proximal to each other produce similar outputs, i.e. rewards are spatially correlated. As this results in similar rewards between nearby tiles, spatial correlation provides a context that can be utilized via function learning. Learning this underlying function while searching for rewards simplifies the task as one does not have to sample each option individually in order to make good decisions. Instead, learning a function enables *generalization* broadly beyond the visited places. This requires to sample from both currently less certain options in order to learn more about the underlying function and options that are promising to produce high reward. This frames the *exploration-exploitation* trade-off, which is typically studied using the multi-armed bandit framework (Auer et al., 2002). The multi-armed bandit is a metaphor for a row of slot machines in a casino, where each slot machine has an independent payoff distribution. Solutions to the problem propose different policies for how to learn about which arms are better to play (exploration), while also efficiently playing known high-value arms to maximize reward (exploitation) (Steyvers et al., 2009; Bechara et al., 1997), with the common assumption that each arm of the bandit has its own reward distribution that needs to be learned independently.

In the real world, it is often infeasible to exhaustively explore all possibilities (Gershman & Daw, 2017) and still humans and other animals are able to quickly learn in and adapt to unfamiliar environments, where the exact same situation is rarely encountered twice (Lee et al., 2014). This is in contrast to more traditional approaches to *associative learning*, which assume a finite set of states and actions, and then independently learn about the distribution of rewards for each state (Sutton & Barto, 1998). This approach falls short in more realistic scenarios with vast state spaces or a limited search horizon, where it is impossible to observe the outcomes of all possible states and actions (Lake et al., 2016; Wilson et al., 2014).

I propose that *function learning* provides a mechanism to generalize prior experience to unobserved states (Tesauro, 1992). The function learning approach relates different state-action contexts to each other by approximating a global value function over all contexts, including unobserved outcomes (Gershman & Daw, 2017). This allows for generalization to vast and potentially infinite state spaces, based on a small, finite number of observations. Additionally, the function learning approach can also scale to problems with complex sequential dynamics (e.g., games such as Go), and has been used in tandem with *restricted search* methods such as Monte Carlo tree search for navigating intractably large search trees (Silver et al., 2016). While restricted search methods have been proposed as models of human reinforcement learning (Huys et al., 2015; Solway & Botvinick, 2015), here I focus on situations where a rich model of the structure of the environment can simplify planning and generalization (Guez et al., 2013).

I model this behavior using a Gaussian Process regression framework paired with different sampling strategies that trade off between exploration and exploitation. In two experiments, I show that people's search behavior is predicted well by a Gaussian Process function learning model paired with an optimistic sampling strategy. I conclude by arguing that participants' behavior in such tasks is surprisingly adaptive and that function approximation equips them with a powerful tool for generalization guiding them through complex spaces.

6.2 Spatially correlated multi-armed bandits

The multi-armed bandit can be easily adapted to a function learning setting by adding spatial correlation to rewards and placing the arms either into a one- or a two-dimensional grid. Each tile represents a playable arm of the bandit, which are initially blank and display the numerical reward value after an arm has been chosen. Traditionally, the goal in an multi-armed bandit task is to maximize cumulative payoffs by sequentially choosing one of the *N*-arms of the bandit that stochastically generate rewards (Steyvers et al., 2009), with learning happening independently for each arm. In the case applied here, because proximate arms generate similar rewards, there is an opportunity to form inductive beliefs about unobserved rewards by learning about an underlying value function. This allows to study how people generate beliefs about unobserved rewards and how this generalization influences their search behavior.

The spatially correlated multi-armed bandit is related to the optimal foraging context (Krebs, 1980), whereby a forager is not only guided by the search for resources, but also by the need to acquire information about the distribution of resources in the environment. This creates a natural trade-off between exploration and exploitation (March, 1991), where an effective search policy needs to adequately balance exploring areas with higher uncertainty, while also exploiting existing information to obtain rewards. One key difference in the presented task is that the decision-maker must determine *where* to search, and not only whether to stay or to leave a patch.

To describe participants' behavior in such tasks, both a model of learning and a sampling strategy for where to search next are required. An example of how these two might work together is shown in Figure 6.1.

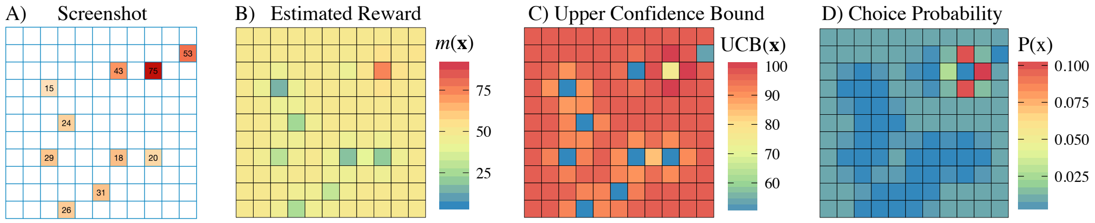


Figure 6.1: Overview of the Gaussian Process function learning model, combined with an upper confidence bound sampling strategy (specified using median participant parameter estimates from Experiment 11. **A)** Screenshot of Experiment 11. Participants were allowed to select any tile until the search horizon was exhausted. **B)** Estimated reward as predicted by a Gaussian Process function learning engine, based on the sampled points in Panel A. (Not shown, the estimated uncertainty). **C)** Upper confidence bound of predicted rewards. **D:** Choice probabilities after a softmax choice rule, $P(\mathrm{x}) = \exp(\text{UCB}(\mathrm{x})/\tau)/\sum_{j=1}^{N}\exp(\text{UCB}(\mathrm{x}_j)/\tau)$, where τ is the temperature parameter (i.e., lower temperature values lead to more precise predictions).

6.3 Models of learning in complex spaces

I will assess two kinds of models to describe how people learn within spatially-correlated bandit tasks . The *function learning model* uses the spatial correlation between rewards in order to learn about an underlying value function mapping inputs to expected rewards and their attached uncertainty. It is able to generalize from sampled to unexplored options. The *associative learning model* does not learn about an underlying function, but rather learns about each input individually by associating inputs with previously generated rewards. It is not able to generalize to unexplored options.

6.3.1 Function learning model

The function learning model is set up as a type of Gaussian Process regression. Let $f : \mathcal{X} \to \mathbb{R}$ denote a function over input space $\mathcal{X}$ that maps to real-valued scalar outputs. As described in Chapter 3, the function can be modeled as a random draw from a Gaussian process:

$$f \sim \mathcal{GP}(m, k), \tag{6.1}$$

where m is a mean function specifying the expected output of the function given input x, and k is a kernel (or covariance) function specifying the covariance between outputs. A common choice of kernel function when exploring and exploiting unknown functions is the radial basis function kernel.

$$k(\mathrm{x}, \mathrm{x}') = \sigma_f^2 \exp\left(-\frac{|\mathrm{x} - \mathrm{x}'|^2}{2\lambda^2}\right), \tag{6.2}$$

where λ governs how quickly correlations between points x and x$'$ decay towards zero as their distance increases. The length-scale λ will be treated as a free parameter to estimate the extent to which people generalize between different points in our model comparison procedure, with smaller values indicating generalization over shorter spatial extents.

This model learns a function that maps locations, for example in a grid or an array of arms of a bandit, onto expected rewards and their attached uncertainties. It assumes that rewards are generated by an underlying function

$$y_{i,t} = f(x_{i,t}) + \varepsilon_t \tag{6.3}$$

with noise $\varepsilon_{i,t} \sim \mathcal{N}(0, \sigma_\varepsilon^2)$.

The Gaussian Process function learning model manages to predict outcomes for a broad space of both sampled and not yet observed arms and thus provides a powerful mechanism for generalization.

6.3.2 Associative learning model

The associative learning model is set up as a simple mean tracking approach towards learning about the distribution of rewards for each arm individually. This model assumes that the average reward associated to an input is constant over time (i.e., no temporal dynamics), which is the case in the search task. Thus, the associative learning model computes a time-invariant posterior distribution for the mean μ_j of each option j including its variance σ_j. The implemented version assumes that rewards are normally distributed with a known variance but unknown mean

$$y_{i,t} \sim \mathcal{N}(\mu_i, \sigma^2_\epsilon) \tag{6.4}$$

where the prior distribution of the mean is again a normal distribution. This implies that the posterior distribution for each mean is also a normal distribution:

$$p(\mu_j|\mathcal{D}_{t-1}) = \mathcal{N}(\mu_{j,t-1}, \sigma^2_{j,t-1}) \tag{6.5}$$

The estimated mean $\mu_{j,t}$ and variance $\sigma^2_{j,t}$ for a given option j are only updated when selected at trial t:

$$\mu_{j,t} = \mu_{j,t-1} + \delta_{j,t}G_{j,t}\left[y_t - \mu_{j,t-1}\right] \tag{6.6}$$

$$\sigma^2_{j,t} = \left[1 - \delta_{j,t}G_{j,t}\right]\sigma^2_{j,t-1} \tag{6.7}$$

where $\delta_{j,t} = 1$ if option j was chosen on trial t, and 0 otherwise. Intuitively, the estimated mean of the chosen option $\mu_{j,t}$ is updated based on the difference between the observed value y_t and the expected mean $\mu_{j,t-1}$, multiplied by $G_{j,t}$. At the same time, the estimated variance $\sigma^2_{j,t}$ is reduced by a factor of $1 - G_{j,t}$, where $G_{j,t}$ is defined as:

$$G_{j,t} = \frac{\sigma^2_{j,t-1}}{\sigma^2_{j,t-1} + \vartheta^2}, \tag{6.8}$$

where ϑ^2 is the error variance, which is estimated as a free parameter.

Intuitively, the estimated mean of the chosen option $\mu_{j,t}$ is updated based on the difference between the observed value y_t and the expected mean $\mu_{j,t-1}$, multiplied by $G_{j,t}$. At the same time, the estimated variance $\sigma_{j,t}$ is reduced by a factor of $1 - G_{j,t}$, which is in the range $[0, 1]$. The error variance (ϑ^2) can be interpreted as an inverse sensitivity, where smaller values result in more substantial updates to the mean $\mu_{j,t}$, and larger reductions of uncertainty $\sigma_{j,t}$. The prior mean of the mean value of payoffs is set to $\mu_{j,0} = 50$ and the prior variance to $\sigma^2_{j,0} = 500$

6.4 Adaptive sampling strategies

I consider various computational models for describing human sampling strategies. All of these strategies make sequential predictions about where people are likely to search for rewards. As both learning models generate normally distributed predictions about the expectation $\mu(\mathrm{x})$ and the uncertainty $\sigma(\mathrm{x})$ for each arm, which are available to the sampling strategies for evaluating the quality, $q(\mathrm{x})$, and ultimately making a prediction about where to sample next, i.e. which arm participants are likely to sample on a given trial.

The *Variance Greedy* (VG) strategy values an arm using only the estimated uncertainty

$$q_{VG}(\mathrm{x}) = \sigma(\mathrm{x}) \tag{6.9}$$

and is an efficient step-wise (greedy) approximation of *information gain* (Srinivas et al., 2010), which seeks to learn the global value function as rapidly as possible. VG achieves at least a constant fraction of the optimal information gain value (Krause & Guestrin, 2005); however, it fails to adequately trade-off between exploration and exploitation, because effort is wasted exploring the function where $f(\mathrm{x})$ is small and only learning about the function is not the objective of the task, but rather to both explore and exploit the function over time.

The *Mean Greedy* (MG) strategy is also step-wise greedy, valuing arms using only the estimated mean reward

$$q_{MG}(\mathrm{x}) = \mu(\mathrm{x}) \tag{6.10}$$

although this strategy carries no known guarantees and is prone to getting stuck in local optima.

Upper confidence bound sampling (UCB) as described in Chapter 3 combines the VG and MG strategies

$$q_{UCB}(\mathrm{x}) = \mu(\mathrm{x}) + \beta\sigma(\mathrm{x}) \tag{6.11}$$

where the exploration factor β determines how the reduction of uncertainty trades off against exploiting high expected rewards. This is sometimes referred to as optimistic "sampling with confidence" as it inflates expectations with respect to the upper confidence bounds (Srinivas et al., 2010), creating a natural balance between exploration and exploitation. The parameter β will be optimized as a free parameter.

There are also various other sampling strategies that combine expectations and uncertainties but in other ways than the upper confidence bound sampling strategy described in Chapter 3.

The *Expected Improvement* (EXI) evaluates each option by *how much* (in expectation) it promises to be better than the best outcome ($\mathrm{x}^+ = \arg\max_{\mathrm{x}_i \in \mathrm{x}_{1:t}} f(\mathrm{x}_i)$) observed so far:

$$q_{EXI}(\mathrm{x}) = \begin{cases} \Phi(Z)(\mu(\mathrm{x}) - f(\mathrm{x}^+)) + \sigma(\mathrm{x})\varphi(Z), & \text{if } \sigma(\mathrm{x}) > 0 \\ 0, & \text{if } \sigma(\mathrm{x}) = 0 \end{cases} \tag{6.12}$$

where $\Phi(\cdot)$ is the normal CDF, $\varphi(\cdot)$ is the normal PDF, and $Z = (\mu(\mathrm{x}) - f(\mathrm{x}^+))/\sigma(\mathrm{x})$.

The *Probability of Improvement* (POI) strategy evaluates an option based on *how likely* it will be better than the best outcome (x^+) observed so far:

$$q_{POI}(\mathrm{x}) = P\left(f(\mathrm{x}) \geq f(\mathrm{x}^+)\right) = \Phi\left(\frac{\mu(\mathrm{x}) - f(\mathrm{x}^+)}{\sigma(\mathrm{x})}\right) \tag{6.13}$$

where $\Phi(\cdot)$ is the normal CDF. This sampling strategy assesses the probability of one option to generate a higher utility than the best option observed so far. Similar strategies have been used in experiments involving multi-attribute choices before (Gershman et al., 2017).

The *Probability of Maximum Utility* (PMU, Speekenbrink & Konstantinidis, 2015) samples each option according to the probability that it results in the highest reward of all options in a particular context.

$$q_{PMU}(\mathrm{x}) = P\left(y_j \geq y_i, \forall i \neq j\right) \tag{6.14}$$

This sampling strategy will be implemented by Monte Carlo sampling from the posterior predictive distribution of a learning model for each option, and evaluating how often a given option turns out to be the maximum over 1,000 generated samples. It is a form of probability matching and can be implemented by sampling from each option's predictive distribution once, and then choosing the option with the highest sampled pay-off. This strategy can be seen as a form of probability matching (Neimark & Shuford, 1959) and can be implemented by sampling from each option's predictive distribution once, and then choosing the option with the highest sampled pay-off. Even though this sampling strategy seems relatively simple, it describes human choices in restless bandit tasks well (Speekenbrink & Konstantinidis, 2015). It is also closely related to Thompson sampling (May et al., 2012), which samples from the posterior distribution of the mean rather than the predictive distribution of rewards. Thus, while Thompson sampling "probability matches" the expected rewards of each arm, the probability of maximum utility rule matches to actual rewards that might be obtained.

All possible combinations of the learning models and sampling strategies will be compared by how well they predict participants' behavior within different versions of the spatially-correlated multi-armed bandit.

6.4.1 Heuristic strategies

Local search predicts that search decisions have a tendency to stay local to the previous choice. This can be incorporated by using the inverse Manhattan distance (IMD) to quantify locality:

$$\mathrm{IMD}(\mathrm{x}, \mathrm{x}') = \sum_{i=1}^{n} |x_i - x'_i|, \tag{6.15}$$

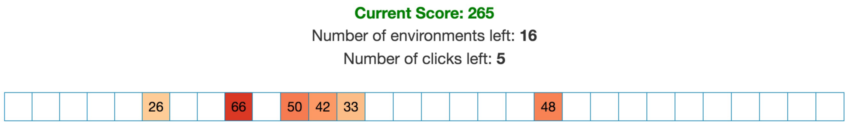
where x and x′ are vectors in $\mathbb{R}^n$. For the special case where $\mathrm{x} = \mathrm{x}'$, I set $\mathrm{IMD}(\mathrm{x}, \mathrm{x}') = 1$.

With the exception of the local search model itself, all other models can also be localized. Localization can be introduced by weighting a model's predicted values of each option $q(\mathrm{x})$ by the inverse Manhattan distance (IMD) to the previously revealed tile. This is equivalent to a multiplicative combination with the Local Search model, without the introduction of any additional free parameters. Localized models will be indicated with an asterisk (e.g., function learning*).

A form of a win-stay lose-sample (WSLS) heuristic (Bonawitz et al., 2014) is implemented, where a *win* is defined as finding a payoff with a higher or equal value than the previous best. When the decision-maker "wins", it is assumed that any tile with a Manhattan distance ≤ 1 is chosen (i.e., a repeat or any of the four cardinal neighbors) with equal probability. *Losing* is defined as the failure to improve, and results in choosing any unrevealed tile with equal probability.

6.5 Experiment 10: Univariate function exploration-exploitation

The first variant of the spatially-correlated multi-armed bandit presents participants with a uni-variate multi-armed bandit problem with spatially correlated rewards. The problem space is represented by a one-dimensional array of 1×30 possible arms, i.e. 30 different options to sample. Participants can click to reveal unexplored tiles or re-click previously uncovered tiles to further explore or exploit known rewards. A screenshot of Experiment 10 is shown in Figure 6.2.



Current Score: 265

Number of environments left: **16**

Number of clicks left: **5**

Figure 6.2: Screenshot of Experiment 10. Univariate spatially correlated multi-armed bandit.

6.5.1 Methods

Participants

81 participants were recruited from Amazon Mechanical Turk (25 Female; mean age= 33 ± 11). Each participant was paid a participation fee of \$0.50 and a performance contingent bonus up to \$1.50. On average, participants earned \$1.14 ± 0.13 and spent around 8 minutes on the task.

Design

A 2×2 between subject design was used, where participants were randomly assigned to one of two different pay-off conditions and one of two different classes of environments. Each grid represented a uni-variate function, with each observation including normally distributed noise, $\varepsilon \sim \mathcal{N}(0, 1)$. The task was presented over 16 blocks on different grid worlds drawn from the same class of environments. In each block, participants had either a *Short* (5 clicks) or *Long* (10 clicks) search horizon to interact with the grid. The search horizon alternated between blocks (within subject), with initial horizon length counterbalanced between subjects. Per block, observations were scaled to a uniformly sampled maximum value in the range of 65 to 85, so that the value of the global optima could not be easily guessed (e.g., a value of 100). Figure 6.3 shows the design of Experiment 10.

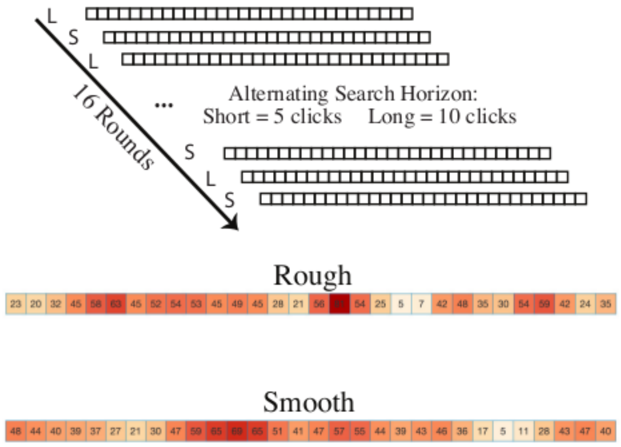


Figure 6.3: Design of Experiment 10. Environments could be either smooth or rough (between subject manipulation). Sample horizons could be either short (5 clicks) or long (10 clicks, within subject manipulation, counter-balanced).

Materials and procedure

Before starting, participants were shown four fully revealed grids in order to familiarize themselves with the task. Example environments were drawn from the same class of environments assigned to the participant (Smooth or Rough) and underwent the same random scaling of observations. Additionally, three comprehension questions were used to ensure full understanding of the task.

At the beginning of each of the 16 blocks, one random tile was revealed and participants could use their mouse to click any of the 30 tiles until the search horizon was exhausted, including re-clicking previously revealed tiles. Clicking an unrevealed tile displayed the numerical value of the reward along with a corresponding color aid, where darker colors indicated higher point values. Previously revealed tiles could also be re-clicked, although there were variations in the observed value due to noise. For repeat clicks, the most recent observation was displayed numerically, while hovering over the tile would display the entire history of observations. The color of the tile corresponded to the mean of all previous observations.

Payoff conditions

Performance will be compared under two different payoff conditions, requiring either a balance between exploration and exploitation (*Accumulators*) or a more exploration focused context of having to find the overall maximum without accruing rewards on every trial (*Maximizers*). Previous work has shown that people can adapt (sometimes with difficulty) to different payoff conditions in information acquisition tasks (Meder & Nelson, 2012).

In each payoff condition, participants received a performance contingent bonus of up to $1.50. Importantly, participants in the *Accumulator* group were told to "gain as many points as possible across all 16 grids" and were given a bonus based on the sum of their overall points. Participants in the *Maximizer* condition were told to "learn where the largest reward is" and were given a bonus using the ratio of the highest observed reward to the global optimum, $(\frac{\max y_t}{y^*})^4$, taken to the power of 4 to exaggerate differences in the upper range of performance and for parity in expected earnings across payoff conditions. All 16 blocks were weighted equally, using noisy but unscaled observations to assign a bonus of up to $1.50. Subjects were informed in dollars about the bonus earned at the end of each block. Repeated participation was not allowed.

Smoothness of the environment

Two different classes of environments were used, corresponding to different levels of smoothness (Fig. 6.2). All environments were sampled from a Gaussian Process prior parameterized with a radial basis function kernel, where the length-scale parameter (λ) determines the rate at which the correlations of rewards decay over distance. I sampled 20 *Smooth* environments ($\lambda = 2$) and 20 *Rough* environments ($\lambda = 1$). Subjects performed the task on 16 grids randomly drawn (without replacement) from their assigned class of environments, while the four fully revealed environments used to familiarize subjects with the task were the remaining 4 environments.

Search horizons

The length of the search horizon influences the value of information learned about the environment, with respect to the assigned payoff condition. Longer horizons provide more opportunities for exploiting acquired information, thereby making early exploration more valuable. I chose two horizon lengths (Short$=5$ and Long$=10$) that were fewer than the total number of tiles on the grid (total number of arms was 30), and varied within subject (alternating between blocks).

6.5.2 Results

Figure 6.4 shows task performance. In all conditions, performance improved as a function of the trial number (i.e., with each additional click), as measured by both the overall correlation between average reward and trial number ($r = 0.66, p < 0.001$) and between the maximum observed reward and trial number ($r = .91, p < 0.001$). There were no learning effects across blocks (i.e., over successive grids), indicated by a lack of correlation between average reward and block number ($r = -0.05, p > 0.5$), or between maximum reward and block number ($r = -0.05, p > 0.1$). Performance improved as more information was revealed (i.e., over trials), but not over additional blocks of identically parameterized environments.

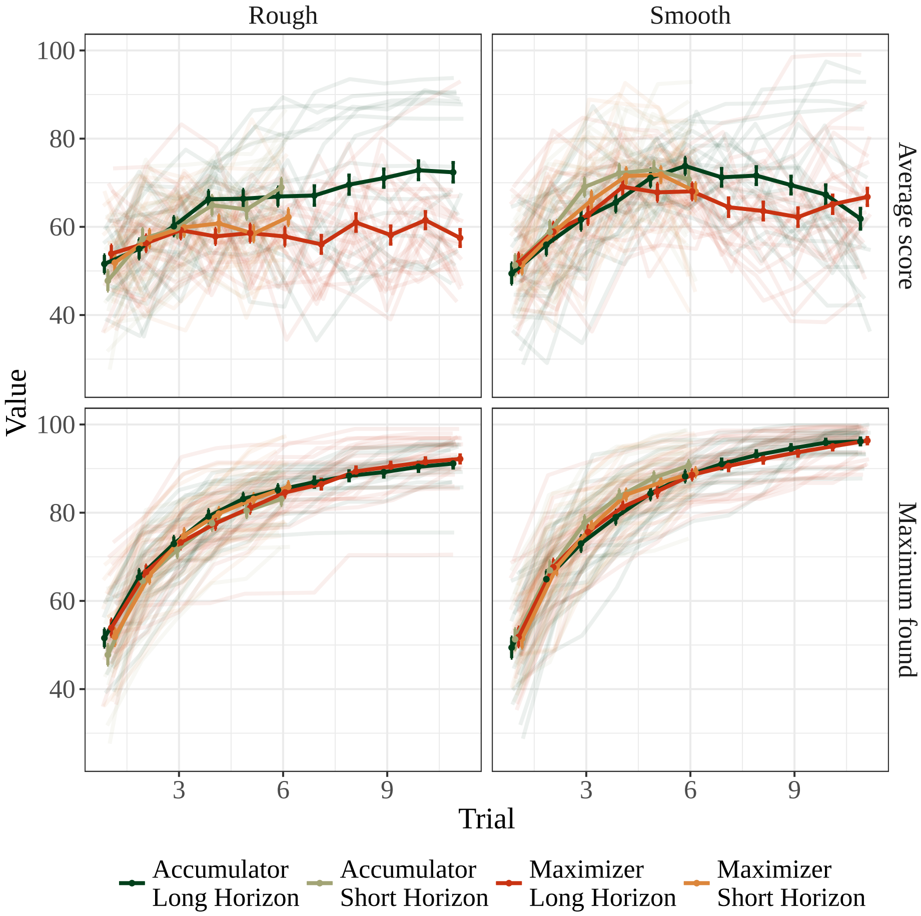


Figure 6.4: Results of Experiment 10.

Payoff conditions

Payoff conditions influenced search behavior, with participants in the Maximum Reward condition displaying more variance in the locations sampled ($t(79) = -3.26, p < 0.05$, $d = 0.73$). Participants in the *Accumulator* group achieved a higher average reward than participants in the *Maximizer* group ($t(79) = 2.89, p < 0.01, d = 0.65$). However, *Maximizers* did not reveal a higher maximum value than *Accumulators* ($t(79) = 0.28$,

$p > 0.7, d = 0.06$). Thus, a strategy balancing exploration and exploitation, at least for human learners, may find the global maximum *en passant*.

Environment and horizon

Independent of the payoff condition, participants assigned to Smooth environments achieved higher average rewards ($t(79) = 3.55, p < 0.001, d = 0.79$) and higher maximum rewards ($t(79) = 3.46, p < 0.001, d = 0.77$) than those assigned to the Rough environments, suggesting that stronger correlations of payoffs make the task easier. Interestingly, longer horizons did not lead to better overall performance in the Average Reward condition ($t(80) = 0.60, p > 0.5, d = 0.07$), although participants given longer horizons found larger maximum rewards for all payoffs and environment conditions ($t(80) = 9.75$, $p < 0.001, d = 1.08$). There may be a less-is-more-effect (Medvec et al., 1995), with longer horizons leading to over-exploration, given the goal of maximizing average rewards.

6.5.3 Cross-validation

The cross-validation procedure assesses all 27 models resulting from combining all models of learning with all sampling strategies and also comparing the heuristic strategies as well.

I use maximum likelihood estimation (MLE) for parameter estimation and cross-validation to measure out-of-sample predictive accuracy. A softmax choice rule transforms each model's prediction into a probability distribution over options:

$$p(\mathrm{x}) = \frac{\exp(q(\mathrm{x})/\tau)}{\sum_{j=1}^{N} \exp(q(\mathrm{x}_j)/\tau)}, \tag{6.16}$$

where $q(\mathrm{x})$ is the predicted value of each option x for a given model, and τ is a free temperature parameter. Lower values of τ indicate more concentrated probability distributions, corresponding to more precise predictions.

In general, for the Gaussian Process-based function learning models I estimate the length-scale, λ, of the radial basis function kernel and for the associative learning models σ^2_ϵ, the error variance. The upper confidence bound sampling strategy has a free parameter β, governing the exploration bonus. Additionally, all models include τ as a free parameter modeling the concentration of predictions within the softmax transformation.

All models were fitted—per participant—using cross-validated MLE, with either a Differential Evolution algorithm (Mullen et al., 2009) or a grid search if the model contained only a single parameter. Parameter estimates are constrained to positive values in the range $[\exp(-5), \exp(5)]$. Cross-validation is performed by first separating participant data according to horizon length, which alternated between rounds within subject. For each participant, half of the rounds corresponded to a short horizon and the other half corresponded to a long horizon. Within all rounds of each horizon length, leave-one-out cross-validation is used to iteratively form a training set by leaving out a single round, computing a MLE on the training set, and then generating out of sample predictions on the remaining round. This is repeated for all combinations of training set and test set, and for both short and long horizon sets for each participant individually. The cross-validation procedure will yield one set of parameter estimates per round, per participant, and out of sample predictions for 120 choices per participant. In total, cross validation required approximately 25,000 hours of computation, or about 2 days distributed across a 716 CPU cluster.

