Part I

Introduction to Bayesian Methods

Cambridge University Press & Assessment
978-1-108-83499-5 — Bayesian Methods for Interaction and Design
Edited by J.H. Williamson , A. Oulasvirta , P.O. Kristensson , N. Banovic
Excerpt
More Information

1
An Introduction to Bayesian Methods for Interaction Design

John H. Williamson

Abstract

Bayesian modelling has much to offer those working in human–computer interaction, but many of the concepts are alien. This chapter introduces Bayesian modelling in interaction design. The chapter outlines the philosophical stance that sets Bayesian approaches apart, as well as a light introduction to the nomenclature and computational and mathematical machinery. We discuss specific models of relevance to interaction, including probabilistic filtering, non-parametric Bayesian inference, approximate Bayesian computation and belief networks. We include a worked example of a Fitts' law modelling task from a Bayesian perspective, applying Bayesian linear regression via a probabilistic program. We identify five distinct facets of Bayesian interaction: probabilistic interaction in the control loop; Bayesian optimisation at design time; analysis of empirical results with Bayesian statistics; visualisation and interaction with Bayesian models; and Bayesian cognitive modelling of users. We conclude with a discussion of the pros and cons of Bayesian approaches, the ethical implications therein and suggestions for further reading.

1.1 Introduction

We assume that most readers will be coming to this text from an interaction design background and are looking to expand their knowledge of Bayesian approaches, and we started from this framing when structuring this chapter. Some readers may be coming the other way, from a Bayesian statistics background to interaction design. These readers will find interesting problems and applications of statistical methods in interaction design.

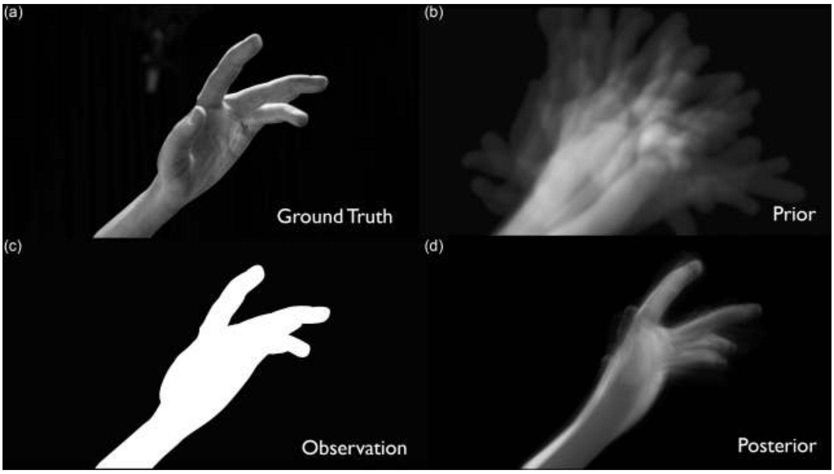

Figure 1.1 Which hand pose, generated the silhouette (c)? We cannot resolve a unique answer from this **observation**. Instead, we can start from a **prior** set (b) of viable hand poses and infer the distribution of *likely* poses given the observed silhouette (the **posterior** belief about poses) (d). **Uncertainty** about the pose is represented in the prior and reduced (but still present) in the posterior, inferred after observing the silhouette.

This book discusses how Bayesian approaches can be used to build models of human interactions with machines. Modelling is the cornerstone of good science, and actionable computational models of interactive systems are the basis of *computational interaction* [42]. A Bayesian approach makes *uncertainty* a first-class element of a model and provides the technical means to reason about uncertain beliefs computationally.

Human–computer interaction is rife with uncertainty. Explicitly modelling uncertainty is a bountiful path to better models and ultimately better interactive systems. The Bayesian world view gives an elegant and compelling basis to reason about the problems we face in interaction, and it comes with a superbly equipped wardrobe of computational tools to apply the theory. Bayesian approaches can be engaged across the whole spectrum of interaction, from the most fine-grained, pixel-level modelling of a pointer to questions about the social impact of always-on augmented reality. Everyone involved in interaction design, at every level, can benefit from these ideas. Thinking about interaction in Bayesian terms can be a refreshing perspective to re-examine old problems.

