Assignment 3 for #70240413
"Statistical Machine Learning"

Instructor: Prof. Jun Zhu

May 29, 2017

1 Preparation

- Case if you are not familiar with python, please scan the python tutorial to get started with the language: `https://docs.python.org/2.7/tutorial/`
- Case if you are not familiar with numpy (scientific computation in python), please scan the numpy tutorial: `https://docs.scipy.org/doc/numpy-dev/user/quickstart.html`
- Case if you are not familiar with Tensorflow, learn it by these tutorials:
  - `https://www.tensorflow.org/get_started/mnist/beginners`
  - `https://www.tensorflow.org/get_started/mnist/pros`
  - `https://www.tensorflow.org/get_started/mnist/mechanics`
  - `https://www.tensorflow.org/programmers_guide/dims_types`
  - `https://www.tensorflow.org/programmers_guide/variables`
  - `https://www.tensorflow.org/programmers_guide/variable_scope`
- To get started with ZhuSuan, follow this tutorial on variational autoencoders: `http://zhusuan.readthedocs.io/en/latest/vae.html`. Then learn the basic concepts through `http://zhusuan.readthedocs.io/en/latest/concepts.html`

2 Variational inference for 2-D Gaussian Mixture

2.1 Model

Here is the generative model

$$
\begin{aligned}
Z^{(i)} &\sim \text{Discrete}(\boldsymbol{\pi}) && (1)\\
\boldsymbol{X}^{(i)}|Z^{(i)} &\sim \mathcal{N}(\boldsymbol{\mu}_{Z^{(i)}}, diag\{\boldsymbol{\sigma}^2_{Z^{(i)}}\}) && (2)
\end{aligned}
$$

where Z and $\boldsymbol{X}$ are random variables, $\{\boldsymbol{x}^{(i)}\}_{i=1}^{N}$ are N observations of $\boldsymbol{X}$. Discrete($\cdot$) is a discrete distribution on $\{1, 2, \cdots, K\}$ and satisfies $\mathbb{P}(Z = j) = \pi_j$. $p(\boldsymbol{X}|Z)$ is a multivariate Gaussian with a diagonal covariance structure where the diagonal of the covariance matrix is $\boldsymbol{\sigma}^2_Z$. $\boldsymbol{\pi} \in \mathbb{R}^K, \boldsymbol{\mu}_j \in \mathbb{R}^D, \boldsymbol{\sigma}^2_j \in \mathbb{R}^D, j \in \{1, 2, \cdots, K\}$ are the parameters to be learned. N, K and D are non-negative integer constants. In the problem we fix $D = 2$. You need to use variational inference (for latent variable Z) to get the maximum likelihood estimations of $\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\sigma}^2$.

Hint: Just follow the tutorial of VAE. Use one-hot representations for Z (zs.OnehotDiscrete). Replace $q(z|x)$ with a discrete variational posterior and zs.sgvb with zs.nvil to deal with discrete latent variables.

2.2 Requirements

Your code need to include:

1. Implement the generative model with ZhuSuan and generate data using your model. (You can refer to the tutorial of VAE in ZhuSuan.)
2. Design appropriate variational posterior distribution for your model with ZhuSuan.
3. Implement the whole algorithm (model-inference-learning) with ZhuSuan. The training data is the generated data in [1]. (Recommended inference algorithm: zs.nvil)
4. Visualization of your clusters learned. (E.g., you can use matplotlib.)

Your report need to include:

1. Write the variational lower bound (for latent variable Z). Can we use zs.sgvb (The algorithm introduced in VAE) to estimate the gradient?
2. A detailed description of the problem using model-inference-learning architecture, especially the design of the variational posterior distribution.
3. Fix $N = 100$, $K = 3$, generate data and finish training. Show your results numerically and visually. Compare the learned clusters and the ground truth.

*4. Investigate how N, K and the true values of parameters influence your results. (E.g., the distances of $\boldsymbol{\mu}_j$'s) [1]

[1] Problems marked * is optional with bonus.

