Lecture notes on Bayesian machine learning

What is BML, why use it, and how to implement it.

Rémi Bardenet and Julyan Arbel

Contents

I What is Bayesian machine learning?

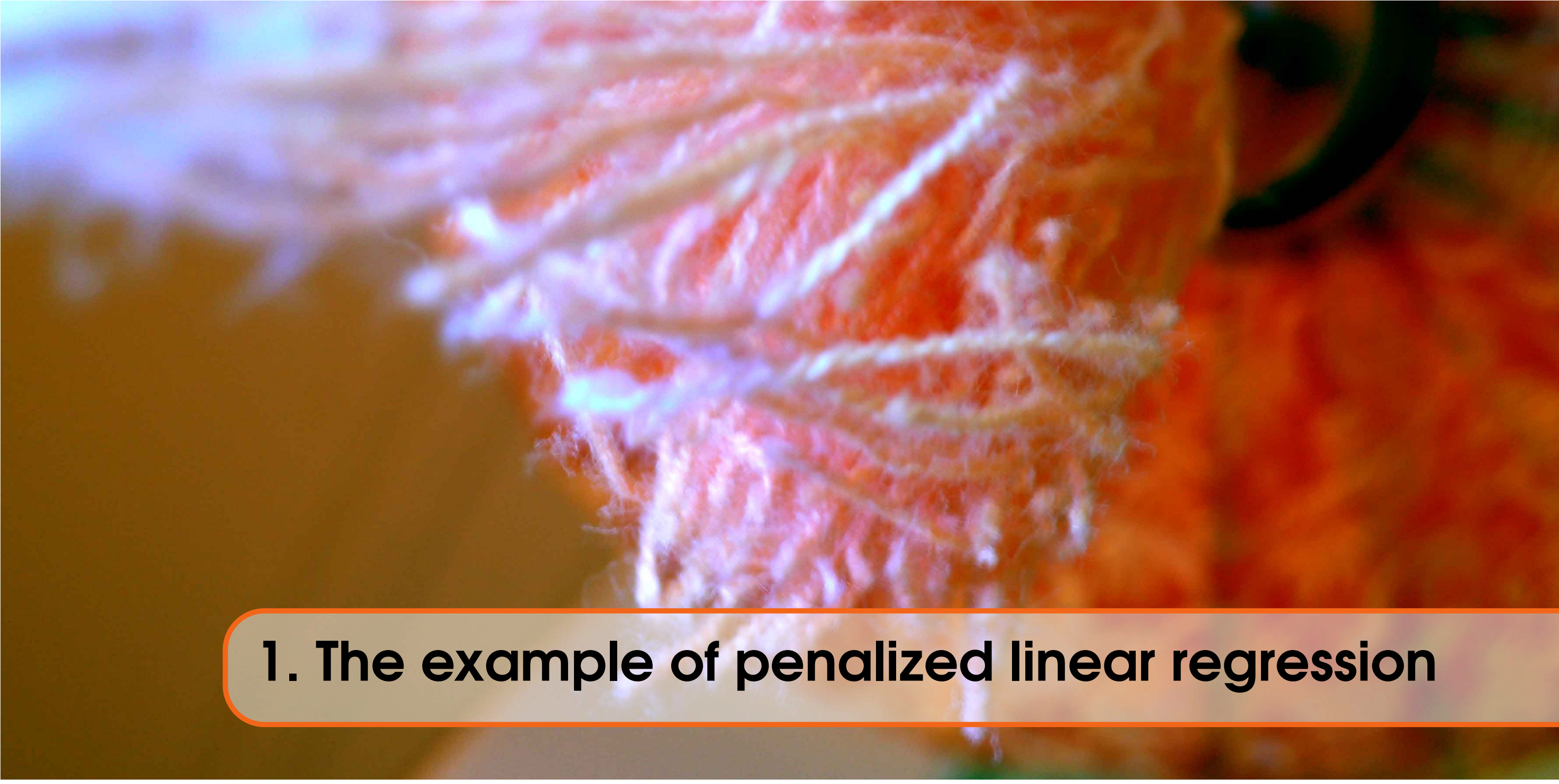
II Foundations

III Implementing Bayesian machine learning

IV Bayesian nonparametrics

V Bayesian deep learning

I What is Bayesian machine learning?



1. The example of penalized linear regression

In this chapter, we introduce different historical approaches to learning and inference on a running example, penalized linear regression. We shall exagerate the stances of some famous scientists, for the sake of illustration.

1.1 Fisher does linear regression

Say we have a data set (x_i, y_i), $1 \leq i \leq N$, where $x_i \in \mathbb{R}^d$ are the *features* and $y_i \in \mathbb{R}$ the *response*. We want to study the influence of features on the response, and we ask British statistician Ronald Fisher (1890–1962) for help. He recommends positing a simple statistical model, i.e. a parametrized collection of PDFs for $y|x$. Calling the parameter θ and abusively denoting the parametrized distributions by $y|x, \theta$, we posit

$$p(y_i|x_i, \theta) = \mathcal{N}(y_i|x_i^T\theta, \sigma^2), 1 \leq i \leq N,$$

i.i.d., with known σ for simplicity. To characterize the influence of features on the response, Fisher recommends that we estimate θ, along with a confidence interval to communicate our uncertainty. Fisher was the one to introduce the maximum likelihood estimator

$$\widehat{\theta}_{\mathrm{MLE}} \triangleq \arg\max_{\theta} p(\mathbf{y}|X, \theta) = \arg\max_{\theta} \prod_{i=1}^{N} \mathcal{N}(y_i|x_i^T\theta, \sigma^2).$$

Now

$$\log \prod_{i=1}^{N} \mathcal{N}(y_i|x_i^T\theta, \sigma^2) = \log \mathcal{N}(\mathbf{y}|X\theta, \sigma^2 I) \propto -\|\mathbf{y} - X\theta\|^2,$$

where the proportionality sign allows us to drop additive terms that do not depend on θ. This entails that $\widehat{\theta}_{\mathrm{MLE}}$ is nothing but the least squares estimator

$$\widehat{\theta}_{\mathrm{MLE}} = (X^T X)^{-1} X^T \mathbf{y},$$

where we assumed that X has full rank. There is nothing inherently good in using the MLE rather than another estimator, but it often has good *frequentist* properties, i.e., properties established by integrating over the posited data generation model. For instance, it is easy to prove the following.

Proposition 1.1.1 Under $\mathbf{y} \sim \mathscr{N}(X\theta, \sigma^2 I)$, and assuming X has full rank,

$$\widehat{\theta}_{\text{MLE}} \sim \mathscr{N}(\theta, \sigma^2 (X^T X)^{-1}). \tag{1.1}$$

Proof. Left as an exercise. ■

Fisher then argues that (1.1) gives you a wealth of properties that make $\widehat{\theta}_{\text{MLE}}$ an interesting estimator. For starters, $\widehat{\theta}_{\text{MLE}}$ is unbiased ($\mathbb{E}\widehat{\theta}_{\text{MLE}} = \theta$) and $\widehat{\theta}_{\text{MLE}} \to \theta$ in probability. Moreover, the (random) ellipsoid

$$\mathscr{E}_\alpha = \{\theta \in \mathbb{R}^d \text{ such that } \sigma^{-2}(\theta - \widehat{\theta}_{\text{MLE}})^T X^T X (\theta - \widehat{\theta}_{\text{MLE}}) \leq \alpha\}$$

contains θ with known probability $1 - \delta(\alpha)$. It thus makes sense, to Fisher, to find the smallest value $\alpha_{0.99}$ such that $1 - \delta(\alpha) \geq 0.99$, and report the *confidence region* $\mathscr{E}_{\alpha_{0.99}}$ to quantify his uncertainty about θ. The property that, under repetition of the data generation, the (random) ellipsoid $\mathscr{E}_{\alpha_{0.99}}$ contains θ about 99% of the time, is called *coverage*.

1.2 Wald does linear regression

Imagine that you had asked Hungarian-American statistician Abraham Wald (1902–1950) for help instead of Fisher. He would have argued that estimating θ, or outputing a region of $\mathbb{R}^d$ that encodes your uncertainty, are two different actions to be taken under uncertainty.

Consider estimation. Wald would ask you to specify the loss $L(\theta, \widehat{\theta})$ that you incur by estimating θ by $\widehat{\theta} \in \mathbb{R}^d$. You might answer $L_\alpha(\theta, \widehat{\theta}) = \|\theta - \widehat{\theta}\|^\alpha$ for some $\alpha > 0$. Wald would then say that the accuracy of an estimator $\widehat{\theta} = \widehat{\theta}(\mathbf{y})$ is characterized by its risk function

$$R(\cdot, \widehat{\theta}) : \theta \mapsto \mathbb{E}_{\mathbf{y}|X,\theta} L(\theta, \widehat{\theta}(\mathbf{y})). \tag{1.2}$$

If the risk function of an estimator is smaller than that of another estimator, we say that the former dominates the latter. Wald's minimum requirement for an estimator is that the estimator is *admissible*, i.e., that it is not dominated. Maybe surprisingly, the MLE for regression with the squared loss L_2 is *not* admissible! This is an extension by **TBC** of an important theorem known as the James-Stein theorem (**TBC**). We leave the original James-Stein theorem as Exercise **??**. We shall see an even simpler estimator that dominates the MLE in Section **??**.

So what should we do, according to Wald, if the MLE is no option? It would be natural to find an estimator with as small as possible a risk function. Unfortunately, the risk function being a function, many pairs of estimators are incomparable. Wald might recommend to sum up a risk function by a single number, and look, for instance, for an estimator minimizing the worst-case risk, a so-called *minimax* estimator

$$\widehat{\theta}_{\text{minimax}} \in \arg\inf_{\widehat{\theta}} \sup_{\theta} R(\theta, \widehat{\theta}). \tag{1.3}$$

Another solution to sum up the risk function is to integrate it against a measure for which the risk is integrable. This leads to an estimator that we call the *Bayes* estimator, anticipating over Chapter 2,

$$\widehat{\theta}_{\text{Bayes}} \in \arg\min_{\widehat{\theta}} \mathbb{E}_\theta R(\theta, \widehat{\theta}) = \mathbb{E}_\theta \mathbb{E}_{\mathbf{y}|X,\theta} R(\theta, \widehat{\theta}). \tag{1.4}$$

We now have a joint distribution over $\theta, \mathbf{y}$. As long as the loss $L(\theta, \widehat{\theta}(\mathbf{y}))$ is integratble w.r.t. this joint, the towering property of the expectation yields

$$\mathbb{E}_\theta \mathbb{E}_{\mathbf{y}|X,\theta} L(\theta, \widehat{\theta}) = \mathbb{E}_{\mathbf{y}} \mathbb{E}_{\theta|X,\mathbf{y}} L(\theta, \widehat{\theta}). \tag{1.5}$$

Since $\widehat{\theta}$ is only a function $\mathbf{y}$, minimizing (1.5) boils down to setting

$$\widehat{\theta}_{\text{Bayes}} = \widehat{\theta}_{\text{Bayes}}(\mathbf{y}) = \arg\inf_{\widehat{\theta}} \mathbb{E}_{\theta|X,\mathbf{y}} L(\theta, \widehat{\theta}).$$

For the squared loss $L = L_2$, the Bayes estimator is thus the mean of the *posterior* distribution, i.e. $\mathbb{E}_{\theta|X,\mathbf{y}}\theta$. For the one-loss $L = L_1$, the Bayes estimator is a generalized median of the same posterior distribution. In particular, note that the Bayes estimator depends on the loss function, and is not necessarily the posterior expectation.

1.3 Savage does linear regression

1.4 Hoerl and Kennard do linear regression

2. Maximizing expected utility

After formally defining decision problems, we show that basic machine learning problems such as classification, regression, model choice, etc. are decision problems. Then we introduce subjective expected utility, the single unique guideline of all Bayesians, and go over its consequences for ML decisions. Justifying subjective expected utility shall wait until Chapter II.

