CS 6190: Probabilistic Machine Learning Spring 2022

Homework 4
Meysam Alishahi (U1323606)

Handed out: 31 March, 2022
Due: 11:59pm, 15 April, 2022

- You are welcome to talk to other members of the class about the homework. I am more concerned that you understand the underlying concepts. However, you should write down your own solution. Please keep the class collaboration policy in mind.
- Feel free discuss the homework with the instructor or the TAs.
- Your written solutions should be brief and clear. You need to show your work, not just the final answer, but you do not need to write it in gory detail. Your assignment should be **no more than 10 pages**. Every extra page will cost a point.
- Handwritten solutions will not be accepted.
- The homework is due by **midnight of the due date**. Please submit the homework on Canvas.

Practice [100 points + 100 bonus]

1. [20 points] Suppose we have a scalar distribution,

$$p(z) \propto \exp(-z^2)\sigma(10z+3).$$

(a) [3 points] Although the normalization constant is not analytical, we can use Gauss-Hermite quadrature to calculate an accurate approximation. Please base on the example in "data/example-code/gmq_example.py", calculate the numerical approximation of the normalization constant, and report its value. With the normalization constant, please draw the density curve of $p(z)$, in the range $z \in [-5, 5]$.

Answer: Setting degree (number of points) to 300, the normalization constant is 1.169.

(b) [5 points] Implement the Laplace approximation of $p(z)$, and report the mean and variance of your Gaussian distribution. Draw the density of your Laplace approximation in the same plot as in (a).

Answer: To use the Laplace approximation, we need to compute the MAP and the second derivative at that point. Setting

$$h(z) = -\log p(z) = z^2 - \log \sigma(10z+3),$$

we can compute

$$h'(z) = 2z - 10(1 - \sigma(10z+3))$$

and

$$h''(z) = 2 + 100\sigma(10z+3)(1-\sigma(10z+3)).$$

Using an optimization method such as *"L-BFGS-B"* to minimize $h(z)$ (equivalently maximizing $p(z)$), we obtain $\mu = z_{max} = 0.09471726$ and thus $\sigma^2 = 1/h''(z_{max}) = 0.25917088$. Therefore, Laplace approximation density is $\mathcal{N}(0.095, 0.259)$.

(c) [10 points] Use the local variational inference method and EM-style updates as we discussed in the class (for logistic regression) to implement the variational approximation to $p(z)$. Report the form of your approximate distribution, and draw its density in the same plot as above.

Answer: It is proven that

$$\sigma(z) \geq \sigma(\xi) \exp\left\{\frac{z-\xi}{2} - \lambda(\xi)(z^2-\xi^2))\right\},$$

where

$$\lambda(\xi) = \frac{1}{2\xi}\left[\sigma(\xi) - \frac{1}{2}\right].$$

We can therefore write

$$\begin{aligned} p(z) &= \frac{1}{Z}\exp(-z^2)\sigma(10z+3) \\ &\geq \frac{1}{Z}\exp(-z^2)\sigma(\xi)\exp\left\{\frac{10z+3-\xi}{2} - \lambda(\xi)((10z+3)^2-\xi^2))\right\} \\ &\propto \mathcal{N}\big(z|0, \frac{1}{\sqrt{2}}\big)h(z,\xi). \end{aligned}$$

and accordingly,

$$\begin{aligned} \log p(z) &\geq \log\left(\mathcal{N}\big(z|0, \frac{1}{\sqrt{2}}\big)h(z,\xi)\right) + \text{Const} \\ &\geq -z^2 + \log\sigma(\xi) + \frac{10z+3-\xi}{2} - \lambda(\xi)\left((10z+3)^2-\xi^2)\right) + \text{Const} \\ &= -(1+100\lambda(\xi))z^2 + (5-60\lambda(\xi))z + \text{Const}(\xi). \\ &= -\frac{1}{2\sigma^2}(z^2 - 2\mu z +) + \cdots \end{aligned}$$

To apply the local variational inference method, we should solve

$$\max_{q,\xi} \mathbb{E}_{q(z)} \log\frac{\mathcal{N}\big(z|0, \frac{1}{\sqrt{2}}\big)h(z,\xi)}{q(z)}.$$

To use EM-algorithm, we first maximize it to find $q(z)$ given ξ fixed (E-step) and then we consider $q(z)$ fixed and maximized it to find ξ (M-step).

