Variational inference

Scalable solutions

Helge Langseth and Thomas Dyhre Nielsen

Oct. 2018

Introduction

Plan for this weeks

Day 1: Bayesian networks – Definition and inference

- Definition of Bayesian networks: Syntax and semantics
- Exact inference
- Approximate inference using MCMC

Day 2: Variational inference – Introduction and basis

- Approximate inference through the *Kullback-Leibler divergence*
- *Variational Bayes*
- The *mean-field* approach to Variational Bayes

Day 3: Variational Bayes – cont'd

- Solving the VB equations
- Introducing Exponential families

Day 4: Scalable Variational Bayes

- Variational message passing
- Stochastic gradient ascent
- Stochastic variational inference

Day 5: Current approaches and extensions

- Variational Auto Encoders
- Black Box variational inference
- Probabilistic Programming Languages

VB w/ MF: algorithm

Algorithm:

- We have observed $\mathbf{X} = \mathbf{x}$, and have access to the full joint $p(\mathbf{z}, \mathbf{x})$.
- We posit a *variational family* of distributions $q_j(\cdot \mid \boldsymbol{\lambda}_j)$, i.e., we choose the distributional form, while wanting to optimize the parameterization $\boldsymbol{\lambda}_j$.
- The posterior approximation is assumed to factorize according to the mean-field assumption, and we use the $\mathrm{KL}\left(q(\mathbf{z}) || p(\mathbf{z} \mid \mathbf{x})\right)$ as our objective.

Algorithm:

Repeat until negligible improvement in terms of $\mathcal{L}\left(q\right)$:

- For each j:
  - Calculate $\mathbb{E}_{q_{\neg j}}\left[\log p(\mathbf{z}, \mathbf{x})\right]$ using current estimates for $q_i(\cdot \mid \boldsymbol{\lambda}_i)$, $i \neq j$.
  - Choose $\boldsymbol{\lambda}_j$ so that $q_j(z_j \mid \boldsymbol{\lambda}_j) \propto \exp\left(\mathbb{E}_{q_{\neg j}}\left[\log p(\mathbf{z}, \mathbf{x})\right]\right)$.
- Calculate the new $\mathcal{L}\left(q\right)$.

As we realized last time, calculations of $\mathbb{E}_{q_{\neg j}}\left[\log p(\mathbf{z}, \mathbf{x})\right]$ and $\mathcal{L}\left(q\right)$ are quite tedious – and apparently must be done separately for each model we make.

This **harms the applicability** of variational inference, even under the **quite restrictive** mean field assumption.

The Exponential Family

Definition of ExpFam models

Consider a distribution $f_{\mathbf{X}}(\mathbf{x} \mid \boldsymbol{\theta})$, and assume it can be written as:

$$f_{\mathbf{X}}(\mathbf{x} \mid \boldsymbol{\eta}) = \exp\left(h(\mathbf{x}) + \boldsymbol{\eta}^{\mathsf{T}}\mathbf{t}(\mathbf{x}) - A(\boldsymbol{\eta})\right)$$

Here $h(\mathbf{x})$ is the log base measure, $\boldsymbol{\eta}$ the natural parameters, $\mathbf{t}(\mathbf{x})$ the sufficient statistics, and $A(\boldsymbol{\eta})$ the log partition function.

Positives with the Exponential Family:

- It is the only family of distributions with finite-sized sufficient statistics*;
- It is the only family of distributions that has conjugate priors;
- It simplifies the operations of variational inference;
- It has simple mathematical procedures for calculating moments, MLEs, Bayesian posteriors, ...

* Under certain regularity conditions...

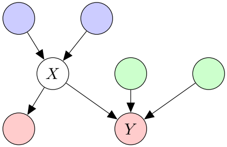
Examples

Distributions that are in the ExpFam family of distributions, and therefore have a shared overarching theory include Bernoulli, Gamma, Normal, Poisson, χ^2, Beta, Dirichlet, Categorical, Multinomial, and Wishart.

Variational message passing

Referring to the conditioning parameters as parents, consider a conditional distribution in exponential form

$$\log f(x \mid \mathrm{pa}(x)) = h_x(x) + \boldsymbol{\eta}_X(\mathrm{pa}(x))^{\mathsf{T}} \mathbf{t}_x(x) - A_x(\mathrm{pa}(x))$$



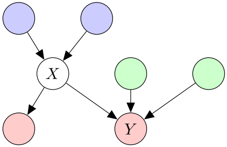
The Conjugate Exponential Family (abridged version)

Referring to the conditioning parameters as parents, consider a conditional distribution in exponential form

$$\log f(x \mid \mathrm{pa}(x)) = h_x(x) + \boldsymbol{\eta}_X(\mathrm{pa}(x))^\intercal \mathbf{t}_x(x) - A_x(\mathrm{pa}(x))$$

with a child y also in exponential form:

$$\log f(y \mid x, \mathrm{cp}(x)) = h_y(y) + \boldsymbol{\eta}_y(x, \mathrm{cp}(x))^\intercal \mathbf{t}_y(y) - A_y(x, \mathrm{cp}(x))$$



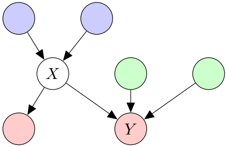
The Conjugate Exponential Family (abridged version)

Referring to the conditioning parameters as parents, consider a conditional distribution in exponential form

$$\log f(x \mid \mathrm{pa}(x)) = h_x(x) + \boldsymbol{\eta}_X(\mathrm{pa}(x))^\mathsf{T} \mathbf{t}_x(x) - A_x(\mathrm{pa}(x))$$

with a child y also in exponential form:

$$\log f(y \mid x, \mathrm{cp}(x)) = h_y(y) + \boldsymbol{\eta}_y(x, \mathrm{cp}(x))^\mathsf{T} \mathbf{t}_y(y) - A_y(x, \mathrm{cp}(x))$$



Conjugacy

Conjugacy requires that $\log f(x \mid \mathrm{pa}(x))$ and $\log f(y \mid x, \mathrm{cp}(x))$ have the same functional form wrt. x and so the latter can be written as:

$$\log f(y \mid x, \mathrm{cp}(x)) = \boldsymbol{\eta}_{xy}(y, \mathrm{cp}(x))^\mathsf{T} \mathbf{t}_x(x) - A_{xy}(y, \mathrm{cp}(x))$$

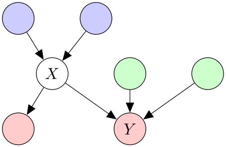
The Conjugate Exponential Family (abridged version)

Referring to the conditioning parameters as parents, consider a conditional distribution in exponential form

$$\log f(x \mid \mathrm{pa}(x)) = h_x(x) + \boldsymbol{\eta}_X(\mathrm{pa}(x))^\mathsf{T}\mathbf{t}_x(x) - A_x(\mathrm{pa}(x))$$

with a child y also in exponential form:

$$\log f(y \mid x, \mathrm{cp}(x)) = h_y(y) + \boldsymbol{\eta}_y(x, \mathrm{cp}(x))^\mathsf{T}\mathbf{t}_y(y) - A_y(x, \mathrm{cp}(x))$$



Conjugacy

Conjugacy requires that $\log f(x \mid \mathrm{pa}(x))$ and $\log f(y \mid x, \mathrm{cp}(x))$ have the same functional form wrt. x and so the latter can be written as:

$$\log f(y \mid x, \mathrm{cp}(x)) = \boldsymbol{\eta}_{xy}(y, \mathrm{cp}(x))^\mathsf{T}\mathbf{t}_x(x) - A_{xy}(y, \mathrm{cp}(x))$$

- $\log f(y \mid x, \mathrm{cp}(x))$ is linear in $\mathbf{t}_x(x)$ and $\mathbf{t}_y(y)$ (and in $\mathbf{t}_z(z)$ for any $z \in \mathrm{cp}(x)$).
- $\rightsquigarrow \log f(y \mid x, \mathrm{cp}(x))$ is a multi-linear function

The Conjugate Exponential Family: Example

Assume that X is normal distributed with mean m and precision b:

$$\log f(x \mid m, b) = \underbrace{-\frac{1}{2}\log(2\pi)}_{h(x)} + \underbrace{\begin{bmatrix} b \cdot m \\ -\frac{b}{2} \end{bmatrix}^\mathsf{T}}_{\boldsymbol{\eta}(m,b)^\mathsf{T}} \underbrace{\begin{bmatrix} x \\ x^2 \end{bmatrix}}_{\mathbf{t}(x)} - \underbrace{\left(\frac{b \cdot m^2}{2} - \frac{1}{2}\log b\right)}_{A(\boldsymbol{\eta}(m,b))}$$

Assume that X is normal distributed with mean m and precision b:

$$\log f(x \mid m, b) = \underbrace{-\frac{1}{2}\log(2\pi)}_{h(x)} + \underbrace{\begin{bmatrix} b \cdot m \\ -\frac{b}{2} \end{bmatrix}^{\mathsf{T}}}_{\boldsymbol{\eta}(m,b)^{\mathsf{T}}} \underbrace{\begin{bmatrix} x \\ x^2 \end{bmatrix}}_{\mathbf{t}(x)} - \underbrace{\left(\frac{b \cdot m^2}{2} - \frac{1}{2}\log b\right)}_{A(\boldsymbol{\eta}(m,b))}$$

Normal prior for mean

We can then express $f(x \mid m, b)$ in terms of m as

$$\log f(x \mid m, b) = -\frac{1}{2}\log(2\pi) + \begin{bmatrix} b \cdot x \\ -\frac{b}{2} \end{bmatrix}^{\mathsf{T}} \begin{bmatrix} m \\ m^2 \end{bmatrix} - \left(\frac{b \cdot x^2}{2} - \frac{1}{2}\log b\right)$$

Thus, conjugacy implies that the prior distribution over m is a normal distribution.

