Variational Inference and Optimization I

Arto Klami
University of Helsinki, Finland

ProbAI, June 15, 2021

Introduction

- Assistant Professor at Department of Computer Science, University of Helsinki, Finland
- Leading the *Multi-source Probabilistic Inference* group[1]
- Research topics: Statistical machine learning, approximate Bayesian inference, ML applications in ultrasonics, spectral imaging and others

FCAI Finnish Center for Artificial Intelligence

REAL AI FOR REAL PEOPLE IN THE REAL WORLD

REAL AI

Real AI solutions will bring AI systems closer to human cognition. We do not attempt to replicate human cognition but provide ambitious and realistic technical solutions.

REAL PEOPLE

Real People will use AIs as companions and assistants – and AIs need guidance from people. AIs we create can interact with humans in ways that are understandable, trustworthy, and ethical.

REAL WORLD

For AIs to be able to perform efficiently in the Real World, they need to work with scarce data and in situations where large, labeled training data sets are rare.

[1] https://www.helsinki.fi/en/researchgroups/multi-source-probabilistic-inference

Introduction

Working with variational approximations since roughly 2007

- Bayesian interpretation of canonical correlation analysis (CCA) and inter-battery factor analysis (IBFA) [Klami et al., 2013]
- Generized Matrix Factorization setups: Group factor analysis (GFA) [Klami et al., 2015], collective matrix factorization (CMF) [Klami et al., 2014], unsupervised object matching [Klami, 2013]
- VI fo non-conjugate models; exponential family CCA [Klami et al., 2010], topic models [Virtanen et al., 2012], Polya-gamma augmentations [Klami, 2014]
- Efficient reparameterization gradients [Sakaya and Klami, 2017]
- Calibrating variational approximation for decision problems [Kusmierczyk et al., 2019]

Contents

Today: Classical variational inference

- Distributional approximations as alternative for MCMC
- Preliminaries: Exponential family and conjugate priors, Gibbs sampling
- Mean-field approximation for conditionally conjugate models
- Coordinate-ascent variational inference

Tomorrow: Modern variational inference

- Working towards VI for broader model families
- Gradient-based optimization with stochastic estimates
- Score function estimator
- Reparameterization gradients
- Amortized inference and VAE

For a good reference of today's material, see Blei et al. [2017]

Main learning objective: Know enough of variational inference to be able to read any scientific paper on the topic, also on a technical level

Bayesian Machine Learning

1. Define a probabilistic model $p(\boldsymbol{x}, \boldsymbol{\theta})$ by specifying the prior $p(\boldsymbol{\theta})$ and the likelihood $p(\boldsymbol{x}|\boldsymbol{\theta})$, often in form of a hierarchical model
2. Find the posterior distribution $p(\boldsymbol{\theta}|\mathcal{D})$ of the model parameters, using the Bayes' rule

$$p(\boldsymbol{\theta}|\mathcal{D}) = \frac{p(\mathcal{D}, \boldsymbol{\theta})}{p(\mathcal{D})}$$

3. To use the model, integrate over the posterior distribution. For example to predict new samples, use $p(\boldsymbol{x}'|\mathcal{D}) = \int p(\boldsymbol{x}'|\boldsymbol{\theta})p(\boldsymbol{\theta}|\mathcal{D})d\boldsymbol{\theta}$

Note: "Being Bayesian" means explicitly accounting for the uncertainty as captured by the posterior, **not** using priors in your model. If we use Maximum Likelihood (ML) or Maximum a Posterior (MAP) estimates we still do probabilistic ML, but not Bayesian ML

Example: Linear regression

Linear regression can be expressed as a probabilistic model, for instance, as follows:

$$\begin{aligned} p(y_n|\mathbf{w}, \mathbf{x}_n, \tau_y) &= \mathcal{N}(\mathbf{w}^T\mathbf{x}_n + b, \tau_y), \\ p(\mathbf{w}|\tau_w) &= \mathcal{N}(0, \tau_w), \\ p(b|\tau_b) &= \mathcal{N}(0, \tau_b), \\ p(\tau_y|\alpha_0, \beta_0) &= \text{Gamma}(\alpha_0, \beta_0) \end{aligned}$$

- Likelihood: y are normally distributed around a linear function of covariates $\mathbf{x}$
- Prior: (1) The weights are assume to be independent and somewhat small, centered around zero. (2) The bias is also centered around zero, but often with $\tau_b < \tau_{w0}$. (3) The precision follows a gamma prior, forcing it to be positive.
- All y_n are independent conditional on $\mathbf{x}_n$ and the parameters
- We could also have prior for $\mathbf{x}_n$ if we wanted to

Example: Linear regression

The whole model is characterised by the joint probability of all variables

$$p(\{y_n\}, \mathbf{w}, \tau_y, b | \{\mathbf{x}_n\}, \tau_b, \tau_w, \alpha_0, \beta_0) = p(\mathbf{w}|\tau_w) p(b|\tau_b) p(\tau_y|\alpha_0, \beta_0) \prod_{n=1}^{N} p(y_n|\mathbf{x}_n, \mathbf{w}, \tau_y)$$

which would typically be written in log-space so that the product becomes a sum

We need to infer the posterior $p(\mathbf{w}, b, \tau_y | \{\mathbf{x}_n, y_n\}_{n=1}^{N}, \tau_b, \tau_w, \alpha_0, \beta_0)$, and eventually we care about interpretation of the parameters or predictions $p(y|\mathbf{x}) = \int p(y|\mathbf{x}, \boldsymbol{\theta}) p(\boldsymbol{\theta}|\mathcal{D}) d\boldsymbol{\theta}$ (that follow some non-normal distribution)

Bayesian inference

Unlike majority of machine learning, Bayesian inference is not an optimization problem. Instead, we just apply the Bayes' rule

For extremely simple models this can be done analytically, but we are interested in more complex models and hence need approximate techniques

MCMC is the common general-purpose solution, but today we talk about distributional approximations instead, sometimes motivated by

- Computational efficiency
- Well-behaving objective
- Deterministic solution
- Easier integration into existing pipelines

