DATA20023 Bayesian Machine Learning

Exercise 3: Distributional approximations and mixture models **Due Feb 14**

1 Distributional approximations for Poisson-gamma

Consider again the same model we studied in Exercise 2 with

$$\begin{aligned} x_n &\sim \text{Poisson}(uv), \\ u &\sim \text{Gamma}(5, 1), \\ v &\sim \text{Gamma}(4, 4). \end{aligned}$$

where individual observations x_n are conditionally independent of each other and we have the observations $\mathbf{x} = [3, 4, 3, 9, 10, 3, 2, 3, 3, 2]$. Check the model solutions for joint log-probability and conditional distributions if you did not manage to derive them by yourself.

(a) Form **Laplace approximation** for the model. First find the MAP estimate (by any means you want) and then determine the precision of the approximation by forming the second derivatives of the log-probability and evaluating them at the MAP estimate. You should do this by deriving the second derivatives by hand, but you can verify your solution using automatic differentiation as shown in the lecture notebook. No need to use any transformation here, but rather make the approximation directly for u and v even though we know the marginals will not be normal distributions.

Plot the marginals of the posterior approximation to see how well it works. You can compare the result against the MCMC results obtained last time – is this a good approximation?

(b) Derive update equations for **Mean-field variational approximation** for this model, using the approximation $q(u, v|\lambda) = q(u|a, b)q(v|c, d)$ where $\lambda = \{a, b, c, d\}$ are the parameters of the approximation. Remember that we already know the conditional distributions that are the starting point for these derivations, and you should already know which distribution family the terms $q(u|a, b)$ and $q(v|a, b)$ are.

Implement the variational approximation and plot the posterior marginals also for this approximation. How does it differ from the Laplace approximation?

2 Mixture models

2.1 Material

This exercise is about *mixture of Gaussians* (MoG) explained in Section 11.2 of Murphy's book. Read the section in detail, but ignore 11.2.4 about mixture of experts. See also Section 21.6 about variational approximation for this model.

2.2 Model and Gibbs sampler

We use a mixture of univariate normal distributions defined as

$$\begin{aligned} p(\boldsymbol{\pi}|\alpha) &= \text{Dirichlet}(\alpha) \\ p(\mu_k|\tau_0) &= \mathcal{N}(0, \tau_0) \\ p(\tau_k|\alpha_0, \beta_0) &= \text{Gamma}(\alpha_0, \beta_0) \\ p(z_n|\boldsymbol{\pi}) &= \text{Categorical}(\boldsymbol{\pi}) \\ p(x_n|\mu_{z_n}, \tau_{z_n}, z_n) &= \mathcal{N}(\mu_{z_n}, \tau_{z_n}) \end{aligned}$$

where τ always refers to precision. The joint likelihood and the conditional distributions needed for a Gibbs sampler are provided below as a starting point to make derivation of variational updates easier. Read this part carefully and think about how the conditional distributions were derived.

If we write the joint likelihood of cluster allocations and data as

$$p(x_n, z_n|\dots) = \left(\prod_k p(z_n|\boldsymbol{\pi}) p(x_n|\mu_k, \tau_k)\right)^{\mathbb{I}[z_n=k]},$$

we get a nice expression for the log-density as

$$\begin{aligned} &\log \left[\prod_n \prod_k \left(p(z_n|\boldsymbol{\pi}) p(x_n|\mu_{z_n}, \tau_{z_n})\right)^{\mathbb{I}[z_n=k]}\right] \left[\prod_k p(\mu_k) p(\tau_k)\right] p(\boldsymbol{\pi}) \\ &= \sum_n \sum_k \mathbb{I}[z_n = k] \left[\log \pi_k + \left(-\frac{1}{2}\log(2\pi) + \frac{1}{2}\log \tau_d - \frac{1}{2}\tau_d (x_d - \mu_d)^2\right)\right] \\ &\quad + \sum_k \left(-\frac{1}{2}\log(2\pi) + \frac{1}{2}\log \tau_0 - \frac{1}{2}\tau_0 \mu_k^2 + \alpha_0 \log(\beta_0) - \Gamma(\alpha_0) + (\alpha_0 - 1)\log(\tau_k) - \beta_0 \tau_k\right) \\ &\quad + \sum_k (\alpha - 1)\log \pi_k - \log B(\alpha). \end{aligned}$$

Note that π in $\log(2\pi)$ is the mathematical constant, not the parameter.