Assume that X is normal distributed with mean m and precision b:

$$\log f(x \mid m, b) = \underbrace{-\frac{1}{2}\log(2\pi)}_{h(x)} + \underbrace{\begin{bmatrix} b \cdot m \\ -\frac{b}{2} \end{bmatrix}^{\mathsf{T}}}_{\boldsymbol{\eta}(m,b)^{\mathsf{T}}} \underbrace{\begin{bmatrix} x \\ x^2 \end{bmatrix}}_{\mathbf{t}(x)} - \underbrace{\left(\frac{b \cdot m^2}{2} - \frac{1}{2}\log b\right)}_{A(\boldsymbol{\eta}(m,b))}$$

Normal prior for mean

We can then express $f(x \mid m, b)$ in terms of m as

$$\log f(x \mid m, b) = -\frac{1}{2}\log(2\pi) + \begin{bmatrix} b \cdot x \\ -\frac{b}{2} \end{bmatrix}^{\mathsf{T}} \begin{bmatrix} m \\ m^2 \end{bmatrix} - \left(\frac{b \cdot x^2}{2} - \frac{1}{2}\log b\right)$$

Thus, conjugacy implies that the prior distribution over m is a normal distribution.

Gamma prior for precision

We can also express $f(x \mid m, b)$ in terms of b:

$$\log f(x \mid m, b) = -\frac{1}{2}\log(2\pi) + \begin{bmatrix} -\frac{1}{2}(x-m)^2 \\ \frac{1}{2} \end{bmatrix}^{\mathsf{T}} \begin{bmatrix} b \\ \log(b) \end{bmatrix} - 0$$

Thus, implying that b follows a gamma distribution.

Variational message passing

Variational message passing is a variational-based inference algorithm for Bayesian networks:

- Applies to conjugate-exponential models (can be generalized to non-conjugate models)
- Works by sending local messages between nodes in the graphical model.

Advantage

No need for the tedious model-specific variational derivations that we saw during the last lecture.

Given a Bayesian network that factorizes according to

$$p(x_i, \dots x_n) = \prod_{i=i}^{N} p\left(x_i \mid \mathrm{pa}(x_i)\right)$$

we saw last time that the mean-field approximation for a variable X_j is given by

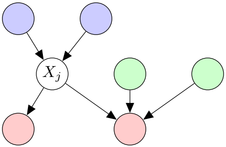
$$\begin{aligned} \log q(x_j) &= \mathbb{E}\left[\sum_{i=i}^{N} \log p\left(x_i \mid \mathrm{pa}(x_i)\right)\right] + c \\ &= \mathbb{E}\left[\log p\left(x_j \mid \mathrm{pa}(x_j)\right)\right] + \mathbb{E}\left[\sum_{y \in \mathrm{ch}(x_j)} \log p\left(y \mid x_j, \mathrm{cp}(x_j)\right)\right] + c' \end{aligned}$$

Given a Bayesian network that factorizes according to

$$p(x_i, \ldots x_n) = \prod_{i=i}^{N} p\left(x_i \mid \mathrm{pa}(x_i)\right)$$

we saw last time that the mean-field approximation for a variable X_j is given by

$$\log q(x_j) \quad = \quad \mathbb{E}\left[\log p\left(x_j \mid \mathrm{pa}(x_j)\right)\right] + \mathbb{E}\left[\sum_{y \in \mathrm{ch}(x_j)} \log p\left(y \mid x_j, \mathrm{cp}(x_j)\right)\right] + c'$$



Therefore, the only contributions to $q(x_j)$ come from

- X_j's parents through the term $\mathbb{E}\left[\log p\left(x_j \mid \mathrm{pa}(x_j)\right)\right]$.
- X_j's children through the term $\mathbb{E}\left[\sum_{y \in \mathrm{ch}(x_i)} \log p\left(y \mid \mathrm{pa}(y)\right)\right]$.
- X_j's co-parents through the term $\mathbb{E}\left[\sum_{y \in \mathrm{ch}(x_i)} \log p\left(y \mid x_j, \mathrm{cp}(x_j)\right)\right]$.

This is exactly **the Markov Blanket** for X_j, which we also exploited during Gibbs sampling and our previous VI derivations.

VMP expressed in exponential form

The distributions involved in the mean-field approximation for variable X_j

$$\log q(x_j) = \mathbb{E}\left[\log p\left(x_j \mid \mathrm{pa}(x_j)\right)\right] + \mathbb{E}\left[\sum_{y \in \mathrm{ch}(x_j)} \log p\left(y \mid x_j, \mathrm{cp}(x_j)\right)\right] + c'$$

can be expressed in exponential form:

- $\log p(x_j \mid \mathrm{pa}(x_j)) = h_{X_j}(x_j) + \boldsymbol{\eta}_{X_j}(\mathrm{pa}(x_j))^{\mathsf{T}} \mathbf{t}(x_j) - A_{X_j}(\boldsymbol{\eta}_{X_j}(\mathrm{pa}(x_j)))$
- $\log p(y \mid x_j, \mathrm{cp}(x_j)) = h_Y(y) + \boldsymbol{\eta}_Y(x_j, \mathrm{cp}(x_j))^{\mathsf{T}} \mathbf{t}(y) - A_Y(\boldsymbol{\eta}_{X_j}(x_j, \mathrm{cp}(x_j)))$

VMP expressed in exponential form

The distributions involved in the mean-field approximation for variable X_j

$$\log q(x_j) = \mathbb{E}\left[\log p\left(x_j \mid \mathrm{pa}(x_j)\right)\right] + \mathbb{E}\left[\sum_{y \in \mathrm{ch}(x_j)} \log p\left(y \mid x_j, \mathrm{cp}(x_j)\right)\right] + c'$$

can be expressed in exponential form:

- $\log p(x_j \mid \mathrm{pa}(x_j)) = h_{X_j}(x_j) + \boldsymbol{\eta}_{X_j}(\mathrm{pa}(x_j))^\intercal \mathbf{t}(x_j) - A_{X_j}(\boldsymbol{\eta}_{X_j}(\mathrm{pa}(x_j)))$
- $\log p(y \mid x_j, \mathrm{cp}(x_j)) = h_Y(y) + \boldsymbol{\eta}_Y(x_j, \mathrm{cp}(x_j))^\intercal \mathbf{t}(y) - A_Y(\boldsymbol{\eta}_{X_j}(x_j, \mathrm{cp}(x_j)))$

Due to conjugacy, we can rewrite $\log p(y \mid x_j, \mathrm{cp}(x_j))$ in terms of $\mathbf{t}(x_j)$:

$$\log p(y \mid x_j, \mathrm{cp}(x_j)) = \boldsymbol{\eta}_{X_j,Y}(y, \mathrm{cp}(x_j))^\intercal \mathbf{t}(x_j) - A_{X_j,Y}(\boldsymbol{\eta}_{X_j,Y}(y, \mathrm{cp}(x_j)))$$

Thus, we end up with:

$$\log q(x_j) = \mathbb{E}\left[h_{X_j}(x_j) + \boldsymbol{\eta}_{X_j}(\mathrm{pa}(x_j))^\intercal \mathbf{t}(x_j) - A_{X_j}(\boldsymbol{\eta}_{X_j}(\mathrm{pa}(x_j)))\right]$$
$$+ \mathbb{E}\left[\sum_{Y \in \mathrm{ch}(x_j)} \boldsymbol{\eta}_{X_j,Y}(Y, \mathrm{cp}(x_j))^\intercal \mathbf{t}(x_j) - A_{X_j,Y}(\boldsymbol{\eta}_{X_j,Y}(Y, \mathrm{cp}(x_j)))\right] + c'$$

The distributions involved in the mean-field approximation for variable X_j

$$\log q(x_j) = \mathbb{E}\left[\log p\left(x_j \mid \mathrm{pa}(x_j)\right)\right] + \mathbb{E}\left[\sum_{y \in \mathrm{ch}(x_j)} \log p\left(y \mid x_j, \mathrm{cp}(x_j)\right)\right] + c'$$

can be expressed in exponential form:

- $\log p(x_j \mid \mathrm{pa}(x_j)) = h_{X_j}(x_j) + \boldsymbol{\eta}_{X_j}(\mathrm{pa}(x_j))^{\mathsf{T}}\mathbf{t}(x_j) - A_{X_j}(\boldsymbol{\eta}_{X_j}(\mathrm{pa}(x_j)))$
- $\log p(y \mid x_j, \mathrm{cp}(x_j)) = h_Y(y) + \boldsymbol{\eta}_Y(x_j, \mathrm{cp}(x_j))^{\mathsf{T}}\mathbf{t}(y) - A_Y(\boldsymbol{\eta}_{X_j}(x_j, \mathrm{cp}(x_j)))$

Due to conjugacy, we can rewrite $\log p(y \mid x_j, \mathrm{cp}(x_j))$ in terms of $\mathbf{t}(x_j)$:

$$\log p(y \mid x_j, \mathrm{cp}(x_j)) = \boldsymbol{\eta}_{X_j,Y}(y, \mathrm{cp}(x_j))^{\mathsf{T}}\mathbf{t}(x_j) - A_{X_j,Y}(\boldsymbol{\eta}_{X_j,Y}(y, \mathrm{cp}(x_j)))$$

Rearranging and absorbing into constant c gives:

$$\log q(x_j) = \left[\mathbb{E}(\boldsymbol{\eta}_{X_j}(\mathrm{pa}(x_j))) + \sum_{Y \in \mathrm{ch}(x_j)} \mathbb{E}(\boldsymbol{\eta}_{X_j,Y}(Y, \mathrm{cp}(x_j)))\right]^{\mathsf{T}}\mathbf{t}(x_j) + h_{X_j}(x_j) + c$$

Calculating the expectations in VMP

We have:

$$\log q(x_j) = \left[\mathbb{E}(\boldsymbol{\eta}_{X_j}(\mathrm{pa}(x_j))) + \sum_{Y \in \mathrm{ch}(x_j)} \mathbb{E}(\boldsymbol{\eta}_{X_j,Y}(Y, \mathrm{cp}(x_j)))\right]^{\mathsf{T}} \mathbf{t}(x_j) + h_{X_j}(x_j) + c$$

