Advanced Probabilistic Machine Learning and Applications

Martina Contisciani and Caterina De Bacco

November 3, 2021

Tutorial 3: Bayesian Mixture Models and Gibbs sampling

Exercise 1: Categorical Mixture Model (CMM)

In this tutorial, we will continue working with the CMM and the twitter dataset presented in Tutorial 2. Here, we will use different versions of the Gibbs sampling algorithm to find the posterior distributions of the cluster assignments $\mathbf{Z}$ and model parameters $(\boldsymbol{\pi}, \boldsymbol{\theta})$.

As a recap, the dataset $\mathbf{X} = \{\mathbf{x}_1, \dots, \mathbf{x}_N\}^\top$ describes a set of N documents, here tweets generated by U users. Each tweet has been cleaned and pre-processed (lemmatisation, lowerization, and stemming) using a dictionary of words I, and it is represented as $\mathbf{x}_n = (x_{n1}, \dots, x_{nW_n})$, i.e. as a vector of W_n words. Each word $x_{nj} \in \{1, \dots, |I|\}$ is described by its position in the dictionary. We introduce also a set $\mathbf{Z}$ of latent variables which represent the cluster assignments.

We work with the following Bayesian Mixture Model:

$$p(\mathbf{x}, \mathbf{z}, \boldsymbol{\pi}, \boldsymbol{\theta}) = p(\boldsymbol{\pi}|\boldsymbol{\alpha})p(\boldsymbol{\theta}|\boldsymbol{\gamma}) \prod_{n=1}^{N} p(z_n|\boldsymbol{\pi})p(\mathbf{x}_n|z_n, \boldsymbol{\theta}),$$

where

$$p(\boldsymbol{\pi}|\boldsymbol{\alpha}) = Dir(\boldsymbol{\pi}|\boldsymbol{\alpha}) \quad p(\boldsymbol{\theta}|\boldsymbol{\gamma}) = \prod_{k=1}^{K} Dir(\boldsymbol{\theta}_k|\boldsymbol{\gamma})$$

$$p(z_n|\boldsymbol{\pi}) = Cat(z_n|\boldsymbol{\pi}) \quad p(\mathbf{x}_n|z_n, \boldsymbol{\theta}) = \prod_{j=1}^{W_n} Cat(x_{nj}|\boldsymbol{\theta}_{z_n}).$$

Remember that the conjugate prior of a Categorical distribution is the Dirichlet distribution, and notice the prior distributions for each $\boldsymbol{\theta}_k$ share the same set of parameters.

1. Algorithm 1 approximates (using samples) the posterior distribution $p(\mathbf{z}, \boldsymbol{\pi}, \boldsymbol{\theta}|\mathbf{x})$. Derive the conditional distributions needed to sample in steps (1), (2), and (3) of the Gibbs sampling algorithm.
2. Algorithm 2 approximates (using samples) the posterior distribution $p(\mathbf{z}, \boldsymbol{\theta}|\mathbf{x})$. Derive the conditional distributions needed to sample in steps (1) and (2) of the $\boldsymbol{\pi}$ collapsed Gibbs sampling algorithm.
3. Algorithm 3 approximates (using samples) the posterior distribution $p(\mathbf{z}|\mathbf{x})$. Derive the conditional distribution needed to sample in step (1) of the $\boldsymbol{\pi}, \boldsymbol{\theta}$ collapsed Gibbs sampling algorithm.
4. Implement the log-likelihood of the model, i.e. $\log p(\mathbf{x}|\mathbf{z}, \boldsymbol{\theta})$, in the jupyter notebook.
5. Implement Algorithm 1, i.e. the posterior distributions obtained in point 1) and fill in the function *fit_no_collapsed_Gibbs*. Then, train the algorithm for 80 iterations with a burn in period $\tau_{burn-in} = 20$ iterations.

6. Using your implementation of Algorithm 1, and the implementations of Algorithm 2 and 3 provided in the jupyter notebook, explain the differences in convergence speed of the algorithms in terms of number of iterations and time. What is the reason behind those differences?

Algorithm 1: Gibbs sampling algorithm

Initialize cluster assignments $\mathbf{Z}$ and model parameters $\boldsymbol{\pi}$, $\boldsymbol{\theta}$;
for $\tau = 1, \ldots, N_{it}$ **do**
  Sample $\boldsymbol{\pi} \sim p(\boldsymbol{\pi}|\mathbf{x}, \mathbf{z}, \boldsymbol{\theta}) = p(\boldsymbol{\pi}|\mathbf{z})$; (1)
  for $k = 1, \ldots, K$ **do**
    Sample $\boldsymbol{\theta}_k \sim p(\boldsymbol{\theta}_k|\mathbf{x}, \mathbf{z}, \boldsymbol{\pi}) = p(\boldsymbol{\theta}_k|\mathbf{x}, \mathbf{z})$; (2)
  end
  for $n = 1, \ldots, N$ **do**
    Sample $z_n \sim p(z_n|\mathbf{x}, \mathbf{z}_{-n}, \boldsymbol{\pi}, \boldsymbol{\theta}) = p(z_n|\mathbf{x}_n, \boldsymbol{\pi}, \boldsymbol{\theta})$; (3)
  end
end

Algorithm 2: $\boldsymbol{\pi}$ collapsed Gibbs sampling algorithm

Initialize cluster assignments $\mathbf{Z}$ and model parameters $\boldsymbol{\theta}$;
for $\tau = 1, \ldots, N_{it}$ **do**
  for $k = 1, \ldots, K$ **do**
    Sample $\boldsymbol{\theta}_k \sim p(\boldsymbol{\theta}_k|\mathbf{x}, \mathbf{z})$; (1)
  end
  for $n = 1, \ldots, N$ **do**
    Sample $z_n \sim p(z_n|\mathbf{x}, \mathbf{z}_{-n}, \boldsymbol{\theta}) = p(z_n|\mathbf{x}_n, \mathbf{z}_{-n}, \boldsymbol{\theta})$; (2)
  end
end

Algorithm 3: $\boldsymbol{\pi}, \boldsymbol{\theta}$ collapsed Gibbs sampling algorithm

Initialize cluster assignments $\mathbf{Z}$;
for $\tau = 1, \ldots, N_{it}$ **do**
  for $n = 1, \ldots, N$ **do**
    Sample $z_n \sim p(z_n|\mathbf{x}, \mathbf{z}_{-n})$; (1)
  end
end