For this chapter, we mostly used two references. (Parmigiani and Inoue, 2009) focuses on ideas and is a formidable entry point to (Bayesian) decision theory, while (Schervish, 2012) is a great textbook-level reference for advanced reading, with mathematical details.

2.1 ML problems are decision problems

Formally, a decision problem is defined as

1. a set $\mathscr{S}$ of *states*. For technical reasons, we also require a σ-algebra $\Sigma_{\mathscr{S}}$ that makes $(\mathscr{S}, \Sigma_{\mathscr{S}})$ a Borel space (Schervish, 2012).
2. a set of *rewards* $\mathscr{R}$. We also require a σ-algebra $\Sigma_{\mathscr{R}}$, which should contain all singletons.
3. a set $\mathscr{A}$ of measurable functions from $\mathscr{S}$ to $\mathscr{R}$, called *actions*.
4. a utility function $u : \mathscr{R} \to \mathbb{R}$.

We think of states as encoding all information about the situation at hand. Actions are what we are tasked to choose, and picking action a while the situation is described by a given state s leads to reward $a(s)$. Note that we assume here that the same set of actions $\mathscr{A}$ remains available in every state s.[1]

Most basic ML problems are of this kind; see Figure 2.1 for a few classical formalizations. Note that most choices made in this table are arbitrary, and correspond to the simplest variant of each problem. For instance, in classification, one might penalize false negatives and false positives differently; see Exercises. Note also that, since Wald, 1950 and as done in Section **??**, it is also customary to define loss functions instead of utilities, as $L(a,s) = -u(a(s))$. At this point of the document, both notations are as expressive, and we shall use them interchangeably. The distinction

[1] In future versions of the course, we might make the framework more general, to include, e.g. Markov decision processes.

	$\mathscr{S}$	$\mathscr{R}$	$\mathscr{A}$	$u(r)$
Regression	$(\mathscr{X}\times\mathscr{Y})^{n+1}$	$\mathscr{Y}=\mathbb{R}$	$\{a_g : s\mapsto y-g(x;x_{1:n},y_{1:n})\}$	$-\|r\|^2$
Classification	$(\mathscr{X}\times\mathscr{Y})^{n+1}$	$\mathscr{Y}=\{0,1\}$	$\{a_g : s\mapsto y-g(x;x_{1:n},y_{1:n})\}$	$\mathbb{1}_{\{r=0\}}$
Point estimation	$\mathscr{Y}^n\times\Theta$	Θ	$\{a_g : s\mapsto \theta-g(y_{1:n})\}$	$-\|r\|^2$
Interval estimation	$\mathscr{Y}^n\times\Theta$	$\{0,1\}\times\mathbb{R}_+$	$\{a_g : s\mapsto (\mathbb{1}_{\{\theta\in g(y_{1:n})\}}, \lvert g(y_{1:n})\rvert)\}$	$r_1+\gamma r_2$
Model choice	$\mathscr{Y}^n\times\left(\cup_{m=1}^M\{m\}\times\Theta_m\right)$	$\{0,1\}$	$\{a_g : s\mapsto \mathbb{1}_{\{m=g(y_{1:n})\}}\}$	r

Figure 2.1: Some classical formalizations of ML problems as decision problems. Actions are labeled by functions g (“predictors"), the domain and codomain of which should be obvious from the definition; for instance g outputs a $\{0,1\}$ label in classification, and a Borel subset of Θ in “interval" estimation.

will come later in Chapter II, when we discuss state-dependent utilities.

2.2 Bayesians maximize expected utility

By definition, a Bayesian is someone who, facing a decision problem, picks a joint distribution p over $(\mathscr{S},\Sigma_{\mathscr{S}})$ and commits to choosing actions according to the following principle. After observing $T\in\Sigma_{\mathscr{S}}$, they will pick action

$$a^\star \in \arg\max_{a\in\mathscr{A}} \mathbb{E}\left[u(a(s))|T\right]. \tag{2.1}$$

Usually, one states (2.1) in the particular case where $T=\mathscr{S}$, i.e.

$$a^\star \in \arg\max_{a\in\mathscr{A}} \mathbb{E}_{s\sim p} u(a(s)). \tag{2.2}$$

Prescription (2.2) is called the *subjective* expected utility principle (SEU), to insist on the fact that p is an arbitrary choice from the decision maker. We choose to rather define a Bayesian as one following the conditional SEU (2.1), because for subtle reasons to be discussed in Section **??**, abiding by (2.2) does not imply that we should use conditional expectations to select actions after we have observed an event. At this stage, we also have not discussed how Bayesians choose p, and there are many ways to do so; see Section **??**. Finally, to give an example of Bayesian decision, the ridge regression estimator is the Bayes action for a particular decision problem and joint distribution over states; see Section **??**.

2.2.1 Posterior subjective expected utility

In ML, it is customary to split the state variable into (s_O,s_U), where $O,U\subset\{1,\dots,\dim\mathscr{S}\}$ are disjoint subsets that respectively index observed states (“data") and unknown states. In particular, we can rewrite (2.2) as

$$a^\star \in \arg\max_{a\in\mathscr{A}} \mathbb{E}_{s_O}\mathbb{E}_{s_U|s_O} u(a(s)). \tag{2.3}$$

Now assume that actions are labeled by a “predictor", which maps s_O to one or several of the unknown variables of interest, say s_I for some $I\subset U$. This is the case for all rows in Figure 2.1. In classification or regression, for instance, the variable of interest is the new label y, and actions are labeled by measurable predictors of this variable of interest: evaluating $g(x;x_{1:n},y_{1:n})$ is thought of as training a given algorithm (say, an SVM) over $\{(x_i,y_i), 1\leq i\leq n\}$ and evaluating the corresponding predictor at the new feature vector x. Now, to maximize (2.3) over $\mathscr{A}$, it is enough maximize it over g. And since g is a function of s_O only, the optimal g is

$$g^\star : s_O \mapsto \arg\max_{g} \mathbb{E}_{s_U|s_O} u(a_g(s)). \tag{2.4}$$

Indeed, by maximizing the innermost expectation in (2.2) for each fixed value of s_O, we maximize the whole expectation. The resulting $g^\star$ is called the *Bayes decision rule*. The expectation in (2.4) is called *posterior* expected utility, in reference to the fact that, in practice, we often compute it for a single value of s_O, *after* s_O has been observed.

2.3 Specifying a joint model

We assume here familiarity with probabilistic graphical models, to the point of telling from a graph whether two sets of nodes are independent given a third one. The reader needing a recap is referred to (Murphy, 2012, Sections 10.1 to 10.5).

2.4 The Bayes decision rule for common ML problems

II

Foundations

3. Many (incompatible) reasons to be a Bayesian

So far, the main appeal of subjective expected utility (SEU) has been its conceptual simplicity, and the fact that it answers all decision problems in the same manner. In this chapter, we review some attempts at justifying SEU more formally, i.e. show that SEU logically follows from some simple, consensual principle. When you hear someone say that “Non-Bayesians are incoherent”, that “a prior encapsulates the information available *before* an experiment is made, so that the prior cannot depend on data", or that “the posterior is the experimenter’s updated belief after the experiment", or that “Non-Bayesians violate the likelihood principle", they are all referring to one or the other of the formal justifications below. Not all these justifications are compatible, and none can really claim to be uniformly superior, so it is important to know upon what arguments you are resting your interpretation of the Bayesian procedure.

3.1 Because you abide by the likelihood principle

The “formal" likelihood principle (Berger and Wolpert, 1988) is a semi-formal justification that deals primarily with the estimation problem. We say *half-formal* because some of the notions it deals with are left without rigorous mathematical definition, although this is not the main weak point (Berger and Wolpert, 1988, Discussions).

3.1.1 The formal LP

Consider two statistical experiments

$$E_i = (\mathscr{Y}_i, s, \{p_i(\cdot|\vartheta), \vartheta \in \Theta\}), \quad i = 1,2.$$

Assume that for some realizations $\mathbf{y}_1$ and $\mathbf{y}_2$,

$$p_1(\mathbf{y}_1|\cdot) \propto p_2(\mathbf{y}_2|\cdot).$$

Note that we are using bold characters to insist on $\mathbf{y}_1$ and $\mathbf{y}_2$ being vectors concatenating (possibly many, of arbitrary dimension) observations, the label $i = 1,2$ is only there to indicate which experiment we consider. In particular, $\mathbf{y}_1$ and $\mathbf{y}_2$ may differ in dimension.

Now, assuming that there exists a quantity $\mathrm{Ev}(E,x)$ that encapsulates the "evidence on θ arising from E and x", the formal LP principle is the requirement that

$$\mathrm{Ev}(E_1,\mathbf{y}_1) = \mathrm{Ev}(E_2,\mathbf{y}_2).$$

As a corollary, $\mathrm{Ev}(E,\mathbf{y})$ can depend on x solely through $p(\mathbf{y}|\cdot)$.

3.1.2 SEU satisfies the LP

Letting $\mathscr{S} = \mathscr{Y}^n \times \theta$, SEU satisfies the LP as long as the joint distribution over states has either p_1 or p_2 as its conditional of $\mathbf{y}$ given θ. Indeed, let

$$p_i(s_i) = p_i(\mathbf{y}_i,\theta) = p_i(\mathbf{y}_i|\theta)p(\theta) = Zp_i(\theta|\mathbf{y}_i), \quad i = 1,2.$$

Note how we use a common prior. Then for $a : \mathscr{S} \to \mathscr{Z}$,

$$\int L(a,s_1)\frac{p_1(\mathbf{y}_1|\theta)p(\theta)}{Z}\mathrm{d}\theta \propto \int L(a,s_2)\frac{p_2(\mathbf{y}_2|\theta)p(\theta)}{Z}\mathrm{d}\theta,$$

so that the posterior expected losses are the same in both experiments, and Bayes actions coincide. However, note that full expected utilities are different in general,

$$\int L(a,s_1)p_1(\mathbf{y}_1|\theta)p(\theta)\mathrm{d}\mathbf{y}_1\mathrm{d}\theta \neq \int L(a,s_2)p_2(\mathbf{y}_2|\theta)p(\theta)\mathrm{d}\mathbf{y}_2\mathrm{d}\theta.$$

3.1.3 The stopping rule principle

The same kind of computations shows that SEU with a particular choice of joint distribution is immune to data-dependent stopping rules. This can also be seen as a consequence of the LP (Berger and Wolpert, 1988), but we stick to SEU with some conditions on its joint distribution of states for simplicity.

Assume that we want to model the following inference problem. We collect data one item at a time, independently from some distribution $y_i|\theta$, until the first $n \in \mathbb{N}$ such that $y_1,\dots,y_n \in A_n$, and then you want to estimate θ. We model this by $\mathscr{S} = \Theta \times \cup_{n\geq 1}\mathscr{Y}^n$, and decide to take a joint distribution p such that $y_1, y_2, \dots$ are independent given θ, just like we assume the data generating mechanism works. Then the Bayes action $a^\star = a_{g^\star}$ minimizes

$$\begin{aligned}
\mathbb{E}L(a,s) &= \mathbb{E}\left[L(a,s)\sum_n \mathbb{1}_{\{N=n\}}\right] \\
&= \sum_n \mathbb{E}\left[L(a,s)\mathbb{1}_{\{N=N\}}\right] \\
&= \sum_n \int L(a,(\theta,y_{1:n}))\mathbb{1}_{\{y_{1:n}\in A_n\}}\prod_{k<n}\mathbb{1}_{\{y_{1:k}\notin A_k\}}p(y_{1:n}|\theta)p(\theta)\mathrm{d}y_{1:n}\mathrm{d}\theta. \\
&= \sum_n \int \mathrm{d}y_{1:n}\mathbb{1}_{\{y_{1:n}\in A_n\}}\prod_{k<n}\mathbb{1}_{\{y_{1:k}\notin A_k\}}\int L(a,(\theta,y_{1:n}))p(y_{1:n}|\theta)p(\theta),
\end{aligned}$$

where we used the monotone convergence theorem and Fubini's theorem (assuming, e.g., that the loss is bounded). So, to find the minimizer $g^\star$ defined on $\cup_n\mathscr{Y}^n$ of the overall expected loss, it is enough, for each n, to define $g^\star(y_{1:n})$ as the usual Bayes rule for fixed n, i.e. as the minimizer of the inner integral. In other words, as long as the prior $p(\theta)$ does not depend on data, the Bayes decision is immune to data-dependent stopping rules: just act as if there were no stopping rule.

