Bachelor Wiskunde

Bachelor thesis

Diverging Posteriors

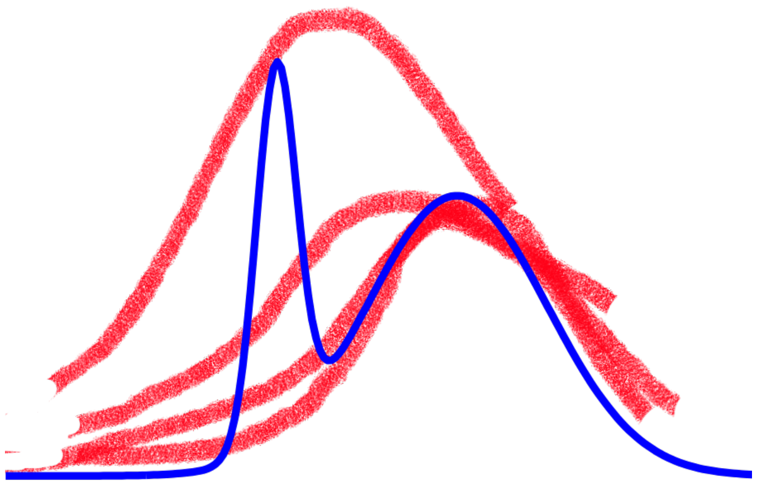
The practical-minded road to approximate inference

by

Victor van den Elzen

June 8, 2017

Supervisor: dr. Bartek Knapik



Department of Mathematics
Faculty of Sciences
VU UNIVERSITY AMSTERDAM

Abstract

First, we introduce Bayesian inference and the problem of computation. We consider symbolic methods, which are limited but fast, and sampling based methods, which are flexible but slow. Unsatisfied, we introduce the idea of approximate inference, which is behind Variational Bayesian inference and Expectation Propagation. As background we review Shannon entropy, Kullback-Leibler divergence and α-divergence. We explore that for approximate inference each divergence has different properties and different algorithms. For solving practical problems, one can now make an informed choice between various probabilistic programming languages.

Title: Diverging Posteriors
The practical-minded road to approximate inference
Author: Victor van den Elzen, ven500@student.vu.nl, 2509763
Supervisor: dr. Bartek Knapik
Date: June 8, 2017

Department of Mathematics
VU University Amsterdam
de Boelelaan 1081, 1081 HV Amsterdam
`http://www.math.vu.nl/`

Contents

1. Overview

A central problem in statistics is the problem of statistical inference: determining a distribution based on data and some model assumptions. We will focus only on Bayesian inference, which is the subject of Chapter 2. Bayesian inference is combining a prior distribution with the likelihood of the data using Bayes' theorem to create a posterior distribution. Given a model, data and prior distributions for the variables in the model we can in principle calculate the posterior distribution over those variables.

In Chapter 3 we look at the practical problems of calculating the normalizing constant, calculating certain characteristics and marginalizing nuisance variables that we are not interested in. All of these problems come down to integrating over the posterior distribution. This is difficult symbolically except in special cases. It is also difficult numerically except with very few variables because of the curse of dimensionality. We can sample using a Markov Chain Monte Carlo (MCMC) method, but in practice these methods are quite slow. It can even give the idea that Bayesian inference is impractical.

A different way to handle these problems is to approximate the distributions involved with easier distributions. From now we will focus on this way, which is much faster. The goal is to find the best approximation distribution from some tractable family, like the normal distributions. But what do we mean by best approximation? And how do we find it?

To answer these questions, in Chapter 4 we review Shannon entropy, Kullback-Leibler divergence and α-divergence. These divergences give us a way to talk about what distributions are good approximations.

In Chapter 5 we see that these divergences are useful in different cases. The exclusive Kullback-Leibler divergence prefers to approximate one mode. The inclusive Kullback-Leibler divergence prefers to approximate the mean and variance. The α-divergence is an extension that can choose in between the above two.

The method to optimize the exclusive Kullback-Leibler divergence is called Variational Bayesian inference, which we consider in Section 5.2. The method to optimize the inclusive Kullback-Leibler divergence is called Expectation Propagation, which we consider in Section 5.3. The method to optimize the exclusive Kullback-Leibler divergence is called Power EP, which we consider in Section 5.4.

With all this knowledge in hand, one can make an informed choice of probabilistic programming language for their inference problem. We list some for each method in Chapter 6.

2. Bayesian inference

2.1. Inference using Bayes' Theorem

The goal of statistical inference is to make statements about probabilities and distributions based on limited data points.

First, recall the following basic theorem of probability (originally [3], modern versions [28] §3.3):

Theorem 2.1 (Bayes' Theorem)**.** *Let A and B be events with $P(B) \neq 0$, then*

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}.$$

If we know all possible events $A_1, ..., A_n$ of some discrete sample space then

$$P(B) = \sum_{i=1}^{n} P(B|A_i)P(A_i),$$

which allows us to write

$$P(A_i|B) = \frac{P(B|A_i)P(A_i)}{\sum\limits_{i=1}^{n} P(B|A_i)P(A_i)}.$$

We are often interested in the probability of some event E_1 from a discrete sample space $\{E_1, ..., E_n\}$ given some data D, i.e. $P(E_1|D)$. Using Bayes' Theorem we can calculate this if we know two things:

- the probability of E_1 without the data, i.e. $P(E_1)$
- the probability of D given some event E_i, i.e. $P(D|E_i)$

That is called Bayesian inference.

Example 2.2. Say we bought a trick coin that lands on the same "trick" side 90% of the time. However, we do not know if the trick side is heads or tails, so we try it out and get heads. What is the probability that the trick side is heads?

Our events are E_1 = heads is the trick side and E_2 = tails is the trick side. Our sample space is $\{E_1, E_2\}$. Our data is D = we got heads.

To calculate $P(E_1|D)$, we first need to know the probability without any data $P(E_1)$. It seems reasonable to assume that in that case $P(E_1) = P(E_2)$. Since that's the whole sample space $P(E_1) + P(E_2) = 1$, so $P(E_1) = 0.5$. We know that $P(D|E_1) = 0.9$ and conversely $P(D|E_2) = 0.1$. Then

$$P(E_1|D) = \frac{P(D|E_1)P(E_1)}{P(D|E_1)P(E_1) + P(D|E_2)P(E_2)} = \frac{0.9 \cdot 0.5}{0.9 \cdot 0.5 + 0.1 \cdot 0.5} = 0.9.$$

We can write Bayes' Theorem in terms of density functions ([27] §1.2). If we have events $x \in X$ and $y \in Y$, we can replace $P(B|A_i)$ by $f(x|y)$ and $P(A_i)$ by $g(y)$ to get

Theorem 2.3 (Bayes' Theorem for discrete distributions)**.**

$$g(y|x) = \frac{f(x|y)g(y)}{\sum\limits_{y \in Y} f(x|y)g(y)}.$$

This extends without problems to continuous distributions as

Theorem 2.4 (Bayes' Theorem for continuous distributions)**.**

$$g(y|x) = \frac{f(x|y)g(y)}{\displaystyle\int_Y f(x|y)g(y)\mathrm{d}y}.$$

In Bayesian inference the different terms of Theorem 2.4 are named.

- $g(y|x)$ is called the *posterior* distribution.
- $f(x|y)$ is called the *likelihood* function or the sampling distribution.
- $g(y)$ is called the *prior* distribution.
- $\int_Y f(x|y)g(y)\mathrm{d}y$ is called the *normalization factor* or, in some contexts, the partition function.

The normalization factor is there so that $g(y|x)$ is a proper distribution with $\int_Y g(y|x)\mathrm{d}y = 1$. While it can be interesting, we usually leave it implicit as in

$$g(y|x) \propto f(x|y)g(y).$$

Here $\propto$ means that both sides are proportional, that is $g(y|x) = \kappa f(x|y)g(y)$ for some constant κ.