Simple algebraic manipulation of the joint density gives the conditional distributions:

$$\begin{aligned} p(\pi|\boldsymbol{z}, \alpha) &= \text{Dirichlet}(\alpha + \sum_k \mathbb{I}[z_n = k]) = \text{Dirichlet}(\alpha + N_k), \\ p(\tau_k|\dots) &= \text{Gamma}(\alpha_0 + \frac{1}{2}N_k, \beta_0 + \frac{1}{2}\sum_n \mathbb{I}(z_n = k)(x_n - \mu_k)^2), \\ p(\mu_k|\dots) &= \mathcal{N}(\frac{\tau_k \sum_n \mathbb{I}[z_n = k]\mu_k}{t_0 + N_k \tau}, t_0 + N_k \tau), \\ p(z_n = k|x_n, \dots) &= \frac{e^{\nu_{nk}}}{\sum_j e^{\nu_{nj}}} \end{aligned}$$

where

$$\nu_{nk} = \log p(z_n = k|x_n, \dots) \propto \log \pi_k + \left(-\frac{1}{2}\log(2\pi) + \frac{1}{2}\log \tau_k - \frac{1}{2}\tau_k(x_n - \mu_k)^2\right)$$

and N_k is always the number of samples currently allocated for each cluster. In most cases these conditionals follows directly from the simple Gaussian model we covered in the lectures, but we just need to consider only the samples that belong to this cluster.

2.3 Variational approximation

Building on he information provided above, derive a **mean-field variational approximation** for the model using the factorized approximation

$$q(\boldsymbol{z}, \boldsymbol{\pi}, \boldsymbol{\tau}, \boldsymbol{\mu}) = \prod_n [q(z_n|\boldsymbol{\xi}_n)]\, q(\boldsymbol{\pi}|\boldsymbol{\eta}) \prod_k [q(\tau_k|a_k, b_k) q(\mu_k|m_k, t_k)]$$

where the approximation terms are categorical distribution, Dirichlet distribution, Gamma distribution, and normal distribution (in order of appearance).

(a) Derive the update rules for coordinate-ascent variational inference for the model. That is, derive the updates needed for updating the parameters of each approximation term in turn. Remember that the updates are formed as expectations of the conditional distributions provided above, and note that Section 21.6 of Murphy gives the right answers for a slightly different mixture model – you can consult the derivations there but should remember that our model is a simpler case.

(b) Implement the approximation and evaluate it on synthetic data sampled from a model with $K = 3$, $\boldsymbol{\pi} = [0.35, 0.2, 0.45]$, $\mu_1 = 0, \mu_2 = -1.5, \mu_3 = 1.5$, $\tau_1 = 1$, $\tau_2 = 5$ and $\tau_3 = 5$; the code below creates the data and illustrates is using histograms. You can use $\alpha = 1$, $\alpha_0 = \beta_0 = 0.5$ and $\tau_0 = 0.5$ as the hyperparameters or try some other values.

```
import numpy as np
import scipy.stats as stats

K = 3; N= 500; x = np.zeros((N,))
pi_true = np.array([0.35, 0.2, 0.45]);
mu_true = np.array([0., -1.5, 1.5]);
tau_true = np.array([1., 5., 5.])

z_true = stats.multinomial(1,pi_true).rvs(N) # using one-hot encoding
colors = ['r','g','b']
plt.subplot(1,2,1)
for k in range(3):
    x[z_true[:,k]==1] = stats.norm(mu_true[k], 1./np.sqrt(tau_true[k])).rvs(sum(z_true[:,k]))
    plt.hist(x[z_true[:,k]==1], color=colors[k], bins=np.linspace(-4.,4.,50))
plt.subplot(1,2,2)
plt.hist(x,bins=np.linspace(-4.,4.,50))
```

(c) Inspect the results, at least by printing the final parameters of the key approximation terms and ideally also graphically. Does the method work well?

(d) Bonus: You can also derive and implement evaluation of the **evidence lower bound** (ELBO) for this model. You get extra points for that, which can be used for compensating missing points in other exercises.