3.1.4 Pros and cons of the LP

- The LP is compelling to many (Berger and Wolpert, 1988), but it has its downsides.
- Being Bayesian is not the only way to abide by the LP.
- I am personally uncomfortable with the stopping rule principle, probably because my frequentist intuition is still too strong.
- It is hard to make fully formal: is $\mathrm{Ev}(E,x)$ even meaningful? See answer by LeCam to (Berger and Wolpert, 1988).
- It assumes we want to specify a likelihood, this prevents model-free Bayesianism.
- It separates the roles of the likelihood and the prior. For LP-abiding Bayesians, the prior is not allowed to depend on data.

3.2 Because you place coherence above all things: subjective Bayes

The literature on the foundations of subjective Bayes is rich, and we refer to (Parmigiani and Inoue, 2009) for entry points. A major milestone was obtained by **Sav53**, building on work of **VoMo** and **Ram**. Savage gave a list of properties (the so-called *Savage axioms* of coherence) that a binary relation $\succ$ on the action space $\mathscr{A}$ should satisfy in order for it to have an essentially unique expected utility representation. More precisely, $\succ$ satisfies the Savage axioms if and only if there is a bounded utility function $u : \mathscr{S} \rightarrow \mathscr{R}_+$ and a (finitely additive) probability measure p on $\mathscr{S}$ such that for $a, b \in \mathscr{A}$,

$$a \succ b \Leftrightarrow \int u(a(s))\mathrm{d}p(s) > \int u(b(s))\mathrm{d}p(s);$$

u is unique up to affine transformations. The pair (u, p) thus characterizes the behaviour of a Savage-abiding decision maker, whose preferred action is the one minimizing the expected utility. This is the sense of the word *subjective* in SEU: the probability p, as well as the utility u, live in the mind of the DM. This is also the sense in which p can be interpreted as the DM's degree of belief. Finally, note that there is no constraint on the probability p: any pair (u, p) corresponds to a coherent decision-maker. In other words, Savage tells you that to be coherent you have to follow SEU, but it does not help you choose a joint model over states, let alone a prior.

Savage's result is beautiful, in that it builds a probability measure p and a utility function u over states from a coherent ranking of actions. Following Savage, there is no need to assume that the phenomenon we are studying is probabilistic, or to dissentangle different forms of uncertainty: a coherent decision maker *must* have a utility and a probability in mind! We now take a closer look at Savage's axioms as an example of prescritive subjective Bayesian framework, and examine some common criticisms.

3.2.1 A closer look at the axioms

Since axiom numbers 'Px' in the original paper of Savage, 1954 are often referred to as such, I use them as labels. Yet I introduce axioms in the same order as Parmigiani and Inoue, 2009.

An important thing to notice before looking at his axioms, is that Savage considers the action set $\mathscr{A}$ to be all functions from $\mathscr{S}$ to $\mathscr{Z}$. In particular, constant actions, i.e. actions of the type $s \mapsto z$ for some $z \in \mathscr{Z}$, play a crucial role in Savage's derivation, even if they do not correspond to actual actions available to the decision maker (DM) in most cases. In the sequel, we abusively denote by z the constant action $s \mapsto z$.

Axiom P1 (Preference). *$\succ$ is complete and transitive.*

Since $\succ$ is complete, we can thus check whether $z_1 \succ z_2$ for any two constant actions z_1, z_2.

Axiom P5 (No total indifference). $\exists z_1, z_2 \in \mathscr{Z}$ *such that* $z_1 \succ z_2$.

■ **Example 3.1** If we take binary classification as a running example, where $\mathscr{Z} = \{0,1\}$, the only two constant actions are the “ideal" classifier 0 that is always right and the one that is always wrong. Most of us will likely elicit $0 \succ 1$, to say that we prefer being always right to being always wrong. Note that none of the two constant actions usually belong to the "pratically accessible" actions $\{a_g : s \mapsto y - g(x; x_{1:n}, y_{1:n}), g \in \mathscr{G}\}$. ■

Now for $a, b \in \mathscr{A}$, $T \in \mathscr{S}$, define action a_T^b by $a_T^b(s) = a(s)1_{s\in T} + b(s)1_{s\in T}$.

Axiom P2 (Sure Thing principle). *$\forall a, b, h_1, h_2 \in \mathscr{A}$ and $\forall T \subset \mathscr{S}$,*

$$a_{T^c}^{h_1} \succ b_{T^c}^{h_1} \Leftrightarrow a_{T^c}^{h_2} \succ b_{T^c}^{h_2}.$$

Here, Savage directly formulates that if two acts coincide on part of $\mathscr{S}$, then preference should only depend on their values where they differ. In particular, under P2 (Sure Thing principle), we can define a new family of preference relations, called *conditional preferences*. Let $T \subset \mathscr{S}$, and define $a \succ b|T$ by

$$c_T^a \succ c_T^b \tag{3.1}$$

for some (equivalently all) $c \in \mathscr{A}$. This family of conditional preferences will be discussed later, when we consider how to rank actions after some T has been observed.

■ **Example 3.2 — Continuation of Example 3.1.** ■

Define now a null state as a $T \subset \mathscr{S}$ such that $a \sim b|T$ for all $a, b \in \mathscr{A}$. Intuitively, this means that preferences are insensitive to T obtaining.

Axiom P3 (No reduction to conditional indifference). *If T is not null, then $a_T^{z_1} \succ a_T^{z_2}|T$ iff $z_1 \succ z_2$.*

Note that by P2 (Sure Thing principle), the values of a and b outside T do not matter. Axiom P3 (No reduction to conditional indifference) makes sure that no preference among consequences can be reduced to indifference conditional on a non-null event.

Axiom P4 (Separation). *Assume that $z_1 \succ z_2$ and $z_1' \succ z_2'$. Let $T_1, T_2 \subset \mathscr{S}$, and*

$$a = z_{2T_1}^{z_1}, b = z_{2T_2}^{z_1}, a' = z'^{z_1'}_{2T_1}, b' = z'^{z_1'}_{2T_2}.$$

Then $a \succ b \Leftrightarrow a' \succ b'$.

In words, for two actions that are both piecewise constant taking the same set of values, I can change the constant values without altering the preference, as long as I preserve the order between outcomes. Intuitively, if outcomes were money, your willingness to bet on T_1 obtaining rather than T_2 obtaining does not depend on how much money you make/lose in each of these two bets.

We can now define a binary relation $T_1 \succ T_2$ over subsets of $\mathscr{S}$ that we could interpret as “T_1 is more likely to me that T_2". We say $T_1 \succ T_2$ if for all $z_1, z_2 \in \mathscr{Z}$ such that $z_1 \succ z_2$,

$$z_{2T_1}^{z_1} \succ z_{2T_2}^{z_1}.$$

In the betting metaphor, you’re more willing to bet on T_1 obtaining than T_2 for the same rewards.

These first five axioms are enough to get the following representation.

Theorem 3.2.1 — Qualitative representation. If P1 (Preference), P2 (Sure Thing principle), P3 (No reduction to conditional indifference), P4 (Separation), and P5 (No total indifference) hold, then $\succ$ on $\mathscr{S}$ is a qualitative probability, that is

- $\succ$ is negatively transitive: $T_1 \preccurlyeq T_2 \preccurlyeq T_3$ implies $T_1 \preccurlyeq T_3$,
- $\forall R \subset \mathscr{S}, T \succ \emptyset$.

- If $T_1 \cap U = T_2 \cap U = \emptyset$, then $T_1 \succ T_2$ iff $T_1 \cup U \succ T_2 \cup U$.

We could stop here and work with beliefs specified by pairwise rankings of events. But Savage, 1954 goes forward, and keeps adding axioms to get a more usual *quantitative* probability. In particular, we need an additional structural axiom on $\mathscr{S}$. There are variations of this “partition axiom", and Savage chooses to embed it in its Archimedean axiom.

Axiom P6 (Archimedean). *$\forall a, b \in \mathscr{A}$ such that $a \succ b$, and $\forall z \in \mathscr{Z}$, there exists a finite partition of $\mathscr{S} = T_1 \cup \cdots \cup T_M$ such that for all T_i, either $a^z_{T_i} \succ b$ or $a \succ b^z_{T_i}$.*

An important consequence of P6 (Archimedean) is that $\mathscr{S}$ must be rich enough to be splittable into tiny pieces suiting any pair of actions. This is in general not possible for a discrete space, and we will generally have to use a continuous state space.

Theorem 3.2.2 — Qualitative representation. Under P1 (Preference), P2 (Sure Thing principle), P3 (No reduction to conditional indifference), P4 (Separation), P5 (No total indifference), and P6 (Archimedean), there exists a unique finitely additive probability measure Π on $2^{\mathscr{S}}$ such that

- $T_1 \succ T_2$ iff $\Pi(T_1) > \Pi(T_2)$,
- $\forall T_1 \subset \mathscr{S}$ and $k \in [0,1]$, there exists $T_2 \subset \mathscr{S}$ such that $\Pi(T_2) = k\Pi(T_1)$.

Now to obtain a full expected utility representation, we need a final axiom.

Axiom P7 (No aversion to risk). *$\forall T \subset \mathscr{S}$,*

- *If $\forall s \in T$, $a \succ b(s)|T$, then $a \succ b|T$.*
- *If $\forall s \in T$, $a(s) \succ b|T$, then $a \succ b|T$.*

I call this *no aversion to risk*, since it intuitively means that if you prefer a to all certain consequences of bn then you prefer a to b.

Theorem 3.2.3 — Qualitative representation. Under P1 (Preference), P2 (Sure Thing principle), P3 (No reduction to conditional indifference), P4 (Separation), P5 (No total indifference), P6 (Archimedean), and P7 (No aversion to risk), there exists a unique finitely additive probability measure satisfying the results of Theorem 3.2.2. Furthermore, there is a unique (up to positive affine transformations) bounded utility function $u : \mathscr{Z} \to \mathbb{R}$ such that

$$a \succ b \Leftrightarrow \int u(a(s))d\pi > \int u(b(s))d\pi.$$

Note that the probability measure in Theorem 3.2.3 is finitely additive, and defined on the Boolean algebra $2^{\mathscr{S}}$ of all subsets of $\mathscr{S}$. In particular, the integrals in Theorem 3.2.3 are *not* Lebesgue integrals, see e.g. Kreps, 1988.

3.2.2 Major criticisms

There are several points that can be raised against using Savage’s result to justify Bayesian learning, which we roughly rank by increasing seriousness. First, the resulting probability measure is only finitely additive: it is hard to bring a σ-algebra like $\Sigma_{\mathscr{S}}$ into the picture with a consensual coherence axiom that does not involve an infinite number of choices from the DM. This is a minor inconvenience, as we can always restrict actions to a set of measurable functions and choose a *bona fide* probability distribution. The price is that we might be unable to represent all coherent behaviours, and there might be interesting statistical stances to be had with finitely additive probability ().

A second objection is the boundedness of the utility function, first noticed by (Fishburn, 1970). Common losses such as the squared loss need to be trimmed to fit into the picture without modifying the axioms. At the cost of some less natural axioms, one can however accomodate unbounded loss functions (**Wak93**).

A third objection is the pivotal role played by constant actions in Savage's (and actually von Neumann and Morgenstern's) proof. Constant actions are those that assign the same reward $r \in \mathscr{R}$ to all states $s \in \mathscr{S}$. The utility function in approaches derived from **VoMo** is defined by finding the weight in a convex combination of two extreme constant actions; see e.g. (Parmigiani and Inoue, 2009, Chapter 3). But constant actions are not part of the action sets that we considered in Section **??**. In binary classification, for instance, there are two constant actions: the ideal classifier $a^\star$ that always predicts the right label, and the worst possible classifier $a_\star$, which consistently predicts the wrong label. The utility function of any reward r is defined in reference to these two idealized classifiers. It would be more satisfying to have a set of axioms that does not give such a key role to *non-physical* classifiers.

