Variational Inference

Thibault Randrianarisoa

University of Toronto, Winter 2026

February 9, 2026

Overview

- Variational Inference
- M-projection
- I-projection
- Naive mean-field approach

Posterior Inference for Latent Variable Models

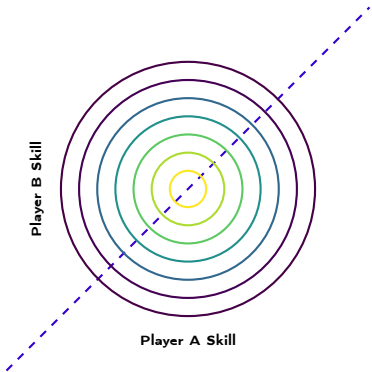
Example of latent variable models: the TrueSkill model (cf. Assignment 2)

These models have a factorization $p(x, z) = p(z)p(x|z)$ where:

- x are the observations or data,
- z are the unobserved (latent) variables.
- $p(z)$ is usually called the **prior**
- $p(x|z)$ is usually called the **likelihood**
- The conditional distribution of the unobserved variables given the observed variables (aka the **posterior**) is

$$p(z|x) = \frac{p(x|z)p(z)}{p(x)} = \frac{p(x|z)p(z)}{\int p(x, z)dz}$$

Prior in TrueSkill model



Says we are very uncertain about both players' skill.

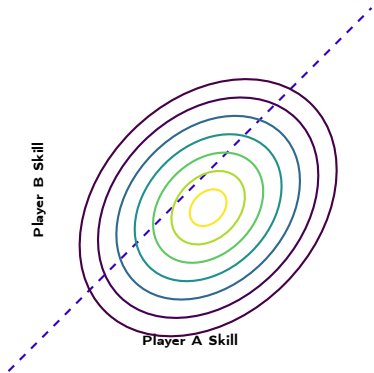
Likelihood in TrueSkill model

A beats B 5 times:

This is the part of the model that gives meaning to the latent variables.

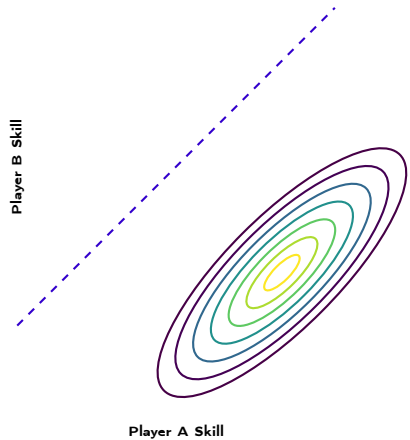
Posterior in TrueSkill model

Posterior after A beats B 5 times:



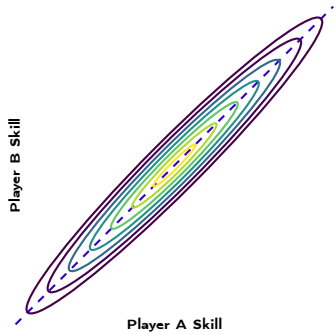
The posterior is not Gaussian anymore.

Posterior after A beats B 10 times:



Now the posterior is certain that A is better than B.

Posterior after both beat each other 10 times:



Now the posterior is certain that neither player is much better than the other, but is uncertain how good they both are in an absolute sense. (Why?)

What is hard to compute about the posterior?

The integral $p(x) = \int p(x, z)dz$ is intractable when z is large.

As a result, these operations become expensive:

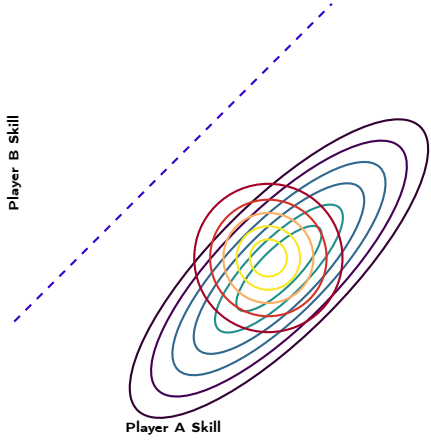
- Computing a posterior probability: $p(z|x) = \frac{p(z)p(x|z)}{p(x)}$
- Computing the evidence / marginal likelihood $p(x) = \int p(z, x)dz$
  - Useful for choosing between models, or fitting model parameters.
- Computing marginals of $p(z_1|x) = \int p(z_1, z_2, \ldots z_D|x)dz_2, dz_3, \ldots dz_D$
  - E.g. finding the posterior over a single tennis player's skill given all games.
- Sampling $z \sim p(z|x)$
  - e.g. summarizing which hypotheses are likely given the data, making predictions, and decisions.