As seen before, both $\boldsymbol{\eta}_{X_j}(\mathrm{pa}(x_j))$ and each $\boldsymbol{\eta}_{X_j,Y}(y, \mathrm{cp}(x_j)))$ are multi-linear functions of the natural statistics vectors of their dependent variables, hence:

$$\mathbb{E}(\boldsymbol{\eta}_{X_j}(\mathrm{pa}(x_j))) = \tilde{\boldsymbol{\eta}}_{X_j}(\{\mathbb{E}(\mathbf{t}(X_i))\}_{X_i \in \mathrm{pa}(X_j)})$$

$$\mathbb{E}(\boldsymbol{\eta}_{X_j Y}(y, \mathrm{cp}(x_j))) = \tilde{\boldsymbol{\eta}}_{X_j,Y}(\mathbb{E}(\mathbf{t}(Y)), \{\mathbb{E}(\mathbf{t}(X_i))\}_{X_i \in \mathrm{cp}(X_j)}).$$

Calculating the expectations in VMP

We have:

$$\log q(x_j) = \left[\mathbb{E}(\boldsymbol{\eta}_{X_j}(\mathrm{pa}(x_j))) + \sum_{Y \in \mathrm{ch}(x_j)} \mathbb{E}(\boldsymbol{\eta}_{X_j,Y}(Y, \mathrm{cp}(x_j)))\right]^{\mathsf{T}} \mathbf{t}(x_j) + h_{X_j}(x_j) + c$$

As seen before, both $\boldsymbol{\eta}_{X_j}(\mathrm{pa}(x_j))$ and each $\boldsymbol{\eta}_{X_j,Y}(y, \mathrm{cp}(x_j)))$ are multi-linear functions of the natural statistics vectors of their dependent variables, hence:

$$\mathbb{E}(\boldsymbol{\eta}_{X_j}(\mathrm{pa}(x_j))) = \tilde{\boldsymbol{\eta}}_{X_j}(\{\mathbb{E}(\mathbf{t}(X_i))\}_{X_i \in \mathrm{pa}(X_j)})$$

$$\mathbb{E}(\boldsymbol{\eta}_{X_j Y}(y, \mathrm{cp}(x_j))) = \tilde{\boldsymbol{\eta}}_{X_j,Y}(\mathbb{E}(\mathbf{t}(Y)), \{\mathbb{E}(\mathbf{t}(X_i))\}_{X_i \in \mathrm{cp}(X_j)}).$$

The expectations over the individual natural statistics can be found using the log-normalizer trick:

$$\nabla A_X(\boldsymbol{\eta}) = \mathbb{E}(\mathbf{t}(X)).$$

Calculating the expectations in VMP

We have:

$$\log q(x_j) = \left[\mathbb{E}(\boldsymbol{\eta}_{X_j}(\mathrm{pa}(x_j))) + \sum_{Y\in\mathrm{ch}(x_j)} \mathbb{E}(\boldsymbol{\eta}_{X_j,Y}(Y,\mathrm{cp}(x_j)))\right]^{\mathsf{T}} \mathbf{t}(x_j) + h_{X_j}(x_j) + c$$

As seen before, both $\boldsymbol{\eta}_{X_j}(\mathrm{pa}(x_j))$ and each $\boldsymbol{\eta}_{X_j,Y}(y,\mathrm{cp}(x_j)))$ are multi-linear functions of the natural statistics vectors of their dependent variables, hence:

$$\mathbb{E}(\boldsymbol{\eta}_{X_j}(\mathrm{pa}(x_j))) = \tilde{\boldsymbol{\eta}}_{X_j}(\{\mathbb{E}(\mathbf{t}(X_i))\}_{X_i\in\mathrm{pa}(X_j)})$$

$$\mathbb{E}(\boldsymbol{\eta}_{X_jY}(y,\mathrm{cp}(x_j))) = \tilde{\boldsymbol{\eta}}_{X_j,Y}(\mathbb{E}(\mathbf{t}(Y)),\{\mathbb{E}(\mathbf{t}(X_i))\}_{X_i\in\mathrm{cp}(X_j)}).$$

The expectations over the individual natural statistics can be found using the log-normalizer trick:

$$\nabla A_X(\boldsymbol{\eta}) = \mathbb{E}(\mathbf{t}(X)).$$

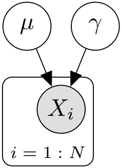
New natural parameter vector and implied message passing scheme

$$\boldsymbol{\eta}^*_{X_j} = \tilde{\boldsymbol{\eta}}_{X_j}\Big(\overbrace{\{\mathbb{E}(\mathbf{t}(X_i))\}_{X_i\in\mathrm{pa}(X_j)}}^{\text{Messages from parents: } \mathbf{m}_{X_i\to X_j}}\Big) + \sum_{Y\in\mathrm{ch}(X_j)} \underbrace{\tilde{\boldsymbol{\eta}}_{X_j,Y}(\mathbb{E}(\mathbf{t}(Y)),\{\mathbb{E}(\mathbf{t}(X_i))\}_{X_i\in\mathrm{cp}(X_j)})}_{\text{Messages from children: } \mathbf{m}_{Y\to X_j}}$$

VMP algorithm

1. Initialize each variational distribution $q(x_i)$ by its moment vector $\mathbb{E}(\mathbf{t}(x_i))$.
2. For each variable X_i:
    1. Retrieve messages from all parents and children.
    2. Computed new natural parameter vector $\boldsymbol{\eta}^*_{x_i}$.
    3. Computed new moment parameters $\mathbb{E}(\mathbf{t}(x_i))$.
3. Calculate ELBO if needed (not described here!)
4. Repeat from step 1 unless termination criteria reached.

Model specification



$$\log p(\gamma \mid \alpha, \beta) = \begin{bmatrix} -\beta \\ \alpha - 1 \end{bmatrix}^{\mathsf{T}} \begin{bmatrix} \gamma \\ \log(\gamma) \end{bmatrix} + c_1$$

$$\log p(\mu \mid m, b) = \begin{bmatrix} b \cdot m \\ -\frac{b}{2} \end{bmatrix}^{\mathsf{T}} \begin{bmatrix} \mu \\ \mu^2 \end{bmatrix} + c_2$$

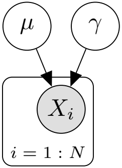
$$\log p(x_i \mid \mu, \gamma) = \begin{bmatrix} \gamma\mu \\ -\frac{\gamma}{2} \end{bmatrix}^{\mathsf{T}} \begin{bmatrix} x_i \\ x_i^2 \end{bmatrix} + c_3$$

$$= \begin{bmatrix} \gamma x_i \\ -\frac{\gamma}{2} \end{bmatrix}^{\mathsf{T}} \begin{bmatrix} \mu \\ \mu^2 \end{bmatrix} + c_4$$

$$= \begin{bmatrix} -\frac{1}{2}x_i^2 + x_i\mu - \frac{1}{2}\mu_i^2 \\ \frac{1}{2} \end{bmatrix}^{\mathsf{T}} \begin{bmatrix} \gamma \\ \log(\gamma) \end{bmatrix} + c_5$$

VMP example

Model specification



$$\log p(\gamma \mid \alpha, \beta) = \begin{bmatrix} -\beta \\ \alpha - 1 \end{bmatrix}^\intercal \begin{bmatrix} \gamma \\ \log(\gamma) \end{bmatrix} + c_1$$

$$\log p(\mu \mid m, b) = \begin{bmatrix} b \cdot m \\ -\frac{b}{2} \end{bmatrix}^\intercal \begin{bmatrix} \mu \\ \mu^2 \end{bmatrix} + c_2$$

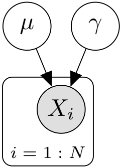
$$\begin{aligned} \log p(x_i \mid \mu, \gamma) &= \begin{bmatrix} \gamma\mu \\ -\frac{\gamma}{2} \end{bmatrix}^\intercal \begin{bmatrix} x_i \\ x_i^2 \end{bmatrix} + c_3 \\ &= \begin{bmatrix} \gamma x_i \\ -\frac{\gamma}{2} \end{bmatrix}^\intercal \begin{bmatrix} \mu \\ \mu^2 \end{bmatrix} + c_4 \\ &= \begin{bmatrix} -\frac{1}{2}x_i^2 + x_i\mu - \frac{1}{2}\mu_i^2 \\ \frac{1}{2} \end{bmatrix}^\intercal \begin{bmatrix} \gamma \\ \log(\gamma) \end{bmatrix} + c_5 \end{aligned}$$

Updating $q(\mu)$

1. Calc. message from co-parent (γ):

$$\mathbf{m}_{\gamma \to x_i} = \begin{bmatrix} \mathbb{E}(\gamma) \\ \mathbb{E}(\log(\gamma)) \end{bmatrix}$$

Model specification



$$\log p(\gamma \mid \alpha, \beta) = \begin{bmatrix} -\beta \\ \alpha - 1 \end{bmatrix}^\mathsf{T} \begin{bmatrix} \gamma \\ \log(\gamma) \end{bmatrix} + c_1$$

$$\log p(\mu \mid m, b) = \begin{bmatrix} b \cdot m \\ -\frac{b}{2} \end{bmatrix}^\mathsf{T} \begin{bmatrix} \mu \\ \mu^2 \end{bmatrix} + c_2$$

$$\begin{aligned} \log p(x_i \mid \mu, \gamma) &= \begin{bmatrix} \gamma\mu \\ -\frac{\gamma}{2} \end{bmatrix}^\mathsf{T} \begin{bmatrix} x_i \\ x_i^2 \end{bmatrix} + c_3 \\ &= \begin{bmatrix} \gamma x_i \\ -\frac{\gamma}{2} \end{bmatrix}^\mathsf{T} \begin{bmatrix} \mu \\ \mu^2 \end{bmatrix} + c_4 \\ &= \begin{bmatrix} -\frac{1}{2}x_i^2 + x_i\mu - \frac{1}{2}\mu_i^2 \\ \frac{1}{2} \end{bmatrix}^\mathsf{T} \begin{bmatrix} \gamma \\ \log(\gamma) \end{bmatrix} + c_5 \end{aligned}$$

Updating $q(\mu)$

1. Calc. message from co-parent (γ):

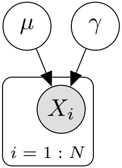
$$\mathbf{m}_{\gamma \to x_i} = \begin{bmatrix} \mathbb{E}(\gamma) \\ \mathbb{E}(\log(\gamma)) \end{bmatrix}$$

