Lecture notes on Bayesian machine learning

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

Rémi Bardenet and Julyan Arbel

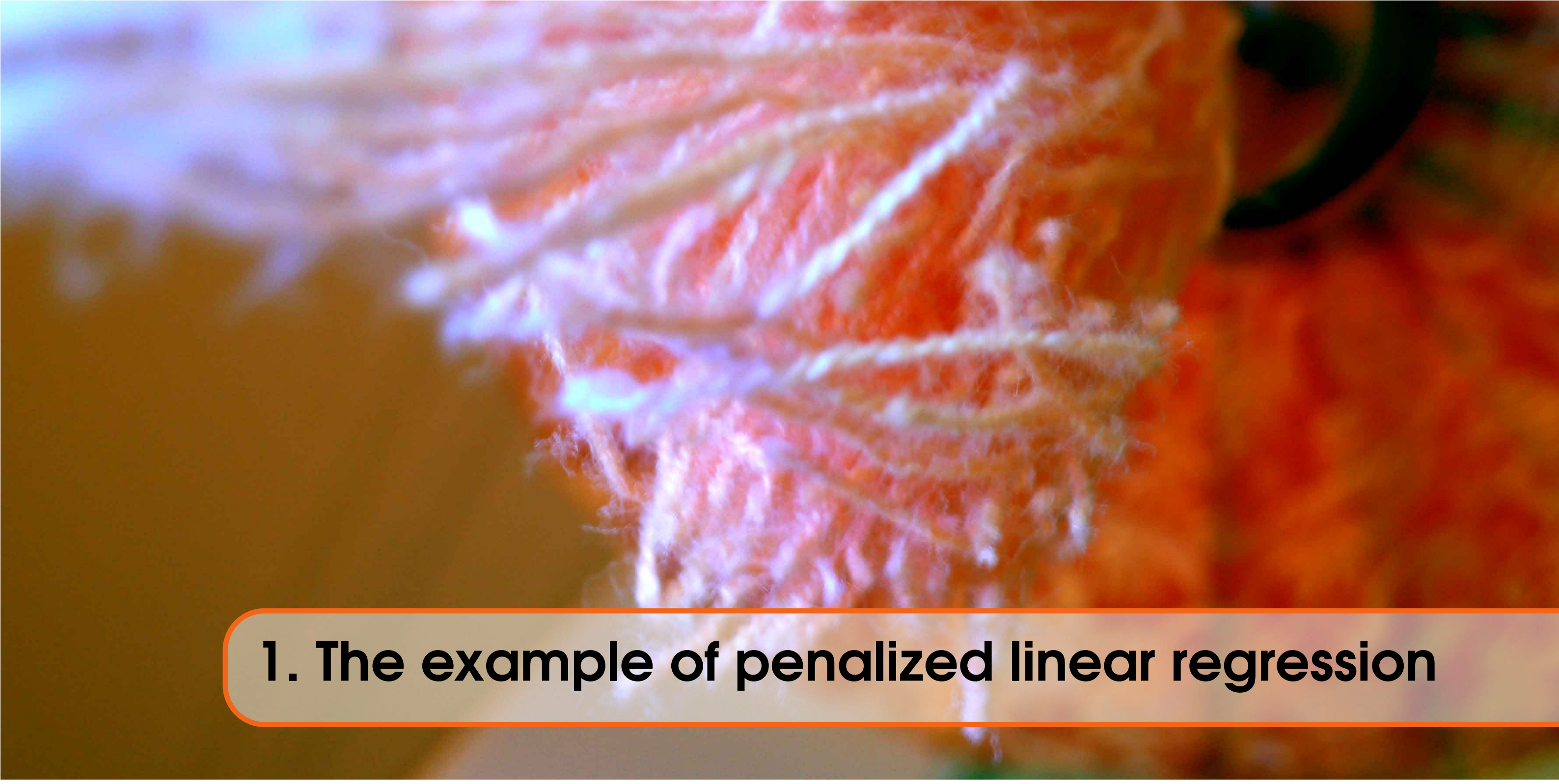
Contents

III Implementing Bayesian machine learning

IV Nonparametric Bayes

V Bayesian deep learning

I What is Bayesian machine learning?



