Introduction to Probabilistic Machine Learning

Summer School on Methods for Statistical Evaluation of AI

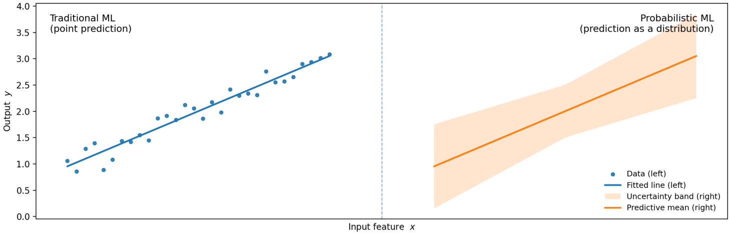
Andrés Masegosa

[Material made by Thomas D. Nielsen and Helge Langseth]

Introduction

From Predictions to Uncertainty-Aware Predictions

- Traditional ML: learns a function $f(x) \approx y$
  - Outputs: single prediction
  - Ignores uncertainty
- Probabilistic ML:
  - Models **distributions**, not just points
  - Explicitly represents **uncertainty**
  - Principles: *Models + Inference*



Prediction as a distribution (uncertainty included).

Why Uncertainty Matters in Practice

In the real world, predictions are not enough. We need to know *how confident* they are.

- **Healthcare:** Misdiagnosis risk - uncertainty flags when to call a human expert.
- **Autonomous driving:** Preventing Fatal Errors - uncertainty flags when driving under conditions not included in the train data.
- **General ML systems:** Uncertainty enables robustness, outlier detection, and better decision-making.

Probabilistic ML = Predictions + Confidence $\rightarrow$ *Safer, more trustworthy AI.*

Bayesian Machine Learning = Probabilistic model + Bayesian inference

- **Likelihood-part:** A probabilistic model typically defined by $p(\mathbf{y} \mid \mathbf{x}, \boldsymbol{\theta})$.
- **Prior**: $p(\boldsymbol{\theta})$ reflects our *a priori* belief about the parameters $\boldsymbol{\theta}$.

Bayesian Machine Learning = Probabilistic model + Bayesian inference

- **Likelihood-part:** A probabilistic model typically defined by $p(\mathbf{y} \mid \mathbf{x}, \boldsymbol{\theta})$.
- **Prior**: $p(\boldsymbol{\theta})$ reflects our *a priori* belief about the parameters $\boldsymbol{\theta}$.

Now we can calculate the posterior over $\boldsymbol{\theta}$ given training data $\mathcal{D}$,

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

... and, e.g., the predictive distribution of a new observation $\mathbf{x}'$:

$$p(\mathbf{y}' \mid \mathbf{x}' \mathcal{D}) = \int_{\boldsymbol{\theta}} p(\mathbf{y}' \mid \mathbf{x}', \boldsymbol{\theta}) p(\boldsymbol{\theta} \mid \mathcal{D})\, \mathrm{d}\boldsymbol{\theta}.$$

Bayesian Machine Learning = Probabilistic model + Bayesian inference

- **Likelihood-part:** A probabilistic model typically defined by $p(\mathbf{y} \mid \mathbf{x}, \boldsymbol{\theta})$.
- **Prior**: $p(\boldsymbol{\theta})$ reflects our *a priori* belief about the parameters $\boldsymbol{\theta}$.

Now we can calculate the posterior over $\boldsymbol{\theta}$ given training data $\mathcal{D}$,

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

... and, e.g., the predictive distribution of a new observation $\mathbf{x}'$:

$$p(\mathbf{y}' \mid \mathbf{x}'\mathcal{D}) = \int_{\boldsymbol{\theta}} p(\mathbf{y}' \mid \mathbf{x}', \boldsymbol{\theta}) p(\boldsymbol{\theta} \mid \mathcal{D})\, \mathrm{d}\boldsymbol{\theta}.$$

Being Bayesian means maintaining a distribution over θ.
Using a point-estimate for $\boldsymbol{\theta}$ is not **Bayesian** ML.