In the E-step, same as Mean-Field (or using complete square method), we obtain

$$\begin{aligned} q(z) &\propto \exp\left\{\log\left(\mathcal{N}\big(z|0, \frac{1}{\sqrt{2}}\big)h(z,\xi^{\text{old}})\right)\right\} \\ &\propto \mathcal{N}(z|\mu_0, \sigma_0), \end{aligned}$$

where

$$\mu_0 = \mu_0(\xi^{\text{old}}) = \frac{5-60\lambda(\xi^{\text{old}})}{2+200\lambda(\xi^{\text{old}})} \qquad \text{and} \qquad \sigma_0 = \sigma_0(\xi^{\text{old}}) = \frac{1}{\sqrt{2+200\lambda(\xi^{\text{old}})}}. \tag{1}$$

Therefore, **E-step update** will be

$$q(z) = \mathcal{N}\left(z|\mu_0(\xi^{\text{old}}), \sigma_0(\xi^{\text{old}})\right). \tag{2}$$

Now, in the M-step, we assume $q(z)$ is given and optimize

$$\begin{aligned}\mathcal{Q}(\xi, \xi^{\text{old}}) &= \mathbb{E}_{q(z)} \log \frac{\mathcal{N}(z|0, \frac{1}{\sqrt{2}})h(z,\xi)}{q(z)} \\ &= \mathbb{E}_{q(z)} \log h(z,\xi) + \text{Const} \\ &= \mathbb{E}_{q(z)} \left[\log \sigma(\xi) - \frac{\xi}{2} - \lambda(\xi)\big((10z+3)^2 - \xi^2\big)\right] + \text{Const} \\ &= \log \sigma(\xi) - \frac{\xi}{2} - \lambda(\xi)\left(100\sigma_0^2 + (10\mu_0+3)^2 - \xi^2\right)\big) + \text{Const.}\end{aligned}$$

We now set the derivative with respect to ξ equal to zero which implies

$$0 = \lambda'(\xi)\left(100\sigma_0^2 + (10\mu_0+3)^2 - \xi^2\right).$$

Therefore, using the same trick as explained in the class ($\lambda'(\xi) \neq 0$ for $\xi > 0$), the **M-step update** will be

$$\xi^{\text{new}} = \sqrt{100\sigma_0^2 + (10\mu_0+3)^2}. \tag{3}$$

Using these updates rules, we obtain the variational approximation

$$q(z) = \mathcal{N}(z|\mu = 0.307, \sigma = 0.329).$$

(d) [2 points] By comparing the "ground-truth" (from (a)) and the approximations (from (b,c)), what do you observe and conclude?
Answer:

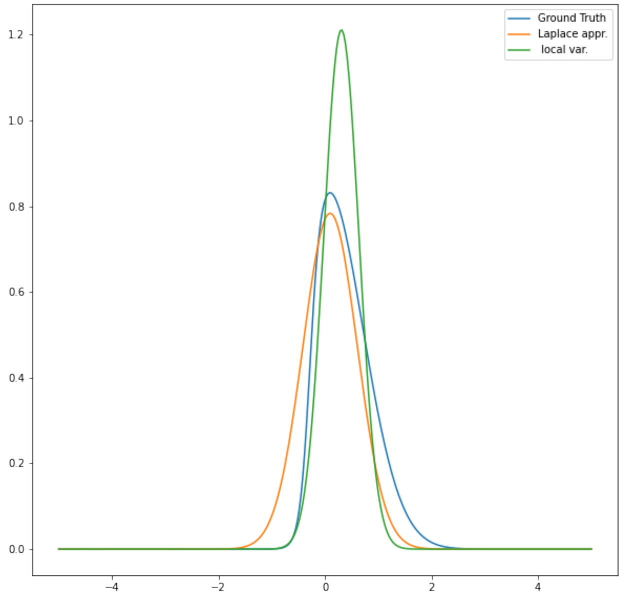

Figure 1: Ground-truth, Laplace, and local variational approximations

2. [50 points] Let us work on a real-world dataset we have met before. Please download the data from the folder"data/bank-note". The features and labels are listed in the file "data-desc.txt". The training data are stored in the file "train.csv", consisting of 872 examples. The test data are stored in "test.csv", and comprise of 500 examples. In both the training and testing datasets, feature values and labels are separated by commas. We assign the feature weight vector $\mathbf{w}$ a standard normal prior $\mathcal{N}(\mathbf{0}, \mathbf{I})$.

