Variational Inference: A Review for Statisticians

Xiang Cui, Yutong Dai, Shuhui Guo

December 27, 2018

1 Introduction

In the application generative probabilistic models, often introducing latent variables help govern the distribution of the data. In Bayesian statistics, one central task is to make inferences on the latent variables. Consider a joint density of latent variables $\mathbf{z} = z_{1:m}$ and observations $\mathbf{x} = x_{1:n}$,

$$p(\mathbf{z}, \mathbf{x}) = p(\mathbf{z})p(\mathbf{x}|\mathbf{z}) \tag{1}$$

where $p(\mathbf{z})$ is the prior density and $p(\mathbf{x}|\mathbf{z})$ is the likelihood. In Bayesian inference, one is interested to find the posterior $p(\mathbf{z}|\mathbf{x})$. However, in serious applications, this posterior is often hard to calculate. Markov chain Monte Carlo (MCMC) sampling is a traditional strategy, which can generate asymptotically exact samples from the target distribution (Robert and Casella, 2013). However, it scales poorly to large datasets and complex models.
Variational inference (VI) has been proposed as an alternative method to approximate Bayesian inference which could scale to large dataset. Rather than use sampling in MCMC, variational inference uses the following optimization to do approximation.

$$q^*(\mathbf{z}) = \underset{q(\mathbf{z})\in\mathscr{Q}}{\operatorname{argmin}} \mathrm{KL}(q(\mathbf{z})||p(\mathbf{z}|\mathbf{x})) \tag{2}$$

where $\mathscr{Q}$ is a family of approximate densities, which is both flexible enough to capture the information in $p(\mathbf{z}|\mathbf{x})$ and computationally tractable. And the Kullback-Leibler (KL) divergence measures the divergence of the approximate density $q(\mathbf{z})$ from the true posterior $p(\mathbf{z}|\mathbf{x})$. Since VI solves an optimization problem, it is faster than MCMC. Therefore, variational inference is more appropriate for large-scale problems, such as document analysis, computational neuroscience, and computer vision. However, one has to point out that Variational inference might find local optimal and the approximated density often underestimated the variance of the true posterior density.
In this report, we will apply variational inference on the mixture of Gaussians and the numerical experiments can provide insights on its mechanism.

2 Variational Inference

2.1 Approximate Inference

Let $\mathbf{x} = x_{1:n}$ be a set of observations and $\mathbf{z} = z_{1:m}$ be a set of latent variables, with joint density $p(\mathbf{z}, \mathbf{x})$. The inference problem is to compute the conditional density of the latent variables given the observation. The posterior $p(\mathbf{z}|\mathbf{x})$ is calculated by

$$p(\mathbf{z}|\mathbf{x}) = \frac{p(\mathbf{z}, \mathbf{x})}{p(\mathbf{x})} \tag{3}$$

The goal of variational inference is to find a $q(\mathbf{z})$ in a family of approximate densities $\mathscr{Q}$, satisfying the optimization

$$q^*(\mathbf{z}) = \underset{q(\mathbf{z})\in\mathscr{Q}}{\operatorname{argmin}} \mathrm{KL}(q(\mathbf{z})||p(\mathbf{z}|\mathbf{x})) \tag{4}$$

where the KL divergence is calculated as

$$\mathrm{KL}(q(\mathbf{z})||p(\mathbf{z}|\mathbf{x})) = \mathbb{E}[\log q(\mathbf{z})] - \mathbb{E}[\log p(\mathbf{z}|\mathbf{x})] \tag{5}$$

where all expectations are taken with respect to $q(\mathbf{z})$. Expand the conditional,

$$\mathrm{KL}(q(\mathbf{z})||p(\mathbf{z}|\mathbf{x})) = \mathbb{E}[\log q(\mathbf{z})] - \mathbb{E}[\log p(\mathbf{z}, \mathbf{x})] + \log p(\mathbf{x}) \tag{6}$$

Based on the above equation, the optimization problem is not computable because the evidence $\log p(\mathbf{x})$ is required to compute. Therefore, an alternative optimization objective is proposed as

$$\mathrm{ELBO}(q) = \mathbb{E}[\log p(\mathbf{z}, \mathbf{x})] - \mathbb{E}[\log q(\mathbf{z})] \tag{7}$$

This function is called the evidence lower bound (ELBO). It lower-bounds the (log) evidence because

$$\log p(\mathbf{x}) = \mathrm{KL}(q(\mathbf{z})||p(\mathbf{z}|\mathbf{x})) + \mathrm{ELBO}(q) \tag{8}$$

The bound then follows from the fact that $\mathrm{KL}(\cdot) \geq 0$ (Kullback and Leibler, 1951). Maximizing the ELBO equals to minimizing the KL divergence.