2. Send messages from all x_i:

$$\mathbf{m}_{x_i \to \mu} = \begin{bmatrix} \mathbb{E}(\gamma) x_i \\ -\frac{1}{2}\,\mathbb{E}(\gamma) \end{bmatrix}$$

VMP example

Model specification



$$\log p(\gamma \mid \alpha, \beta) = \begin{bmatrix} -\beta \\ \alpha - 1 \end{bmatrix}^\mathsf{T} \begin{bmatrix} \gamma \\ \log(\gamma) \end{bmatrix} + c_1$$

$$\log p(\mu \mid m, b) = \begin{bmatrix} b \cdot m \\ -\frac{b}{2} \end{bmatrix}^\mathsf{T} \begin{bmatrix} \mu \\ \mu^2 \end{bmatrix} + c_2$$

$$\begin{aligned} \log p(x_i \mid \mu, \gamma) &= \begin{bmatrix} \gamma\mu \\ -\frac{\gamma}{2} \end{bmatrix}^\mathsf{T} \begin{bmatrix} x_i \\ x_i^2 \end{bmatrix} + c_3 \\ &= \begin{bmatrix} \gamma x_i \\ -\frac{\gamma}{2} \end{bmatrix}^\mathsf{T} \begin{bmatrix} \mu \\ \mu^2 \end{bmatrix} + c_4 \\ &= \begin{bmatrix} -\frac{1}{2}x_i^2 + x_i\mu - \frac{1}{2}\mu_i^2 \\ \frac{1}{2} \end{bmatrix}^\mathsf{T} \begin{bmatrix} \gamma \\ \log(\gamma) \end{bmatrix} + c_5 \end{aligned}$$

Updating $q(\mu)$

1. Calc. message from co-parent (γ):

$$\mathbf{m}_{\gamma \to x_i} = \begin{bmatrix} \mathbb{E}(\gamma) \\ \mathbb{E}(\log(\gamma)) \end{bmatrix}$$

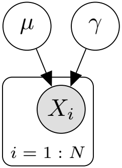
2. Send messages from all x_i:

$$\mathbf{m}_{x_i \to \mu} = \begin{bmatrix} \mathbb{E}(\gamma) x_i \\ -\frac{1}{2}\,\mathbb{E}(\gamma) \end{bmatrix}$$

3. Update natural parameter of $q(\mu)$:

$$\boldsymbol{\eta}_\mu^* = \begin{bmatrix} b \cdot m \\ -\frac{b}{2} \end{bmatrix} + \sum_{i=1}^{N} \mathbf{m}_{x_i \to \mu}$$

Model specification



$$\log p(\gamma \mid \alpha, \beta) = \begin{bmatrix} -\beta \\ \alpha - 1 \end{bmatrix}^\intercal \begin{bmatrix} \gamma \\ \log(\gamma) \end{bmatrix} + c_1$$

$$\log p(\mu \mid m, b) = \begin{bmatrix} b \cdot m \\ -\frac{b}{2} \end{bmatrix}^\intercal \begin{bmatrix} \mu \\ \mu^2 \end{bmatrix} + c_2$$

$$\begin{aligned} \log p(x_i \mid \mu, \gamma) &= \begin{bmatrix} \gamma\mu \\ -\frac{\gamma}{2} \end{bmatrix}^\intercal \begin{bmatrix} x_i \\ x_i^2 \end{bmatrix} + c_3 \\ &= \begin{bmatrix} \gamma x_i \\ -\frac{\gamma}{2} \end{bmatrix}^\intercal \begin{bmatrix} \mu \\ \mu^2 \end{bmatrix} + c_4 \\ &= \begin{bmatrix} -\frac{1}{2}x_i^2 + x_i\mu - \frac{1}{2}\mu_i^2 \\ \frac{1}{2} \end{bmatrix}^\intercal \begin{bmatrix} \gamma \\ \log(\gamma) \end{bmatrix} + c_5 \end{aligned}$$

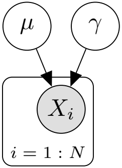
Updating $q(\gamma)$

1. Calc. message from co-parent (μ):

$$\mathbf{m}_{\mu \to x_i} = \begin{bmatrix} \mathbb{E}(\mu) \\ \mathbb{E}(\mu^2) \end{bmatrix}$$

(based on updated parameters.)

Model specification



$$\log p(\gamma \mid \alpha, \beta) = \begin{bmatrix} -\beta \\ \alpha - 1 \end{bmatrix}^{\mathsf{T}} \begin{bmatrix} \gamma \\ \log(\gamma) \end{bmatrix} + c_1$$

$$\log p(\mu \mid m, b) = \begin{bmatrix} b \cdot m \\ -\frac{b}{2} \end{bmatrix}^{\mathsf{T}} \begin{bmatrix} \mu \\ \mu^2 \end{bmatrix} + c_2$$

$$\log p(x_i \mid \mu, \gamma) = \begin{bmatrix} \gamma\mu \\ -\frac{\gamma}{2} \end{bmatrix}^{\mathsf{T}} \begin{bmatrix} x_i \\ x_i^2 \end{bmatrix} + c_3$$

$$= \begin{bmatrix} \gamma x_i \\ -\frac{\gamma}{2} \end{bmatrix}^{\mathsf{T}} \begin{bmatrix} \mu \\ \mu^2 \end{bmatrix} + c_4$$

$$= \begin{bmatrix} -\frac{1}{2}x_i^2 + x_i\mu - \frac{1}{2}\mu_i^2 \\ \frac{1}{2} \end{bmatrix}^{\mathsf{T}} \begin{bmatrix} \gamma \\ \log(\gamma) \end{bmatrix} + c_5$$

Updating $q(\gamma)$

1. Calc. message from co-parent (μ):

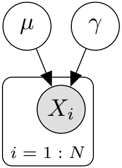
$$\mathbf{m}_{\mu \to x_i} = \begin{bmatrix} \mathbb{E}(\mu) \\ \mathbb{E}(\mu^2) \end{bmatrix}$$

(based on updated parameters.)

2. Send messages from all x_i:

$$\mathbf{m}_{x_i \to \gamma} = \begin{bmatrix} -\frac{1}{2}x_i^2 + x_i\,\mathbb{E}(\mu) - \frac{1}{2}\,\mathbb{E}(\mu^2) \\ \frac{1}{2} \end{bmatrix}$$

Model specification



$$\log p(\gamma \mid \alpha, \beta) = \begin{bmatrix} -\beta \\ \alpha - 1 \end{bmatrix}^\mathsf{T} \begin{bmatrix} \gamma \\ \log(\gamma) \end{bmatrix} + c_1$$

$$\log p(\mu \mid m, b) = \begin{bmatrix} b \cdot m \\ -\frac{b}{2} \end{bmatrix}^\mathsf{T} \begin{bmatrix} \mu \\ \mu^2 \end{bmatrix} + c_2$$

$$\log p(x_i \mid \mu, \gamma) = \begin{bmatrix} \gamma\mu \\ -\frac{\gamma}{2} \end{bmatrix}^\mathsf{T} \begin{bmatrix} x_i \\ x_i^2 \end{bmatrix} + c_3$$

$$= \begin{bmatrix} \gamma x_i \\ -\frac{\gamma}{2} \end{bmatrix}^\mathsf{T} \begin{bmatrix} \mu \\ \mu^2 \end{bmatrix} + c_4$$

$$= \begin{bmatrix} -\frac{1}{2}x_i^2 + x_i\mu - \frac{1}{2}\mu_i^2 \\ \frac{1}{2} \end{bmatrix}^\mathsf{T} \begin{bmatrix} \gamma \\ \log(\gamma) \end{bmatrix} + c_5$$

Updating $q(\gamma)$

1. Calc. message from co-parent (μ):

$$\mathbf{m}_{\mu \to x_i} = \begin{bmatrix} \mathbb{E}(\mu) \\ \mathbb{E}(\mu^2) \end{bmatrix}$$

(based on updated parameters.)

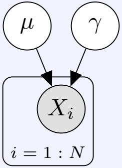
2. Send messages from all x_i:

$$\mathbf{m}_{x_i \to \gamma} = \begin{bmatrix} -\frac{1}{2}x_i^2 + x_i\,\mathbb{E}(\mu) - \frac{1}{2}\,\mathbb{E}(\mu^2) \\ \frac{1}{2} \end{bmatrix}$$

3. Update natural parameter of $q(\gamma)$:

$$\boldsymbol{\eta}_\gamma^* = \begin{bmatrix} -\beta \\ \alpha - 1 \end{bmatrix} + \sum_{i=1}^{N} \mathbf{m}_{x_i \to \gamma}$$

Code task: Implement VMP in the model defined on the previous slide

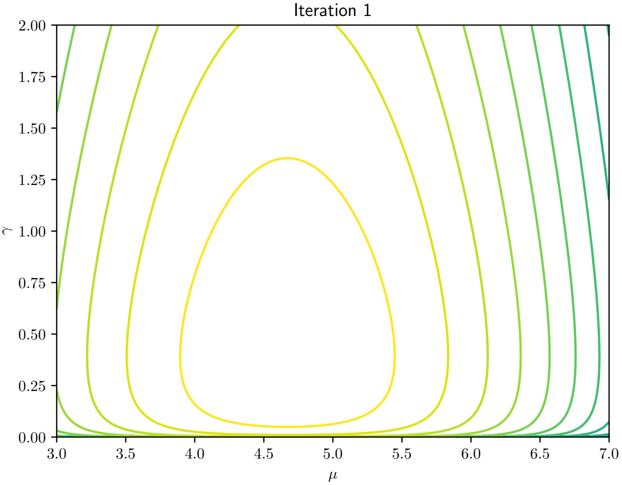


- We will implement VMP in this model, writing source-code that heavily relies on the specific model at hand:
  - Only Gaussian and Gamma distributed variables.
  - A Gaussian can only have another Gaussian as child ($\mu \rightarrow x_i$).
  - The parent of a Gaussian can be Gaussian ($\mu \rightarrow x_i$) or Gamma distributed ($\gamma \rightarrow x_i$) .
- We have already looked at translations from moment parameters to natural parameters. Now we need the translation also going the other way.
- Most of the "supporting code" is already made available to you; start from `students_VMP.ipynb`.