Example 2.5. Suppose we have a package and we want to know its weight. We pick it up and take it to a scale. Based on the feel we think it is about 5kg, plus or minus 500g. More formally, our prior is distributed normally like $\mathcal{N}(5, 0.5^2)$ with density function $g(y) \propto e^{-\frac{(y-5)^2}{2 \cdot 0.5^2}}$. The scale says it's 4kg, and we know the scale to be accurate to 200g. Formally, given a certain weight y, the results from the scale are distributed like $\mathcal{N}(y, 0.2^2)$. Then the likelihood of getting 4kg is

$f(4|y) \propto e^{-\frac{(4-y)^2}{2\cdot 0.2^2}}$. So the posterior for the weight being y based on our feel and the scale is

$$\begin{aligned}
g(y|4) \propto & f(4|y)g(y) \propto \\
& e^{-\frac{(4-y)^2}{2\cdot 0.2^2}} e^{-\frac{(y-5)^2}{2\cdot 0.5^2}} = \\
& e^{-\frac{(4-y)^2}{2\cdot 0.2^2}-\frac{(y-5)^2}{2\cdot 0.5^2}} = \\
& e^{-\frac{(0.2+0.5)y^2-2(5\cdot 0.2^2+4\cdot 0.5^2)y+5^2\cdot 0.2^2+4^2\cdot 0.5^2}{2\cdot 0.2^2\cdot 0.5^2}} \propto \\
& e^{-\frac{\left(y-\frac{5\cdot 0.2^2+4\cdot 0.5^2}{0.2^2+0.5^2}\right)^2}{2\frac{0.2^2\cdot 0.5^2}{0.2^2+0.5^2}}} = \\
& e^{-\frac{(y-4.138...)^2}{2\cdot 0.185...^2}}.
\end{aligned}$$

Our posterior is distributed like $\mathcal{N}(4.138..., 0.185...^2)$. We can see that our posterior is a combination of the prior and the likelihood (see Figure 2.1).

We see that the normal distribution has the nice property that when we multiply the density functions of two normal distributions, the result is again the density function of a normal distribution. In general if $n(\mu, \sigma^2)$ is the density function for $\mathcal{N}(\mu, \sigma^2)$, then

$$n\left(\frac{\mu_1\sigma_2^2+\mu_2\sigma_1^2}{\sigma_1^2+\sigma_2^2}, \frac{\sigma_1^2\sigma_2^2}{\sigma_1^2+\sigma_2^2}\right) \propto n\left(\mu_1, \sigma_1^2\right) n\left(\mu_2, \sigma_2^2\right).$$

We can simplify this by using the *precision* instead of the variance with the substitution $\tau = 1/\sigma^2$ giving

$$n\left((\mu_1\tau_1+\mu_2\tau_2)/(\tau_1+\tau_2), (\tau_1+\tau_2)^{-1}\right) \propto n\left(\mu_1, \tau_1^{-1}\right) n\left(\mu_2, \tau_2^{-1}\right).$$

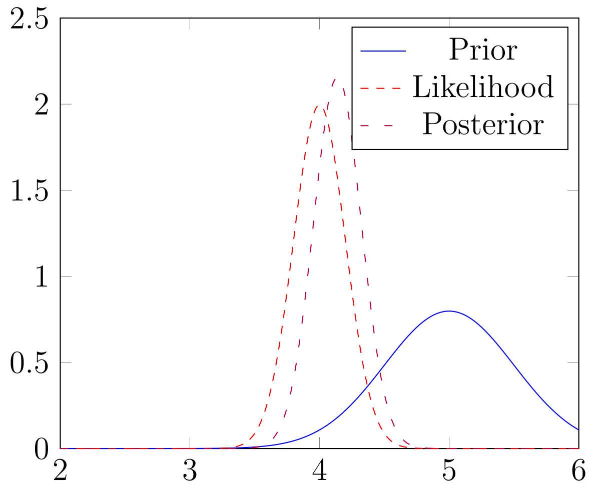
A nice property of Bayes' theorem is that when we have two independent pieces of data such that the likelihood $f(x_1, x_2|y) = f_1(x_1|y)f_2(x_2|y)$ we can do the computation in multiple steps:

$$g(y|x_1) \propto f_1(x_1|y)g(y)$$

$$g(y|x_1, x_2) \propto f_2(x_2|y)g(y|x_1).$$

Also the the order of these steps does not matter. This will become crucial in Section 5.3 when we consider Expectation propagation.

Figure 2.1.: Illustration of example 2.5



2.2. Reasonable priors

A key point in getting Bayesian inference off the ground is selecting a reasonable prior distribution. We can distinguish different kinds of priors:

- Uninformative or objective priors that try to be unbiased and minimize the information and influence of the prior.
  - Priors based on Laplace's Principle of Indifference, like those based on symmetry (such as in Example 2.2) and maximum entropy ([12], [27] §3.2.3).
  - Priors that minimize information like the Jeffrey's prior which minimizes Fisher information ([14], [27] §3.5.2).
  - Priors that maximize the influence of the data like reference priors ([4], [27] §3.5.4).
- Informative or subjective priors contain some information not captured by the model. This may be considered either problematic or helpful.
  - Priors based on "gut feelings" or experience that cannot be completely explained (such as in Example 2.5). These may be elicited from subject experts or be personal feelings ([27] §3.2.1).
  - Priors based on data that is related in a way that is not captured by the model. For example data from different time periods or similar locations may be used as starting points ([27] §3.2.1).
  - Priors based on the same data as used in the model, known as "Empirical Bayes". ([7], [27] §10.4). This is a bit strange because it is "double dipping". Empirical Bayes works best when doing many individual inferences on a large amount of data.

Some of these priors are not proper distributions and are known as “improper priors”.

The selection of a prior can be difficult not just mathematically, but also philosophically and when results are to be shared with others. One piece of good news is the Bernstein-von Mises theorem. It guarantees that when using parametric models with enough data, the posteriors will converge no matter what (non-pathological) prior is used. Practically, we can say that if we have a reasonable amount of data the prior is less important and we can get away with any of the above ones. One strategy is to do an analysis multiple times with different priors.

For the rest of the thesis we will assume a proper prior distribution, and make no more comments on prior selection.

3. Difficulties in practical computation of Bayesian inference

Looking back on Example 2.5, multiplying the distribution functions was easy. We were lucky enough to be able to recognize our posterior as a normal distribution, which meant we could find our normalization factor $\int_{-\infty}^{\infty} e^{-\frac{(y-4.138\ldots)^2}{2\cdot 0.185\ldots^2}} \mathrm{d}y$ easily: both the trick (looking at the integral multiplied by itself in polar coordinates) and the result ($\sqrt{2\pi\sigma^2}$) are well known.

However, so far this is only for normal priors and normal likelihoods. Calculating this integral in general is still a problem.

Furthermore, we are often interested not only in the posterior distribution, but also related values. For example the mean of the posterior distribution is of special interest. Again, this is an integral over the distribution, this time multiplied by the integrating variable: $\int_\Omega y g(y|x) \mathrm{d}y$, where Ω is the sample space of the distribution. Another thing we might like to look at is the distribution with some variables removed, by integrating them out (marginalization): $g(y_1|x) = \int_Y g(y_1, y_2|x) \mathrm{d}y_2$.

The computation of these kinds of integrals is a central problem in Bayesian inference.

3.1. Symbolic computation

We noticed in Example 2.5 that when our prior and likelihood are both normal distributions, then the posterior is also a normal distribution.

In general if we have a prior from family $\mathcal{P}$ and a likelihood from family $\mathcal{L}$, such that inference gives us a posterior from $\mathcal{P}$, then $\mathcal{P}$ is called a *conjugate prior* to $\mathcal{L}$. For example the family of normal distributions with a known variance is conjugate prior to itself. A different example is that the family of beta distributions is conjugate prior to the family of binomial distributions.

When we are in the conjugate prior situation Bayesian inference is quite easy symbolically. The distributions and integrals are well known, and we can repeat our inference procedure indefinitely. It is therefore tempting to use these priors when the likelihoods allow it. However this prior is not always the one we would choose otherwise. Further, practically we only have useful conjugate priors when the likelihoods are of an exponential family. That means distributions like binomial, Poisson, normal, Gamma and Beta that we can write as

$$f(x|y) = h(x)e^{\eta(y)\cdot T(x)-A(y)}$$

for some functions h, η, T and A. See [27] §3.3 for more information.

When we are not in the conjugate prior situation, symbolically solving the integrals that arise is at least difficult and often impossible. And even when the symbolic integral is possible, that does not always give us a practical way to compute the numerical value. See Appendix A for a practical implementation of the following examples.