2.2 The Mean-Field Variational Family

We focus on the mean-field variational family to complete the specification of the optimization problem(4). The mean-field variational family refers to the family where the latent variables are mutually independent and each governed by a distinct factor in the variational density. A generic member of the mean-field variational family is

$$q(\mathbf{z}) = \prod_{j=1}^{m} q_j(z_j) \tag{9}$$

Each latent variable z_j is governed by its own variational factor, the density $q_j(z_j)$. In optimization, these variational factors are chosen to maximize the ELBO.

2.3 Coordinate Ascent Mean-Field Variational Inference

In solving this approximation optimization problem, coordinate ascent variational inference (CAVI) (Bishop, 2006) is one of the most commonly used algorithms. CAVI can iteratively optimize each factor of the mean-field variational density with the others fixed and find a local optimum for ELBO.
To derive the algorithm, we first need one result. Consider the jth latent variable z_j. The complete conditional of z_j is its conditional density given all of the other latent variables in the model and observations, $p(z_j|\mathbf{z}_{-j}, \mathbf{x})$. Fix the other variational factors $q_l(z_l), l \neq j$. The optimal $q_j(z_j)$ is

$$q_j^*(z_j) \propto \exp\{\mathbb{E}_{-j}[\log p(z_j|\mathbf{z}_{-j}, \mathbf{x})]\} \tag{10}$$

The expectation in equation (10) is with respect to the variational density over $\mathbf{z}_{-j}$, that is, $\prod_{l \neq j} q_l(z_l)$. Equivalently, equation (10) is proportional to the exponentiated log of the joint,

$$q_j^*(z_j) \propto \exp\{\mathbb{E}_{-j}[\log p(z_j, \mathbf{z}_{-j}, \mathbf{x})]\} \tag{11}$$

Because all the latent variables are independent in the mean-field family, the expectations on the right hand side of (11) do not involve the jth variational factor. Thus this is a valid coordinate update. This equation underlie the CAVI

algorithm, presented in **Algorithm 1**. Through the iterations, the ELBO of equation (7) will go uphill, eventually finding a local optimum.

Algorithm 1: Coordinate ascent variational inference (CAVI)

Input: A model $p(\mathbf{x}, \mathbf{z})$, a data set $\mathbf{x}$
Output: A variational density $q(\mathbf{z}) = \prod_{j=1}^{m} q_j(z_j)$
Initialization Variational factors $q_j(z_j)$
while *the ELBO has not converged* **do**
  for $j \in \{1, \ldots, m\}$:
  Set $q_j(z_j) \propto \exp[\mathbb{E}_{-j}[\log p(z_j|\mathbf{z}_{-j}, \mathbf{x})]]$
  end
  Compute ELBO(q)
end
return $q(z)$

Algorithm 1 CAVI is closely related to Gibbs sampling. The Gibbs sampler works in a way that iteratively samples the latent variables conditionally. And for CAVI, it uses the same complete conditional in equation (10).

3 Bayesian Mixture of Gaussians

3.1 Unit-Variance Univariate Gaussian

We will use multivariate Mixture of Gaussians with general model setup in our application. The paper(Blei et al., 2017) gives a variational inference method of Bayesian mixture of unit-variance univariate Gaussian. Hence, we only give a brief summary here.

For a Bayesian mixture of unit-variance univariate Gaussians, there are K mixture components, corresponding to K Gaussian distributions with means $\boldsymbol{\mu} = \{\mu_1, \ldots, \mu_K\}$. The mean parameters are drawn independently from a common prior $p(\mu_k)$, which is assumed to be a Gaussian $\mathcal{N}(0, \sigma^2)$, where the prior variance σ^2 is a parameter. To generate an observation x_i from the model, a cluster assignment c_i is first chosen. It indicates which latent cluster x_i comes from and is drawn from a categorical distribution over $\{1, \ldots, K\}$. c_i is encoded as an indicator K-vector, all zeros except for a one in the position corresponding to x_i's cluster. Then x_i is drawn from the corresponding Gaussian $\mathcal{N}(c_i^{\mathrm{T}}\boldsymbol{\mu}, 1)$. The full hierarchical model is

$$\mu_k \sim \mathcal{N}(0, \sigma^2) \quad k = 1, \ldots, K; \quad c_i \sim Categorical(1/K, \ldots, 1/K) \quad i = 1, \ldots, n \tag{12}$$

$$x_i|c_i, \boldsymbol{\mu} \sim \mathcal{N}(c_i^{\mathrm{T}}\boldsymbol{\mu}, 1) \quad i = 1, \ldots, n; \quad p(\boldsymbol{\mu}, \mathbf{c}, \mathbf{x}) = p(\boldsymbol{\mu}) \prod_{i=1}^{n} p(c_i)p(x_i|c_i, \boldsymbol{\mu}) \tag{13}$$

The latent variables are $\mathbf{z} = \{\boldsymbol{\mu}, \mathbf{c}\}$, the K class means and n class assignments.

