

Running head: EXPLORING INDUCTIVE BIASES

Bayesian models as tools for exploring inductive biases

Thomas L. Griffiths

Department of Psychology

University of California, Berkeley

Keywords: Bayesian models, inductive biases, function learning, bias-variance tradeoff

Word count: 6812

Address for correspondence:

Tom Griffiths

University of California, Berkeley

Department of Psychology

3210 Tolman Hall # 1650

Berkeley CA 94720-1650

E-mail: tom_griffiths@berkeley.edu **Phone:** (510) 642 7134 **Fax:** (510) 642 5293

Bayesian models as tools for exploring inductive biases

Generalization – reasoning from the properties of observed entities to those of entities as yet unobserved – is at the heart of many aspects of human cognition. As several of the chapters in this book indicate, generalization plays a key role in language learning, where learners need to make judgments about the linguistic properties of utterances (such as their meaning or grammaticality) using their previous experience with a language. It also underlies our ability to form and use categories, allowing us to identify which objects are likely to belong to a category based on a few examples, and is central to learning about causal relationships, where we predict how one event will influence another by drawing on past instances of those events. Language learning, categorization, and causal induction are three of the most widely-studied topics in cognitive science, and allow us to communicate about, organize, and intervene on our environment. They are also three examples of problems for which human performance exceeds that of automated systems, setting the standard to which artificial intelligence and machine learning research aspires. This raises a natural question: What makes people so good at generalization?

The ubiquity and importance of generalization in cognitive science derives in part from its close relationship to inductive inference. The definition of generalization given above emphasizes its relationship to the classic problem of induction (e.g., Hume, 1739/1978), in which one forms predictions about future events based on examples from the past, such as anticipating that the sun will rise tomorrow because of the many days on which the sun has risen in the past. We can also think of generalization in terms of extracting a general rule from these events that can be used to make further predictions. For instance, Figure 1 shows a collection of (x, y) pairs produced by a simple function $y = f(x)$ together with additive noise. The dotted line shows a new value of x , for which the value of y is unknown. Predicting the value of y for this new x based on the other

(x, y) pairs is a problem of generalization. In forming a prediction, you might take into account the different forms that the function $f(x)$ relating x and y might take on. Identifying the underlying function is an inductive problem. This simple example clearly has parallels in human causal learning, where people might want to understand the nature of the relationship between a cause and an effect, and there is an extensive literature exploring how people learn functions of this kind (for a review, see Busemeyer, Byun, DeLosh, & McDaniel, 1997).

The connection between generalization and induction means that the question of how to produce good generalizations becomes the question of what makes a good inductive learner. One reason why inductive problems such as language learning, categorization, and causal induction are particularly interesting from the perspective of cognitive science is that the hypotheses under consideration are not directly determined by the observed data. In order to choose among the many hypotheses that might provide equally good accounts of the data, the learner has to inject some subjective preferences for those hypotheses. This observation, which has a long history in cognitive science (e.g., Kant, 1781/1964; Helmholtz, 1866/1962), has two consequences. First, the way that people solve inductive problems can be uniquely revealing about the structure of the mind, as the conclusions people reach are a function of both data and their internal dispositions, which I will refer to as “inductive biases.” Second, these biases make the difference between a good inductive learner and a bad one, so understanding the consequences of having different inductive biases will provide an answer to the question of what makes a good inductive learner.

In this chapter, I explore the question of what makes a good inductive learner, emphasizing the role of inductive biases and the potential of Bayesian models of cognition as tools for revealing those biases. As a first step in this exploration, I consider the possibility that what makes a good inductive learner is simply being able to entertain a

richer set of hypotheses as to the structure contained in observed data. An example suffices to illustrate that this is not the whole story: richer hypothesis spaces can come with serious disadvantages. This motivates the discussion of a simple but deep result in mathematical statistics – the bias-variance tradeoff – which provides a way to understand the relevance of inductive biases to human learning. I then argue that probabilistic models in which inductive inference is performed using Bayes’ rule provide a way to formalize these biases, and summarize two methods of using Bayesian models to determine human inductive biases: making models with different biases and testing them against human data, and using the predictions of Bayesian models to design experiments that are explicitly intended to reveal inductive biases.

Inductive inference and the richness of the hypothesis space

At an abstract computational level (cf. Anderson, 1990; Marr, 1982), we can formalize inductive problems as follows. The goal of the learner is to generalize accurately from observed data. Denoting the data d , the learner seeks to identify the hypothesis h that will result in the greatest generalization accuracy. The hypothesis h will be selected from a set of hypotheses $\mathcal{H}$, which I will refer to as the “hypothesis space.” A learning algorithm is a procedure for using the data d to select a hypothesis h from a hypothesis space $\mathcal{H}$. To make these ideas more concrete, in the simple function learning problem from Figure 1, the data d are the (x, y) pairs, a hypothesis h is a function relating x and y , and a simple learning algorithm is selecting the function that most closely matches the points in d from the set of functions in the hypothesis space $\mathcal{H}$. Using this formal framework, the question of what makes a good inductive learner reduces to asking what makes a good learning algorithm.

Theories of cognitive development and the history of formal models of learning in cognitive science both suggest a simple answer to this question: that better learning

algorithms are characterized by richer hypothesis spaces. Piaget’s influential account of cognitive development suggested that as children develop, they become capable of entertaining more sophisticated hypotheses about the structure of their environment (e.g., Inhelder & Piaget, 1958). This change in representational capacity is viewed as the major force allowing children to acquire the knowledge associated with the adult state, a claim that has drawn criticism on logical grounds (e.g., Fodor, 1980). A similar trend appears in the history of models of human learning based on artificial neural networks. Despite initial enthusiasm about simple neural networks such as Rosenblatt’s (1958) perceptron, which automatically learned a set of weights for linearly mapping its inputs to its outputs, Minsky and Papert (1969) pointed out that these models were severely limited in the input-output mappings that they could learn. Specifically, perceptrons were only able to learn distinctions between sets of inputs that were linearly separable. This problem could be addressed by using multilayer perceptrons in which the outputs of one layer underwent a nonlinear transformation and became the inputs of the next, and Rumelhart, Hinton, and Wilson (1986) introduced the backpropagation algorithm for training multilayer neural networks. The procedure for selecting a hypothesis was based on the same criteria as that used in the perceptron: choosing the weights that allowed the model to most closely match the training data. The innovation was the expansion of the hypothesis space to make it possible to overcome the constraint of linear separability.