Variational methods

Variational inference is closely related to the calculus of variations, developed in the 1700s by Euler, Lagrange. It is an approximate inference method where we seek a tractable (e.g., factorized) approximation to the target intractable distribution.

To be more formal, variational inference works as follows:

1. Choose a tractable distribution $q(z) \in \mathcal{Q}$ from a feasible set $\mathcal{Q}$. This distribution will be used to approximate $p(z|x)$.
   - For example, $q(z) = \mathcal{N}(z|\mu, \Sigma)$. Try to choose $\mathcal{Q}$ that contains a $q(z)$ that is a good approximation of the true posterior $p(z|x)$.
2. Encode some notion of "difference" between $p(z|x)$ and q that can be efficiently estimated. Usually we will use the KL divergence.
3. Minimize this difference. Usually we will use an iterative optimization method.



- Whatever feasible set $\mathcal{Q}$ we choose, it's usually not the case that there is any $q \in \mathcal{Q}$ that exactly matches the true posterior.
- But computing the true posterior is intractable, so we have to take a shortcut somewhere.

How to measure closeness: KL divergence

We will measure the difference between q and p using the **Kullback-Leibler divergence**

$$\mathrm{KL}(q(z)||p(z|x)) = \int q(z) \log \frac{q(z)}{p(z|x)} dz = \mathbf{E}_{z\sim q} \log \frac{q(z)}{p(z|x)}$$

Properties of the KL Divergence (see the tutorial):

- $\mathrm{KL}(q||p) \geqslant 0$
- $\mathrm{KL}(q||p) = 0 \Leftrightarrow q = p$
- $\mathrm{KL}(q||p) \neq \mathrm{KL}(p||q)$
- KL divergence is not a metric, since it is not symmetric.

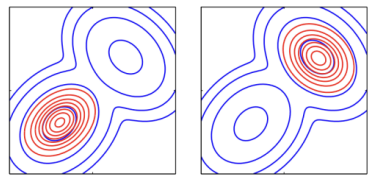
Which direction of KL to use? $\mathrm{KL}(q||p)$ vs $\mathrm{KL}(p||q)$: we will go with the tractable one.

Information (I-)Projection

I-projection: $q^* = \arg\min_{q \in \mathcal{Q}} \mathrm{KL}(q||p) = \mathbf{E}_{x \sim q(x)} \log \frac{q(x)}{p(x)}$.

- $p \approx q \implies \mathrm{KL}(q||p)$ small
- I-projection underestimates support, and does not yield the correct moments.
- $\mathrm{KL}(q||p)$ penalizes q having mass where p has none (but not vice versa).

Below: $p(x)$ is mixture of two 2D Gaussians and $\mathcal{Q}$ is the set of all 2D Gaussian distributions (with arbitrary covariance matrices)



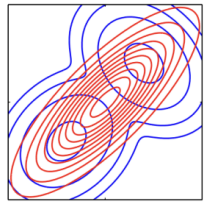
p=Blue, q^*=Red (two equivalently good solutions!)

Moment (M-)projection

M-projection: $q^* = \arg\min_{q \in \mathcal{Q}} \mathrm{KL}(p||q) = \mathbf{E}_{x \sim p(x)} \log \frac{p(x)}{q(x)}$.

- $p \approx q \implies \mathrm{KL}(p||q)$ small
- $\mathrm{KL}(p||q)$ penalizes q missing mass where p has some.
- M-projection yields a distribution $q(x)$ with the correct mean and covariance if $\mathcal{Q}$ is the family of Gaussians.