For estimation, the mean-field variational family contains approximate posterior densities of the form

$$q(\boldsymbol{\mu}, \mathbf{c}) = \prod_{k=1}^{K} q(\mu_k; m_k, s_k^2) \prod_{i=1}^{n} q(c_i; \varphi_i) \tag{14}$$

In this equation, the factor $q(\mu_k; m_k, s_k^2)$ is a Gaussian distribution on the kth mixture component's mean parameter with mean m_k and variance s_k^2. The factor $q(c_i; \varphi_i)$ is a distribution on the ith observation's mixture assignment with probabilities φ_i.

There are two types of variational parameters: categorical parameters φ_i for approximating the posterior cluster assignment of the ith data point and the Gaussian parameters m_k and s_k^2.

Algorithm 2 presents the CAVI for the Bayesian mixture of Gaussians.

Algorithm 2: CAVI for Gaussian mixture model

Input: Data $x_{1:n}$, number of components K, prior variance of component means σ^2
Output: Variational densities $q(\mu_k; m_k, s_k^2)$(Gaussian) and $q(c_i; \varphi_i)$(K-categorical)
Initialization Variational parameters $\mathbf{m} = m_{1:K}, \mathbf{s}^2 = s_{1:K}^2$, and $\varphi = \varphi_{1:n}$
while *the ELBO has not converged* **do**
  for $j \in \{1, \ldots, n\}$
  set $\varphi_{ik} \propto \exp\{\mathbb{E}[\mu_k; m_k, s_k^2]x_i - \mathbb{E}[\mu_k^2; m_k, s_k^2]/2\}$
  end
  for $k \in \{1, \ldots, K\}$
  set $m_k \leftarrow \frac{\sum_i \varphi_{ik} x_i}{1/\sigma^2 + \sum_i \varphi_{ik}}$ and $s_k^2 \leftarrow \frac{1}{1/\sigma^2 + \sum_i \varphi_{ik}}$
  end
  Compute $\text{ELBO}(\mathbf{m}, \mathbf{s}^2, \varphi)$
end
return $q(\mathbf{m}, \mathbf{s^2}, \varphi)$

Once the variational density is fitted, it can be used as the posterior. For example, we can obtain a posterior decomposition of the data. We assign points to their most likely mixture assignment $\hat{c}_i = \text{argmax}_k \varphi_{ik}$ and estimate cluster means with their variational means m_k.

3.2 Multivariate Gaussian Mixture Model

For the model and CAVI algorithm presented in the paper (Blei et al., 2017), it could only deal with unit-variance univariate Gaussian, but the dataset we deal with is multivariate mixture gaussians, we need a more general CAVI algorithm. Hence, we consider a CAVI algorithm for the multivariate mixture gaussian in the Book (Nasrabadi, 2007). The observed dataset is denoted by $\mathbf{X} = \{x_1, \ldots, x_n\}$. For each observation x_n, we have a corresponding latent variable z_n comprising a 1-of-K binary vector with elements z_{nk} for $k = 1, \ldots, K$. We denote the latent variables by $\mathbf{Z} = \{z_1, \ldots, z_N\}$. The mixing coefficient is given by $\boldsymbol{\pi} = \{\pi_1, \ldots, \pi_K\}$, where $p(z_{nk} = 1) = \pi_k$ and $\sum_{k=1}^K \pi_k = 1$. Given the mixing coefficient, the conditional distribution of $\mathbf{Z}$ could be written in the form;

$$p(\mathbf{Z}|\boldsymbol{\pi}) = \prod_{n=1}^N \prod_{k=1}^K \pi_k^{z_{nk}} \tag{15}$$

And given the latent variables and component parameters, the conditional distribution of the observed data vectors could be given by:

$$p(\mathbf{X}|\mathbf{Z}, \boldsymbol{\mu}, \boldsymbol{\Lambda}) = \prod_{n=1}^N \prod_{k=1}^K \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Lambda}_k^{-1})^{z_{nk}} \tag{16}$$

where $\boldsymbol{\mu}_k$ and $\boldsymbol{\Lambda}_k$ represent the mean vector and the precision matrix for the k-th mixture component. Then the joint distribution of all the random variables could be written as:

$$p(\mathbf{X}, \mathbf{Z}, \boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Lambda}) = p(\mathbf{X}|\mathbf{Z}, \boldsymbol{\mu}, \boldsymbol{\Lambda}) p(\mathbf{Z}|\boldsymbol{\pi}) p(\boldsymbol{\pi}) p(\boldsymbol{\mu}|\boldsymbol{\Lambda}) p(\boldsymbol{\Lambda}) \tag{17}$$

where $\mathbf{X} = \{x_1, \ldots, x_n\}$ is the observed data. The prior over the mixing coefficient $\boldsymbol{\pi}$ is a Dirichlet distribution:

$$p(\boldsymbol{\pi}) = \text{Dir}(\boldsymbol{\pi}|\boldsymbol{\alpha}_0) = C(\boldsymbol{\alpha}_0) \prod_{k=1}^K \pi_k^{\alpha_0 - 1} \tag{18}$$