Prediction error (computed as log loss) is summed up over all rounds, and is reported as *predictive accuracy*, using a pseudo-R^2 measure that compares the total log loss prediction error for each model to that of a random model.

$$R^2 = 1 - \frac{\log \mathcal{L}(\mathcal{M}_k)}{\log \mathcal{L}(\mathcal{M}_{\text{rand}})}, \tag{6.17}$$

where $\log \mathcal{L}(\mathcal{M}_{\text{rand}})$ is the log loss of a random model (i.e., picking options with equal probability) and $\log \mathcal{L}(\mathcal{M}_k)$ is the log loss of model k's out-of-sample prediction error. Intuitively, $R^2 = 0$ corresponds to prediction accuracy equivalent to chance, while $R^2 = 1$ corresponds to theoretical perfect prediction accuracy, since $\log \mathcal{L}(\mathcal{M}_k) / \log \mathcal{L}(\mathcal{M}_{\text{rand}}) \to 0$ when $\log \mathcal{L}(\mathcal{M}_k) \ll \log \mathcal{L}(\mathcal{M}_{\text{rand}})$. This measure is similar to McFadden's pseudo-R^2 (McFadden, 1973), although it uses a completely random model $\mathcal{M}_{\text{rand}}$ as baseline.

Model comparison results

Results of the full model comparison for Experiment 10 are shown in Figure 6.5.

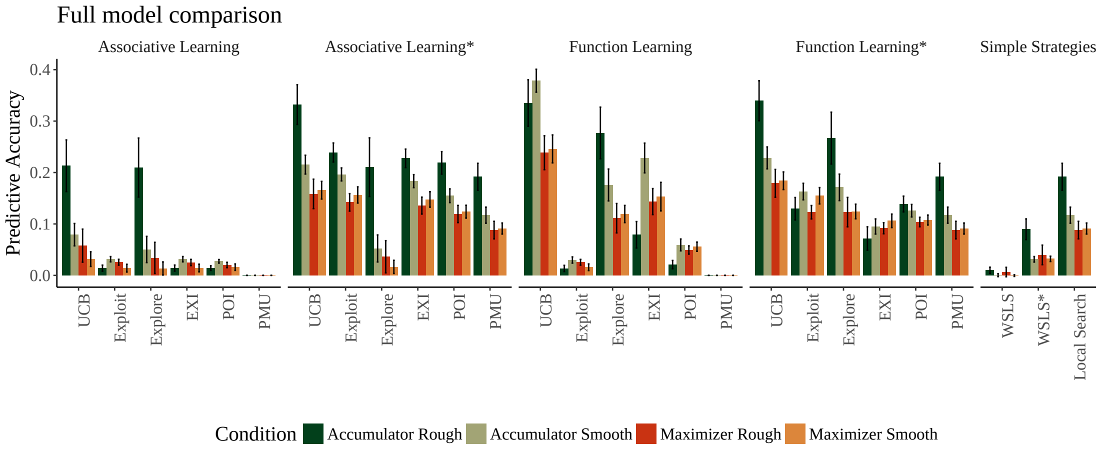


Figure 6.5: Model comparison results of Experiment 10.

The function learning models generally predicted participants' behavior better than the associative learning models ($t(80) = 15.81, p < 0.001, d = 1.76$) and better than the simple strategies ($t(80) = 9.22, p < 0.001, d = 1.02$). Within the function learning models, the upper confidence bound sampling strategy performed better than all other sampling strategies ($t(80) = 15.34, p < 0.001, d = 1.71$). Even though localization improved both the function learning model's ($t(80) = 5.05, p < 0.001, d = 0.56$) and the associative learning model's ($t(80) = 17.1, p < 0.001, d = 1.9$) performance, assessing only the upper confidence bound sampling strategies, the non-localized model performed better than the localized function learning model ($t(80) = 5.0495, p < 0.001, d = 0.56$). Furthermore, decomposing the upper confidence bound sampling strategy into pure exploitation (mean greedy) or pure exploration (variance greedy) components revealed that both high expectations of reward and the reduction of uncertainty are necessary components for the function learning model to predict human search behavior, with mean greedy ($t(80) = -8.85, p < .001$, $d = 0.98$) and variance greedy ($t(80) = -16.63, p < .001, d = 1.85$) performing worse than the combined UCB algorithm. In total, 48 participants were best predicted by the function learning model combined with a upper confidence bound sampling strategy.

Extracted parameter estimates

Looking more closely at the parameter estimates of the Gaussian Process-based function learning model with upper confidence bound sampling (Figure 6.6; median estimates per participant) showed that people tend to underestimate the extent of spatial correlations, with estimated λ values significantly lower than the ground truth for both Smooth (mean estimate: $\hat{\lambda} = 0.82, t(41) = -17.60, p < .001, d = 2.71$; compared to ground truth $\lambda_{Smooth} = 2$) and Rough environments ($\hat{\lambda} = 0.78, t(38) = -3.89, p < .001, d = 0.62$; compared to $\lambda_{Rough} = 1$), which could be interpreted as a tendency to avoid overgeneralization (Gershman & Daw, 2017). The exploration bonus of UCB sampling (β) was robustly

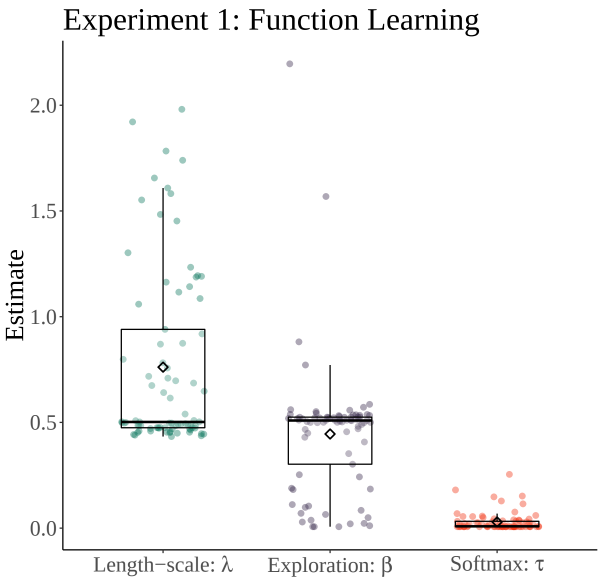
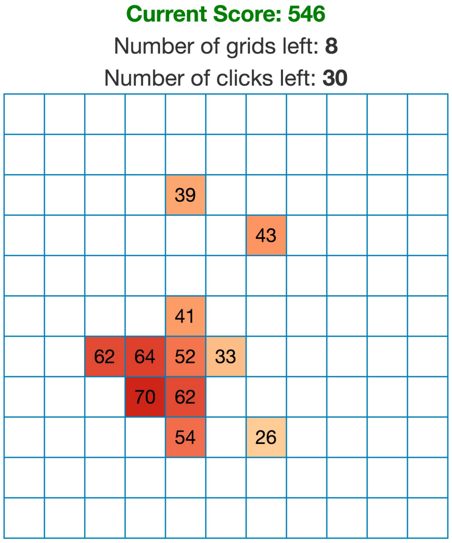


Figure 6.6: Parameter estimates of Experiment 10.

estimated above 0 ($\hat{\beta} = 0.47$, $t(80) = 12.78$, $p < .001$, $d = 1.42$, compared to the lower estimation bound), indicating participants valued the exploration of uncertain options, along with exploiting high expectations of reward. Additionally, the estimates of the softmax temperature were very low ($\hat{\tau} = 0.01$), corresponding to more precise model predictions.

6.6 Experiment 11: Bivariate function exploration-exploitation

The second variant of the spatially-correlated multi-armed bandit presents participants with a bi-variate spatially correlated bandit. Therein, the problem space was represented as a two-dimensional grid, measuring 11×11, resulting in 121 unique tiles in total. As in the univariate experiment, participants could click to reveal unexplored tiles or re-click previously uncovered tiles. A screenshot of Experiment 11 is shown in Figure 6.7.



Current Score: 546

Number of grids left: **8**

Number of clicks left: **30**

				39						
						43				
				41						
		62	64	52	33					
			70	62						
				54		26				

Figure 6.7: Screenshot of Experiment 11. Bivariate spatially correlated multi-armed bandit.

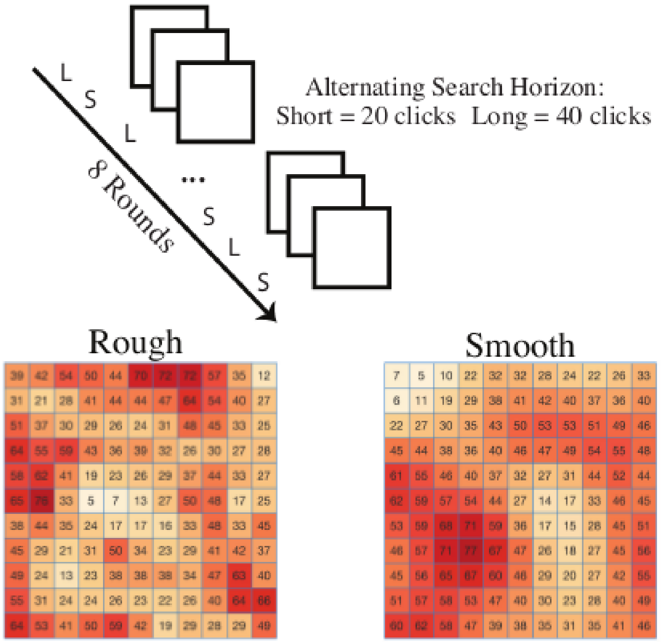
6.6.1 Methods

Participants

80 participants were recruited from Amazon Mechanical Turk (25 Female; mean age ± SD 32 ± 9). Each participant was paid a participation fee of $0.50 and a performance contingent bonus up to $1.50. Subjects earned on average $1.64 ± 0.20 and spent on average 8 minutes on the task.

Design

As before, a 2×2 between subject design was used, where participants were randomly assigned to one of two different pay-off structures (*Maximizers* vs. *Accumulators*) and one of two different classes of environments (*Smooth* vs. *Rough*). Each grid represented a bi-variate function, with each observation including normally distributed noise, $\varepsilon \sim \mathcal{N}(0, 1)$. The task was presented over 8 blocks on different grid worlds drawn from the same class of environments. In each block, participants had either a *Short* (20 clicks) or *Long* (40 clicks) search horizon to interact with the grid. The search horizon alternated between blocks (within subject), with initial horizon length counterbalanced between subjects. Observations were again scaled for every block to a uniformly sampled maximum value in the range of 65 to 85 as before. The design of Experiment 11 is shown in Figure 6.8.


Alternating Search Horizon:
Short = 20 clicks Long = 40 clicks
L
S
L
...
S
L
S
8 Rounds

Rough

39	42	54	50	44	70	72	72	57	35	12
31	21	28	41	44	44	47	64	54	40	27
51	37	30	29	26	24	31	48	45	33	25
64	55	59	43	36	39	32	26	30	27	28
58	62	41	19	23	26	29	37	44	33	27
65	76	33	5	7	13	27	50	48	17	25
38	44	35	24	17	17	16	33	48	33	45
45	29	21	31	50	34	23	29	41	42	37
49	24	13	23	38	38	38	34	47	63	40
55	31	24	24	26	23	22	26	40	64	66
64	53	41	50	59	42	19	29	28	29	49

Smooth

7	5	10	22	32	32	28	24	22	26	33
6	11	19	29	38	41	42	40	37	36	40
22	27	30	35	43	50	53	53	51	49	46
45	44	38	36	40	46	47	49	54	55	48
61	55	46	40	37	32	27	31	44	52	44
62	59	57	54	44	27	14	17	33	46	45
53	59	68	71	59	36	17	15	28	45	51
46	57	71	77	67	47	26	18	27	45	56
45	56	65	67	60	46	29	20	27	42	55
51	57	58	53	47	40	30	23	28	40	49
60	62	58	47	39	38	35	31	35	41	46

Figure 6.8: Design of Experiment 11. Environments could be either smooth or rough (between subject manipulation). Sample horizons could be either short (20 clicks) or long (40 clicks, within subject manipulation, counter-balanced).

Materials and procedure

Materials and procedures were almost identical to the uni-variate case. Participants were shown four fully revealed grids in order to familiarize themselves with the task before they started. Example environments were drawn from the same class of environments assigned to the participant (Smooth or Rough). As before, three comprehension questions were used to ensure full understanding of the task.

At the beginning of each of the 8 blocks, one random tile was revealed and participants could use their mouse to click any of the 30 tiles until the search horizon was exhausted, including re-clicking previously revealed tiles.

Payoff conditions

Performance is compared under two different payoff conditions, *Accumulators*) or *Maximizers*. In each payoff condition, participants received a performance contingent bonus of up to $1.50. All 8 blocks were weighted equally.

Smoothness of the environment

All environments were sampled from a bivariate Gaussian Process prior parameterized with a radial basis function kernel. I sampled 20 *Smooth* environments using $\lambda = 2$ and 20 *Rough* environments using $\lambda = 1$. Subjects performed the task on 8 grids randomly drawn (without replacement) from their assigned class of environments, while the four fully revealed environments used to familiarize subjects with the task were randomly drawn from the remaining 12 environments without replacement.

Search horizons

The two horizon lengths were 20 for Short and 40 for Long blocks and varied within subject (alternating between blocks) as in Experiment 10.

6.6.2 Results

Figure 6.9 shows task performance. Performance improved as a function of the trial number as measured by both the overall correlation between average reward and trial number ($r = 0.32, p < 0.001$) and between the maximum observed reward and trial number ($r = 0.83, p < 0.001$). There were no learning effects across blocks, indicated by a lack of correlation between average reward and block number ($r = 0.19, p > 0.5$), or between maximum reward and block number ($r = -0.37, p > 0.1$). Thus, as in the univariate experiment, performance improved as more information was revealed, but not over additional blocks of identically parameterized environments.

Payoff conditions

Payoff conditions influenced search behavior, with participants in the Maximizer condition displaying more variance in the locations sampled ($t(78) = -2.48, p < 0.01, d = 0.55$). Participants in the *Accumulator* group did not achieved a higher average reward than participants in the *Maximizer* group ($t(78) = 1.32, p > 0.1.\ d = 0.29$) and *Maximizers* did not reveal a higher maximum value than *Accumulators* ($t(78) = 1.48, p > 0.1, d = 0.33$).

Environment and horizon

Independent of the payoff condition, participants assigned to Smooth environments achieved higher average rewards ($t(78) = 6.55, p < 0.001, d = 1.47$) and higher maximum rewards ($t(78) = 5.45, p < 0.001, d = 1.22$) than those assigned to the Rough environments, suggesting that stronger correlations of payoffs make the task easier. As before, longer horizons did not lead to better overall performance in the Accumulator condition ($t(79) = 0.96$, $p > 0.3, d = 0.11$), although participants given longer horizons found larger maximum

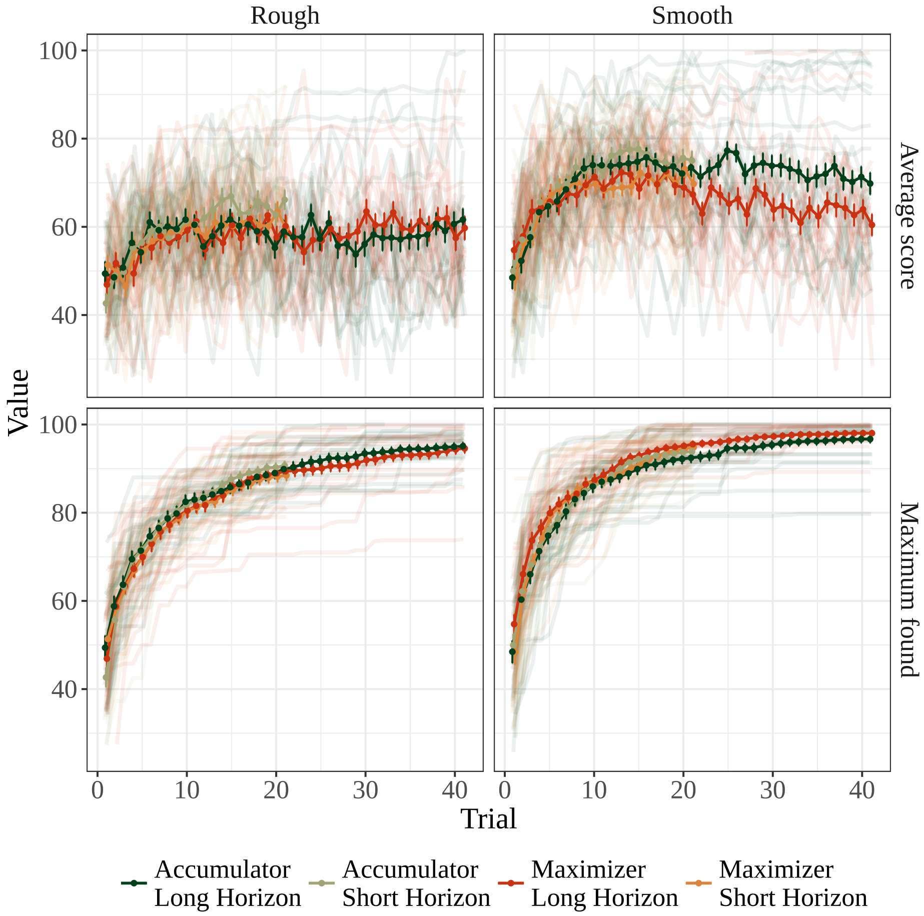


Figure 6.9: Results of Experiment 11.

rewards for all payoffs and environment conditions ($t(79) = 9.98, p < 0.001, d = 1.12$).

6.6.3 Cross-validation results

Cross-validation was performed as in Experiment 10 for all 27 models and each participant individually. In total, the cross validation for the bivariate experiment took around 52,000 hours or about 3 days on distributed across a 716 CPU cluster.

Results of the full model comparison for Experiment 11 are shown in Figure 6.10.

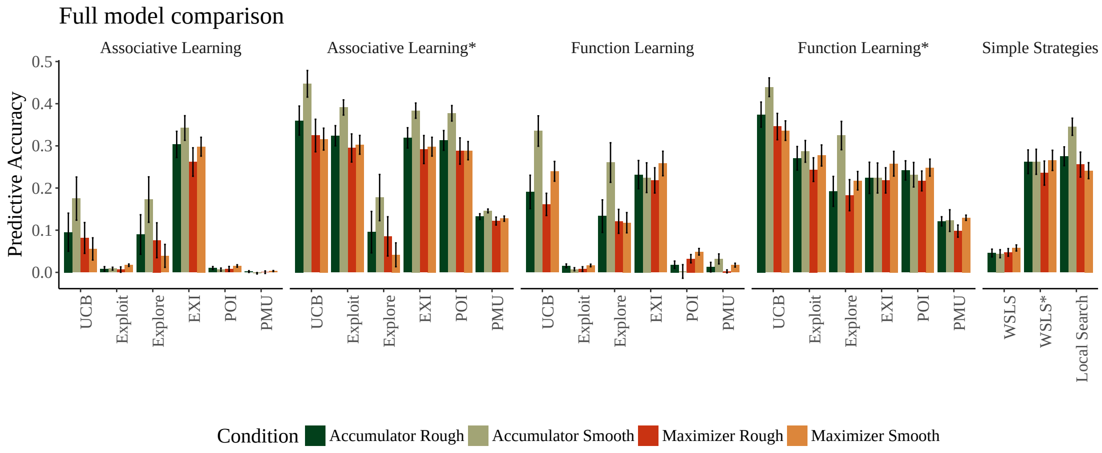


Figure 6.10: Model comparison results of Experiment 11.

Again, the function learning model performed better than the associative learning model ($t(79) = -5.01, p < 0.001, d = 0.56$). However, this time further localization of the models improved both the associative ($t(79) = 27.46, p < 0.001, d = 3.07$) and the function learning model ($t(79) = 19.12, p < 0.001\ d = 2.13$). Again, the upper confidence bound sampling strategy was by far the best sampling strategy ($t(79) = 12.57, p < 0.001$, $d = 1.41$). The localized function learning* model paired with a upper confidence bound sampling strategy predicted 59 participant best overall and performed significantly better than the localized associative learning* model ($t(70) = -2.18, p < 0.05, d = 0.24$; UCB only). Again, both components of the UCB strategy—the expected reward ($t(79) = -6.44, p < .001, d = 0.72$) and the attached uncertainty ($t(79) = -14.32, p < .001$, $d = 1.60$)—were necessary to predict choices.

6.6.4 Parameter estimates

Further extracting the median parameter estimates per participant from the localized Gaussian Process-based function learning* model (Figure 6.11) showed that participants again un-

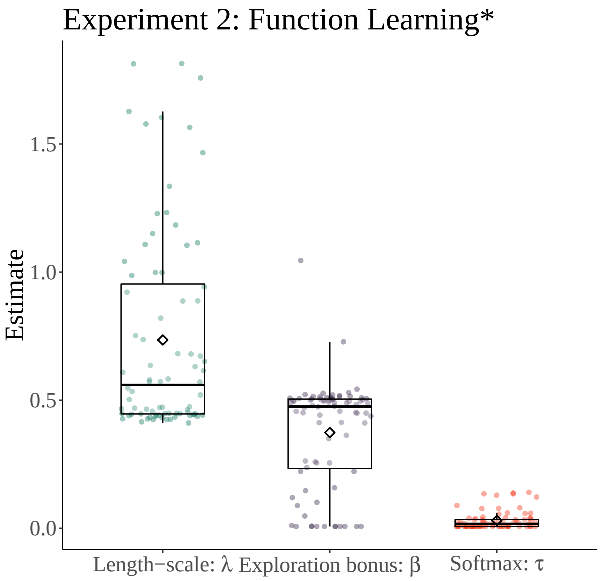


Figure 6.11: Parameter estimates of Experiment 11.

derestimated the strength of the underlying spatial correlation in both Smooth ($\hat{\lambda} = 0.92$, $t(42) = -14.62, p < .001, d = 2.22$; comparison to $\lambda_{Smooth} = 2$) and Rough environments ($\hat{\lambda} = 0.78, t(36) = -5.31, p < .001, d = 0.87$; comparison to $\lambda_{Rough} = 1$), suggesting a robust tendency to avoid overgeneralization. The estimated exploration bonus β was again greater than 0 ($\hat{\beta} = 0.45$, $t(79) = 27.02, p < .001$, $d = 3.02$, compared to the lower estimation bound), while the estimated softmax temperature parameter τ was slightly larger than in Experiment 10, $\hat{\tau} = 0.09$.

6.7 Discussion

Generalization is a powerful tool that helps guide human decision making, supporting faster learning rates and increased performance. In this chapter, I examined human behavior in a reward acquisition task within two type of grid worlds containing spatially correlated rewards. These environments were designed such that the number of states vastly exceeded the available search horizon. However, the presence of spatial correlation provided traction for generalization, making it possible to learn an underlying value function based on spatial context. Participants performed better with stronger spatial correlations, reached similar performance in short compared to long horizons, and were sometimes able to find the optimal option just by balancing exploration and exploitation.

Using predictive cross-validation, I found that participants were best described by a Gaussian Process-based *function learning* model, which relates different search options by their spatial contexts. The advantage of function learning is that it provides a mechanism for generalization from a small number of observations to a vast number of possibilities. The recovered parameter estimates indicate that there was a systematic tendency to undergeneralize the strength of spatial correlations, yet comparisons with associative learning models provide evidence that generalization is a robust and recoverable phenomenon (also see Appendix).

Participants seem to deliberately trade off between exploration and exploitation as they are predicted well by an upper confidence bound sampling strategy. Even though this strategy is both *optimistic* (by assuming uncertainty is always positive) and *myopic*, it is nonetheless the only known algorithm with performance guarantees (Srinivas et al., 2010) in this setting. Similar approaches have previously been applied to problems in ecology (Gotovos et al., 2013), robotics (Berkenkamp et al., 2016) and biology (Desautels, 2014). The finding that human behavior is best predicted by this model shows that humans are capable of learning and adapting their behavior in complex and uncertain environments, even if an optimal solution is unobtainable.

The results presented in this chapter have tried to bridge a gap in the current reinforcement learning literature by stressing the importance of generalization, specifically in situations in which humans have to learn and adapt to environments where not all outcomes can be observed. This has been one of the first studies to predict individual decisions in such a complex search task, including a rigorous comparison between 27 models that yielded robust model comparisons and parameter estimates. Interestingly, recent findings have connected both spatial and conceptual representations to a common neural substrate (Constantinescu et al., 2016), opening up the opportunity to apply the function learning model to assess generalization and decision making in further contextualized settings. This will be the focus of the next chapter.

"In time, and as one comes to benefit from experience, one learns that things will not turn out as well as one hoped."

Jerome Bruner, 1966

7

Contextual Multi-Armed Bandits

In this chapter, I investigate human behavior within a paradigm in which contextual cues are related to an option's expected reward by an underlying function.

7.1 Introduction

Chapter 6 focused on search behavior in various grid worlds in a spatially correlated multi-armed bandit paradigm. As different options' rewards were related by their spatial location, learning a function mapping space to expected rewards provided an opportunity for generalization. Thus, the spatial location of an option acted as a context guiding participants' decision making. However, contextual cues do not only come as spatial information but can frequently also contain conceptual information. Coming back to the restaurant example from before, if you are trying to find a good restaurant to dine in tonight, you might not even know the location of different restaurants, but rather use other contextual cues such as the price of the food on the menu or recommendations from friends and online rating pages.

This chapter investigates human behavior in such problems within a paradigm called the contextual multi-armed bandit task. Learning and decision making within contextual multi-armed bandit tasks require the same two elements as before: learning a function that maps the observed features of options to their expected rewards, and a sampling strategy that uses these expectations to choose between the options. Function learning is important as it allows one to generalize previous experiences to novel situations. For example, it allows to predict the quality of a new restaurant from experiences with other restaurants with a similar number of customers and a similarly priced menu. The decision (or sampling) strategy is important because not only should one attempt to choose options that are currently most rewarding, but also take into account how much one can learn in order to make good choices in the future. This again means taking into account the exploration-exploitation trade-off, where exploration means learning about the function that relates features to rewards and exploitation means using that functional knowledge to generate high rewards.

In what follows, I will introduce the contextual multi-armed bandit paradigm in more detail and propose several models to describe how people solve such tasks. I will then describe three experiments which explore how people perform within different variants of the task. The results will show that participants are able to learn within this paradigm, approximating the function in an efficient way (Srinivas et al., 2010; Lucas et al., 2015) and using their knowledge to sensitively balance exploration and exploitation. However, the extent to which participants are able to learn and generalize the underlying function depends on the complexity of the task.

7.2 Contextual multi-armed bandits

A contextual multi-armed bandit task (CMAB) is a game in which on each round, an agent is presented with a context (a set of features) and a number of options which each offer an unknown reward. The expected rewards associated to each option depend on the context through an unknown function. The context can contain general features that apply to all options (e.g., the city the restaurants are in) or specific features that apply to single options (e.g., the exact menu and its price). The agent's task is to choose those options that will accumulate the highest reward over all rounds of the game. The rewards are stochastic, such that even if the agent had complete knowledge of the task, a choice would still involve a kind of gamble. In this respect, choosing an option can be seen as choosing a slot machine (a one-armed bandit) to play, or, equivalently, choosing which arm of a multi-armed bandit to play. After choosing an option in a round, the agent receives the reward of the chosen option but is not informed of the foregone rewards that could have been obtained from the other options. For an agent who ignores the context, the task would appear as a restless bandit task (e.g., Speekenbrink & Konstantinidis, 2015), as the rewards associated with an arm will vary over time due to the changing context. However, learning the function that maps the context to (expected) rewards will make these changes in rewards predictable and thereby choosing the optimal arm easier. In order to choose wisely, the agent should thus learn about the underlying function. Sometimes, this may require her to choose an option which is not expected to give the highest reward on a particular round, but one that might provide useful information about the function, thus choosing to explore rather than to exploit.

Contextual multi-armed bandit tasks provide a scenario in which a participant has to learn a function in order to maximize the outputs of that function over time by making wise choices. They are a natural extension of both the classic multi-armed bandit task, which is a CMAB with an invariant context throughout, and the restless bandit task, which is a CMAB with time as the only contextual feature.

While the CMAB is novel in the psychological literature (though see Stojic et al., 2015, for a related task), where few tasks explicitly combine function learning and experience-based decision making, there are certain similarities with tasks used in previous research. For example, recent studies in experience-based decision-making provided participants with descriptions about the underlying distributions that generate rewards (e.g., Lejarraga & Gonzalez, 2011; Weiss-Cohen et al., 2016). Just as in the CMAB, this presents a naturalistic decision environment in which different sources of information (e.g., descriptions and participants' own experience) need to be integrated in order to choose between alternatives or courses of action.