A fourth and related objection is that the utility in Savage's result is independent of the state: $u(a(s))$ only depends on the state through $a(s)$. This means that the notation $L(a,s) = -u(a(s))$ is inappropriate, as it lets the user think that the loss of action a can depend on the state s. In textbook ML tasks, this is not much of a problem, but in real applications, this can prove limiting; see (Parmigiani and Inoue, 2009, Chapter XXX).

Maybe the most important weakness of axiomatic constructions like Savage's is that they do not prescribe what action to choose after some part of the state is observed. Using the conditional SEU is only a natural candidate for what to do, not more. We should not be fooled by the name "conditional expectation": call it "strange mathematical construction" and then try to think if this is how you want to change your behaviour once you observe an event. There are ways to justify conditioning by means of sequential Dutch books (), but none that is not controversial in the general case of an infinite state space. Worse: it is doubtful that this can be done, since in practice, we all refrain from conditioning from time to time. For instance, if you observe that your classifier has poor test error, you change the family $\mathscr{G}$ of classifiers that you consider. Unless you included the choice of $\mathscr{G}$ in your state space, this kind of "external" move, which looks at some calibration property of a subjective model, is not constrained by Savage-like constructions. Modern attempts to build Bayesian statistics with more focus on calibration include the prequential view of **Daw**. One could also argue that PAC-Bayes goes along those lines, by abandoning conditioning in favour of better frequentist properties; more about this in Section **??**.

3.3 Because you like coherence and consensus: objective Bayes

3.4 Because you are a Waldian frequentist in disguise

3.4.1 On the consistency of Bayesian estimators

Restrict to parametric Bernstein von-Mises and its consequences.

3.4.2 Complete class theorems

3.4.3 PAC-Bayes statistical learning

III

Implementing Bayesian machine learning

4. Markov chain Monte Carlo

Maximizing expected utility requires computing integrals. Numerical integration consists in finding T nodes x_t and weights w_t, such that

$$\mathscr{E}_T(f) \triangleq \int f(x)\mathrm{d}\mu(x) - \sum_{t=1}^{T} w_t f(\theta_t) = o_{T\to\infty}(1), \quad \forall f : \mathbb{R}^d \to \mathbb{R} \in \mathscr{C},$$

where $\mathscr{C}$ is a large class of functions. For any smooth f, Riemann-like (i.e., grid-based) integration leads to $\mathscr{E}_T(f) \sim \frac{\sqrt{d}}{T^{1/d}}$, so that grids become essentially useless beyond $d = 3$. Monte Carlo methods are randomized constructions of nodes and weights. Since $\mathscr{E}_T(f)$ is then random, statements on the error shall be with high probability, in expectation, about the asymtotic fluctuations, etc.

We shall see that Monte Carlo methods (i) can accomodate settings where the target μ in (**??**) is only available through the evaluation of a unnormalized density $\mathrm{d}\mu(x) = \pi(x)\mathrm{d}x = \pi_u(x)/Z\mathrm{d}x$, as is almost always the case in Bayesian learning; and that (ii) some Monte Carlo methods have an error that scales only polynomially with the dimension of the support of the integrand.

References.
The reference book for basic Monte Carlo methods is (Robert and Casella, 2004). For theoretical results on MCMC, we refer to (Douc, Moulines, and Stoffer, 2014, Chapters 5 to 7). For more recent methods, we shall refer to papers. For demos of MCMC samplers, see `https://chi-feng.github.io/mcmc-demo/`. Finally, we shall not cover deterministic alternatives to Monte Carlo methods in large dimension, see quasi-Monte Carlo methods (Dick and Pilichshammer, 2010).

4.1 Basic Monte Carlo

If we knew how to sample from π, we could take $x_t \sim \pi$ i.i.d., $w_t = 1/T$. Chebyshev's inequality would lead to

$$\mathbb{P}\left(\mathscr{E}_T(f) \geq \alpha \frac{\sigma(f)}{\sqrt{T}}\right) \leq \frac{1}{\alpha^2}, \quad \forall \alpha > 0,$$

as soon as $\sigma(f)^2 := \mathbb{E}_{X\sim\pi}[f(X) - \int f(x)\pi(x)\mathrm{d}x] < +\infty$.

In practice, we never have access to a sampler of π, so we choose an instrumental distribution $q(x)\mathrm{d}x$, sample x_t i.i.d. from q. If we can evaluate π, then $w_t = \pi(x_t)/q(x_t)$ leads to an unbiased estimator called the *importance sampling*. Its error can also be controlled by the Chebyshev inequality. But more often than not, we can only evaluate an unnormalized density π_u. An alternative is to then set $w_t \propto \pi_u(x_t)/q(x_t)$ and normalize the weights so that $\sum_t w_t = 1$. This leads to the *self-normalized importance sampling estimator*

$$\widehat{I}_T^{\text{NIS}} = \frac{\sum_{t=1}^T \frac{\pi_u(\theta_t)}{q(\theta_t)} f(x_t)}{\sum_{t=1}^T \frac{\pi_u(\theta_t)}{q(\theta_t)}} \to \frac{\int f(x)\pi_u(x)\mathrm{d}x}{\int \pi_u(x)\mathrm{d}x} = \int f(x)\pi(x)\mathrm{d}x,$$

where the convergence is almost sure (apply the strong law of large numbers to both the numerator and the denominator).

Proposition 4.1.1 — CLT for NIS. The NIS estimator satisfies

$$\sqrt{T}\left(\widehat{I}_T^{\text{NIS}} - \int f(x)\pi(x)\mathrm{d}x\right) \to_d \mathscr{N}(0, \sigma^2_{\text{NIS}}(f))$$

where f is such that

$$\sigma^2_{\text{NIS}}(f) \triangleq TBC < \infty.$$

Proof. See exercise sheet. ■

Unfortunately, while NIS does accomodate unnormalized targets, it does not solve the curse of dimensionality: even when π and q are both Gaussian, only with different covariance matrices, one can show that

$$\log \sigma^2_{\text{NIS}}(f) = \Theta(d).$$

4.2 The Metropolis-Hastings algorithm

Metropolis-Hastings (MH) is the achetypal MCMC algorithm, and is still the main building block of modern MCMC algorithms. The idea is to take the nodes as the truncated history of a Markov chain (X_t), which we build so as to guarantee that $\mathscr{E}_T(f) \to 0$. To see how to build (X_t), remember first the law of large numbers for Markov chains.

Proposition 4.2.1 — LLN for Markov chains; see e.g. Douc, Moulines, and Stoffer, 2014. Let $(X_t)_{t\in\mathbb{N}}$ be a Markov chain with Markov kernel P. If

1. There exists μ s.t.
$$\int \mathrm{d}\mu(x)P(x,B) = \mu(B).$$
2. For any A with $\mu(A) > 0$, for any $\theta \in \Theta$,
$$\mathbb{P}_x\left(\sum_{t=0}^{\infty} 1_{\theta_t \in A} = +\infty\right) = 1,$$

then for any initial distribution μ_0 of X_0, almost surely

$$\frac{1}{T}\sum_{t=1}^{T} f(\theta_t) \to \int f \mathrm{d}\mu,$$

where $f \in L^1(\mu)$.

The first condition states that π is an *invariant distribution* of the chain: if $X_t \sim \mu$, then $X_{t+1} \sim \mu$. The second condition is called *Harris recurrence*: starting from any x, i.e. $X_0 \sim \delta_x$, then the chain returns an infinite number of times to A almost surely, as soon as A is charged by μ. Intuitively, this makes sure that there is an infinity of nodes on A, so that the integral of f on A is accurately estimated.

4.3 Gibbs sampling

4.4 Hamiltonian Monte Carlo

We closely follow Bou-Rabee and Sanz-Serna, 2018, who give a clear introduction to the ingredients of HMC. To facilitate going back and forth between the notes and (Bou-Rabee and Sanz-Serna, 2018), we temporarily adopt their notational conventions: the variable over which we wish to integrate is $q \in \mathbb{R}^d$, while $x \in \mathscr{X}$ denotes a generic variable, later taken to be $x = (p,q) \in \mathbb{R}^{2d}$. Note that vanilla HMC is limited to continuous variables.

4.4.1 An abstract variant of Metropolis-Hastings

Let S be a linear involution of $\mathscr{X} \subset \mathbb{R}^{2d}$, such that $\eta \circ S = \eta$ for some (possibly unnormalized) PDF η. Let further $\Phi : \mathbb{R}^{2d} \to \mathbb{R}^{2d}$ be a C^1-diffeomorphism that is reversible w.r.t. to S, that is, $S \circ \Phi = \Phi^{-1} \circ S$. Now let

$$\alpha(x) \triangleq 1 \wedge \frac{\eta(\Phi(x))}{\eta(x)} |\Phi'(x)|, \tag{4.1}$$

and consider the Markov kernel

$$P_{aHMC}(x,A) = \alpha(x) 1_{\Phi(x)\in A} + (1-\alpha(x)) 1_{S(x)\in A}.$$

Algorithmically, this “abstract" HMC kernel corresponds to accepting $\Phi(x)$ with probability $\alpha(x)$, and otherwise setting the new Markov state to $S(x)$.

Proposition 4.4.1 P_{aHMC} leaves η invariant.

Proof. Left as an exercise. ■

4.4.2 An augmented target

Consider the PDF on $\mathbb{R}^{2d}$ defined by $\tilde{\pi}(q,p) = \frac{1}{2}\mathscr{N}(p|0,M) \times \pi(q)$. Clearly, the p-marginal is Gaussian, while the q-marginal is π. If we manage to obtain an MCMC chain for $\tilde{\pi}$, i.e. a chain with a Markov kernel that leaves $\tilde{\pi}$ invariant, then simply discarding the p-component of every realization will yield a chain that is invariant w.r.t. π. This is an example of *augmentation* of the state space: unlike for the collapsed Gibbs sampling of Section **??**, we augment the dimensionality of the problem in the hope to make sampling easier. Our hope is justified here by the fact that we know how to efficiently move in $\mathbb{R}^{2d}$ along the level lines of the augmented density $\tilde{\pi}$: this is were Hamiltonian dynamics come into play.

4.4.3 Hamiltonian dynamics

Let $H(q,p) = \log \tilde{\pi}(q,p)$, and consider the differential equation

$$\frac{\mathrm{d}}{\mathrm{d}t}\begin{pmatrix} x \\ p \end{pmatrix} = J\nabla H(q,p), \qquad \text{where } J = \begin{pmatrix} 0_d & -I_d \\ I_d & 0_d \end{pmatrix}. \tag{4.2}$$

In particular, if ∇H is Lipschitz, which we shall always assume in this section, the Cauchy-Lipschitz theorem yields a unique solution to (4.2) passing through (q_0, p_0) at $t = 0$, which we denote by $t \mapsto \phi_t(q_0, p_0)$. We shall further assume that ϕ_t is well defined for all t, and call ϕ_t the *Hamiltonian flow*. By definition, the Hamiltonian flow preserves the Hamiltonian, since

$$\frac{\mathrm{d}}{\mathrm{d}t} H(\phi_t(q_0,p_0)) = \nabla H(\phi_t(q_0,p_0))^T J \nabla H(\phi_t(q_0,p_0)) = 0,$$

J being skew-symmetric. In other words, the flow ϕ_t follows level lines of H.

4.4.4 Ideal HMC and numerical HMC

The ideal HMC is the concatenation of two kernels. Given (q_n, p_n), first resample p from its conditional under $\widetilde{\pi}$; i.e. $p' \sim \mathscr{N}(0,M)$. This obviously leaves $\widetilde{\pi}$ invariant, as in Gibbs sampling. Then set $(q_{n+1}, p_{n+1}) = \phi_T(q_n, p')$. In words, one step of the corresponding Markov chain consists of sampling a random momentum variable from the corresponding conditional, and then following the Hamiltonian flow ϕ_t up to time $T > 0$. Note that, unlike Gibbs sampling, this second step changes both variables.