And, as this book illustrates, it can also be transformational in practically delivering the human–computer interactions of the future.

A Note on This Chapter
This chapter is intended to be a high-level look at Bayesian approaches *from the point of view of an interaction designer*. Where possible, I have omitted mathematical terminology; the Appendix of the book gives a short introduction to standard terminology and notation. In some places I have provided skeleton code in Python. This is not intended to be executable, but to be a readable way to formalise the concepts for a computer scientist audience, and should be interpretable even if you are not familiar with Python. All data and examples are synthetic.

The chapter is structured as follows:

- A short introduction to Bayesian inference for the unfamiliar.
- A high-level discussion of the distinctive aspects of Bayesian modelling.
- A detailed worked example of Bayesian modelling in an HCI problem.
- A short summary of Bayesian algorithms and techniques particularly relevant to interaction, including approximate Bayesian computation, Bayesian optimisation and probabilistic filtering.
- A discussion of the important facets of Bayesian interaction design.
- Finally, a reflection on the implications of these ideas as well as recommendations for further reading.

1.1.1 What Are Bayesian Methods?

Bayesian methods is a broad term. In this book, the ideas are linked by the fundamental property of representing *uncertain belief* using probability distributions, and updating those beliefs with evidence. The underpinning of probability theory puts this on a firm theoretical basis, but the concept is simple: we represent what we know about the specific aspects of the world with a **distribution** that tells us how likely possible configurations of the world are, and then refine belief about these possibilities with data. We can repeat this process as required, accumulating evidence and reducing our uncertainty. In its simplest form, this boils down to simply counting potential configurations of the world, then adjusting those counts to be compatible with some observed data. This idea has been become vastly more practical as computational power has surged, making the efficient ‘counting of possibilities’ a feasible task for complex problems.

Why Model at All?

‘I am never content until I have constructed a mechanical model of the subject I am studying. If I succeed in making one, I understand, otherwise I do not.’ – Lord Kelvin, *Baltimore Lectures on Molecular Dynamics and the Wave Theory of Light.* 1884
For the twenty-first century, replace ‘mechanical’ with ‘computational’.

Modelling creates a simplified version of a problem that we can more easily manipulate, and could be mathematical, computational or physical in nature. Good science depends on good models. Models can be shared, criticised and re-used. Fields of study where there is healthy exchange of models can ‘ratchet’ constructively, one investigation feeding into the next. In interaction design, modelling has been relatively weak. When models have been used, they have often been descriptive in nature rather than causal. One of the motivations for a Bayesian approach is in the adoption of statistical models that are less about describing or predicting the superficial future state of the world and more about predicting the underlying state of the world. The other motivation is to build and work with models that properly account for uncertainty.

We can consider the relative virtues of models, in terms of their authenticity to the real-world phenomena, their complexity or their mathematical convenience. However, for the purposes of human–computer interaction, several virtues are especially relevant:

- Models that are *generative* and can be executed in a computer simulation to produce synthetic data.
- Models that are *computational* and can be manipulated, transformed and validated algorithmically; for example, written as programs.
- Models that are conveniently *parameterisable* and ideally have parameters that are meaningful and interpretable.
- Models that are *causal* and describe the underlying origins of phenomena rather than predict the manifestations of phenomena.
- Models that preserve and propagate *uncertainty*.
- Models that fit well with software engineering practices to deploy them, whether embedded in an interaction loop, in design tools or in analyses of evaluations.

1.1.2 What Is Distinctive about *Bayesian* Modelling?

Bayesian modelling has several salient consequences:

- We can often directly use *simulators* of the process we believe to be generating the world that we observe, instead of relying on abstract, fixed models that can be difficult to shoehorn into interaction problems. For example, we might be able to use a detailed, agent-based simulation of pedestrian movement rather than a standard regression model.
- We reason from belief to evidence, not the other way around. This subtle difference means that we have a way to easily fuse information from many sources. This can range from sensor fusion in an inertial measuring unit to meta-reviews of surveys in the literature.
- We have a *universal* approach to solving problems that gives us a simple and consistent way to formulate questions and reason our way to answers.
- We also have a *universal* language with which to exchange and combine information: the probability distribution. Want to plug a language model into a gesture recogniser? No problem – exchange probability distributions.
- That same freedom and flexibility to model, and the need to represent distributions rather than values, implies technical difficulties. The devil is in the details.