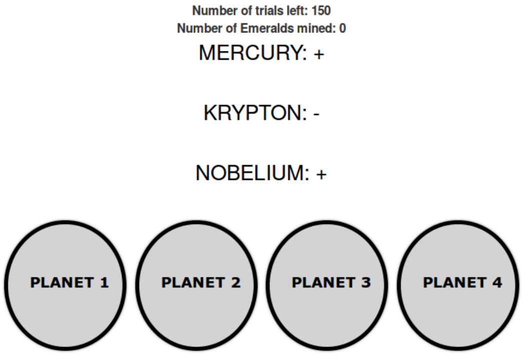
Another related paradigm is multiple cue probability learning (MCPL, Kruschke & Johansen, 1999; Speekenbrink & Shanks, 2008) in which participants are shown an array of cues that are probabilistically related to an outcome and have to learn the underlying function mapping the cues' features to expected outcomes. Especially when the outcome is a categorical variable, such as in the "Weather Prediction Task" (Gluck et al., 2002; Speekenbrink et al., 2008) as described in Chapter 2, making a prediction is structurally similar to a decision between multiple arms (possible predictions) that are rewarded (correct prediction) or not (incorrect prediction). Just as in the CMAB, multiple-cue probability learning and probabilistic category learning tasks require people to learn a function which maps multiple cues or features to expected outcomes. An important difference however is that in these latter tasks there is a strong dependency between the options: there is only one correct prediction, and hence there is a perfect (negative) correlation between the rewards for the options. Whether a current choice was rewarded or not thus provides information about whether the non-chosen options would have been rewarded. This dependency weakens the need for exploration, especially when the outcome is binary, in which case there is no need for exploration at all. In CMAB tasks, there is a stronger impetus for exploration, as the rewards associated to arms are generally conditionally independent, given the context. Knowing that a particular option was rewarded thus does not provide immediate information whether another option would have been rewarded. Another major difference is that MCPL tasks generally require participants to learn the whole function. In CMAB tasks, learning the function is only necessary insofar as it helps to generalize from observed outcomes and thus leads to better decisions. To solve the exploration-exploitation dilemma, it may suffice to learn the function well only in those regions that promise to produce high rewards. Moreover, as I will describe later, each option can be governed by its own function relating context to rewards. To the best of my knowledge, simultaneous learning of multiple functions has not previously been investigated.

Another area of related research comes from the associative learning literature, where it has been shown that context can act as an additional cue to maximize reward (cf Bouton & King, 1983; Gershman et al., 2010). In one example of this, Gershman & Niv (2015) showed how generalization based on context (the average reward of options in an environment) can explain how participants react to novel options in the same environment, such that a high-reward context leads people to approach novel options, while a low-reward context leads to avoidance of novel options. The CMAB paradigm introduced here is related to such situations, but instead of a single, constant context, varies the contexts such that good performance requires learning the underlying contextual function.

7.3 General CMAB task

In the experimental implementation of the CMAB task, participants are told they have to mine for "Emeralds" on different planets. Moreover, it is explained that at each time of mining, the galaxy is described by 3 different environmental factors, "Mercury", "Krypton", and "Nobelium", that have different effects on different planets. Participants are then told they have to maximize their production of Emeralds over time by learning how the different environmental factors influence the planets and choosing the planet they think will produce the highest outcome in light of the available factors. Participants were explicitly told that different planets can react differently to specific environmental factors. A screenshot of the CMAB task can be seen in Figure 7.1.

As each planet responds differently to the contexts, they can be seen as arms of a multi-armed bandit that are related to the context by different functions. The reward of an op-


Number of trials left: 150
Number of Emeralds mined: 0

MERCURY: +

KRYPTON: -

NOBELIUM: +

Figure 7.1: Screenshot of the CMAB task depicting Experiment 12.

tion j is given as

$$y_{j,t} = f(a_t = j, x_t) = f_j(x_t) + \varepsilon_{j,t} \tag{7.1}$$

with $\varepsilon_{j,t} \sim \mathcal{N}(0, 5)$. The task consists of 150 trials in which a random context is drawn and participants choose a planet to mine on[*].

The three experiments differ in the functions f_j and whether the environmental factors defining the context were binary or continuous. This is specified in more detail when describing the experiments. Source code for the experimental set-up is available online.[†]

7.4 Models of learning and decision making in CMAB-tasks

7.4.1 Contextual and context-blind learning

I will specify two different classes of learning within CMAB-tasks, contextual and context-blind learning. Whereas contextual models learn and make use of how the provided context

[*] The initial trial had the same context x_1 for all participants. Afterwards, the values of the context x_t were sampled at random

[†] https://github.com/ericschulz/contextualbandits

maps onto expected rewards over time, context-blind models only learn the reward for each option separately without taking functional knowledge into account. This is similar to the distinction between associative and function learning models of the previous chapter.

There will be two kinds of context-blind models, a Bayesian mean tracker and Kalman filtering. The *Bayesian mean-tracking* model assumes that the average reward associated to each option is constant over time and simply computes a posterior distribution over the mean μ_j of each option j. Here, this is implemented as a relatively simple version of such a model which assumes rewards are normally distributed with a known variance but unknown mean and the prior distribution for that mean is again a normal distribution. This corresponds exactly to the associative learning model from chapter 6 and implies that the posterior distribution for each mean is also a normal distribution:

$$p(\mu_{j,t}|\mathcal{D}_{t-1}) = \mathcal{N}(\mu_{j,t}, \sigma^2_{j,t}) \tag{7.2}$$

where the mean $\mu_{j,t}$ represents the currently expected outcome for a particular arm j and the variance $\sigma^2_{j,t}$ represents the uncertainty attached to that expectation. The posterior distribution can be computed through a mean-stable version of the Kalman filter, which I will describe next.

Unlike the Bayesian mean tracking model, which computes the posterior distribution of a time-invariant mean μ_j after each new observation, the *Kalman filter* is a suitable model for tracking a time-varying mean $\mu_{j,t}$ which I here assume varies according to a simple random walk

$$\mu_{j,t+1} = \mu_{j,t} + \zeta_t \qquad \zeta_t \sim \mathcal{N}(0, \sigma^2_\zeta) \tag{7.3}$$

Kalman filtering models have been used to successfully describe participants' choices in a restless bandit task (Speekenbrink & Konstantinidis, 2015) and have also been proposed as a model unifying many findings within the literature of context-free associative learning (Kruschke, 2008; Gershman, 2015). In this model, the posterior distribution of the mean is again a normal distribution

$$p(\mu_{j,t}|\mathcal{D}_{t-1}) = \mathcal{N}(\mu_{j,t}, \sigma^2_{j,t}) \tag{7.4}$$

with mean

$$\mu_{j,t} = \mu_{j,t-1} + \delta_{j,t}G_{j,t}(y_t - \mu_{j,t-1}) \tag{7.5}$$

where y_t is the received reward on trial t and $\delta_{j,t} = 1$ if arm j was chosen on trial t, and 0 otherwise. The "Kalman gain" term is computed as

$$G_{j,t} = \frac{\sigma^2_{j,t-1} + \sigma^2_\zeta}{\sigma^2_{j,t-1} + \sigma^2_\zeta + \sigma^2_\varepsilon} \tag{7.6}$$

where $\sigma^2_{k,t}$ is the variance of the posterior distribution and the mean $\mu_{j,t}$ is computed as

$$v_{j,t} = (1 - \delta_{j,t}G_{j,t})(\sigma^2_{j,t-1} + \sigma^2_\zeta) \tag{7.7}$$

Prior means and variances were initialized to $\mu_{j,0} = 0$ and $\sigma^2_{j,0} = 1000$, while the innovation variance σ^2_ζ and error variance σ^2_ε were free parameters. The Bayesian mean-tracking model is obtained from the Kalman filter model by setting the innovation variance to $\sigma^2_\zeta = 0$, implying the underlying mean is not assumed to change over time.

Contextual learning models will be implemented as Gaussian Process regression for which it will be assumed that participants learn a separate function $f_j(x)$ that maps contexts

x to rewards y for each option j. There will be two different kernels specifying how people learn within CMAB tasks. The first is the radial basis function kernel

$$k(x, x') = \sigma_f^2 \exp\left(-\frac{|x - x'|^2}{2\lambda^2}\right) \tag{7.8}$$

for which λ indicates the level of generalization participants might perform and is again estimated as a free parameter. The second kernel is a linear kernel

$$k(x, x') = \vartheta_1 (x - \vartheta_2)(x' - \vartheta_2) \tag{7.9}$$

which is the same as performing Bayesian linear regression and thus a simple rule-based function learning model to compare the more universal radial basis function kernel against.

7.4.2 Sampling strategies

As in Chapter 6, the models of learning return a predictive distribution for each option which then has to be transformed to actual choices. Thus, I will again compare different sampling strategies which I recapitulate quickly.

The *upper confidence bound* (UCB) sampling strategy defines a trade-off between an option's expected value and the associated uncertainty and chooses the option for which the upper confidence bound of the mean is highest. It has a free parameter β, which determines the width of confidence interval. The UCB sampling strategy can be described as a selection strategy with an exploration bonus, where the bonus dynamically depends on the confidence interval of the estimated mean reward at each time point. I have found it to be the best sampling strategy within the spatially correlated multi-armed bandit paradigm within the previous chapter.

Another decision strategy is the *probability of improvement* (PoI) rule, which calculates the probability for each option to lead to an outcome higher than the option that is currently believed to have the highest expected value (Kushner, 1964). Intuitively, this algorithm estimates the probability of one option to generate a higher utility than another option, which is taken to be the best option observed so far.

The PoI rule focusses solely on the probability that an option provides a higher outcome than another; whether the difference in outcomes is large or small does not matter. The *expected improvement* (EXI) rule is similar to the PoI rule, but does take the magnitude of the difference in outcomes into account and compares options to the current favorite in terms of the expected increase of outcomes (Mockus et al., 1978).

The fourth decision strategy is the *probability of maximum utility* (PMU) rule (Speekenbrink & Konstantinidis, 2015). This strategy chooses each option according to the probability that it results in the highest reward out of all options in a particular context. It is again implemented by Monte Carlo-sampling from the posterior predictive distribution.

As participants' decisions are expected to be more noisy reflections of the provided decision strategies, I use a softmax transformation to map the value of each option according to the decision rule into probabilities of choice:

$$p(a_t = j|s_t, \mathcal{D}_{t-1}) = \frac{\exp\{\tau^{-1} \cdot \mathrm{q}(a = j|s_t, \mathcal{D}_{t-1})\}}{\sum_{i=1}^{n} \exp\{\tau^{-1} \cdot \mathrm{q}(a = i|s_t, \mathcal{D}_{t-1})\}} \tag{7.10}$$

The temperature parameter $\tau > 0$ governs how consistently participants choose according to the values generated by the different model-sampling strategy combinations and will be estimated as a free parameter.

7.4.3 Model comparison

All models are compared in terms of their out-of-sample predictive performance, assessing the accuracy of their one-step-ahead predictions and comparing them to that of a *random model* which picks each option with the same probability. The procedure is as follows: for each participant, a given model is fitted by maximum likelihood to the first $t-1$ trials with a differential evolution optimization algorithm (using 100 epochs, cf. Mullen et al., 2009). Afterwards, the fitted model is used to predict the choice on trial t. As repeating this procedure for every trial is computationally too expensive, models' predictive accuracy for every participant will be assessed on trials $t = \{10, 30, 50, 70, 90, 110, 130, 150\}$. The one-step-ahead predictive accuracy measure compares each model $\mathcal{M}_k$ to a random model $\mathcal{M}_{\text{rand}}$:

$$R_p^2 = 1 - \log \mathcal{L}(\mathcal{M}_k) / \log \mathcal{L}(\mathcal{M}_{\text{rand}}) \tag{7.11}$$

where $\mathcal{L}(\mathcal{M})$ denotes the likelihood of model $\mathcal{M}$ (i.e., the probability of a participants' choices as predicted by fitted model $\mathcal{M}$). This measure returns values between 0 (accuracy equal to the random model) and 1 (accuracy infinitely larger than the random model) and is exactly the same measure of predictive validity as used in Chapter 6.

7.5 Experiment 12: Binary features

The goal of the first contextual multi-armed bandit experiment was to test whether participants can learn to make good decisions in a CMAB task at all. For this purpose, I set up a relatively simple contextual bandit scenario in which the context only consists of binary features.

7.5.1 Participants

Forty-seven participants (26 male) with an average age of 31.9 years ($SD = 8.2$) were recruited via Amazon Mechanical Turk and received $0.3 plus a performance-dependent bonus. The experiment took 12 minutes to complete on average and the average reward was $0.73±0.07.

7.5.2 Task

There were four different arms that could be played (planets that could be mined). In addition, three discrete variables, $x_{i,t}$, $i = 1, 2, 3$, were introduced as the general context. The three variables defining the contexts could either be on ($x_{i,t} = 1$) or off ($x_{i,t} = -1$). The outcomes of the four arms were dependent on the context as follows:

$$f_1(x_t) = 50 + 15 \times x_{1,t} - 15 \times x_{2,t} \tag{7.12}$$

$$f_2(x_t) = 50 + 15 \times x_{2,t} - 15 \times x_{3,t} \tag{7.13}$$

$$f_3(x_t) = 50 + 15 \times x_{3,t} - 15 \times x_{1,t} \tag{7.14}$$

$$f_4(x_t) = 50 \tag{7.15}$$

On each trial, the probability that a contextual feature was on or off was set to $p(x_{i,t} = 1) = p(x_{i,t} = -1) = 0.5$, making each of the 8 possible contexts equally likely to occur on a given trial. The functions f_j were designed such that the expected reward of each arm over all possible contexts equals $\mathbb{E}[y_{j,t}] = 50$. This means that the only way to gain higher rewards than the average of 50 is by learning how the contextual features influence the rewards. More formally, this implies that no arm achieves first-order stochastic dominance. Moreover, including the context-independent fourth arm that returns the mean with added noise helps to distinguish even further between learning and not learning the context: this arm has the same expected value as all the other arms but a lower variance and therefore achieves second-order stochastic dominance over the other arms. As such, a context-blind and risk-averse learner would prefer this arm over time.

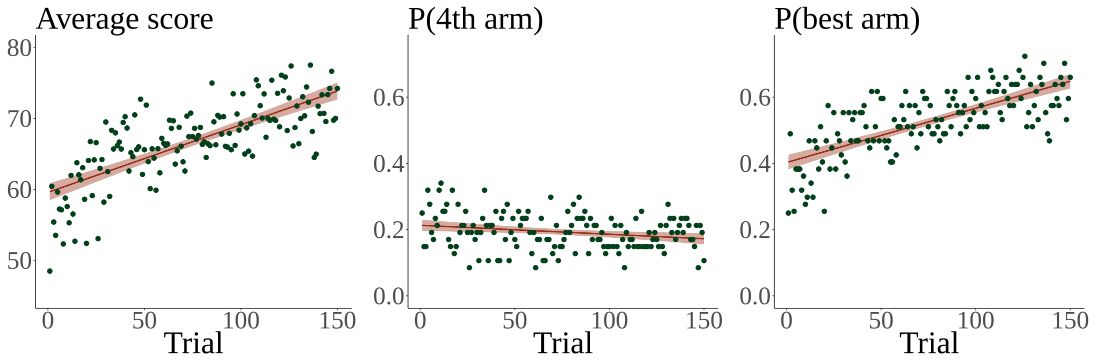
7.5.3 Procedure

After giving their informed consent, participants received instructions to the experiment. Participants were told that they had to mine for "Emeralds" on different planets. Moreover, it was explained that at each time each of the 3 different environmental factors could either be on (+) or off (-) and had different effects on different planets. Participants were told that they had to maximize the overall production of Emeralds over time by learning how the different elements influence the planets and then picking the planet they thought would produce the highest outcome, given the status (on or off) of the elements. It was explicitly noted that different planets can react differently to different elements. After reading the instructions, participants performed the CMAB task. There were a total number of 150 trials and participants were paid $0.3 + total score/(150 × 100).

7.5.4 Results

Behavioral results

Participants gained 66.78 points (SD=13.02) per round on average throughout the task. Participants' average scores were significantly above the chance level of 50 ($t(46) = 8.83$, $p < 0.01$, $d = 1.29$). 34 out of 47 participants performed better than chance according to a simple t-test with $\alpha = 0.05$ and $\mu_0 = 50$. Thus, participants seemingly learned to take the context into account, obtaining higher rewards than expected if they were ignoring the context.



(a) Average scores per round. **(b)** Proportion of 4th arm. **(c)** Proportion of best arm.

Figure 7.2: Results of the binary CMAB task of Experiment 12. Average overall score (a), proportion of choosing the fourth arm (b), and proportion of choosing the best arm (c)

Over time, participants made increasingly better choices (see Figure 7.2a), as indicated by a significant correlation between the average score (over participants) and trial number, $r = 0.74, p < 0.01$. The proportion of participants choosing the non-contextual option (the option that did not respond to any of the contextual features, indicated as the 4th arm) decreased over time ($r = -0.22, p < 0.05$, Figure 7.2b), another indicator that participants learned the underlying functions. The proportion of participants choosing the best option for the current context increased during the task ($r = 0.72, p < 0.01$, see Figure 7.2a).

As a final assessment of whether participants took the context into account, I tested whether the outcome and chosen arm on the previous trial were related to the choice of arm on the current trial. If participants relied on the context, one would not expect previous outcomes and choices to be related to current choices. If participants ignored the context and for instance relied on (average) rewards or a win-stay-lose-shift strategy, a positive relation between current choices and previous choices and outcomes would be expected. A hierarchical multinomial regression (where trials were nested within participants) with chosen arms as dependent variable and outcomes and chosen arm on trial $t - 1$ as predictors showed no significant effects of the predictors. Again, this indicates participants learned the underlying function instead of using more simplistic (and in this case not very useful) heuristic memorization techniques such as testing similar arms in sequences or changing to a particular arm after a particular score.

Modeling results

To determine which combination of learning model and sampling strategy best captures participants' choices, the one-step-ahead predictive comparison was used. For each participant and model, the pseudo-R^2 at the eight test trials was computed. Higher R^2-values indicate better model performance. The results are shown in Figure 7.3.

Overall, the best performing model was the GP learning model with a radial basis function kernel with the PoI sampling strategy. Aggregating over sampling strategies, the contextual models produced significantly better one-step-ahead predictions than the context-blind models ($t(46) = 6.82, p < 0.001, d = 0.99$). Additionally, the GP-model with an RBF kernel performed better than the linear model ($t(46) = 7.88, p < 0.001, d = 1.15$). Distinguishing the different sampling strategies turned out to be harder than comparing the different learning models. Aggregating over models, the probability of maximum utility strategy performed marginally better than all other sampling strategies ($t(46) = 3.10$, $p < 0.05, d = 0.45$). The probability of improvement sampling strategy numerically predicted participants' choices best out of all the sampling strategies when combined with the radial basis function kernel ($t(46) = 4.84, p < 0.001, d = 0.71$).

The median parameter estimates of the GP-RBF model over all sampling strategies per participant, are shown in Figure 7.4.

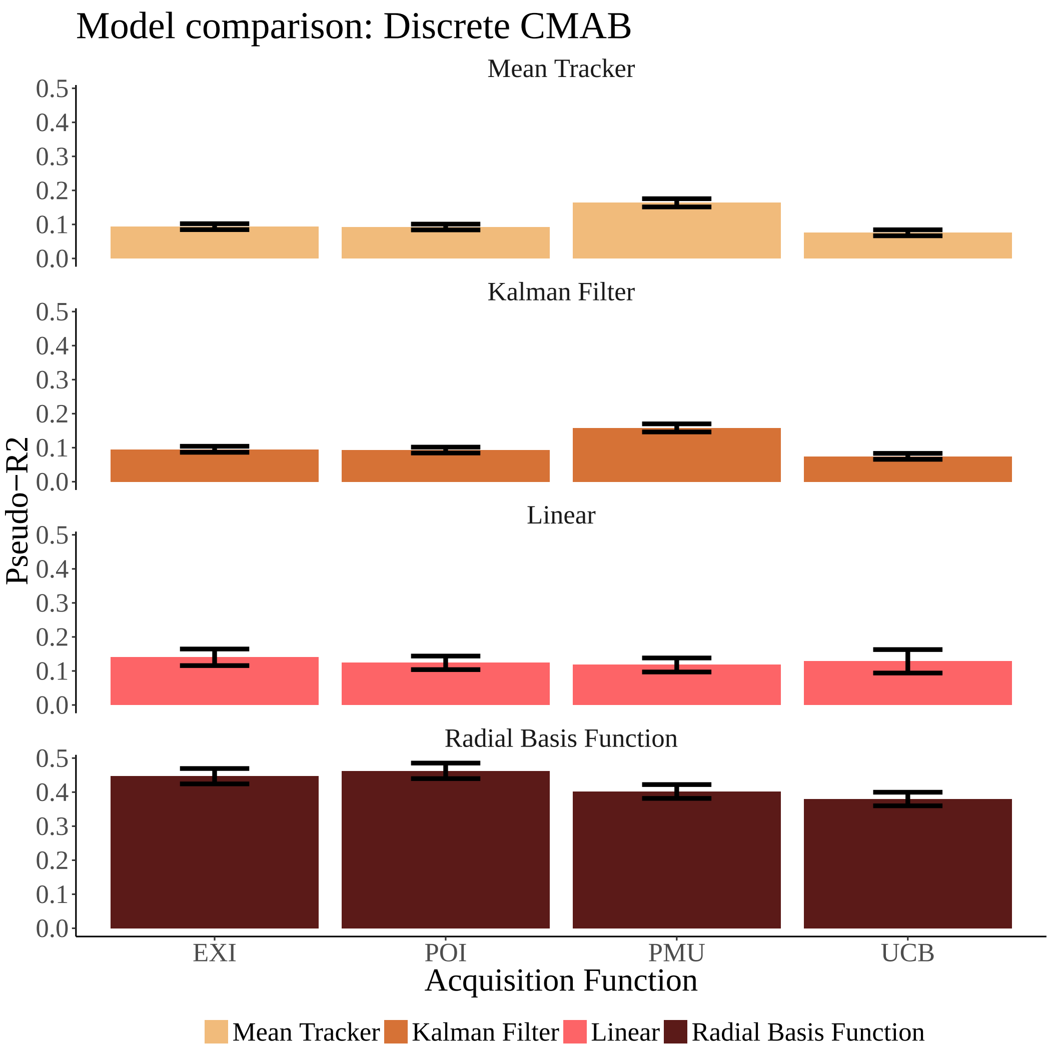


Figure 7.3: Predictive accuracy of the models for the CMAB task with discrete cues in Experiment 12. Error bars represent the standard error of the mean.

The median noise variance ($\hat{\sigma}^2_\varepsilon = 3.08$) was reasonably close to the underlying observation noise variance of $\sigma\varepsilon^2 = 5$, albeit smaller in general ($t(46) = -4.71, p < 0.01$, $d = 0.69$); thus, participants seemed to underestimate the overall noise in the observed outcomes. The estimates of the length-scale parameter clustered around the mean value of $\hat{\lambda} = 6.12$. An RBF kernel can emulate a linear kernel by setting a very high length-scale. As

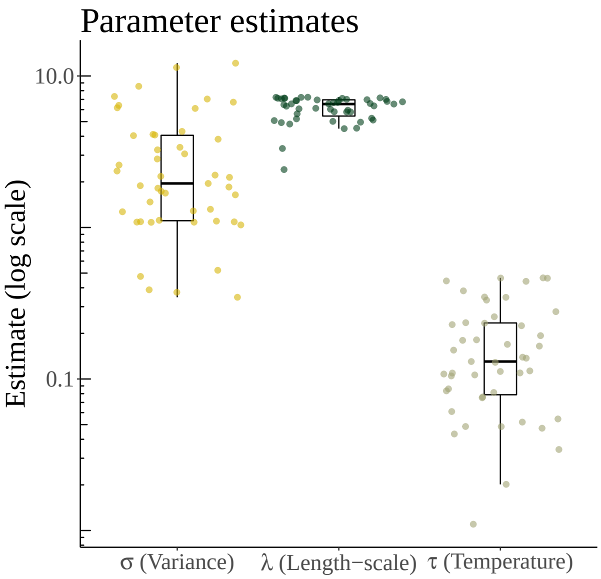


Figure 7.4: Median estimates of the error variance σ, the length-scale λ, and the temperature parameter τ for the Gaussian Process regression radial basis function kernel model.

the true underlying functions were linear in the experiment, we could thus expect high values for $\hat{\lambda}$. In that light, a value of six for the estimated length-scale seems surprisingly small, as it indicates that the dependencies between input points are expected to decay rather quickly, i.e. that participants generalized more locally than what was necessary. The overall temperature parameter was relatively low (mean estimate: $\hat{\tau}^{-1} = 0.085$), indicating that participants quite consistently chose the options with the highest predicted rewards.

According to the best fitting model in the cognitive modeling exercise, people learn the relation between context and outcomes by relying on a more general function approximator than linear regression. By using a Probability of Improvement decision strategy, participants compare the option which is thought to have the highest average rewards in the current context, to relatively lesser known options in that context, determining how probable these are to provide a higher reward. This strategy is in agreement with prior findings in simpler multi-attribute choice tasks (for example, Carroll & De Soete, 1991).

7.6 Experiment 13: Continuous-linear features

Experiment 12 contained only 8 unique contexts. In this case, a memorization strategy could have been feasible, such that participants may have simply memorized the expected rewards for each option in each context, rather than having inferred a more general model of the underlying function (although further behavioral checks spoke against this explanation). The goal of the next experiment was therefore to assess whether the findings from Experiment 12 would generalize to a task with a larger number of unique contexts, in which memorizing input-output pairs is less feasible. For this purpose, Experiment 12 used the same task, but with continuous rather than binary features comprising the contexts.

7.6.1 Participants

Fifty-nine participants (30 male) with a mean age of 32.4 (SD=7.8) were recruited via Amazon Mechanical Turk and received \$0.3 as a basic reward and a performance-dependent bonus of up to \$0.5. Participants earned an average amount of \$0.69 ± 0.08. Average time to complete the experiment was around 13 minutes.

7.6.2 Task and Procedure

The task was identical to that of Experiment 12, only this time the context contained continuous features with an underlying linear function mapping inputs to outputs:

$$f_1(x_t) = 50 + 3 \times x_{1,t} - 3 \times x_{2,t} \tag{7.16}$$

$$f_2(x_t) = 50 + 3 \times x_{2,t} - 3 \times x_{3,t} \tag{7.17}$$

$$f_3(x_t) = 50 + 3 \times x_{3,t} - 3 \times x_{1,t} \tag{7.18}$$

$$f_4(x_t) = 50 \tag{7.19}$$

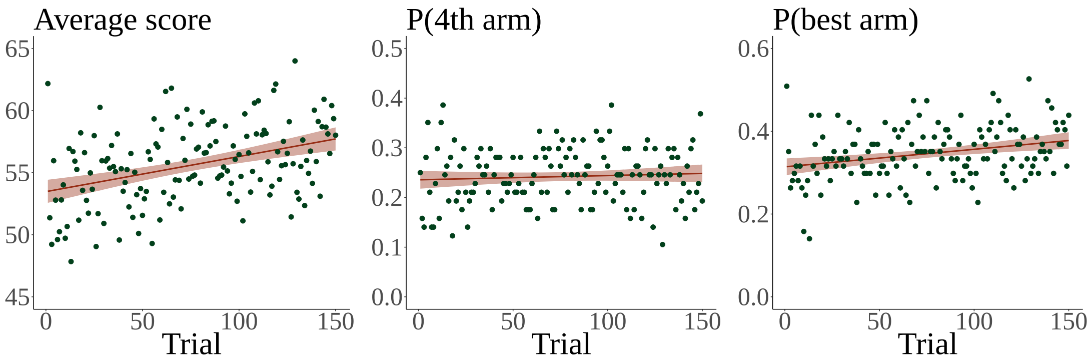
The values of the context variables $x_{j,t}$ were sampled randomly from a uniform distribution $x_{j,t} \sim \mathcal{U}(-10, 10)$. The values were rounded to whole numbers and shown in their numerical form to participants. Again, the expected value (over all contexts) for each option was 50, so no option achieved first-order stochastic dominance, while the fourth option achieved second-order stochastic dominance, as the variance of its rewards was the lowest.

7.6.3 Results

Behavioral results

The average obtained reward across participants was 59.84 (SD = 9.41), significantly higher than chance ($t(58) = 7.17, p < 0.01, d = 0.78$). As in Experiment 12, participants were able to take the context into account in order to increase their performance above chance level.

Performance increased over trials, $r = 0.39, t(58) = 3.64, p < 0.01$, although this was not as pronounced as in Experiment 12 (see Figure 7.5a). While the proportion of participants choosing the fourth option did not decrease significantly over time ($r = 0.05$,



(a) Average scores per round.

(b) Proportion of 4th arm.

(c) Proportion of best arm.

Figure 7.5: Results of the binary CMAB task of Experiment 13. Average overall score (a), proportion of choosing the fourth arm (b), and proportion of choosing the best arm (c)

$p > 0.05$), the proportion of choosing the best option given the context significantly increased over trials ($r = 0.33, p < 0.01$, see Figure 7.5c).