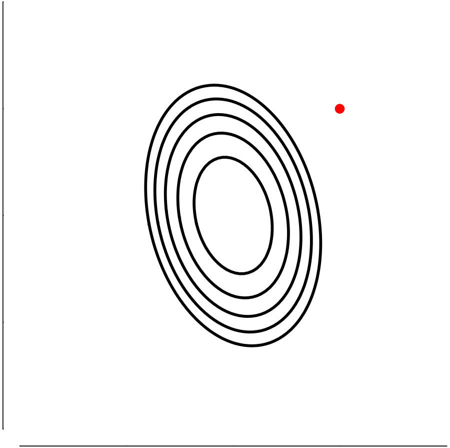
Bayesian inference using MCMC

In the end we mostly care about expectations of some function over the posterior $\mathbb{E}_{p(\boldsymbol{\theta}|\boldsymbol{x})}[f(\boldsymbol{\theta})]$, so Monte Carlo approximation can be used instead

Draw $\boldsymbol{\theta}_m$ from $p(\boldsymbol{\theta}|\boldsymbol{x})$ using some algorithm that hopefully gives good enough samples and compute

$$\mathbb{E}_{p(\boldsymbol{\theta}|\boldsymbol{x})}[f(\mathbf{x})] \approx \frac{1}{M} \sum_{m=1}^{M} f(\boldsymbol{\theta}_m)$$

Usually $\boldsymbol{\theta}_m$ are drawn using Markov Chain Monte Carlo (MCMC) algorithms, which provide the samples sequentially

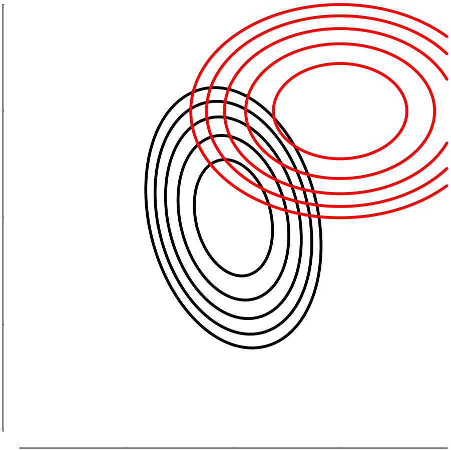


Bayesian inference using optimization

Even though MCMC algorithms provide samples from the posterior, the quality is quantified only implicitly. There is no clear learning objective, we just rely on the sampler being good enough

We can convert the search for the posterior distribution into an optimization problem as well

- Choose some parametric family of distributions $q(\boldsymbol{\theta}|\boldsymbol{\lambda})$
- Find $q(\boldsymbol{\theta}|\boldsymbol{\lambda})$ that is close to $p(\boldsymbol{\theta}|\mathcal{D})$, by minimizing some dissimilarity measure wrt $\boldsymbol{\lambda}$



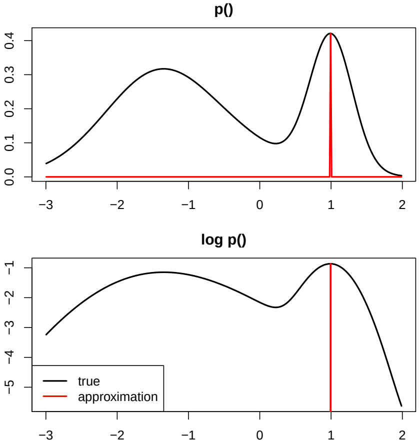
Maximum a posteriori (or maximum likelihood)

Let's stop being Bayesian for a while. Then we can find $\boldsymbol{\theta}$ for which $p(\boldsymbol{\theta}|\mathcal{D})$ is maximized, the MAP estimate, by ignoring $p(\mathcal{D})$ and optimizing

$$\max_{\boldsymbol{\theta}} \left[\log p(\mathcal{D}|\boldsymbol{\theta}) + \log p(\boldsymbol{\theta})\right]$$

which resembles standard machine learning formulation of a loss and a regulariser

Finds the mode of the posterior distribution, but completely ignores uncertainty



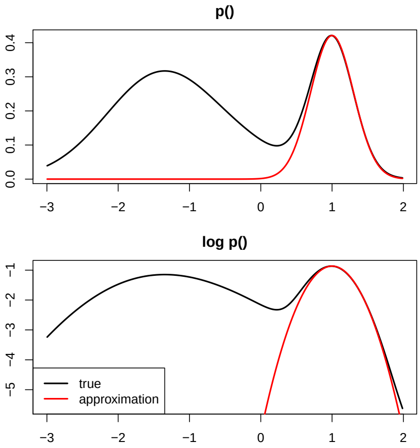
Distributional approximation: Laplace approximation

Laplace approximation is the simplest distributional approximation and it captures the uncertainty around the mode

Computationally and conceptually easy

- First find the MAP estimate (by any optimization algorithm, e.g. SGD)
- Then estimate the variation around that point, based on the curvature of the log-posterior around that point

By definition finds unimodal approximation, interpretable either as Taylor series for log-density or a Gaussian approximation



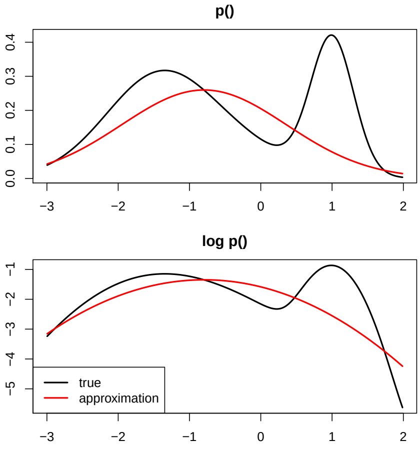
Distributional approximation: Variational approximation

Variational inference directly solves for a suitable distribution that is close to the posterior:

- Postulate a parametric form $q(\boldsymbol{\theta}|\boldsymbol{\lambda})$
- Optimize for $\boldsymbol{\lambda}$ such that $q(\boldsymbol{\theta}|\boldsymbol{\lambda}) \approx p(\boldsymbol{\theta}|\mathcal{D})$
- ...by minimizing a suitable distance measure $d(q(\boldsymbol{\theta}|\boldsymbol{\lambda}), p(\boldsymbol{\lambda}|\mathcal{D}))$ and making sure $\int q(\boldsymbol{\theta}|\boldsymbol{\lambda})d\boldsymbol{\theta} = 1$

Important:

- $\boldsymbol{\theta}$ always refers here to parameters of the model, which are unknown – we want to learn the distribution over them
- $\boldsymbol{\lambda}$ are parameters of our approximation – we optimize over these



