ECE ILLINOIS
Department of Electrical and Computer Engineering

Assignment 3: Generative Models

Corresponding TA: Raymond Yeh

Overview

Release Date: 10/19/17

Due Date: 11/09/17

In this assignment you will:

- Derive and learn the concepts on expectation maximization (EM) and the k-means algorithm.
- Implement GMM train with k-means or EM.
- Implement semi-supervised learning algorithm with k-means on hand-written digits.
- Implement VAE using TensorFlow to generate hand-written digits.

Written Section

Problem 1. Given a training dataset $\mathcal{D} = \{(x_1), (x_2), \dots\}$. Let r_{ik} be the indicator of whether i^{th} example is in k^{th} cluster, and let μ_k be the k^{th} cluster center.

- a) What cost function does kMeans optimize?
- b) Assume that r_{ik} are fixed, show that $\mu_k = \frac{1}{\sum_{i \in D} r_{ik}} \sum_{i \in D} r_{ik} x_i$ minimized the cost function you wrote in a).
- c) Write out the psuedo code of the kMeans algorithm.
- d) Show that at each update step of the kMeans algorithm, the cost function you wrote in a) decreases monotonically.

Problem 2. Recall that Jensen's Inequality states: For a real function f, numbers $x_1, \dots x_n$ in its domain, and positive weights a_i and $\sum_{i=1}^{n} a_i = 1$ then

$$f(\sum_{i=1}^{n} a_i x_i) \leq \sum_{i=1}^{n} a_i f(x_i)$$

- a) Prove Jensen's Inequliaty for the case of $n = 2$ using definition of convexity.
- b) Prove Jensen's Inequliaty for the case of n, using mathematical induction, and result from a)

Programming Section

Part 1: Setup

- First remote connect to a ews machine.

```
ssh (netid)@remlnx.ews.illinois.edu
```

You may need vpn if you are not on a campus network.

- Load python module, this will also load pip and virtualenv

```
module load python/3.4.3
```

- Reuse the virtual environment from mp1.

```
source ~/ece544na_fa2017/bin/activate
```

- Copy mp3 into your svn directory, and change directory to mp3.

```
svn cp https://subversion.ews.illinois.edu/svn/fa17-ece544/_shared/mp3 .
cd mp3
```

- Create data directory and Download the data into the data directory.

```
mkdir data
wget --user (netid) --ask-password
https://courses.engr.illinois.edu/ece544na/fa2017/secure/assignment3_data.zip -O data/assignment3_data.zip
```

- Unzip assignment3_data.zip

```
unzip assignment1_data.zip -d .
```

- Prevent svn from checking in the data directory.

```
svn propset svn:ignore data .
```

- Follow the same steps as mp1 to checkout and setup for mp3 and install the requirements.

Part 2: Exercise 1: Semi-supervised learning with Gaussian Mixture Model.

In this exercise you will build a system to classify between hand-written digits (0 to 9), using labeled and unlabeled data. The main motivation for this approach is that labeled data tends to be expensive to collect.

To use the unlabeled data, we first formulate the problem as a clustering problem. (i.e. arrange the data into k groups according to some distance metric). In the ideal case, the distance for the examples with the same label should be small, then we will only need one labeled example for each cluster.

In `main.py`, the overall program structure is provided for you.

Part 2.1 Reading in data

The data has been provided to you in csv format.

Each row is a hand-written digit, the first element is the label (0-9), and the remaining elements are the pixel intensities; The image is of size 28x28 in gray scale. For the unlabeled data, a dummy label of -1 is included.

Revelant Files: `utils/io_tools.py`,

~~Part 2.2 Gaussian Mixture Model~~

~~Review Lecture 14 for GMM formulation, and more details are provided in the function docstring.~~

Revelant Files:

~~`models/gaussian_mixture_model.py`~~

Part 2.3 Optimization (K-means)

Review Lecture 13 for K-means formulation, and more details are provided in the function docstring.