One can formulate the intuition that, since the second step just follows a level line of H, the ideal HMC kernel leaves $\widetilde{\pi}$ invariant. This is indeed the case (Bou-Rabee and Sanz-Serna, 2018), but since ϕ_t is usually not available in closed form, the ideal HMC kernel cannot be implemented, and we will skip the proof of its invariance.

In practice, one has to approximate the Hamiltonian flow ϕ_t, and there is a large literature in numerical analysis on the subject, with integrators showcasing many interesting properties. In terms of notation, denote by h a stepsize parameter, and $n = \lfloor T/h \rfloor$, so that we think of the numerical integrator $\psi_h^n(x,p)$ as an approximation to $\phi_T(x,p)$. There exists numerical integrators ϕ_h^n that are (i) C^1 diffeomorphisms, (ii) are reversible w.r.t. to momentum flip, and are volume-preserving, i.e. $|\det(\psi_h^n)'(q,p)| = 1$. Since the momentum flip preserves $\widetilde{\pi}$, we can replace the second step of the ideal HMC algorithm by the abstract HMC algorithm of Section 4.4.1, with $\eta = \widetilde{\pi}$, $\Phi = \phi_h^n$ and S the momentum flip. Because the numerical integrator is volume-preserving, the acceptance probability becomes suprisingly simple, as

$$\alpha_{HMC}((q,p),(q',p')) = 1 \wedge \mathrm{e}^{-H(q,p)-H(q',p')}.$$

Intuitively, the acceptance step compensates for the fact that we did not exactly follow the level lines of H.

One common numerical integrator satisfying all the required properties is the leapfrog (aka velocity Verlet in (Bou-Rabee and Sanz-Serna, 2018)) integrator defined as $\psi_h^n = \psi_h \circ \dots \psi_h$, where $(p',q') = \psi_h(p,q)$ is defined by

$$\begin{aligned} p_{1/2} &= p + \frac{h}{2}\nabla \log \pi(q) \\ q' &= a + hM^{-1}p_{1/2} \\ p' &= p_{1/2} + \frac{h}{2}\nabla \log \pi(q'); \end{aligned}$$

see (Bou-Rabee and Sanz-Serna, 2018, Section 3) for more information, and `https://chi-feng.github.io/mcmc-demo/`for an interactive demo.

4.4.5 On the ergodicity of HMC

4.4.6 An ubiquitous variant: NUTS

NUTS for auto-tuning, etc.

4.5 MCMC practice: convergence diagnostics

Convergence diagnostics. Discuss the output of pymc.

4.6 Alternative methods

4.6.1 Other randomized quadratures

More kernels, SMC samplers, PDMPs

4.6.2 Deterministic quadrature in large dimensions

Herding QMC, randomized QMC

5. Variational Bayes

Monte Carlo methods are randomized numerical quadratures that approximate the target measure $\mathrm{d}\mu(x) = \pi(x)\mathrm{d}x$ by a random discrete measure with a finite number of atoms. Alternately, one might try to approximate μ by directly miniziming some distance between μ and a candidate approximation $q(x)\mathrm{d}x$ in some parametrized subset $\mathscr{Q} \subset \mathscr{M}_1$ of non-atomic candidate approximations. Since our ultimate goal is to approximate expected utilities with respect to μ, see Section **??**, one may wish to find a $q \in \mathscr{Q}$ that minimizes the worst-case error we would make by replacing π with q in the integration of some class $\mathscr{F}$ of functions. Formally, we might wish to minimize

$$q \mapsto d_{\mathscr{F}}(\pi, q) \triangleq \sup_{f \in \mathscr{F}} \left| \int f(x)\pi(x)\mathrm{d}x - \int f(x)q(x)\mathrm{d}x \right| \tag{5.1}$$

When $\mathscr{F}$ is a reproducing kernel Hilbert space and $\mathscr{Q}$ is the set of weighted measures with N atoms, this leads to kernel herding; see Section 4.6.

In this chapter, we rather consider minimizing the KL divergence between q and π, i.e.

$$q^\star \in \arg\min_{q \in \mathscr{Q}} \mathrm{KL}(q, \pi) := \mathbb{E}_q \log \frac{q(x)}{\pi(x)} = \int q(x) \log \frac{q(x)}{\pi(x)} \mathrm{d}x. \tag{5.2}$$

Once (5.2) has been solved, one can use the resulting $q^\star$ as a plug-in replacement for the target π. Unfortunately, the (reverse) KL divergence does not have the form (5.1), and in general, $q^\star$ will not be guaranteed to lead to a controlled integration error. Actually, the combined use of the reverse KL and of $\mathscr{Q} \subsetneq \mathscr{M}_1$ usually leads an optimal $q^\star$ with a smaller support than π. This also makes it difficult to use $q^\star$ as a proposal distribution in importance sampling.

While replacing (5.1) by (5.2) might seem disappointing at first sight, our sacrifice will not be in vain. We shall see in Sections 5.1 to 5.3 that minimizing (5.2) can be computationally tractable in settings where no efficient Monte Carlo method has been proposed yet, in particular in very high dimensions such as when fitting a deep neural network. There is research in frequentist theoretical results that support using (5.2), generalized formulations of Bayesian inference that justify (5.2), and case-specific empirical demonstrations that $q^\star$ retains some of the desireable properties of the posterior; we survey some of these results in Section **??**. Closing the gap between VB and Bayes is an exciting current research topic.

5.1 The evidence lower bound (ELBO)

Remember that the density π is often known only though the evaluation of an unnormalized version π_u, i.e., $\pi_u(\theta) = Z\pi(\theta), \forall\theta$. If we are to carry out (5.2), we thus need to know how to express $\mathrm{KL}(q,\pi)$ using only π_u.

Lemma 1. *Let* $J(q) := \int q(\theta) \log \frac{q(\theta)}{\pi_u(\theta)} \mathrm{d}\theta$. *Then*

$$J(q) = KL(q,\pi) - \log Z. \tag{5.3}$$

Proof. Left as an exercise. ■

Two remarks are in order. First, since Z does not depend on q, the optimization problem in (5.2) is equivalent to $\min J(q)$. Letting $L(q) = -J(q)$, (5.2) is further equivalent to $\max L(q)$. Second and the nonnegativity of the KL divergence implies that

$$L(q) \leq \log Z. \tag{5.4}$$

In Bayesian inference,

$$\pi_u(\theta) = p(y_{1:N}|\theta)p(\theta),$$

so that $Z = p(y_{1:N})$. Furthermore, (5.4) says that $L(q)$ is *a lower bound for the (logarithm of the) evidence*, shortened in ELBO. Most VB algorithms in the literature are cast as maximizing the ELBO.

5.2 Mean-field inference

The most common variational family is the so-called *mean-field approximation*. If you need to approximate a posterior over parameters $\theta \in \mathbb{R}^d$ and latent variables $z_1, \dots, z_N$, this means taking

$$\mathscr{Q} = \left\{ \theta \mapsto \prod_{d=1}^{D} q_d(\theta_d) \prod_{i=1}^{N} q_i(z_i) \right\}. \tag{5.5}$$

In other words, we approximate π with a separable PDF. Note that (5.5) only specifies the structure of the variational approximations. This is enough to derive the abstract form of the VB updates, which we do in the remainder of this section. In practice, though we further make explicit parametric choices for the individual factors, as we shall see in Section 5.2.1.

The whole motivation of the mean-field variational family is that if your target has simple conditionals, coordinate-wise optimization of the ELBO is easy. Indeed, write $x = (\theta, z) \in \mathbb{R}^p$ and let $1 \leq i \leq p$. Writing $q(x) = q_i(x_i)q_{\setminus i}(x_{\setminus i})$, and keeping track only of the additive terms that depend on q_i, it comes

$$\begin{aligned} L(q) &= \iint q_i(x_i)q_{\setminus i}(x_{\setminus i}) \left[\log \pi_u(x) - \left(\log q_i(x_i) + \log q_{\setminus i}(x_{\setminus i})\right)\right] \mathrm{d}x_i \mathrm{d}x_{\setminus i} \\ &\propto \int q_i(x_i) \left[\int q_{\setminus i}(x_{\setminus i}) \log \pi_u(x) \mathrm{d}x_{\setminus i}\right] dx_i - \int q_i(x_i) \log q_i(x_i) \mathrm{d}x_i \\ &= -KL(q_i, \phi_i), \end{aligned}$$

where

$$\phi_i(x_i) = \exp\left[\int q_{\setminus i}(x_{\setminus i}) \log \pi_u(x) \mathrm{d}x_{\setminus i}\right] \tag{5.6}$$

is an unnormalized PDF. By a fundamental property of the KL, the ELBO L is thus maximized by setting $q_i \propto \phi_i$. The bottleneck is thus to be able to compute ϕ_i in (5.6). Like in deriving conditionals in Gibbs sampling, this is the part where conjugate distributions play a role. In practice, the choice of $\mathscr{Q}$ is often made so that this step is easy, as we shall see in Section 5.2.1.

Finally, note that taking the variational family to be (5.5) is akin to assuming independence of all variables under the posterior. Combined with the fact that reverse KL (5.2) penalizes $q^\star$ putting a lot of mass where π does not, this often implies a gross underestimation of the support of the target (and thus of posterior uncertainty), along with the built-in ignorance of posterior correlations; see Figure **??**. While separability makes algorithmic derivations easier, we thus usually rather aim for “as separable as required by computation". In other words, if, for modeling reasons, you believe that there is correlation under π of some subset of the variables, say $x_i, i \in I$, you should try to keep these variables correlated in your variational approximation, by rather defining

$$\mathscr{Q} = \left\{ \theta \mapsto q_I(x_I) \prod_{i \notin I} q_i(x_i) \right\},$$

for some nonseparable q_I. Murphy, 2012, Chapter 23 calls this *structured mean-field*.

5.2.1 Mean-field VB for LDA

Recall the latent Dirichlet allocation model from Section **??**, for which

$$\begin{aligned}
&\log p(y, z, \pi, B) \qquad (5.7)\\
&= \sum_{i=1}^{N} \left[\log p(\pi_i|\alpha) + \sum_{\ell=1}^{L_i} \Big(\log p(z_{i\ell}|\pi_i) + \log p(y_{i\ell}|z_{i\ell}, B) \Big) \right] + p(B|\gamma)\\
&\propto \sum_{i=1}^{N} \left[\sum_{k=1}^{K} \alpha_k \log \pi_{ik} + \sum_{\ell=1}^{L_i} \left(\sum_{k=1}^{K} 1_{z_{i\ell}=k} \log \pi_{ik} + \sum_{v=1}^{V} \sum_{k=1}^{K} 1_{y_{i\ell}=v} 1_{z_{i\ell}=k} \log b_{kv} \right) \right]\\
&\quad + \sum_{k=1}^{K} \sum_{v=1}^{V} \gamma_v \log b_{kv}. \qquad (5.8)
\end{aligned}$$

We want to fit a mean-field approximation

$$\mathscr{Q} = \left\{ \prod_{i=1}^{N} \left[\mathrm{Dir}(\pi_i|\widetilde{\pi}_i) \prod_{\ell=1}^{L_i} \mathrm{Cat}(z_{i\ell}|\widetilde{z}_{i\ell}) \right] \prod_{k=1}^{K} \mathrm{Dir}(B_{k:}|\widetilde{B}_{k:}) \right\}.$$

Tilded variables parametrize the variational approximation q, and optimizing over q will thus be implemented as an optimization over these parameters. As we shall see, the Dirichlet distributions are chosen to make the following computations easy thanks to conjugacy.

To implement VB, we need to compute (5.6) for every coordinate, that is, we need to integrate the log joint distribution (5.8) with respect to all variables but one, for every choice of that singled out variable.

Singling out π_i.