Variational inference

Today we go through the full derivation of variational inference for a specific class of conditionally conjugate models and mean-field approximation family, but before we can do so we need to spend a bit of time talking about useful concepts

- Exponential family of distributions and conjugacy
- Gibbs sampling as a simpler algorithm for conditionally conjugate models
- Measuring the dissimilarity between distributions

Detour: Exponential family

Exponential family is a set of probability distributions that can be expressed in common form

$$p(\boldsymbol{x}|\boldsymbol{\theta}) = h(\boldsymbol{x})e^{\eta(\boldsymbol{\theta})T(\boldsymbol{x})+A(\boldsymbol{\theta})}$$

where a specific distribution is determined by the choice of the four terms:

- η is called natural parameter
- $T(\boldsymbol{x})$ is sufficient statistic of the data
- canonical form if $\eta(\boldsymbol{\theta}) = \boldsymbol{\theta}$
- $A(\boldsymbol{\theta})$ is the log-partition function (some sources use $g(\boldsymbol{\theta}) = e^{A(\boldsymbol{\theta})}$)

Looks a bit intimidating, but is highly useful in Bayesian ML:

- Almost all probability distributions you know of belong to the exponential family
- Unifies theoretical analysis: Lots of results have been proven for the exponential family
- Can also unify derivations and implementations

Exponential family: Conjugate prior

For every likelihood $p(\boldsymbol{x}|\boldsymbol{\theta})$ that belongs to exponential family, there exists a conjugate prior $p(\boldsymbol{\theta})$ such that the posterior $p(\boldsymbol{\theta}|\boldsymbol{x})$ is in the same distribution family as the prior

If we have likelihood

$$p(\boldsymbol{x}|\eta) = h(\boldsymbol{x})e^{\eta T(\boldsymbol{x}) - A(\eta)}$$

and prior

$$p(\eta|\xi,\nu) = f(\eta,\nu)e^{\eta\xi - \nu A(\eta)}$$

then their product (un-normalized posterior) is

$$p(\eta|\boldsymbol{x},\xi,\nu) \propto h(\boldsymbol{x})f(\eta,\nu)e^{\eta(T(\boldsymbol{x})+\xi)-(\nu+1)A(\eta)}$$

that has the same form as the prior because $h(\boldsymbol{x})$ does not depend on the parameter η and it can be incorporated into $f(\eta,\nu)$

Because we know the distribution we also know the normalizing constant

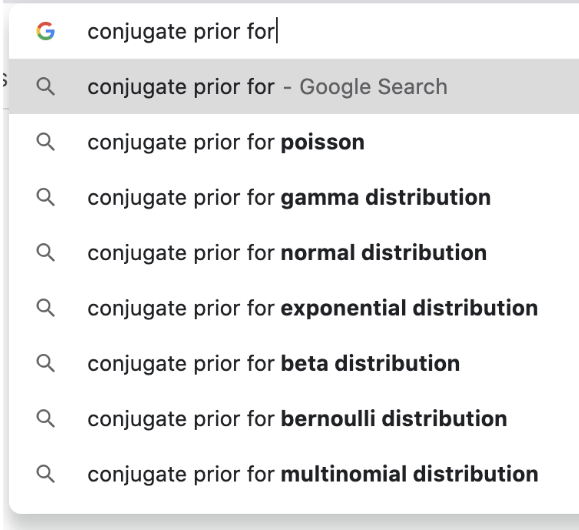
Conjugate prior

Examples:

- Conjugate prior for mean of normal distribution (conditional on known variance) is normal distribution
- Conjugate prior for precision of normal distribution is gamma distribution, and for variance we have inverse-gamma distribution
- Conjugate prior for the parameters of Bernoulli and Binomial are beta distribution
- Conjugate prior for the parameters of Poisson and Exponential distribution are gamma distribution

Conjugate prior

You can typically find these with simple Google search:



conjugate prior for

conjugate prior for - Google Search

conjugate prior for **poisson**

conjugate prior for **gamma distribution**

conjugate prior for **normal distribution**

conjugate prior for **exponential distribution**

conjugate prior for **beta distribution**

conjugate prior for **bernoulli distribution**

conjugate prior for **multinomial distribution**

Conditional conjugacy

For fully conjugate models Bayesian inference can be done analytically, but we cannot usually define a conjugate prior for the joint distribution of all model parameters

As nice compromise between tractability and flexibility is obtained by using models where each conditional distribution has a conjugate prior

If $p(\boldsymbol{x}|\boldsymbol{\theta}_d, \boldsymbol{\theta}_{-d})$ is in exponential family then $p(\boldsymbol{\theta}_d|\boldsymbol{x}, \boldsymbol{\theta}_{-d})$ has the same distribution as $p(\boldsymbol{\theta}_d|\boldsymbol{\theta}_{-d})$, so we only need to assume conjugacy conditional on other parameters

For example, our earlier linear regression model had this property:

$$\begin{aligned}
p(y_n|\mathbf{w}, \mathbf{x}_n, \tau_y) &= \mathcal{N}(\mathbf{w}^T\mathbf{x}_n + b, \tau_y), \\
p(b|\tau_b) &= \mathcal{N}(0, \tau_b), \\
p(\mathbf{w}|\tau_w) &= \mathcal{N}(0, \tau_w), \\
p(\tau_y|\alpha_0, \beta_0) &= \text{Gamma}(\alpha_0, \beta_0)
\end{aligned}$$

Exponential family: Winging it

We will not work using the exponential family notation today, but instead just rely on basic facts about conditionally conjugate models

We can then derive inference algorithms in standard parameterization as long as we

- Know which distributions belong to the exponential family
- Remember that for every one there is a conjugate prior and that the posterior has the same form
- To find out the posterior, identify all relevant terms and afterwards find out the normalising term

Exponential family: Winging it

Example:

$$p(x|\lambda) = \text{Poisson}(\lambda) \qquad \log p(x|\lambda) = x \log \lambda - \lambda - \log x!,$$

$$p(\lambda) = \text{Gamma}(\alpha, \beta) \qquad \log p(\lambda) = \alpha \log \beta - \log \Gamma(\alpha) + (\alpha - 1) \log \lambda - \beta\lambda$$