Example: Linear regression

A Bayesian linear regression with univariate explanatory variables:

Likelihood – $p(\mathcal{D} \mid \boldsymbol{\theta})$: $p(y_i \mid x_i, \mathbf{w}, \sigma_y^2) = \mathcal{N}\left(w_0 + w_1 \cdot x_i, \sigma_y^2\right)$

Note! The observation noise, σ_y^2, is known, so the parameter-set is simply $\boldsymbol{\theta} = \{\mathbf{w}\}$.

Prior – $p(\boldsymbol{\theta})$: $p(\mathbf{w}) = \mathcal{N}\left(\mathbf{0}, \sigma_w^2\right)$

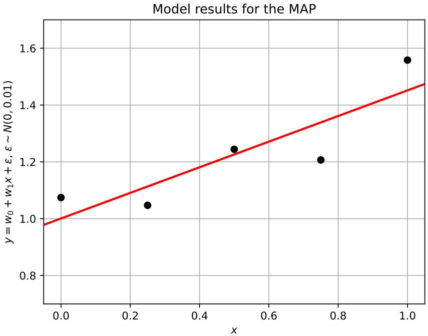
Bayesian Linear regression – Full model:

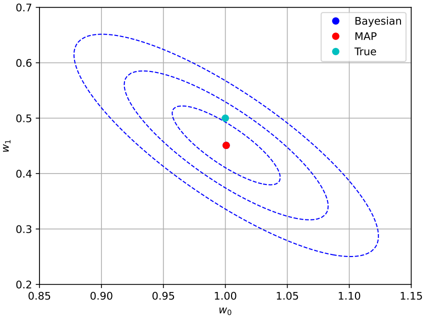
$$p(\mathcal{D}, \boldsymbol{\theta}) = p\left(\{y_i\}_{i=1}^n, \mathbf{w} \mid \{\mathbf{x}_i\}_{i=1}^n, \sigma_y^2, \sigma_w^2\right) = \overbrace{p(\mathbf{w} \mid \sigma_w^2)}^{p(\boldsymbol{\theta})} \overbrace{\prod_{i=1}^n p(y_i \mid \mathbf{w}, \mathbf{x}_i, \sigma_y^2)}^{p(\mathcal{D} \mid \boldsymbol{\theta})}$$

Bayes linear regression with some fake data:

- We have generated $N = 5$ examples from $y_i = 1.0 + 0.5 \cdot x_i + \epsilon_i$, $\epsilon_i \sim \mathcal{N}\left(0, 0.1^2\right)$.
- Weights unknown a priori, so here we use the vague priors $w_j \sim \mathcal{N}\left(0, 10^2\right)$.

Results for the MAP and the fully Bayesian model:





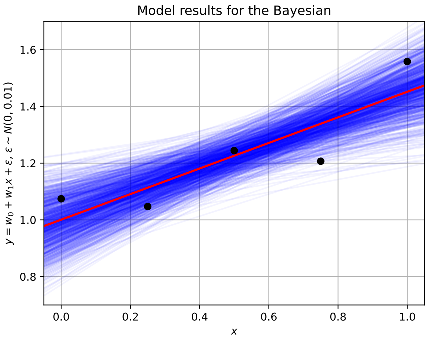
- **MAP:** Reasonable point estimate; No model uncertainty;
- **Bayes:** Model uncertainty around same MAP estimate;

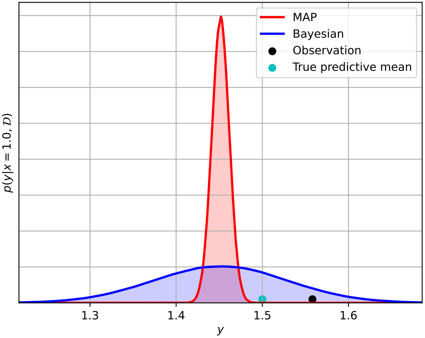
Example: Linear regression – MAP vs (fully) Bayesian

Bayes linear regression with some fake data:

- We have generated $N = 5$ examples from $y_i = 1.0 + 0.5 \cdot x_i + \epsilon_i$, $\epsilon_i \sim \mathcal{N}\left(0, 0.1^2\right)$.
- Weights unknown a priori, so here we use the vague priors $w_j \sim \mathcal{N}\left(0, 10^2\right)$.