Below: $p(x)$ is mixture of two 2D Gaussians and $\mathcal{Q}$ is the set of all 2D Gaussian distributions (with arbitrary covariance matrices)



p=Blue, q^*=Red

Maximum (relative) entropy interpretation

Consider the constrained optimization problem

$$\begin{aligned} &\text{maximize } -\mathrm{KL}(q||p) \\ &\text{subject to } \mathbf{E}_{x\sim q(x)}[f_i(x)] = t_i \text{ for } i = 1, \ldots, k. \end{aligned}$$

Theorem: Maximum entropy principle

Exponential family of distributions with sufficient statistics $\mathbf{f}(x) = (f_1(x), \ldots, f_k(x))$ and base measure p maximize the relative entropy $-\mathrm{KL}(p||q)$ over all distributions satisfying: $\mathbf{E}_{x\sim q(x)}[f_i(x)] = t_i$ for $i = 1, \ldots, k$.

The solution takes the form

$$q^*(x) \propto p(x) e^{\sum_{i=1}^{k} \lambda_i f_i(x)}.$$

Drawback of M-projection

- In M-projection (without the constraints), if $\mathcal{Q}$ is the set of exponential distributions with sufficient statistics $\mathbf{f}(x)$ (and any base measure $h(x)$), then the expected sufficient statistics wrt $q^*(x)$ is the same as that wrt $p(x)$
- ⚠M-projection require expectation wrt p to match the expected sufficient statistics, hence intractable.
- Most variational inference algorithms make use of the I-projection instead and the family $\mathcal{Q}$ is parameterized

$$\mathcal{Q} = \{q_\psi : \ \psi \in \Psi\}$$

ELBO and its properties

ELBO: Evidence Lower Bound

- Evaluating $\mathrm{KL}(q_\phi(z)||p(z|x))$ directly is intractable because of the integral over z and the term $p(z|x)$, which is intractable to normalize.
- We can still "optimize" this KL without knowing the normalization constant $p(x)$.
- We solve a surrogate optimization problem: maximize the **evidence lower bound (ELBO)**, to be introduced in a second.
- Indeed, maximizing the ELBO is equivalent to minimizing

$$\mathrm{KL}(q_\phi(z)||p(z|x)).$$

ELBO: Evidence Lower Bound

Maximizing the ELBO is the same as minimizing $\mathrm{KL}(q_\phi(z)||p(z|x))$.

$$\begin{aligned}
\mathrm{KL}(q_\phi(z)||p(z|x)) &= \mathbf{E}_{z\sim q_\phi} \log \frac{q_\phi(z)}{p(z|x)} \\
&= \mathbf{E}_{z\sim q_\phi}\left[\log\left(q_\phi(z)\cdot\frac{p(x)}{p(z,x)}\right)\right] \\
&= \mathbf{E}_{z\sim q_\phi}\left[\log\frac{q_\phi(z)}{p(z,x)}\right] + \mathbf{E}_{z\sim q_\phi}\log p(x) \\
&:= -\mathcal{L}(\phi) + \log p(x)
\end{aligned}$$

Where $\mathcal{L}(\phi)$ is the **ELBO**:

$$\mathcal{L}(\phi) = \mathbf{E}_{z\sim q_\phi}\Big[\log p(z,x) - \log q_\phi(z)\Big]$$

ELBO: Evidence Lower Bound

Recall: $\mathrm{KL}(q_\phi(z)||p(z|x)) = -\mathcal{L}(\phi) + \log p(x)$.

- Rearranging, we get
$$\mathcal{L}(\phi) + \mathrm{KL}(q_\phi(z)||p(z|x)) = \log p(x)$$
- Because $\mathrm{KL}(q_\phi(z)||p(z|x)) \geqslant 0$,
$$\mathcal{L}(\phi) \leqslant \log p(x)$$
- Maximizing the ELBO $\Rightarrow$ minimizing $\mathrm{KL}(q_\phi(z)||p(z|x))$.
- Note: $\mathcal{L}(\phi) = \mathbf{E}_{z\sim q_\phi}[\log p(z,x)] + \mathbf{E}_{z\sim q_\phi}[-\log q_\phi(z)]$, so
$$\text{ELBO} = \text{expected log-joint} + \text{entropy}$$
- Sometimes we write $\mathcal{L}(\phi|x)$ or $\mathcal{L}(\theta,\phi|x)$ if $p(z,x)$ depends on a parameter θ.

Estimating gradients of the ELBO

Maximizing ELBO

Recall: $\nabla \mathcal{L}(\phi)$ gives the direction of the steepest ascent of $\mathcal{L}(\phi)$.