And for the mean and precision of each Gaussian component, we use the independent Gaussian-Wishart prior:

$$p(\boldsymbol{\mu}, \boldsymbol{\Lambda}) = p(\boldsymbol{\mu}|\boldsymbol{\Lambda}) p(\boldsymbol{\Lambda}) \tag{19}$$

$$= \prod_{k=1}^K \mathcal{N}(\boldsymbol{\mu}_k|\mathbf{m}_0, (\beta_0 \boldsymbol{\Lambda}_k)^{-1}) \mathcal{W}(\boldsymbol{\Lambda}_k|\mathbf{W}_0, v_0) \tag{20}$$

since this represents the conjugate prior distribution when both the mean and precision are unknown.
To estimate the posterior $p(\mathbf{Z}, \boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Lambda}|\mathbf{X})$, we consider a variational distribution which factorizes between the latent variables and the parameters, so that:

$$q(\mathbf{Z}, \boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Lambda}) = q(\mathbf{Z})q(\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Lambda}) \tag{21}$$

This is the only assumption that we need to make in order to obtain a tractable practical solution to the Bayesian Gaussian mixture model. The functional form of the factor $q(\mathbf{Z})$ and $q(\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Lambda})$ will be determined automatically by optimizing the variation distribution, using the general result (11).
Then we derive the optimal sequential update equations for these factors: For the factor $q(\mathbf{Z})$, using the general result (11), the log of the optimized factor is given by:

$$\ln q^\star(\mathbf{Z}) = \mathbb{E}_{\boldsymbol{\pi},\boldsymbol{\mu},\boldsymbol{\Lambda}}[\ln p(\mathbf{X}, \mathbf{Z}, \boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Lambda})] + \text{const} = \mathbb{E}_{\boldsymbol{\pi}}[\ln p(\mathbf{Z}|\boldsymbol{\pi})] + \mathbb{E}_{\boldsymbol{\pi},\boldsymbol{\Lambda}}[\ln p(\mathbf{X}|\mathbf{Z}, \boldsymbol{\mu}, \boldsymbol{\Lambda})] + \text{const} \tag{22}$$

Substituting the two conditional distribution and absorbing the terms independent of $\mathbf{Z}$, they get:

$$\ln q^\star(\mathbf{Z}) = \sum_{n=1}^{N}\sum_{k=1}^{K} z_{nk} \ln \rho_{nk} + \text{const} \tag{23}$$

where:

$$\ln \rho_{nk} = \mathbb{E}[\ln \pi_k] + \frac{1}{2}\mathbb{E}[\ln|\boldsymbol{\Lambda}_k|] - \frac{D}{2}\ln(2\pi) - \frac{1}{2}\mathbb{E}_{\boldsymbol{\mu}_k,\boldsymbol{\Lambda}_k}[(\mathbf{x}_n - \boldsymbol{\mu}_k)^{\mathrm{T}}\boldsymbol{\Lambda}_k(\mathbf{x}_n - \boldsymbol{\mu}_k)] \tag{24}$$

$$\mathbb{E}_{\boldsymbol{\mu}_k,\boldsymbol{\Lambda}_k}[(\mathbf{x}_n - \boldsymbol{\mu}_k)^{\mathrm{T}}\boldsymbol{\Lambda}_k(\mathbf{x}_n - \boldsymbol{\mu}_k)] = D\beta_k^{-1} + v_k(\mathbf{x}_n - \mathbf{m}_k)^{\mathrm{T}}\mathbf{W}_k(\mathbf{x}_n - \mathbf{m}_k) \tag{25}$$

$$\ln\widetilde{\Lambda}_k \equiv \mathbb{E}[\ln|\boldsymbol{\Lambda}_k|] = \sum_{i=1}^{D}\psi(\frac{v_k + 1 - i}{2}) + D\ln 2 + \ln|\mathbf{W}_k| \tag{26}$$

$$\ln\widetilde{\pi}_k \equiv \mathbb{E}[\ln \pi_k] = \psi(\alpha_k) - \psi(\hat{\alpha}) \tag{27}$$

Taking the exponential of both sides of (23) and normalize this distribution, we could get the optimal solution for the factor $q(\mathbf{Z})$:

$$q^\star(\mathbf{Z}) = \prod_{n=1}^{N}\prod_{k=1}^{K} r_{nk}^{z_{nk}} \quad \text{where} \quad r_{nk} = \frac{\rho_{nk}}{\sum_{j=1}^{K}\rho_{nj}} \tag{28}$$