Results for the MAP and the fully Bayesian model:





- **MAP:** Reasonable point estimate; No model uncertainty; Predictive uncertainty degenerated to observation noise: poor fit wrt. true value and observation.
- **Bayes:** Model uncertainty around same MAP estimate; Captures model uncertainty well; Predictive distribution reasonable.

Bayesian inference is in principle easy using Bayes' rule:

$$p(\boldsymbol{\theta} \mid \mathcal{D}) = \frac{p(\boldsymbol{\theta})\, p(\mathcal{D} \mid \boldsymbol{\theta})}{p(\mathcal{D})} = \frac{p(\mathcal{D}, \boldsymbol{\theta})}{\int_{\boldsymbol{\theta}} p(\boldsymbol{\theta})\, p(\mathcal{D} \mid \boldsymbol{\theta})\, \mathrm{d}\boldsymbol{\theta}}$$

Note! This can only be solved analytically for **some simple models** (e.g., linear regression), but typically not for the really interesting models.

We need to approximate $p(\boldsymbol{\theta} \mid \mathcal{D})$

What we want:

- Computationally efficient;
- Well-founded approach;
- Easy integration with other frameworks.

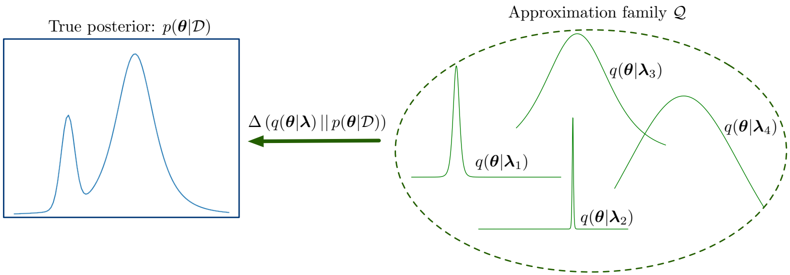
What we don't want:

- Non scalable solutions;
- Widely applicable.

FUNDAMENTAL assumption:
It will always be *computationally efficient* to evaluate $p(\mathcal{D}, \boldsymbol{\theta})$ at any given point $\{\mathcal{D}, \boldsymbol{\theta}\}$, e.g., using the simple factorization $p(\mathcal{D}, \boldsymbol{\theta}) = p(\boldsymbol{\theta}) \cdot p(\mathcal{D} \mid \boldsymbol{\theta}) = p(\boldsymbol{\theta}) \prod_i p(\mathbf{x}_i \mid \boldsymbol{\theta})$.

Variational Bayes: Approximate inference by optimization

Variational Inference: Approximate the true posterior distribution $p(\boldsymbol{\theta} \mid \mathcal{D})$ with a **variational distribution** from a tractable family of distributions $\mathcal{Q}$. The family is indexed by the parameters $\boldsymbol{\lambda}$.



Approximate inference through optimization

- **General goal:** Somehow approximate $p(\boldsymbol{\theta} \mid \mathcal{D})$ with a $q(\boldsymbol{\theta} \mid \mathcal{D})$.
    - **Note!** We use $q(\boldsymbol{\theta})$ as a short-hand for $q(\boldsymbol{\theta} \mid \mathcal{D})$.

Formalization of approximate inference through optimization:

Given a family of tractable distributions $\mathcal{Q}$ and a distance measure between distributions Δ, choose

$$\hat{q}(\boldsymbol{\theta}) = \arg\min_{q \in \mathcal{Q}} \Delta\big(q(\boldsymbol{\theta}) \,||\, p(\boldsymbol{\theta} \mid \mathcal{D})\big).$$

Decisions to be made:

1. How to define $\Delta(\cdot||\cdot)$ so that we end up with a high-quality solution?
    - How to work with $\Delta\big(q(\boldsymbol{\theta}) \,||\, p(\boldsymbol{\theta} \mid \mathcal{D})\big)$ when we don't know what $p(\boldsymbol{\theta} \mid \mathcal{D})$ is?
2. How to define a family of distributions $\mathcal{Q}$ that is both flexible enough to generate good approximations and restrictive enough to support efficient calculations?