(a) [7 points] Implement the standard Laplace approximation to the posterior distribution of the feature weight vector. Report your approximate posterior. Now, use Gauss-Hermite quadrature to implement the calculation of the predictive distribution. Please be careful: **you need to do a proper variable transformation in the integral before applying the Gauss-Hermite quadrature because you integrate with a Gaussian like $\mathcal{N}(x|\mu, \sigma^2)$ rather than $\exp(-x^2)$!** Now we test the performance with two measures. First, we calculate the inner-product between the posterior mean of the weight vector and the feature vector of each test example, and throw the inner-product into the sigmoid function to calculate the probability that the test example is positive. If the probability is no less than 0.5, we classify the example to be positive (i.e., 1) otherwise we classify the example to be negative (i.e., 0). Report the prediction accuracy. Second, we calculate the average predictive likelihood of the test samples, namely we evaluate the predictive density value of each test sample based on the predictive distribution and then take an average. Note that in Bayesian learning, the predictive likelihood includes all the information of the (approximate) posterior distribution, hence is more preferred in the evaluation.

Answer: Note that we have added 1 as the fisrt entry to each feature vector to consider the bias term.

Standard Laplace approximation to the posterior distribution $p(\mathbf{w}|\mathbf{t})$ is

$$q(\mathbf{w}) = \mathcal{N}(\mathbf{w}|\boldsymbol{\mu}_0, \boldsymbol{\Sigma}_0)$$

where

$$\boldsymbol{\mu}_0 = [2.85594016, -2.69321758, -1.59105678, -1.8992619, -0.17689812]^\top$$

and

$$\boldsymbol{\Sigma}_0 = \begin{bmatrix} 0.13679752 & -0.05634869 & -0.02708631 & -0.0406641 & 0.02068531 \\ -0.05634869 & 0.11964241 & 0.05318788 & 0.07181374 & 0.00963915 \\ -0.02708631 & 0.05318788 & 0.05011859 & 0.05299433 & 0.02319023 \\ -0.0406641 & 0.07181374 & 0.05299433 & 0.06217003 & 0.0196692 \\ 0.02068531 & 0.00963915 & 0.02319023 & 0.0196692 & 0.0382288 \end{bmatrix}.$$

Performance on the test set using $\boldsymbol{\mu}_0 = \mathbf{w}_{MAP}$:
Accuracy: 99%

Calculation of the predictive distribution:
Note that we know $\mathbf{w}|\mathbf{t} \sim \mathcal{N}(\boldsymbol{\mu}_0, \boldsymbol{\Sigma}_0)$ (approximately). For a given feature vector $\mathbf{x}$, setting $a = \mathbf{x}^\top\mathbf{w}$, we obtain

$$a|\mathbf{t} \sim \mathcal{N}(\mathbf{x}^t\boldsymbol{\mu}_0, \mathbf{x}^\top\boldsymbol{\Sigma}_0\mathbf{x}).$$

Therefore, setting $m = \mathbf{x}^t\boldsymbol{\mu}_0$ and $s^2 = \mathbf{x}^\top\boldsymbol{\Sigma}_0\mathbf{x}$,

$$\begin{aligned} p(t=1|\mathbf{x}) &= \int \sigma(a)p(a|\mathbf{t})\mathrm{d}a \\ &\approx \int \sigma(a)\mathcal{N}(a|m, s^2)\mathrm{d}a \\ &= \frac{1}{\sqrt{\pi}}\int e^{-z^2}\sigma(\sqrt{2}sz + m)\mathrm{d}z. \end{aligned}$$

Now, we can apply Gauss-Hermite quadrature to compute this integral.

Performance on the test set using predictive distribution:
Accuracy: 99%
Average Predictive Likelihood: 0.974324189967849

(b) [3 points] Implement Laplace approximation with the diagonal Hessian. Report the approximate posterior distribution of the feature weights, the prediction accuracy and average predictive likelihood.

