

BML lecture #3: variational Bayes

<http://github.com/rbardenet/bml-course>

Rémi Bardenet

CNRS & CRIS^tAL, Univ. Lille, France



1 Introduction

2 Variational inference

3 Back to LDA

1 Introduction

2 Variational inference

3 Back to LDA

What comes to *your* mind when you hear "variational inference"?

1 Introduction

2 Variational inference

3 Back to LDA

Turning integration into optimization over measures

Variational Bayesian inference (VB) consists in approximating

$$\int f(\theta)\pi(\theta)d\theta \approx \int f(\theta)q^*(\theta)d\theta$$

with $q^* \in \arg \min_{q \in \mathcal{Q}} \text{distance}(\pi, q)$. Often we take

$$\text{distance}(\pi, \tilde{q}) = \text{KL}(\tilde{q}, \pi) := \int q(\theta) \log \frac{q(\theta)}{\pi(\theta)} d\theta.$$

for computational convenience.

But remember we can only evaluate $\pi_u = Z\pi\ldots$

► Show that $J(q) := \int q(\theta) \log \frac{q(\theta)}{\pi_u(\theta)} d\theta = \text{KL}(q, \pi) - \log Z$.

► In particular, $L(q) = -J(q) \leq \log Z$. For

$$\pi_u(\theta) = p(\text{data}|\theta)p(\theta),$$

$L(q)$ is thus a lower bound for the evidence $p(\text{data})$ (ELBO).

- ▶ The most common approach is the mean-field approximation

$$\mathcal{Q} = \{\theta \mapsto \prod_{d=1}^D q_d(\theta_d)\}.$$

- ▶ Include all variables over which you integrate, e.g.

$$q(\theta, z_{1:n}) = \prod_{d=1}^D q_d(\theta_d) \prod_{i=1}^N q_i(z_i).$$

- ▶ Try to keep some dependence if it is key in your application.
- ▶ If your original model has NEF conditionals, **coordinate-wise maximization of $q \mapsto L(q)$ is easy.**

1 Introduction

2 Variational inference

3 Back to LDA

$$\begin{aligned}
 & \log p(y, z, \pi, B) \\
 &= \sum_{i=1}^N \left[\log p(\pi_i | \alpha) + \sum_{\ell=1}^{L_i} \left(\log p(z_{i\ell} | \pi_i) + \log p(y_{i\ell} | z_{i\ell}, B) \right) \right] + p(B | \gamma) \\
 &\propto \sum_{i=1}^N \left[\sum_{k=1}^K \alpha_k \log \pi_{ik} + \sum_{\ell=1}^{L_i} \left(\sum_{k=1}^K 1_{z_{i\ell}=k} \log \pi_{ik} + \sum_{v=1}^V \sum_{k=1}^K 1_{y_{i\ell}=v} 1_{z_{i\ell}=k} \log b_{kv} \right) \right] \\
 &\quad + \sum_{k=1}^K \sum_{v=1}^V \gamma_k \log b_{kv}.
 \end{aligned}$$

Lemma (exercise)

Let $\Psi(\cdot) := \Gamma'(\cdot)/\Gamma(\cdot)$ be the digamma function. Then

$$\mathbb{E}_{\text{Dir}(\theta|\eta)} \log \theta_i = \Psi(\eta_i) - \Psi(\|\eta\|_1).$$

$$\log p(y, z, \pi, B)$$

$$\propto \sum_{i=1}^N \left[\sum_{k=1}^K \alpha_k \log \pi_{ik} + \sum_{\ell=1}^{L_i} \left(\sum_{k=1}^K 1_{z_i \ell = k} \log \pi_{ik} + \sum_{v=1}^V \sum_{k=1}^K 1_{y_i \ell = v} 1_{z_i \ell = k} \log b_{kv} \right) \right] + \sum_{k=1}^K \sum_{v=1}^V \gamma_k \log b_{kv}.$$

$$\log p(y, z, \pi, B)$$

$$\propto \sum_{i=1}^N \left[\sum_{k=1}^K \alpha_k \log \pi_{ik} + \sum_{\ell=1}^{L_i} \left(\sum_{k=1}^K 1_{z_{i\ell}=k} \log \pi_{ik} + \sum_{v=1}^V \sum_{k=1}^K 1_{y_{i\ell}=v} 1_{z_{i\ell}=k} \log b_{kv} \right) \right] + \sum_{k=1}^K \sum_{v=1}^V \gamma_k \log b_{kv}.$$

$$\begin{aligned} \log p(y, z, \pi, B) \\ \propto \sum_{i=1}^N \left[\sum_{k=1}^K \alpha_k \log \pi_{ik} + \sum_{\ell=1}^{L_i} \left(\sum_{k=1}^K 1_{z_i \ell = k} \log \pi_{ik} + \sum_{v=1}^V \sum_{k=1}^K 1_{y_i \ell = v} 1_{z_i \ell = k} \log b_{kv} \right) \right] + \sum_{k=1}^K \sum_{v=1}^V \gamma_k \log b_{kv}. \end{aligned}$$

Using counts keeps space and time complexity low

- ▶ Storing $\tilde{z}_{i\ell k}$ requires $\mathcal{O}(NK \sum_i L_i)$ space. In practice, one works with (sparse) count data

n_{iv} = number of times word v appears in document i ,

and variables c_{ivk} , thus reducing storage costs (and the dimension of the underlying integral!) to $\mathcal{O}(NVK)$.

- ▶ For hidden variable models, **EM** is VB with

$$q(z, \theta) = \pi(z|\theta)\delta_{\tilde{\theta}}(\theta).$$

- ▶ **Variational EM** is VB with

$$q(z, \theta) = q(z)\delta_{\tilde{\theta}}(\theta).$$

- ▶ VB for any PGMs with NEF arrows is **variational message passing**.
- ▶ Rather approximating

$$\pi(\theta) \approx \prod_{f=1}^F q_f(\theta)$$

leads to **expectation propagation**.

- ▶ These days, **ADVI with stochastic gradients** is the default VI choice in probabilistic programming software like PyMC3, Stan, or PyRo.

- [1] A. Kucukelbir et al. “Automatic differentiation variational inference”. In: *The Journal of Machine Learning Research* 18.1 (2017), pp. 430–474.
- [2] K. Murphy. *Machine learning: a probabilistic perspective*. MIT Press, 2012.