A hierarchical multinomial regression assessing whether either a given chosen arm or a given output was predictive for choosing a particular arm next in, again revealed no significant effect (all $p > 0.05$), indicating that participants did not seem to use any temporal-dependent heuristics. This means that repetitions of sequences or more simplistic choice patterns after a given score are not the main driver of participants' behavior.

Modeling results

Cross validation results are shown in Figure 7.6. The best performing model incorporates again a GP learning component, but now coupled with a UCB decision strategy. Contextual models again performed better than the context-blind models ($t(58) = 3.11, p < 0.01$, $d = 0.40$). Yet, the linear model performed significantly worse than all of the other models ($t(58) = -2.84, p < 0.01, d = 0.37$). The Gaussian Process model with a radial basis function kernel performed significantly better than all of the other candidate models

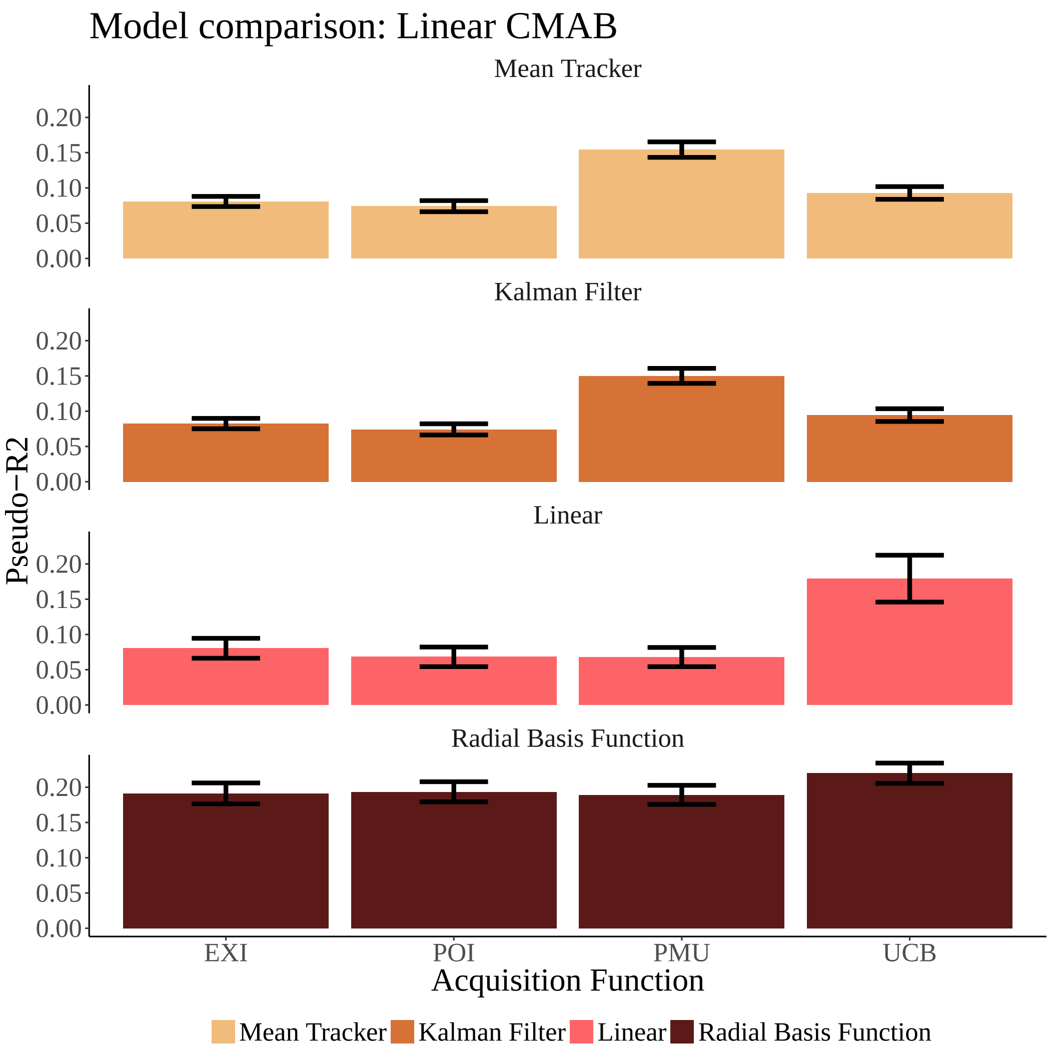


Figure 7.6: Predictive accuracy of the models for the CMAB task with continuous-linear cues in Experiment 13. Error bars represent the standard error of the mean.

($t(58) = 3.95, p < 0.001, d = 0.51$). Thus, as in Experiment 12, participants were best predicted by a GP-based learning model with a universal kernel function.

Regarding the sampling strategy, the UCB strategy performed better than the other strategies, significantly so for the linear learning model,$t(58) = 3.38$,$p < 0.001$, $d =$ $d = 0.63$, and the GP-learning model, $t(58) = 2.43, p < 0.05, d = 0.32$. The probability of maximum utility sampling strategy resulted into better predictive performance for both the Bayesian mean tracker ($t(58) = 8.05, p < 0.001, d = 1.05$) and Kalman filter learning model ($t(58) = 7.22, p < 0.001, d = 0.94$). This mirrors previous findings on restless bandit tasks (Speekenbrink & Konstantinidis, 2015) and could indicate that some people switched to a non-contextual strategy within this more difficult set-up.

The median estimates of the GP-RBF-learning model for each participant are shown in Figure 7.7. The mean estimated temperature parameter was $\hat{\tau}^{-1} = 0.049$, which again indicates that participants mostly consistently chose the options with the highest predicted utility. The estimated error variance was $\hat{\sigma}_{\epsilon}^{2} = 5.07$ on average, which was very close to the actual variance of $\sigma_{\epsilon}^{2} = 5$ ($t(58) = 0.16, p > 0.05, d = 0.02$). The estimated length-scale parameter was clustered tightly around a value of $\hat{\lambda} = 10.31$. This indicates a tendency

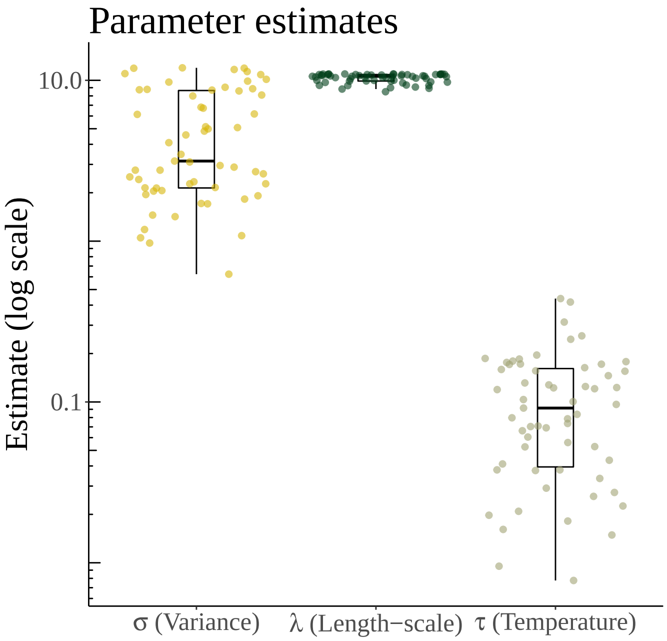


Figure 7.7: Median estimates of the error variance σ, the length-scale λ, and the temperature parameter τ for the GP-RBF model.

towards further extrapolation than in Experiment 12, but is still quite far removed from the level of extrapolation a linear function would provide.

7.7 Experiment 14: Continuous-non-linear features

The previous contextual multi-armed bandit experiments showed that most participants were able to learn how a multi-featured context differentially affects the rewards associated to decision alternatives. The goal of the third experiment was to investigate whether this would still be the case in an even more complicated environment with continuous valued features which are related to the options' rewards through non-linear functions. In order to cover a wide range of non-linearities, the functions relating contexts to rewards were themselves sampled from a Gaussian Process prior.

7.7.1 Participants

60 participants (28 female) with a mean age of 29 (SD=8.2) were recruited via Amazon Mechanical Turk and received $0.3 as a basic reward and a performance-dependent reward of up to $0.5. Participants earned $0.67 ± 0.04 on average. The experiment took around 12 minutes to complete.

7.7.2 Task and Procedure

The task was identical to that of Experiment 13, apart from the functions mapping inputs to outputs, which were drawn from a Gaussian process prior:

$$f_1(x_t) = 50 + f_1(x_{1,t}, x_{2,t}) \tag{7.20}$$

$$f_2(x_t) = 50 + f_2(x_{2,t}, x_{3,t}) \tag{7.21}$$

$$f_3(x_t) = 50 + f_3(x_{3,t}, x_{1,t}) \tag{7.22}$$

$$f_4(x_t) = 50 \tag{7.23}$$

$$f_j \sim \mathcal{GP}(m, K), j = 1, \ldots, 3 \tag{7.24}$$

where the mean function μ was set to 0 the kernel generating Σ was a radial basis kernel with a length-scale of $\lambda = 2$.

As in Experiment 13, the features were described numerically and could take values between -10 and 10. These values were sampled from a uniform distribution $x_{i,t} \sim \mathcal{U}(-10, 10)$. As before, the average expectation for all planets was 50 and the variance for the fourth arm was the lowest.

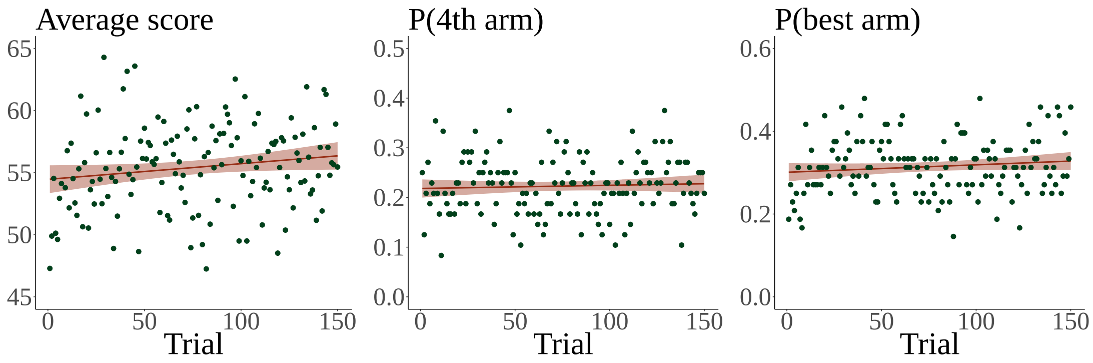
The procedure was identical to the one of Experiment 13.

7.7.3 Results

Behavioral results

On average, participants obtained rewards of 55.35 (SD = 6.33), which is again significantly above chance level, $t(59) = 5.85, p < 0.01, d = 0.85$.

Average scores increased over trials, $r = 0.19, p < 0.01$, but to a lesser extent than in



(a) Average scores per round.

(b) Proportion of 4th arm.

(c) Proportion of best arm.

Figure 7.8: Results of the binary CMAB task of Experiment 14. Average overall score (a), proportion of choosing the fourth arm (b), and proportion of choosing the best arm (c)

Experiment 13 (see Figure 7.8b), which might be due to the increase in difficulty of the task. The proportion of participants choosing the best option in a context did not increase much over trials, $r = 0.12, p < 0.05$ (see Figure 7.8c). The proportion of choosing the non-contextual arm did not significantly decrease over time, $r = 0.04, p > 0.05$. Overall, these results seem to indicate that participants struggled more than in any of the two prior experiments to perform well within the continuous non-linear task.

Modeling results

Modeling results are shown in Figure 7.9. Overall, the best performing model is the GP-RBF learning model paired with a upper confidence bound sampling strategy.

The contextual models predicted participants' choices only slightly better than the context-blind models ($t(59) = 2.7047, p < 0.01, d = 0.37$), but this was due to the linear model generating worse predictions than all of the other models ($t(59) = -3.59, p < 0.001$, $d = 0.50$). The Gaussian Process model parameterized with a radial basis function kernel generated better predictions than all of the other models ($t(59) = 4.13, p < 0.001$,

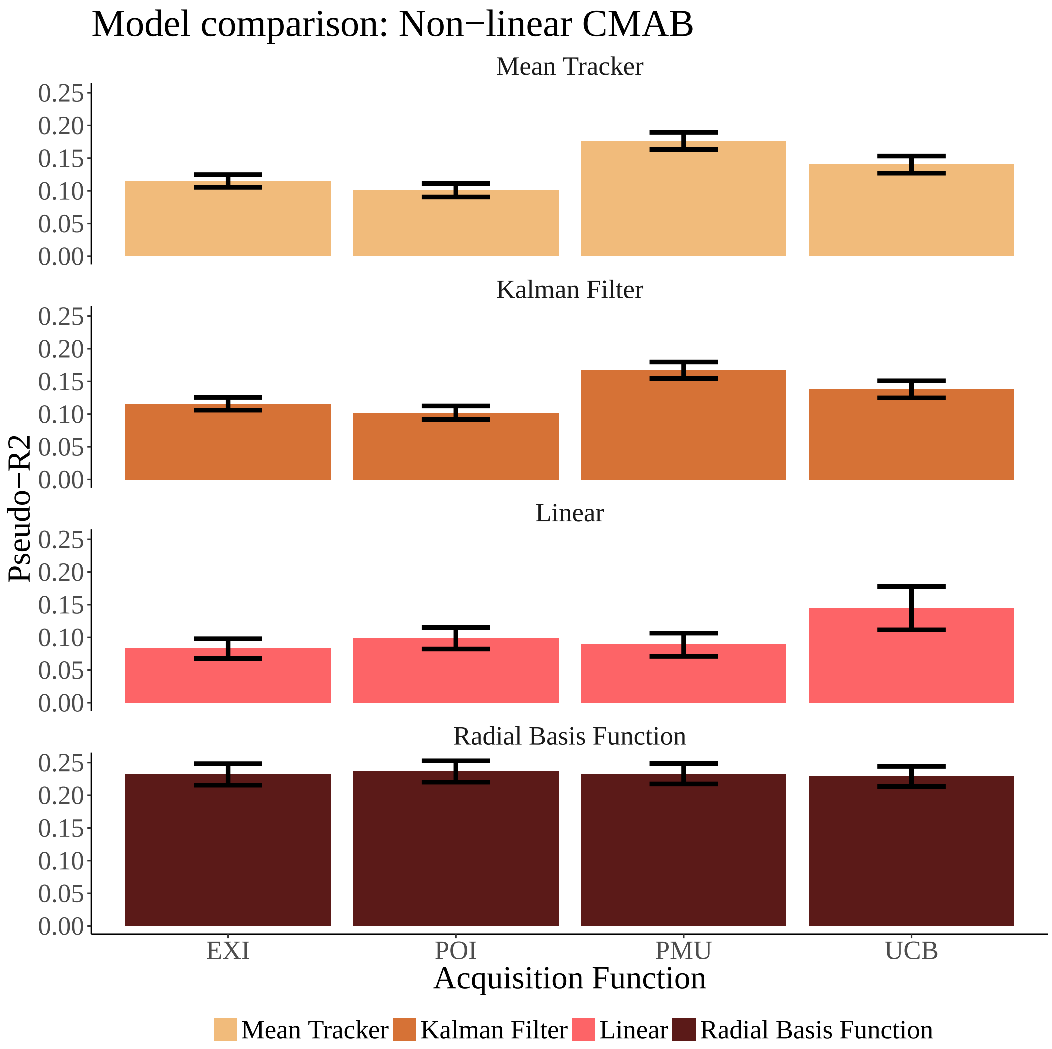


Figure 7.9: Predictive accuracy of the models for the CMAB task with continuous-non-linear cues in Experiment 14. Error bars represent the standard error of the mean.

$d = 0.57$). The probability of maximum utility sampling strategy generated better predictions within the the Kalman filter ($t(59) = 1.90, p < 0.05, d = 0.26$) learning models. There was no meaningful difference between different sampling strategies for the Gaussian Process model.

Figure 7.10 shows the median estimates of the GP-RBF learning model for all participants. The low average estimated temperature parameter $\hat{\tau} = 0.06$ again indicates that

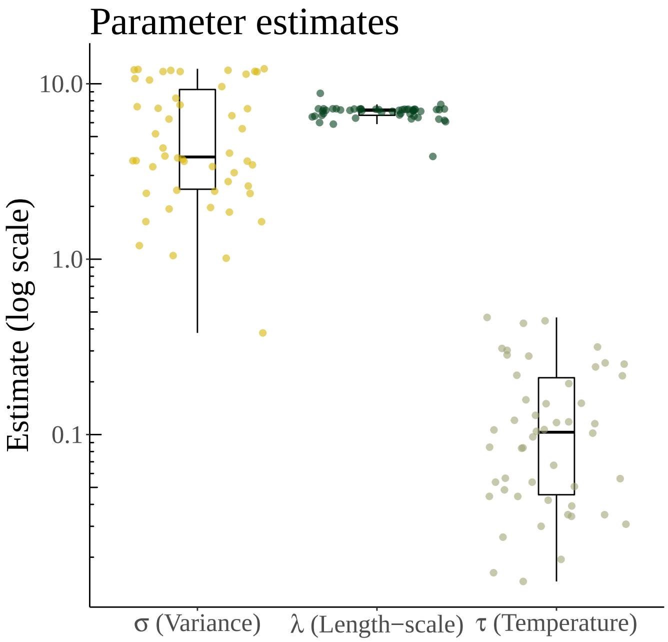


Figure 7.10: Median estimates of the error variance σ, the length-scale λ, and the temperature parameter τ for the GP-RBF model.

participants consistently chose the option with higher predicted rewards. The estimated length-scale clustered tightly along a value of $\hat{\lambda} = 6.86$ and the estimated noise variance of $\hat{\sigma}^2_\varepsilon = 5.71$ was again indistinguishable from the underlying true variance of $\sigma^2_\varepsilon = 5$ ($t(59) = 1.09, p > 0.05, d = 0.02$).

As this experiment required participants to learn three different non-linear functions, it might have been too difficult to learn the functions, so that some of them reversed back to learning in a context-free manner. Moreover, it seems less feasible in this scenario to distinguish between the different sampling strategies.

7.8 Inter-experimental model comparison

To assess how participants adapted to the different task environments, this section assesses how model performance and parameter estimates vary between different experiments. For this analysis, the focus will be on the model with a GP learning component and a UCB decision strategy because this strategy described participants reasonably well in all of the experiments and comes with the additional benefit that the parameters are very interpretable. For example, higher β-estimates are an indicator of more exploration behavior, higher λ-estimates indicate further generalization (i.e., the gradient of functional generalization), and higher noise parameters model an increasing tendency to perceive the underlying function as noisy. 7.11 shows the mean estimates of this model across all three experiments.

The overall predictive performance of the model was significantly higher within the first experiment as compared to the other two experiments ($t(152) = 4.52, p < 0.01, d = 0.79$). There was no meaningful difference between the continuous-linear and the continuous-non-linear experiment ($t(105) = -0.28, p > 0.05, d = 0.05$). Comparing the exploration-parameter β across experiments revealed that there was a negative correlation between the tendency to explore and the complexity of the task (ranked from discrete to non-linear) with $r = -0.18, p < 0.05$. This means that participants seem to explore less as the task becomes more difficult. The assumed noise term σ was estimated to be lower for the discrete experiment than for the continuous-linear experiments ($t(140) = 3.3, p < 0.01, d = 0.87$) which in turn was smaller than the estimated variance of the continuous-nonlinear experiment ($t(163) = 2.22, p < 0.05, d = 0.36$). Therefore, more difficult tasks can lead to a higher level of subjective noise within the learned functions. The length-scale parameter λ did not differ significantly between the three experiments (all $p > 0.5$). This indicates

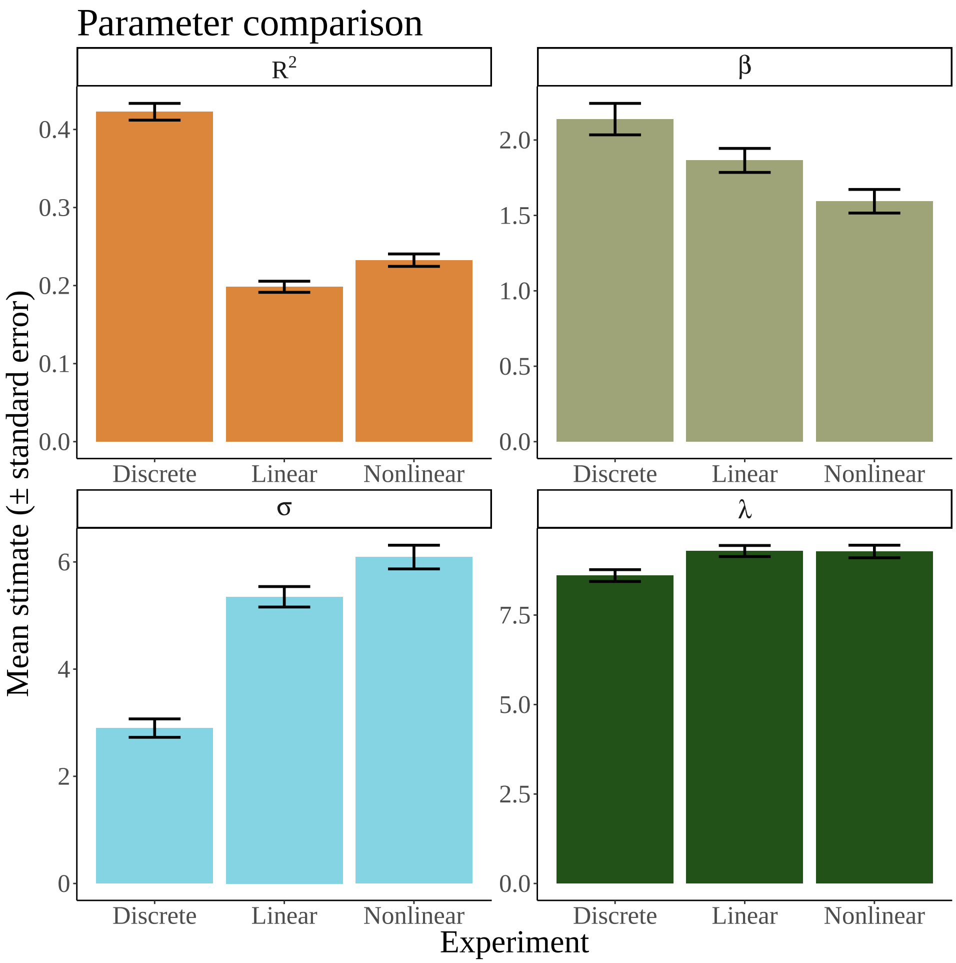


Figure 7.11: Mean estimates of the predictive performance R^2, the exploration parameter β, the error variance σ, and the length-scale λ across all experiments. Error bars represent the standard error of the mean.

that participants seem to approach diverse function learning tasks with a similar assumption about the underlying smoothness. While the assumed smoothness was less than the objective smoothness of the environment in the first two experiments, it was slightly higher in the last experiment.

In summary, comparing parameter estimates of the GP-RBF learning and a upper confidence bound sampling strategy between experiments showed that (1) the model captures participants' behavior best for the more simple discrete experiment, (2) participants seem to explore less in more difficult experiments, (3) the length-scale parameter modeling the assumed smoothness of the environment seems to be relatively stable across experiments, indicating a general approach to learning about unknown functions, and (4) the continuous-non-linear experiment was hard for participants as the model captured their behavior less well and assumed more noise overall.

7.9 Discussion

In this chapter, I have introduced the contextual multi-armed bandit (CMAB) task as a paradigm to investigate behavior in situations where participants have to learn functions and simultaneously make decisions according to the predictions of those functions. The CMAB is a natural extension of past research on function learning, experience-based decision making in multi-armed bandit tasks, as well as the spatially-correlated multi-armed bandit introduced in Chapter 6. In three experiments, I assessed people's performance in a CMAB task where a general context affected the rewards of options differently (i.e. each option had a different function relating contexts to rewards). Even though learning multiple functions simultaneously is likely to be more complex than learning a single function (as is common in previous studies on function learning and multiple cue probability learning), on average, participants were able to perform better than expected if they were unable to take generalize from the provided context. This was even the case in a rather complex situation where the functions were sampled from a general class of non-linear functions, although performance dropped considerably as compared to environments with linear functions.

The presented experiments focused on a general context which differentially affected the outcomes of options. This is different than the CMAB task of Stojic et al. (2015), in which the features had different values for each option, while the function relating the contexts to rewards was the same for each option. Future studies could combine these paradigms and incorporate both option-specific (e.g., the type of restaurant) as well as general (e.g., the area in which the restaurants are located) contextual features, possibly allowing these to interact (e.g., a seafood restaurant might be preferable to a pizzeria in a fishing village, but not a mountain village).

Modeling function learning as Gaussian Process regression allowed to incorporate both rule-based and similarity-based learning in a single framework. In all three environments, participants' learning appeared to rely on a universal learning and generalization strategy based on a Gaussian Process regression with a radial basis function kernel. This is a universal function learning engine and assumes rather smooth functions to generalize from previous observations to similar contexts. This is close to a combination of exemplar learning, in which past observations are memorized and averaged when making new predictions, and similarity-based extrapolation.

The results regarding the sampling strategy were somewhat less consistent. When the features comprising the contexts were binary, people appeared to rely on a strategy in which they focus on the probability of improvement over past outcomes. In environments with continuous contextual features and linear and non-linear functions, they appeared to trade-off their expectations and uncertainties more explicitly, and their decisions were best described by an upper confidence bound sampling strategy. Participants may have adapted their decision strategy to the task at hand. In a relatively simple scenario with binary contextual features, they appeared to focus on trying to improve upon past outcomes. When there are few unique and distinct contexts, it might be possible to memorize the average outcomes for those contexts, such that trying to maximally improve upon the current best option may be a feasible and efficient strategy. As the environment becomes more complicated, they seemed to balance their expectations and uncertainties more explicitly. Upper confidence bound sampling is not only the only sampling strategy with provable good regret (Srinivas et al., 2012), but it has also been proposed as a dynamic shaping bonus within the exploratory choice literature before (Daw et al., 2006).

The environment involving non-linear functions sampled from a Gaussian Process prior was more difficult than the others, and a proportion of participants appeared unable to learn the functions, performing more in line with a context-blind learning strategy (Kalman filter) that treats the task as a restless bandit in which the expected rewards fluctuate over time but where these fluctuations are not predictable from the changes in context. The combination of a Kalman filter learning model with a "probability of maximum utility" decision strategy that described these participants best has been found to describe participants behavior well in an actual restless bandit task (Speekenbrink & Konstantinidis, 2015) and here might have indicated the limits of participants' learning abilities.

While previous research on function learning has found a bias towards linear functions in such environments (e.g., Lucas et al., 2015), this was not the case in the presented experiments. This could be due to the increased complexity of learning multiple functions simultaneously, or due to participants' learning the functions with the purpose of making good decisions, rather than to accurately predict the outcomes as such. While good performance in standard function learning experiments requires accurate knowledge of a function in its whole domain, more course-grained knowledge may suffice in a CMAB task, where it might be enough to know which function has the maximum output for a given context.

Although the true functions relating contexts to rewards were smoother than those assumed by the estimated length-scale parameters within two of the three experiments, it is not clear whether this constitutes as a bias (in the psychological sense) or might be a useful prior to have in the real world. Approaching a task with a relatively less smooth kernel may be wise when there is uncertainty about the level of smoothness in the environment. Furthermore, assuming unsmooth functions and learning in a situation of smoother than expected functions might lead to smaller errors than the other way around, that is, expecting smooth functions and having to learn in unsmooth environments. Further probing the effects of mismatched priors on generated rewards will be the focus of Chapter 8.

"And now, at last, I come to the humanly significant and exciting problem: namely, what are the conditions which favor narrow strip-maps and what are those which tend to favor broad comprehensive maps not only in rats but also in men?"

Edward C. Tolman, 1948

8

From behavior to theory

This chapter investigates the previously reported result that participants underestimate smoothness using a Bayesian optimization simulation with mismatched priors.

8.1 Introduction

One of the main findings of the last two chapters was that the estimated λ-parameter, which governs generalization within the radial basis function kernel, tends to be smaller than expected given the environments in which participants had to search for rewards. I have found this to be the case for both the spatially correlated and the contextual multi-armed bandit and across different versions of both tasks. As λ scales the correlation between points as their distance grows, smaller λ-estimates indicate that people generalize to a lesser extend than what they could for the underlying smoothness of the task. This raises the following questions: Why is the estimated length-scale parameter frequently smaller than it could be? Do people act irrationally when they do not tune their tendency to generalize to the underlying environment? Or might there be another explanation for the found mismatch between the smoothness participants seem to expect and the smoothness they observe?