Since we know that $p(\lambda|x)$ is a gamma distribution, it is enough to find the terms in front of $\log \lambda$ and λ:

$$\log \lambda : x + (\alpha - 1) = (\alpha + x - 1) = (\alpha' - 1) \text{ for } \alpha' = \alpha + x$$

$$\lambda : -1 - \beta = \beta' \text{ for } \beta' = \beta + 1$$

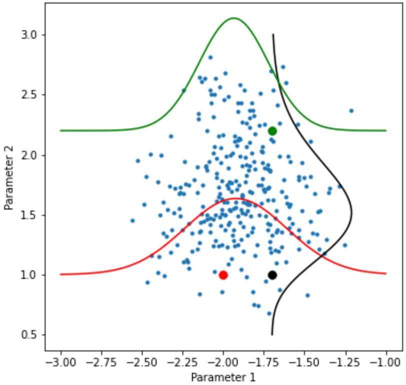
Alternative: Just consult a suitable text book

Gibbs sampling

A common algorithm for doing inference for conditionally conjugate models is Gibbs sampling

MCMC algorithm that iteratively samples every parameter $\boldsymbol{\theta}_d$ from the conditional distribution $p(\boldsymbol{\theta}_d | \boldsymbol{x}, \boldsymbol{\theta}_{-d})$ where all other parameters are assumed known and fixed to their current values

...which is easy for conditionally conjugate models, since we know these conditionals are of the same distribution as the corresponding prior



Example: Normal model

Let us consider one very simple conditionally conjugate model, a normal distribution with unknown mean and precision (inverse variance)

$$\begin{aligned} p(\boldsymbol{x}|\mu, \tau) &= \mathcal{N}(\mu, \tau^{-1}), \\ p(\mu|\mu_0, \tau_0) &= \mathcal{N}(\mu_0, \tau_0^{-1}), \\ p(\tau|\alpha_0, \beta_0) &= \text{Gamma}(\alpha_0, \beta_0) \end{aligned}$$

The full posterior $p(\mu, \tau|\mathcal{D})$ is two-dimensional with some dependency between the two parameters and we do not have a joint conjugate prior

Gibbs sampling only needs $p(\mu|\mathcal{D}, \tau)$ and $p(\tau|\mathcal{D}, \mu)$ that are easy to derive

Example: Normal model

Let's start with $p(\tau|\boldsymbol{x},\mu)$. The log prior is

$$\alpha_0 \log \beta_0 - \log \Gamma(\alpha_0) + (\alpha_0 - 1) \log \tau - \beta_0 \tau$$

and hence we only need to find $\log \tau$ and τ terms in the log likelihood (and verify that there are no other terms involving τ). The log likelihood for N observations is

$$\frac{N}{2} \log \tau - \frac{N}{2} \log(2\pi) - \frac{1}{2} \tau \sum_n (\boldsymbol{x}_n - \mu)^2$$

and hence the multipliers for the two terms are $\frac{N}{2}$ and $\frac{1}{2} \sum_n (\boldsymbol{x}_n - \mu)^2$

We immediately recognize this as a Gamma distribution with parameters $\alpha_0 + \frac{N}{2}$ and $\beta_0 + \frac{1}{2} \sum_n (\boldsymbol{x}_n - \mu)^2$, and hence sample new τ from that

Sanity check: Mean of gamma is $\frac{\alpha}{\beta}$ so increasing N increases precision and bigger difference between $\boldsymbol{x}_n$ and μ decreases it

Example: Normal model

The other term, $p(\mu|\boldsymbol{x},\tau)$ is slightly trickier. The log prior is

$$\frac{1}{2}\left(\log\tau_0 - \log(2\pi) - \tau_0(\mu-\mu_0)^2\right) = C + \frac{1}{2}\tau_0(\mu^2 - 2\mu\mu_0 + \mu_0^2) = C' + \frac{1}{2}\tau_0\mu^2 - \tau_0\mu_0\mu$$

so we need terms in front of μ^2 and μ

Dropping already now the terms not involving μ, the likelihood can be written as

$$D + \frac{1}{2}\tau N\mu^2 - \tau(\sum_n \boldsymbol{x}_n)\mu$$

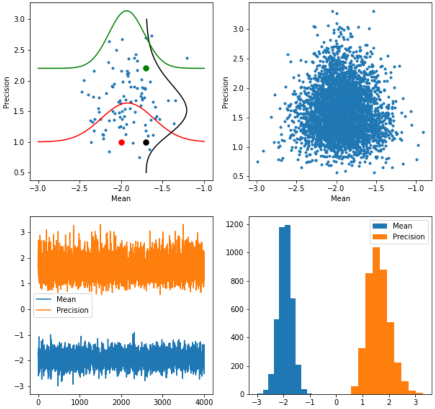
and hence we have $\frac{1}{2}(\tau_0 + N\tau)$ in front of μ^2 and $(\tau_0\mu_0 + \tau\sum_n \boldsymbol{x}_n)$ in front of μ

After some additional manipulation (completing the square) we get

$$p(\mu|\mathcal{D},\tau) = \mathcal{N}(\frac{\tau_0\mu_0 + \tau\sum_n \boldsymbol{x}_n}{\tau_0 + N\tau}, \tau_0 + N\tau)$$

Sanity check: With increasing data, the precision goes up and the mean converges to the mean of the observations

Gibbs in action



```
def Gibbs(x, mu, prec, T):
   samples = [mu, prec]
   for t in range(T):
     # Sample mean conditional on precision
     S2 = len(x)*prec + 1.0/prior_sigma**2.0
     m = (prec*np.sum(x) + 1.0/prior_sigma**2.0 * prior_mu) / S2
     mu = np.random.normal(m, np.sqrt(1/S2))

     # Sample precicision conditional on mean
     prec = np.random.gamma(prior_alpha + 0.5*len(x),
                scale = 1.0 / (prior_beta + 0.5*np.sum((x - mu)**2.0)))

     samples = np.vstack([samples, [mu, prec]])
 return samples
```

Re-cap of exponential family and Gibbs sampling

To summarize:

- For conditionally conjugate models, we can easily derive the conditional distribution of each parameter with all others fixed
- Gibbs sampling conducts inference using these, repeatedly sampling each parameter at a time
- Produces a set of samples that asymptotically follow the true posterior

Next up: Continue from here to variational approximation as a distributional approximation alternative that ends up being closely related to Gibbs sampling

Towards VI: KL divergence

The goal of VI is to find $q(\boldsymbol{\theta}|\boldsymbol{\lambda}) \approx p(\boldsymbol{\theta}|\mathcal{D})$ and to do so we need a distance measure for quantifying the similarity