The idea that richer hypothesis spaces are the key to better inductive inference provides an attractively simple answer to the question of what makes a good inductive learner. However, it is straightforward to demonstrate that richer hypothesis spaces are not sufficient to guarantee good inductive inferences, and can actually prove detrimental. Consider three possible learning algorithms for solving the function learning problem depicted in Figure 1. As mentioned above, here the data are the (x, y) pairs and the hypotheses are functions relating x and y , producing predictions $h(x)$. The first learning

algorithm selects the linear function $h(x) = \theta_0 + \theta_1 x$ that minimizes the squared error between the prediction of the function at x and the true value of y , $(h(x) - y)^2$. The second and third learning algorithms use the same procedure to select a quadratic function $h(x) = \theta_0 + \theta_1 x + \theta_2 x^2$, and an 8th degree polynomial $h(x) = \theta_0 + \sum_{k=1}^8 \theta_k x^k$ respectively. Since the procedure for selecting a function is the same in each case, all that differs is the hypothesis space $\mathcal{H}$ from which the function is selected. Identifying functions with their parameters, the hypothesis space of the first learning algorithm is $\mathcal{H}_1 = (\theta_0, \theta_1) \in \mathcal{R}^2$, where $\mathcal{R}$ is the set of real numbers. The hypothesis space of the second learning algorithm is $\mathcal{H}_2 = (\theta_0, \theta_1, \theta_2) \in \mathcal{R}^3$, and is strictly richer than $\mathcal{H}_1$, since any linear function can be produced by setting $\theta_2 = 0$ (i.e. $\mathcal{H}_1 \subset \mathcal{H}_2$). Likewise, $\mathcal{H}_3 = (\theta_0, \dots, \theta_8) \in \mathcal{R}^9$, and is strictly richer than $\mathcal{H}_2$.

The consequences of using these three learning algorithms are shown in Figure 2.¹ For these data, the linear hypothesis space $\mathcal{H}_1$ produces a function that does not correspond particularly closely to the data, the quadratic hypothesis space $\mathcal{H}_2$ produces lower error, and the set of 8th degree polynomials $\mathcal{H}_3$ results in very little error on any of the points. So, richer hypothesis spaces produce functions that match the data more closely, as should be expected since these spaces provide more options from which to select. However, this does not guarantee that algorithms using richer hypothesis spaces will always do a better job of generalization. Looking at the three functions in Figure 2, which would you select to make your prediction of y for the new value of x ? In fact, these data were generated from a quadratic function, very similar to that found using $\mathcal{H}_2$. Consequently, in this case, the algorithm using $\mathcal{H}_2$ would generalize better than that using $\mathcal{H}_3$. The more powerful learning algorithm, the one with the richer hypothesis space, would perform worse. So, why is a richer hypothesis space not sufficient for improved generalization?

The bias-variance tradeoff

A simple explanation for why richer hypothesis spaces can be problematic is that having more options makes it possible to fit noise in the data. The points shown in Figure 1 were generated by adding random noise to the true underlying function. As a consequence, a learning algorithm can “overfit” the data, capturing the noise as well as the systematic variation produced by the underlying function. This makes the algorithm extremely sensitive to the specific points that are observed, and leads to a great deal of variability in its predictions. However, the example considered above also illustrates that richer hypothesis spaces can be beneficial: the learning algorithm considering only linear functions performed poorly, because the true function (in this case a quadratic) was outside its hypothesis space. So, richer hypothesis spaces seem to aid generalization in some cases, and hinder it in others.

We can begin to understand how the assumptions made by different learning algorithms affect generalization by formally analyzing the factors contributing to generalization performance. In this section, I will briefly summarize an elegant result from the statistics literature, showing that generalization error decomposes into two parts – the bias of a learning algorithm, and its variance (Geman, Bienenstock, & Doursat, 1992). While I will focus on the case of learning functions, this kind of analysis can be performed for simple inductive problems of many kinds, including categorization problems (Domingos, 2000; Friedman, 1997; James & Hastie, 1997; Tibshirani, 1996), and the general spirit of the results is recapitulated in many other formal analyses of learning (Vapnik, 1995; Kearns & Vazirani, 1994). The presentation here follows that of other discussions of the bias-variance tradeoff in the machine learning literature (Bishop, 2006; Hastie, Tibshirani, & Friedman, 2001).

In the function learning example, our data are produced by selecting a set of x values from a distribution $p(x)$, and then generating corresponding y values from a distribution

$p(y|x)$ that is Gaussian with mean $f(x)$ and standard deviation σ . This defines a joint distribution $p(x, y)$ on (x, y) pairs. A learning algorithm returns a hypothesis h , which we will take to be a function $y = h(x)$, for data d consisting of n points drawn from this distribution. The generalization error, GE , associated with such a function will be the average of the error associated with a particular (x, y) pair over the distribution $p(x, y)$, or

$$GE = \int \int (y - h(x))^2 p(x, y) dx dy = E_{p(x, y)} [(y - h(x))^2] \quad (1)$$

where $E_{p(\cdot)}[\cdot]$ is the expectation of its argument over the distribution $p(\cdot)$. With some algebra, we can simplify this to

$$E_{p(x, y)} [(y - h(x))^2] = E_{p(x, y)} [(y - f(x) + f(x) - h(x))^2] \quad (2)$$

$$= E_{p(x, y)} [(y - f(x))^2] + E_{p(x, y)} [(f(x) - h(x))^2] \quad (3)$$

$$= \sigma^2 + E_{p(x)} [(f(x) - h(x))^2] \quad (4)$$

where the second line uses the linearity of expectation and the fact that $E_{p(x, y)} [(y - f(x))(f(x) - h(x))]$ is zero, and the third line uses the definition of $p(y|x)$ and the observation that y does not appear in $f(x) - h(x)$ (see Geman et al., 1992, for details).

The basic conclusion suggested by Equation 4 is that generalization error can be attributed to a combination of intrinsic error due to the noise in y , represented by σ^2 , and systematic error resulting from the difference between the true function $f(x)$ and the function selected by the learning algorithm, $h(x)$. A good hypothesis – one that produces low generalization error – will thus be one that makes $(f(x) - h(x))^2$ small for arbitrary x . However, we need to take into account the fact that the hypothesis $h(x)$ chosen by the learning algorithm depends on the data d . To do this, we need to compute the expected

generalization error of a learning algorithm, EGE , over data d . This is simply

$$EGE = E_{p(d)}[GE] \quad (5)$$

where $p(d)$ is the probability distribution that results from drawing n points from $p(x, y)$ and GE is given in Equation 1. A good learning algorithm – one that produces low expected generalization error – will thus be one for which the expectation of its error, $(f(x) - h(x))^2$, over possible datasets d is small. We can express this as

$$E_{p(d)} [(f(x) - h(x))^2] = (f(x) - E_{p(d)}[h(x)])^2 + E_{p(d)} [(h(x) - E_{p(d)}[h(x)])^2] \quad (6)$$

where the derivation is similar derivation to that used in Equations 2-4, adding and subtracting $E_{p(d)}[h(x)]$ in the same way we added and subtracted $f(x)$ above.