Standard choice when working with probability distributions

The **Kullback-Leibler divergence** is the standard distance measure:

$$\mathrm{KL}\left(f||g\right) = \int_{\boldsymbol{\theta}} f(\boldsymbol{\theta}) \log\left(\frac{f(\boldsymbol{\theta})}{g(\boldsymbol{\theta})}\right) \mathrm{d}\boldsymbol{\theta} = \mathbb{E}_{\boldsymbol{\theta}\sim f}\left[\log\left(\frac{f(\boldsymbol{\theta})}{g(\boldsymbol{\theta})}\right)\right].$$

Notice that while $\mathrm{KL}\left(f||g\right)$ obeys the positivity criterion, it satisfies neither symmetry nor the triangle inequality. It is thus **not a proper distance measure**.

Two alternative KL definitions: $\mathrm{KL}(q||p)$ or $\mathrm{KL}(p||q)$?

Information-projection

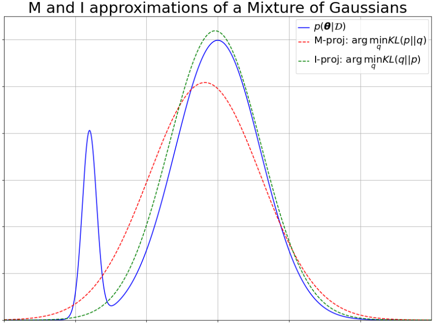
- Minimizes $\mathrm{KL}(q||p) = -\mathbb{E}_{\boldsymbol{\theta}\sim q}[\log p(\boldsymbol{\theta} \mid \mathcal{D})] - \mathcal{H}_q$.
- Preference given to q that has:
  1. High q-probability allocated to p-probable regions.
  2. Small q in any region where p is small. "$p(\boldsymbol{\theta} \mid \mathcal{D}) \approx 0 \implies q(\boldsymbol{\theta}) \approx 0$".
  3. High entropy ($\sim$ variance)

Moment-projection

- Minimizes $\mathrm{KL}(p||q) = -\mathbb{E}_{\boldsymbol{\theta}\sim p}[\log q(\boldsymbol{\theta})] - \mathcal{H}_p$.
- Preference given to q that has:
  1. High q-probability allocated to p-probable regions.
  2. $q(\boldsymbol{\theta}) > 0$ in any region where p is non-negligible. "$p(\boldsymbol{\theta} \mid \mathcal{D}) > 0 \implies q(\boldsymbol{\theta}) > 0$"
  3. No explicit focus of entropy

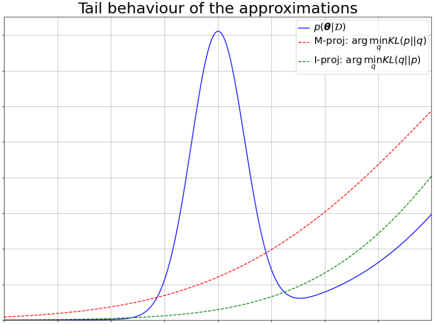
Cheat-sheet:

- **KL-divergence:** $\mathrm{KL}(f||g) = \mathbb{E}_f\left[\log\left(\frac{f(\boldsymbol{\theta})}{g(\boldsymbol{\theta})}\right)\right] = -\mathbb{E}_f\left[\log\left(g(\boldsymbol{\theta})\right)\right] - \mathcal{H}_f$.
- **Entropy:** $\mathcal{H}_f = -\int_{\boldsymbol{\theta}} f(\boldsymbol{\theta}) \log\left(f(\boldsymbol{\theta})\right) \mathrm{d}\boldsymbol{\theta} = -\mathbb{E}_f\left[\log\left(f(\boldsymbol{\theta})\right)\right]$.
- **Intuition:** Cheat a bit (measure-zero, limit-zero-rates, etc.) and think *"If $g(\boldsymbol{\theta}_0) \approx 0$, then $-\mathbb{E}_{\boldsymbol{\theta}\sim f}[\log g(\boldsymbol{\theta})]$ becomes 'huge' unless $f(\boldsymbol{\theta}_0) \approx 0$"* because $\lim_{x\to 0^+} \log(x)$ diverges, while $\lim_{x\to 0^+} x \cdot \log(x) = 0$.