Example 3.1. Consider the following model: we sample data generated from $\mathcal{C}(\theta, 1)$, a Cauchy distribution with location parameter θ. That gives us our likelihood. Our prior on θ is the standard normal distribution $\mathcal{N}(0, 1^2)$. Then the posterior given some data $x_1, ..., x_n$ is ([27] Ex 6.1.1)

$$g(\theta|x_1, ..., x_n) \propto e^{\frac{-\theta^2}{2}} \prod_{i=1}^{n} \frac{1}{1+(x_i-\theta)^2}.$$

The normalization constant $Z = \int_{-\infty}^{\infty} g(\theta|x_1, ..., x_n)\mathrm{d}\theta$ cannot be solved symbolically, even allowing special functions like `erf`, The mean $\int_{-\infty}^{\infty} \theta g(\theta|x_1, ..., x_n)\mathrm{d}\theta/Z$ can also not be solved symbolically.

Example 3.2. Consider the following scenario: to sample, first a biased coin is flipped, then we sample either the left or the right distribution according to the outcome of the coin. This is a mixture model, a special case of a hierarchical model. More explicitly call the weight of the biased coin p, the left distribution $\mathcal{N}(\mu_1, \sigma_1^2)$ and the right distribution $\mathcal{N}(\mu_2, \sigma_2^2)$. Let us combine the parameters as $\theta = (p, \mu_1, \sigma_1, \mu_2, \sigma_2)$. Then the likelihood is

$$f(x|\theta) \propto pe^{\frac{-(x-mu_1)^2}{2\sigma_1^2}} + (1-p)e^{\frac{-(x-mu_2)^2}{2\sigma_2^2}}.$$

Finding a prior $g(\theta)$ for all the parameters that makes it possible to symbolically do inference is possible using a combination of the normal, inverse-gamma and beta distributions ([27], Ex 6.1.5). Then, skipping over the details, the posterior given data $x_1, ..., x_n$ can be written as

$$\begin{aligned} f(\mu_1, \sigma_1, \mu_2, \sigma_2|x_1, ..., x_n) &\propto g(\theta) \prod_{i=1}^{n} \left(pe^{\frac{-(x-mu_1)^2}{2\sigma_1^2}} + (1-p)e^{\frac{-(x-mu_2)^2}{2\sigma_2^2}} \right) \\ &= \sum_{c\in\{\text{left,right}\}^n} \zeta(c) \end{aligned}$$

where $\zeta(c)$ is a complicated expression involving special functions for the normal, inverse-gamma and beta distributions. The most important point is that this posterior involves a sum over every possible choice of left and right for n times, for a total of 2^n summation terms. This means that for larger values of n it becomes practically impossible to actually compute this sum, let alone the normalization constant and the mean!

3.2. Numerical methods

When we can't use symbolic integration, we can try to use numerical methods.

For one-dimensional integrals we can simply use the trapezoidal rule or a more advanced technique like Runge-Kutta [31]. These involve sampling the function at regularly spaced points. For example, the total error for Runge-Kutta is proportional to the step size between points h, on the order of h^4. On an interval of length L the number of points is $n = L/h$. This means in practice that numerically computing one-dimensional integrals is very easy and fast.

For higher-dimensional integrals that we get from inference with multiple variables the situation is different. When we have d dimension and a box like $[0, L]^d$ then a grid of sample points has the number of points $n = (L/h)^d$. This means that for dimensions higher than about 4 the amount of points needed becomes too high to practically compute ([27], §6.2.1). Unfortunately different schemes to place points in a regular pattern don't overcome this "curse of dimensionality".

Fortunately, our analysis was about the *total* error, but we are usually more interested in the *relative* error. A grid can still be too much to ask even if we only have 2 points per dimensions, giving a total number of points $n = 2^d$. One try is to use random points $\theta_1, ..., \theta_n$ sampled from our prior $g(\theta)$ over our space Θ (for example the box $[0, L]^d$). Then the normalization factor is

$$\int_\Theta f(x|\theta)g(\theta)\mathrm{d}\theta \approx \frac{1}{n}\sum_{i=1}^{n} f(x|\theta_i).$$

This is called a *Monte Carlo method* due to its randomness, and converges almost surely as n goes to infinity. If the variance of $g(\theta|x)$ (the posterior) is finite, we can bound the relative error to the order $1/\sqrt{n}$ due to the Central Limit Theorem.

How does this Monte Carlo method do in practice? A big problem is that often many of the random points are in areas of low probability. This causes the convergence to be very slow due to high constant factors. We can try to sample more methodically using stratified sampling methods ([16] Ch. 8) or give the method a hint with an importance function ([27], §6.2.2). The most promising method is to pick up hints where to look from the sampled points themselves using *Markov Chain Monte Carlo.*

The idea of Markov Chain Monte Carlo (MCMC) is to create a Markov chain which is more likely to go towards areas of high density while still having the average of the points converge to the integral we are looking for ([27] §6.3). Fortunately, such Markov chains turn out to be fairly easy to create in various ways including Metropolis-Hastings methods and Gibbs sampling methods. They are usually many times faster than Monte Carlo methods and have excellent theoretical convergence properties. Practically, they are easy to implement and parallelizable. The success of MCMC for Bayesian inference has made them almost synonymous with Bayesian inference itself.

Still, MCMC is not perfect. It is not possible to say that any finite sample has converged to or is near the real answer. There are always unexplored areas of parameter space that may drastically alter the answer. Further, while MCMC is often fast enough to be practical it is still many times slower than simple symbolic solutions or Frequentist methods that were designed to be calculated by hand. While it is hard to say anything concrete because every problem is different, we can say that 2 to 10 seconds is on the low end [9], up to 30 minutes is possible for more complicated models with hundreds of thousand of data points, and finally there is advice that calculations should be run overnight at least and preferably as long as possible ([10] §1.11.3). These running times are too slow for (soft) real-time use, exploratory data analysis or fitting many different models or data.

3.3. Approximate inference - a third way

So far we have seen two methods of computing inference. The symbolic way can be very fast but is limited in the choice of models and priors. The MCMC way is somewhat slow but very flexible in the choice of models and priors. We would like to have something in between those extremes, that we can use when we need something more flexible than symbolic methods but significantly faster than MCMC.

This third way is *approximate inference*, which is the topic for the rest of the thesis. The idea is to approximate the all distributions by those that can easily be used symbolically, in practice almost always normal distributions. Then we want to find the best approximation to the posterior distribution.

Approximate inference is fast. For example, an approximate inference method called ADVI is “orders of magnitude faster than NUTS, a state-of-the-art MCMC algorithm” [17]. Where MCMC uses minutes, approximate inference uses seconds. This is a qualitative difference that enables application of Bayesian inference in situations where there is less time or where there are larger problems.

Approximate inference is fairly flexible. It is applicable when the posterior can be well approximated by a normal distribution. The likelihoods in many complicated models naturally tend to a normal distribution due to the central limit theorem. Example 3.2 is an instance of this. In such a situation, the Bernstein-von Mises theorem tells us that the posterior will also tend to a normal distribution. Furthermore, we often only want certain characteristics of the posterior like the mean or the variance. Then “well approximated” only means that those characteristics are close.

The big questions then that will be answered in the rest of the thesis are:

1. What do we mean exactly by “best approximation”?
2. How do we find the best approximation?

4. Entropy and divergence of distributions

4.1. Entropy

We will first examine the concept of *entropy* as defined by Shannon [29], also know as *Shannon entropy* or *Shannon information*. In the original formulation, entropy is given by the function $H(p)$ going from a discrete probability distribution p over $\{1, ..., n\}$ to a non-negative real number such that

1. H is continuous in the $p(i)$, and
2. if all $p(i)$ are the same $(1/n)$, then H is monotonically increasing in n, and
3. the entropy of a joint distribution of two independent distributions is the sum of the entropy of the two independent distributions.

This function is then

$$H(p) = -\sum_{i=1}^{n} p(i) \log_B p(i) \tag{4.1}$$

which is unique up to a multiplicative constant, namely the base B. B is conventionally set to 2 or e.

The motivation for this definition is in the context of coding information over some communication channel. With $B = 2$, the expected number of bits needed to communicate a single event from a discrete distribution using the optimal (minimal) code is its entropy.