At this point, it would be convenient for us to first define the following statsitcs:

$$N_k = \sum_{n=1}^{N} r_{nk}, \quad \bar{\mathbf{x}}_k = \frac{1}{N_k}\sum_{n=1}^{N} r_{nk}\mathbf{x}_n \tag{29}$$

$$\mathbf{S}_k = \frac{1}{N_k}\sum_{n=1}^{N} r_{nk}(\mathbf{x}_n - \bar{\mathbf{x}}_k)(\mathbf{x}_n - \bar{\mathbf{x}}_k)^{\mathrm{T}} \tag{30}$$

Then we could consider the factor $q(\boldsymbol{\pi}, \mathbf{u}, \boldsymbol{\Lambda})$, again using the general result (11) we have:

$$\ln q^\star(\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Lambda}) = \ln p(\boldsymbol{\pi}) + \sum_{k=1}^{K}\ln p(\boldsymbol{\mu}_k, \boldsymbol{\Lambda}_k) + \mathbb{E}_{\mathbf{Z}}[\ln p(\mathbf{Z}|\boldsymbol{\pi})] + \sum_{k=1}^{K}\sum_{n=1}^{N}\mathbb{E}[z_{nk}]\ln\mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Lambda}_k^{-1}) + \text{const} \tag{31}$$

where we could see that the right-hand side of this expression could decompose into a sum of terms involving π only and another sum of terms involving $\boldsymbol{\mu}$ and $\boldsymbol{\Lambda}$, which imply the variational posterior $q(\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Lambda})$ could factorize to give $q(\boldsymbol{\pi})q(\mu, \boldsymbol{\Lambda})$, hence we could get:

$$q(\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Lambda}) = q(\boldsymbol{\pi})\prod_{k=1}^{K} q(\boldsymbol{\mu}_k, \boldsymbol{\Lambda}_k) \tag{32}$$

Identifying the terms on the right hand of (31), we could get:

$$\ln q^{\star}(\boldsymbol{\pi}) = (\alpha_0 - 1)\sum_{k=1}^{K}\ln\pi_k + \sum_{k=1}^{K}\sum_{n=1}^{N} r_{nk}\ln\pi_k + \text{const} \tag{33}$$

Take the exponential on both sides, we could identify:

$$q^{\star}(\boldsymbol{\pi}) = \text{Dir}(\boldsymbol{\pi}|\boldsymbol{\alpha}) \quad \alpha_k = \alpha_0 + N_k \tag{34}$$

Also by identifying the terms on the right hand of (31), we could get:

$$q^{\star}(\boldsymbol{\mu}_k, \boldsymbol{\Lambda}_k) = \mathcal{N}(\boldsymbol{\mu}_k|\mathbf{m}_k, (\beta_k\boldsymbol{\Lambda}_k)^{-1})\mathcal{W}(\boldsymbol{\Lambda}_k|\mathbf{W}_k, v_k) \tag{35}$$

where

$$\beta_k = \beta_0 + N_k, \quad \mathbf{m}_k = \frac{1}{\beta_k}(\beta_0\mathbf{m}_0 + N_k\bar{\mathbf{x}}_k) \tag{36}$$

$$\mathbf{W}_k^{-1} = \mathbf{W}_0^{-1} + N_k\mathbf{S}_k + \frac{\beta_0 N_k}{\beta_0 + N_k}(\bar{\mathbf{x}}_k - \mathbf{m}_0)(\bar{\mathbf{x}}_k - \mathbf{m}_0)^{\mathrm{T}}, \quad v_k = v_0 + N_k \tag{37}$$

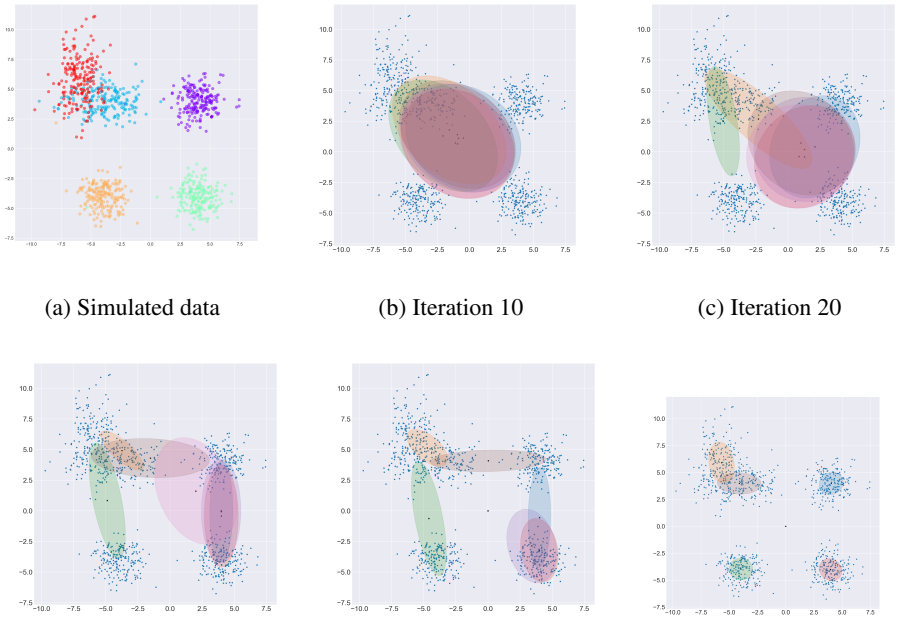
In order to test for convergence, we also use the ELBO in equation (6).
In general, the optimization of the variational distribution involves two stages: the first stage is using the current model parameters to evaluate the moments in (25), (26), (27); the second stage is to keep these moment as constant and recompute the different model parameters through (24), (28), (34), (35),(36),(37). Cycling the above two stages until the ELBO value converges could give us the optimal solution for the variational distribution.