Example: Approximating a Mix-of-Gaussians by a single Gaussian

- Moment projection – optimizing $\mathrm{KL}\,(p||q)$ – has slightly larger variance.
- Similar mean values, but Information projection – optimizing $\mathrm{KL}\,(q||p)$ – focuses mainly on the most prominent mode.



Example: Approximating a Mix-of-Gaussians by a single Gaussian

- Moment projection – optimizing $\mathrm{KL}\,(p||q)$ – has slightly larger variance.
- Similar mean values, but Information projection – optimizing $\mathrm{KL}\,(q||p)$ – focuses mainly on the most prominent mode.
- M-projection is **zero-avoiding**, while I-projection is **zero-forcing**.

Variational Bayes setup

VB uses information projections:

Variational Bayes relies on **information projections**, i.e., approximates $p(\boldsymbol{\theta} \mid \mathcal{D})$ by

$$\hat{q}(\boldsymbol{\theta}) = \arg\min_{q \in \mathcal{Q}} \mathrm{KL}\left(q(\boldsymbol{\theta}) || p(\boldsymbol{\theta} \mid \mathcal{D})\right)$$

- **Positives:**
  - Clever interpretation when used for Bayesian machine learning.
    - We will end up with an objective that lower-bounds the marginal log likelihood, $\log p(\mathcal{D})$.
  - Very efficient when combined with cleverly chosen $\mathcal{Q}$.
- **Negatives:**
  - May result in *zero-forcing* behaviour.
    - Typical choice of $\mathcal{Q}$ can make this issue even more prominent.

Notice how we can rearrange the KL divergence as follows:

$$\mathrm{KL}\left(q(\boldsymbol{\theta})||p(\boldsymbol{\theta}\,|\,\mathcal{D})\right) \quad = \quad \mathbb{E}_{\boldsymbol{\theta}\sim q}\left[\log \frac{q(\boldsymbol{\theta})}{p(\boldsymbol{\theta}\,|\,\mathcal{D})}\right]$$

Notice how we can rearrange the KL divergence as follows:

$$\boxed{\mathrm{KL}\left(q(\boldsymbol{\theta})||p(\boldsymbol{\theta}\,|\,\mathcal{D})\right)} = \mathbb{E}_{\boldsymbol{\theta}\sim q}\left[\log\frac{q(\boldsymbol{\theta})}{p(\boldsymbol{\theta}\,|\,\mathcal{D})}\right] = \mathbb{E}_{\boldsymbol{\theta}\sim q}\left[\log\frac{q(\boldsymbol{\theta})\cdot p(\mathcal{D})}{p(\boldsymbol{\theta}\,|\,\mathcal{D})\cdot p(\mathcal{D})}\right]$$

ELBO: Evidence Lower-BOund

Notice how we can rearrange the KL divergence as follows:

$$\begin{aligned}
\boxed{\mathrm{KL}\left(q(\boldsymbol{\theta}) || p(\boldsymbol{\theta} \mid \mathcal{D})\right)} \quad &= \quad \mathbb{E}_{\boldsymbol{\theta}\sim q}\left[\log \frac{q(\boldsymbol{\theta})}{p(\boldsymbol{\theta} \mid \mathcal{D})}\right] = \mathbb{E}_{\boldsymbol{\theta}\sim q}\left[\log \frac{q(\boldsymbol{\theta}) \cdot p(\mathcal{D})}{p(\boldsymbol{\theta} \mid \mathcal{D}) \cdot p(\mathcal{D})}\right] \\
&= \quad \log p(\mathcal{D}) - -\mathbb{E}_{\boldsymbol{\theta}\sim q}\left[\log \frac{q(\boldsymbol{\theta})}{p(\boldsymbol{\theta}, \mathcal{D})}\right]
\end{aligned}$$