The most common choice is Kullback-Leibler (KL) divergence

$$d_{KL}(q,p) = \int_{\boldsymbol{\theta}} q(\boldsymbol{\theta}) \log \frac{q(\boldsymbol{\theta})}{p(\boldsymbol{\theta})} d\boldsymbol{\theta}$$

which is a dissimilarity measure between probability distributions $q(\cdot)$ and $p(\cdot)$, with

- $d_{KL}(p,q) \geq 0$
- $d_{KL}(p,q) = 0$ if and only if $p(\boldsymbol{\theta}) = q(\boldsymbol{\theta}) \; \forall \; \boldsymbol{\theta}$
- It is not symmetric and hence not a metric: $D_{KL}(q,p) \neq D_{KL}(p,q)$

(Note: There are other divergences as well, and practical algorithms using those, but KL divergence has a lot of nice properties for VI)

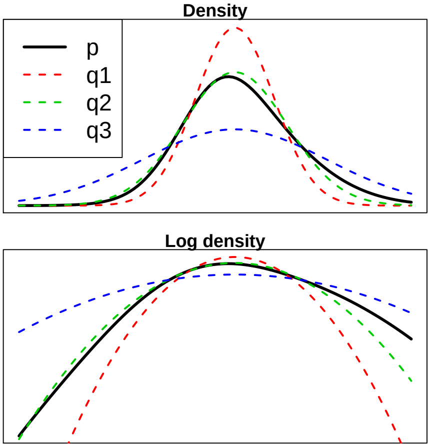
KL divergence

For $q(x) = 0$, we have $q(x) \log \frac{q(x)}{p(x)} = 0$, but for $p(x) \to 0$ when $q(x) > 0$ the value tends to ∞

Hence, the support of $q(\cdot)$ has to be within the support of $p(\cdot)$

Consequently, $q(\cdot)$ minimizing the divergence to $p(\cdot)$ will have less fat tails, and in general smaller variance

Here $d(q_2|p) < d(q_1|p) \ll d(q_3|p)$



Variational approximation

Now we can finally start talking about variational inference, as a method that finds $q(\boldsymbol{\theta}|\boldsymbol{\lambda}) \approx p(\boldsymbol{\theta}|\mathcal{D})$ by minimizing $D_{KL}(q(\boldsymbol{\theta}|\boldsymbol{\lambda})|p(\boldsymbol{\theta}|\mathcal{D}))$

The order is important: We can integrate over $q(\boldsymbol{\theta})$ (because we choose it), but not over the unknown posterior

VI fits the approximation "inside" the posterior, to avoid putting probability mass for $p(\boldsymbol{\theta}|\mathcal{D}) \approx 0$, and hence always underestimates variance

Note: We will write $q(\boldsymbol{\theta}) = q(\boldsymbol{\theta}|\boldsymbol{\lambda})$ without explicitly denoting the variational parameter to compress the notation during the derivation

Variational approximation

The definition gives the learning objective (to be minimized)

$$\mathcal{L}_{KL}(\boldsymbol{\lambda}) = \int q(\boldsymbol{\theta}) \log \frac{q(\boldsymbol{\theta})}{p(\boldsymbol{\theta}|\mathcal{D})} d\boldsymbol{\theta}$$

which we cannot compute since the posterior $p(\boldsymbol{\theta}|\mathcal{D})$ is unknown

To proceed, re-write it as

$$\mathcal{L}_{KL}(\boldsymbol{\lambda}) = \int q(\boldsymbol{\theta}) \log \frac{q(\boldsymbol{\theta})p(\mathcal{D})}{p(\boldsymbol{\theta}|\mathcal{D})p(\mathcal{D})} d\boldsymbol{\theta} = \int q(\boldsymbol{\theta}) \log \frac{q(\boldsymbol{\theta})p(\mathcal{D})}{p(\boldsymbol{\theta}, \mathcal{D})} d\boldsymbol{\theta}$$

to convert it into a function of joint density $p(\theta, \mathcal{D})$ (easy to compute) and marginal likelihood $p(\mathcal{D})$ (even harder to compute, but not a function of $\boldsymbol{\theta}$)

Variational approximation

Since $p(\mathcal{D})$ does not depend on $\boldsymbol{\theta}$ we can take it out of the integral to get

$$\begin{aligned}\mathcal{L}_{KL}(\boldsymbol{\lambda}) &= \int q(\boldsymbol{\theta}) \log \frac{q(\boldsymbol{\theta})}{p(\boldsymbol{\theta}, \mathcal{D})} d\boldsymbol{\theta} + \log p(\mathcal{D}) \\ &= -\mathbb{E}_{q(\boldsymbol{\theta})} [\log p(\boldsymbol{\theta}, \mathcal{D})] - \mathcal{H}(q(\boldsymbol{\theta})) + \log p(\mathcal{D})\end{aligned}$$

$p(\mathcal{D})$ does not depend on $\boldsymbol{\lambda}$ either, so we can ignore it in optimization and get a learning objective consisting of two computable terms

- Negative expected log-density over the approximation, pushing probability mass for parameters maximizing the likelihood
- Negative entropy of the approximation, preventing collapsing the distribution to a point distribution (Entropy quantifies the uncertainty of a distribution)

However, we cannot actually compute the objective value!

Variational approximation

Instead of minimizing the KL divergence we can alternatively re-write the objective as maximizing a lower bound for the marginal likelihood

$$\log p(\mathcal{D}) = \mathcal{L}_{KL}(\boldsymbol{\lambda}) - D_{KL}(q(\boldsymbol{\theta})|p(\boldsymbol{\theta}, \mathcal{D}))$$

where writing $p(\boldsymbol{\theta}, \mathcal{D}) = p(\mathcal{D}|\boldsymbol{\theta})p(\boldsymbol{\theta})$ leads to

$$\log p(\mathcal{D}) = \mathbb{E}_{q(\boldsymbol{\theta})}\left[\log p(\mathcal{D}|\boldsymbol{\theta})\right] - D_{KL}(q(\boldsymbol{\theta})|p(\boldsymbol{\theta})) + \mathcal{L}_{KL}(\boldsymbol{\lambda})$$