This chapter investigates how much mismatched expectations about the generalizability of the underlying function affect performance in function exploration-exploitation tasks. More specifically, I use simulations to assess what kind of error is worse, expecting smooth and having to optimize unsmooth functions or expecting unsmooth and having to optimize smooth functions. Furthermore, my simulations will assess the extend to which a Bayesian optimization algorithm performs worse if assumptions are mismatched as compared to getting the expected smoothness right. The results of these simulations will have consequences as to whether or not the previously found human behavior can be seen as bias or as adaptive behavior.

8.2 Mismatched Bayesian optimization

The goal of Bayesian optimization is to find the maximum of a target function $f(x)$ on some bounded set $\mathcal{X}$, which I will take to be a subset of $\mathbb{R}^D$ (Brochu et al., 2010; Snoek et al., 2012; Shahriari et al., 2016) by using a Gaussian Process ($\mathcal{GP}$) as a probabilistic model for $f(x)$ and then exploiting this model via a sampling strategy to decide where in $\mathcal{X}$ to evaluate the function next (Ghahramani, 2015). This set-up is exactly the same as in the paradigms used in Chapters 6 and 7. Bayesian optimization algorithms are frequently also applied for problems of black box function optimization*. Even though different sampling strategies have been compared statistically before (cf. Brochu et al., 2010; Snoek et al., 2012) and Bayesian optimization is frequently applied to optimize neural network hyperparameters (Bergstra et al., 2011), fitness landscapes (Romero et al., 2013), or even to replace more traditional numerical methods (Hennig & Hauberg, 2014), surprisingly little is known about how choosing a particular kernel affects Bayesian optimization routines in practice.

In fact, many practitioners seem to either use the radial basis function kernel or a Matérn kernel in order to encode assumptions about smooth or medium smooth target functions (to the best of my knowledge first proposed by Snoek et al., 2012). Thus, assessing how different kernels affect the performance of the Bayesian optimization routine is not only an important psychological question, it also matters for practitioners of Bayesian optimization in order to decide when to use which kernel, potentially providing a case in which cognitive psychology can aid machine learning research.

Given two kernels k_1 and k_2, I will define mismatched optimization as optimizing a func-

*It is called black box as the underlying function or at least some important characteristics of it are either unknown or expensive to evaluate.

tion $f \sim \mathcal{GP}(0, k_1(x_s, x_s)$ sampled from a GP-prior parameterized by k_1 using a GP parameterized by k_2, iff k_1 and k_2 are not the same kernel. This means that a target function is sampled from a Gaussian Process parameterized by one kernel and then optimized with a GP parameterize by another kernel. As the kernel can encode assumptions about the expected smoothness of the target function, mismatched optimization occurs if these assumptions are misaligned. I will refer to the kernel that has generated the target function as the *teacher* and to the kernel that is used to optimize the target function as the *student*.

8.2.1 Sampling strategies

A simulation assessing the mismatch of assumptions in Bayesian optimization might not only vary the used kernels, but also the chosen sampling strategy. In the following simulations, I will assess 6 sampling strategies in terms of their performance within mismatched optimization scenarios. Some of these strategies have been used in the model comparisons before, whereas others were too costly to be evaluated in a psychological setting. Let me quickly recapitulate these sampling strategies one more time.

Upper confidence bound (UCB) sampling picks the next point that currently has the highest upper confidence bound (Srinivas et al., 2010):

$$\text{UCB}(x) = \mu(x) + \beta\sigma(x) \tag{8.1}$$

UCB sampling is an optimistic strategy that samples based on an explicit exploration-exploitation trade-off, and has been proven to produce sub-linear regret (Srinivas et al., 2012), where regret is measured by the difference between the currently produced output and the best possible output on every trial t

$$r_t = f(x_{\max}) - f(x_t) \tag{8.2}$$

Moreover, upper confidence bound sampling has been the strategy that has predicted participants' behavior best in the experiments presented in Chapters 6 and 7, indicating that a trade-off between exploration and exploitation captures human behavior well.

The Probability of Improvement (POI) sampler picks the next point that currently shows the highest probability of being better than an incumbent point which is normally set to $x^+ = \arg\max_{x_i \in x_{1:t}} f(x_i)$—i.e., the best outcome observed so far (Kushner, 1964):

$$\text{POI}(x) = P(f(x) \geq f(x^+ + \xi)) = \Phi\left(\frac{\mu(x) - f(x^+ - \xi)}{\sigma(x)}\right) \tag{8.3}$$

where $\Phi(\cdot)$ is the normal cumulative distribution function and $\xi \geq 0$ is a trade-off parameter controlling exploration. This sampling strategy predicted participants' choices best within the contextual multi-armed bandit experiment with discrete binary cues in Chapter 7.

The Expected Improvement (EXI) assesses each point by how much in expectation it promises to be better than an incumbent point x^+ (Močkus, 1975):

$$\text{EXI}(x) = \begin{cases} (\mu(x) - f(x^+) - \xi)\Phi(Z) + \sigma(x)\phi(Z), & \text{if } \sigma(x) \geq 0 \\ 0, & \text{if } \sigma(x) = 0 \end{cases} \tag{8.4}$$

where $\phi(\cdot)$ is the normal probability density function and $Z = (\mu(x) - f(x^+) - \xi)/\sigma(x)$.

The remaining three sampling strategies are all based on relatively costly Monte Carlo methods. They are included in the simulations because they are frequently applied in computational optimization scenarios. Yet, I have not used them as candidate sampling strategies of human search in the experiments presented earlier. This was mostly due to the fact that these acquisition functions are computationally expensive to run such that it would not have been possible to use them for each participant individually while also optimizing kernel hyper-parameters. For example, the probability of maximum utility sampling strategy, which can be seen as a more simplistic case of the following strategies, already takes three times as long as all the other sampling strategies when parameters were optimized at the same time, running for about 4 hours per participant.

Predictive entropy search (PES) is an information-based acquisition function that samples points that promise to maximally reduce the differential entropy of an unknown optimizer $x^* = \arg\max_{x \in \mathcal{X}} f(x)$ (Hennig & Schuler, 2012; Hernández-Lobato et al., 2014). After having observed data $\mathcal{D}_n$, the posterior distribution of x^* is $p^*(x|\mathcal{D})$. The predictive entropy search sampler then is

$$\text{PES}(x) = H(y|\mathcal{D}_n, x) - \mathbb{E}_{x^*|\mathcal{D}_n}\left[H(y|\mathcal{D}_n, x, x^*)\right] \tag{8.5}$$

where $H(x^*|\mathcal{D}_n)$ denotes the differential entropy of $p^*(x|\mathcal{D})$. The differential entropy can be approximated by Monte Carlo sampling from the posterior $\mathcal{GP}$ (Hernández-Lobato et al., 2014).

Minimum regret point (MRP) sampling samples the point producing the lowest simple regret (Metzen, 2016). The expected simple regret (ER) of selecting x under $p(f)$ can be expressed as

$$\mathbb{E}_{p(f)}[R_f(x)] = \mathbb{E}_{p(f)}[f(x_{\max}) - f(x)]. \tag{8.6}$$

MRP picks the point which with minimal regret under $p_n(f)$, $\tilde{x} = \arg\min_x ER(p_n)(x)$.

$$\mathrm{MRP}(x) = \min \mathrm{ER}(p_n)(\tilde{x}) - \mathbb{E}_{y|p_n(f)}, x[\min_{\tilde{x}} \mathrm{ER}(p_n^{[\mathcal{X},y]})(\tilde{x})] \tag{8.7}$$

where $p_n^{[\mathcal{X},y]} = p(f|\mathcal{D} \cup \{x^q, y\})$ is the updated probability distribution of f after having queried x^q and observing $y = f(x^q) + \varepsilon$. MRP sampling can also be modified to account for the inherent uncertainty about $\tilde{x}$ via marginalization leading to Minimum Regret Search (MRS) as described in detail by Metzen (2016).

$$\mathrm{MRS}(x) = \mathbb{E}_{\tilde{x}\sim p_n^*}[\mathrm{ER}(p_n)(\tilde{x})] - \mathbb{E}_{y|p_n(f),x^q}[\mathbb{E}_{p^*_{\mathcal{D}_n \cup \{(\mathcal{X}^q,y)\}}}[\mathrm{ER}(p_n^{[\mathcal{X},y]})(\tilde{x})]] \tag{8.8}$$

8.2.2 Encoding Smoothness

I will again use the Matérn class of kernel functions to encode the expected smoothness of the target function explicitly as in Chapters 3-4. To recaptitulate this kernel class one more time, the Matérn covariance between two points separated by $\tau = |x - x'|$ distance units is

$$k \quad (\tau) = \sigma_f^2 \frac{2^{1-\nu}}{\Gamma(\nu)} \left(\sqrt{2\nu}\frac{\tau}{\lambda}\right)^{\nu} K_\nu \left(\sqrt{2\nu}\frac{\tau}{\lambda}\right) \tag{8.9}$$

where Γ is the gamma function, K_ν is the modified Bessel function of the second kind, and λ and ν are non-negative covariance parameters. A $\mathcal{GP}$ with a Matérn covariance has sample paths that are $\nu - 1$ times differentiable. When $\nu = p + 0.5$, the Matérn kernel can be written as a product of an exponential and a polynomial of order p. When $\nu = p + 0.5$, the Matérn kernel can be written as a product of an exponential and a polynomial of order p.

$$k_{p+0.5}(\tau) = \sigma_f^2 \exp\left(-\frac{\sqrt{2\nu}\tau}{\lambda}\right)\frac{\Gamma(p+1)}{(2p+1)} \times \sum_{i=0}^{p} \frac{(p+i)!}{i!(p-i)!}\left(\frac{\sqrt{8\nu}\tau}{\lambda}\right)^{p-i} \tag{8.10}$$

I will compare two extreme cases in the following simulations, the Ornstein-Uhlenbeck process (a Matérn 1/2 kernel),

$$k(\tau) = \sigma_f^2 \exp(-\tau/\lambda) \tag{8.11}$$

that results when setting $p = 0$ and the radial basis function kernel that results when $p \to \infty$,

$$k(\tau) = \sigma_f^2 \exp(-\tau^2/2\lambda^2). \tag{8.12}$$

Figure 3.5 of Chapter 3 showed how differently smooth priors result into different levels of posterior generalization, i.e. how much information one point provides about proximal points.

8.2.3 Past research on mismatched assumptions

How mismatch affects the performance of Gaussian Process regression methods has been investigated before. In fact, the derivations of learning curves discussed in Chapter 3 have also been applied to mismatched scenarios.

Sollich (2002) pioneered this work by deriving approximations to the learning curves for the case of mismatched models and found that, for large input space dimensionality, there are universal plateaux in the learning curves, with transitions in between that can exhibit arbitrarily many over-fitting maxima. In lower dimensions, plateaux also appear, and the learning curve remains dependent on the mismatch between student and teacher even in the asymptotic limit of a large number of training examples. In particular, these results showed that learning with excessively strong smoothness assumptions is unwise: a student with a radial basis function kernel will learn a rougher teacher function from a Matérn family only "logarithmically slowly".

Later, Sollich (2005) investigated whether GPs can be made robust against mismatch effects by optimizing the student's model hyper-parameters to maximize the data likelihood. He derived an approximation for the average evidence and used it to predict the optimal hyper-parameter values and the resulting generalization error. This showed that, in lower-dimensional learning scenarios, evidence maximization eliminates logarithmically slow learning and recovers the optimal scaling of the decrease of generalization error with training set size.

Furthermore, van der Vaart & van Zanten (2008) derived rates of contraction of posterior distributions on nonparametric or semiparametric models based on Gaussian Processes and showed that smoother functions show faster contraction rates than rough ones and that having the right expectation about the underlying smoothness improves the rate of convergence.

Srinivas et al. (2012) showed that GP-UCB has sublinear regret for cases when the true underlying function lies in a low Reproducible Kernel Hilbert Space norm, even in the agnostic setting of not precisely defining the kernel function.

To the best of my knowledge, no further theoretical proof for Bayesian optimization in the mismatched setting exists.

Therefore, I will utilize extensive simulations of mismatched optimizations in order to assess the effect of a misspecification of the target function's smoothness empirically. Hopefully, this will then shed some light onto human generalization behavior found within the spatially correlated and the contextual multi-armed bandit described before while also being informative for applied settings of Bayesian optimization.

8.3 Simulation 1: Simple mismatch

In the first simulation, the target function

$$y_t = f(x_t) + \varepsilon \tag{8.13}$$

was created by sampling

$$f \sim \mathcal{GP}(0, k(x, x')) \tag{8.14}$$

and $\varepsilon \sim \mathcal{N}(0, 10^{-3})$ from a teacher kernel and a student kernel was used to optimize it. This procedure is repeated 100 times for every student-teacher-sampling strategy combination resulting in 4 different comparisons for every sampling strategy, i.e. all combinations of the Ornstein-Uhlenbeck (Matérn 1/2) and the radial basis function kernel as either the teacher or the student kernel.

8.3.1 Simple regret

In the first simulation, I fix the length-scale parameter to $\lambda = 0.1$ for both student and teacher kernels. Figure 8.1 shows the median regret calculated over 100 simulations for all of the sampling strategies over 100 trials.

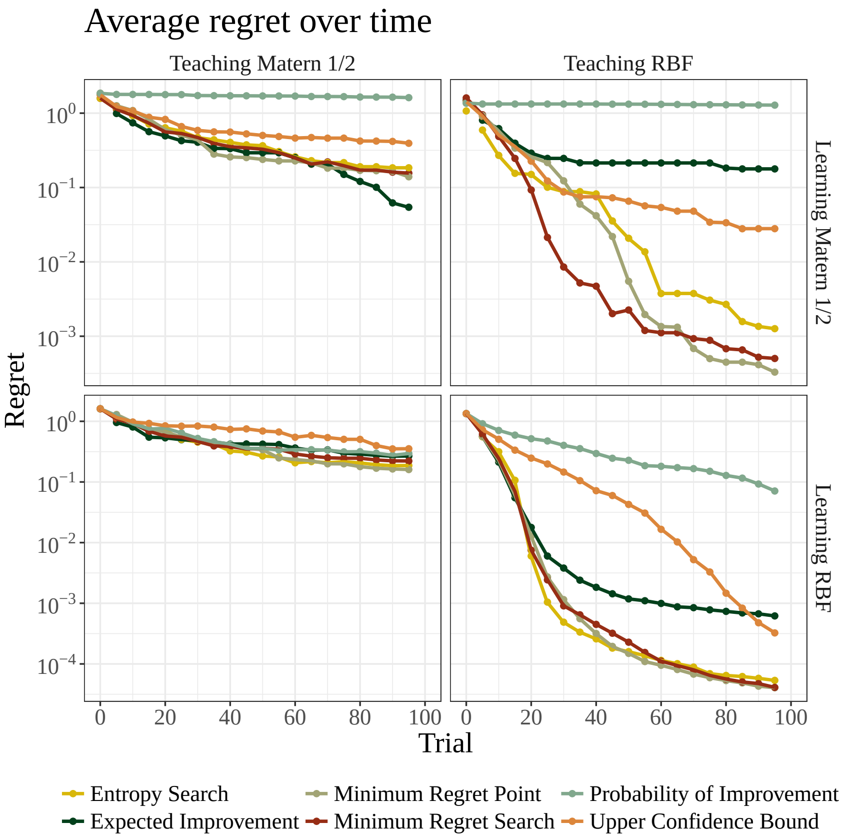


Figure 8.1: Median regret over time for different sampling strategies in matched and mismatched conditions.

Maximizing an unsmooth function is harder than maximizing a smooth function. The lowest regret achieved after 100 trials is within the region of 10^{-5} produced by the minimum regret search sampling strategy for the scenario in which a radial basis function student maximizes a radial basis function teacher. The upper confidence bound sampling algorithm reaches a regret within the region of 10^{-3} within the same scenario. A Matérn 1/2 (i.e. Ornstein-Uhlenbeck) student trying to optimize a radial basis function teacher can also perform reasonably well. The lowest median regret reached in such a scenario is in the region of 10^{-4}, again by the minimum regret search acquisition function. The upper confidence bound sampling algorithm achieved a regret in the margin of 10^{-2} in this scenario. Optimizing an unsmooth function turned out to be harder overall. In the matched case, with a Matérn 1/2 teacher and a Matern 1/2 student, the best regret achieved was in the region of $10^{-1.5}$, again achieved by the minimum regret sampling strategy. The upper confidence bound sampling strategy achieved a regret in the region of 10^{-1} in this scenario. Using a radial basis function kernel student to optimize a Matern 1/2 kernel lead to the worst regret overall. In this scenario, the minimum regret search only achieved a median regret of 0.16 and the upper confidence bound sampling algorithm of 0.35 after 100 trials.

These results show that mismatched optimization leads to worse performance as compared to expecting the right level of smoothness. The worst possible regret results from expecting a smooth function and in reality optimizing a very rough function. Overall, the information-based sampling strategies performed best. Notably, mismatch mattered more than specifying the right sampling strategy. This is surprising as debates in the machine learning literature frequently concern the choice of an adequate sampling strategy and not specifying the right kernel.

8.4 Simulation 2: Regret in higher dimensions

Simulation 1 focused on the most simplistic case of having to optimize a univariate function. However, the previous experiments also showed an effect of undergeneralization when participants had to optimize a function on a bivariate grid or using a context incorporating three variables. Therefore, the next simulation focused on mismatch in higher dimensions. The same simulation as before was repeated but this time over 50 trials and for $d = \{2, 3, 4\}$ dimensions of the input space over which the function had to be optimized. Figure 8.2 shows the median root mean squared difference between each sampling strategy's

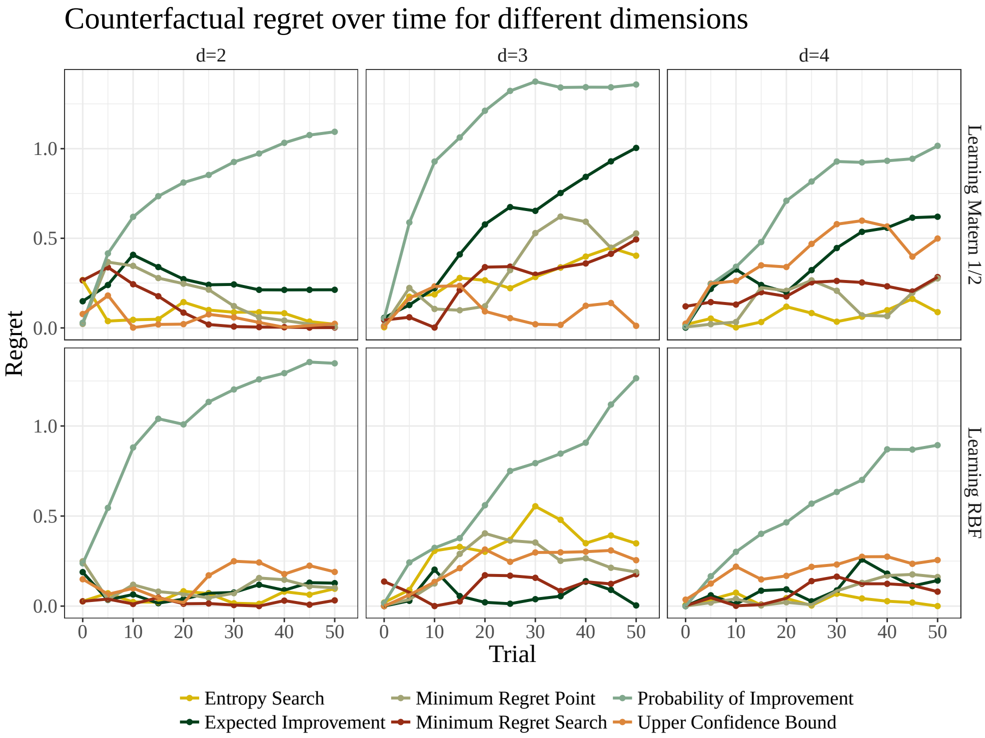


Figure 8.2: Median counter-factual regret over 50 trials for different input-dimensions.

performance in the mismatched setting compared to the same strategy's performance in the matched setting for each dimension. This provides a relative measure of counter-factual regret, i.e. by how much could each of the strategies have done better on each trial if it had been specified with the right prior about the target function's smoothness. It can be seen that, whereas most sampling strategies seem to recover from misspecified assumptions in the bi-variate case, the effect of misspecifications exacerbates over time in higher dimensions.

Interestingly, when just looking at the performance of the upper confidence bound sampling strategy, the counter-factual regret of a Matérn 1/2-student for $d = \{2, 3\}$ dimensions decreases again and is indistinguishable from 0 after 50 trials, even though this is not completely the case for $d = 4$. However, the counter-factual regret of a radial basis function kernel trying to optimize Matérn 1/2 never goes back to 0 again. In some cases, expecting a rough function and having to optimize a smoother functions is better than the other way around. However, this is not necessarily often the case. For example, looking at the information-based sampling strategies in the cases when an RBF kernel is used to optimize functions sampled from a Matérn 1/2-kernel, the regret does not seem to go up in higher dimensions. Thus, there are also cases when expecting smoother functions does not hurt performance.

8.5 Simulation 3: Optimizing hyper-parameters

Sollich (2005) has shown that some of the effects caused by mismatched assumptions can be overcome by optimizing hyper-parameters in the pure learning case. Therefore, I next assessed if this also holds true for the optimization case, when the student kernel's hyper-parameters ϑ are optimized via the log marginal likelihood

$$\log p(y|X, \vartheta) = -\frac{1}{2} y^\top K_y^{-1} y - \frac{1}{2} \log |K_y| - \frac{n}{2} \log 2\pi. \tag{8.15}$$

As before, I set the length-scale of the teacher kernel to $\lambda = 0.1$. However, this time the student's length-scale is optimized on every step of the optimization routine. The simulation is again run over 100 trials and repeated 100 times for each sampling strategy. Figure 8.3

shows the resulting median regret per trial for all sampling strategies and every student-teacher combination.

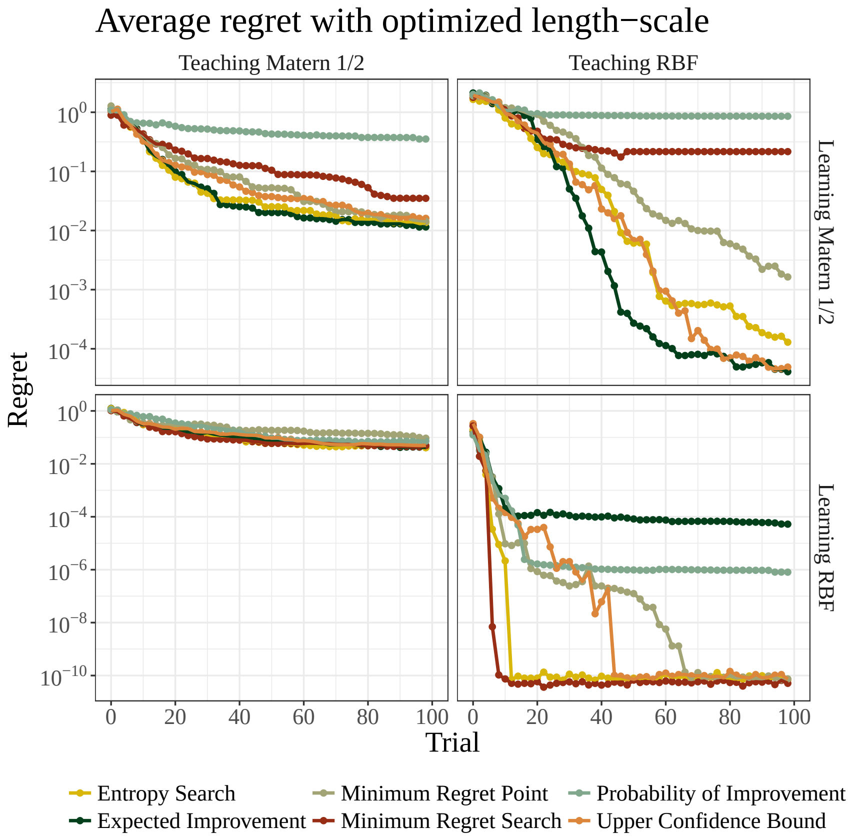


Figure 8.3: Median regret over time for different sampling strategies in matched and mismatched conditions.

The resulting regret shows that mismatch still affects the performance of the Bayesian optimization routine. In cases where the teacher and student both come from a Matérn 1/2 kernel, the lowest overall reached regret is in the region of 10^{-2}, whereas the best regret when trying to optimize the Matérn 1/2 kernel by using a RBF kernel is $10^{-1.5}$. The effect of mismatch is even stronger in the case of an RBF-teacher, where the best overall regret in the matched case is 10^{-10} but only $10^{-4.5}$ for the mismatched case. This means that optimizing hyper-parameters leads to better performance in all cases, no matter if mismatch occured or not. However, having the right assumptions about the target function's smoothness matters even more in cases when the hyper-parameters are optimized. Given that participants' estimates of λ seemed to be relatively stable across experiments in Chapters 6-7, they likely did not adapt much to the underlying smoothness, but rather approached different settings with similar expectations.

8.6 Simulation 4: The effect of λ on regret

Throughout all of the experiments in Chapters 6-7 I have used a radial basis function kernel to describe participants' generalization. As the estimated λ, which I interpreted as a gradient of functional generalization, turned out to be lower than what the underlying function allowed for, I next assess the effect of mismatched λs directly.

Two kernels, a student and a teacher, were used. Both kernels were set up to be radial basis function kernels, where the teacher kernel was parameterized with a length-scale λ_0 and the student kernel with a length-scale λ_1. For situations in which $\lambda_0 \neq \lambda_1$, the optimization routine can be seen as mismatched before. If $\lambda_1 > \lambda_0$, then the student kernel tends to overgeneralize from the observed points, whereas if $\lambda_1 < \lambda_0$ the student kernel tends to undergeneralize. Every possible combination between $\lambda_0 = \{0.1, 0.2, \cdots, 1\}$ and $\lambda_1 = \{0.1, 0.2, \cdots, 1\}$ was created leading to 100 different combinations of student-teacher optimization scenarios. For every run of the simulation, I sampled a target function from a univariate Gaussian Process parameterized by λ_0 and then used a Gaussian Process parameterized by λ_1 and one of the 6 sampling strategies to optimize the target function over 50 trials. This was repeated for 100 runs for every student-teacher-sampling strategy combination leading to 60,000 runs of the simulation in total (i.e., 10 different λ_0-values times 10 different λ_1-values times 6 sampling strategies).

Figure 8.4 show the median regret for 100 runs for all sampling strategies and every λ_0-λ_1-combination at trial number $N = \{1, 5, 10, 15, 50\}$.

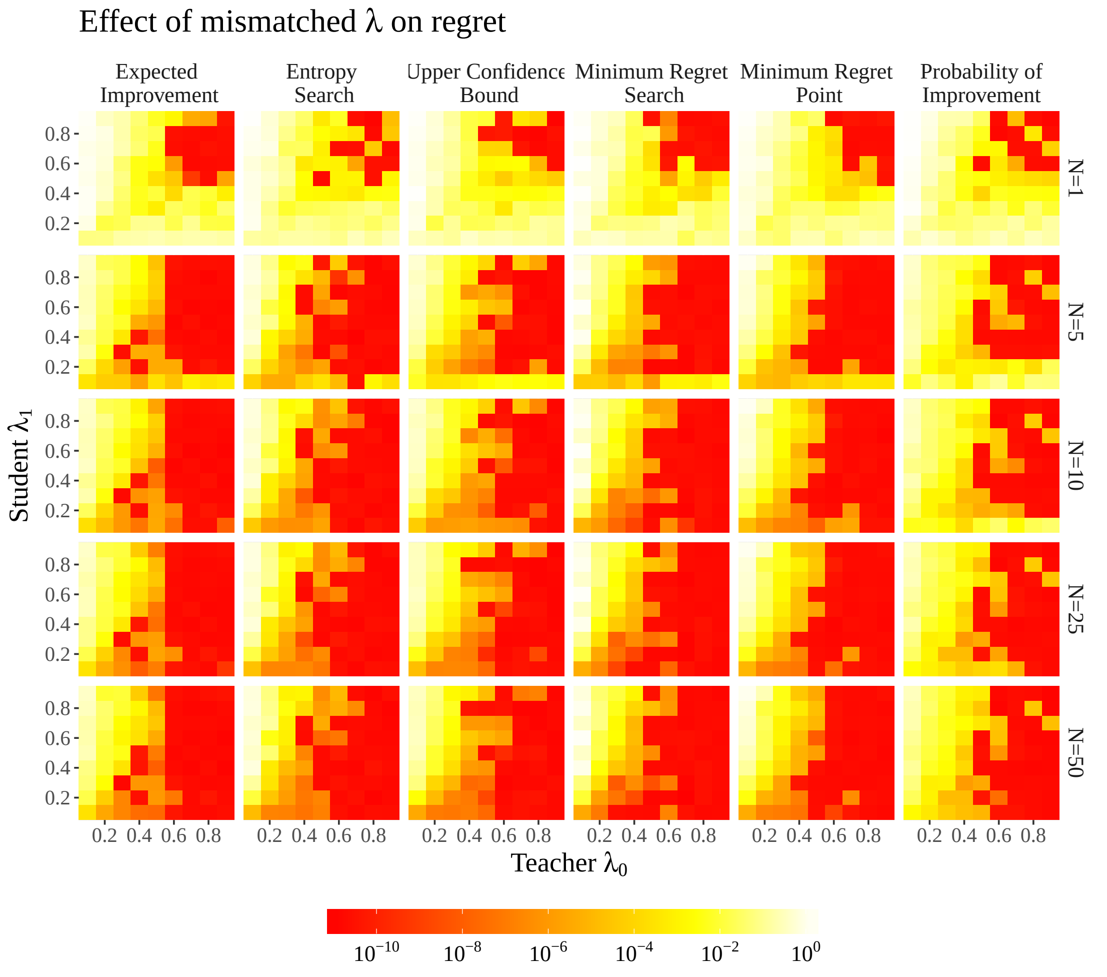


Figure 8.4: Median regret for different student-teacher combinations and sampling strategies at trial number $N = \{1, 5, 10, 15, 50\}$.