Data

Four data points sampled from a normal distribution with mean 5 and variance 1.

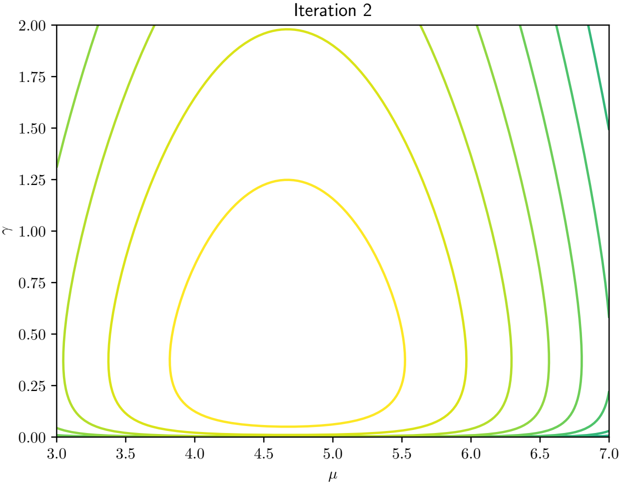
Posteriors



Data

Four data points sampled from a normal distribution with mean 5 and variance 1.

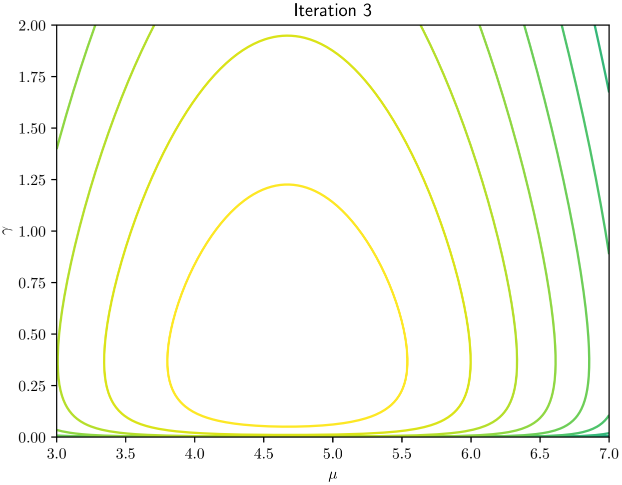
Posteriors



Data

Four data points sampled from a normal distribution with mean 5 and variance 1.

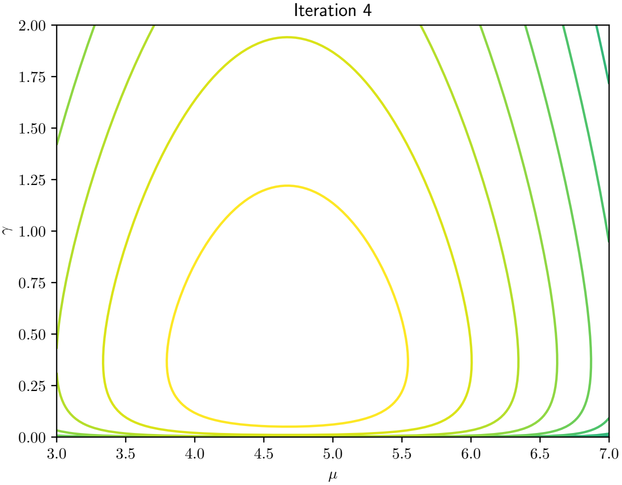
Posteriors



Data

Four data points sampled from a normal distribution with mean 5 and variance 1.

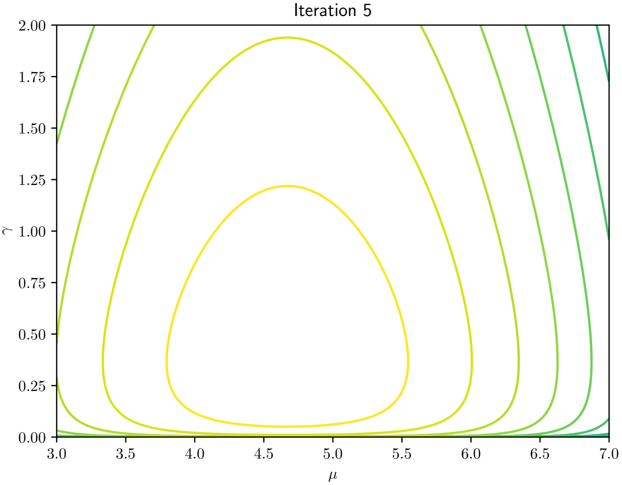
Posteriors



Data

Four data points sampled from a normal distribution with mean 5 and variance 1.

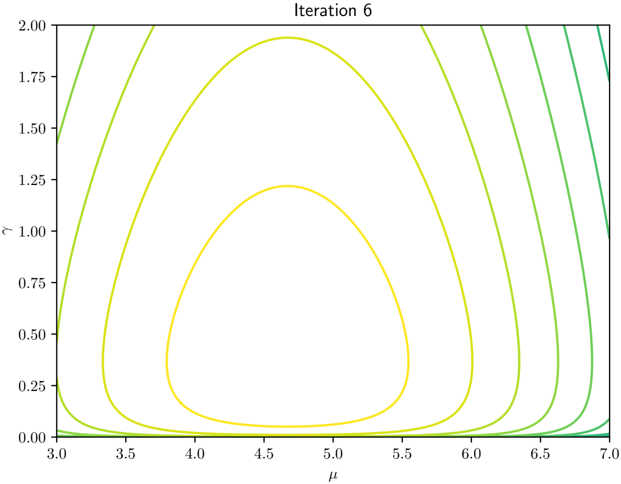
Posteriors



Data

Four data points sampled from a normal distribution with mean 5 and variance 1.

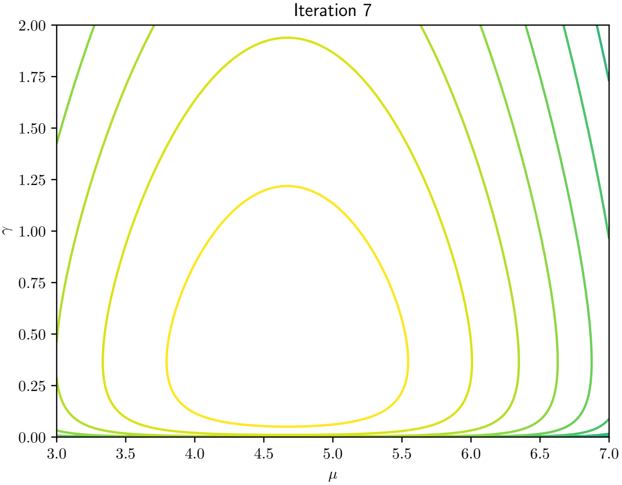
Posteriors



Data

Four data points sampled from a normal distribution with mean 5 and variance 1.

Posteriors

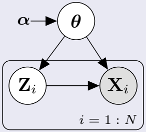


Stochastic Variational Inference

Setup

Model of interest

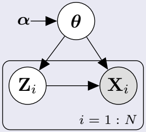
- $\boldsymbol{\theta}$ are *global* hidden variables, using $\boldsymbol{\alpha}$ as (hyper-)parameters.
- $\mathbf{Z}_i$ is a vector of latent variables *local* to $\mathbf{X}_i$
  - $\mathbf{Z}_i$ describes the internal structure of $\mathbf{X}_i$ (like in a factor analysis model).
  - Notice that $\{\mathbf{X}_i, \mathbf{Z}_i\} \perp\!\!\!\perp \{\mathbf{X}_j, \mathbf{Z}_j\} \,|\, \boldsymbol{\theta}$ for $i \neq j$.
- All distributions belong to the (conjugate) Exponential Family.
- $\mathbf{X}_i$ is observed, hence we have a data-set $\mathcal{D} = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$.



Setup

Model of interest

- $\boldsymbol{\theta}$ are *global* hidden variables, using $\boldsymbol{\alpha}$ as (hyper-)parameters.
- $\mathbf{Z}_i$ is a vector of latent variables *local* to $\mathbf{X}_i$
  - $\mathbf{Z}_i$ describes the internal structure of $\mathbf{X}_i$ (like in a factor analysis model).
  - Notice that $\{\mathbf{X}_i, \mathbf{Z}_i\} \perp\!\!\!\perp \{\mathbf{X}_j, \mathbf{Z}_j\} \,|\, \boldsymbol{\theta}$ for $i \neq j$.
- All distributions belong to the (conjugate) Exponential Family.
- $\mathbf{X}_i$ is observed, hence we have a data-set $\mathcal{D} = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$.



Example of use:

- $\boldsymbol{\theta}$ represents *topics* for text document and which words are used for each topic.
- $\mathbf{X}_i$ is a text document represented by a bag-of-words.
- $\mathbf{Z}_i$ encodes which topics $\mathbf{X}_i$ discusses.

Goals:

- Infer the local latent representation Z_i for each observation $\mathbf{x}_i$
- Infer the global representation $\boldsymbol{\theta}$. **Typically this is the most important goal.**

VMP for this model

Algorithm

1. Initialize all variational parameters randomly.
2. Repeat
   (a) For each local variational parameter-vector $\boldsymbol{\eta}_{\mathbf{z}_j}$:
   $$\boldsymbol{\eta}_{\mathbf{z}_j} \leftarrow \tilde{\boldsymbol{\eta}}_{z_j}(\mathbf{m}_{\boldsymbol{\theta}\rightarrow\mathbf{z}_j}) + \mathbf{m}_{x_i\rightarrow z_i}.$$
   (b) Update the variational parameters for global parameter $\boldsymbol{\theta}$:
   $$\boldsymbol{\eta}_{\boldsymbol{\theta}} \leftarrow \boldsymbol{\eta}_{\boldsymbol{\theta}}(\boldsymbol{\alpha}) + \sum_{i=1}^{N}(\mathbf{m}_{\mathbf{z}_i\rightarrow\boldsymbol{\theta}} + \mathbf{m}_{\mathbf{x}_i\rightarrow\boldsymbol{\theta}}).$$
3. Until we converge wrt. $\mathcal{L}(q)$.