Equation 6 might seem relatively arcane at first glance, but it actually has a simple and intuitive interpretation. The first term on the right hand side reflects differences between the true function $f(x)$ and the predictions made by the learning algorithm when averaged over all datasets. This is a measure of the *bias* of the learning algorithm – its overall capacity to capture the right form of the function. Formally, the bias at a point x is defined to be

$$\text{bias} = f(x) - E_{p(d)}[h(x)]. \quad (7)$$

The second term on the right hand side expresses the degree to which the hypothesis selected by the learning algorithm changes as a function of d . This is the *variance* of the learning algorithm, by analogy to the variance of a probability distribution, being the expected squared difference between the predictions and their average. Formally, the

variance at a point x is

$$\text{variance} = E_{p(d)} \left[\left(h(x) - E_{p(d)}[h(x)] \right)^2 \right]. \quad (8)$$

Thus, we can rewrite Equation 6 as

$$E_{p(d)} \left[(f(x) - h(x))^2 \right] = \text{bias}^2 + \text{variance} \quad (9)$$

making it clear that the performance of the learning algorithm can be described in terms of just these two factors. It follows from Equations 4 and 5 that the expected generalization error is just σ^2 plus the expectation of the sum of the squared bias and the variance with respect to $p(x)$.

The decomposition of the expected generalization error into bias and variance provides a simple framework for understanding the properties of different learning algorithms. I will illustrate the contributions of these factors by returning to the three learning algorithms introduced above, which differ only in the hypothesis spaces that they use. These learning algorithms are simple enough that we can directly compute their bias and variance, if we make some assumptions about the distribution $p(x, y)$ from which (x, y) pairs are being generated. I took $f(x)$ to be the quadratic function used above, fixed a value for σ^2 , and took $p(x)$ to be uniform over the range shown in the graphs in Figures 1 and 2. I then generated 100 samples with $n = 10$ from this distribution, and applied the learning algorithms to these data, producing 100 instances of $h(x)$ for each algorithm. These 100 hypotheses are thus samples from the distribution on $h(x)$ produced by generating data from $p(d)$. The results are shown in Figure 3, together with a function produced by averaging these predictions, and the true function $f(x)$.

Figure 3 provides an intuitive illustration of the meaning of bias and variance, and how they contribute to generalization error. The bias of a learning algorithm is the

difference between its average predictions and the true function. Its variance is the amount of variation that it shows around that average. The hypothesis space of linear functions results in a high bias, with the average function differing significantly from the true function, but reasonably low variance. The hypothesis space of 8th degree polynomials results in low bias, with the average function being close to the true function, but incredibly high variance, with very different predictions being produced by different samples of d . The hypothesis space of quadratic functions has both low bias and low variance, producing an average function close to the true function and exhibiting little variation across samples.

The decomposition of expected generalization error into bias and variance provides us with a way to understand how richer hypothesis spaces can sometimes help generalization, and sometimes hurt it. Richer hypothesis spaces help because they reduce bias. Going from linear functions to quadratic functions adds enough flexibility to that it becomes possible to actually match the true function. Richer hypothesis space hurt because they increase variance. Going from quadratic functions to 8th degree polynomials adds so much flexibility that it becomes possible to overfit the data, producing highly variable predictions. This transition from one source of error to another is the *bias-variance tradeoff*, and much of the work in designing learning algorithms is about trying to hit the sweet spot between bias and variance for a given problem.

Before considering the implications of this analysis for understanding human inductive inference, it is worth noting two subtle points about how bias and variance depend on different factors involved in learning. The first is that this formal notion of bias is relative. From Equation 7, it should be clear that this quantity depends intimately on the nature of the true function, $f(x)$. If the true function were linear, then the algorithm using the linear hypothesis space would have a low bias. If the true function were cubic, then the algorithm using the quadratic hypothesis space would have a high bias. The

dependence of the bias on the true function is somewhat counter-intuitive, so I use the term “inductive biases” to describe the dispositions that guide learners in solving inductive problems, reflecting the extent to which they favor one kind of hypothesis over another. These inductive biases will be the same regardless of the true function. For example, the learning algorithm using the linear hypothesis space will always produce a linear function, and we can describe its inductive biases in these terms.

The second subtlety of the bias-variance tradeoff is that variance is strongly affected by the amount of data provided to the learning algorithm. Figure 4 shows the results of repeating the procedure used to construct Figure 3, but with $n = 100$ rather than $n = 10$. Now, whether the hypothesis space is quadratic functions or 8th degree polynomials has little effect on the variance of the learning algorithm: the data are sufficient to strongly determine a solution either way. The sweet spot between bias and variance thus changes as the amount of data increases, with richer hypothesis spaces being less problematic when data are plentiful.

Implications for human inductive inference

The bias-variance tradeoff suggests that the answer to the question of what makes a good inductive learner is going to depend on the kind of problems that the learner will face. If the learner will be provided with only small amounts of data, then variance is a real concern and the only way to guarantee accurate generalization is by having inductive biases that match the problem at hand (i.e. that make the bias, in the formal sense, small). If the learner will be provided with large amounts of data, and needs to be able to solve a variety of problems, then variance is a less significant issue and bias is dominant: the learner needs to be flexible enough to accommodate the different solutions that could be needed for different problems.

Interestingly, these two perspectives map loosely onto the two extreme positions

found in discussion of human inductive biases in cognitive science: that the relevant biases are strong and specific to particular learning domains (e.g., Chomsky, 1965), or that the biases are weak, and the result of domain-general learning mechanisms (e.g., Elman et al., 1996; Rumelhart & McClelland, 1986; Rogers & McClelland, 2004). Arguments for strong biases emphasize limitations in the amount or the quality of data provided to learners, setting up the problem as one in which variance is dominant. Domain-specificity is a natural corollary of this view, since different domains will provide different targets for learning, and require different inductive biases. Arguments for weak biases focus on the possibility of a single learning algorithm that can be applied across many domains – something that requires flexibility, such as that exhibited in the large hypothesis space of nonlinear functions that can be produced by multilayer neural networks. These approaches downplay the issue of the amount of data available to learners. For example, algorithms such as backpropagation that seek to minimize error on some training data treat the amount of data and the number of iterations of learning equivalently, meaning that a small but representative sample will result in the same predictions as a larger sample in which the same points appear many times.

More generally, the bias-variance tradeoff makes it clear that understanding human inductive inference means understanding human inductive biases. The key to good generalization is having inductive biases that provide a good compromise between bias and variance across a range of problems. By identifying the biases that guide human inductive inference, we can examine whether either of these two extreme views is correct. We can also explore the continuum of positions between these two extremes. However, in order to do so we need a good way to systematically and transparently characterize the inductive biases of a learner.