Answer: It means that we assume $q(\mathbf{w}) = \prod_{i=1}^{d} q(w_i)$. In this case, for each $i \in [d]$, we want to approximate

$$\log q(w_i) \approx \log q(\mu_i) - \frac{1}{2}\sigma_i^{-2}(w - \mu_i).$$

Using the same approach as in class was done, we obtain that the μ_0 would be the same as previous part and the $\mathbf{\Sigma}_0$ is a diagonal matrix whose digonal entries are the same as prevous part, i.e.,

$$\boldsymbol{\mu}_0 = [2.85594016, -2.69321758, -1.59105678, -1.8992619, -0.17689812]^\top$$

and

$$\mathbf{\Sigma}_0 = \begin{bmatrix} 0.13679752 & 0 & 0 & 0 & 0 \\ 0 & 0.11964241 & 0 & 0 & 0 \\ 0 & 0 & 0.05011859 & 0 & 0 \\ 0 & 0 & 0 & 0.06217003 & 0 \\ 0 & 0 & 0 & 0 & 0.0382288 \end{bmatrix}.$$

Performance on the test set using predictive distribution:
Accuracy: 99%
Average Predictive Likelihood: 0.9139297763783802

(c) [20 points] Implement variational logistic regression we introduced in the class. Use EM-style updates. Report the variational posterior of the feature weight vector you obtained (i.e., a multivariate Gaussian). Report the prediction accuracy and average predictive likelihood.
Answer: We say that the E-updates are:

$$\mathbf{m}_N = S_N \left(S_0^{-1} m_0 + \sum_{n=1}^{N} (t_n - \frac{1}{2} \right) \phi_n$$

and

$$\mathbf{s}_N = s_0^{-1} + 2\sum_{n=1}^{n} \lambda(\xi_n)\phi_n \boldsymbol{\phi}_n^\top.$$

Also, the M-step is:

$$(\xi_n^{\text{new}})^2 = \boldsymbol{\phi}_n^\top \mathbb{E}(S_N + \mathbf{m}_N \mathbf{m}_N^\top)\boldsymbol{\phi}_n.$$

Implementing these updates, we obtain

$$\boldsymbol{\mu} = [2.89882616, -2.77086021, -1.63751078, -1.95377589, -0.19364377]$$

and

$$\mathbf{\Sigma}_0 = \begin{bmatrix} 0.0287931 & -0.00270207 & -0.00196961 & -0.0041901 & 0.00504233 \\ -0.00270207 & 0.00360754 & -0.00021745 & 0.00078054 & -0.001883 \\ -0.00196961 & -0.00021745 & 0.00189673 & 0.00157568 & 0.00167853 \\ -0.0041901 & 0.00078054 & 0.00157568 & 0.0022857 & 0.00045613 \\ 0.00504233 & -0.001883 & 0.00167853 & 0.00045613 & 0.00640488 \end{bmatrix}.$$

Accuracy: 0.99%
Predictive Likelihood: 0.9770066184539451

(d) [15 points] Implement variational logistic regression we introduced in the class. But this time, you will use the fully factorized posterior, $q(\mathbf{w}) = \prod_i q(w_i)$ where w_i is i-th element in the weight vector $\mathbf{w}$. In the E step, please use the standard mean-field update to alternatively optimize each $q(w_i)$ given all the others fixed. Report your variational posterior (i.e., diagonal

Gaussian), the prediction accuracy and average predictive likelihood on the test data.
Answer: We already know that (see page 499 of the book: Equations 10.149 - 10.151)

$$
\begin{aligned}
p(\mathbf{t}, \mathbf{w}) &= p(\mathbf{w})p(\mathbf{t}|\mathbf{w}) \\
&\geq p(\mathbf{w}) \underbrace{\prod_{n=1}^{N} \sigma(\xi_n) \exp\left\{\mathbf{w}^\top \boldsymbol{\phi}_n t_n - (\mathbf{w}^\top \boldsymbol{\phi}_n + \xi_n)/2 - \lambda(\xi_n)\left([w^\top \boldsymbol{\phi}_n]^2 - \xi_n^2\right)\right\}}_{=h(\mathbf{w},\xi)}
\end{aligned}
$$

and thus

$$
\begin{aligned}
\log p(\mathbf{t}, \mathbf{w}) &= \log p(\mathbf{w})p(\mathbf{t}|\mathbf{w}) \\
&\geq \log p(\mathbf{w})h(\mathbf{w}, \boldsymbol{\xi}) \\
&= -\frac{1}{2}\mathbf{w}^\top \mathbf{w} + \sum_{n=1}^{N} \Big(\log \sigma(\xi_n) + \\
&\quad \mathbf{w}^\top \boldsymbol{\phi}_n t_n - (\mathbf{w}^\top \boldsymbol{\phi}_n + \xi_n)/2 - \lambda(\xi_n)\left([w^\top \boldsymbol{\phi}_n]^2 - \xi_n^2\right) \Big). \qquad (4)
\end{aligned}
$$