Computational problem:

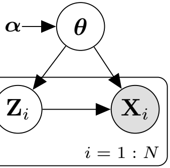
- In the first iteration, each $\boldsymbol{\eta}_{\mathbf{z}_j}$ is using the random initialization of $\boldsymbol{\eta}_{\boldsymbol{\theta}}$.
  - This may jeopardize the results of $\boldsymbol{\eta}_{\mathbf{z}_j}$, and this waste computation.
  - We do N local updates using $\boldsymbol{\eta}_{\boldsymbol{\theta}}$.
  - If N is large this can be a considerable loss of computational time.
- In turn, $\boldsymbol{\eta}_{\boldsymbol{\theta}}$ will be updated based on poorly adjusted $\boldsymbol{\eta}_{\mathbf{z}_i}$ values, and the whole process converges slowly.

The VMP message received at θ

Recall the VMP architecture

$\boldsymbol{\eta}_{\boldsymbol{\theta}}$ is updates according to

$$\boldsymbol{\eta}_{\boldsymbol{\theta}}(\boldsymbol{\alpha}) + \sum_{i=1}^{N} (\mathbf{m}_{\mathbf{z}_i \to \boldsymbol{\theta}} + \mathbf{m}_{\mathbf{x}_i \to \boldsymbol{\theta}})$$



Opportunity for speed-up through parallelization

- If we distribute the dataset $\mathcal{D} = \{\mathbf{x}_i, \ldots, \mathbf{x}_N\}$ on several computational nodes, the node using data-partition $\mathcal{D}_j$ will calculate

$$\sum_{i:\mathbf{x}_i \in \mathcal{D}_j} (\mathbf{m}_{\mathbf{z}_i \to \boldsymbol{\theta}} + \mathbf{m}_{\mathbf{x}_i \to \boldsymbol{\theta}})$$

- In a map-reduce organization, the master-node sums the messages from the slaves, updates $\boldsymbol{\eta}_{\boldsymbol{\theta}}$, and distributes the $\boldsymbol{\theta}$-message back to the slaves.
- Each slave updates its local latent variables $\{\boldsymbol{\eta}_{\mathbf{Z}_i}\}_{i:\mathbf{x}_i \in \mathcal{D}_j}$.

"Crazy" idea: Subsampling:

Instead of distributing the dataset we just **subsample** a dataset ("minibatch") from $\mathcal{D}$, say a single observation $\mathbf{x}_i$ and use that subsample to update $\boldsymbol{\lambda}_{\boldsymbol{\theta}}$:

1. Initialize all variational parameters randomly to $\boldsymbol{\lambda}$.
2. Repeat forever
   (a) Select one $\mathbf{x}_i$ randomly from $\mathcal{D}$.
   (b) Update $\boldsymbol{\eta}_{\mathbf{z}_i}$:
   $$\boldsymbol{\eta}_{\mathbf{z}_i} \leftarrow \tilde{\boldsymbol{\eta}}_{z_i}(\mathbf{m}_{\boldsymbol{\theta}\rightarrow\mathbf{z}_i}) + \mathbf{m}_{\mathbf{x}_i\rightarrow\mathbf{z}_i}$$
   (c) Update the variational parameters for $\boldsymbol{\theta}$:
   $$\boldsymbol{\eta}_{\boldsymbol{\theta}} \leftarrow \boldsymbol{\eta}_{\boldsymbol{\theta}}(\boldsymbol{\alpha}) + \mathbf{m}_{\mathbf{z}_i\rightarrow\boldsymbol{\theta}} + \mathbf{m}_{\mathbf{x}_i\rightarrow\boldsymbol{\theta}}$$

"Crazy" idea: Subsampling:

Instead of distributing the dataset we just **subsample** a dataset ("minibatch") from $\mathcal{D}$, say a single observation $\mathbf{x}_i$ and use that subsample to update $\boldsymbol{\lambda}_{\boldsymbol{\theta}}$:

1. Initialize all variational parameters randomly to $\boldsymbol{\lambda}$.
2. Repeat forever
   (a) Select one $\mathbf{x}_i$ randomly from $\mathcal{D}$.
   (b) Update $\boldsymbol{\eta}_{\mathbf{z}_i}$:
   $$\boldsymbol{\eta}_{\mathbf{z}_i} \leftarrow \tilde{\boldsymbol{\eta}}_{z_i}(\mathbf{m}_{\boldsymbol{\theta}\rightarrow\mathbf{z}_i}) + \mathbf{m}_{\mathbf{x}_i\rightarrow\mathbf{z}_i}$$
   (c) Update the variational parameters for $\boldsymbol{\theta}$:
   $$\boldsymbol{\eta}_{\boldsymbol{\theta}} \leftarrow \boldsymbol{\eta}_{\boldsymbol{\theta}}(\boldsymbol{\alpha}) + \mathbf{m}_{\mathbf{z}_i\rightarrow\boldsymbol{\theta}} + \mathbf{m}_{\mathbf{x}_i\rightarrow\boldsymbol{\theta}}$$

Bad news – and some good:

Bad news: This does not work!
However, the **good news** is that we can fix it!

A small side-step: Gradient Ascent

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

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

$$\boldsymbol{\lambda}^{(t+1)} \leftarrow \boldsymbol{\lambda}^{(t)} + \rho \cdot \nabla_{\boldsymbol{\lambda}} f\left(\boldsymbol{\lambda}^{(t)}\right)$$

$\boldsymbol{\lambda}^{(t)}$ converges to a (local) optimum of $f(\cdot)$ if:

- f is “sufficiently nice”;
- The learning-rate ρ is “sufficiently small”.

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

If we have access to $\mathbf{g}(\boldsymbol{\lambda})$ – an **unbiased estimate** of the gradient – it still works!

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

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

$\boldsymbol{\lambda}_t$ converges to a (local) optimum of $f(\cdot)$ if:

- f is “sufficiently nice”;
- $\mathbf{g}(\boldsymbol{\lambda})$ is a random variable with $\mathbb{E}[\mathbf{g}(\boldsymbol{\lambda})] = \nabla_{\boldsymbol{\lambda}} f(\boldsymbol{\lambda})$ and finite variance.
- The learning-rates $\{\rho_t\}$ is a Robbins-Monro – sequence:
    - $\sum_t \rho_t = \infty$
    - $\sum_t \rho_t^2 < \infty$

The (Euclidian) gradient points in the direction of the solution of

$$\arg\max_{\mathrm{d}\boldsymbol{\lambda}} f(\boldsymbol{\lambda} + \mathrm{d}\boldsymbol{\lambda}) \qquad \text{subject to } ||\mathrm{d}\boldsymbol{\lambda}||_2 < \epsilon$$

- This, however, fails to recognize that we work with probability distributions:
  - The distributions $\mathcal{N}(\mu = 0, \tau = 10^{-6})$ and $\mathcal{N}(\mu = 10, \tau = 10^{-6})$ are "close" as both are virtually uniform on $\mathbb{R}$, but have distance 10 parameter space.
  - $\mathcal{N}(\mu = 0, \tau = 10^{6})$ and $\mathcal{N}(\mu = 1, \tau = 10^{6})$ are "separated", even though their distance in $\boldsymbol{\lambda}$-space is only 1.

Natural gradients

The (Euclidian) gradient points in the direction of the solution of

$$\arg\max_{\mathrm{d}\boldsymbol{\lambda}} f(\boldsymbol{\lambda} + \mathrm{d}\boldsymbol{\lambda}) \qquad \text{subject to } ||\mathrm{d}\boldsymbol{\lambda}||_2 < \epsilon$$

- This, however, fails to recognize that we work with probability distributions:
  - The distributions $\mathcal{N}(\mu = 0, \tau = 10^{-6})$ and $\mathcal{N}(\mu = 10, \tau = 10^{-6})$ are "close" as both are virtually uniform on $\mathbb{R}$, but have distance 10 parameter space.
  - $\mathcal{N}(\mu = 0, \tau = 10^{6})$ and $\mathcal{N}(\mu = 1, \tau = 10^{6})$ are "separated", even though their distance in $\boldsymbol{\lambda}$-space is only 1.
- **Natural gradients** take the information geometry into account.
- The natural gradients are found by pre-multiplying with the inverse Fisher matrix:

$$\tilde{\nabla}_{\boldsymbol{\lambda}} f(\boldsymbol{\lambda}) = \mathbf{H}_{\boldsymbol{\lambda}}^{-1} \nabla_{\boldsymbol{\lambda}} f(\boldsymbol{\lambda})$$

where $\mathbf{H}_{\boldsymbol{\lambda}}$ is defined as $\mathbf{H}_{\boldsymbol{\lambda}} = -\,\mathbb{E}_X\left[\nabla_{\boldsymbol{\lambda}}^2 \log f(X \mid \boldsymbol{\lambda}) \mid \boldsymbol{\lambda}\right]$.

Natural gradients

The (Euclidian) gradient points in the direction of the solution of

$$\arg\max_{\mathrm{d}\boldsymbol{\lambda}} f(\boldsymbol{\lambda} + \mathrm{d}\boldsymbol{\lambda}) \qquad \text{subject to } ||\mathrm{d}\boldsymbol{\lambda}||_2 < \epsilon$$

- This, however, fails to recognize that we work with probability distributions:
  - The distributions $\mathcal{N}(\mu = 0, \tau = 10^{-6})$ and $\mathcal{N}(\mu = 10, \tau = 10^{-6})$ are "close" as both are virtually uniform on $\mathbb{R}$, but have distance 10 parameter space.
  - $\mathcal{N}(\mu = 0, \tau = 10^{6})$ and $\mathcal{N}(\mu = 1, \tau = 10^{6})$ are "separated", even though their distance in $\boldsymbol{\lambda}$-space is only 1.
- **Natural gradients** take the information geometry into account.
- The natural gradients are found by pre-multiplying with the inverse Fisher matrix:

$$\tilde{\nabla}_{\boldsymbol{\lambda}} f(\boldsymbol{\lambda}) = \mathbf{H}_{\boldsymbol{\lambda}}^{-1} \nabla_{\boldsymbol{\lambda}} f(\boldsymbol{\lambda})$$

  where $\mathbf{H}_{\boldsymbol{\lambda}}$ is defined as $\mathbf{H}_{\boldsymbol{\lambda}} = -\,\mathbb{E}_X\left[\nabla^2_{\boldsymbol{\lambda}} \log f(X \,|\, \boldsymbol{\lambda}) \,|\, \boldsymbol{\lambda}\right]$.
- The same operation can obviously also be done in a sub-sample setting, with the same $\mathbf{H}_{\boldsymbol{\lambda}}$ – since $\mathbf{H}_{\boldsymbol{\lambda}}$ is data-independent.