We start by singling out $\pi_i \in \Delta_K$ for some $1 \leq i \leq N$, denoting the corresponding expectation by $\mathbb{E}_{\setminus \pi_i}$. We are confident that we shall be able to identify the functional form (and thus the normalization constant) of the resulting distribution, and thus we do not keep track of additive variable that do not imply π_i. This yields

$$\begin{aligned}
\mathbb{E}_{\setminus \pi_i} \log p(y, z, \pi, B) &\propto \sum_{k=1}^{K} \alpha_k \log \pi_{ik} + \sum_{\ell=1}^{L_i} \mathbb{E}_{z_{i\ell}} \sum_{k=1}^{K} 1_{z_{i\ell}=k} \log \pi_{ik}\\
&= \sum_{k=1}^{K} \alpha_k \log \pi_{ik} + \sum_{\ell=1}^{L_i} \sum_{k=1}^{K} \widetilde{z}_{i\ell k} \log \pi_{ik}
\end{aligned}$$

and we recognize the log PDF of a Dirichlet distribution in π_i, with parameters

$$\widetilde{\pi}_i \triangleq \left(\alpha_k + \sum_{\ell=1}^{L_i} \widetilde{z}_{i\ell k}\right)_{1\leq k\leq K}.$$

Singling out $z_{i\ell}$.
To compute $\mathbb{E}_{\setminus z_{i\ell}} \log p(y,z,\pi,B)$, we need to be able to compute expectation of log weights w.r.t. a Dirichlet distribution.

Lemma 2. *Let $\Psi(\cdot) := \Gamma'(\cdot)/\Gamma(\cdot)$ be the digamma function. Let $\widetilde{\eta} \in \Delta_M$ be a probability distribution over $\{1,\dots,M\}$. Then, for $m \in \{1,\dots,M\}$,*

$$\mathbb{E}_{Dir(\eta|\widetilde{\eta})} \log \eta_m = \Psi(\widetilde{\eta}_m) - \Psi(\|\widetilde{\eta}\|_1) \triangleq \Psi_m(\widetilde{\eta}).$$

Proof. Left as an exercise. ■

Now we derive

$$\begin{aligned}\mathbb{E}_{\setminus z_{i\ell}} \log p(y,z,\pi,B) &\propto \sum_{k=1}^{K} 1_{z_{i\ell}=k} \mathbb{E}_{\pi_i} \log \pi_{ik} + \sum_{v=1}^{V}\sum_{k=1}^{K} 1_{y_{i\ell}=v} 1_{z_{i\ell}=k} \mathbb{E}_{B_{k:}} \log b_{kv} \\ &= \sum_{k=1}^{K} 1_{z_{i\ell}=k} \Psi_k(\widetilde{\pi}_i) + \sum_{v=1}^{V}\sum_{k=1}^{K} 1_{y_{i\ell}=v} 1_{z_{i\ell}=k} \Psi_v(\widetilde{B}_{k:})\end{aligned}$$

We recognize a categorical distribution with parameters

$$\widetilde{z}_{i\ell} \propto \left(\exp\left[\Psi_k(\widetilde{\pi}_i) + \Psi_{y_{i\ell}}(\widetilde{B}_{k:})\right]\right)_{1\leq k\leq K}.$$

Once again, the normalization constant can be guessed after doing the computation, since necessarily

$$\widetilde{z}_{i\ell} = \left(\frac{\exp\left[\Psi_k(\widetilde{\pi}_i) + \Psi_{y_{i\ell}}(\widetilde{B}_{k:})\right]}{\sum_{k=1}^{K} \exp\left[\Psi_k(\widetilde{\pi}_i) + \Psi_{y_{i\ell}}(\widetilde{B}_{k:})\right]}\right)_{1\leq k\leq K}.$$

Singling out $B_{k:}$
In the same vein,

$$\mathbb{E}_{\setminus B_{k:}} \log p(y,z,\pi,B) \propto \sum_{i=1}^{N}\sum_{\ell=1}^{L_i}\sum_{v=1}^{V} 1_{y_{i\ell}=v} \mathbb{E}_{z_{i\ell}} 1_{z_{i\ell}=k} \log b_{kv} + \sum_{v=1}^{V} \gamma_v \log b_{kv}$$

and we recognize a Dirichlet with parameters

$$\widetilde{B}_{k:} \triangleq \left(\gamma_v + \sum_{i=1}^{N}\sum_{\ell=1}^{L_i} 1_{y_{i\ell}=v} \widetilde{z_{i\ell k}}\right)_{1\leq v\leq V}.$$

This concludes the derivation of VB for LDA.

5.2.2 Mean-field VB for marginal LDA

As an exercise, derive the updates for the marginalized LDA model of Section **??**; see Murphy, 2012, Chapter 27.3 for the solution.

5.2.3 VB generalizes the EM algorithm

TBC

5.3 Gradient-based algorithms

An alternate approach to finding a simple $\mathscr{Q}$ leading to closed-form updates is to directly run a gradient algorithm on the ELBO (5.4). Take $\mathscr{Q} = \{q(\cdot|\phi), \phi \in \Phi\}$, where for all x, $\phi \mapsto q(x|\phi)$ is differentiable. Then, assuming the necessary regularity conditions, Paisley, Blei, and Jordan, 2012 note that gradient of the ELBO can be rewritten using the so-called *score function trick* as

$$\begin{aligned}\nabla_\phi L(q) &= \nabla_\phi \mathbb{E}_{x\sim q(\cdot|\phi)} \log \frac{\pi_u(x)}{q(x|\phi)} \\ &= \int \log \pi_u(x) \nabla_\phi q(x|\phi) \mathrm{d}x + \nabla_\phi H[q(\cdot|\phi)]. \\ &= \mathbb{E}_{x\sim q(\cdot|\phi)} \left[\log \pi_u(x) \nabla_\phi \log q(x|\phi)\right] + \nabla_\phi H[q(\cdot|\phi)].\end{aligned}$$

The first term can be estimated by vanilla Monte Carlo. The entropy term can usually be differentiated in closed form; if not, it can be estimated by vanilla Monte Carlo as well. Overall, we can plug an unbiased estimator of the gradient of the ELBO in any stochastic gradient algorithm. More often than not, the ELBO as a function of ϕ is not convex, though, and one has to be happy with searching for local optima. Moreover, vanilla Monte Carlo estimators of (**??**) have been reported to have high variance even in simple models (Paisley, Blei, and Jordan, 2012).

Variance reduction for ELBO gradients has been a field of active research. Paisley, Blei, and Jordan, 2012 propose to use control variates, while D. P. Kingma and M. Welling, 2014 and a large body of follow-up work propose *reparametrization tricks* that work as follows. Assume that there exists a (deterministic) smooth and invertible function f such that $f(\varepsilon, \phi) \sim q(\cdot|\phi)$ whenever $\varepsilon \sim p(\varepsilon)$, with ε easy to sample. Now rewrite

$$\nabla_\phi L(q) = \nabla_\phi \mathbb{E}_{\varepsilon\sim p} \log \pi_u(f(\varepsilon, \phi)) + \nabla_\phi H[q(\cdot|\phi)].$$

This time the gradient can be passed under the integral in the first term, without relying on the score function trick, and we obtain

$$\nabla_\phi L(q) = \mathbb{E}_{\varepsilon\sim p} \nabla_\phi \log \pi_u(f(\varepsilon, \phi)) + \nabla_\phi H[q(\cdot|\phi)].$$

As long as we can compute gradients of $\log \pi_u$, we can compute the gradient in the expectation using the chain rule. This suggests a second vanilla Monte Carlo estimator, drawing $\varepsilon_i \sim p$ i.i.d. In practice, the resulting estimator has been found to have much lower variance (Rezende, Mohamed, and Wierstra, 2014), like in variational auto-encoders (D. P. Kingma and M. Welling, 2014). I haven't seen a completely convincing explanation why and when variance reduction happens with the reparametrization trick in general, though. Finally, note that we again assumed that the entropy of q could be differentiated in closed form, but the entropy term can also be treatead using the reparametrization trick if needed. We shall do so in the next section, for the sake of illustration.

5.3.1 VB for deep networks

One of the hot applications of gradient-based VB is for Bayesian deep learning, which has generated a huge literature in a short amount of time; see e.g. recent NeurIPS tutorials and workshops for pointers. For instance, Blundell et al., 2015 proceed as follows. We consider networks as generative models, so consider the softmax (classification) or squared (regression) loss. A network thus corresponds to a likelihood $p(\mathbf{y}|w)$. We take a prior $p(w)$ for the weights, and want to fit $q(w|\phi)$ to the posterior $\pi(w) \propto p(\mathbf{y}|w)p(w) = \pi_u(w)$. The gradient of the reparametrized ELBO writes

$$\begin{aligned}\nabla_\phi L(q(\cdot|\phi)) &= \mathbb{E}_\varepsilon \nabla_\phi \log \frac{\pi_u(f(\varepsilon, \phi))}{q(f(\varepsilon, \phi)|\phi)} \\ &\approx \frac{1}{N_\varepsilon} \sum_{i=1}^{N_\varepsilon} \nabla_\phi \log \frac{\pi_u(f(\varepsilon_i, \phi))}{q(f(\varepsilon_i, \phi)|\phi)}.\end{aligned}$$

Now notice that $\log \pi_u(w) = \log p(w) + \sum_{i=1}^{N_y} \log p(y_i|w)$, so that one can further uniformly draw (with or without replacement) a minibatch of data points B, and further obtain an unbiased estimator

$$\nabla_\phi L(q(\cdot|\phi)) \approx \frac{N_y}{N_\varepsilon |B|} \sum_{i=1}^{N_\varepsilon} \sum_{y \in B} \nabla_\phi \left[\frac{1}{N_y} \log p(f(\varepsilon_i, \phi)) + \log p(y|f(\varepsilon_i, \phi)) - \frac{1}{N_y} \log q(f(\varepsilon_i, \phi)|\phi) \right].$$

Note that following Blundell et al., 2015, we do not assume that the entropy can be differentiated in closed form, but the method applies *mutatis mutandis*. Note also that it is not obvious that replacing the entropy by its closed-form would reduce the variance of the estimator.

Now the key argument is that the gradient inside the sum can be computed using the chain rule, backprogagation, and the (assumed known) gradient of q. As an example, assume $w \in \mathbb{R}^d$, $\phi = (\mu, \sigma) \in \mathbb{R}^{d+1}$, so that $f(\phi, \varepsilon) = \mu + \sigma\varepsilon \sim \mathcal{N}(\mu, \sigma I_d)$. For $y \in B$, Let $F(w) = \log p(y|w) + \log p(w)$. The gradient of F is provided by backpropagation. Now the chain rule yields

$$\begin{aligned} \nabla_\phi (F(f(\varepsilon, \cdot))(\phi_0) &= J_{f(\varepsilon,\cdot)}(\phi_0)^T \nabla F(f(\varepsilon, \phi_0)) \\ &= \begin{pmatrix} I_d & \varepsilon \end{pmatrix}^T \nabla F(f(\varepsilon, \phi_0)). \end{aligned}$$

5.4 Theoretical guarantees

5.5 An alternative derivation of VB, and generalizations

We defined VB as minimizing $q \mapsto KL(q, \pi)$ over $\mathcal{Q}$. There is an equivalent definition that sheds some light on the relationship of VB with Bayes, and leads to natural generalizations of VB; see (Knoblauch, Jewson, and Damoulas, 2022) and references therein.