1. The example of penalized 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.

	$\mathscr{S}$	$\mathscr{Z}$	$\mathscr{A}$	$u(r)$
Regression	$(\mathscr{X}\times\mathscr{Y})^n\times\mathscr{X}\times\mathscr{Y}$	$\mathscr{Y}=\mathbb{R}$	$\{a_g : s\mapsto y-g(x;x_{1:n},y_{1:n})\}$	$-\Vert r\Vert^2$
Classification	$(\mathscr{X}\times\mathscr{Y})^n\times\mathscr{X}\times\mathscr{Y}$	$\mathscr{Y}=\{0,1\}$	$\{a_g : s\mapsto y-g(x;x_{1:n},y_{1:n})\}$	$\mathbb{1}_{\{r=0\}}$
Point estimation	$\mathscr{Y}^n\times\Theta$	Θ	$\{a_g : s\mapsto \theta-g(y_{1:n})\}$	$-\Vert r\Vert^2$
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 chooses action

$$a^\star \in \arg\max_{a\in\mathscr{A}} \mathbb{E}_{s\sim p} u(a(s)). \tag{2.1}$$

The principle of decision-making encoded by (2.1) is called *expected utility*, or *subjective* expected utility, to insist on the fact that p is an arbitrary choice from the decision maker. At this stage, we 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 **??**.

In ML, it is customary to split the state variable into (s_O,s_U), where $O,U\subset\{1,\dots,\dim\mathscr{S}\}$ respectively index observed states (“data") and unknown states. In particular, we can rewrite (2.1) as

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

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 (2.2) over $\mathscr{A}$, it is enough maximize it over g. And since g is a function of s_O only, the optimal g is

$$g^\star : s_O \mapsto \arg\max_{g} \mathbb{E}_{s_U\vert s_O} u(a_g(s)). \tag{2.3}$$

Indeed, by maximizing the innermost expectation in (2.1) for each fixed value of s_O, we maximize the whole expectation. The resulting $g^\star$ is called the *Bayes decision rule*.

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 half-formal justification that deals primarily with the estimation problem. It is only half-formal because it deals with notions that are hard to rigorously define.

3.1.1 The formal LP

Consider two statistical experiments

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

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,

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

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 $\mathscr{S}=\Theta\times\cup_{n\geq1}\mathscr{Y}^n$, 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

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

3.4.2 Complete class theorems

3.4.3 PAC-Bayes statistical learning

III Implementing Bayesian machine learning

4. Markov chain Monte Carlo

4.1 Monte Carlo methods

Importance sampling, self-normalized importance sampling, SMC, SMC samplers, MCMC (discuss ergodicity and CLT).

4.2 The Metropolis-Hastings algorithm

Basic algorithm, detailed balance. Adaptive MH.

4.3 Gibbs sampling

4.4 Hamiltonian Monte Carlo

We closely follow **BoSa18**, who give a clear introduction to the ingredients of HMC. To facilitate going back and forth between the notes and (**BoSa18**), 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 $x = (p,q)$. 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 $\widetilde{\pi}(q,p) = \frac{1}{2}\mathcal{N}(p|0,M) \times \pi(q)$. Clearly, the p-marginal is Gaussian, while the q-marginal is π. If we manage to obtain an MCMC chain for $\widetilde{\pi}$, i.e. a chain with a Markov kernel that leaves $\widetilde{\pi}$ 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 $\widetilde{\pi}$: this is were Hamiltonian dynamics come into play.

4.4.3 Hamiltonian dynamics

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

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

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. $p' \sim \mathcal{N}(0,M)$. 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 (**BoSa18**), 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 (**BoSa18**)) 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 (**BoSa18**) 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.

5. Variational Bayes

While Monte Carlo methods are randomized numerical quadratures, *variational Bayes* (VB) stands for finding an approximation to the target distribution within some prespecified set $\mathscr{Q}$ of distributions. This is done by minimizing some notion of distance between the target π and the so-called variational approximation $q \in \mathscr{Q}$. For instance, a common choice is to solve

$$q^\star \in \arg\min_{q\in\mathscr{Q}} \mathrm{KL}(q,\pi) := \mathbb{E}_q \log \frac{q(\theta)}{\pi(\theta)} = \int q(\theta) \log \frac{q(\theta)}{\pi(\theta)} \mathrm{d}\theta, \tag{5.1}$$

where π and q are PDFs w.r.t. some common measure $\mathrm{d}\theta$. Once (5.1) has been solved, one can either use the resulting $q^\star$ as a plug-in replacement for the target π, or, say, use $q^\star(\theta)\mathrm{d}\theta$ as a proposal distribution in importance sampling.