Code Task: Maximum Likelihood in a simple Gaussian model

We are looking for the ML estimators for them mean μ and precision τ in a simple Gaussian model.

Recall that

$$f(\mu, \tau) = \sum_{i=1}^{N} \log p(x_i \mid \mu, \tau) = -\frac{N}{2}\log(2\pi) + \frac{N}{2}\log\tau - \frac{\tau}{2}\sum_{i=1}^{N}(x_i - \mu)^2$$

This should give us easy access to both $\nabla_{(\mu,\tau)} f(\mu, \tau)$ as well as the Fisher information matrix $-\mathbb{E}_X\left[\nabla^2_{(\mu,\tau)} \log p(X \mid \mu, \tau) \mid \mu, \tau\right]$.

You are asked to fix `calculate_gradient` in the file
`students_ML_via_SGD.ipynb`.

Examine the effect of

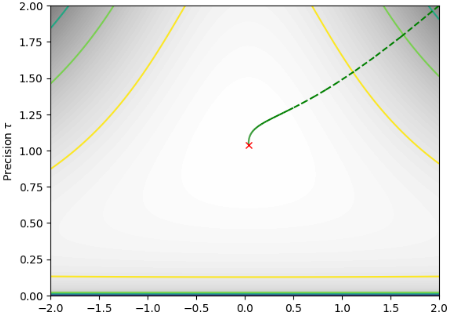
- Changing the batch-size (all data or just a single observation per step)
- Switching between natural and Euclidian gradients.

Example: Maximum log likelihood in a Gaussian model

We have access to $N = 1000$ observations from a Gaussian distribution with unknown mean μ and precision τ. Use $\boldsymbol{\lambda} = [\mu, \tau]^\intercal$.

$$\begin{aligned} f(\boldsymbol{\lambda}) &= \sum_{i=1}^{N} \log p(x_i \,|\, \boldsymbol{\lambda}) = \frac{N}{2}\log\tau - \frac{N}{2}\log(2\pi) - \frac{\tau}{2}\sum_{i=1}^{N}(x_i-\mu)^2 \\ \nabla_{\boldsymbol{\lambda}} &= \begin{bmatrix} -N\tau\mu + \tau\sum_{i=1}^{N} x_i \\ \frac{N}{2\tau} - \frac{1}{2}\sum_{i=1}^{N}(x_i-\mu)^2 \end{bmatrix} \end{aligned}$$

Cost of calculation: $O(N)$



Example: Maximum log likelihood in a Gaussian model

We consider the same maximum likelihood problem, but instead of the gradient based on the full sample, we only have a **mini-batch of a single example** x_t at iteration t:

$$\mathbf{g}(\boldsymbol{\lambda} \,|\, x_t) = N \cdot \begin{bmatrix} -\tau\mu + \tau x_t \\ \frac{1}{2\tau} - \frac{1}{2}\left(x_t - \mu\right)^2 \end{bmatrix}$$

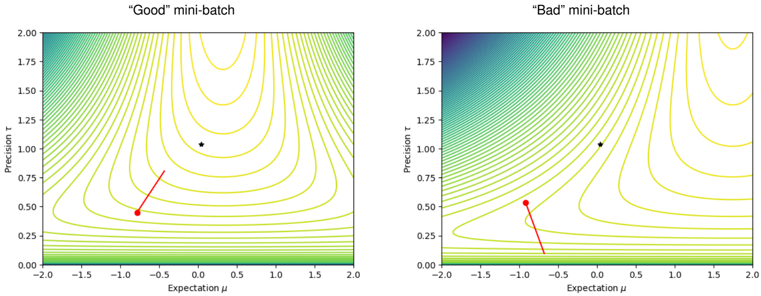
Randomness in $\mathbf{g}$ is a consequence of the random data selection process, and it follows that $\mathbb{E}[\mathbf{g}(\boldsymbol{\lambda})] = \nabla_{\boldsymbol{\lambda}} f(\boldsymbol{\lambda})$ because we re-scaled through a multiplication of N.

Example: Maximum log likelihood in a Gaussian model

We consider the same maximum likelihood problem, but instead of the gradient based on the full sample, we only have a **mini-batch of a single example** x_t at iteration t:

$$\mathbf{g}(\boldsymbol{\lambda} \,|\, x_t) = N \cdot \begin{bmatrix} -\tau\mu + \tau x_t \\ \frac{1}{2\tau} - \frac{1}{2}\left(x_t - \mu\right)^2 \end{bmatrix}$$

Randomness in $\mathbf{g}$ is a consequence of the random data selection process, and it follows that $\mathbb{E}[\mathbf{g}(\boldsymbol{\lambda})] = \nabla_{\boldsymbol{\lambda}} f(\boldsymbol{\lambda})$ because we re-scaled through a multiplication of N.

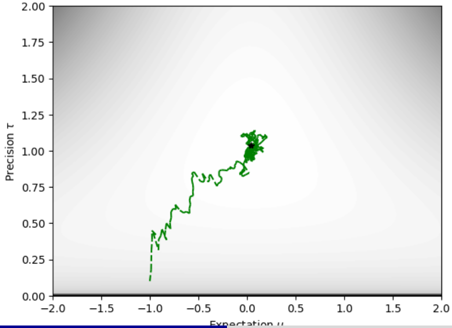


Example: Maximum log likelihood in a Gaussian model

We consider the same maximum likelihood problem, but instead of the gradient based on the full sample, we only have a **mini-batch of a single example** x_t at iteration t:

$$\mathbf{g}(\boldsymbol{\lambda} \,|\, x_t) = N \cdot \begin{bmatrix} -\tau\mu + \tau x_t \\ \frac{1}{2\tau} - \frac{1}{2}\left(x_t - \mu\right)^2 \end{bmatrix}$$

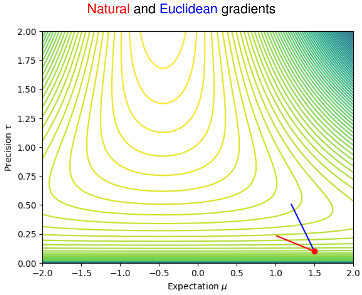
Randomness in $\mathbf{g}$ is a consequence of the random data selection process, and it follows that $\mathbb{E}[\mathbf{g}(\boldsymbol{\lambda})] = \nabla_{\boldsymbol{\lambda}} f(\boldsymbol{\lambda})$ because we re-scaled through a multiplication of N.

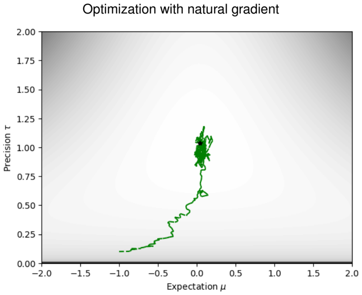


Example: Maximum log likelihood in a Gaussian model

We consider the same maximum likelihood problem. The Fisher information matrix is given by

$$\begin{aligned}\mathbf{H}_{\boldsymbol{\lambda}} &= -\mathbb{E}_X\left[\nabla^2_{\boldsymbol{\lambda}} \log p(X \mid \boldsymbol{\lambda}) \mid \boldsymbol{\lambda}\right] \\ &= -\mathbb{E}_X\left(\begin{bmatrix} -\tau & X-\mu \\ X-\mu & -\frac{1}{2\tau^2}\end{bmatrix}\right) = \begin{bmatrix} \tau & 0 \\ 0 & \frac{1}{2\tau^2}\end{bmatrix}\end{aligned}$$





The Stochastic Variational Inference (SVI) algorithm

1. Initialize $\boldsymbol{\lambda}^{(0)}$; Set $t = 1$; Let $\{\rho_t\}_{t=1}^{\infty}$ be a Robbins-Moreno – sequence.
2. Repeat forever
   (a) Select a single observation $\mathbf{x}_i$ from $\mathcal{D}$.
   (b) Compute its local variational parameter-vector $\boldsymbol{\eta}_i$:
   $$\boldsymbol{\eta}_{\mathbf{z}_j} \leftarrow \tilde{\boldsymbol{\eta}}_{z_j}(\mathbf{m}_{\boldsymbol{\theta}\to\mathbf{z}_j}) + \mathbf{m}_{\mathbf{x}_i\to\mathbf{z}_i}$$
   (c) Update the variational parameters for $\boldsymbol{\theta}$:
   $$\boldsymbol{\lambda}_{\boldsymbol{\theta}}^{(t)} \leftarrow (1-\rho_t)\boldsymbol{\lambda}_{\boldsymbol{\theta}}^{(t-1)} + \rho_t\left(\boldsymbol{\eta}_{\boldsymbol{\theta}}(\boldsymbol{\alpha}) + N\cdot(\mathbf{m}_{\mathbf{z}_i\to\boldsymbol{\theta}} + \mathbf{m}_{\mathbf{x}_i\to\boldsymbol{\theta}})\right)$$
   (d) $t \leftarrow t+1$

- The algorithm holds **theoretical guarantees** of convergence – it can be seen as a **stochastic gradient ascent** algorithm over $\mathcal{L}(q)$.
- We work in natural parameter space – this **simplifies the calculations** considerably, and **improves performance**.
- SVI offers **substantive improvements** over VMP on massive data-sets – $\boldsymbol{\lambda}_{\boldsymbol{\theta}}$ can often converge before all data is read once.