Algorithm 3: CAVI for Multivariate Gaussian mixture model

Input: Data $X = x_{1:N}$, number of components K, prior for $\boldsymbol{\pi}$ in (18), independent Gaussian-Wishart prior for $\boldsymbol{\mu}, \boldsymbol{\Lambda}$ in (20)
Output: Variational densities for latent variables $q(\mathbf{Z})$ and for parameter $q(\boldsymbol{\pi})$ and $q(\boldsymbol{\mu}_k, \boldsymbol{\Lambda}_k)$
Initialization Variational parameters in $q(\mathbf{Z})$, $q(\boldsymbol{\pi})$ and $q(\boldsymbol{\mu}_k, \boldsymbol{\Lambda}_k)$
while *the ELBO has not converged* **do**
  Evaluate the moment $\mathbb{E}_{\boldsymbol{\mu}_k,\boldsymbol{\Lambda}_k}[(\mathbf{x}_n - \boldsymbol{\mu}_k)^{\mathrm{T}}\boldsymbol{\Lambda}_k(\mathbf{x}_n - \boldsymbol{\mu}_k)]$, $\mathbb{E}[\ln|\boldsymbol{\Lambda}_k|]$ and $\mathbb{E}[\ln\pi_k]$ as in (25), (26), (27)
  Update the optimal density for $q^*(Z)$ as in (24), (28)
  Update the statistics in (29), (30)
  Update the optimal density for $q^*(\boldsymbol{\pi})$ as in (34)
  Update the optimal density for $q^*(\boldsymbol{\mu}_k, \boldsymbol{\Lambda}_k)$ as in (35),(36),(37)
  Compute ELBO as in (7)
end
return $q(\mathbf{Z})$, $q(\boldsymbol{\pi})$, $q(\boldsymbol{\mu}, \boldsymbol{\Lambda})$

4 Experiment

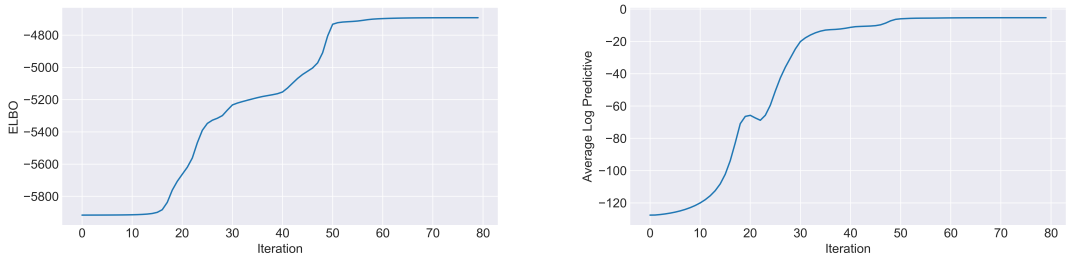
4.1 Simulation



(a) Simulated data (b) Iteration 10 (c) Iteration 20

(d) Iteration 30 (e) Iteration 40 (f) Converged

Figure 1: Apply CAVI on a simulated two dimensional mixture of Gaussians training. The figure (a) shows the original data, where we created 5 Gaussians with different mean vectors and covariance matrices. The ellipses are 1σ contours of estimated covariance matrices.



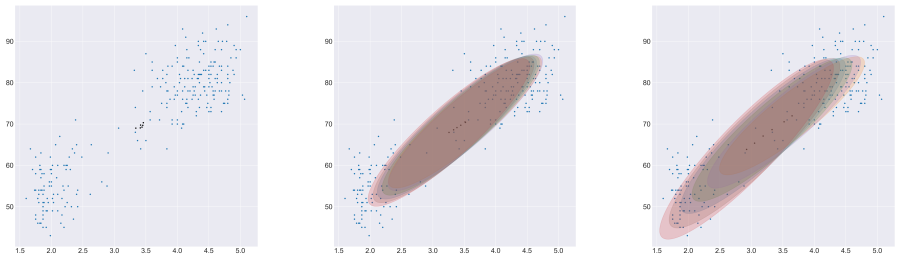
(a) ELBO (b) Average Log Predictive

Figure 2: The figure (a) shows successive improvement on the evidence lower bound while the figure (b) shows the average of the log predictive calculated on the simulated testing dataset.