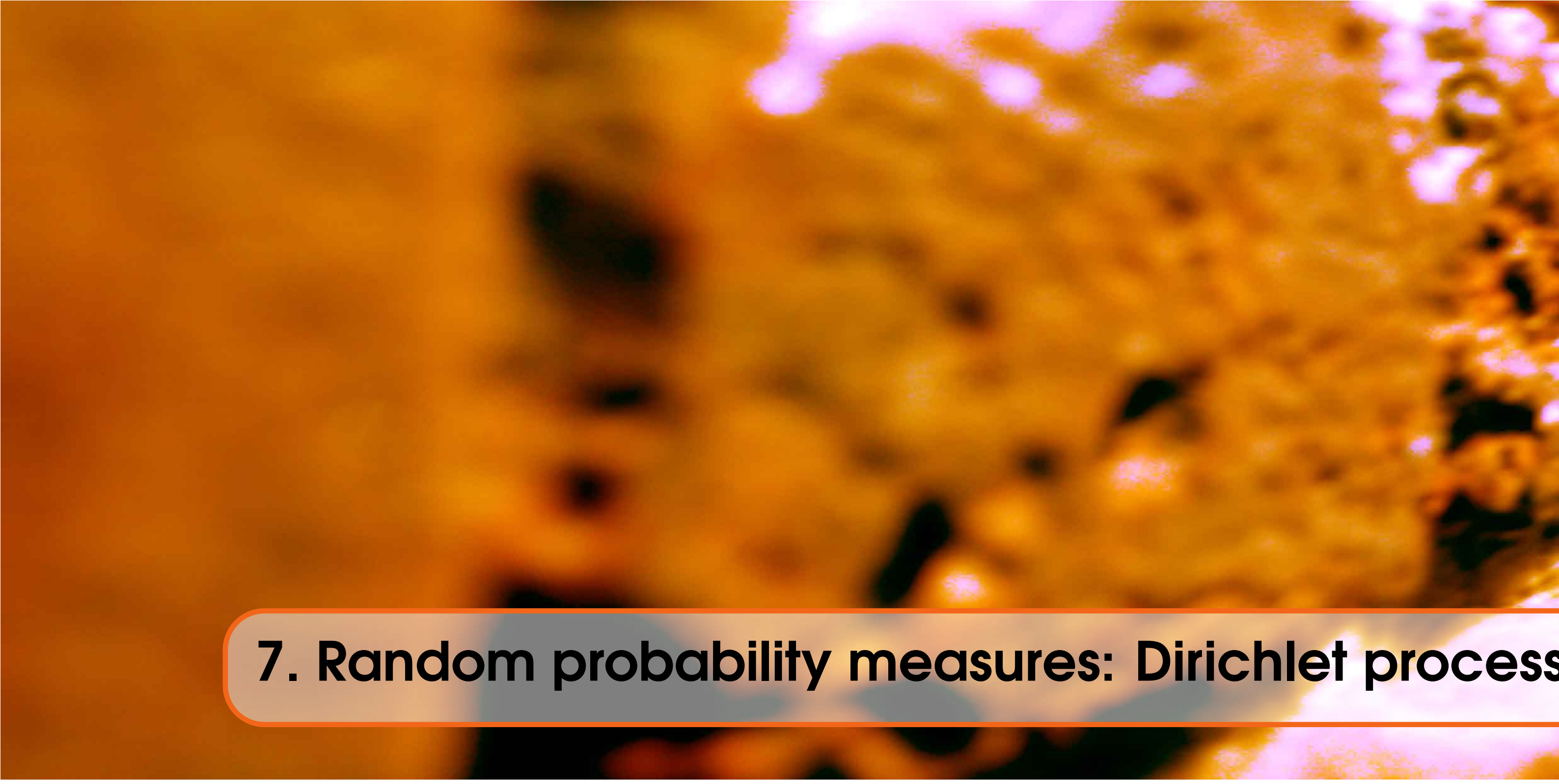
We consider the problem of inference for simplicity. Given a loss function ℓ, a divergence D, and a set of distributions $\mathcal{Q}$, consider the optimization problem $P(\ell, D, \mathcal{Q})$ defined by

$$q^\star \in \arg\min_{q \in \mathcal{Q}} \mathbb{E}_q \sum_{i=1}^{N} \ell(\theta, x_i) + D(q(\theta)\mathrm{d}\theta, p(\theta)d\theta). \qquad (P(\ell, D, \mathcal{Q}))$$

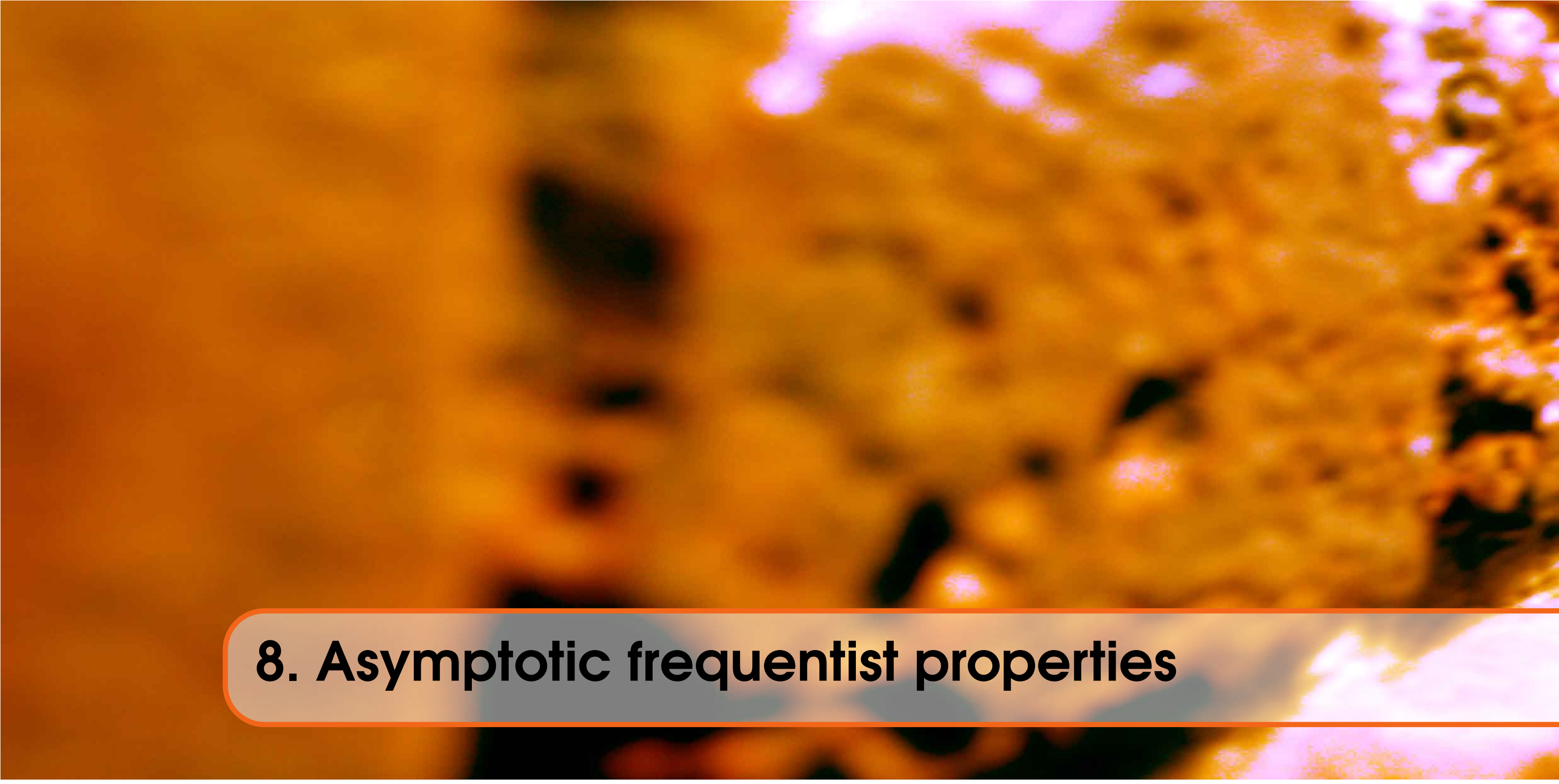
A classical result going back at least to (Donsker and Varadhan, 1975) implies that taking $\ell(\theta, x_i) = -\log p(y_{1:N}|\theta)$, $D = \mathrm{KL}$, and $\mathcal{Q} = \mathcal{M}_1$ leads to $q^\star(\theta) = \pi(\theta) \propto p(y_{1:N}|\theta)p(\theta)$. Moreover, taking $\mathcal{Q} \subsetneq \mathcal{M}_1$ leads to VB, i.e. $q^\star \in \arg\min \mathrm{KL}(q, \pi)$. In that sense, VB corresponds to the same optimization problem as Bayes, but with a constraint on where the "posterior" may lie. Knoblauch, Jewson, and Damoulas, 2022 examine justifications behind what they call the *optimization-centric* view on Bayesian inference, and how the choices of ℓ and D in ($P(\ell, D, \mathcal{Q})$) impacts the results of VB. In particular, when the prior has a structure that does not represent what we expect from a posterior, like when using an uncorrelated Gaussian prior on the weights of a neural network, while we expect large correlations across layers and multimodality from a posterior. Knoblauch, Jewson, and Damoulas, 2022 demonstrate that empirical performance in regression benefits from replacing the KL by Renyi's α-divergence. Interestingly, this benefit depends on α being in a certain finite range, which allows both a weaker dependence to the prior, and the support of the resulting $q^\star$ not collapsing to a single point.