The last exponential toppings

Some nice features

Remember the definition or the Exponential Family:

$$f_X(x \,|\, \boldsymbol{\eta}) = \exp\left(h(\mathbf{x}) + \boldsymbol{\eta}^\intercal \mathbf{t}(\mathbf{x}) - A(\boldsymbol{\eta})\right)$$

Maximum likelihood estimator for $\boldsymbol{\eta}$ from a dataset $\mathcal{D} = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$:

- First notice that

$$\log f(\mathcal{D} \,|\, \boldsymbol{\eta}) = \sum_i h(\mathbf{x}_i) + \boldsymbol{\eta}^\intercal \sum_i \mathbf{t}(\mathbf{x}_i) - N \cdot A(\boldsymbol{\eta}).$$

- We thus look for

$$\boldsymbol{\eta}^* = \arg\max_{\boldsymbol{\eta}} \sum_i \boldsymbol{\eta}^\intercal \mathbf{t}(\mathbf{x}_i) - N \cdot A(\boldsymbol{\eta})$$

- Since $\frac{\mathrm{d}A(\boldsymbol{\eta})}{\mathrm{d}\boldsymbol{\eta}} = \mathbb{E}\left[\mathbf{t}\left(\mathbf{X}\right)\right]$, it follows that $\boldsymbol{\eta}^*$ ensures *moment matching* for $\mathbf{t}(\mathbf{X})$:

$$\mathbb{E}\left[\mathbf{t}\left(\mathbf{X}\right)\right] = \frac{1}{N} \sum_i \mathbf{t}(\mathbf{x}_i)$$

- … and convexity of $A(\boldsymbol{\eta})$ ensures that this is the unique global optimum.

Some nice features

Remember the definition or the Exponential Family:

$$f_X(x \,|\, \boldsymbol{\eta}) = \exp\left(h(\mathbf{x}) + \boldsymbol{\eta}^\intercal \mathbf{t}(\mathbf{x}) - A(\boldsymbol{\eta})\right)$$

Maximum likelihood estimator for $\boldsymbol{\eta}$ from a dataset $\mathcal{D} = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$:

- Example: The multivariate Gaussian has $t(X) = [\mathbf{X}, \mathbf{X}\mathbf{X}^\intercal]$.
- Remember that $\mathbb{E}\left[\mathbf{t}\left(\mathbf{X}\right)\right] = \left[\mathbb{E}(\mathbf{X}), \mathbb{E}(\mathbf{X}\mathbf{X}^\intercal)\right] = \left[-\frac{1}{2}\boldsymbol{\eta}_2^{-1}\boldsymbol{\eta}_1, -\frac{1}{2}\boldsymbol{\eta}_2^{-1}\right]$.
- This gives us

$$\boldsymbol{\eta}_1^* = \left(\sum_i \mathbf{x}_i\right) \cdot \left(\sum_i \mathbf{x}_i \mathbf{x}_i^\intercal\right)^{-1} \text{ and } \boldsymbol{\eta}_2^* = -\frac{N}{2}\left(\sum_i \mathbf{x}_i \mathbf{x}_i^\intercal\right)^{-1}.$$

- While harder to interpret than the *moment parameter* MLEs,

$$\boldsymbol{\mu}^* = \frac{1}{N}\sum_i \mathbf{x}_i \text{ and } \boldsymbol{\Sigma}^* = \frac{1}{N}\sum_i \mathbf{x}_i \mathbf{x}_i^\intercal,$$

the *natural parameter* MLEs are found with a unifying theory.

Conjugacy plays a crucial role in Bayesian inference:

- Assume we observe variables $\mathcal{D} = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$ from the model $f(\mathbf{x} \,|\, \boldsymbol{\eta})$.
- Prior knowledge about $\boldsymbol{\eta}$ is encoded in $f(\boldsymbol{\eta} \,|\, \boldsymbol{\nu})$, we seek the posterior $f(\boldsymbol{\eta} \,|\, \mathcal{D}, \nu) \propto f(\boldsymbol{\eta} \,|\, \boldsymbol{\nu}) \prod_i f(\mathbf{x}_i \,|\, \boldsymbol{\eta})$.
- $f(\boldsymbol{\eta} \,|\, \boldsymbol{\nu})$ is a conjugate for $f(\mathbf{x} \,|\, \boldsymbol{\eta})$ if $f(\boldsymbol{\eta} \,|\, \mathcal{D}, \boldsymbol{\nu})$ has the same form as $f(\boldsymbol{\eta} \,|\, \boldsymbol{\nu})$ when both are seen as functions of $\boldsymbol{\eta}$.
- Then inference amounts to “updating” $\boldsymbol{\nu}$ to a new value, defined by $\boldsymbol{\nu}$ and $\mathcal{D}$.

Examples:

- Prior Multinomial + Multinomial likelihood .
- Prior Beta(p) + Likelihood Bernoulli($x \,|\, p$) – also for Binomial for fixed n.
- Prior Normal(μ) + Likelihood Normal($x \,|\, \mu$) with mean μ and known variance.
- Prior Gamma(τ) + Likelihood Normal($x \,|\, \tau$) with known mean and variance τ^{-1}.

Conjugacy plays a crucial role in Bayesian inference:

- Assume we observe variables $\mathcal{D} = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$ from the model $f(\mathbf{x} \,|\, \boldsymbol{\eta})$.
- Prior knowledge about $\boldsymbol{\eta}$ is encoded in $f(\boldsymbol{\eta} \,|\, \boldsymbol{\nu})$, we seek the posterior $f(\boldsymbol{\eta} \,|\, \mathcal{D}, \nu) \propto f(\boldsymbol{\eta} \,|\, \boldsymbol{\nu}) \prod_i f(\mathbf{x}_i \,|\, \boldsymbol{\eta})$.
- $f(\boldsymbol{\eta} \,|\, \boldsymbol{\nu})$ is a conjugate for $f(\mathbf{x} \,|\, \boldsymbol{\eta})$ if $f(\boldsymbol{\eta} \,|\, \mathcal{D}, \boldsymbol{\nu})$ has the same form as $f(\boldsymbol{\eta} \,|\, \boldsymbol{\nu})$ when both are seen as functions of $\boldsymbol{\eta}$.
- Then inference amounts to “updating” $\boldsymbol{\nu}$ to a new value, defined by $\boldsymbol{\nu}$ and $\mathcal{D}$.

The Exponential Family:

- Assume a likelihood-model $f_x(\mathbf{x} \,|\, \boldsymbol{\eta}) = \exp\left(h(\mathbf{x}) + \boldsymbol{\eta}^\intercal \mathbf{t}(\mathbf{x}) - A_x(\boldsymbol{\eta})\right)$.
- Then the conjugate prior is $f_\eta(\boldsymbol{\eta} \,|\, \boldsymbol{\tau}, \nu) = \exp\left(h_\eta(\boldsymbol{\eta}) + \boldsymbol{\tau}^\intercal \boldsymbol{\eta} - \nu A_x(\boldsymbol{\eta}) - A_\eta(\boldsymbol{\tau})\right)$.
- Notice how the prior is not in ExpFam form; can be done by setting

$$\tilde{\boldsymbol{\tau}} = [\boldsymbol{\tau}^\intercal, -\nu]^\intercal, \ \tilde{\mathbf{t}}_\eta(\boldsymbol{\eta}) = [\boldsymbol{\eta}^\intercal, A_x(\boldsymbol{\eta})]^\intercal, \ \tilde{A}_\eta(\tilde{\boldsymbol{\tau}}) = A_\eta(\boldsymbol{\tau}).$$

from which we obtain $f_\eta(\boldsymbol{\eta} \,|\, \tilde{\boldsymbol{\tau}}) = \exp\left(h_\eta(\boldsymbol{\eta}) + \tilde{\boldsymbol{\tau}}^\intercal \tilde{\mathbf{t}}_\eta(\boldsymbol{\eta}) - \tilde{A}_\eta(\tilde{\boldsymbol{\tau}})\right)$.

$\rightsquigarrow$ specific form can be exploited during, e.g., SVI for calculating the gradients.

Model formulation

Assume a likelihood-model $f_x(\mathbf{x} \mid \boldsymbol{\eta}) = \exp\left(h_x(\mathbf{x}) + \boldsymbol{\eta}^\intercal \mathbf{t}(\mathbf{x}) - A_x(\boldsymbol{\eta})\right)$, from which we have observed the dataset $\mathcal{D} = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$.

The prior is $f_\eta(\boldsymbol{\eta} \mid \boldsymbol{\tau}, \nu) = \exp\left(h_\eta(\boldsymbol{\eta}) + \boldsymbol{\tau}^\intercal \boldsymbol{\eta} - \nu A_x(\boldsymbol{\eta}) - A_\eta(\boldsymbol{\tau})\right)$.

$$
\begin{aligned}
\log f_\eta(\boldsymbol{\eta} \mid \mathcal{D}, \boldsymbol{\tau}, \nu) &\propto \log f_x(\mathcal{D} \mid \boldsymbol{\eta}) + \log f_\eta(\boldsymbol{\eta} \mid \boldsymbol{\tau}, \nu) \\
&= \sum_i h_x(\mathbf{x}_i) + \boldsymbol{\eta}^\intercal \sum_i \mathbf{t}(\mathbf{x}_i) - N \cdot A_x(\boldsymbol{\eta}) \\
&\quad + \quad h_\eta(\boldsymbol{\eta}) + \boldsymbol{\tau}^\intercal \boldsymbol{\eta} - \nu \cdot A_x(\boldsymbol{\eta}) - A_\eta(\boldsymbol{\tau}) \\
&\propto h_\eta(\boldsymbol{\eta}) + \boldsymbol{\eta}^\intercal \left\{\boldsymbol{\tau} + \sum_i \mathbf{t}(\mathbf{x}_i)\right\} - \{N + \nu\} \cdot A_x(\boldsymbol{\eta}) - A_\eta(\boldsymbol{\tau}) \\
\log f_\eta(\boldsymbol{\eta} \mid \mathcal{D}, \boldsymbol{\tau}, \nu) &= \log f_x\left(\boldsymbol{\eta} \mid \boldsymbol{\tau} + \sum_i \mathbf{t}(\mathbf{x}_i), \nu + N\right)
\end{aligned}
$$

Interpretation:

Posterior is found directly by collecting the summed sufficient statistics and no. observations. This **always** works for **every** conjugate exponential family.