A different explanation is that H is an increasing function of the number of ways to realize a distribution. Let's say we have m different objects and buckets of size $m_i = m/p(i)$, where we restrict m such that all m_i are integer. Dividing the objects between the buckets can be done in $W(m) = \frac{m!}{m_1! \ldots m_n!}$ different ways. Then ([13] §11.4)

$$\lim_{m\to\infty} \frac{1}{m} \log W(m) = H(p).$$

The extension of the definition of H to a continuous distribution $p(x)$ called *differential entropy* is

$$H(p) = -\int_{-\infty}^{\infty} p(x) \log_B p(x) \mathrm{d}x. \tag{4.2}$$

There is a subtlety here that Equation 4.2 is not the limit of Equation 4.1 for finer and finer discrete distributions p, which would be infinite [12].

Example 4.1. The normal distribution is $p(x) = (2\pi\sigma^2)^{-1/2}e^{-\frac{(x-\mu)^2}{2\sigma^2}}$. Its entropy in base e is

$$\begin{aligned}
H(p) &= -\int_{-\infty}^{\infty} p(x)\log p(x)\mathrm{d}x \\
&= -\int_{-\infty}^{\infty} p(x)\log\left((2\pi\sigma^2)^{-1/2}e^{-\frac{(x-\mu)^2}{2\sigma^2}}\right)\mathrm{d}x \\
&= -\int_{-\infty}^{\infty} p(x)\left(-\frac{1}{2}\log(2\pi\sigma^2) + -\frac{(x-\mu)^2}{2\sigma^2}\right)\mathrm{d}x \\
&= \frac{1}{2}\log(2\pi\sigma^2)\int_{-\infty}^{\infty} p(x)\mathrm{d}x + \frac{1}{2\sigma^2}\underbrace{\int_{-\infty}^{\infty}(x-\mu)^2 p(x)\mathrm{d}x}_{\text{variance}} \\
&= \frac{1}{2}\log(2\pi\sigma^2) + \frac{1}{2\sigma^2}\sigma^2 \\
&= \frac{1}{2}\log(2\pi\sigma^2 e).
\end{aligned}$$

4.2. Kullback-Leibler divergence

Building on the idea of entropy, Kullback and Leibler defined the "information in x for discrimination" between two distributions p and q as $\log\frac{p(x)}{q(x)}$ [18].

Then the "mean information for discrimination" over discrete p is

$$\mathrm{KL}(p \mid\mid q) = \sum_{i=1}^{n} p(i)\log_B\frac{p(i)}{q(i)}.$$

This is what we now call the *Kullback-Leibler divergence* from q to p.

We can see this as a more general version of entropy. Using a uniform distribution $u(x) = 1/n$, the entropy $H(p) = \log_B n - \mathrm{KL}(p \mid\mid u)$. Let us interpret this in terms of coding in base 2. $H(p)$ is the optimal number of bits to encode p. $\log_2 n$ is the optimal number of bits to encode a uniform distribution. Then $\mathrm{KL}(p \mid\mid u)$ is the number of bits wasted when using the uniform code instead of the optimal code for p. In general, $\mathrm{KL}(p \mid\mid q)$ is the number of bits wasted when using the optimal code for q to encode p.

Just like entropy, a different explanation is that $\mathrm{KL}(p \mid\mid q)$ is a monotone function of the number of ways to realize p, but now weighted by the distribution q. The weighted number of ways is $W_q(m) = \frac{m!}{m_1!\dots m_n!}q(1)^{m_1}\dots q(n)^{m_n}$. Then ([15] §3.1.3)

$$\lim_{m\to\infty}\frac{1}{m}\log W_q(m) = -\mathrm{KL}(p \mid\mid q).$$

For continuous p and q the definition of Kullback-Leibler divergence is

$$\mathrm{KL}(p \mid\mid q) = \int_{-\infty}^{\infty} p(x)\log_B\frac{p(x)}{q(x)}\mathrm{d}x.$$

Kullback-Leibler divergence has many nice properties. One of them is that $\mathrm{KL}(p \,||\, q) \geq 0$ with $\mathrm{KL}(p \,||\, q) = 0$ if and only if $p = q$ almost everywhere, which is the definition of a *divergence*. However, the Kullback-Leibler divergence is not symmetric, $\mathrm{KL}(p \,||\, q) \neq \mathrm{KL}(q \,||\, p)$, and it does not obey the triangle inequality in either direction. So it is not a *metric*.

The asymmetry will be considered in depth when applying the Kullback-Leibler divergence to approximating distributions.

Example 4.2 (Moment matching)**.** What is the best normal approximation to a distribution $p(x)$? One answer (see Section 5.1) is that it is the normal distribution with parameters μ and σ^2 that minimizes $\mathrm{KL}(p \,||\, \phi(x|\mu, \sigma^2))$, where

$$\phi(x|\mu,\sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\dfrac{(x-\mu)^2}{2\sigma^2}}.$$

We can solve this by finding the stationary points, where the derivatives with respect to μ and σ^2 are zero. Let's look at the derivative with respect to μ:

$$\begin{aligned}
\frac{\mathrm{d}}{\mathrm{d}\mu}\mathrm{KL}(p \,||\, \phi(x|\mu,\sigma^2)) &= \frac{\mathrm{d}}{\mathrm{d}\mu} \int_{-\infty}^{\infty} p(x) \log \frac{p(x)}{\phi(x|\mu,\sigma^2)} \mathrm{d}x \\
&= \frac{\mathrm{d}}{\mathrm{d}\mu} - \int_{-\infty}^{\infty} p(x) \log \phi(x|\mu,\sigma^2) \mathrm{d}x \\
&= \frac{\mathrm{d}}{\mathrm{d}\mu} \int_{-\infty}^{\infty} p(x) \frac{(x-\mu)^2}{2\sigma^2} \mathrm{d}x \\
&= -\frac{1}{\sigma^2} \int_{-\infty}^{\infty} p(x)(x-\mu) \mathrm{d}x \\
&= -\frac{1}{\sigma^2} \left(\int_{-\infty}^{\infty} x p(x) \mathrm{d}x - \mu \right).
\end{aligned}$$

This derivative is equal to zero exactly when $\mu = \int_{-\infty}^{\infty} x p(x) \mathrm{d}x$. The second derivative is $1/\sigma^2 > 0$, so this is indeed a minimum. That means that the mean of the best normal approximation is the same as the mean of p.

In a similar way we can find that the optimal variance σ^2 for this μ is exactly the variance of p. That completely determines the normal distribution.

Approximating p with a normal distribution with the same mean and variance is called *moment matching*.

4.3. Amari α-divergence

In the field of information geometry, we have a further generalization of entropy and Kullback-Leibler divergence called α-divergence [1]. The simplest formula for this divergence is [33]

$$\mathrm{D}_\alpha(p \mid\mid q) = \frac{1}{(1-\alpha)\alpha}\left(1 - \int_{-\infty}^{\infty} p(x)^\alpha q(x)^{1-\alpha}\mathrm{d}x\right).$$

This is a divergence because $\mathrm{D}_\alpha(p \mid\mid q) \geq 0$ with $\mathrm{D}_\alpha(p \mid\mid q) = 0$ if and only if $p = q$ almost everywhere. The values for $\alpha = 0$ and $\alpha = 1$ are defined by continuous extension, which turn out to be

$$\mathrm{D}_1(p \mid\mid q) = \mathrm{KL}(p \mid\mid q), \text{ and}$$
$$\mathrm{D}_0(p \mid\mid q) = \mathrm{KL}(q \mid\mid p).$$

In this way the α-divergence is an extension of the Kullback-Leibler divergence, which interpolates between the two directions for $\alpha \in (0, 1)$.

The other special value is $\alpha = 1/2$, which is the only value for which the α-divergence is symmetric. In general we have a sort of dual symmetry $\mathrm{D}_\alpha(p \mid\mid q) = \mathrm{D}_{1-\alpha}(q \mid\mid p)$.