Compared to MCMC, the big plus of VB is that some modification of the optimization problem (5.1) can often be implemented in large-scale settings where either the number of data items or the dimension of the problem are large, e.g. in deep learning. The major downside of VB is the difficulty to provide theoretical guarantees on its results. One reason for this is that the distance to minimize, such as the reverse KL divergence in (5.1), is often chosen for computational convenience rather than for its guarantees on integrating functions of interest. Another reason is that approximating complex target distributions requires a large set $\mathscr{Q}$ of variational approximations, which usually makes (5.1) a difficult optimization problem that has to be further modified before an efficient implementation is possible.

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 (5.1), 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*

$$J(q) = KL(q,\pi) - \log Z. \tag{5.2}$$

Proof. Left as an exercise. ■

Two remarks are in order. First, since Z does not depend on q, the optimization problem in (5.1) is equivalent to $\min J(q)$. Letting $L(q) = -J(q)$, (5.1) is further equivalent to $\max L(q)$. Second and the nonnegativity of the KL divergence implies that

$$L(q) \leq \log Z. \tag{5.3}$$

In Bayesian inference,

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

so that $Z = p(y_{1:N})$. Furthermore, (5.3) 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

$$\mathscr{Q} = \left\{ \theta \mapsto \prod_{d=1}^{D} q_d(\theta_d) \prod_{i=1}^{N} q_i(z_i) \right\}. \tag{5.4}$$

In other words, we approximate π with a separable PDF. Note that (5.4) 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

$$\phi_i(x_i) = \exp\left[\int q_{\setminus i}(x_{\setminus i}) \log \pi_u(x) \mathrm{d}x_{\setminus i}\right] \tag{5.5}$$

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 (5.5). 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 (5.4) is akin to assuming independence of all variables under the posterior. Combined with the fact that the log term in the reverse KL (5.1) favours a $q^\star$ with a support that includes that of π, this often implies a gross overestimation of the support of the target, along with the expected ignorance of 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

$$\begin{aligned}
&\log p(y, z, \pi, B) && (5.6)\\
&\quad = \sum_{i=1}^{N} \left[\log p(\pi_i | \alpha) + \sum_{\ell=1}^{L_i} \left(\log p(z_{i\ell} | \pi_i) + \log p(y_{i\ell} | z_{i\ell}, B) \right) \right] + p(B|\gamma) \\
&\quad \propto \sum_{i=1}^{N} \left[\sum_{k=1}^{K} \alpha_k \log \pi_{ik} + \sum_{\ell=1}^{L_i} \left(\sum_{k=1}^{K} 1_{z_{i\ell}=k} \log \pi_{ik} + \sum_{v=1}^{V} \sum_{k=1}^{K} 1_{y_{i\ell}=v} 1_{z_{i\ell}=k} \log b_{kv} \right) \right] \\
&\qquad + \sum_{k=1}^{K} \sum_{v=1}^{V} \gamma_v \log b_{kv}. && (5.7)
\end{aligned}$$

We want to fit a mean-field approximation

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

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 (5.5) for every coordinate, that is, we need to integrate the log joint distribution (5.7) 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

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

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

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

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 (5.3). 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, **PaBlJo12** 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 **PaBlJo12**.

Variance reduction for ELBO gradients has been a field of active research. **PaBlJo12** propose to use control variates, while **KiWe14** 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

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

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 (**ReMoWi14**), like in variational auto-encoders (**KiWe14**). 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.

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, **BCKW15** 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

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

Note that following **BCKW15**, 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