In Variotional approximation, we want to solve

$$
\max_{q(\mathbf{w}),\boldsymbol{\xi}} \overbrace{\mathbb{E}_{q(\mathbf{w})} \log \left\{ \frac{p(\mathbf{w})h(\mathbf{w}, \boldsymbol{\xi})}{q(\mathbf{w})} \right\}}^{=\mathcal{L}(\mathbf{q},\boldsymbol{\xi})}. \qquad (5)
$$

We have the assumption that the posterior is the fully factorized $\mathbf{q}(\mathbf{w}) = \prod_i q(w_i)$. Plug this assumption in omptimization 5 and using EM-algorithm, trying to find the posterior $q(w_i)$, in the E-step, we maximize the above objective function $\mathcal{L}(\mathbf{q}, \boldsymbol{\xi})$ with respect to each $q(w_i)$ in turn assuming that the other factors $q_j(w_j)$ (for $j \neq i$) and $\boldsymbol{\xi}$ are fixed. In this case,

$$
\begin{aligned}
\mathcal{L}(q_i) &= \int \mathbf{q}(\mathbf{w}) \log \left\{ \frac{p(\mathbf{w})h(\mathbf{w}, \boldsymbol{\xi})}{q(\mathbf{w})} \right\} \mathrm{d}\mathbf{w} \\
&= \int q(w_i) \Big\{ \underbrace{\int \prod_{j \neq i} q(w_j) \log p(\mathbf{w})h(\mathbf{w}, \boldsymbol{\xi}) \mathrm{d}\mathbf{w}_{\neg i}}_{=\mathbb{E}_{\prod_{j\neq i} q(w_j)}[\log p(\mathbf{w})h(\mathbf{w},\boldsymbol{\xi})]} \Big\} \mathrm{d}w_i - \int q(w_i) \log q(w_i) \mathrm{d}w_i + \text{Const.} \\
&= \int q(w_i) \underbrace{\mathbb{E}_{\prod_{j\neq i} q(w_j)} \left[\log p(\mathbf{w})h(\mathbf{w}, \boldsymbol{\xi})\right]}_{=\log \tilde{g}(w_i,\boldsymbol{\xi})+C} \mathrm{d}w_i - \int q(w_i) \log q(w_i) \mathrm{d}w_i + \text{Const} \\
&= -\mathrm{KL}\big(q(w_i) \| \tilde{g}(w_i, \boldsymbol{\xi})\big) + \text{Const.} \qquad (6)
\end{aligned}
$$

Note that, for a fixed $\boldsymbol{\xi}$, we defined a new distribution $\tilde{g}(w_i, \boldsymbol{\xi})$ so that

$$
\log \tilde{g}(w_i, \boldsymbol{\xi}) = \mathbb{E}_{\prod_{j\neq i} q(w_j)} \left[\log p(\mathbf{w})h(\mathbf{w}, \boldsymbol{\xi})\right] + \text{Const.}
$$

By Equation 6, maximizing $\mathcal{L}(q_i)$, we are indeed minimizing Kullback-Leibler divergence $\mathrm{KL}\big(q(w_i) \| \tilde{g}(w_i, \boldsymbol{\xi})\big)$ which is simply done by setting

$$
q(w_i) = \tilde{g}(w_i, \boldsymbol{\xi}) \propto \exp\left\{ \mathbb{E}_{\prod_{j\neq i} q(w_j)} \left[\log p(\mathbf{w})h(\mathbf{w}, \boldsymbol{\xi})\right] \right\}.
$$

So, we need to compute

$$
\mathbb{E}_{\prod\limits_{j\neq i} q(w_j)} \left[\log p(\mathbf{w})h(\mathbf{w}, \boldsymbol{\xi})\right].
$$