4.4. Other divergences

The Kullback-Leibler divergence is very important and the basis for basically all approximate inference methods. However, there are other divergences that could be used. One of the drawbacks of the Kullback-Leibler divergence is that it is not symmetric, and thus not a metric. The original paper [18] suggests the symmetrized version

$$\mathrm{KL}(p \mid\mid q) + \mathrm{KL}(q \mid\mid p).$$

A different symmetrized version is Jensen-Shannon divergence [19]

$$\mathrm{KL}(p \mid\mid \frac{p+q}{2}) + \mathrm{KL}(q \mid\mid \frac{p+q}{2}).$$

We encountered the Hellinger distance already in disguise: [25] [33]

$$\mathrm{Hel}^2(p, q) = \frac{1}{2}\int_{-\infty}^{\infty}\left(\sqrt{p(x)} - \sqrt{q(x)}\right)^2 \mathrm{d}x = \frac{1}{4}\mathrm{D}_{1/2}(p \mid\mid q).$$

The Wasserstein distance or the Earth Mover Distance is a metric that has recently been used in machine learning problems [2]:

$$\inf_{\gamma \in \Pi(p,q)} \mathrm{E}_{(x,y)\sim y}\|x - y\|.$$

One might also look to different generalizations of the Kullback-Leibler divergence such as the confusingly similar Rényi α-divergence [8]:

$$\frac{1}{1-\alpha}\int_{-\infty}^{\infty} p(x)^\alpha q(x)^{1-\alpha}\mathrm{d}x.$$

This divergence is also equal to to $\mathrm{KL}(p \mid\mid q)$ when $\alpha = 1$, but unlike the Amari α-divergence it has no connection to the Kullback-Leibler divergence for $\alpha = 0$.

5. Approximations using different divergences

Now that we know some divergences we can try to answer the first question raised in section 3.3: what do we mean exactly by “best approximation”?

The standard answer is that we mean the approximation with the minimal Kullback-Leibler divergence to the true posterior. However, there is an ambiguity here. Let’s say we have a posterior $g(y|x)$ and an approximation $\tilde{g}(y|x)$ from some tractable family like the normal distribution. Should we minimize $\mathrm{KL}(g \mid\mid \tilde{g})$ or $\mathrm{KL}(\tilde{g} \mid\mid g)$? Following [23] we will call $\mathrm{KL}(g \mid\mid \tilde{g})$ the *inclusive* and $\mathrm{KL}(\tilde{g} \mid\mid g)$ the *exclusive* divergence, for reasons made clear below.

From an interpretational point of view, $\mathrm{KL}(g \mid\mid \tilde{g})$ seems to make the most sense. We often consider the first argument of the Kullback-Leibler divergence to be the “true” distribution, and the second argument the mistaken or simplified model (e.g. [15]).

From a practical point of view, we will examine the inclusive and exclusive divergence on an example and see that they are appropriate for different characteristics. Then we will examine the optimization problems generated by the inclusive and exclusive divergence.

5.1. Inclusive and exclusive divergence

The difference between the inclusive and exclusive divergence is most clear in an example that can’t be well-approximated by a normal distribution. In Figure 5.1, we see what happens with the best approximation of a mixture of two normal distributions by a single normal distribution.

We can see that the inclusive divergence tries to capture both parts of the mixture while the exclusive divergence only captures one part. This is exaggerated at more

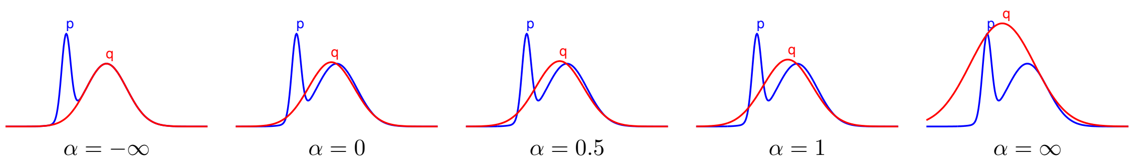


Figure 5.1.: The best normal approximation for a mixture for different values of α. $\alpha = 0$ corresponds to the exclusive divergence. $\alpha = 1$ corresponds to the inclusive divergence. Image copied from [23].

extreme values of α. We can explain this by looking at the form of the divergences.

For the inclusive divergence

$$\mathrm{KL}(g \,||\, \tilde{g}) = \int_{-\infty}^{\infty} g(y|x) \log \frac{g(y|x)}{\tilde{g}(y|x)} \mathrm{d}x,$$

the divergence goes to infinity when $\tilde{g}$ is zero but g is not zero in some region. So the best approximation will have mass anywhere the true posterior has mass. We call this behavior *zero avoiding.*

On the other hand for the exclusive divergence

$$\mathrm{KL}(\tilde{g} \,||\, g) = \int_{-\infty}^{\infty} \tilde{g}(y|x) \log \frac{\tilde{g}(y|x)}{g(y|x)} \mathrm{d}x$$

the divergence goes to infinity when g is zero but $\tilde{g}$ is not zero in some region. So the best approximation will have *no* mass anywhere the true posterior has *no* mass. We call this behavior *zero forcing.*

These behaviors extend to the α-divergence

$$\mathrm{D}_{\alpha}(g \,||\, \tilde{g}) = \frac{1}{(1-\alpha)\alpha} \left(1 - \int_{-\infty}^{\infty} \frac{g(x)^{\alpha}}{\tilde{g}(x)^{\alpha-1}} \mathrm{d}x \right).$$

When $\alpha > 1$, the α-divergence is zero avoiding. When $\alpha < 0$, the α-divergence is zero forcing.

In terms of characteristics, the inclusive divergence is the one that has the correct mean and variance [23]. In fact the normal distribution that minimizes the inclusive divergence has the same mean and variance as the true posterior. That is known as moment matching, see Example 4.2. However, the maximum (mode) of the inclusive divergence is often very different from the true posterior. The exclusive divergence gives us a distribution where the maximum is close to a local maximum of the true posterior, namely the maximum of the part of the mixture with the most weight.

These properties give us an idea of when we should use each divergence. We will come back to this choice when examining the best α for α-divergence minimization.

5.2. Variational Bayesian inference (exclusive)

The approximate inference method that optimizes the exclusive divergence is called variational Bayesian inference, sometimes Variational Inference (VI) or Variational Bayes (VB) for short. The optimization problem is

$$\arg\min_{\tilde{g} \in \mathcal{F}} \mathrm{KL}(\tilde{g} \,||\, g).$$

Here $\mathcal{F}$ is some tractable family. Although more complicated families are possible, to be concrete we will use the so called *mean-field approximation* where our approximations are fully factored normal distributions. Then

$$\tilde{g}(y|x) = \tilde{g}(\mu_1, ..., \mu_n, \sigma_i, ..., \sigma_n|x_1, ..., x_n) = \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi\sigma_i^2}} e^{-\frac{(x_i-\mu_i)^2}{2\sigma_i^2}}$$

for some μ_i, σ_i which are the variables that we will optimize.

Now the first problem is that we do not know what $g(y|x)$ is exactly. Indeed the whole point is to avoid having to calculate the normalization factor $Z_g = \int_Y f(x|y)g(y)\mathrm{d}y$! However we know the denormalized part $f(x|y)g(y) = g(y|x)Z_g$. So let us rewrite our optimization objective in terms of the what we know:

$$\begin{aligned}
\mathrm{KL}(\tilde{g} \mid\mid g) &= \int_Y \tilde{g}(y|x) \log \frac{\tilde{g}(y|x)}{g(y|x)} \mathrm{d}y \\
&= \int_Y \tilde{g}(y|x) \log \tilde{g}(y|x) \mathrm{d}y - \int_Y \tilde{g}(y|x) \log g(y|x) \mathrm{d}y \\
&= \int_Y \tilde{g}(y|x) \log \tilde{g}(y|x) \mathrm{d}y - \int_Y \tilde{g}(y|x) \log \frac{g(y|x)Z_g}{Z_g} \mathrm{d}y \\
&= \int_Y \tilde{g}(y|x) \log \tilde{g}(y|x) \mathrm{d}y - \int_Y \tilde{g}(y|x) \log(g(y|x)Z_g) \mathrm{d}y + \int_Y \tilde{g}(y|x) \log Z_g \mathrm{d}y \\
&= \int_Y \tilde{g}(y|x) \log \tilde{g}(y|x) \mathrm{d}y - \int_Y \tilde{g}(y|x) \log(g(y|x)Z_g) \mathrm{d}y + \log Z_g.
\end{aligned}$$

Because the Kullback-Leibler divergence is non-negative

$$\mathrm{ELBO}(\tilde{g}) = \int_Y \tilde{g}(y|x) \log(g(y|x)Z_g) \mathrm{d}y - \int_Y \tilde{g}(y|x) \log \tilde{g}(y|x) \mathrm{d}y$$

is a lower bound for $\log Z_p$, where ELBO stands for "evidence lower bound" [5]. Also because $\log Z_g$ is constant with respect to $\tilde{g}$, for optimization we can use the ELBO like

$$\arg\min_{\tilde{g}\in\mathcal{F}} \mathrm{KL}(\tilde{g} \mid\mid g) = \arg\max_{\tilde{g}\in\mathcal{F}} \mathrm{ELBO}(\tilde{g}).$$

Now the last term of the ELBO is just the entropy of $\tilde{g}$ which is usually well known. In our specific case it is the sum of the entropy of normal distributions, $\sum_{i=1}^{n} \log \sqrt{2\pi e \sigma_i^2}$.