The results of this simulation revealed several interesting factors influencing regret. First of all, regret decreases fastest when smooth teachers are optimized by equally smooth students as can be seen in the right upper corner of the heat-maps in Figure 8.4. Furthermore, there are cases in which a student can recover from mismatched assumptions over time, producing similar regret as if a completely matched student had been used. Importantly, this happens much more frequently for students that were parameterized to be less smooth than the teacher than the other way around, as can be seen on the right lower part of the heat-maps throughout all optimizations. The value of λ_1 producing the lowest average regret over all trials was $\hat{\lambda}_1 = 0.2$ over all sampling strategies and $\hat{\lambda}_1 = 0.2$ for the upper confidence bound sampling strategy. Estimating the best possible alignment between λ_0 and λ_1 to produce the lowest regret revealed that under estimating λ_0 by an average of about 0.19 over all sampling strategies and by 0.21 for the upper confidence bound sampling strategy can potentially produce the best regret. Thus, these results provide strong evidence that lower values of λ do not necessarily constitute a bias but can sometimes lead to competitive overall regret. This is most likely due to the additional flexibility gained by assuming less smooth functions. If the true value of λ_0 is unknown, and optimizing λ_1 unfeasible, erring on the side of assuming a rougher environment thus seems wise.

8.7 Discussion

Bayesian optimization is not only a popular method to optimize black box functions, I have also found it to predict participants' search behavior in both spatially correlated and contextual multi-armed bandit tasks. However, in both of these paradigms the estimated length-scale parameters showed that participants seem to underestimate the underlying smoothness of the environment, i.e. they did not generalize as much as they could have. In this chapter, I have tried to assess the effect of this underestimation using various simulations manipulating the difference of smoothness between a teaching kernel from which a target function is generated and a student kernel that is used to optimize the target function.

First of all, from a machine learning perspective, the results suggest that the wrong prior assumptions about a target function's smoothness can lead to a large increase in regret, that this effect gets worse in higher dimensions, and remains even if hyper-parameters are optimized. Moreover, often-times choosing the right prior matters far more than choosing the sampling strategy. Thus, taking the term "black box optimization" too literally can lead to noticeable under-performance. My results thus tell a cautionary tale about how thinking hard about prior assumptions can sometimes be more important than choosing a particular sampling strategy.

Secondly, from a psychological perspective, the results suggest that underestimating the underlying smoothness is not always maladaptive. For example, in the simple regret simulation, assuming extremely rough functions still leads to comparably competitive performance when optimizing an extremely smooth function. Moreover, in higher dimensions, the counter-factual regret of the upper confidence bound sampling strategy was relatively robust to misspecification and (at least for lower dimensions) was able to fully recover from misspecified priors. In the scenario where the length-scale is optimized on every step, underestimating the underlying smoothness can lead to under-performance. However, it is likely not the case that participants adapt their length-scale according to the evidence seen on every trial, especially given very stable estimates across the different settings in the multi-armed bandit experiments. In the final simulation I generated different levels of mismatch by parameterizing radial basis function kernel teachers and students with different values of the length-scale λ. The results showed that undergeneralization can be beneficial for all Bayesian optimization routines, with the best possible λ_1 to be used for optimizing a teacher with λ_0 to be –on average– smaller by 0.19 (at trial 100).

The simulations presented here have tried to shed more light onto the finding of participants' consistently lower than necessary length-scale parameter estimates in all of the bandit experiments presented previously. The motivation was to assess if the apparent undergeneralization found in such tasks is practically damaging participants' performance and by how much. I have found that it is not always bad for a Bayesian optimization routine to be pessimistic, that is to expect rough functions and in fact being confronted with smoother functions. Sometimes being pessimistic about the world one needs to learn in can pay off in the end. Thus, under-generalization might not be a bug but rather a feature of human behavior.

Nonetheless, the question of why people seem to undergeneralize as much as they do needs to be addressed further. One mechanism proposed in the literature is that people might try to avoid overgeneralization from encountered exemplars (Gershman & Daw, 2017). Such a mechanism could be especially important if a learning algorithm is paired with a naive and optimistic sampling approach such as upper confidence bound sampling or within scenarios requiring to avoid very bad outcomes. Other explanations such as memory restrictions or sparse updates could also be investigated. Further disentangling these mechanisms should be the goal of future experiments on human behavior in contextualized bandit settings.

"And then will follow the period of slow and steady progress, varied by a certain amount of wholesome interruption."

Edward Titchener, 1914

9 Conclusion

This chapter concludes the thesis by summarizing the main findings, discussing possible shortcomings, and proposing future directions.

9.1 Lessons learned

I began this thesis by posing the fundamental computational problem of generalization: how do people know what to do in novel situations? I then proposed that Gaussian Process regression offers a unifying theory to explain human generalization across a wide range of cognitive domains. I will now briefly summarize the main findings of this thesis, examine potential shortcomings, and propose further experiments and steps towards theory unification.

9.1.1 Gaussian process learning curves match perceived predictability

The first experiment assessed if the characteristics theoretically influencing the generalizability of functions also influence participants' predictability judgments and found this to be the case: participants perceive the smoothness of functions as more important than the attached noise when making predictability judgments. Thus, predictions derived from learning curves match participants' behavior.

A shortcoming of this study is that it only asked participants' to judge about different functions' predictability without letting them generate actual predictions later on. Future studies could let participants make predictability judgments first and afterwards reward them based on how well they can actually predict the presented functions, taking into account their prior judgments.

Moreover, as the derived learning curves assumed a match between the student and the teacher kernel, future experiments could also assess the effect of mismatched learning on both predictability judgments and function learning performance. In particular, it would be interesting to assess what happens when participants learn in either rougher, smoother, or dynamically changing environments, before assessing them on the same level of smoothness. This could be used to develop and test different hierarchies of function learning models, such as warm starting (i.e., adjusting hyper-parameters over environments Poloczek et al., 2016), mismatched learning (i.e., assuming a fixed length-scale that is either too small or too large for the test set), or even "learning-to-learn"-algorithms (Chen et al., 2017) within environments of the same class over time.

Currently, the design of the predictability experiments also has been fully passive, whereas I could also ask participants to sample points actively. As learning curves for uncertainty-guided sampling with Gaussian Process regression exist (Seo et al., 2000), it would be straight forward to again compare theoretical predictions with human behavior in such paradigms.

Putting these concerns aside, I believe that studies on predictability, and associated therewith theoretically derived learning curves, will continue to produce elegant insights into human generalization.

9.1.2 Function learning and generalization is compositional

In the fifth chapter of this thesis, I defined and tested a compositional theory of human function learning and generalization. Within 8 experimental paradigms, from simple pattern completion to a working memory task, I have probed how a pre-defined compositional grammar of functional inductive biases explains human behavior. The results and analyses of all experiments converged clearly on the same interpretation, namely that participants do indeed have compositional inductive biases.

Other approaches of structure learning and generalization such as approximating the gradient of functional generalization explicitly or just assuming a smooth radial basis function kernel were unable to explain the richness of behavioral findings across all experiments. A compositional theory of human generalization provides a plausible and unique description of human generalization.

As is often the case, these results answer some but also generate many new questions. Most important is probably the question whether a compositional theory of human generalization offers a universal theory of generalization. Shepard's original proposal was to assess the gradient of generalization and later on researchers such as Carroll and Brehmer proposed rule-based accounts of function learning. One could argue that a compositional theory of generalization runs the risk of falling into the same complexity trap as more traditional rule-based accounts: as many possible base kernels exist (many more than the three proposed in Chapter 5), the main question becomes how participants choose and combine kernels in order to make sense and generalize from encountered data. Thus, my proposal of compositional inductive biases might move the quest for a universal rule of human cognition from an area that is a bit like physics (i.e. finding a particular rule or framework that predicts behavior across many set-ups) to one that is more akin to chemistry (i.e. trying to find the elements of generalization and the rules they adhere to). To this I say that proposing compositionality as a core principle of generalization is not opposed to assessing the gradient of functional generalization. Indeed, in order to truly understand human behavior across a wide range of domains, we will need both a top-down theory that tells us what the "human kernel" might look like, as well as a bottom-up model describing how participants arrive at a particular form of generalization over repeated experience. My results about compositionally in function learning imply that human generalization is compositional, even if my model is not exactly perfect.

The empirical finding of compositional inductive biases paves the way for multiple follow-up studies. An obvious one is defining and assessing a compositional theory of function exploration-exploitation which I will describe in more detail below. Another one concerns the distribution of compositional priors and how idiosyncratic and task-specific these are.

Lastly, I have only assessed compositionality by combining different kernels in order to make the search tree "wider". However, compositionality could also be assessed by making the tree deeper. This could be done by parsing the results computed from one function into another function, resulting in a Deep Gaussian Process (Damianou & Lawrence, 2013). Deep GPs can overcome some problems such as mismatch under stationarity assumptions, but also leads to further inferential complexities at run time.

Taken together, I believe that the results presented in Chapter 5 make a strong case that a successful quest for a theory of human generalization must adhere to the principle of compositionality.

9.1.3 Function learning guides search through complex tasks

Within both a spatially correlated and a multi-armed bandit task, I have found that participants' search behavior is generally best predicted by a combination of a Gaussian Process function learning model and the optimistic upper confidence bound sampling strategy. This not only means that participants' search in complex spaces is guided by a function mapping contexts to expected rewards and their attached uncertainties, but also that they solve the exploration-exploitation dilemma by balancing between sampling highly valued options while also valuing exploration in order to reduce uncertainty. As a Gaussian Process function learning model paired with the upper confidence bound sampling strategy is also the only computational algorithm in such scenarios with known performance guarantees, participants' behavior is surprisingly adaptive.

The experiments presented in Chapters 6-7 open the way for further cross-experimental comparisons. As recent neuroscientific evidence has connected both spatial and conceptual representations to a common neural substrate (Constantinescu et al., 2016), a model of generalization could also further unite contextual and spatial search problems. For example, the bivariate spatially correlated multi-armed bandit can be seen as a contextual multi-armed bandit problem with 121 arms and 2 continuous features. This makes the task comparable to the one used by Stojic et al. (2015). The results of such a comparison could test if the same model that predicts behavior in spatially correlated grids also describes behavior in contextual bandits, if the extracted parameters, in particular the estimated extent of generalization, are similar, and whether or not the neural representation of such tasks is analogous. The Gaussian Process function learning model could also be adjusted by extending the model's search horizon as has been proposed in the Bayesian optimization community before (González et al., 2016).

The results presented in Chapters 6-7 show that marrying powerful yet interpretable function learning techniques with methods commonly applied in the reinforcement learning literature can further advance our understanding of adaptive behavior and generalization in complex environments.

9.1.4 Under-generalization is sometimes beneficial

Chapter 8 has addressed the question if the empirically found phenomenon of participants' low λ-parameter estimates and the resulting under-generalization is always disadvantageous or if –indeed– there are situations in which being pessimistic about the underlying smoothness can be beneficial. The results of mismatched optimization simulations revealed that –even though expecting the right level of smoothness is usually best– there are situations in which it is easier to recover from mismatched priors if those have been defined pessimistically rather than optimistically. A quote commonly attributed to Albert Einstein says :"I'd rather be an optimist and a fool than a pessimist and right". As it turns out –at least in function exploration-exploitation scenarios– even Einstein can be wrong sometimes.

There are several remedies that could be applied to further overcome mismatch in Bayesian optimization. One is to integrate out the estimated hyper-parameters of the kernel by applying slice sampling (Snoek et al., 2012). Whether or not integrating out hyper-parameters can indeed overcome mismatch is an open empirical question.

Another way to overcome mismatch could be to utilize a compositional approach in which differently smooth kernels are combined additively and the weights are optimized using the log-evidence. When I tried this in a simulation for functions sampled from an Ornstein-Uhlenbeck kernel, a simple mixture of a radial basis function and an Ornstein-Uhlenbeck kernel was not able to produce lower regret than just using the Ornstein-Uhlenbeck kernel alone (see Appendix A8). This is most likely due to the fact that in optimization scenarios, algorithms only sample from very restricted points of the input space, i.e. points that promise to produce high outputs. Therefore, such sampling strategies do not provide enough information to optimize mixture weights.

By further assessing the effects of mismatched priors, both within computational simulations and psychological experiments, we may create new insights bridging psychology and machine learning research.

9.2 Open questions

There also remain open theoretical questions, the most important of which I want to address here.

9.2.1 Is this a process level model?

As Gaussian Process regression is a Bayesian algorithm that scales cubically with the number of training points, the question about whether or not my proposal equips us with a process level model seems adequate. Let me respond to this concern in two ways.

First, I do not see Gaussian Process regression as a process model *per se*, but rather as a computational level theory*. However, the borders between different levels of analysis become blurred relatively quickly. Consider Shepard's paradigmatic example of a universal law for which he defers to Newton's law of gravity. Even though Newton's law predicts the behavior of atoms, planets, and even galaxies, it is nonetheless not a process level model *per se*. It does not explicitly state how (i.e., by which underlying mechanism) different objects attract or repel each other. In fact, in his lectures on "The Character of Physical Law", Richard Feynman elaborates on three different ways by which Newton's law could be interpreted. Gravity could either refer to forces generated at a distance, to a potential at an object's center, or be defined by the shortest path between two objects. All of these interpretations are mathematically equivalent to Newton's law, but "psychologically feel very different" (Feynman & Wilczek, 1964). Regardless, Newton's law describes and predicts a wide range of physical phenomena. While I do not claim that the theory of Gaussian Process-based generalization forward is comparable to Newton's universal law, I think the mechanism of action is similar. By providing a computational framework to assess different forms of generalization, I hope to find empirical commonalities, be it compositionality or under-generalization, that build up a theory of generalization which predicts behavior across a wide range of paradigms.

*Based on Marr & Hildreth (1980)'s levels of analysis, the computational level (What does it do and why?), the algorithmic level (What is the representation and algorithm?), and the implementation level (What is the physical realization?).

There also exist several methods to be applied to transform Gaussian Process regression into a more plausible form of what is commonly referred to as a psychological process model. One such way is to speed up inference by approximation, for example via sparse sampling (Herbrich et al., 2003)[†]. This is similar to sampling-based process models that are currently gaining popularity as computationally rational solutions of resource-limited agents to complex problems (Gershman et al., 2015a; Dasgupta et al., 2017a; Bramley et al., 2017). Another approach could be to computationally embrace the linearity bias that is frequently observed in traditional function learning settings. As inference for linear regression is computationally cheap (it scales linearly with the number of data points), the linearity bias could be re-defined as a computationally less taxing "linearity heuristic". Participants' behavior might correspond well to models that try to jump to the limiting parametric form in order to save computations (Gramacy & Lee, 2008). Thus, there are multiple ways by which Gaussian Process regression could be made more "human-like".

9.2.2 Why not one model to rule them all?

Can the compositional model of generalization assessed in Chapter 5 be applied to the function exploration-exploitation scenarios assessed in Chapters 6-7?[‡]

One easy way to assess if compositions play a role when optimizing functions is to let participants explore and exploit the univariate functions created as the "matched set" in Chapter 5. If participants perform better when optimizing the compositional as compared to the non-compositional set, then this will provide first evidence that compositions also

[†] Similar savings can also be achieved by re-using previous computations (Dasgupta et al., 2017b).

[‡] Indeed, the chapters of this thesis are not sorted chronologically but rather topically. The function exploration-exploitation experiments were in fact conceptualized prior to defining and assessing the compositional theory of function learning. Thus, developing and assessing a theory of how compositional building blocks are combined when having to optimize unknown functions is a logical next step.

matter in exploration-exploitation situations. The task will then be to build a computational model of compositional optimization. One possible problem for this could be that the restricted set of observations that the sampling strategies produce might not be informative enough to discover sufficient structure. Thus, a compositional approach to optimization will likely need an explicit exploration bonus towards uncovering compositions or strong compositional priors. This could be accomplished by adding a strategic exploration mechanism that chooses between maximization, simple exploration, or exploration to discover compositional structure.

In the long run, I believe that combining compositional search with adequate sampling strategies will create both psychologically plausible and computationally powerful reinforcement learning models that can generalize to new tasks.

9.3 Generalizing generalization?

There are also a few more general directions in which the present research can be taken.

9.3.1 Further theory unification

A Gaussian Process-based theory of generalization offers many opportunities for further theory unification. For example, the associative learning models as specified in Chapters 6-7 can be reformulated as a GP regression model (Reece & Roberts, 2010) and hence implemented as a function learning model. In addition, when the length-scale of the radial basis function kernel approaches 0 ($\lambda \rightarrow 0$), the function learning model effectively assumes state-independence, as in traditional reinforcement learning models. Thus, there may be a continuum of reinforcement learning models, ranging from the traditional assumption of state independence to the opposite extreme of complete state interdependence.

Moreover, a Gaussian Process is also equivalent to a Bayesian Neural Network with infinite nodes (Neal, 1996), suggesting a further link to distributed function learning models (LeCun et al., 2015). Indeed, one explanation for the impressive performance of Deep Reinforcement Learning (Mnih et al., 2015) is that neural networks are specifically a powerful type of function approximator (Schölkopf, 2015).

As Bayesian reinforcement can also be seen as approximate dual control (Dayan & Sejnowski, 1996), approximating and propagating an underlying value function can also be modeled by Gaussian Process regression (Klenske & Hennig, 2015; Schulz et al., 2017a). Given that the successor representation of reinforcement learning (Dayan, 1993) has recently been shown to play a role in learning, memory and control (Gershman et al., 2012; Momennejad et al., 2017) and that it is possible to define a kernel using the inverse graph Laplacian (the successor representation under a random walk Kondor & Lafferty, 2002), assessing diffusion kernels with an exponential decay over graphs in reinforcement learning situations seems timely.

Sampling models of choice and decision making (Smith, 2000) could also be re-cast as a special case of Gaussian Process regression. For example, it is possible to think about Gaussian Processes applied to decision making as accumulating evidence at different time scales, i.e. not only over trials but within a trial. The generative model underlying the drift diffusion model is in fact a Gaussian Process, but defined in terms of intra-trial evidence dynamics. Thus, it is possible to define a single covariance function that incorporates both inter-trial and intra-trial dynamics. One implication of this is that any neural circuitry programmed to do Gaussian evidence accumulation can also be programmed to do learning across trials when configured with the appropriate recurrent connections.

9.3.2 Scaling up

Carroll already said that participants in Shepard's task do not only learn functions, but also "programs". I think making the compositional approach wider and deeper is only a first stepping stone towards thoroughly exploring the human ability to generalize. In order to really understand human generalization, I think what we need is a compositional approach that is closer to functional programming than to kernel compositions. This could be achieved by recasting human problem solving as a form of program induction (Dechter et al., 2013; Ellis et al., 2016), for example in paradigms that assess creative problem solving and hypothesis testing. In another line of research, I have shown how heuristic decision making strategies can emerge from compositional approximately Bayesian inference (Schulz et al., 2016c), which –I believe– could also be adapted to compositional functional programming more generally.

9.4 Conclusion

Situations in which generalization is required are ubiquitous. No matter if we perform inferences about patterns in the environment or decide where to dine tonight, there can be little doubt that generalization is a powerful ability aiding us to generate expectations in novel situations, a core ingredient of human intelligence. The controversial question is how we achieve this and what a unifying theory of human generalization might look like.

In this thesis, I have proposed to use Gaussian Process regression as method to investigate human generalization. The results of my experiments showed that generalization is compositional, that it guides decision making through complex tasks, and that the human tendency to under-generalize can sometimes be beneficial rather than harmful. Thus, the gradient of generalization is not the result of inferences someone makes, but a core ingredient of those inferences.

Of course, these have only been investigations into a few domains that were immediately accessible to the chosen function learning approach. The next steps are now to test how much these results themselves generalize both theoretically and empirically to other domains. I am confident that doing so systematically and rigorously will bring us closer to uncover universal rules of cognition— "possibly, behind the diverse behaviors of humans and animals, as behind the various motions of planets and stars, we may discern the operation of universal laws" (Shepard, 1987).

A
Appendix

A.1 Forecastibility

The Shannon entropy of a function f can be measured by the uncertainty of its spectrum S_f:

$$H(f) = -\int_{-} \quad S_f(\lambda) \ln S_f(\lambda) d\lambda \tag{A.1}$$

For any stationary function,

$$
\begin{aligned}
H(f) &\leq H(\text{white noise}) \\
&= -\int_{-\pi}^{\pi} \frac{1}{2\pi} \ln \frac{1}{2\pi} d\lambda \\
&= \ln 2\pi, \qquad \text{(A.2)}
\end{aligned}
$$

with equality if and only if f is white noise, the least forecastable signal with a uniform spectrum. The forecastibility measure $\Omega(f)$ is then defined as

$$\Omega(f) = 1 - \frac{H(f)}{\ln(2\pi)}. \qquad \text{(A.3)}$$

I follow the approach put forwad by Goerg (2013) and estimate $\Omega(f)$ by first estimating the spectrum S_f, normalizing it so that it integrates to 1, and then plugging it back into Eq. A.1. I use the periodogram as an unbiased estimator of S_f (see Goerg, 2013, for details).

A.2 Wavelet transform similarity measure

A discrete wavelet Haar Transform performs a scale-wise decomposition of the time series in such a way that most of the energy of the time series can be represented by a few coefficients. The main idea is to replace the original series by its wavelet approximation coefficients a in an appropriate scale, and then to measure the dissimilarity between the wavelet approximations. A detailed description of wavelet methods for time series analysis can be found in Percival & Walden (2006). I use the R-package `TSclust` (Montero & Vilar, 2014) to find the appropriate scale of the transform. I then measure the dissimilarity between two series x_1 and x_2 by the Euclidean distance at the selected scale:

$$d(x_1, x_2) = ||\mathrm{a}_1 - \mathrm{a}_2||. \tag{A.4}$$

A.3 Model details for spatially correlated mulit-armed bandit experiments

Table A.1 shows every model's parameter estimates and predictive accuracy for both Experiment 10 and 11 presented in Chapter 6.

Table A.1: Modeling results of Experiments 10 and 11.

Model	Experiment 10						Experiment 11					
	Model Comparison		Parameter Estimates				Model Comparison		Parameter Estimates			
	Predictive Accuracy	Participants Best Described	Length Scale (λ)	Exploration Bonus (β)	Error Variance ($\sqrt{\vartheta^2}$)	Softmax Temperature (τ)	Predictive Accuracy	Participants Best Described	Length Scale (λ)	Exploration Bonus (β)	Error Variance ($\sqrt{\vartheta^2}$)	Softmax Temperature (τ)
associative learning												
Upper Confidence Bound	0.09	1	–	2.54	1.01	0.03	0.1	0	–	0.97	1.96	0.02
Pure Exploitation	0.07	1	–	–	54.6	54.6	0.1	0	–	–	148.41	148.41
Pure Exploration	0.02	0	–	–	0.6	0.05	0.01	0	–	–	15.9	0.03
Expected Improvement	0.02	0	–	–	0.59	0.01	0.01	0	–	–	1.56	0.02
Probability of Improvement	0.02	0	–	–	0.42	0.01	0.1	0	–	–	0.01	0.11
Probability of Maximum Utility	0.00	0	–	–	0.62	7.63	0	0	–	–	0.54	0.01
associative learning*												
Upper Confidence Bound	0.21	1	–	20.12	0.01	28.12	0.36	12	–	44.08	0.07	15.79
Pure Exploitation	0.07	0	–	–	54.6	0.01	0.1	0	–	–	148.41	148.41
Pure Exploration	0.18	0	–	–	0.01	1.6	0.33	3	–	–	0.58	0.43
Expected Improvement	0.17	0	–	–	0.01	0.62	0.32	0	–	–	0.63	0.14
Probability of Improvement	0.15	0	–	–	0.06	0.18	0.32	0	–	–	0.01	0.09
Probability of Maximum Utility	.012	0	–	–	0.67	0.46	0.13	0	–	–	0.36	0.01
function learning												
Upper Confidence Bound	0.29	48	0.5	0.51	–	0.01	0.24	4	0.54	0.47	–	0.02
Pure Exploitation	0.16	6	1.94	–	–	0.15	0.16	0	1.55	–	–	0.11
Pure Exploration	0.02	0	0.11	–	–	0.03	0.01	0	0.17	–	–	0.55
Expected Improvement	0.15	9	0.56	–	–	0.01	0.23	0	0.67	–	–	0.05
Probability of Improvement	0.05	0	3.43	–	–	0.18	0.02	0	0.87	–	–	0.09
Probability of Maximum Utility	0.00	0	0.69	–	–	7.17	0.02	0	0.49	–	–	0.01
function learning*												
Upper Confidence Bound	0.23	10	0.96	0.54	–	0.16	0.38	60	0.76	0.49	–	0.09
Pure Exploitation	0.16	1	7.13	–	–	0.12	0.23	0	14.4	–	–	0.06
Pure Exploration	0.14	3	0.08	–	–	0.32	0.27	0	0.17	–	–	.19
Expected Improvement	0.09	1	0.71	–	–	0.11	0.23	1	0.67	–	–	0.05
Probability of Improvement	0.12	0	7.14	–	–	0.2	0.24	0	0.84	–	–	0.09
Probability of Maximum Utility	0.12	0	0.67	–	–	0.46	0.12	0	0.46	–	–	0.01
Win-Stay Lose-Sample	0.00	0	–	–	–	3.72	0.05	0	–	–	–	0.32
Win-Stay Lose-Sample*	0.05	0	–	–	–	0.73	0.26	0	–	–	–	0.22
Local Search	0.12	0	–	–	–	0.46	0.28	0	–	–	–	0.22

Note: Parameter estimates are the mean over all participants. There were 81 participants in Experiment 10 and 80 participants in Experiment 11.

A.4 Model recovery

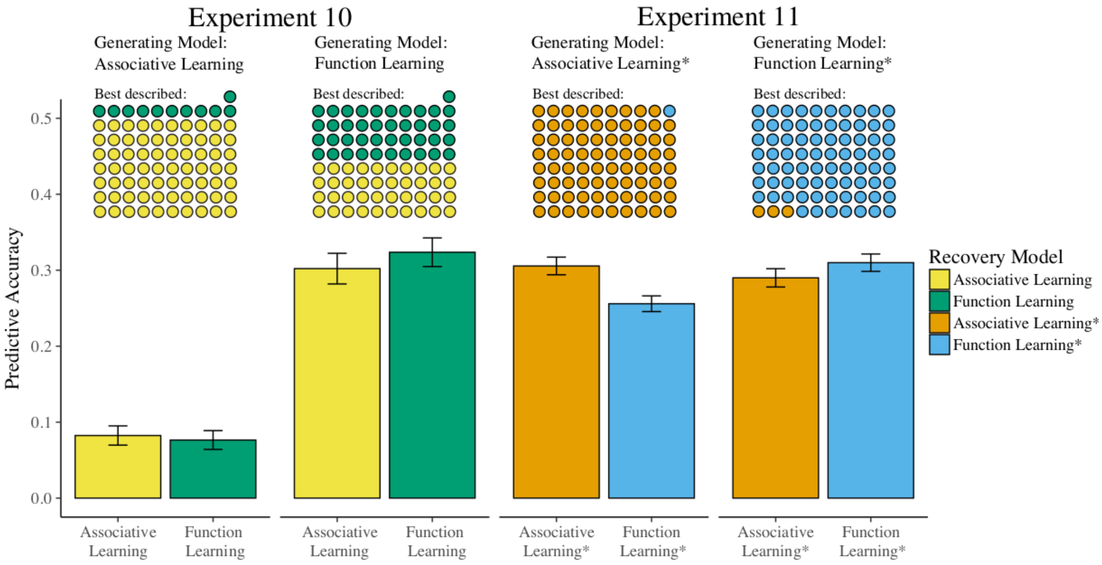


Figure A.1: Model recovery results, where data was generated by the specified generating model using individual participant parameter estimates. The recovery process used the same cross-validation method used in the model comparison. I report the predictive accuracy of each candidate recovery model (bars including standard error) and the number of simulated participants best described (icon array). For both generating and recovery models, I used UCB sampling. Table A.1 reports the median values of the cross-validated parameter estimates used to specify each generating model.