Since KL-divergence is always positive, we get an alternative objective (to be maximized)

$$\mathcal{L}_{ELBO}(\boldsymbol{\lambda}) := \mathbb{E}_{q(\boldsymbol{\theta})}\left[\log p(\mathcal{D}|\boldsymbol{\theta})\right] - D_{KL}(q(\boldsymbol{\theta})|p(\boldsymbol{\theta}))$$

that combines (a) expected log-likelihood and (b) divergence between the approximation and the prior

Variational approximation

The evidence lower bound

$$\mathcal{L}_{ELBO}(\boldsymbol{\lambda}) := \mathbb{E}_{q(\boldsymbol{\theta})}\left[\log p(\mathcal{D}|\boldsymbol{\theta})\right] - D_{KL}(q(\boldsymbol{\theta})|p(\boldsymbol{\theta})) \qquad \leq \log p(\mathcal{D})$$

only contains expectations of known parts of the model over the approximation. It is a well-defined learning objective, but we still need to be able to both evaluate and optimize for it

Note that comparing $\mathcal{L}_{ELBO}$ for different models does not really make sense – it differs from $\log p(\mathcal{D})$ by a term whose magnitude can vary a lot

Evaluation: To compute the expectations, we need to assume $q(\boldsymbol{\theta})$ is sufficiently simple

Optimization:

- Today: Coordinate ascent and closed-form analytic updates
- Tomorrow: Stochastic gradient descent and Monte Carlo approximations

Evaluating the bound

Already evaluating

$$\mathcal{L}_{ELBO}(\boldsymbol{\lambda}) := \mathbb{E}_{q(\boldsymbol{\theta})}\left[\log p(\mathcal{D}|\boldsymbol{\theta})\right] - D_{KL}(q(\boldsymbol{\theta})|p(\boldsymbol{\theta}))$$

requires computing an integral over the approximating family, and it is not necessarily easy

If $\boldsymbol{\theta} \in \mathbb{R}^D$, we have a D-dimensional integral, and we need to compute the expectation of $\log p(\mathcal{D}|\boldsymbol{\theta})$ – it is a scalar function, but depends on all D parameters and all N data samples and may be slow to compute

Mean-field approximation

The most common approach for ensuring we can compute various expectations over $q(\boldsymbol{\theta})$ is to assume it factorizes into a product of lower-dimensional terms

This is called the mean-field approximation, which in the fully factorized case is

$$q(\boldsymbol{\theta}|\boldsymbol{\lambda}) = \prod_{d=1}^{D} q_d(\theta_d|\boldsymbol{\lambda}_d)$$

where each $\boldsymbol{\lambda}_d$ can be multi-dimensional (e.g. μ and σ^2 for univariate normal distribution)

Now $\mathbb{E}_{q(\boldsymbol{\theta})}[\log p(\mathcal{D}|\boldsymbol{\theta})]$ becomes a nested collection of D one-dimensional integrals

$$\mathbb{E}_{q(\boldsymbol{\theta})}[\log p(\mathcal{D}|\boldsymbol{\theta})] = \int \dots \int \prod_{d=1}^{D} q_d(\theta_d|\boldsymbol{\lambda}_d) \log p(\mathcal{D}|\boldsymbol{\theta}) d\theta_1 \dots d\theta_D,$$

which is typically considerably easier to compute

Mean-field approximation

For the mean-field approximation the objective can be re-written as

$$\mathcal{L}_{ELBO}(\boldsymbol{\lambda}) = \int \dots \int q(\boldsymbol{\theta}) \log p(\mathcal{D}|\boldsymbol{\theta}) d\theta_1 \dots d\theta_D - \sum_{d=1}^{D} D_{KL}(q(\theta_d|\lambda_d)|p(\theta_d))$$

where the KL divergences separate because they are independent, but the expected log-likelihood still involves the whole joint distribution

To optimize this we can use coordinate ascent, which allows maximizing the objective wrt to any of $q(\theta_d|\lambda_d)$ at a time (quite much like Gibbs sampling proceeds)

However, since we did not define $q(\theta_d|\lambda_d)$ there is no obvious parameterization. Instead, we need to perform variational calculus to optimize over distributions

Detour: Coordinate ascent

Coordinate ascent is one of the simplest iterative optimization algorithms

To maximize some objective $\mathcal{L}(\boldsymbol{\theta})$ for $\boldsymbol{\theta} \in \mathbb{R}^D$, we iteratively

1. Select one dimension $d \in [1, \ldots, D]$
2. Maximize $\mathcal{L}(\boldsymbol{\theta})$ by modifying only θ_d, keeping all other dimensions of $\boldsymbol{\theta}$ fixed

Solving the one-dimensional optimization problems is often quite easy; closed-form updates, line search, ...

Naturally only converges to a local optimum for non-convex objectives

Detour: Langrange multipliers

Optimization problems of type

$$\max_{\boldsymbol{\lambda}} \; f(\boldsymbol{\lambda})$$
$$\text{s.t. } g(\boldsymbol{\lambda}) = 0$$

can be solved by forming the lagrangian

$$\mathcal{L}_L(\boldsymbol{\lambda}) = f(\boldsymbol{\lambda}) - \alpha g(\boldsymbol{\lambda}),$$

where $\alpha \geq 0$ is a langrange multiplier, and solving for $\nabla \mathcal{L}_L(\boldsymbol{\lambda}) = 0$

"Proof": We can improve the loss along the gradient projected on the constraint, unless the projection is zero. This happens when $\nabla f(\boldsymbol{\lambda}) = \alpha \nabla g(\boldsymbol{\lambda})$ for some α

Mean-field approximation

So, let's optimize for $q(\theta_d|\lambda_d)$ such that $\int q(\theta_d|\lambda_d) = 1$ (it is a distribution)

The lagrangian of the optimization problem is then

$$\int \dots \int q(\boldsymbol{\theta}) \log p(\mathcal{D}, \boldsymbol{\theta}) d\theta_1 \dots d\theta_D - \int q(\theta_d) \log q(\theta_d) d\theta_d - \alpha \left(\int q(\theta_d) d\theta_d - 1 \right)$$

Start by re-writing the first term as

$$\int_{q_d} q(\theta_d) \mathbb{E}_{q_{-d}(\boldsymbol{\theta}_{-d}|\boldsymbol{\lambda}_{-d})} \left[\log p(\mathcal{D}, \boldsymbol{\theta}) \right] d\theta_d$$

where the inner expectation is computed over all other terms in the approximation

Note that we only have one entropy here – the others do not depend on this $q(\theta_d)$