So the last problem is the term $\int_Y \tilde{g}(y|x) \log(g(y|x)Z_g)\mathrm{d}y$. Sometimes $g(y|x)Z_g = f(x|y)g(y)$ has a form that is simplified by the logarithm, like a product of exponential family distributions:

$$\log \prod_{i=1}^{n} h_i(x) e^{\eta_i(y)\cdot T_i(x) - A_i(y)} = \sum_{i=1}^{n} \left(\log h_i(x) + \eta_i(y) \cdot T_i(x) - A_i(y)\right).$$

In that case, we can often symbolically find an optimization scheme [32]. For example, our mean-field approximation lets us derive a coordinate descent scheme [5]. But even when $g(y|x)Z_g$ is complicated, we can use a Monte Carlo method to find a noisy but unbiased estimate of the gradient of the ELBO with respect to the y [17] [26]. Then using these estimated gradients we can optimize $\tilde{g}$ using stochastic gradient descent [6].

One final remark is that unfortunately this optimization problem is generally not convex [5]. It is not the fault of the Kullback-Leibler divergence which is convex in both its arguments, but of our space $\mathcal{F}$. A linear combination of normal distributions is not a normal distribution again, but rather a mixture of normal distributions. This means that the optimization methods above are not guaranteed to find the global optimum. Indeed, often they will find a local optimum instead. The exact optimum found depends on the starting point and the details of the optimization process. Luckily these local optima are usually close to the global optimum, so the method is still practical.

5.3. Expectation Propagation (inclusive)

The approximate inference method that optimizes the inclusive divergence is called Expectation Propagation (EP). The optimization problem is

$$\arg\min_{\tilde{g}\in\mathcal{F}} \mathrm{KL}(g \mid\mid \tilde{g}).$$

Like in 5.2 we will take $\mathcal{F}$ to be the tractable family of fully factored normal distributions.

The trick used in 5.2 doesn't quite work the same way. Let's write out our optimization objective in terms of $g(y|x)Z_g$:

$$\begin{aligned}\mathrm{KL}(g \mid\mid \tilde{g}) &= \int_Y g(y|x)\log\frac{g(y|x)}{\tilde{g}(y|x)}\mathrm{d}y \\ &= \int_Y g(y|x)\log g(y|x)\mathrm{d}y - \int_Y g(y|x)\log\tilde{g}(y|x)\mathrm{d}y \\ &= \int_Y g(y|x)\log g(y|x)\mathrm{d}y - \frac{1}{Z_g}\int_Y g(y|x)Z_g\log\tilde{g}(y|x)\mathrm{d}y\end{aligned}$$

The first term is the entropy of the true posterior, which we can ignore for optimization purposes because it is constant with respect to $\tilde{g}$. That leaves an unknown multiplicative constant, and $\int_Y g(y|x)Z_g\log\tilde{g}(y|x)\mathrm{d}y$, a different integral than in 5.2. In this case the logarithm does not simplify $g(y|x)Z_g$, so this integral is just as hard if not harder than the integral for Z_g. A numerical method based on unbiased estimators of the gradients of $g(y|x)$ with respect to the y has not been developed in the literature.

Instead of trying to optimize the inclusive divergence directly, Expectation Propagation optimizes the *dual* optimization problem [21] [30] based on the dual form of the Kullback-Leibler divergence

$$\mathrm{KL}(g \mid\mid \tilde{g}) = \int_Y g(y|x)\frac{g(y|x)}{\tilde{g}(y|x)}\mathrm{d}y = \max_v \int_Y g(y|x)v(y)\mathrm{d}y - \log\int_Y \tilde{g}(y|x)e^{v(y)}\mathrm{d}y.$$

It was not immediately recognized that this was the case, because it's quite technical. Instead, Expectation Propagation was developed based on a simpler heuristic idea.

One of the first thing that comes to mind when considering approximate inference is to simply approximate the prior $g(y)$ and the likelihood $f(x|y)$ with normal distributions, for example using moment matching. Then $g(y|x) \approx \tilde{g}(y|x) \propto \tilde{f}(x|y)\tilde{g}(y)$. This works but the approximation is poor. When we have multiple independent data points this idea turns into

$$\tilde{g}(y|x) \propto \tilde{g}(y) \prod_{i=1}^{n} \tilde{f}(x_i|y). \tag{5.1}$$

An improvement is instead to approximate $g(y)$ with $\tilde{g}(y)$, then try to compute the moments of $\tilde{g}(y)f(x_1|y)$ and moment match them with a normal distribution to produce $\tilde{g}(y|x_1)$. Then we can repeat the procedure with $\tilde{g}(y|x_1)$ and $f(x_2|y)$ to produce $\tilde{g}(y|x_1, x_2)$, and so forth until $\tilde{g}(y|x_1, ..., x_n)$. This gives the *Assumed Density Filtering* algorithm [20]. Unfortunately this algorithm is sensitive to the order of the data, and the approximation still isn't very good.

The idea of the Expectation Propagation is to start at an initial approximation, like Equation 5.1, and then keep improving [20]. Because the prior $g(y)$ has no special computational status compared to the $f(x_i|y)$, we define $t_0(y) = g(y)$ and $t_i(y) = f(x_i|y)$. Then $g(y|x_1, ...x_n) \propto \prod_{i=0}^{n} t_i(y)$. We call $\tilde{t}^{\setminus i} = \prod_{j \neq i} \tilde{t}_j(y)$ the cavity distribution of t_i; it can efficiently be computed as $\tilde{g}/\tilde{t}_i$. To improve an approximation $\tilde{t}_i$ for some i, we first compute a new $\tilde{g}$ by moment matching $t_i\tilde{t}^{\setminus i}$ with a normal distribution. Then we update $\tilde{t}_i$ to be the unique normal distribution such that $\tilde{g} \propto \prod_{i=0}^{n} \tilde{t}_i(y)$, or equivalently $\tilde{t}_i \propto \tilde{g}/\tilde{t}^{\setminus i}$. To finish the algorithm, we simply keep improving $\tilde{t}_i$'s until convergence.

The only computational difficulty in the algorithm is matching the moments of $t_i\tilde{t}^{\setminus i}$ with a normal distribution. We only need the expectation and the variance. This can be done symbolically for many distributions. For example, Infer.NET has full support for doing Expectation Propagation using normal, Bernoulli and Poisson distributions among others [24]. A different approach is to compute the expectation and variance numerically [11].

Expectation Propagation does not always converge. When it does not converge, or converges slowly, this is an indication that the posterior is not well approximated. However, a convergent algorithm may be required, so there exist extensions which always converge [11].

5.4. α-divergence optimization

Our final optimization problem is

$$\arg\min_{\tilde{g} \in \mathcal{F}} \mathrm{D}_\alpha(g \mid\mid \tilde{g}).$$

This generalizes both the exclusive and the inclusive optimization problems. We might expect to need a new, difficult method to solve this. But in fact we can solve this optimization problem with a simple modification of Expectation Propagation, called Power EP [22].

Intuitively, we look only at a fraction α of each $\tilde{t}_i$ when improving it. We define the modified cavity distribution $\tilde{t}_\alpha^{\setminus i} = (\tilde{t}_i)^{1-\alpha} \prod_{j \neq i} \tilde{t}_j$; it can efficiently be computed as $\tilde{g}/(\tilde{t}_i)^\alpha$. To improve an approximation $\tilde{t}_i$ for some i, we first compute a new $\tilde{g}$ by moment matching $(t_i)^\alpha \tilde{t}^{\setminus i}$ with a normal distribution. Then we update $\tilde{t}_i$ to be the normal distribution proportional to $(\tilde{g}/\tilde{t}_\alpha^{\setminus i})^{1/\alpha}$.