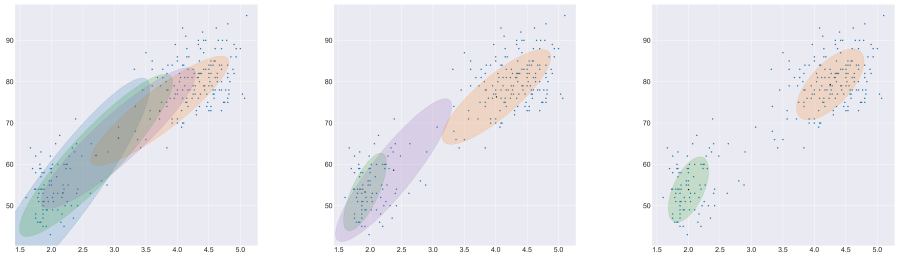
4.2 Old Faithful dataset

The Old Faithful dataset describes the waiting time between eruptions and the duration of the eruption for the Old Faithful geyser in Yellowstone National Park. It's a data frame with 272 observations on 2 variables, one is eruption time in mins and one is waiting time to next eruption. The data is plotted in Figure 3.

We apply the CAVI algorithm 3 to fit a Gaussian mixture models with 6 clusters to the Old Faithful dataset. Figure 3 shows the variational density of the components as the CAVI algorithm processes. We could see after convergence, only two components whose expected mixing coefficient are numerically distinguishable form zero.

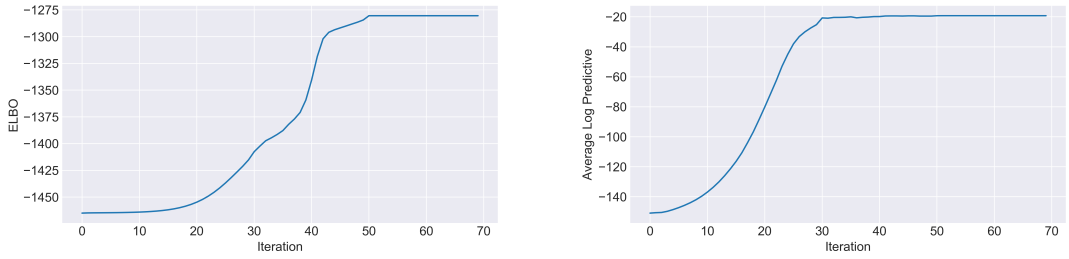


(a) Iteration 0 (b) Iteration 10 (c) Iteration 20



(d) Iteration 30 (e) Iteration 40 (f) Converged

Figure 3: Apply CAVI on the old faithful dataset. The ellipses are 1σ contours of estimated covariance matrices. Components whose expected mixing coefficient are numerically indistinguishable from zero are not plotted.



(a) ELBO (b) Average Log Predictive

Figure 4: The figure (a) shows successive improvement on the evidence lower bound while the figure (b) shows the average of the log predictive calculated on the original old faithful dataset, since there is no testing dataset.

4.3 Image Grouping

4.3.1 Grouping using CAVI

Consider the task of grouping image according to their color profiles for the imageCLEF dataset (Villegas and Paredes, 2014). The dataset was initially a database of over 31 million images was created by querying Google, Bing and

Yahoo!. Then a subset of 250,000 images was selected from this database by choosing the top images from a ranked list. We randomly select **5 thousand** images from the imageCLEF collection as our dataset to do the image grouping. One approach to group image according to their color profiles is to compute the color histogram of the images. Figure 5 shows the blue, green and red channel histograms of one images from the imageCLEF data. Each histogram is a vector of length 256, counting the number of pixel in the range of (0 to 255) in each channel. Concatenating the three color histogram gives a 768-dimensional representation of each image, regardless of its original size in pixel-space.

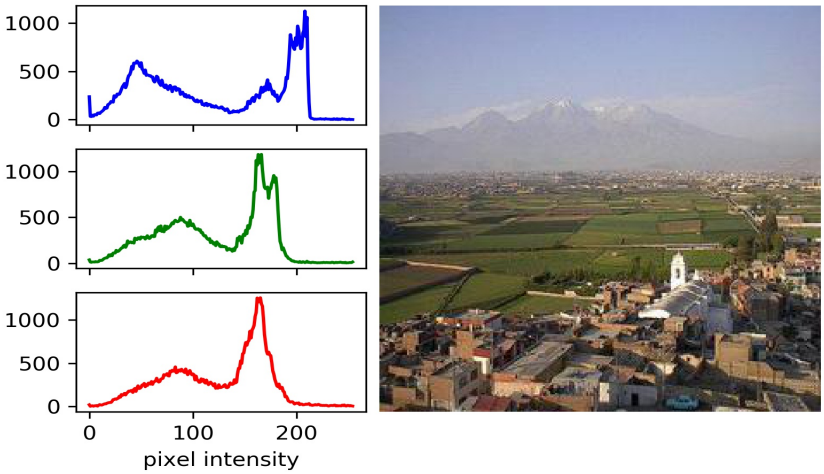


Figure 5: The left three histograms shows pixel counts on Blue, Green and Red channels for the right picture (from the imageCLEF datasets) respectively. The pixel intensity ranges from 0-255. So we use a 768 dimensional vector to characterize an image.