Mean-field approximation

Now we have the objective

$$\int q(\theta_d)\left[\mathbb{E}_{q_{-d}}[\log p(\mathcal{D},\boldsymbol{\theta})] - \log q(\theta_d) - \alpha\right] d\theta_d + \alpha$$

which can be solved by differentiating w.r.t. to $q(\theta_d)$ (yes, we can do that) and solving for $\nabla q(\theta_d) = 0$

The derivative is

$$\mathbb{E}_{q_{-d}}[\log p(\mathcal{D},\boldsymbol{\theta})] - \log q(\theta_d) - \alpha + \frac{q(\theta_d)}{q(\theta_d)}$$

and solving for $\nabla q(\theta_d) = 0$ gives

$$\log q(\theta_d) = \mathbb{E}_{q_{-d}}[\log p(\mathcal{D},\boldsymbol{\theta})] + C$$

and hence

$$q(\theta_d) \propto e^{\mathbb{E}_{q_{-d}}[\log p(\mathcal{D},\boldsymbol{\theta})]}$$

Mean-field approximation

Instead of

$$q(\theta_d) \propto e^{\mathbb{E}_{q_{-d}}[\log p(\mathcal{D},\boldsymbol{\theta})]}$$

we can just as well use

$$q(\theta_d) \propto e^{\mathbb{E}_{q_{-d}}[\log p(\theta_d|\boldsymbol{\theta}_{-d},\mathcal{D})]}$$

since

$$p(\mathcal{D},\boldsymbol{\theta}) = p(\boldsymbol{\theta}_d|\mathcal{D},\boldsymbol{\theta}_{-d})p(\mathcal{D},\boldsymbol{\theta}_{-d}),$$

where the second term is constant wrt to θ_d and can be moved into the normalizing constant

These are the same conditional distributions we used for Gibbs sampling! Instead of drawing a sample, we now just compute the expectation of the distribution over all other variables

If deriving updates from scratch you can use either $p(\mathcal{D},\boldsymbol{\theta})$ or $p(\boldsymbol{\theta}_d|\boldsymbol{\theta}_{-d},\mathcal{D})$, but if you already derived the latter than naturally use them

Gibbs vs Mean-field VI

For conditionally conjugate models, we can derive a Gibbs sampler and mean-field VI in analogous fashion

For both we need the conditional distributions $p(\theta_d|\boldsymbol{\theta}_{-d}, \mathcal{D})$ and we iteratively loop through every parameter at a time, but the specific steps are different

- Gibbs: Keep all other parameters $\boldsymbol{\theta}_{-d}$ fixed and select a new random sample from the conditional distribution for θ_d
- VI: Update parameters of $q(\theta_d|\lambda_d)$ using a deterministic update rule. For all $\boldsymbol{\theta}_{-d}$ that appear in the conditional distribution we integrate over the current $q(\boldsymbol{\theta}_{-d})$

Mean-field approximation: Example

For our example model

$$
\begin{aligned}
p(x|\mu,\tau) &= \mathcal{N}(\mu,\tau), \\
p(\mu|\mu_0,\tau_0) &= \mathcal{N}(\mu_0,\tau_0), \\
p(\tau|\alpha_0,\beta_0) &= \text{Gamma}(\alpha_0,\beta_0)
\end{aligned}
$$

the mean-field approximation with $\lambda = \{m, t, \alpha, \beta\}$ is

$$q(\boldsymbol{\theta}|\boldsymbol{\lambda}) = q_\mu(\mu|m,t)q_\tau(\tau|\alpha,\beta)$$

where the first term is a normal distribution with precision t, and the second term is a gamma distribution

These are not assumptions: The specific distributions follow from the model and the mean-field assumption

Mean-field approximation: Example

To fit the approximation we need

1. Update rule for q_μ
2. Update rule for q_τ
3. Computation of $\mathcal{L}_{ELBO}$

Mean-field approximation: Example

Start with $q_\tau(\alpha, \beta)$. From Gibbs updates we remember that the conditional distribution is

$$\text{Gamma}(\alpha_0 + \frac{N}{2}, \beta_0 + \frac{1}{2}\sum_n (x_n - \mu)^2)$$

with log-density (using $\alpha' = \alpha_0 + \frac{N}{2}$)

$$C + \alpha' \log \beta + (\alpha' - 1) \log \tau - \left(\beta_0 + \frac{1}{2}\sum_n (x_n - 2x_n\mu + \mu^2)\right)\tau$$

Now we need integral of this over $q_\mu(m, t)$. The only terms that depend on μ are μ and μ^2, which have simple expressions

$$\mathbb{E}[\mu] = m, \qquad \mathbb{E}[\mu^2] = m^2 + t^{-1},$$

All terms that are independent of μ can be taken outside of the integral, so in the end we have

$$\alpha = \alpha_0 + \frac{N}{2}, \qquad \beta = \beta_0 + \frac{1}{2}\sum_n (x_n - 2x_n m + m^2 + t^{-1})$$

Mean-field approximation: Example

For $q_\mu(m, t)$ the required conditional distribution is

$$p(\mu|x, \tau) = \mathcal{N}(\frac{\tau_0 + \tau \sum_n x_n}{\tau'}, \tau')$$

with $\tau' = \tau_0 + N\tau$.

The log-density involves both τ (in front of the quadratic term) and $\log \tau$, but the latter only appears in the normalization constant of $q_\mu(m, t)$ so we do not even need it

In practice, we can push everything else except τ outside the integral over $q_\tau(\alpha, \beta)$ and hence only need the mean $\mathbb{E}[\tau] = \frac{\alpha}{\beta}$ to get the updates

$$t = \tau_0 + N\frac{\alpha}{\beta}, \qquad m = \frac{\tau_0 + \frac{\alpha}{\beta} \sum_n x_n}{t}$$