This algorithm works for any α other than $\alpha = 0$, the exclusive divergence. We can approximate $\alpha = 0$ with very small values of α, and in the limit $\alpha \to 0$ this converges [11]. This modification can also make the convergence better or the algorithm easier to compute.

A modern version of α-divergence optimization is called Black-Box α-Divergence Minimization [11]. As the name implies, this method works on arbitrary likelihoods and priors by using numerical methods, such as Monte Carlo estimation, automatic differentiation and stochastic gradient descent. This method is tailored to large data sets, using a simplification called factor tying. This simplification is accurate in the case of many independent data points, being equal in the limit to infinity. It serves to make the method convergent and memory-efficient.

In their paper [11] Hernández-Lobato et al. perform several experiments to see which value of α gives the best result for various problems. The first experiment is a probit regression where $\alpha = 1$ is very slightly best using the log-likelihood loss (Figure 5.4). The second experiment is a neural network regression where $\alpha = 0.5$ is best using the log-likelihood loss (Figure 5.4). The third experiment is a neural network classification where $\alpha = -1$ is best using the average test error and the log-likelihood loss (Figure 5.4). The final experiment is a neural network regression where $\alpha = 0.5$ is best using the average test error and the log-likelihood loss (Figure 5.4).

As we see there is no single best α value, although $\alpha = 0.5$ (the Hellinger distance) seems like a decent all-round choice. However, it is clearly valuable to have the ability to choose α according to each problems needs, whether it needs more attention to modes or means.

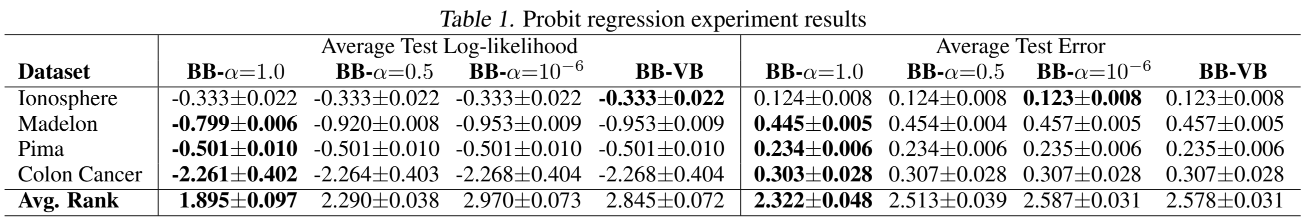


Table 1. Probit regression experiment results

	Average Test Log-likelihood				Average Test Error			
Dataset	**BB-α=1.0**	**BB-α=0.5**	**BB-α=10^{-6}**	**BB-VB**	**BB-α=1.0**	**BB-α=0.5**	**BB-α=10^{-6}**	**BB-VB**
Ionosphere	-0.333±0.022	-0.333±0.022	-0.333±0.022	**-0.333±0.022**	0.124±0.008	0.124±0.008	**0.123±0.008**	0.123±0.008
Madelon	**-0.799±0.006**	-0.920±0.008	-0.953±0.009	-0.953±0.009	**0.445±0.005**	0.454±0.004	0.457±0.005	0.457±0.005
Pima	**-0.501±0.010**	-0.501±0.010	-0.501±0.010	-0.501±0.010	**0.234±0.006**	0.234±0.006	0.235±0.006	0.235±0.006
Colon Cancer	**-2.261±0.402**	-2.264±0.403	-2.268±0.404	-2.268±0.404	**0.303±0.028**	0.307±0.028	0.307±0.028	0.307±0.028
Avg. Rank	**1.895±0.097**	2.290±0.038	2.970±0.073	2.845±0.072	**2.322±0.048**	2.513±0.039	2.587±0.031	2.578±0.031

Figure 5.2.: First Black-Box α-Divergence Minimization experiment results, copied from [11].

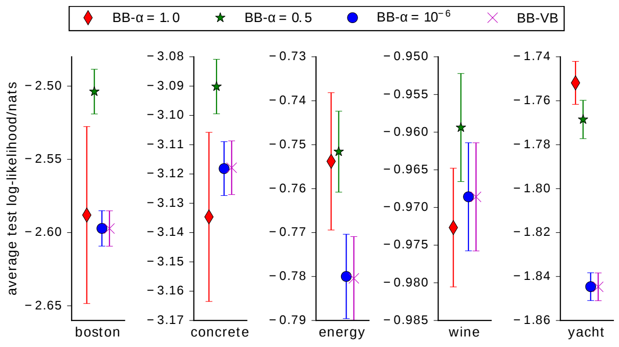


Figure 5.3.: Second Black-Box α-Divergence Minimization experiment results, copied from [11].

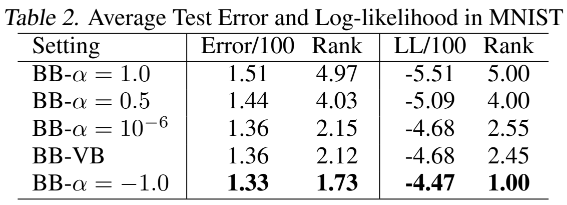


Table 2. Average Test Error and Log-likelihood in MNIST

Setting	Error/100	Rank	LL/100	Rank
BB-$\alpha = 1.0$	1.51	4.97	-5.51	5.00
BB-$\alpha = 0.5$	1.44	4.03	-5.09	4.00
BB-$\alpha = 10^{-6}$	1.36	2.15	-4.68	2.55
BB-VB	1.36	2.12	-4.68	2.45
BB-$\alpha = -1.0$	**1.33**	**1.73**	**-4.47**	**1.00**

Figure 5.4.: Third Black-Box α-Divergence Minimization experiment results, copied from [11].

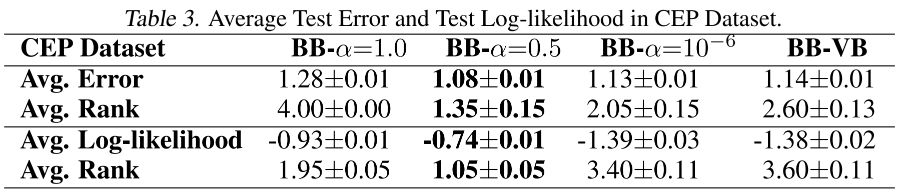


Table 3. Average Test Error and Test Log-likelihood in CEP Dataset.

CEP Dataset	**BB-α=1.0**	**BB-α=0.5**	**BB-α=10^{-6}**	**BB-VB**
Avg. Error	1.28±0.01	**1.08±0.01**	1.13±0.01	1.14±0.01
Avg. Rank	4.00±0.00	**1.35±0.15**	2.05±0.15	2.60±0.13
Avg. Log-likelihood	-0.93±0.01	**-0.74±0.01**	-1.39±0.03	-1.38±0.02
Avg. Rank	1.95±0.05	**1.05±0.05**	3.40±0.11	3.60±0.11

Figure 5.5.: Fourth Black-Box α-Divergence Minimization Black-Box Alpha experiment results, copied from [11].

6. Probabilistic programming languages

We considered several methods for Bayesian inference in this thesis. In passing, a few computer tools were mentioned that can help us apply these methods. The following tools allow us to describe our model and data in some way and then calculate probabilities and distributions. They are called *probabilistic programming languages*.

For symbolic methods to inference, we can use general purpose computer algebra systems like Mathematica, Maple and Maxima. There is also a more specialized tool called PSI Solver (2016, `http://psisolver.org`).

For Markov Chain Monte Carlo methods we have several high-quality options:

- BUGS (1989, `http://www.mrc-bsu.cam.ac.uk/software/bugs/`)
- JAGS (2007, `http://mcmc-jags.sourceforge.net`)
- Stan (2012, `http://mc-stan.org`)
- emcee (2012, `http://dan.iel.fm/emcee/`)

For variational methods (exclusive divergence) we have:

- ADVI (2015, in Stan `http://mc-stan.org`)
- Edward (2016, `http://edwardlib.org`)

For Expectation Propagation (inclusive divergence) we have Infer.NET (2008, `http://infernet.azurewebsites.net`).