I present model recovery results that assess whether or not the predictive model comparison procedure of experiments 10-11 (but also, conceptually, of Experiments 12-14) allows to correctly identify the true underlying model. To assess this, I generated data based on each individual participant's parameter estimates. More specifically, for each participant and round, I use the cross-validated parameter estimates to specify a given model, and then generate new data resembling participants' data. I generate data using the associative learning and the function learning model for Experiment 10 and the associative learning* model and the function learning* model for Experiment 11. In all cases, I use the UCB sampling strategy in conjunction with the specified learning model. I then utilize the same cross-validation method as before in order to determine if it is possible to successfully identify which model has generated the underlying data. Figure A.1 shows the cross-validated predictive performance (bars) for the simulated data, along with the number of simulated participants best described (icon array).

In the simulation for Experiment 10, the predictive model comparison procedure shows that the associative learning model is a better predictor for data generated from the same underlying model, whereas the function learning model is only marginally better at predicting data generated from the same underlying model. This suggests that the main model comparison results are robust to Type I errors, and provides evidence that the better predictive accuracy of the function learning model on participant data is unlikely due to overfitting.

When the associative learning model generates data using participant parameter estimates, the same associative learning model achieves an average predictive accuracy of $R^2 = .1$ and describes 70 out 81 simulated participants best. On the same generated data, the function learning model achieves an average predictive accuracy of $R^2 = .09$ and only describes 11 out of 81 simulated participants best.

When the function learning model has generated the underlying data, the same function learning model achieves a predictive accuracy of $R^2 = .4$ and describes 41 out of 81 simulated participants best, whereas the associative learning model achieves a predictive accuracy of $R^2 = .39$ and describes 40 participants best. This makes the finding of the function learning as the best predictive model even stronger as –technically– the associative learning model could mimic parts of the function learning behavior.

In the simulations for Experiment 11, I use the localized version of each type of learning model for both generation and recovery, since in both cases, localization improved predictive accuracy of human participants (Table A.1). Here, I find very clear recoverability in all cases, with the recovering model best predicting the vast majority of simulated participants when it is also the generating model (Fig. A.1).

When the associative learning* model generated the data, the associative learning* model achieves a predictive accuracy of $R^2 = .31$ and predicts 79 out of 80 simulated participants best, whereas the function learning* model predicts only a lone simulated participant better, with an average predictive accuracy of $R^2 = .26$.

If the function learning* model generated the underlying data, the same function learning* model achieves a predictive accuracy of $R^2 = .34$ and describes 77 out of 80 simulated participants best, whereas the associative learning* model only describes 3 out of 80 simulated participants better, with a average predictive accuracy of $R^2 = .32$.

In all of the these simulations, the model that has generated the underlying data is also the best performing model, as assessed by its predictive accuracy and the number of simulated participants predicted best. Thus, I can confidently say that the cross-validation procedure distinguishes between the two assessed model classes. Moreover, in the cases where the function learning or function learning* model has generated the underlying data, the predictive accuracy of the same model is not perfect (i.e., $R^2 = 1$), but rather closer to the predictive accuracy I have found on the actual participant data (Table A.1).

A.5 Parameter Recovery

Another important question is whether or not the reported parameter estimates of the models are reliable and recoverable. I address this question by assessing the recoverability of the three parameters of the function learning model, the length-scale λ, the exploration factor β, and the temperature parameter τ of the softmax choice rule. I use the results from the model recovery simulation described above, and correlate the empirically estimated

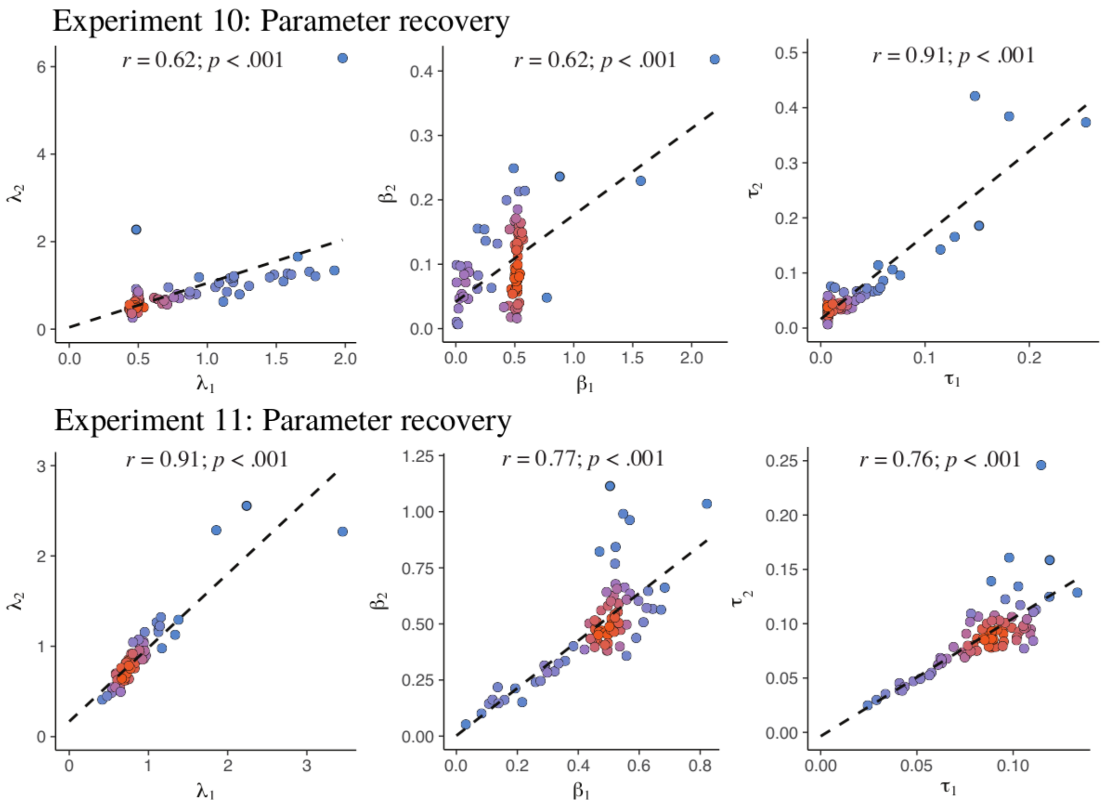


Figure A.2: Parameter recovery results. The generating parameter estimate is on the x-axis and the recovered parameter estimate is on the y-axis. The generating parameter estimates are from the cross-validated participant parameter estimates, which were used to simulate data (see Model Recovery). Recovered parameter estimates are the result of the cross-validated model comparison (see Model Comparison) on the simulated data. While the cross-validation procedure yielded k-estimates per participant, one for each round ($k_{exp1} = 16$; $k_{exp2} = 8$), I show the median estimate per (simulated) participant. The dashed line shows a linear regression on the data, while the Pearson correlation and p-value is shown above the plot. For readability, colors represent the bivariate kernel density estimate, with red indicating higher density.

parameters used to generate data (i.e., the estimates based on participants' data), with the parameter estimates of the recovering model (i.e., the MLE from the cross-validation procedure on the simulated data). I assess whether the recovered parameter estimates are similar to the parameters that were used to generated the underlying data. I present parameter recovery results for the function learning model for Experiment 10 and the function learning* model for Experiment 11, both using the UCB sampling strategy. I report the results in Figure A.2, with the generating parameter estimate on the x-axis and the recovered parameter estimate on the y-axis.

For Experiment 10, the correlation between the generating and the recovered length-scale λ is $r = .62, p < .001$, the correlation between the generating and the recovered exploration factor β is $r = 0.62, p < .001$, and the correlation between the generating and the recovered softmax temperature parameter τ is $r = 0.91, p < .001$. For Experiment 11, the correlation between the generating and the recovered λ is $r = 0.91, p < .001$, for β the correlation is $r = 0.77, p < .001$, and for τ the correlation is $r = 0.76, p < .001$.

These results show that the correlation between the generating and the recovered parameters is incredibly high for both experiments and for all parameters. Thus, I have found strong evidence to support the claim that the reported parameter estimates of the Gaussian Process-based function learning model (Table A.1) are recoverable, reliable, and therefore interpretable. Importantly, I find that estimates for β (exploration bonus) and τ (softmax temperature) are indeed recoverable, providing evidence for the existence of a *directed* exploration bonus, as a separate phenomena from random, undirected exploration (Daw et al., 2006).

A.6 Model details for contextual mulit-armed bandit experiments

Table A.2 shows the full parameter estimates and predictive performance for all models applied in the CMAB with binary features assessed in Experiment 12.

Table A.2: Model results of CMAB with binary features

Model	Predictive R^2	#best	λ	β	σ	τ
GP-RBF						
Upper Confidence Bound	0.38	3	7.82	2.13	2.89	1.33
Expected Improvement	0.45	11	7.46	-	4.32	3.98
Probability of Improvement	0.46	15	7.39	-	3.37	8.43
Probability of Maximum Utility	0.40	3	8.40	-	3.62	4.21
GP-Linear						
Upper Confidence Bound	0.13	5	-	1.31	-	4.17
Expected Improvement	0.14	3	-	-	-	1.39
Probability of Improvement	0.12	1	-	-	-	1.67
Probability of Maximum Utility	0.12	2	-	-	-	1.38
Kalman Filter						
Upper Confidence Bound	0.07	0	-	2.06	3.21	0.13
Expected Improvement	0.10	0	-	-	3.60	3.98
Probability of Improvement	0.09	0	-	-	9.43	0.72
Probability of Maximum Utility	0.16	1	-	-	1.17	2.69
Mean Tracker						
Upper Confidence Bound	0.08	0	-	2.12	4.44	0.34
Expected Improvement	0.09	0	-	-	6.37	8.32
Probability of Improvement	0.09	0	-	-	3.45	0.75
Probability of Maximum Utility	0.16	3	-	-	6.45	2.67

Table A.3 shows the full parameter estimates and predictive performance for all models applied in the CMAB with continuous-linear features assessed in Experiment 13.

Table A.3: Model results of linear CMAB with continous features

Model	Predictive R^2	#best	λ	β	σ	τ
GP-RBF						
Upper Confidence Bound	0.22	16	8.44	1.84	5.34	2.39
Expected Improvement	0.19	4	8.43	-	5.99	1.11
Probability of Improvement	0.19	4	7.39	-	5.38	6.07
Probability of Maximum Utility	0.19	1	7.90	-	4.77	0.54
GP-Linear						
Upper Confidence Bound	0.18	9	-	0.95	-	3.22
Expected Improvement	0.08	3	-	-	-	0.64
Probability of Improvement	0.07	1	-	-	-	0.43
Probability of Maximum Utility	0.06	2	-	-	-	0.74
Kalman Filter						
Upper Confidence Bound	0.09	2	-	2.72	0.13	0.28
Expected Improvement	0.08	3	-	-	3.60	0.61
Probability of Improvement	0.07	1	-	-	9.43	3.47
Probability of Maximum Utility	0.15	8	-	-	1.17	6.04
Mean Tracker						
Upper Confidence Bound	0.09	0	-	2.94	5.69	0.25
Expected Improvement	0.08	3	-	-	6.37	9.19
Probability of Improvement	0.08	0	-	-	6.45	0.66
Probability of Maximum Utility	0.15	6	-	-	6.69	3.51

Table A.4 shows the full parameter estimates and predictive performance for all models applied in the CMAB with continuous non-linear features assessed in Experiment 14.

Table A.4: Model results of non-linear CMAB with continous features

Model	Predictive R^2	#best	λ	β	σ	τ
GP-RBF						
Upper Confidence Bound	0.23	9	8.94	1.59	6.09	8.86
Expected Improvement	0.23	6	7.49	-	7.17	15.7
Probability of Improvement	0.24	2	7.73	-	6.23	5.19
Probability of Maximum Utility	0.23	8	7.09	-	6.33	5.44
GP-Linear						
Upper Confidence Bound	0.14	4	-	1.08	-	3.88
Expected Improvement	0.08	3	-	-	-	0.93
Probability of Improvement	0.10	2	-	-	-	0.81
Probability of Maximum Utility	0.09	3	-	-	-	0.93
Kalman Filter						
Upper Confidence Bound	0.14	2	-	2.19	0.14	0.41
Expected Improvement	0.12	0	-	-	3.79	5.97
Probability of Improvement	0.10	1	-	-	9.11	1.75
Probability of Maximum Utility	0.17	5	-	-	0.99	2.89
Mean Tracker						
Upper Confidence Bound	0.14	1	-	2.36	6.41	0.41
Expected Improvement	0.12	3	-	-	8.43	10.39
Probability of Improvement	0.10	0	-	-	6.45	1.97
Probability of Maximum Utility	0.18	4	-	-	6.38	2.97

A.7 Sublinear regret of GP-UCB

The regret bounds (up to polylog factors) for the upper confidence bound sampling strategy were derived by (Srinivas et al., 2012). Let T be the time horizon, d the dimension of the data, and ν the Matérn parameter, then the regret bound for the linear kernel is

$$R_T = d\sqrt{T}, \tag{A.5}$$

the regret bound for the radial basis function kernel is

$$R_T = \sqrt{T(\log T)^{d+1}}, \tag{A.6}$$

and the regret bound for the Matérn kernel is

$$R_T = T\frac{\nu + d(d+1)}{2\nu + d(d+1)} \tag{A.7}$$

A.8 Regret of a mixture kernel

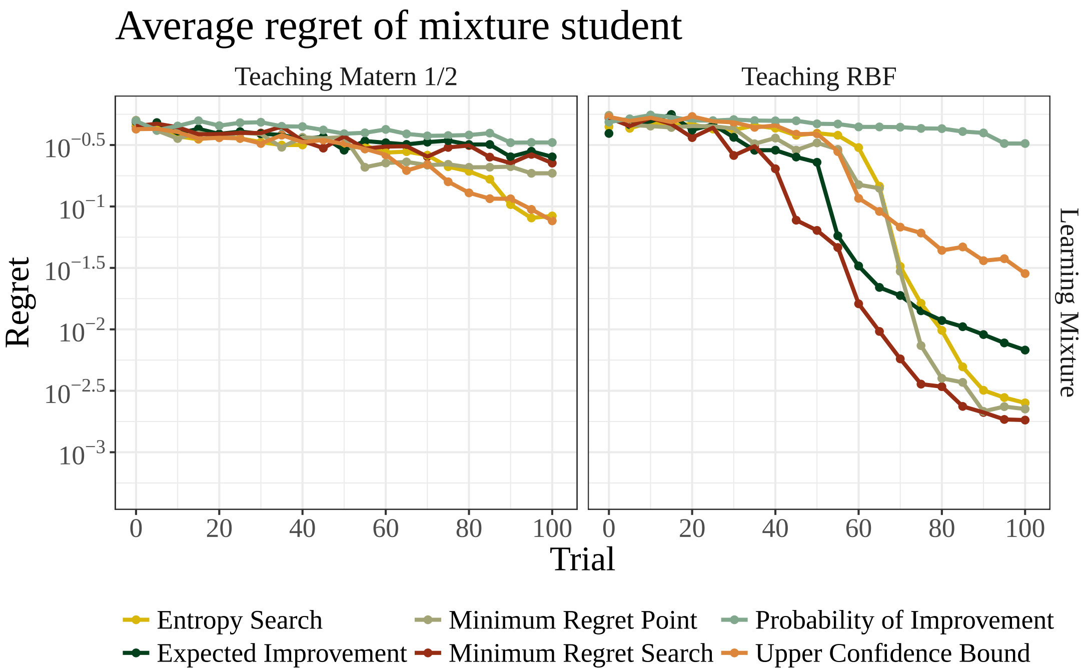


Figure A.3: Median regret over time for different sampling strategies using a mixture kernel student.

Figure A.3 shows the median regret over 100 runs for a mixture kernel, with a Matérn

1/2 and radial basis function kernel component, optimizing either a RBF or a Matérn 1/2 target function. Whereas the mixture kernel reaches a regret in the region of 10^{-1} when optimizing a Matérn 1/2 kernel, the best regret in the matched setting was $10^{-1.5}$. Whereas the mixture kernel reaches a regret in the region of $10^{-2.5}$, the best regret in the matched setting was in the region of 10^{-4}. Thus, the mixture regret does not recover the performance of its matching component.

References

Andreassen, P. B. & Kraus, S. J. (1990). Judgmental extrapolation and the salience of change. *Journal of Forecasting*, 9(4), 347–372.

Archer, K. J. & Kimes, R. V. (2008). Empirical characterization of random forest variable importance measures. *Computational Statistics & Data Analysis*, 52(4), 2249–2260.

Auer, P., Cesa-Bianchi, N., & Fischer, P. (2002). Finite-time analysis of the multiarmed bandit problem. *Machine Learning*, 47(2), 235–256.

Barker, A. A. (1965). Monte carlo calculations of the radial distribution functions for a proton electron plasma. *Australian Journal of Physics*, 18(2), 119–134.

Bechara, A., Damasio, H., Tranel, D., & Damasio, A. R. (1997). Deciding advantageously before knowing the advantageous strategy. *Science*, 275(5304), 1293–1295.

Bergstra, J. S., Bardenet, R., Bengio, Y., & Kégl, B. (2011). Algorithms for hyper-parameter optimization. In *Advances in Neural Information Processing Systems* (pp. 2546–2554).

Berkenkamp, F., Schoellig, A. P., & Krause, A. (2016). Safe controller optimization for quadrotors with Gaussian processes. In *Robotics and Automation (ICRA), 2016 IEEE International Conference on* (pp. 491–496).: IEEE.

Berry, D. C. & Broadbent, D. E. (1984). On the relationship between task performance and associated verbalizable knowledge. *The Quarterly Journal of Experimental Psychology*, 36(2), 209–231.

Biederman, I. (1987). Recognition-by-components: A theory of human image understanding. *Psychological Review*, 94(2), 115.

Bolger, F. & Harvey, N. (1998). Heuristics and biases in judgmental forecasting. John Wiley & Son.

Bonawitz, E., Denison, S., Gopnik, A., & Griffiths, T. L. (2014). Win-stay, lose-sample: A simple sequential algorithm for approximating bayesian inference. *Cognitive Psychology*, 74, 35–65.

Borji, A. & Itti, L. (2013). Bayesian optimization explains human active search. In *Advances in Neural Information Processing Systems* (pp. 55–63).

Bott, L. & Heit, E. (2004). Nonmonotonic extrapolation in function learning. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 30(1), 38.

Boulesteix, A.-L., Janitza, S., Kruppa, J., & König, I. R. (2012). Overview of random forest methodology and practical guidance with emphasis on computational biology and bioinformatics. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 2(6), 493–507.

Bouton, M. E. & King, D. A. (1983). Contextual control of the extinction of conditioned fear: tests for the associative value of the context. *Journal of Experimental Psychology: Animal Behavior Processes*, 9(3), 248.

Brady, T. F., Konkle, T., & Alvarez, G. A. (2011). A review of visual memory capacity: Beyond individual items and toward structured representations. *Journal of Vision*, 11(5), 4–4.

Brady, T. F. & Tenenbaum, J. B. (2013). A probabilistic model of visual working memory: Incorporating higher order regularities into working memory capacity estimates. *Psychological Review*, 120(1), 85.

Bramley, N. R., Dayan, P., Griffiths, T. L., & Lagnado, D. A. (2017). Formalizing neurath's ship: Approximate algorithms for online causal learning. *Psychological Review*, 124(3), 301.

Brehmer, B. (1971). Subjects' ability to use functional rules. *Psychonomic Science*, 24(6), 259–260.

Brehmer, B. (1976). Learning complex rules in probabilistic inference tasks. *Scandinavian Journal of Psychology*, 17(1), 309–312.

Brehmer, B., Kuylenstierna, J., & Liljergren, J.-E. (1974). Effects of function form and cue validity on the subjects' hypotheses in probabilistic inference tasks. *Organizational Behavior and Human Performance*, 11(3), 338–354.

Brochu, E., Cora, V. M., & De Freitas, N. (2010). A tutorial on Bayesian optimization of expensive cost functions, with application to active user modeling and hierarchical reinforcement learning. *arXiv preprint arXiv:1012.2599*.

Bröder, A. & Schiffer, S. (2003). Bayesian strategy assessment in multi-attribute decision making. *Journal of Behavioral Decision Making*, 16(3), 193–213.

Bruner, J. S. (1966). *Toward a theory of instruction*, volume 59. Harvard University Press.

Bruner, J. S., Goodnow, J. J., & Austin, G. A. (1956). A study of thinking. *Science Editions*.

Bubeck, S., Cesa-Bianchi, N., et al. (2012). Regret analysis of stochastic and nonstochastic multi-armed bandit problems. *Foundations and Trends in Machine Learning*, 5(1), 1–122.

Buffart, H., Leeuwenberg, E., & Restle, F. (1981). Coding theory of visual pattern completion. *Journal of Experimental Psychology: Human Perception and Performance*, 7(2), 241.

Busemeyer, J. R., Byun, E., DeLosh, E. L., & McDaniel, M. A. (1997). Learning functional relations based on experience with input-output pairs by humans and artificial neural networks. In K. Lamberts & D. Shanks (Eds.), *Concepts and Categories* (pp. 405–437). Cambridge: MIT Press.

Byun, E. (1995). Interaction between prior knowledge and type of nonlinear relationship on function learning.

Carroll, J. D. (1963). Functional learning: The learning of continuous functional mappings relating stimulus and response continua. *ETS Research Report Series*, 1963(2).

Carroll, J. D. & De Soete, G. (1991). Toward a new paradigm for the study of multiattribute choice behavior: Spatial and discrete modeling of pairwise preferences. *American Psychologist*, 46(4), 342.

Chase, W. G. & Simon, H. A. (1973). Perception in chess. *Cognitive Psychology*, 4(1), 55–81.

Chater, N. & Oaksford, M. (2008). *The probabilistic mind: Prospects for Bayesian cognitive science*. OUP Oxford.

Chater, N., Tenenbaum, J. B., & Yuille, A. (2006). Probabilistic models of cognition: Conceptual foundations. *Trends in cognitive sciences*, 10(7), 287–291.

Chater, N. & Vitányi, P. (2003). Simplicity: A unifying principle in cognitive science? *Trends in Cognitive Sciences*, 7(1), 19–22.

Chen, Y., Hoffman, M. W., Colmenarejo, S. G., Denil, M., Lillicrap, T. P., Botvinick, M., & de Freitas, N. (2017). Learning to learn without gradient descent by gradient descent.

Chomsky, N. (1957). Syntactic structures. the hague: Mouton.. 1965. aspects of the theory of syntax. *Cambridge, Mass.: MIT Press.(1981) Lectures on Government and Binding*, 6, 1–52.

Collobert, R., Weston, J., Bottou, L., Karlen, M., Kavukcuoglu, K., & Kuksa, P. (2011). Natural language processing (almost) from scratch. *Journal of Machine Learning Research*, 12(Aug), 2493–2537.

Constantinescu, A. O., O'Reilly, J. X., & Behrens, T. E. (2016). Organizing conceptual knowledge in humans with a gridlike code. *Science*, 352(6292), 1464–1468.

Cornford, D., Nabney, I. T., & Williams, C. K. (2002). Modelling frontal discontinuities in wind fields. *Journal of Nonparametric Statistics*, 14(1-2), 43–58.

Cox, G., Kachergis, G., & Shiffrin, R. (2012). Gaussian process regression for trajectory analysis. In *Proceedings of the Cognitive Science Society*.

Crupi, V. & Tentori, K. (2014). State of the field: Measuring information and confirmation. *Studies in History and Philosophy of Science Part A*, 47, 81–90.

Damianou, A. & Lawrence, N. (2013). Deep gaussian processes. In *Artificial Intelligence and Statistics* (pp. 207–215).

Dasgupta, I., Schulz, E., & Gershman, S. J. (2017a). Where do hypotheses come from? *Cognitive Psychology*, 96, 1–25.

Dasgupta, I., Schulz, E., Goodman, N. D., & Gershman, S. J. (2017b). Amortized hypothesis generation. In *Proceedings of the 39th Annual Conference of the Cognitive Science Society*.

Daw, N. D., O'doherty, J. P., Dayan, P., Seymour, B., & Dolan, R. J. (2006). Cortical substrates for exploratory decisions in humans. *Nature*, 441(7095), 876.

Dayan, P. (1993). Improving generalization for temporal difference learning: The successor representation. *Neural Computation*, 5(4), 613–624.

Dayan, P. & Sejnowski, T. J. (1996). Exploration bonuses and dual control. *Machine Learning*, 25(1), 5–22.

Dechter, E., Malmaud, J., Adams, R. P., & Tenenbaum, J. B. (2013). Bootstrap learning via modular concept discovery. In *IJCAI* (pp. 1302–1309).

DeLosh, E. L., Busemeyer, J. R., & McDaniel, M. A. (1997). Extrapolation: The sine qua non for abstraction in function learning. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 23(4), 968.

Desautels, T. A. (2014). *Spinal cord injury therapy through active learning*. PhD thesis, California Institute of Technology.

Duvenaud, D., Lloyd, J. R., Grosse, R., Tenenbaum, J. B., & Ghahramani, Z. (2013). Structure discovery in nonparametric regression through compositional kernel search. In *International Conference on Machine Learning*.

Ebbinghaus, H. (1885). Ueber das gedächtnis.

Eggleton, I. R. (1982). Intuitive time-series extrapolation. *Journal of Accounting Research*, (pp. 68–102).

Ellis, K., Solar-Lezama, A., & Tenenbaum, J. (2016). Sampling for Bayesian Program Learning. In *Advances in Neural Information Processing Systems* (pp. 1297–1305).

Feynman, R. & Wilczek, F. (1964). *The character of physical law*. MIT press.

Field, D. J., Hayes, A., & Hess, R. F. (1993). Contour integration by the human visual system: Evidence for a local "association field". *Vision Research*, 33(2), 173–193.

Fodor, J. (2001). Language, thought and compositionality. *Royal Institute of Philosophy Supplements*, 48, 227–242.

Fodor, J. A. (1975). *The language of thought*, volume 5. Harvard University Press.

Fodor, J. A. & Pylyshyn, Z. W. (1988). Connectionism and cognitive architecture: A critical analysis. *Cognition*, 28(1), 3–71.

Forrest, A. R. (1972). Interactive interpolation and approximation by bézier polynomials. *The Computer Journal*, 15(1), 71–79.

Friston, K. (2010). The free-energy principle: A unified brain theory? *Nature Reviews Neuroscience*, 11(2), 127–138.

Gershman, S. J. (2015). A unifying probabilistic view of associative learning. *PLoS Computational Biology*, 11(11), e1004567.

Gershman, S. J., Blei, D. M., & Niv, Y. (2010). Context, learning, and extinction. *Psychological Review*, 117(1), 197.

Gershman, S. J. & Daw, N. D. (2017). Reinforcement learning and episodic memory in humans and animals: An integrative framework. *Annual Review of Psychology*, 68, 101–128.

Gershman, S. J., Horvitz, E. J., & Tenenbaum, J. B. (2015a). Computational rationality: A converging paradigm for intelligence in brains, minds, and machines. *Science*, 349(6245), 273–278.

Gershman, S. J., Malmaud, J., & Tenenbaum, J. B. (2017). Structured representations of utility in combinatorial domains. *Decision*, 4(2), 67.

Gershman, S. J., Moore, C. D., Todd, M. T., Norman, K. A., & Sederberg, P. B. (2012). The successor representation and temporal context. *Neural Computation*, 24(6), 1553–1568.

Gershman, S. J. & Niv, Y. (2010). Learning latent structure: Carving nature at its joints. *Current Opinion in Neurobiology*, 20, 251–256.

Gershman, S. J. & Niv, Y. (2015). Novelty and inductive generalization in human reinforcement learning. *Topics in Cognitive Science*, 7(3), 391–415.

Gershman, S. J., Norman, K. A., & Niv, Y. (2015b). Discovering latent causes in reinforcement learning. *Current Opinion in Behavioral Sciences*, 5, 43–50.

Ghahramani, Z. (2015). Probabilistic machine learning and artificial intelligence. *Nature*, 521(7553), 452–459.

Gigerenzer, G. (1991). From tools to theories: A heuristic of discovery in cognitive psychology. *Psychological Review*, 98(2), 254.

Gluck, M. A., Shohamy, D., & Myers, C. (2002). How do people solve the "weather prediction" task? individual variability in strategies for probabilistic category learning. *Learning & Memory*, 9(6), 408–418.

Gobet, F. (1998). Expert memory: A comparison of four theories. *Cognition*, 66(2), 115–152.