ELBO: Evidence Lower-BOund

Notice how we can rearrange the KL divergence as follows:

$$\begin{aligned} \mathrm{KL}\left(q(\boldsymbol{\theta})||p(\boldsymbol{\theta}\,|\,\mathcal{D})\right) &= \mathbb{E}_{\boldsymbol{\theta}\sim q}\left[\log \frac{q(\boldsymbol{\theta})}{p(\boldsymbol{\theta}\,|\,\mathcal{D})}\right] = \mathbb{E}_{\boldsymbol{\theta}\sim q}\left[\log \frac{q(\boldsymbol{\theta})\cdot p(\mathcal{D})}{p(\boldsymbol{\theta}\,|\,\mathcal{D})\cdot p(\mathcal{D})}\right] \\ &= \log p(\mathcal{D}) - -\mathbb{E}_{\boldsymbol{\theta}\sim q}\left[\log \frac{q(\boldsymbol{\theta})}{p(\boldsymbol{\theta},\mathcal{D})}\right] = \log p(\mathcal{D}) - \mathcal{L}(q) \end{aligned}$$

Evidence Lower Bound (ELBO): $\mathcal{L}(q) = -\mathbb{E}_{\boldsymbol{\theta}\sim q}\left[\log \frac{q(\boldsymbol{\theta})}{p(\boldsymbol{\theta},\mathcal{D})}\right] = \mathbb{E}_{\boldsymbol{\theta}\sim q}\left[\log \frac{p(\boldsymbol{\theta},\mathcal{D})}{q(\boldsymbol{\theta})}\right]$.

ELBO: Evidence Lower-BOund

Notice how we can rearrange the KL divergence as follows:

$$\mathrm{KL}\left(q(\boldsymbol{\theta})||p(\boldsymbol{\theta}\,|\,\mathcal{D})\right) = \mathbb{E}_{\boldsymbol{\theta}\sim q}\left[\log\frac{q(\boldsymbol{\theta})}{p(\boldsymbol{\theta}\,|\,\mathcal{D})}\right] = \mathbb{E}_{\boldsymbol{\theta}\sim q}\left[\log\frac{q(\boldsymbol{\theta})\cdot p(\mathcal{D})}{p(\boldsymbol{\theta}\,|\,\mathcal{D})\cdot p(\mathcal{D})}\right]$$

$$= \log p(\mathcal{D}) - -\mathbb{E}_{\boldsymbol{\theta}\sim q}\left[\log\frac{q(\boldsymbol{\theta})}{p(\boldsymbol{\theta},\mathcal{D})}\right] = \log p(\mathcal{D}) - \mathcal{L}\left(q\right)$$

Evidence Lower Bound (ELBO): $\mathcal{L}\left(q\right) = -\mathbb{E}_{\boldsymbol{\theta}\sim q}\left[\log\frac{q(\boldsymbol{\theta})}{p(\boldsymbol{\theta},\mathcal{D})}\right] = \mathbb{E}_{\boldsymbol{\theta}\sim q}\left[\log\frac{p(\boldsymbol{\theta},\mathcal{D})}{q(\boldsymbol{\theta})}\right]$.

VB focuses on ELBO:

$$\log p(\mathcal{D}) = \mathcal{L}\left(q\right) + \mathrm{KL}\left(q(\boldsymbol{\theta})||p(\boldsymbol{\theta}\,|\,\mathcal{D})\right)$$

Since $\log p(\mathcal{D})$ is constant wrt. the distribution q it follows:

- We can minimize $\mathrm{KL}\left(q(\boldsymbol{\theta})||p(\boldsymbol{\theta}\,|\,\mathcal{D})\right)$ by maximizing $\mathcal{L}\left(q\right)$
- This is **computationally simpler** because it uses $p(\boldsymbol{\theta},\mathcal{D})$ and not $p(\boldsymbol{\theta}\,|\,\mathcal{D})$.
- $\mathcal{L}\left(q\right)$ is a **lower bound** of $\log p(\mathcal{D})$ because $\mathrm{KL}\left(q(\boldsymbol{\theta})||p(\boldsymbol{\theta}\,|\,\mathcal{D})\right) \geq 0$.