One can implement any of these algorithms in any programming language. It may be helpful to use numerical libraries like Eigen and NumPy/SciPy, or machine learning libraries like Tensorflow and Theano.

7. Summary

We looked at Bayesian inference and its computational problems.

When the model allows for it, we can get exact results using symbolic methods. For low-dimensional problems we can use standard numerical integration. For higher-dimensional problems we can use stochastic sampling methods such as Markov Chain Monte Carlo. When MCMC is too slow, we turn to approximate methods. We can approximate a mode by optimizing the exclusive divergence using Variational Inference. We can approximate the mean and variance by optimizing the inclusive divergence using Expectation Propagation. And we can approximate anything in between by optimizing the α-divergence using Power EP.

Finally, we can use a probabilistic programming language to actually perform the computation using the chosen method.

Bibliography

[1] Shun'ichi Amari, *Differential-Geometrical Methods in Statistic*, 1985, Springer

[2] Martin Arjovsky, Soumith Chintala, Léon Bottou, *Wasserstein GAN*, 2017, `https://arxiv.org/abs/1701.07875`

[3] Thomas Bayes and Richard Price, *An Essay towards solving a Problem in the Doctrine of Chances*, Philosophical Transactions of the Royal Society of London, **53**, 370–418, 1763, `https://dx.doi.org/10.1098%2Frstl.1763.0053`

[4] Jose M. Bernardo, *Reference Posterior Distributions for Bayesian Inference*, 1979, Journal of the Royal Statistical Society, Series B. **41**, 113–147, `http://www.uv.es/~bernardo/1979JRSSB.pdf`

[5] David M. Blei, Alp Kucukelbir, Jon D. McAuliffe *Variational Inference: A Review for Statisticians*, 2016, `https://arxiv.org/abs/1601.00670`

[6] Léon Bottou, *Large-Scale Machine Learning with Stochastic Gradient Descent*, 2010, Proceedings of COMPSTAT'2010, `http://leon.bottou.org/publications/pdf/compstat-2010.pdf`

[7] Bradley Efron, 2010, *Large-Scale Inference: Empirical Bayes Methods for Estimation, Testing and Prediction*, `http://statweb.stanford.edu/~ckirby/brad/LSI/monograph_CUP.pdf`

[8] Tim van Erven, *Rényi Divergence and Kullback-Leibler Divergence*, 2014, IEEE Transactions on Information Theory, 60 (7) `https://arxiv.org/abs/1206.2459`

[9] Andrew Gelman, *Stan is fast*, 2012, `http://andrewgelman.com/2012/08/30/stan-is-fast/`

[10] Charles J. Geyer, *Introduction to MCMC*, 2011, Handbook of Markov Chain Monte Carlo, `http://www.mcmchandbook.net/HandbookChapter1.pdf`

[11] José Miguel Hernández-Lobato, Yingzhen Li, Mark Rowland, Daniel Hernández-Lobato, Thang Bui, Richard E. Turner, *Black-box α-divergence Minimization*, 2016, Proceedings of The 33rd International Conference on Machine Learning, PMLR 48:1511-1520, `https://arxiv.org/abs/1511.03243`

[12] E. T. Jaynes, *Information Theory and Statistical Mechanics*, 1957, The Physical Review, **106** 4, 620-640, `http://bayes.wustl.edu/etj/articles/theory.1.pdf`

[13] E. T. Jaynes, G. Larry Bretthorst, *Probability Theory: the Logic of Science*, 2003, Cambridge

[14] H. Jeffreys, *An Invariant Form for the Prior Probability in Estimation Problems*, 1946, Proceedings of the Royal Society of London. Series A, Mathematical and Physical Sciences. **186**, 453–461, `https://dx.doi.org/10.1098%2Frspa.1946.0056`

[15] Sadanori Konishi, Genshiro Kitagawa, *Information Criteria and Statistical Modeling*, 2007, Springer

[16] Dirk P. Kroese, Thomas Taimre, Zdravko I. Botev, *Handbook of Monte Carlo Methods* 2011, Wiley

[17] A. Kucukelbir, D. Tran, R. Ranganath, A. Gelman and D.M. Blei, *Automatic Differentiation Variational Inference*, 2017, Journal of Machine Learning Research, 18(14), `http://jmlr.org/papers/volume18/16-107/16-107.pdf`

[18] S. Kullback and R.A. Leibler, *On information and sufficiency*, 1951, Annals of Mathematical Statistics, 22 (1): 79–86, `https://projecteuclid.org/euclid.aoms/1177729694`

[19] J. Lin, *Divergence measures based on the shannon entropy*, 1991. IEEE Transactions on Information Theory. 37 (1): 145–151. `https://www.cise.ufl.edu/~anand/sp06/jensen-shannon.pdf`

[20] Thomas Minka, *A family of algorithms for approximate Bayesian inference*, 2001, Thesis (Ph.D.)–Massachusetts Institute of Technology, `https://dspace.mit.edu/handle/1721.1/86583`

[21] Thomas Minka, *The EP energy function and mimization schemes*, 2001, `https://tminka.github.io/papers/ep/minka-ep-energy.pdf`

[22] Thomas Minka, *Power EP*, 2004, Microsoft Research Technical Report MSR-TR-2004-149, `https://www.microsoft.com/en-us/research/publication/power-ep/`

[23] Tom Minka, *Divergence Measures and Message Passing*, 2005, Microsoft Research Technical Report MSR-TR-2005-173, `https://www.microsoft.com/en-us/research/publication/divergence-measures-and-message-passing/`

[24] *Infer.NET: List of factors and constraints*, `http://infernet.azurewebsites.net/docs/list%20of%20factors%20and%20constraints.aspx`

[25] M.S. Nikulin, *Hellinger distance*, Encyclopedia of Mathematics, `http://www.encyclopediaofmath.org/index.php?title=Hellinger_distance&oldid=16453`

[26] Rajesh Ranganath, Sean Gerrish, David M. Blei, *Black Box Variational Inference*, 2014,
http://www.cs.columbia.edu/~blei/papers/RanganathGerrishBlei2014.pdf

[27] Christian P. Robert, *The Bayesian Choice, 2nd ed.*, 2007, Springer

[28] Sheldon Ross, *A First Course in Probability, 8th ed.*, 2010, Pearson

[29] Claude E. Shannon, *A Mathematical Theory of Communication*, 1948, Bell System Technical Journal, 27 (3), http://math.harvard.edu/~ctm/home/text/others/shannon/entropy/entropy.pdf

[30] Martin J. Wainwright and Michael I. Jordan, *Graphical Models, Exponential Families, and Variational Inference*, 2008, Foundations and Trends® in Machine Learning: Vol. 1: No. 1–2, pp 1-305,
https://people.eecs.berkeley.edu/~wainwrig/Papers/WaiJor08_FTML.pdf

[31] Eric W. Weisstein, *Runge-Kutta Method*, MathWorld–A Wolfram Web Resource,
http://mathworld.wolfram.com/Runge-KuttaMethod.html

[32] John Winn, Christopher M. Bishop, *Variational Message Passing*, 2005, Journal of Machine Learning Research 6 (661–694),
http://www.jmlr.org/papers/volume6/winn05a/winn05a.pdf

[33] Huaiyu Zhu and Richard Rohwer, *Information Geometric Measurements of Generalisation*, 1995, Technical Report NCRG/4350, Aston University,
http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.38.881

A. Difficult cases for symbolic inference

We try two difficult symbolic inference problems in PSI Solver commit `da79ca`. For the Cauchy problem (Example 3.1) the integral is not solved. For the mixture problem (Example 3.2) the time and memory usage grows quickly with the data size, making it practically impossible to compute the solution for more than ten data points.

Listing A.1: Cauchy problem

```
def main() {
    theta := gauss(0, 1);
    d := cauchy(theta, 1);
    return d;
}
```

Listing A.2: Mixture example

```
def main() {
    p := uniform(0, 1);
    a := [0.1, 0.2, 0.3, 0.4];
    for i in [0..a.length) {
        left := gauss(0, 1);
        right := gauss(1, 1);
        if(flip(p)) {
            cobserve(right, a[i]);
        } else {
            cobserve(left, a[i]);
        }
    }
    return p;
}
```