1.1.3 How Is This Relevant to Interaction Design?

Everything we do with interactive systems has substantial *uncertainty* inherent in it. We don't know who our users are. We don't know what they want, or how they behave, or even how they tend to move. We don't know where they are, or in what context they are operating. The evidence that we *can* acquire is typically weakly informative and often indirectly related to problems we wish to address. This extends across all levels, from tightly closed control loops to design-time questions or retrospective evaluations. For example:

- Do a user's pointing movements indicate an intention to press button A or B?
- Is now a good moment to pop up a dialog?
- How many touch interaction events will happen in the next 500 ms?
- How tired is the user right now?
- Is it better to allocate a shorter keyboard shortcut to Save or for Refresh?
- Does adding spring-back to a scrolling menu increase or decrease user stress?
- Which volatility visualisation strategy helps users make more rational decisions?
- Is this interactive system more or less likely to polarise society?

We typically have at least partial models of how the human world works: from psychology, physiology, sociology or physics. Good interaction design behooves us to take advantage of all the modelling we can derive from the research of others. Being able to slot together models from disparate fields is essential to advance science. The Bayesian approach of formally incorporating knowledge as priors can make this a consistent and reasonable thing to do.

We are in the business of interacting with computers – so computational methods are universally available to us. We care little about methods that are efficient to be hand-solved algebraically. The blossoming field of *computational* Bayesian statistics means that we can realistically embed Bayesian models in interactive systems or use them to design and analyse empirical studies at the push of a button. We have problems where it is important to pool and fuse information, whether in low-level fusion of sensor streams or in combining survey data from multiple studies. We have fast CPUs and GPUs and software libraries that subsume the fiddly details of inference.

1.1.4 What Does This Give Us?

Why might we consider Bayesian approaches?

- Taming uncertainty by representing and manipulating it grants us robustness, whether this is robustness within a control loop or in the interpretation of the evaluation of a system. Represented uncertainty *regularises* predictions and avoids making extreme inferences based on limited data.
- We explicitly and precisely model *prior beliefs*. This allows knowledge to be encoded, inspected and shared, whether among software components or among researchers.
- A focus on generative models leads us to model constructively, to build models that synthesise what we expect to observe. These models can be strikingly more insightful than models that seek to summarise or describe what we have observed.
- Bayesian inference makes it realistic to fuse information from many sources without ad hoc tricks, and a principled way to deal with missing data and imputation.

Most of all, Bayesian approaches give us a new perspective from which to garner insight into problems of interaction, supported by a bedrock of mathematical and computational tools.

1.1.5 Is This Just for Statistical Analysis?

Bayesian methods are a powerful tool for empirical analysis, and historically Bayesian methods have been used for statistical analyses of the type familiar to HCI researchers in user evaluations. But that is not their only role in interaction design, and arguably not even the most important role they can play. Bayesian methods can be used directly within the control loop as a way of robustly tracking states (for example, using probabilistic filtering). Bayesian optimisation makes it possible to optimise systems that are hard and expensive to measure, such as subjective responses to UI layouts. Bayesian ideas can change the way we think about how users make sense of interfaces, how we should represent uncertainty to them and how we should predict users' cognitive processes.

1.2 A Short Tutorial

Terminology We will use a number of specific terms in the rest of the chapter:

- **model** a simplified representation of a problem that has some parts that can vary. Our models will always be implemented as computer programs.
- **parameter** one variable in a model that partially determines how a model operates.
- **configuration** a collection of parameter values that fully specifies a specific instantiation of a model.
- **observations** values which we observe, i.e. data.
- **probability** a number between 0 and 1 representing how much we believe something.
- **distribution** an assignment of probabilities to possible configurations, defining how likely each is.
- **sample** a specific value, drawn at random according to a probability distribution.
- **prior** a belief about the world before observing data, as a distribution over configurations.
- **posterior** a belief about the world after observing data, as a distribution over configurations.
- **likelihood** a belief about how likely observations are, given a configuration, as a distribution over possible observations.