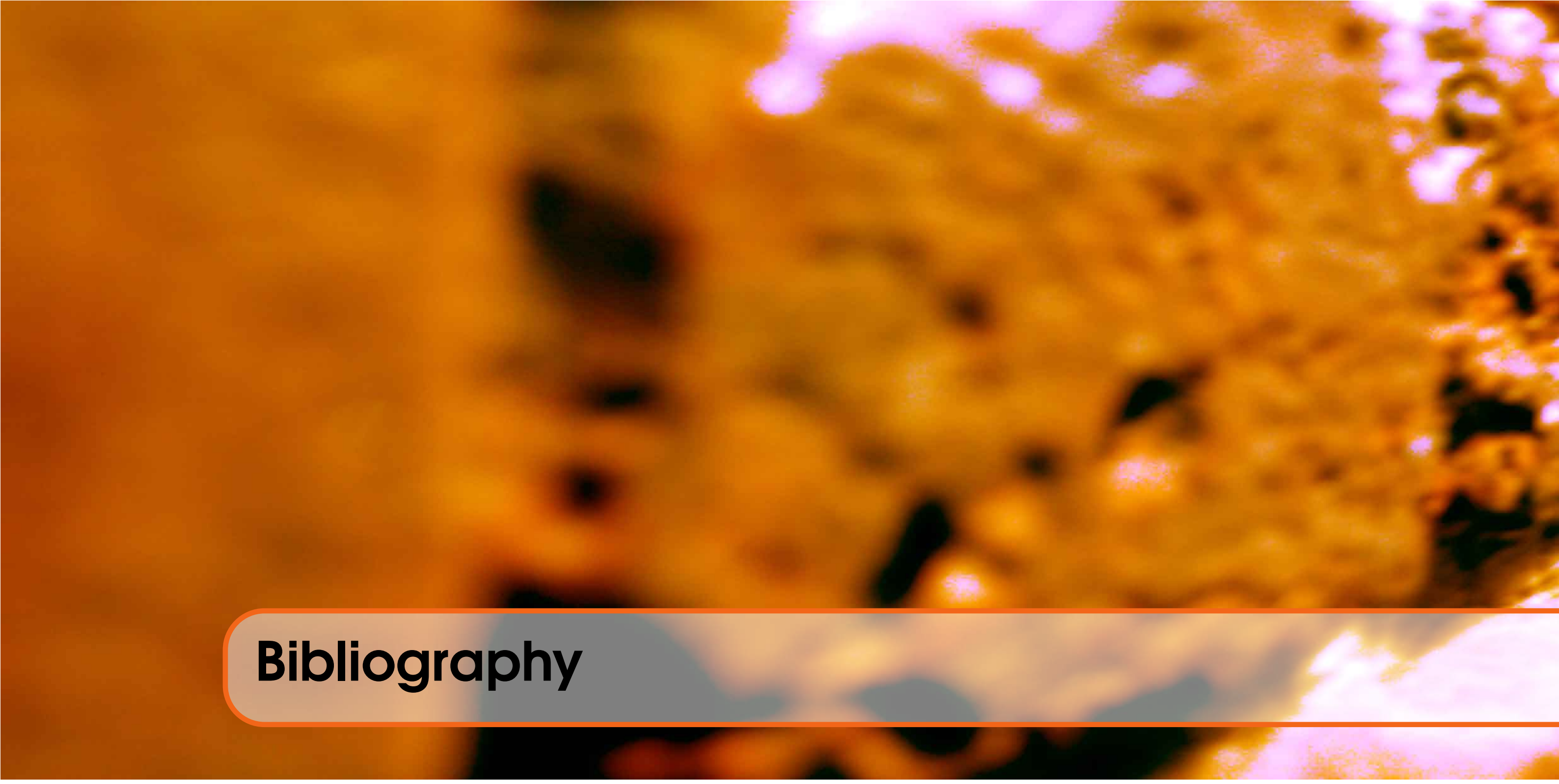
$$\begin{aligned} \nabla_\phi (F(f(\varepsilon, \cdot))(\phi_0) &= J_{f(\varepsilon,\cdot)}(\phi_0)^T \nabla F(f(\varepsilon, \phi_0)) \\ &= \begin{pmatrix} I_d & \varepsilon \end{pmatrix}^T \nabla F(f(\varepsilon, \phi_0)). \end{aligned}$$

5.4 Theoretical guarantees

5.5 Alternatives to the KL divergence

IV Nonparametric Bayes

V Bayesian deep learning



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 13–15).

[2] K. Murphy. *Machine learning: a probabilistic perspective*. MIT Press, 2012 (cited on pages 10, 25, 26).

[3] G. Parmigiani and L. Inoue. *Decision theory: principles and approaches*. Volume 812. John Wiley & Sons, 2009 (cited on page 9).

[4] M. J. Schervish. *Theory of statistics*. Springer Science & Business Media, 2012 (cited on page 9).

[5] A. Wald. *Statistical decision functions*. Wiley, 1950 (cited on page 9).