IV Bayesian nonparametrics

6. Random functions: Gaussian processes

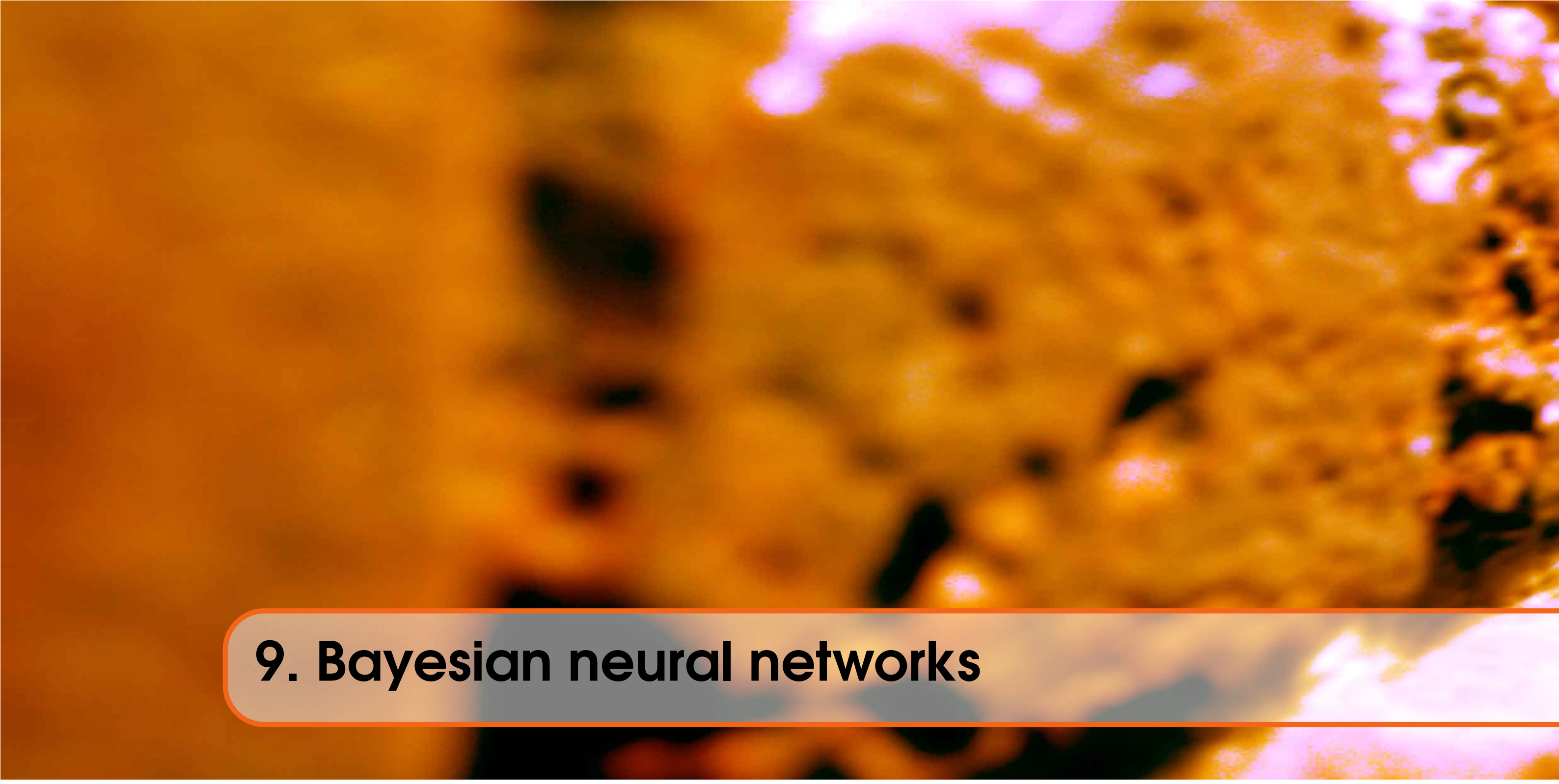


7. Random probability measures: Dirichlet process



8. Asymptotic frequentist properties

V Bayesian deep learning



9. Bayesian neural networks

9.1 Introduction

9.2 Prior distributions

9.2.1 Connection prior/initialization

9.2.2 Neural-network Gaussian process (NN-GP)

9.2.3 Neural tangent kernel (NTK)

9.2.4 Edge of Chaos

9.2.5 Unit priors get heavier with depth

9.2.6 Other priors

9.3 Posterior sampling algorithms

9.3.1 MCMC

9.3.2 Laplace approximation

9.3.3 Monte Carlo dropout

The dropout technique can be reinterpreted as a form of approximate Bayesian variational inference (Gal and Ghahramani, 2016; Diederik P Kingma, Salimans, and Max Welling, 2015). Gal and Ghahramani, 2016 build a connection between dropout and the Gaussian process representation, while Diederik P Kingma, Salimans, and Max Welling (2015) propose a way to interpret Gaussian dropout. They develop a *variational dropout* where each weight of a model has its individual dropout rate. *Sparse variational dropout*, proposed by Molchanov, Ashukha, and Vetrov (2017), extends *variational dropout* to all possible values of dropout rates and leads to a sparse solution. The approximate posterior is chosen to factorize either over rows or over individual entries of the weight matrices. The prior usually factorizes in the same way. Therefore, performing dropout can be used as a Bayesian approximation. However, as noted by Duvenaud et al. (2014), dropout has no regularization effect on infinitely-wide hidden layers. Nalisnick, Hernández-Lobato, and Smyth (2019) propose a Bayesian interpretation of regularization via multiplicative noise, with dropout being the particular case of Bernoulli noise. They find that noise applied to hidden units ties the scale parameters in the same way as the ARD algorithm (Neal, 1996), a well-studied shrinkage prior.

Let us describe Monte Carlo dropout in more details. The idea is simple and consists in performing random sampling at test time. Instead of turning off the dropout layers at test time (as is usually done), hidden units are randomly dropped out according to a Bernoulli(p) distribution. Repeating this operation M times provides M versions of the MAP estimate of the network parameters w^m, $m = 1,\dots,M$ (where some units of the MAP are dropped), yielding an approximate posterior predictive in the form of the equal-weight average:

$$p(y|x,\mathscr{D}^{\setminus}) \approx \frac{1}{M}\sum_{m=1}^{M} p(y|x,w^m). \tag{9.1}$$

Monte Carlo dropout captures some uncertainty from out-of-distribution (OOD) inputs, but is nonetheless incapable of providing valid posterior uncertainty. Indeed, Monte Carlo dropout changes the Bayesian model under study, which modifies also the properties of the approximate Bayesian inference performed. Specifically, Folgoc et al., 2021 shows that the Monte Carlo dropout posterior predictive (9.1) assigns zero probability to the true model posterior predictive distribution.

9.3.4 Variational inference

9.3.5 Expectation propagation

9.3.6 SGD-based methods

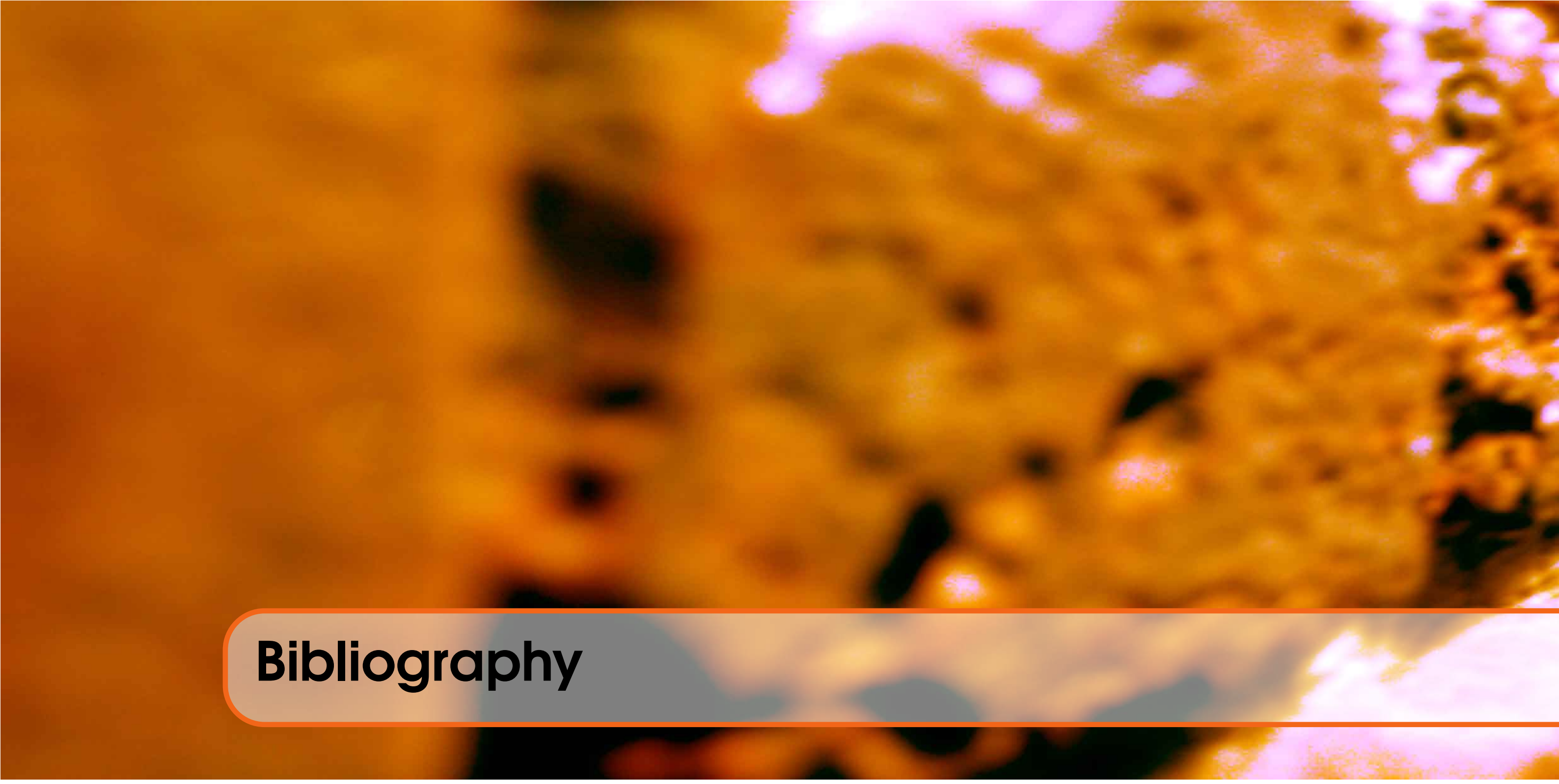
9.3.7 Last-layer methods

9.3.8 Deep ensembles

9.3.9 Cold posteriors

9.4 Generalization

9.4.1 PAC-Bayes



Bibliography

[1] J. O. Berger and R. L. Wolpert. *The likelihood principle: A review, generalizations, and statistical implications*. Volume 6. Institute of Mathematical Statistics, 1988 (cited on pages 19–21).

[2] C. Blundell et al. “Weight uncertainty in neural networks”. In: *International conference on machine learning*. PMLR. 2015, pages 1613–1622 (cited on pages 37, 38).

[3] N. Bou-Rabee and J. M. Sanz-Serna. “Geometric integrators and the Hamiltonian Monte Carlo method”. In: *Acta Numerica* 27 (2018), pages 113–206 (cited on pages 29, 30).

[4] J. Dick and F. Pilichshammer. *Digital Nets and Sequences. Discrepancy Theory and Quasi-Monte Carlo Integration*. Cambridge University Press, 2010 (cited on page 27).

[5] M. D. Donsker and S. R. S. Varadhan. “Asymptotic evaluation of certain Markov process expectations for large time, I”. In: *Communications on Pure and Applied Mathematics* 28.1 (1975), pages 1–47 (cited on page 38).

[6] R. Douc, É. Moulines, and D. Stoffer. *Nonlinear time series*. Chapman-Hall, 2014 (cited on pages 27, 28).

[7] David Duvenaud et al. “Avoiding pathologies in very deep networks”. In: *International Conference on Artificial Intelligence and Statistics*. 2014 (cited on page 49).

[8] P. C. Fishburn. *Utility theory for decision making*. Technical report. Research analysis corp. McLean VA, 1970 (cited on page 24).

[9] Loic Le Folgoc et al. “Is MC Dropout Bayesian?” In: *arXiv preprint arXiv:2110.04286* (2021) (cited on page 50).

[10] Yarin Gal and Zoubin Ghahramani. “Dropout as a Bayesian approximation: Representing model uncertainty in deep learning”. In: *International Conference on Machine Learning*. 2016 (cited on page 49).

[11] D. P. Kingma and M. Welling. “Stochastic gradient VB and the variational auto-encoder”. In: *Second International Conference on Learning Representations, ICLR*. Volume 19. 2014, page 121 (cited on page 37).

[12] Diederik P Kingma, Tim Salimans, and Max Welling. “Variational dropout and the local reparameterization trick”. In: *Advances in Neural Information Processing Systems*. 2015 (cited on page 49).

[13] J. Knoblauch, J. Jewson, and T. Damoulas. “An optimization-centric view on Bayes’ rule: Reviewing and generalizing variational inference”. In: *Journal of Machine Learning Research* 23.132 (2022), pages 1–109 (cited on page 38).

[14] D. Kreps. *Notes on the Theory of Choice*. Westview press, 1988 (cited on page 23).

[15] Dmitry Molchanov, Arsenii Ashukha, and Dmitry Vetrov. “Variational dropout sparsifies deep neural networks”. In: *International Conference on Machine Learning*. 2017 (cited on page 49).

[16] K. Murphy. *Machine learning: a probabilistic perspective*. MIT Press, 2012 (cited on pages 15, 35, 36).

[17] Eric Nalisnick, José Miguel Hernández-Lobato, and Padhraic Smyth. “Dropout as a structured shrinkage prior”. In: *International Conference on Machine Learning*. 2019 (cited on page 49).

[18] Radford M Neal. *Bayesian learning for neural networks*. Springer Science & Business Media, 1996 (cited on page 50).

[19] J. Paisley, D. M. Blei, and M. I. Jordan. “Variational Bayesian Inference with Stochastic Search”. In: *International Conference on Machine Learning (ICML)*. 2012 (cited on page 37).

[20] G. Parmigiani and L. Inoue. *Decision theory: principles and approaches*. Volume 812. John Wiley & Sons, 2009 (cited on pages 13, 21, 24).

[21] D. J. Rezende, S. Mohamed, and D. Wierstra. “Stochastic backpropagation and approximate inference in deep generative models”. In: *International conference on machine learning*. PMLR. 2014, pages 1278–1286 (cited on page 37).

[22] C. P. Robert and G. Casella. *Monte Carlo statistical methods*. Springer, 2004 (cited on page 27).

[23] L. J. Savage. *The Foundations of Statistics*. John Wiley & Sons, 1954 (cited on pages 21, 23).

[24] M. J. Schervish. *Theory of statistics*. Springer Science & Business Media, 2012 (cited on page 13).

[25] A. Wald. *Statistical decision functions*. Wiley, 1950 (cited on page 13).