We use the CAVI Algorithm 3 to fit a Gaussian mixture models with thirty clusters to the image histograms for the 5000 images. Figure 6 shows similarly colored images assigned to three randomly chosen clusters. We could see that the CAVI Algorithm 3 could group the images well according to their color files.

(a) Yellow (b) Green (c) Blue

Figure 6: Classification results based on the mixture of Gaussians. Three subfigures show nine randomly sampled images from three clusters. We name each cluster by its dominant color.

4.3.2 Comparison with MCMC

For this image histogram clustering problem, we compare the CAVI to an implementation in Stan, the same as in Appendix B in (Blei et al., 2017), which No-U-Turn sampler(NUTS)(Carpenter et al., 2017) (Hoffman et al., 2012).

Stan is a probabilistic programming language for statistical inference written in C++. Stan implements gradient-based Markov chain Monte Carlo (MCMC) algorithms for Bayesian inference
When using the CAVI to fit a Gaussian mixture models with thirty clusters to the image histograms for the 5000 images, the ELBO converges in less than 1 minute, which means the CAVI could finish the parameter estimation within 1 minute. However, when using the NUTS in Stan, even with a simpler setting, fitting a diagonal covariance Gaussian mixture models with thirty clusters, it takes for more than two hours to run for only 60 iterations, which has been tested to be far from convergence.
Therefore, we could see that variational inference provides a better approach to approximate bayesian inference when dealing with large dataset and complex models.

5 Discussion and Extension

The mixture Gaussian model is a special case for conditionally conjugate models with local and global variables. Models like this come up frequently in Bayesian statistics and statistical machine learning, where the global variables are parameters and the local variables are latent variables. A close form of coordinate ascent variational inference(CAVI) algorithm could be derived for this general class of models.
Modern applications of probability models may analyze massive dataset and the mean-field variational inference could be scaled to big data using stochastic variational inference(SVI) (Hoffman et al., 2013). The CAVI algorithm could not scale to large dataset since it requires iterating through the entire data set at each iteration. As the data size grows, each iteration becomes too computational expensive. Hence instead of using coordinate ascent, a gradient-based optimization which climbs the ELBO by computing and following its gradient at each iteration was proposed. And that's the key idea of stochastic variational inference(SVI). The algorithm of SVI is simple, it repeatedly (a)subsample a data point from the full dataset, (b) use natural gradient-based stochastic optimization method to update the parameters with the subsampled data. This stochastic optimization could give noisy but cheap-to-compute gradients to reach the optimum of the ELBO. In the paper (Robbins and Monro, 1951), they proved results implying this stochastic optimization could successfully use the noisy unbiased gradients under certain condition. And this SVI could enable the modern machine learning to big data (Bottou and Cun, 2004).

References

David M Blei, Alp Kucukelbir, and Jon D McAuliffe. Variational inference: A review for statisticians. *Journal of the American Statistical Association*, 112(518):859–877, 2017.

Léon Bottou and Yann L Cun. Large scale online learning. In *Advances in neural information processing systems*, pages 217–224, 2004.

Bob Carpenter, Andrew Gelman, Matthew D Hoffman, Daniel Lee, Ben Goodrich, Michael Betancourt, Marcus Brubaker, Jiqiang Guo, Peter Li, and Allen Riddell. Stan: A probabilistic programming language. *Journal of statistical software*, 76(1), 2017.

Bob Carpenter Hoffman, D Matthew, and Andrew Gelman. Stan, scalable software for bayesian modeling. In *Proceedings of the NIPS Workshop on Probabilistic Programming*, 2012.

Matthew D Hoffman, David M Blei, Chong Wang, and John Paisley. Stochastic variational inference. *The Journal of Machine Learning Research*, 14(1):1303–1347, 2013.

Solomon Kullback and Richard A Leibler. On information and sufficiency. *The annals of mathematical statistics*, 22 (1):79–86, 1951.

Nasser M Nasrabadi. Pattern recognition and machine learning. *Journal of electronic imaging*, 16(4):049901, 2007.

Herbert Robbins and S Monro. [a]a stochastic approximation method, [o] annals math. *Statistics*, 22:400–407, 1951.

Christian Robert and George Casella. *Monte Carlo statistical methods*. Springer Science & Business Media, 2013.

Mauricio Villegas and Roberto Paredes. Overview of the imageclef 2014 scalable concept image annotation task. In *CLEF (Working Notes)*, pages 308–328. Citeseer, 2014.