Bayesian inference and inductive biases

Bayesian inference is a formalism that makes the inductive biases of learners particularly clear. It also provides a rational account of how learners should go about revising their beliefs in light of evidence (e.g., Robert, 1994). In this section, I will summarize the basic ideas behind this approach and illustrate them using the example of learning functions. My emphasis here will be on a Bayesian analysis of inductive inference in broad terms. A Bayesian analysis of generalization in the stricter sense of reasoning from observed properties of a set of objects to the unobserved properties of another object is given in Tenenbaum and Griffiths (2001), building on that of Shepard (1987).

The basic assumption behind Bayesian inference is that learners are willing to express their degrees of belief in different hypotheses using probabilities. Several formal arguments can be made in support of this assumption (Cox, 1961; Jaynes, 2003), and once it is accepted the process of revising beliefs becomes straightforward, being a matter of applying *Bayes' rule*. Assume that a learner has a *prior* probability distribution, $p(h)$, specifying the probability assigned to the truth of each hypothesis h in a set of hypotheses $\mathcal{H}$ before seeing d . Bayes' rule states that the probability that should be given to each hypothesis after seeing d – known as the *posterior* probability, $p(h|d)$ – is

$$p(h|d) = \frac{p(d|h)p(h)}{\sum_{h \in \mathcal{H}} p(d|h)p(h)} \quad (10)$$

where $p(d|h)$ – the *likelihood* – indicates how probable d is under hypothesis h . The sum in the denominator simply guarantees that this process produces a probability distribution, ensuring that the resulting probabilities sum to one.

Bayes' rule provides an elegant way to characterize the inductive biases of learners, through the prior distribution over hypotheses $p(h)$. Interpreted probabilistically, priors indicate the kind of world a learner expects to encounter, guiding their conclusions when

provided with data. The prior probability associated with each hypothesis reflects the probability with which a learner expects that hypothesis to be the solution to an inductive problem. We can easily translate the function learning problem that I have been using as a running example into a problem of Bayesian inference. The assumption that y is generated from a Gaussian distribution with mean $h(x)$ and standard deviation σ gives us our likelihood function $p(d|h)$, and means that the likelihood is maximized by minimizing the sum of the squared errors $(y - h(x))^2$. If we define our priors to be uniform over all functions in our hypothesis space, giving constant prior probability to every function in the set, then the hypothesis with maximum posterior probability is simply the hypothesis in that set with highest likelihood. Selecting the hypothesis with maximum posterior probability would thus be equivalent to the learning algorithm that chooses the function in the hypothesis space that minimizes the sum of the squared errors.

Using this simple method of setting a prior – choosing a hypothesis space and assigning probabilities uniformly within that space – and selecting the hypothesis with highest posterior probability produces exactly the same results for the linear, quadratic, and 8th degree polynomial hypothesis spaces as in our example above. However, it helps to explain why we saw those results. The prior expresses the expectations of a learner, and a learner who assigns equal prior probability to every 8th degree polynomial believes that it lives in a world where it is likely to encounter extremely complex functions. While the function shown in Figure 2 (c) seems like a wild conjecture to us, it is just as plausible a priori as any other 8th degree polynomial to the learner. The issue is that its beliefs are poorly calibrated to the quadratic world in which it lives.

More generally, priors can be used not just to limit the set of hypotheses, but to constrain their plausibility. In our function learning example, we can use 8th degree polynomials as our hypothesis space, providing us with the potential to produce complex functions, but define a prior that favors simpler functions. This can be done by giving

higher prior probability to those functions for which the θ_k are small. One approach is to take a Gaussian prior for each θ_k , with a mean of 0 and a standard deviation of $\alpha\beta^k$. For $\beta < 1$, the variance decreases geometrically with k , meaning that the coefficients of higher powers of x are increasingly likely to be small. Figure 5 shows samples from this prior for three different values of β , illustrating how the prior probability of different kinds of functions is affected by this parameter. Figure 6 shows the consequences of choosing the hypothesis with maximum posterior probability using the Gaussian likelihood introduced above together with these three priors.² As with constraining the hypothesis space directly, the resulting predictions vary in how well they correspond to the true function. However, one significant difference is that a learner using any of these three priors could ultimately learn any 8th degree polynomial, if there are enough data that suggest that this complexity is warranted, rather than being stuck within a limited hypothesis space. Assigning graded degrees of plausibility across a large hypothesis space is an effective way to define models that are sufficiently constrained that they can generalize well from small amounts of data, yet sufficiently flexible that they can learn complex functions from large amounts of data. Variants on this approach are widely used in machine learning (Bishop, 2006; Hastie et al., 2001).

Finally, it is important to note that the prior distribution assumed by a learner should not be interpreted simply as reflecting innate constraints on learning. The prior simply collects together all of the factors affecting how plausible a learner finds a particular hypothesis. There are many such factors other than innate constraints that could affect this, such as data from other domains that is independent of the data observed in this domain but might provide a source of hypotheses, or information-processing constraints, such as limitations on working memory. It should also be clear that many learning algorithms, including those used with multilayer neural networks, can be interpreted as having a prior over some hypothesis space. That prior might just be relatively weak, like

the uniform priors over hypothesis spaces of functions discussed above. For example, when viewed from this perspective, the learning algorithms typically used with multilayer neural networks can be interpreted as defining a uniform prior over the hypothesis space of functions that can be expressed through their weights (MacKay, 1995; Neal, 1992).

Revealing inductive biases

Bayesian models provide a way to express a variety of inductive biases, through the prior distribution over hypotheses that they assume. These models make predictions about which hypotheses people should select as a consequence of observing different kinds of data. This suggests at least two ways to explore the inductive biases of human learners: comparing human judgments to Bayesian models that assume different priors, and using the assumptions behind Bayesian models to design experiments that are explicitly intended to reveal the priors of the participants. I will outline these two approaches, providing examples of cases in which they have been successful.

Testing assumed biases

The simplest approach to using Bayesian models to reveal human inductive biases is to construct a set of models that assume different priors, and examine which of these models best characterizes human performance on a task. This approach is consistent with a long tradition of computational modeling in cognitive science, in which parameterized models are fit to human data and compared in order to evaluate claims about the processes behind behavior. For example, a variety of models have been proposed to explain for how people perform function learning tasks like that used in our example (Busemeyer et al., 1997; DeLosh, Busemeyer, & McDaniel, 1997; Kalish, Lewandowsky, & Kruschke, 2004). While not expressed in Bayesian terms, these models differ in the inductive biases that they ascribe to learners, and tests of these models explore what kind of biases seem to give a better account of human judgments.