Revelant Files: `models/kmeans.py`

Part 2.3 Semi-Supervised Learning

For the supervised learning part, we simply count! We assign labeled examples to each cluster which maximizes the posterior probability. Then, we count how many times a digit shows up in each of the cluster. (e.g if the most frequency digit in cluster 1, is digit ‘5’, then our model will predict that any example assigned to cluster 1 to have the label of digit ‘5’).

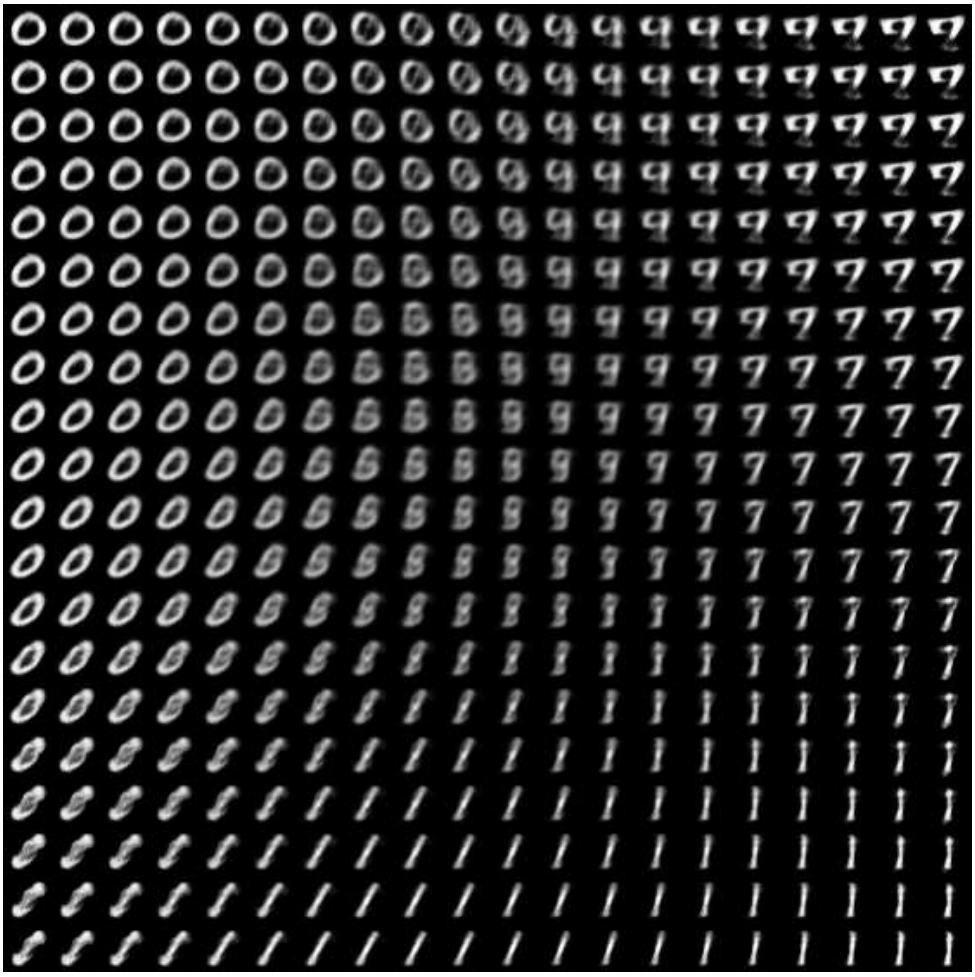
Revelant Files: `models/kmeans.py`, ~~`models/gaussian_mixture_model.py`~~

Part 3: Exercise 2: Unsupervised learning with Variational AutoEncoder

In this exercise, we will learn to generate examples from the data distribution using VAE. We have provided the training loop, visualization, and data reader for you.

Review Lecture 17 for formulation on Variational AutoEncoder. More details are provided in the function docstring.

For the ease of visualization, we have chosen the latent space to be 2 dimension, The visualized results will be something similar to the figure below.



Revelant Files: `models/variational_autoencoder.py`, `main_tf.py`

Part 4: Submit

Submitting your code is simply commiting you code. This can be done with the follow command:

```
svn commit -m "Some meaningful comment here."
```

Note: The assignment will be autograded. It is important that you do not use additional libraries, or change the provided functions input and output.