Hint: About the data generation: First fix N, D, K, then choose $\boldsymbol{\pi} \in \mathbb{R}^K, \boldsymbol{\mu}_j \in \mathbb{R}^D, \boldsymbol{\sigma}_j^2 \in \mathbb{R}^D$ randomly or manually. Do inference using the generated $\{\boldsymbol{x}^{(i)}\}_{i=1}^N$ and "forgetting" the true values of parameters. Finally compare the MLE and true values of parameters.

3 Gaussian Mixture VAE

Consider the combination of Gaussian Mixture model and VAE, we get the generative model

$$\begin{aligned} Z^{(i)} &\sim \text{Discrete}(\boldsymbol{\pi}) & (3)\\ \boldsymbol{H}^{(i)}|Z^{(i)} &\sim \mathcal{N}(\boldsymbol{\mu}_{Z^{(i)}}, diag\{\boldsymbol{\sigma}^2_{Z^{(i)}}\}) & (4)\\ \boldsymbol{X}^{(i)}|\boldsymbol{H}^{(i)} &\sim \text{Bernoulli}(\boldsymbol{f}_{NN}(\boldsymbol{H}^{(i)})) & (5) \end{aligned}$$

where Z, $\boldsymbol{H}$ and $\boldsymbol{X}$ are random variables, $\{x^{(i)}\}_{i=1}^N$ are the observations of $\boldsymbol{X}$. Discrete($\cdot$) is a discrete distribution on $\{1, 2, \cdots, K\}$ and satisfies $\mathbb{P}(Z = j) = \pi_j$. $\boldsymbol{H} \in \mathbb{R}^D$, $\boldsymbol{X} \in \{0,1\}^{784}$. $\boldsymbol{f}_{NN}(\boldsymbol{H})$ is a mapping (neural network) from $\mathbb{R}^D$ to $[0,1]^{784}$. $\boldsymbol{\pi} \in \mathbb{R}^K, \boldsymbol{\mu}_j \in \mathbb{R}^D, \boldsymbol{\sigma}_j^2 \in \mathbb{R}^D$, $j \in \{1, 2, \cdots, K\}$ and the parameters in $\boldsymbol{f}_{NN}(\cdot)$ are the parameters to be learned. N, K and D are nonnegative integer constants.

In the problem we fix $D = 40$, $K = 10$. The dataset is MNIST (See examples/tutorials/vae.py for reference.) You need to compute the maximum likelihood estimations of $\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\sigma}^2$ and the parameters in $\boldsymbol{f}_{NN}(\cdot)$.

3.1 Requirements

Your code need to include:

1. Implement the model with ZhuSuan.

2. There are many ways to do inference for this model. One method is given in the hint below. You can follow the hint or design your own inference algorithm.
   Hint: First sum over Z in the joint likelihood to get a joint likelihood with only $\boldsymbol{H}$ and X

   $$p(X, \boldsymbol{H}) = \sum_Z p(\boldsymbol{X}, \boldsymbol{H}, Z) = p(\boldsymbol{X}|\boldsymbol{H}) \sum_Z p(\boldsymbol{H}|Z)p(Z)$$

   Then write the variational lower bound for $\boldsymbol{H}$ and apply zs.sgvb.

3. Implement the whole algorithm (model-inference-learning) with ZhuSuan. The training data is MNIST.

4. Visualization of your results. Plot some samples: draw z from the discrete distribution on $\{1, 2, \cdots, K\}$, generate multiple $\boldsymbol{x}^{(i)}$'s for each z using your generative model and observe the clustering results.

Your report need to include:

1. A detailed description of the problem using model-inference-learning architecture.

2. Show and analyze your results numerically and visually.

*3. You may find that only several clusters represents meaningful images and some of them are composed of images from different classes. To improve the generative model, try adding some labeled data to it. For instance, choose 10 images from each class and label them with a shared cluster assignments (Z). Add the learning loss of the labeled data to the original unsupervised loss and train the model. See what's the difference.

*4. Can you get pure clusters (each cluster contains almost a single class of images) without the use of labeled data? Think about some regularizers.