Several studies have explicitly used Bayesian models in this way. Anderson (1990; Anderson & Milson, 1989) introduced a probabilistic model of memory, in which retrieval was viewed as a process of inferring whether an item in memory was likely to be needed based on a cue. This model required a prior distribution over items in memory, indicating the probability assigned to a given item being needed at a particular time. This “need probability” was originally estimated using a model based on the behavior of books in a library, but Anderson and Schooler (1991) subsequently measured the probabilities of items being needed as a function of time from three environmental sources: headlines in the New York Times, a corpus of child-directed speech, and people who sent e-mail to Anderson. They were able to show that using priors estimated from these data resulted in accurate predictions of basic memory effects, such as retention curves, the effects of practice, and the spacing of exposures. This approach provides a particularly elegant explanation for these phenomena, grounding them in a prior obtained directly from the environment. Similar methods are commonly used in perception research, showing that human judgments can be accounted for by an ideal observer using knowledge of environmental statistics (e.g., Geisler, Perry, Super, & Gallogly, 2001).

Other analyses have more directly compared the consequences of using different priors. For example, Oaksford and Chater (1994) showed that apparent irrationalities in Wason’s (1966) card selection task could be explained as rational Bayesian inference, but only under the assumption that the predicates involved are rare – that is, true for only a small subset of objects. McKenzie and Mikkelsen (2000) made a similar point in the context of covariation assessment, arguing that longstanding results concerning the relative weight assigned to different kinds of evidence could be explained as the result of a belief that predicates are rare. A number of subsequent studies have tested these explanations, examining the consequence of manipulating the rarity of predicates for people’s judgments in these tasks (Oaksford, Chater, Grainger, & Larkin, 1997; Oaksford,

Chater, & Grainger, 1999; Oaksford & Chater, 2001; McKenzie & Mikkelsen, 2007).

If the goal is to identify people's priors, then inferences from small amounts of data have the potential to be most informative, as they will be most strongly influenced by priors. Griffiths and Tenenbaum (2006) used this approach to show that people have remarkably accurate knowledge of the distributions associated with human lifespans, and use this knowledge in the way prescribed by Bayes' rule (see Figure 7). People were asked to make a prediction based on a single piece of data. For example, if you were assessing the prospects of a 60-year-old man, how much longer would you expect him to live? If you heard that a movie had made \$40 million so far at the box office, how much would you expect it to make in total? The resulting predictions should be based on people's prior knowledge about these domains, and take characteristic forms when the quantities in question follow different distributions. Studying everyday inductive leaps of this kind provides the opportunity to explore the constraints on human inductive inferences using naturalistic tasks that are easily reproduced in a laboratory setting. The fact that these inferences are made from only a small amount of data means that we can measure the effects of prior knowledge directly.

By examining inductive inferences in domains where we know the statistics that should inform the prior, we can examine how well people's prior knowledge is calibrated to their environment. In other settings, we can use the Bayesian framework to explore the consequences of using different kinds of prior knowledge, and work backward from people's judgments to identify possible constraints. While most work in this area has focused on memory and probabilistic reasoning, a similar methodology can be applied to other kinds of inductive problems. For example, one way to explore inductive biases in language learning is to examine what learners with different priors can extract from corpora of child-directed speech. An analysis of this kind for the problem of word segmentation – learning the words that appear in continuous speech – suggests that assumptions about

the nature of the interaction between words can have a significant effect on segmentation performance (Goldwater, Griffiths, & Johnson, 2006). Extending these analyses and connecting them to empirical research in language acquisition is a promising direction for future research.

Designing experiments to reveal biases

The second approach to studying inductive biases involves developing laboratory methods specifically designed to provide information about the constraints that guide people’s inductive inferences. This approach takes inspiration less from the tradition of computational models of cognition, and more from methods like *mechanism design* in theoretical economics (Hurwicz, 1973). The basic idea behind mechanism design is to structure an interaction between agents in a way that provides them with incentives to produce a particular kind of behavior. For example, an auction can be structured so that every bidder should bid the true value that they assign to the prize, revealing information that might be concealed in a more conventional auction (Vickrey, 1961). Mechanism design proceeds from the assumption that the agents are rational utility maximizers, making it possible to predict their behavior in different situations. By analogy, assuming that learners are Bayesian agents can allow us to design tasks intended to reveal their inductive biases.

One line of work that uses this idea grew out of analyzing the properties of a class of models of language evolution by “iterated learning” (Kirby, 2001). These models are based on the idea that every speaker of a language learns that language from another speaker, who had to learn it from somebody else in turn. Formally, we can imagine a sequence of learners, each of whom receives data from the previous learner, forms a hypothesis from those data, and then generates new data that are passed to the next learner. Griffiths and Kalish (2005, in press) analyzed the consequences of iterated

learning in the case where learners form hypotheses by applying Bayes' rule and then sampling a hypothesis from the the resulting posterior distribution, and generate data by sampling from the likelihood function associated with that hypothesis. In this case, the probability that a learner selects a particular hypothesis will converge to the prior probability that the learners assign to that hypothesis as the length of the sequence increases. This result has significant implications for the connection between the biases of individual learners and linguistic universals (Griffiths & Kalish, in press; Kirby, Dowman, & Griffiths, 2007), but goes beyond language evolution, applying to iterated learning with any kind of hypotheses and data. The fact that iterated learning converges to the prior suggests a simple procedure for explore human inductive biases: implement iterated learning in the laboratory with human learners, and examine which hypotheses survive.

A series of recent experiments have provided results suggesting that iterated learning can be used to reveal the inductive biases of human learners. Figure 8 shows the results of an experiment in iterated function learning, where the hypotheses concerned the form of a function causally relating two variables, and the data were values of these variables (Kalish, Griffiths, & Lewandowsky, in press). In this experiment, we ran 32 “families” of learners (i.e. sequences of nine learners, each learning from the previous learner) and found that iterated learning had remarkably consistent consequences, with 28 of the 32 families producing a positive linear function by the end of the experiment (the other four were producing negative linear functions). This is consistent with previous experiments in function learning suggesting that people have an inductive bias favoring linear functions with a positive slope (Brehmer, 1971, 1974; Busemeyer et al., 1997). We have subsequently obtained similar results using an iterated generalization task based on the category structures studied by Shepard, Hovland, and Jenkins (1961), finding that the structures people find easiest to learn quickly dominate (Griffiths, Christian, & Kalish, 2006).

Again, this approach is in its infancy, but has promise as a way to discover the

inductive biases of human learners. Its main advantage over fitting models assuming different priors is that it is nonparametric – it does not require any assumptions about the form of the prior on the part of the modeler. This is important in domains where the stimuli or hypotheses are complex, making it hard to formally describe the prior in a way that would not be an oversimplification. The results shown in Figure 8 illustrate this: it is clear that the prior is one that favors positive linear functions, even though we might not know exactly how to define a prior over the space of functions that people are considering. If we want to explore inductive biases for aspects of natural languages, the structure of categories, and networks of causal relationships, this capacity to deal with complexity could prove extremely valuable.