$\rightsquigarrow$ Look for $\hat{q}(\boldsymbol{\theta}) = \arg\max_{q\in\mathcal{Q}}\mathcal{L}\left(q\right)$.

ELBO: Evidence Lower-BOund

Notice how we can rearrange the KL divergence as follows:

$$\begin{aligned}
\mathrm{KL}\left(q(\boldsymbol{\theta}) || p(\boldsymbol{\theta} \mid \mathcal{D})\right) &= \mathbb{E}_{\boldsymbol{\theta}\sim q}\left[\log \frac{q(\boldsymbol{\theta})}{p(\boldsymbol{\theta} \mid \mathcal{D})}\right] = \mathbb{E}_{\boldsymbol{\theta}\sim q}\left[\log \frac{q(\boldsymbol{\theta}) \cdot p(\mathcal{D})}{p(\boldsymbol{\theta} \mid \mathcal{D}) \cdot p(\mathcal{D})}\right] \\
&= \log p(\mathcal{D}) - -\mathbb{E}_{\boldsymbol{\theta}\sim q}\left[\log \frac{q(\boldsymbol{\theta})}{p(\boldsymbol{\theta}, \mathcal{D})}\right] = \log p(\mathcal{D}) - \mathcal{L}(q)
\end{aligned}$$

Evidence Lower Bound (ELBO): $\mathcal{L}(q) = -\mathbb{E}_{\boldsymbol{\theta}\sim q}\left[\log \frac{q(\boldsymbol{\theta})}{p(\boldsymbol{\theta}, \mathcal{D})}\right] = \mathbb{E}_{\boldsymbol{\theta}\sim q}\left[\log \frac{p(\boldsymbol{\theta}, \mathcal{D})}{q(\boldsymbol{\theta})}\right]$.

Summary:

- We started out looking for $\arg\min_{q\in\mathcal{Q}} \mathrm{KL}\left(q(\boldsymbol{\theta}) || p(\boldsymbol{\theta} \mid \mathcal{D})\right)$.
- Didn't know how to calculate $\mathrm{KL}\left(q(\boldsymbol{\theta}) || p(\boldsymbol{\theta} \mid \mathcal{D})\right)$ because $p(\boldsymbol{\theta} \mid \mathcal{D})$ is unknown.
- Still, we can find the optimal approximation by maximizing $\mathcal{L}(q)$:

$$\arg\max_{q\in\mathcal{Q}} \mathcal{L}(q) = \arg\min_{q\in\mathcal{Q}} \mathrm{KL}\left(q(\boldsymbol{\theta}) || p(\boldsymbol{\theta} \mid \mathcal{D})\right).$$

- It all makes sense: We aim to maximize $\mathcal{L}(q)$, which is a lower-bound of $\log p(\mathcal{D})$.

Variational Bayes w/ Mean Field

The mean field assumption

What we have ...

We now have the first building-block of the approximation:

$$\Delta(q\,||\,p) = \mathrm{KL}\left(q(\boldsymbol{\theta})||p(\boldsymbol{\theta}\,|\,\mathcal{D})\right),$$

and avoided the issue with $p(\boldsymbol{\theta}\,|\,\mathcal{D})$ by focusing on $\mathcal{L}\,(q)$.

We still need the set $\mathcal{Q}$:

Very often you will see the **mean field assumption**, which states that $\mathcal{Q}$ consists of distributions that **factorize** according to the equation

$$q(\boldsymbol{\theta}|\boldsymbol{\lambda}) = \prod_i q_i(\theta_i|\lambda_i).$$

This may seem like a very restricted set, but it often works well anyway ...

Wrapping it all up: The VB algorithm under MF

Setup:

- We have observed $\mathcal{D}$, and can calculate the full joint $p(\boldsymbol{\theta}, \mathcal{D}) = p(\boldsymbol{\theta}) \cdot p(\mathcal{D} \mid \boldsymbol{\theta})$.
- We use the ELBO as our objective, and assume $q(\boldsymbol{\theta})$ factorizes.
- We posit a *variational family* of distributions $q_i(\cdot \mid \boldsymbol{\lambda}_i)$, i.e., we choose the distributional form, while wanting to optimize the parameterization $\boldsymbol{\lambda}_i$.