Using 4, we obtain

$$
\begin{aligned}
\mathbb{E}_{\prod\limits_{j\neq i} q(w_j)}\left[\log p(\mathbf{w})h(\mathbf{w},\boldsymbol{\xi})\right] &= -\frac{1}{2}\mathbb{E}\left(\mathbf{w}^\top\mathbf{w}\right) + \sum_{n=1}^{N}\Big\{\log\sigma(\xi_n) + \mathbb{E}\left(\mathbf{w}^\top\right)\boldsymbol{\phi}_n t_n \\
&\quad - \Big(\mathbb{E}\left(\mathbf{w}^\top\right)\boldsymbol{\phi}_n + \xi_n\Big)/2 - \lambda(\xi_n)\left(\mathbb{E}\left([\mathbf{w}^\top\boldsymbol{\phi}_n]^2\right) - \xi_n^2\right)\Big\} \\
&= -\frac{1}{2}w_i^2 + \sum_{n=1}^{N}\left\{\phi_i^{(n)}(t_n - \frac{1}{2})w_i - \lambda(\xi_n)\Big((w_i\phi_i^{(n)})^2 + 2w_i\phi_i^{(n)}\sum_{j\neq i}\mu_j\phi_j^{(n)}\Big)\right\} + C \\
&= -\frac{1}{2}\left(1 + 2\sum_{n=1}^{N}\lambda(\xi_n)(\phi_i^{(n)})^2\right)w_i^2 \\
&\quad + \sum_{n=1}^{N}\phi_i^{(n)}\left(t_n - \frac{1}{2} - 2\lambda(\xi_n)\sum_{j\neq i}\mu_j\phi_j^{(n)}\right)w_i + C.
\end{aligned}
$$

Therefore, in **M-step**, we do the following updates (for $i = 1,\ldots,d$),

$$
(\sigma_i^{\text{new}})^2 = \frac{1}{1 + 2\sum\limits_{n=1}^{N}\lambda(\xi_n)(\phi_i^{(n)})^2}
$$

and

$$
\mu_i^{\text{new}} = (\sigma_i^{\text{new}})^2\sum_{n=1}^{N}\phi_i^{(n)}\left(t_n - \frac{1}{2} - 2\lambda(\xi_n)\sum_{j\neq i}\mu_j\phi_j^{(n)}\right).
$$

In the M-step, we consider $\mathbf{w}$ fixed and then solve Optimization 5 for $\boldsymbol{\xi}$.

$$
\begin{aligned}
\mathcal{L}(\boldsymbol{\xi}) &= \mathbb{E}_{\mathbf{q}(\mathbf{w})}\log\left\{\frac{p(\mathbf{w})h(\mathbf{w},\boldsymbol{\xi})}{\mathbf{q}(\mathbf{w})}\right\} \\
&= -\frac{1}{2}\mathbb{E}[\mathbf{w}]^\top\mathbb{E}[\mathbf{w}] + \sum_{n=1}^{N}\Big(\log\sigma(\xi_n) + \mathbb{E}[\mathbf{w}]^\top\boldsymbol{\phi}_n t_n - (\mathbb{E}[\mathbf{w}]^\top\boldsymbol{\phi}_n + \xi_n)/2 - \lambda(\xi_n)\left(\mathbb{E}[\mathbf{w}^\top\boldsymbol{\phi}_n]^2 - \xi_n^2\right)\Big) \\
&= \sum_{n=1}^{N}\Big(\log\sigma(\xi_n) - \xi_n/2 - \lambda(\xi_n)\left(\mathbb{E}[\mathbf{w}^\top\boldsymbol{\phi}_n]^2 - \xi_n^2\right)\Big) + \text{Const.}
\end{aligned}
$$

Setting the derivative of $\mathcal{L}(\boldsymbol{\xi})$ to zero, we obtain the M-step update rule as follows:

$$
\begin{aligned}
\xi_n^2 &= \mathbb{E}[\mathbf{w}^\top\boldsymbol{\phi}_n]^2 \\
&= \text{var}(\mathbf{w}^\top\boldsymbol{\phi}_n) + \mathbb{E}^2[\mathbf{w}^\top\boldsymbol{\phi}_n] \\
&= \sum_{i=1}^{d}\text{var}(\mathbf{w}_i)\left(\boldsymbol{\phi}_i^{(n)}\right)^2 + \left(\sum_{i=1}^{d}\mathbb{E}(w_i)\boldsymbol{\phi}_i^{(n)}\right)^2.
\end{aligned}
$$