Goerg, G. (2013). Forecastable component analysis. In *Proceedings of the 30th International Conference on Machine Learning (ICML-13)* (pp. 64–72).

González, J., Osborne, M. A., & Lawrence, N. D. (2016). GLASSES: Relieving the myopia of Bayesian optimisation. In *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics* (pp. 790–799).

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., & Bengio, Y. (2014). Generative adversarial nets. In *Advances in Neural Information Processing Systems* (pp. 2672–2680).

Goodman, N. D., Tenenbaum, J. B., Feldman, J., & Griffiths, T. L. (2008a). A rational analysis of rule-based concept learning. *Cognitive Science*, 32(1), 108–154.

Goodman, N. D., Tenenbaum, J. B., Griffiths, T. L., & Feldman, J. (2008b). Compositionality in rational analysis: Grammar-based induction for concept learning. *The probabilistic mind: Prospects for Bayesian cognitive science*.

Goodwin, P. & Wright, G. (1993). Improving judgmental time series forecasting: A review of the guidance provided by research. *International Journal of Forecasting*, 9(2), 147–161.

Gotovos, A., Casati, N., Hitz, G., & Krause, A. (2013). Active learning for level set estimation. In *International Joint Conference on Artificial Intelligence (IJCAI)* (pp. 1344–1350).

Gramacy, R. B. & Lee, H. K. (2008). Gaussian processes and limiting linear models. *Computational Statistics & Data Analysis*, 53(1), 123–136.

Griffiths, T. L. & Kalish, M. L. (2007). Language evolution by iterated learning with bayesian agents. *Cognitive Science*, 31(3), 441–480.

Griffiths, T. L., Lucas, C., Williams, J., & Kalish, M. L. (2009). Modeling human function learning with Gaussian processes. In *Advances in Neural Information Processing Systems* (pp. 553–560).

Grosse, R., Salakhutdinov, R., Freeman, W., & Tenenbaum, J. (2012). Exploiting compositionality to explore a large space of model structures. In *Uncertainty in Artificial Intelligence*.

Guez, A., Silver, D., & Dayan, P. (2013). Scalable and efficient Bayes-adaptive reinforcement learning based on monte-carlo tree search. *Journal of Artificial Intelligence Research*, 48, 841–883.

Gureckis, T. M. & Markant, D. B. (2012). Self-directed learning: A cognitive and computational perspective. *Perspectives on Psychological Science*, 7(5), 464–481.

Guttman, N. & Kalish, H. I. (1956). Discriminability and stimulus generalization. *Journal of Experimental Psychology*, 51(1), 79.

Harvey, N. & Bolger, F. (1996). Graphs versus tables: Effects of data presentation format on judgemental forecasting. *International Journal of Forecasting*, 12(1), 119–137.

Harvey Teresa Ewart Robert West, N. (1997). Effects of data noise on statistical judgement. *Thinking & Reasoning*, 3(2), 111–132.

Hennig, P. & Hauberg, S. (2014). Probabilistic solutions to differential equations and their application to Riemannian statistics. In *Artificial Intelligence and Statistics* (pp. 347–355).

Hennig, P. & Schuler, C. J. (2012). Entropy search for information-efficient global optimization. *Journal of Machine Learning Research*, 13(Jun), 1809–1837.

Herbrich, R., Lawrence, N. D., & Seeger, M. (2003). Fast sparse Gaussian process methods: The informative vector machine. In *Advances in Neural Information Processing Systems* (pp. 625–632).

Hernández-Lobato, J. M., Hoffman, M. W., & Ghahramani, Z. (2014). Predictive entropy search for efficient global optimization of black-box functions. In *Advances in Neural Information Processing Systems* (pp. 918–926).

Hoerl, A. E. & Kennard, R. W. (1970). Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, 12(1), 55–67.

Holyoak, K. J. (1985). The pragmatics of analogical transfer. *Psychology of learning and motivation*, 19, 59–87.

Honig, W. K. & Urcuioli, P. J. (1981). The legacy of Guttman and Kalish (1956): 25 years of research on stimulus generalization. *Journal of the Experimental Analysis of Behavior*, 36(3), 405–445.

Hoyer, P. O., Janzing, D., Mooij, J. M., Peters, J., & Schölkopf, B. (2009). Nonlinear causal discovery with additive noise models. In *Advances in Neural Information Processing Systems* (pp. 689–696).

Hull, C. L. (1935). The conflicting psychologies of learning—a way out. *Psychological Review*, 42(6), 491.

Huys, Q. J., Lally, N., Faulkner, P., Eshel, N., Seifritz, E., Gershman, S. J., Dayan, P., & Roiser, J. P. (2015). Interplay of approximate planning strategies. *Proceedings of the National Academy of Sciences*, 112(10), 3098–3103.

Hwang, Y., Tong, A., & Choi, J. (2016). Automatic construction of nonparametric relational regression models for multiple time series. In *Proceedings of the 33rd International Conference on Machine Learning*.

Hyndman, R. J., Wang, E., & Laptev, N. (2015). Large-scale unusual time series detection. In *Data Mining Workshop (ICDMW), 2015 IEEE International Conference on* (pp. 1616–1619).: IEEE.

Jäkel, F., Schölkopf, B., & Wichmann, F. A. (2007). A tutorial on kernel methods for categorization. *Journal of Mathematical Psychology*, 51(6), 343–358.

Kac, M. & Siegert, A. (1947). An explicit representation of a stationary Gaussian process. *The Annals of Mathematical Statistics*, (pp. 438–442).

Kalish, M. L. (2013). Learning and extrapolating a periodic function. *Memory & Cognition*, 41(6), 886.

Kalish, M. L., Griffiths, T. L., & Lewandowsky, S. (2007). Iterated learning: Intergenerational knowledge transmission reveals inductive biases. *Psychonomic Bulletin & Review*, 14(2), 288–294.

Kalish, M. L., Lewandowsky, S., & Kruschke, J. K. (2004). Population of linear experts: Knowledge partitioning and function learning. *Psychological Review*, 111(4), 1072.

Kandel, E. R. (2007). *In search of memory: The emergence of a new science of mind*. WW Norton & Company.

Kanizsa, G. (1979). *Organization in vision: Essays on Gestalt perception*. Praeger Publishers.

Kanizsa, G. et al. (1976). Subjective contours. *Scientific American*, 234(4), 48–52.

Kemp, C. (2012). Exploring the conceptual universe. *Psychological Review*, 119(4), 685.

Kemp, C. & Tenenbaum, J. B. (2009). Structured statistical models of inductive reasoning. *Psychological Review*, 116(1), 20.

Keren, G. (1983). Cultural differences in the misperception of exponential growth. *Attention, Perception, & Psychophysics*, 34(3), 289–293.

Klenske, E. D. & Hennig, P. (2015). Dual control for approximate Bayesian reinforcement learning. *arXiv preprint arXiv:1510.03591*.

Koh, K. & Meyer, D. E. (1991). Function learning: Induction of continuous stimulus-response relations. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 17(5), 811.

Kondor, R. I. & Lafferty, J. (2002). Diffusion kernels on graphs and other discrete input spaces. In *International Conference on Machine Learning*, volume 2 (pp. 315–322).

Krause, A. & Guestrin, C. (2005). Optimal nonmyopic value of information in graphical models: Efficient algorithms and theoretical limits.

Krebs, J. R. (1980). Optimal foraging, predation risk and territory defence. *Ardea*, 68, 83–90.

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems* (pp. 1097–1105).

Kruschke, J. K. (2008). Bayesian approaches to associative learning: From passive to active learning. *Learning & Behavior*, 36(3), 210–226.

Kruschke, J. K. & Johansen, M. K. (1999). A model of probabilistic category learning. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 25, 1083–1119.

Kushner, H. J. (1964). A new method of locating the maximum point of an arbitrary multipeak curve in the presence of noise. *Journal of Basic Engineering*, 86(1), 97–106.

Kwantes, P. J. & Neal, A. (2006). Why people underestimate y when extrapolating in linear functions. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 32(5), 1019.

Lake, B. M., Salakhutdinov, R., & Tenenbaum, J. B. (2015). Human-level concept learning through probabilistic program induction. *Science*, 350(6266), 1332–1338.

Lake, B. M., Salakhutdinov, R. R., & Tenenbaum, J. (2013). One-shot learning by inverting a compositional causal process. In *Advances in Neural Information Processing Systems* (pp. 2526–2534).

Lake, B. M., Ullman, T. D., Tenenbaum, J. B., & Gershman, S. J. (2016). Building machines that learn and think like people. *Behavioral and Brain Sciences*, (pp. 1–101).

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, 521(7553), 436–444.

Lee, S. W., Shimojo, S., & O'Doherty, J. P. (2014). Neural computations underlying arbitration between model-based and model-free learning. *Neuron*, 81(3), 687–699.

Lee, T. S. & Yuille, A. L. (2006). Efficient coding of visual scenes by grouping and segmentation. *Bayesian brain: Probabilistic approaches to neural coding*, (pp. 141–185).

Lejarraga, T. & Gonzalez, C. (2011). Effects of feedback and complexity on repeated decisions from description. *Organizational Behavior and Human Decision Processes*, 116(2), 286–295.

Lerer, A., Gross, S., & Fergus, R. (2016). Learning physical intuition of block towers by example. *arXiv preprint arXiv:1603.01312*.

Lew, T. F. & Vul, E. (2015). Ensemble clustering in visual working memory biases location memories and reduces the weber noise of relative positions. *Journal of Vision*, 15(4), 10–10.

Lewandowsky, S., Kalish, M., & Ngang, S. (2002). Simplified learning in complex situations: Knowledge partitioning in function learning. *Journal of Experimental Psychology: General*, 131(2), 163.

Lindley, D. V. (1956). On a measure of the information provided by an experiment. *The Annals of Mathematical Statistics*, (pp. 986–1005).

Lloyd, J. R., Duvenaud, D., Grosse, R., Tenenbaum, J. B., & Ghahramani, Z. (2014). Automatic construction and natural-language description of nonparametric regression models. *arXiv preprint arXiv:1402.4304*.

Lucas, C. G., Griffiths, T. L., Williams, J. J., & Kalish, M. L. (2015). A rational model of function learning. *Psychonomic Bulletin & Review*, 22(5), 1193.

Ludvig, E. A., Sutton, R. S., & Kehoe, E. J. (2008). Stimulus representation and the timing of reward-prediction errors in models of the dopamine system. *Neural Computation*, 20(12), 3034–3054.

March, J. G. (1991). Exploration and exploitation in organizational learning. *Organization Science*, 2(1), 71–87.

Marr, D. & Hildreth, E. (1980). Theory of edge detection. *Proceedings of the Royal Society of London: Biological Sciences*, 207(1167), 187–217.

Mathy, F. & Feldman, J. (2012). What's magic about magic numbers? Chunking and data compression in short-term memory. *Cognition*, 122(3), 346–362.

May, B. C., Korda, N., Lee, A., & Leslie, D. S. (2012). Optimistic Bayesian sampling in contextual-bandit problems. *Journal of Machine Learning Research*, 13(Jun), 2069–2106.

McDaniel, M. A., Dimperio, E., Griego, J. A., & Busemeyer, J. R. (2009). Predicting transfer performance: A comparison of competing function learning models. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 35(1), 173.

McFadden, D. (1973). Conditional logit analysis of qualitative choice behavior. In P. Zarembka (Ed.), *Frontiers in Econometrics* (pp. 105–142). New York: Academic Press.

Meder, B. & Nelson, J. D. (2012). Information search with situation-specific reward functions. *Judgment and Decision Making*, 7, 119–148.

Medvec, V. H., Madey, S. F., & Gilovich, T. (1995). When less is more: Counterfactual thinking and satisfaction among Olympic medalists. *Journal of Personality and Social Psychology*, 69(4), 603.

Metzen, J. H. (2016). Minimum regret search for single-and multi-task optimization. *arXiv preprint arXiv:1602.01064*.

Miller, G. A. (1956). The magical number seven, plus or minus two: Some limits on our capacity for processing information. *Psychological Review*, 63, 81–97.

Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., Graves, A., Riedmiller, M., Fidjeland, A. K., Ostrovski, G., et al. (2015). Human-level control through deep reinforcement learning. *Nature*, 518(7540), 529–533.

Močkus, J. (1975). On Bayesian methods for seeking the extremum. In *Optimization Techniques IFIP Technical Conference* (pp. 400–404).: Springer.

Mockus, J., Tiesis, V., & Zilinskas, A. (1978). The application of Bayesian methods for seeking the extremum. *Towards global optimization*, 2(117-129), 2.

Momennejad, I., Russek, E. M., Cheong, J. H., Botvinick, M. M., Daw, N., & Gershman, S. J. (2017). The successor representation in human reinforcement learning. *bioRxiv*, (pp. 083824).

Montero, P. & Vilar, J. (2014). Tsclust: Time series clustering utilities. *R package version*, 1(1).

Mullen, K. M., Ardia, D., Gil, D. L., Windover, D., & Cline, J. (2009). DEoptim: An R package for global optimization by differential evolution.

Neal, R. M. (1996). Priors for infinite networks. In *Bayesian Learning for Neural Networks* (pp. 29–53). Springer.

Neimark, E. D. & Shuford, E. (1959). Comparison of predictions and estimates in a probability learning situation. *Journal of Experimental Psychology*, 57(5), 294.

Nelson, J. D. (2005). Finding useful questions: On Bayesian diagnosticity, probability, impact, and information gain. *Psychological Review*, 112(4).

Nosofsky, R. M. (1986). Attention, similarity, and the identification–categorization relationship. *Journal of Experimental Psychology: General*, 115(1), 39.

Nosofsky, R. M., Palmeri, T. J., & McKinley, S. C. (1994). Rule-plus-exception model of classification learning. *Psychological Review*, 101(1), 53.

Opper, M. & Vivarelli, F. (1999). General bounds on Bayes errors for regression with Gaussian processes. In *Advances in Neural Information Processing Systems* (pp. 302–308).

Orhan, A. E. & Jacobs, R. A. (2013). A probabilistic clustering theory of the organization of visual short-term memory. *Psychological Review*, 120(2), 297.

Osman, M. (2010). Controlling uncertainty: A review of human behavior in complex dynamic environments. *Psychological Bulletin*, 136(1), 65.

Parpart, P., Schulz, E., Speekenbrink, M., & Love, B. C. (2015). Active learning as a means to distinguish among prominent decision strategies. In *Proceedings of the 37th Annual Meeting of the Cognitive Science Society* (pp. 1829–1834).

Pashler, H. (1988). Familiarity and visual change detection. *Attention, Perception, & Psychophysics*, 44(4), 369–378.

Paulhus, D. L., Lysy, D. C., & Yik, M. S. (1998). Self-report measures of intelligence: Are they useful as proxy iq tests? *Journal of Personality*, 66(4), 525–554.

Pavlov, I. P. (1941). Lectures on conditioned reflexes. vol. ii. conditioned reflexes and psychiatry.

Percival, D. B. & Walden, A. T. (2006). *Wavelet methods for time series analysis*. Cambridge University Press.

Peyton Jones, S. L. (1987). *The implementation of functional programming languages (prentice-hall international series in computer science)*. Prentice-Hall, Inc.

Piantadosi, S. T., Tenenbaum, J. B., & Goodman, N. D. (2016). The logical primitives of thought: Empirical foundations for compositional cognitive models. *Psychological Review*, 123(4), 392.

Poloczek, M., Wang, J., & Frazier, P. I. (2016). Warm starting Bayesian optimization. In *Winter Simulation Conference (WSC), 2016* (pp. 770–781).: IEEE.

Rajeswar, S., Subramanian, S., Dutil, F., Pal, C., & Courville, A. (2017). Adversarial generation of natural language. *arXiv preprint arXiv:1705.10929*.

Rasmussen, C. E. & Williams, C. K. (2006). *Gaussian processes for machine learning*, volume 1. MIT Press Cambridge.

Reece, S. & Roberts, S. (2010). An introduction to Gaussian processes for the Kalman filter expert. In *Information Fusion (FUSION), 2010 13th Conference on* (pp. 1–9).: IEEE.

Ritter, K. (2007). *Average-case analysis of numerical problems*. Springer.

Romero, P. A., Krause, A., & Arnold, F. H. (2013). Navigating the protein fitness landscape with Gaussian processes. *Proceedings of the National Academy of Sciences*, 110(3), E193–E201.

Rouder, J. N., Morey, R. D., Morey, C. C., & Cowan, N. (2011). How to measure working memory capacity in the change detection paradigm. *Psychonomic Bulletin & Review*, 18(2), 324–330.

Sanborn, A. N. & Chater, N. (2017). The sampling brain. *Trends in Cognitive Sciences*, 21(7), 492–493.

Sanborn, A. N., Griffiths, T. L., & Shiffrin, R. M. (2010). Uncovering mental representations with Markov chain Monte Carlo. *Cognitive Psychology*, 60(2), 63–106.

Schmidt, L. A. (2009). *Meaning and compositionality as statistical induction of categories and constraints*. PhD thesis, Massachusetts Institute of Technology.

Schölkopf, B. (2015). Artificial intelligence: Learning to see and act. *Nature*, 518(7540), 486–487.

Schulz, E., Huys, Q. J., Bach, D. R., Speekenbrink, M., & Krause, A. (2016a). Better safe than sorry: Risky function exploitation through safe optimization. In *Proceedings of the Thirty-Eighth Annual Conference of the Cognitive Science Society* (pp. 1140–1145).: Cognitive Science Society.

Schulz, E., Klenske, E., Bramley, N., & Speekenbrink, M. (2017a). Strategic exploration in human adaptive control. In *Proceedings of the 39th Annual Conference of the Cognitive Science Society*.

Schulz, E., Konstantinidis, E., & Speekenbrink, M. (2015a). Exploration-exploitation in a contextual multi-armed bandit task. In *International Conference on Cognitive Modeling* (pp. 118–123).

Schulz, E., Konstantinidis, E., & Speekenbrink, M. (2015b). Learning and decisions in contextual multi-armed bandit tasks. In *Proceedings of the Thirty-Seventh Annual Conference of the Cognitive Science Society* (pp. 2122–2127).: Cognitive Science Society.

Schulz, E., Konstantinidis, E., & Speekenbrink, M. (in press). Putting bandits into context: How function learning supports decision making. *Journal of Experimental Psychology: Learning, Memory, and Cognition.*

Schulz, E., Speekenbrink, M., Hernández-Lobato, J. M., Ghahramani, Z., & Gershman, S. J. (2016b). Quantifying mismatch in Bayesian optimization. In *NIPS Workshop on Bayesian Optimization: Black-box Optimization and beyond.*

Schulz, E., Speekenbrink, M., & Krause, A. (2017b). A tutorial on Gaussian process regression with a focus on exploration-exploitation scenarios. *bioRxiv*, (pp. 095190).

Schulz, E., Speekenbrink, M., & Meder, B. (2016c). Simple trees in complex forests: Growing take the best by approximate bayesian computation. In *Proceedings of the 38th Annual Conference of the Cognitive Science Society*: Cognitive Science Society.

Schulz, E., Tenenbaum, J., Duvenaud, D. K., Speekenbrink, M., & Gershman, S. J. (2016d). Probing the compositionality of intuitive functions. In *Advances in Neural Information Processing Systems* (pp. 3729–3737).

Schulz, E., Tenenbaum, J. B., Duvenaud, D., Speekenbrink, M., & Gershman, S. J. (2016e). Compositional inductive biases in function learning. *bioRxiv.*

Schulz, E., Tenenbaum, J. B., Reshef, D. N., Speekenbrink, M., & Gershman, S. (2015c). Assessing the perceived predictability of functions. In *Proceedings of the 37th Annual Meeting of the Cognitive Science Society* (pp. 2116–2121). Cognitive Science Society.

Seo, S., Wallat, M., Graepel, T., & Obermayer, K. (2000). Gaussian process regression: Active data selection and test point rejection. In *Neural Networks, 2000. IJCNN 2000, Proceedings of the IEEE-INNS-ENNS International Joint Conference on*, volume 3 (pp. 241–246).: IEEE.

Shahriari, B., Swersky, K., Wang, Z., Adams, R. P., & de Freitas, N. (2016). Taking the human out of the loop: A review of bayesian optimization. *Proceedings of the IEEE*, 104(1), 148–175.

Shepard, R. N. (1957). Stimulus and response generalization: A stochastic model relating generalization to distance in psychological space. *Psychometrika*, 22(4), 325–345.

Shepard, R. N. (1958). Stimulus and response generalization: Tests of a model relating generalization to distance in psychological space. *Journal of Experimental Psychology*, 55(6), 509.

Shepard, R. N. (1981). Psychological relations and psychophysical scales: On the status of "direct" psychophysical measurement. *Journal of Mathematical Psychology*, 24(1), 21–57.

Shepard, R. N. (1987). Toward a universal law of generalization for psychological science. *Science*, 237(4820), 1317–1323.

Shepard, R. N., Hovland, C. I., & Jenkins, H. M. (1961). Learning and memorization of classifications. *Psychological monographs: General and applied*, 75(13), 1.

Silver, D., Huang, A., Maddison, C. J., Guez, A., Sifre, L., Van Den Driessche, G., Schrittwieser, J., Antonoglou, I., Panneershelvam, V., Lanctot, M., et al. (2016). Mastering the game of go with deep neural networks and tree search. *Nature*, 529(7587), 484–489.

Simon, H. A. (1978). Rationality as process and as product of thought. *The American Economic Review*, (pp. 1–16).

Simonite, T. (2016). How google plans to solve artificial intelligence.

Skinner, B. F. (1945). The operational analysis of psychological terms. *Psychological Review*, 52(5), 270.

Smith, P. L. (2000). Stochastic dynamic models of response time and accuracy: A foundational primer. *Journal of Mathematical Psychology*, 44(3), 408–463.

Snoek, J., Larochelle, H., & Adams, R. P. (2012). Practical bayesian optimization of machine learning algorithms. In *Advances in Neural Information Processing Systems* (pp. 2951–2959).

Sollich, P. (2002). Gaussian process regression with mismatched models. In *Advances in Neural Information Processing Systems* (pp. 519–526).

Sollich, P. (2005). Can Gaussian process regression be made robust against model mismatch? *Lecture notes in computer science*, 3635, 199.

Sollich, P. & Halees, A. (2002). Learning curves for Gaussian process regression: approximations and bounds. *Neural Computation*, 14(6), 1393–1428.

Solway, A. & Botvinick, M. M. (2015). Evidence integration in model-based tree search. *Proceedings of the National Academy of Sciences*, 112(37), 11708–11713.

Speekenbrink, M., Channon, S., & Shanks, D. R. (2008). Learning strategies in amnesia. *Neuroscience & Biobehavioral Reviews*, 32(2), 292–310.

Speekenbrink, M. & Konstantinidis, E. (2015). Uncertainty and exploration in a restless bandit task. *Topics in Cognitive Science*, 7, 351–367.

Speekenbrink, M. & Shanks, D. R. (2008). Through the looking glass: A dynamic lens model approach to multiple cue learning. In N. Chater & M. Oaksford (Eds.), *The probabilistic mind*, number 18 (pp. 409–429). Oxford University Press.

Srinivas, N., Krause, A., Kakade, S., & Seeger, M. W. (2010). Gaussian process optimization in the bandit setting: No regret and experimental design. In *Proceedings of the 27th International Conference on Machine Learning* (pp. 1015–1022).

Srinivas, N., Krause, A., Kakade, S. M., & Seeger, M. W. (2012). Information-theoretic regret bounds for gaussian process optimization in the bandit setting. *IEEE Transactions on Information Theory*, 58(5), 3250–3265.

Sternberg, S. et al. (1966). High-speed scanning in human memory. *Science*, 153(3736), 652–654.

Steyvers, M., Lee, M. D., & Wagenmakers, E.-J. (2009). A Bayesian analysis of human decision-making on bandit problems. *Journal of Mathematical Psychology*, 53(3), 168–179.

Stojic, H., Analytis, P. P., & Speekenbrink, M. (2015). Human behavior in contextual multi-armed bandit problems. In *Proceedings of the 37th Annual Meeting of the Cognitive Science Society* (pp. 2290–2295).: Cognitive Science Society.

Stone, G. O. (1986). An analysis of the delta rule and the learning of statistical associations. *Parallel distributed processing: Explorations in the microstructure of cognition*, 1, 444–459.

Sutton, R. S. (1996). Generalization in reinforcement learning: Successful examples using sparse coarse coding. *Advances in Neural Information Processing Systems*, (pp. 1038–1044).

Sutton, R. S. & Barto, A. G. (1998). *Reinforcement learning: An introduction*, volume 1. MIT press Cambridge.

Tenenbaum, J. B. & Griffiths, T. L. (2001). Generalization, similarity, and Bayesian inference. *Behavioral and Brain Sciences*, 24(4), 629–640.

Tenenbaum, J. B., Kemp, C., Griffiths, T. L., & Goodman, N. D. (2011). How to grow a mind: Statistics, structure, and abstraction. *Science*, 331(6022), 1279–1285.

Tesauro, G. (1992). Practical issues in temporal difference learning. *Machine learning*, 8(3-4), 257–277.

Thorndyke, P. W. (1977). Cognitive structures in comprehension and memory of narrative discourse. *Cognitive Psychology*, 9(1), 77–110.

Titchener, E. B. (1914). An historical note on the james-lange theory of emotion. *The American Journal of Psychology*, 25(3), 427–447.

Tolman, E. C. (1948). Cognitive maps in rats and men. *Psychological Review*, 55(4), 189.

Tsividis, P. A., Pouncy, T., Xu, J. L., Tenenbaum, J. B., & Gershman, S. J. (2017). Human learning in atari. In *The AAAI 2017 Spring Symposium on Science of Intelligence: Computational Principles of Natural and Artificial Intelligence*.

van der Vaart, A. W. & van Zanten, J. H. (2008). Rates of contraction of posterior distributions based on Gaussian process priors. *The Annals of Statistics*, (pp. 1435–1463).

von Neumann, J., Kent, R., Bellinson, H., & Hart, B. t. (1941). The mean square successive difference. *The Annals of Mathematical Statistics*, 12(2), 153–162.

Wang, Z., Deisenroth, M. P., Amor, H. B., Vogt, D., Schölkopf, B., & Peters, J. (2012). Probabilistic modeling of human movements for intention inference. *Proceedings of Robotics: Science and Systems, VIII*.

Watkins, C. J. C. H. (1989). *Learning from delayed rewards*. PhD thesis, King's College, Cambridge.

Watson, J. B. et al. (1958). *Behaviorism*. Transaction Publishers.

Weiss-Cohen, L., Konstantinidis, E., Speekenbrink, M., & Harvey, N. (2016). Incorporating conflicting descriptions into decisions from experience. *Organizational Behavior and Human Decision Processes*, 135, 55–69.

Williams, C. K. & Rasmussen, C. E. (1996). Gaussian processes for regression. *Advances in Neural Processing Systems*, 8, 514–520.

Williams, C. K. & Rasmussen, C. E. (2006). Gaussian processes for machine learning. *the MIT Press*, 2(3), 4.

Williams, C. K. & Vivarelli, F. (2000). Upper and lower bounds on the learning curve for Gaussian processes. *Machine Learning*, 40(1), 77–102.

Wilson, A. & Adams, R. (2013). Gaussian process kernels for pattern discovery and extrapolation. In *Proceedings of the 30th International Conference on Machine Learning (ICML-13)* (pp. 1067–1075).

Wilson, A. G., Dann, C., Lucas, C., & Xing, E. P. (2015). The human kernel. In *Advances in Neural Information Processing Systems* (pp. 2854–2862).

Wilson, R. C., Geana, A., White, J. M., Ludvig, E. A., & Cohen, J. D. (2014). Humans use directed and random exploration to solve the explore–exploit dilemma. *Journal of Experimental Psychology: General*, 143(6), 2074.

Wu, C. M., Schulz, E., Speekenbrink, M., Nelson, J. D., & Meder, B. (2017). Mapping the unknown: The spatially correlated multi-armed bandit. In *Proceedings of the 39th Annual Conference of the Cognitive Science Society*: Cognitive Science Society.

Wu, C. M., Schulz, E., Speekenbrink, M., Nelson, J. D., & Meder, B. (submitted). Exploration and generalization in vast spaces.

Wundt, W. M. (1896). *Grundriss der Psychologie*. W. Engelmann.

Xu, Y., Honorio, J., & Wang, X. (2017). Statistical efficiency of compositional nonparametric prediction. *arXiv preprint arXiv:1704.01896*.

Zellner, A. (1962). An efficient method of estimating seemingly unrelated regressions and tests for aggregation bias. *Journal of the American statistical Association*, 57(298), 348–368.

Zhao, J., Ngo, N., McKendrick, R., & Turk-Browne, N. B. (2011). Mutual interference between statistical summary perception and statistical learning. *Psychological Science*, 22(9), 1212–1219.

Zhao, J. & Yu, R. Q. (2016). Statistical regularities reduce perceived numerosity. *Cognition*, 146, 217–222.