We will also use the following notation for probability:

- $P(A)$ the probability that event A occurs.
- $P(A, B)$ the probability that both event A and event B occur together.
- $P(A|B)$ the probability that event A occurs, if we *know* that B occurs.

See the Appendix of this book for a more thorough explanation.

1.2.1 An Example of Bayesian Inference

Imagine we have three app variants deployed to a group of users, A, B and C. App A has two buttons on the splash screen, App B has four, and App C has nine. We get a log event that indicates that '3' in the splash screen was pressed, but not which app generated it. Which app was the user using, given this information (Figure 1.2)?

We have an unobserved **parameter** (which app is being used) and observed **evidence** (button 3 was pressed). Let us further assume there are 10 test users using the app: five using A, two using B and three using C. This gives us a **prior** belief about which app is being used (for example, if we knew nothing about the interaction, we expect it is 50% more likely that App C is being used than App B). We also need to assume a *model* of behaviour. We might assume a very simple model that users are equally likely to press any button – the **likelihood** of choosing any button is equal.

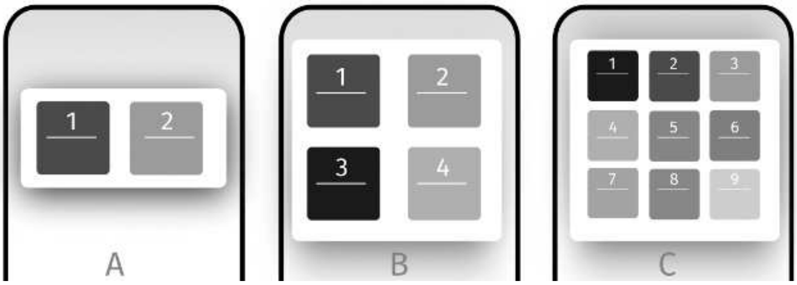

Figure 1.2 Which app was used? We know button 3 was pressed, but not on which app.

This is a problem of Bayesian updating (Figure 1.3); how to move from a prior probability distribution over apps to a posterior distribution over apps, having observed some evidence in the form of a button press.

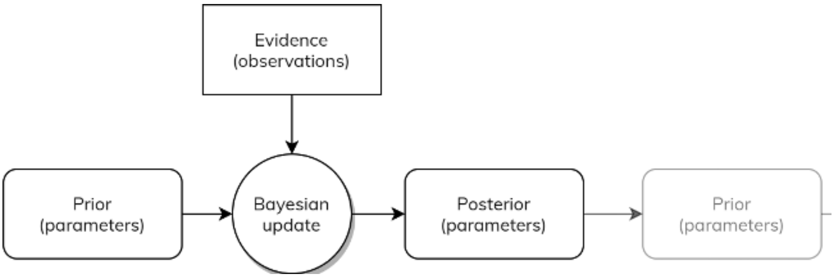

Figure 1.3 Bayesian inference takes a prior probability distribution, which represents beliefs about parameters, incorporates observed evidence and produces a posterior distribution which captures those beliefs that are compatible with the evidence. The posterior from one inference step can form the prior of a subsequent update.

How do we compute this? In this case, we can just count up the possibilities for each app, as shown in Figure 1.4.

We know that button 3 was logged, so:

- There is no possibility that the user was using App A, which has only two buttons.
- If they were using App B, 1/4 of the time they would press 3; and 1/9 of the time if using App C.

These numbers come directly from our assumption that buttons are pressed with equal likelihood, and so the **likelihoods** of seeing button 3 for each app are (A = 0, B = 1/4, C = 1/9). Given our prior knowledge about the number of apps in use, we can multiply these likelihoods by how likely we thought the particular app was *before* observing the '3'. This **prior** was (A = 5/10, B = 2/10, C = 3/10). This gives us: (A = 0 * 5/10, B = 1/4 * 2/10, C = 1/9 * 3/10) = (0, 1/20, 1/30). We can normalise this so it sums to 1 to make it a proper probability distribution: (0, 3/5, 2/5). This is the **posterior** distribution, the probability distribution revised to be compatible with the evidence. We *now* believe there is a 60% chance that App B was used and a 40% chance App C was used.