- We then aim to solve the following continuous maximization problem:

$$\arg\max_{\boldsymbol{\lambda}} \mathcal{L}(\boldsymbol{\lambda}) = \arg\min_{q \in \mathcal{Q}} \mathrm{KL}\left(q(\boldsymbol{\theta}) || p(\boldsymbol{\theta} \mid \mathcal{D})\right).$$

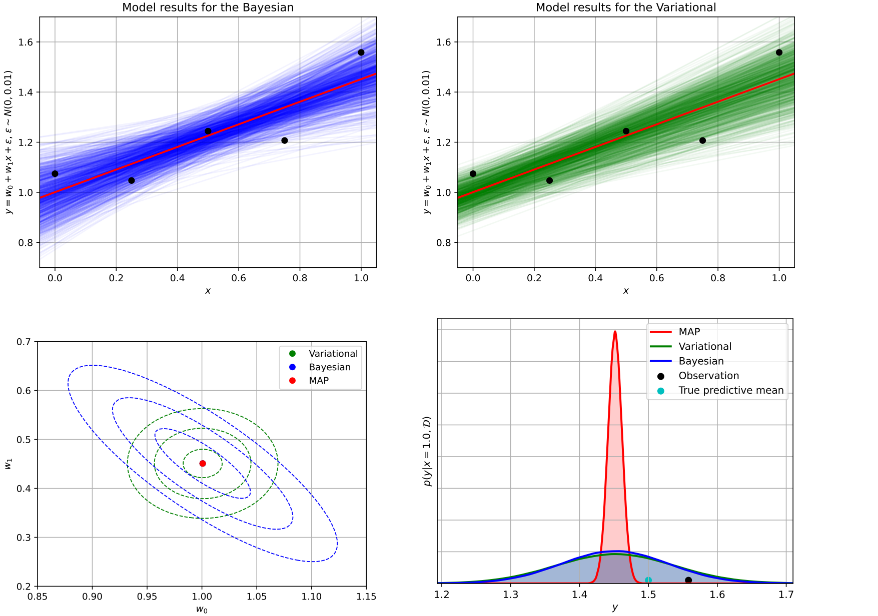
(Stochastic) Gradient ascent algorithm for maximizing a function L ($\boldsymbol{\lambda}$):

1. Initialize $\boldsymbol{\lambda}^{(0)}$ randomly.
2. For $t = 1, \ldots$:

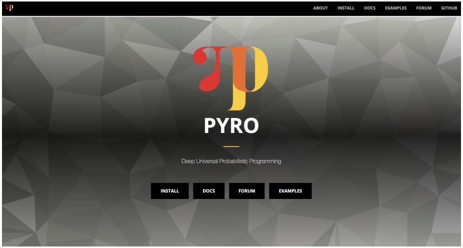
$$\boldsymbol{\lambda}^{(t)} \leftarrow \boldsymbol{\lambda}^{(t-1)} + \rho_t \cdot \mathcal{L}\left(\boldsymbol{\lambda}^{(t-1)}\right)$$

VB-MF example – "sanity check"

Bayes linear regression with likelihood $y_i \mid \{w_0, w_1, x_i, \sigma_y^2\} = \mathcal{N}(w_0 + w_1\, x_i, \sigma_y^2)$.



Probabilistic programming: Pyro



Pyro's main features (`www.pyro.ai`) :

- Initially developed by UBER (the car riding company).
- Community of contributors and a dedicated team at Broad Institute (US).
- Rely on Pytorch (Deep Learning Framework).
- Enable GPU accelaration and distributed learning.

Pyro

Pyro (`pyro.ai`) is a Python library for probabilistic machine learning integrated with PyTorch.

Modeling:
- Directed graphical models
- Neural networks (via `nn.Module`)
- ...

Inference:
- Variational inference – including BBVI, SVI
- Monte Carlo – including Importance sampling and Hamiltonian Monte Carlo
- ...

Criticism:
- Point-based evaluations
- Posterior predictive checks
- ...

https://github.com/PGM-Lab/2025-MSE-AI