Using These updates, we get

$$
\boldsymbol{\mu} = [2.89537268, -2.76699462, -1.63499389, -1.94911109, -0.19317358]
$$

and

$$
\boldsymbol{\sigma}^2 = [0.01532219, 0.00247973, 0.00047451, 0.00058978, 0.00290673].
$$

Indeed, for each $i \in [d]$,

$$
q(w_i) \sim \mathcal{N}(\mu_i, \boldsymbol{\sigma}_i^2).
$$

Accuracy: 99%
Predictive Likelihood: 0.9769986161734513

(e) [5 points] Compare the results of the above four approximations. What do you observe and conclude?
Answer: Although all the above methods perform almost the same based on test (predictive) accuracy, the predictive likelihood for the variational inference surpasses the other methods. This indicates that by using variational approximation methods we get more close to the true posterior distribution.

3. [30 points] Gaussian Mixture Model (GMM). Please download the data "data/faithful/faithful.txt" from Canvas. Each row is a sample, including 2 features. Please normalize the features in each column to be in [-1, 1]. Specifically, denote the column by $\mathbf{x}$; then we compute for each $x_i \leftarrow (x_i - \text{mean}(\mathbf{x}))/(\max(\mathbf{x}) - \min(\mathbf{x}))$.

(a) [20 points] Implement EM algorithm for GMM. Set the number of clusters to 2. Initialize the cluster centers to be [-1, 1] and [1,-1], and the covariance matrix to be $0.1 \cdot \mathbf{I}$ for both clusters. Run your EM algorithm for 100 iterations. For iteration 1, 2, 5, 100, please draw the figures showing the corresponding cluster centers and memberships. Specifically, for each figure, first draw the scatter plots of the data points and your cluster centers. Each data point is assigned to the cluster that has a great posterior probability to include that data point. Please draw the cluster memberships with different colors.
Answer: See Figure 2.

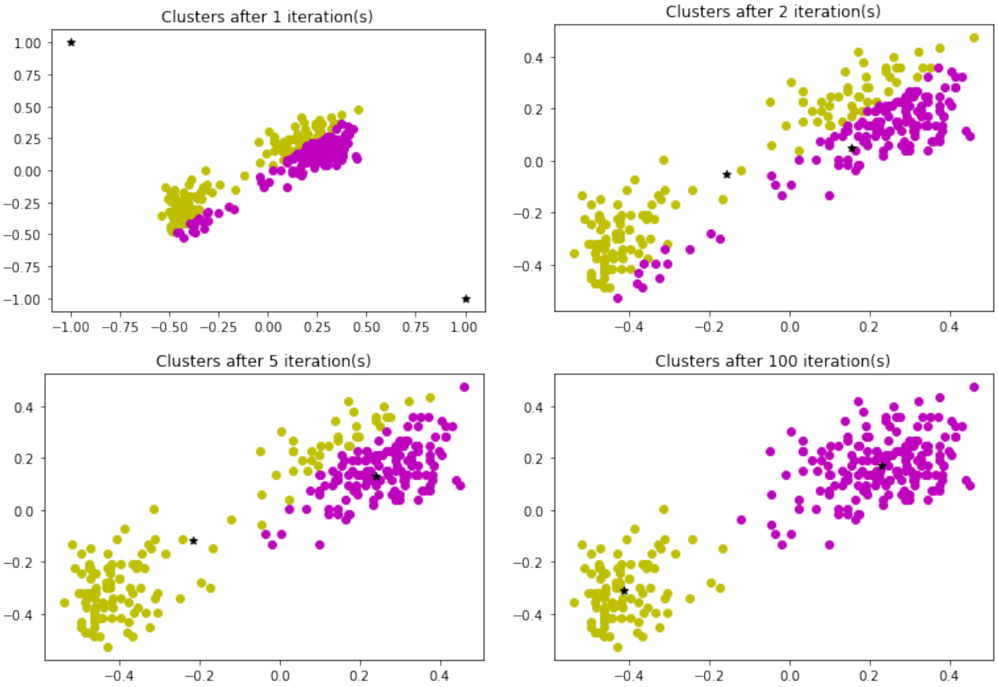

Figure 2: Clusters with starting the centers to be $(-1, 1)$ and $(1, -1)$

(b) [7 points] Now initialize the cluster centers to be [-1, -1] and [1, 1] and covariance matrix to be $0.5 \cdot \mathbf{I}$ for both clusters. Run your EM algorithm for 100 iterations. Draw the figures showing the cluster centers and memberships for iteration 1, 2, 5, 100.
Answer: See Figure 3.