Gradient descent (GD) methods: $\phi_{t+1} = \phi_t + s_t \nabla \mathcal{L}(\phi_t)$.

- We have that $\mathcal{L}(\phi) = \mathbf{E}_{z \sim q_\phi}\Big[\log p(x, z) - \log q_\phi(z)\Big]$.
- We need $\nabla_\phi \mathcal{L}(\phi)$ or an unbiased estimate (from stochastic GD for instance).

How to approximate the gradient of some $\mathbf{E}(f(Y, \phi))$?

- If the distribution of Y independent of ϕ then

$$\nabla_\phi \mathbf{E}(f(Y, \phi)) = \mathbf{E}(\nabla_\phi f(Y, \phi)).$$

- We then have $\nabla_\phi \mathbf{E}(f(Y, \phi)) \approx \frac{1}{m} \sum_{i=1}^{m} \nabla_\phi f(y_i, \phi)$.
- ⚠ Problem: In our case the distribution of z depends on ϕ.

The reparameterization trick

Problem:

$$\nabla_\phi \mathbf{E}_{z\sim q_\phi}\Big[\log p(x,z) - \log q_\phi(z)\Big] \neq \mathbf{E}_{z\sim q_\phi}\Big[\nabla_\phi(\log p(x,z) - \log q_\phi(z))\Big].$$

In some situations there is a trick:

Suppose that $z \sim q_\phi$ has the same distribution as $T(\epsilon, \phi)$, where ϵ is a random variable whose distribution p_0 does not depend on ϕ. In this case, to sample $z \sim q_\phi$ by:

- sampling a random variable $\epsilon \sim p_0$,
- deterministically computing $z = T(\epsilon, \phi)$.

For example, if $z \sim \mathcal{N}(\mu, \sigma^2)$ then $z = \mu + \sigma\epsilon$, where $\epsilon \sim \mathcal{N}(0, 1)$.

- sample $\epsilon \sim \mathcal{N}(0, 1)$.
- $\phi = (\mu, \sigma^2)$. $T(\epsilon, \phi) = \mu + \sigma\epsilon$.

The reparameterization trick

If $z = T(\epsilon, \phi)$, we can write

$$\mathbf{E}_{z \sim q_\phi}\Big[\log p(x,z) - \log q_\phi(z)\Big] = \mathbf{E}_{\epsilon \sim p_0}\Big[\log p(x, T(\epsilon,\phi)) - \log q_\phi(T(\epsilon,\phi))\Big]$$

This makes the density independent of the parameter ϕ, which lets us use simple Monte Carlo: $z = T(\phi, \epsilon)$

$$\begin{aligned}
\nabla_\phi \mathcal{L}(\phi) &= \nabla_\phi \mathbf{E}_{z \sim q_\phi(z)}\big[\log p(x,z) - \log q_\phi(z)\big] \\
&= \nabla_\phi \mathbf{E}_{\epsilon \sim p_0(\epsilon)}\big[\log p(x, T(\phi,\epsilon)) - \log q_\phi(T(\phi,\epsilon))\big] \\
&= \mathbf{E}_{\epsilon \sim p_0(\epsilon)} \nabla_\phi\big[\log p(x, T(\phi,\epsilon)) - \log q_\phi(T(\phi,\epsilon))\big].
\end{aligned}$$

Instead of computing the full gradient (which is in general not possible), we compute a simple Monte Carlo estimate of it.

So generating a sample $\epsilon_1, \ldots, \epsilon_m$ from p_0, we get

$$\nabla_\phi \mathcal{L}(\phi) \approx \frac{1}{m} \sum_{i=1}^{m} \nabla_\phi \big[\log p(x, T(\phi, \epsilon_i)) - \log q_\phi(T(\phi, \epsilon_i)) \big].$$

Example: Bayesian Neural Networks

The distribution $p(z|x)$ may be very complicated:

- z are weights of neural network
- x are all observed outputs: $y_1, y_2, \ldots$. Assume inputs $\mathbf{x}_i$ are fixed.
- $p(z)$ prior on weights, usually standard normal
- $p(x|z) = \prod_i p(y_i|\mathbf{x}_i, z)$
  - for regression: $p(y|\mathbf{x}_i, z) = \mathcal{N}(\text{nnet}(\mathbf{x}_i, z), \sigma^2)$
  - for classification: $p(y|\mathbf{x}_i, z) = \text{Categorical}(y_i|\text{softmax}(\text{nnet}(\mathbf{x}_i, z)))$
- $p(z|x)$ is a collection of plausible sets of parameters that all fit the data.