Conclusion

I began this chapter with the question of what makes people so good at generalization. The example of function learning shows that one simple answer – richer hypothesis spaces – is not sufficient to produce good generalization. The bias-variance tradeoff helps to explain why this is, showing that generalization errors are affected by both the bias of a learning algorithm and the variability in its answers across datasets, and richer hypothesis spaces reduce bias at the cost of increasing variance. Richer hypothesis spaces need to be complemented by inductive biases that constrain inferences from small amounts of data. Bayes’ rule provides a way to state these biases – through the prior distribution over hypotheses – and to define graded degrees of plausibility over rich hypothesis spaces. Analyzing people’s inferences in these terms gives us at least two ways to explore human inductive biases: by comparing Bayesian models with different priors to human behavior, and by designing experiments explicitly intended to reveal the priors of learners.

While my focus in this chapter has been on revealing inductive biases, a critical

question that this raises is where those biases come from. The Bayesian framework provides a natural answer to this question, through a class of models known as hierarchical Bayesian models (Tenenbaum, Griffiths, & Kemp, 2006). The basic idea behind a hierarchical Bayesian model is that the knowledge that we draw upon in solving inductive problems is represented at many levels, and that Bayesian inference can be applied at any of these levels. To the extent that there are principles that are relevant to many inductive inferences within a domain – such as the way that words of different syntactic classes are used, the form that categories usually take, or the nature of the mechanisms underlying particular causal relationships – these principles can be abstracted from experience and used to constrain future inferences. Essentially, people can learn the prior distributions that characterize their environment, and use this knowledge to improve their inferences. By using Bayesian models to chart the inductive biases that guide people’s inferences in different domains, we can begin to explore whether this kind of approach can also account for their origins.

References

- Anderson, J. R. (1990). *The adaptive character of thought*. Hillsdale, NJ: Erlbaum.
- Anderson, J. R., & Milson, R. (1989). Human memory: An adaptive perspective. *Psychological Review*, 96, 703-719.
- Anderson, J. R., & Schooler, L. J. (1991). Reflections of the environment in memory. *Psychological Science*, 2, 396-408.
- Bishop, C. M. (2006). *Pattern recognition and machine learning*. New York: Springer.
- Brehmer, B. (1971). Subjects' ability to use functional rules. *Psychonomic Science*, 24, 259-260.
- Brehmer, B. (1974). Hypotheses about relations between scaled variables in the learning of probabilistic inference tasks. *Organizational Behavior and Human Decision Processes*, 11, 1-27.
- Busmeyer, 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* (p. 405-437). Cambridge: MIT Press.
- Chomsky, N. (1965). *Aspects of the theory of syntax*. Cambridge, MA: MIT Press.
- Cox, R. T. (1961). *The algebra of probable inference*. Baltimore, MD: Johns Hopkins University Press.
- DeLosh, E. L., Busmeyer, J. R., & McDaniel, M. A. (1997). Extrapolation: The sine qua non of abstraction in function learning. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 23, 968-986.
- Domingos, P. (2000). A unified bias-variance decomposition and its applications. In *Proceedings of the seventeenth international conference on machine learning* (p. 231-238). Stanford, CA: Morgan Kaufmann.
- Elman, J. L., Bates, E. A., Johnson, M. H., Karmiloff-Smith, A., Parisi, D., & Plunkett, K. (1996). *Rethinking innateness: A connectionist perspective*. Cambridge, MA: MIT Press.
- Fodor, J. A. (1980). On the impossibility of acquiring more powerful structures. In

- M. Piattelli-Palmarini (Ed.), *Language and learning: The debate between Jean Piaget and Noam Chomsky*. London: Routledge and Kegan Paul.
- Friedman, J. H. (1997). On bias, variance, 0/1 loss, and the curse-of-dimensionality. *Data Mining and Knowledge Discovery*, 1, 55-77.
- Geisler, W. S., Perry, J. S., Super, B. J., & Gallogly, D. P. (2001). Edge co-occurrence in natural images predicts contour grouping performance. *Vision Research*, 41, 711-724.
- Geman, S., Bienenstock, E., & Doursat, R. (1992). Neural networks and the bias-variance dilemma. *Neural Computation*, 4, 1-58.
- Goldwater, S., Griffiths, T. L., & Johnson, M. (2006). Contextual dependencies in unsupervised word segmentation. In *Proceedings of Coling/ACL 2006*.
- Griffiths, T. L., Christian, B. R., & Kalish, M. L. (2006). Revealing priors on category structures through iterated learning. In *Proceedings of the 28th Annual Conference of the Cognitive Science Society*. Mahwah, NJ: Erlbaum.
- Griffiths, T. L., & Kalish, M. L. (2005). A Bayesian view of language evolution by iterated learning. In B. G. Bara, L. Barsalou, & M. Bucciarelli (Eds.), *Proceedings of the Twenty-Seventh Annual Conference of the Cognitive Science Society* (p. 827-832). Mahwah, NJ: Erlbaum.
- Griffiths, T. L., & Kalish, M. L. (in press). A Bayesian view of language evolution by iterated learning. *Cognitive Science*.
- Griffiths, T. L., & Tenenbaum, J. B. (2006). Optimal predictions in everyday cognition. *Psychological Science*, 17, 767-773.
- Hastie, T., Tibshirani, R., & Friedman, J. (2001). *The elements of statistical learning: Data mining, inference, and prediction*. New York: Springer.
- Helmholtz, H. v. (1866/1962). Concerning the perceptions in general. In J. P. C. Southall (Ed.), *Treatise on physiological optics* (Vol. 3). New York: Dover.
- Hume, D. (1739/1978). *A treatise of human nature*. Oxford: Oxford University Press.
- Hurwicz, L. (1973). The design of mechanisms for resource allocation. *American Economic Review*, 63, 1-30.

- Inhelder, B., & Piaget, J. (1958). *The growth of logical thinking from childhood to adolescence*. London: Routledge and Kegan Paul.
- James, G., & Hastie, T. (1997). *Generalizations of the bias/variance decomposition for prediction error* (Tech. Rep.). Stanford, CA: Department of Statistics, Stanford University.
- Jaynes, E. T. (2003). *Probability theory: The logic of science*. Cambridge: Cambridge University Press.
- Kalish, M., Lewandowsky, S., & Kruschke, J. (2004). Population of linear experts: Knowledge partitioning and function learning. *Psychological Review*, 111, 1072-1099.
- Kalish, M. L., Griffiths, T. L., & Lewandowsky, S. (in press). Iterated learning: Intergenerational knowledge transmission reveals inductive biases. *Psychonomic Bulletin and Review*.
- Kant, I. (1781/1964). *Critique of pure reason*. London: Macmillan.
- Kearns, M., & Vazirani, U. (1994). *An introduction to computational learning theory*. Cambridge, MA: MIT Press.
- Kirby, S. (2001). Spontaneous evolution of linguistic structure: An iterated learning model of the emergence of regularity and irregularity. *IEEE Journal of Evolutionary Computation*, 5, 102-110.
- Kirby, S., Dowman, M., & Griffiths, T. L. (2007). Innateness and culture in the evolution of language. *Proceedings of the National Academy of Sciences*, 104, 5241-5245.
- MacKay, D. (1995). Probable networks and plausible predictions - a review of practical bayesian methods for supervised neural networks. *Network: Computation in Neural Systems*, 6, 469-505.
- Mackay, D. J. C. (2003). *Information theory, inference, and learning algorithms*. Cambridge: Cambridge University Press.
- Marr, D. (1982). *Vision*. San Francisco, CA: W. H. Freeman.
- McKenzie, C. R. M., & Mikkelsen, L. A. (2000). The psychological side of Hempel's paradox of confirmation. *Psychonomic Bulletin and Review*, 7, 360-366.
- McKenzie, C. R. M., & Mikkelsen, L. A. (2007). A Bayesian view of covariation assessment. *Cognitive Psychology*, 54, 33-61.