(c) [3 points] Compare the results in (a) and (b), what do you observe and conclude?**Answer:** We can clearly see that the convergence when we initiate the centers as [-1, -1] and [1, 1] is much faster. We can see that the choice [-1, -1] and [1, 1] is more close to the true centers while the

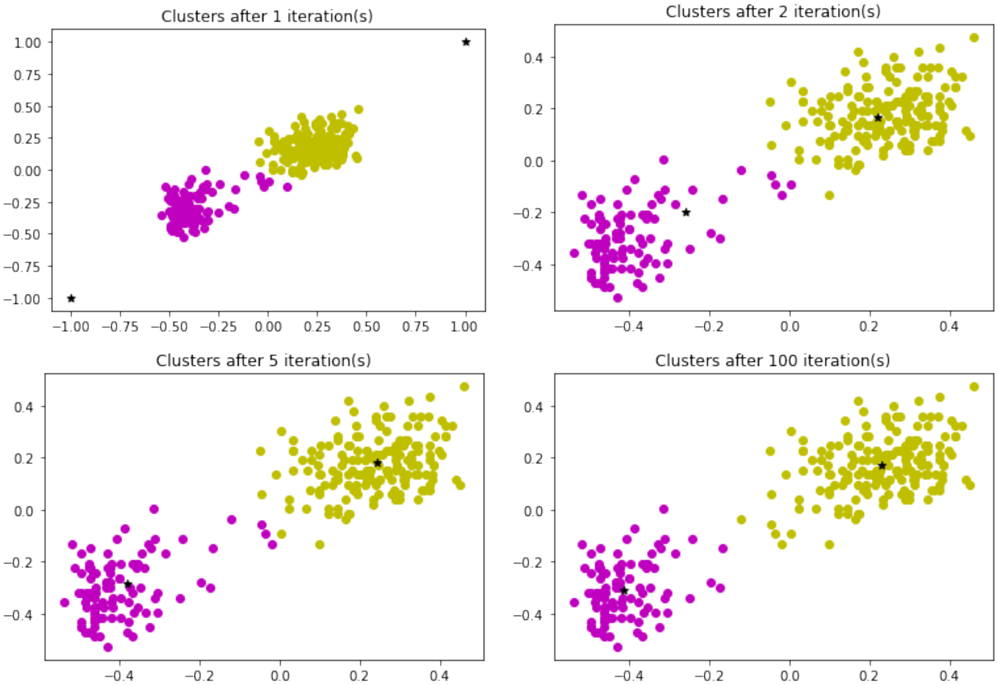

Figure 3: Clusters with starting the centers to be $(-1, -1)$ and $(1, 1)$

points [-1, 1] and [1,-1] are somehow symmetric for all the points so they cannot distinguish between clusters truly.

4. [100 points][**Bonus**] Latent Dirichlet Allocation (LDA). Please download the pre-processed corpus from "data/lda". From "ap.txt", you can see the original Associated Press corpus. "ap.dat" are the processed data which you will work on. Each row in "ap.dat" represents a document. In total, we have $2,246$ documents. The first number in each row is the number of words in the document. The following are a list of **word-id**:**count** items. Word-id starts with 0. The corresponding word list is given in "vocab.txt". The first row corresponds to Word-id 0, second, Word-id 1, and continue.

   (a) [70 points] Implement the mean-field variational EM algorithm for LDA inference as we discussed in the class. Following the orignal LDA paper (`http://www.cs.columbia.edu/~blei/papers/BleiNgJordan2003.pdf`) to implement the perplexity calculation on test documents (Sec. 7.1). Please randomly select 10% documents as the test set, and run your LDA inference algorithm on the remaining 90% documents. Vary the number of topics from {5, 10, 15, 20, 50, 100, 200}. Run your algorithm until convergence or 500 iterations have achieved. Draw a figure to show how the perplexity vary along with the growth of the topic numbers. What do you observe and conclude?

   (b) [30 points] Set the number of topics to 20 and run your variational inference algorithm. Examine the top 15 words (i.e., with the largest probability) in each learned topic distribution. List a few topics which you think is semantically meaningful and explain why.