Note: The number of inputs/outputs may be too large for our gradient computations.

Mean-Field VI

The Mean-Field Variational Family

To complete the specification of the optimization problem, we must define the variational family $\mathcal{Q}$.

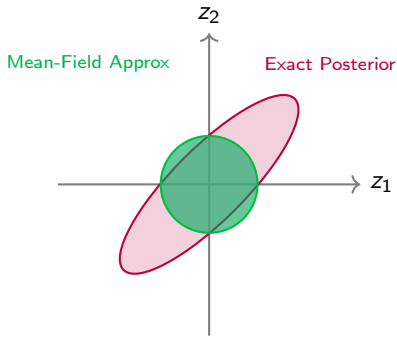
- We focus on the **mean-field variational family**, where the latent variables are mutually independent.
- A generic member of this family is:

$$q(\mathbf{z}) = \prod_{j=1}^{m} q_j(z_j)$$

- Each latent variable z_j is governed by its own distinct factor $q_j(z_j)$.
- ⚠ We do not parameterized anymore, each q_i can take any form.

Visualizing the Mean-Field Approximation

The mean-field family is expressive enough to capture any marginal density, but cannot capture correlation between variables.



- **Purple:** Exact posterior (highly positively correlated, elongated).
- **Green:** Mean-field approximation $q(\mathbf{z}) = q(z_1)q(z_2)$.
- Because q must factorize, its contours must be axis-aligned.
- **Result:** It underestimates the marginal variance to avoid placing mass in low-probability regions (due to the KL divergence direction).

Coordinate Ascent Variational Inference (CAVI)

How do we find the optimal factors $q_j(z_j)$?

- We can use **Coordinate Ascent** (CAVI): iteratively optimize each factor q_j while holding the other distributions on $\mathbf{z}_{-j}$ fixed.
- This climbs the ELBO to a local optimum.

Optimal Coordinate Update

The complete conditional of z_j is its conditional density given all other latent variables and data. The optimal update is:

$$q_j^*(z_j) \propto \exp\left\{\mathbf{E}_{-j}[\log p(z_j, \mathbf{z}_{-j}, \mathbf{x})]\right\}$$

where $\mathbf{E}_{-j}[\cdot]$ denotes the expectation with respect to the currently fixed distributions $\prod_{\ell \neq j} q_\ell(z_\ell)$.

Algorithm: CAVI

Input: A model $p(\mathbf{x}, \mathbf{z})$, a data set $\mathbf{x}$.
Output: A variational density $q(\mathbf{z}) = \prod_{j=1}^{m} q_j(z_j)$.

Initialize: Variational factors $q_j(z_j)$.

1. **While** the ELBO has not converged **do**:
   1. **For** $j \in \{1, \ldots, m\}$ **do**:
      - Set $q_j(z_j) \propto \exp\{\mathbf{E}_{-j}[\log p(z_j | \mathbf{z}_{-j}, \mathbf{x})]\}$
   2. Compute $\text{ELBO}(q) = \mathbf{E}[\log p(\mathbf{z}, \mathbf{x})] - \mathbf{E}[\log q(\mathbf{z})]$
2. **Return** $q(\mathbf{z})$

Connection: CAVI is closely related to Gibbs sampling. While Gibbs samples from the complete conditional, CAVI takes the *expected log* of the complete conditional.

MCMC: Pros & Cons

Pros of MCMC:

- Accurate results (at least asymptotically)
- Flexibility
- No approximation
- Handles multimodal distributions

Cons of MCMC:

- High computational cost
- Requires tuning of hyperparameters
- Convergence issues
- Inefficient in sampling complex dependencies

SVI: Pros & Cons

Pros of SVI:

- Faster convergence
- Scalability
- Ease of use

Cons of SVI:

- Approximate results
- Limited flexibility
- Mode seeking (cf. slide 13)
- Sensitive to choice of hyperparameters