Mean-field approximation: Example

Finally, in order to monitor progress and to verify the updates we need to compute $\mathcal{L}_{ELBO}$, which includes the expected joint likelihood

$$\begin{aligned}\mathbb{E}_{q_\mu q_\tau}\left[\log p(x,\mu,\tau)\right] = &-\frac{1}{2}\log 2\pi + \frac{1}{2}\mathbb{E}_{q_\tau}\left[\log \tau\right] - \frac{1}{2}\mathbb{E}_{q_\tau}\left[\tau\right]\mathbb{E}_{q_w}\left[(x-\mu)^2\right] \\ &- \frac{1}{2}\log 2\pi + \frac{1}{2}\log \tau_0 - \frac{1}{2}\tau_0\mathbb{E}_{q_w}\left[(\mu-\mu_0)^2\right] \\ &- \alpha_0\log\beta_0 + \log\Gamma(\alpha_0) - (\alpha_0-1)\mathbb{E}_{q_\tau}\left[\log\tau_y\right] + \beta\mathbb{E}_{q(\tau)}\left[\tau_y\right]\end{aligned}$$

and additionally the entropies $\mathcal{H}(q_\tau)$ and $\mathcal{H}(q_\mu)$

The likelihood only involves terms we have already seen, except for

$$\mathbb{E}_{q_\tau}\left[\log\tau_y\right] = \psi(\alpha) - \log(\beta),$$

where $\psi(\cdot)$ is the digamma function. The entropies, in turn, are available in literature

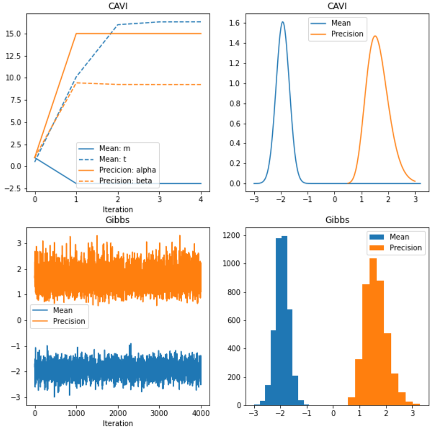
Mean-field approximation: Example

In the end we have

- Closed-form updates that only depend on expectations of the other approximation factors
- Closed-form ELBO that only depends on expectations
- Coordinate ascent algorithm that converges to local optimum of ELBO by non-decreasing updates

The derivation, however, was quite involved, requiring some non-trivial algebraic manipulation and knowledge on non-trivial expectations (log of a gamma variable, second-order terms for normal distribution)

CAVI in action



Main differences between CAVI and Gibbs:

- + CAVI converges fast and we do not need to keep on sampling
- + CAVI returns explicit representation for the result, rather than a set of samples
- +/- Roughly the same computational cost per iteration
  - We had to make the mean field assumption, which may be bad
  - CAVI slightly more difficult to derive, but not much

Using variational approximations

In the end we want to compute expectations $\mathbb{E}_{p(\boldsymbol{\theta}|\boldsymbol{x})}[f(\boldsymbol{\theta})]$. The naive estimate simply plugs in the approximation in place of the posterior

$$\mathbb{E}_{p(\boldsymbol{\theta}|\boldsymbol{x})}[f(\boldsymbol{\theta})] \approx \mathbb{E}_{q(\boldsymbol{\theta})}[f(\boldsymbol{\theta})] \approx \frac{1}{M}\sum_{m=1}^{M} f(\boldsymbol{\theta}_m),$$

but this is possibly severely biased. Nevertheless, this is what people often do

To reduce the bias we can use importance sampling estimator

$$\mathbb{E}_{p(\boldsymbol{\theta}|\boldsymbol{x})}[f(\boldsymbol{\theta})] \approx \frac{\sum_{m=1}^{M} w_m f(\boldsymbol{\theta}_m)}{\sum_{m=1}^{M} w_m}, \text{ for } w_m = \frac{p(\boldsymbol{x}, \boldsymbol{\theta}_m)}{q(\boldsymbol{\theta}_m|\boldsymbol{\theta})},$$

or even better pareto smoothed importance sampling [Yao et al., 2018] that can also be used for evaluating the approximation accuracy

Variational inference, classical

Re-cap

- Distributional approximations explicitly approximate the posterior distribution
- VI does it by minimizing the KL divergence between the approximation and the posterior
- Alternative: Maximize a lower bound for the marginal likelihood
- Mean-field approximation leads to elegant closed-form update rule for coordinate ascent
- The updates use the same conditional distributions that Gibbs needs, but now we integrate over the other parameters instead of conditioning on them

Variational inference, classical

Core limitations

- Underestimates uncertainty
- ELBO is useful for monitoring convergence, but not directly for comparing models
- Moving outside of mean-field is difficult
- Moving outside of conditionally conjugate models is difficult (but not impossible)
- The derivations are slow, error-prone and perhaps a bit waste of time

Tomorrow: How to get rid of some of these limitations with modern tools

References I

David M. Blei, Alp Kucukelbir, and Jon D. McAuliffe. Variational Inference: A Review for Statisticians. *Journal of the American Statistical Association*, 112(518), 2017.

Arto Klami. Bayesian object matching. *Machine Learning*, 92(2):225–250, 2013.

Arto Klami. Polya-gamma augmentations for factor models. In *Proceedings of the Asian Conference on Machine Learning*, 2014.

Arto Klami, Seppo Virtanen, and Samuel Kaski. Bayesian exponential family projections for coupled data sources. In *Proceedings of the 26th Conference on Uncertainty in Artificial Intelligence*, 2010.

Arto Klami, Seppo Virtanen, and Samuel Kaski. Bayesian canonical correlation analysis. *Journal of Machine Learning Research*, 14:965–1003, 2013.

Arto Klami, Guillaume Bouchard, and Abhishek Tripathi. Group-sparse embeddings in collective matrix factorization. In *Proceedings of the International Conference on Learning Representations*, 2014.

Arto Klami, Seppo Virtanen, Eemeli Leppäaho, and Samuel Kaski. Group factor analysis. *IEEE Transactions on Neural Networks and Learning Systems*, 26(9):2136–2147, 2015.

References II

Tomasz Kusmierczyk, Joseph Sakaya, and Arto Klami. Variational Bayesian decision-making for continuous utitilies. In *Advances in Neural Information Processing Systems*, 2019.

Joseph Sakaya and Arto Klami. Importance sampled stochastic optimization for variational inference. In *Proceedings of Uncertainty in Artificial Intelligence*, 2017.

Seppo Virtanen, Yangqing Jia, Arto Klami, and Trevor Darrell. Factorized multi-modal topic model. In *Proceedings of the 28th Conference on Uncertainty in Artificial Intelligence*, 2012.

Yuling Yao, Aki Vehtari, Daniel Simpson, and Andrew Gelman. Yes, but did it work?: Evaluating variational inference. In *Proceedings of the 35th International Conference on Machine Learning*, 2018.