- Minsky, M. L., & Papert, S. A. (1969). *Perceptrons*. Cambridge, MA: MIT Press.
- Neal, R. M. (1992). Connectionist learning of belief networks. *Artificial Intelligence*, 56, 71-113.
- Oaksford, M., & Chater, N. (1994). A rational analysis of the selection task as optimal data selection. *Psychological Review*, 101, 608-631.
- Oaksford, M., & Chater, N. (2001). The probabilistic approach to human reasoning. *Trends in Cognitive Sciences*, 5, 349-357.
- Oaksford, M., Chater, N., & Grainger, R. (1999). Probabilistic effects in data selection. *Thinking and Reasoning*, 5, 193-243.
- Oaksford, M., Chater, N., Grainger, R., & Larkin, J. (1997). Optimal data selection in the reduced array selection task (RAST). *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 23, 441-458.
- Robert, C. P. (1994). *The Bayesian choice: A decision-theoretic motivation*. New York: Springer.
- Rogers, T., & McClelland, J. (2004). *Semantic cognition: A parallel distributed processing approach*. Cambridge, MA: MIT Press.
- Rosenblatt, F. (1958). The Perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*, 65, 386-408.
- Rumelhart, D., & McClelland, J. (1986). On learning the past tenses of English verbs. In J. McClelland, D. Rumelhart, & the PDP research group (Eds.), *Parallel distributed processing: Explorations in the microstructure of cognition* (Vol. 2). Cambridge, MA: MIT Press.
- Rumelhart, D. E., Hinton, G. E., & Wilson, R. J. (1986). Learning representations by back-propagating errors. *Nature*, 323, 533-536.
- Shepard, R. N. (1987). Towards a universal law of generalization for psychological science. *Science*, 237, 1317-1323.
- Shepard, R. N., Hovland, C. I., & Jenkins, H. M. (1961). Learning and memorization of classifications. *Psychological Monographs*, 75. (13, Whole No. 517)
- Tenenbaum, J. B., & Griffiths, T. L. (2001). Generalization, similarity, and Bayesian inference. *Behavioral and Brain Sciences*, 24, 629-641.

- Tenenbaum, J. B., Griffiths, T. L., & Kemp, C. (2006). Theory-based Bayesian models of inductive learning and reasoning. *Trends in Cognitive Science*, 10, 309-318.
- Tibshirani, R. (1996). *Bias, variance, and prediction error for classification rules* (Tech. Rep.). Toronto, ON: Statistics Department, University of Toronto.
- Vapnik, V. N. (1995). *The nature of statistical learning theory*. New York: Springer.
- Vickrey, W. (1961). Counterspeculation, auctions, and competitive sealed tenders. *Journal of Finance*, 16, 8-27.
- Wason, P. C. (1966). Reasoning. In B. Foss (Ed.), *New horizons in psychology*. Harmondsworth, UK: Penguin.

Author Note

This work was supported by grants BCS-0631518 and BCS-0704034 from the National Science Foundation and FA9550-07-1-0351 from the Air Force Office of Scientific Research.

Footnotes

¹Despite the intimidating description as searching through an infinite hypothesis space to find the best-fitting function, this reduces to the familiar problem of linear regression, where appropriate powers of x are used as predictors, and has a simple closed-form solution (e.g., Bishop, 2006).

²Again, while this might seem complicated, it reduces to a simple algebra problem (Bishop, 2006). It is also possible to compute the predictions of a Bayesian model when averaging over the posterior distribution, which provides another way of controlling complexity and leads to some sophisticated solutions to this kind of function learning problem (Mackay, 2003).

Figure Captions

Figure 1. A schematic example of a generalization problem. Given a set of (x, y) pairs, indicated with solid black points, the learner has to predict the value of y for a new value of x , indicated by the dotted line. This can be done by inferring the function relating x and y .

Figure 2. Consequences of selecting function minimizing squared error to observed data, using hypothesis spaces of (a) linear functions (b) quadratic functions and (c) 8th degree polynomials.

Figure 3. Bias and variance. Each panel shows the results of applying a learning algorithm to 100 randomly generated sets of 10 points. The grey lines are the model predictions, the black line is the average of these predictions, and the dotted line is the true function $f(x)$ from which the data were generated. (a) A hypothesis space of linear functions results in a high bias, with the average function differing significantly from the true function, and a moderate amount of variance around that average function. (b) A hypothesis space of quadratic functions results in both low bias and low variance. (c) A hypothesis space of 8th degree polynomials results in low bias, with the average function being close to the truth, but an enormous amount of variance, with predictions depending strongly on the specifics of the data.

Figure 4. Adding more data reduces variance, for all three learning algorithms. The predictions shown here are from 100 sets of 100 randomly generated points. (a) The linear hypothesis space still results in a high bias, but hypothesis spaces of (b) quadratic functions and (c) 8th degree polynomials now result in more similar predictions.

Figure 5. Samples of functions from different priors on the hypothesis space of 8th degree polynomials. All priors were defined by assuming that the parameter θ_k , being the

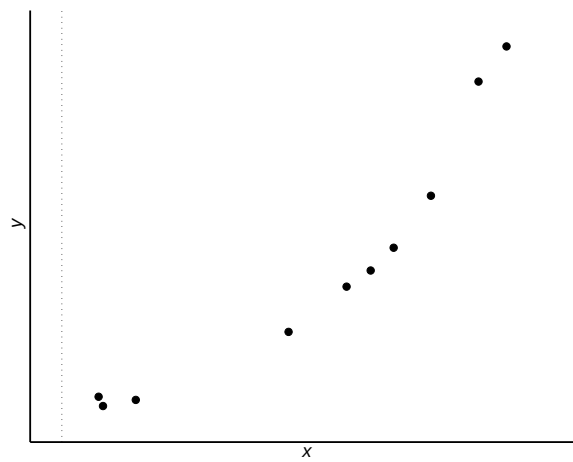
coefficient of x^k , followed a Gaussian distribution with mean zero and standard deviation $\alpha\beta^k$. Varying β varies the extent to which coefficients are expected to be small as k increases. The scale is the same as that used in the other figures exploring this example. (a) $\beta = 0.3$ results in a prior in which functions can have relatively high curvature. (b) $\beta = 0.1$ penalizes higher powers of x more strongly, resulting in less curved functions. (c) $\beta = 0.01$ favors functions that have no curvature and little slope, giving extremely small values to all θ_k except θ_0 .

Figure 6. Consequences of using different priors in making predictions. The final predictions are a compromise between the prior and the likelihood. The priors are those used to generate the functions that appear in the corresponding panels in the previous figure. (a) With a prior that favors functions with high curvature ($\beta = 0.3$), a highly curved function is selected. (b) A prior favoring functions that have a little curvature ($\beta = 0.1$) does well in reproducing the true function, having inductive biases appropriate for the problem at hand. (c) A prior favoring functions with no curvature and little slope does poorly, although the predictions it makes based on the data are notably different from the functions sampled from the prior, having a non-negligible slope. With more data, this prior would eventually be overwhelmed, allowing the learner to make more accurate predictions.

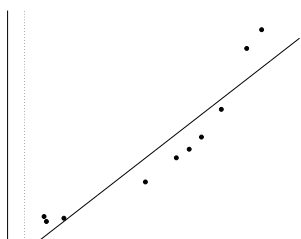
Figure 7. People use different prior distributions when making predictions about different quantities. The upper panels show the empirical distribution of the total duration or extent, t_{total} , for five different everyday phenomena. The values of t_{total} are the hypotheses h to be evaluated, and these distributions are the appropriate priors. The first two distributions (human lifespans and movie runtimes) are approximately Gaussian, the next two (the gross of movies and the length of poems) are approximately power-law, and the last (length of terms of members of the U.S. House of Representatives) is approximately

Erlang. Best-fitting parametric distributions are plotted in black. In the lower panels, black dots show subjects' median predictions for t_{total} when given a single observed sample t of a duration or extent in each of five domains (the data d used when applying Bayes' rule). Judgments are consistent with Bayesian predictions using the empirical prior distribution shown in the upper panel (grey lines), and the best-fitting parametric prior (black lines). Predictions based on a single uninformative prior (dotted lines) are not consistent with these judgments. Adapted from Griffiths and Tenenbaum (2006).

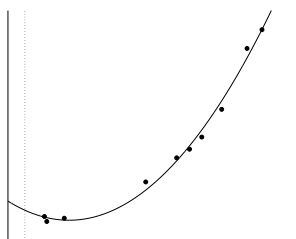
Figure 8. Iterated learning as a method for identifying inductive biases. The leftmost panel in each row shows the set of samples from a function seen by the first learner in a sequence. The (x, y) pairs were each presented as the lengths of two bars on a computer screen. During training, participants predicted the length of one bar from the other before seeing the second bar as feedback. The second panel in each row show the predictions produced by the first learner in a test phase where participants made predictions of y for a range of x values without receiving feedback. These predictions were then used as the training data for the second learner, who produced the predictions shown in the third column. The other panels show the data produced by each generation of learners, each being trained from the predictions produced by the previous learner. Each row shows a single sequence of nine learners, drawn at random from eight “families” of learners run with the same initial data. The rows differ in the functions used to generate the data shown to the first subject. In each case, iterated learning quickly converges to a linear function with positive slope, consistent with findings indicating that human learners are biased towards this kind of function. Adapted from Kalish, Griffiths, and Lewandowsky (in press).



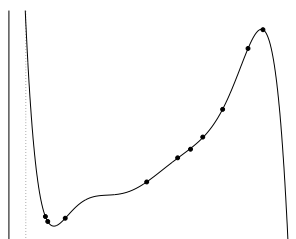
A



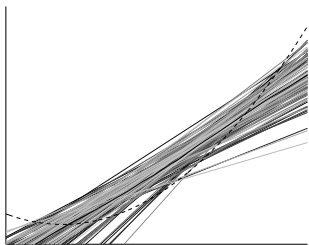
B



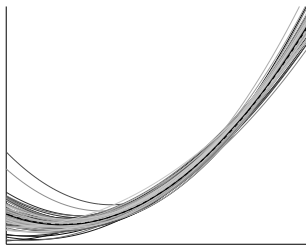
C



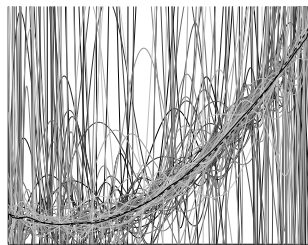
A



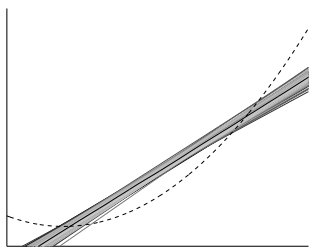
B



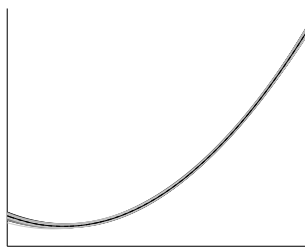
C



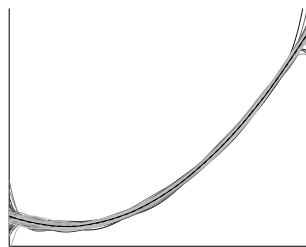
A



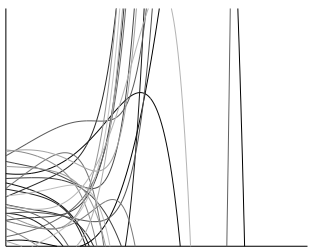
B



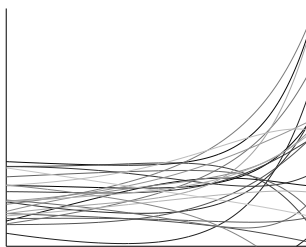
C



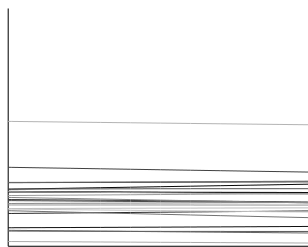
A



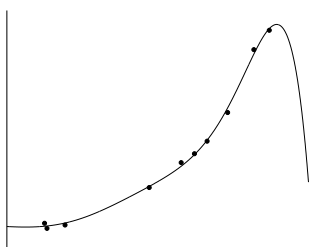
B



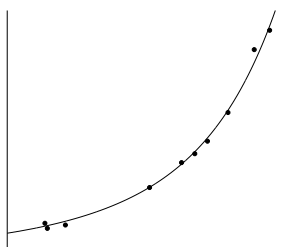
C



A



B



C

