AIML-15: Machine Learning, Homework 3. K-Means, GMM+EM, and Clustering Evaluation

December 1, 2015

General information. Problem solutions should be submitted in PDF format in report style (no source code listings required). All reports must be submitted before December 20 to the moodle (elearning) system. It is advised to use Python as a programming language, but you can use any language of your choice (at your own risk). In case you use Python, free `Anaconda` distribution comes with all needed packages:

`https://www.continuum.io/downloads`

In particular, you might find useful `scikit-learn` general machine learning library and `matplotlib` plotting facilities. When in doubt, read the manual and take a look at the large set of examples:

`http://scikit-learn.org/stable/documentation.html`

`http://scikit-learn.org/stable/auto_examples/`

`http://matplotlib.org/examples/`

Data preparation. In this homework you will work with the `MNIST` [1] dataset composed from 10 classes of handwritten digits. The dataset contains ≈ 70000, 28×28 images. Steps:

- If you are using Python and `scikit-learn`, you can get training data by running:

  ```
  from sklearn.datasets import fetch_mldata
  mnist = fetch_mldata('MNIST_original')
  X = mnist.data     # 70000 by 784 matrix of instances
  y = mnist.target   # 70000 vector of labels
  ```

  This might take some time when you execute it for the first time, because this command will download the dataset.

- If you are using Matlab, you can use `http://www.cs.nyu.edu/~roweis/data/mnist_all.mat`.

- Otherwise, in binary format from `http://yann.lecun.com/exdb/mnist/`.

Clustering with K-Means.

- Select the subset of $\boldsymbol{X}$ and $\boldsymbol{y}$, which belongs only to classes $\{0, 1, 2, 3, 4\}$, with 200 examples per class.
- Cluster $\boldsymbol{X}$ using K-Means into 5 clusters. An example of clustering in Python would be,

```
from sklearn.cluster import KMeans
clusterer = KMeans(5, 'random')
clusterer.fit(X)
```

- Plot obtained cluster centroids as images. You can use `pylab.imshow()`.
- Repeat clustering and visualization for 3 clusters and 10 clusters. Which characteristic of the data is captured by the centroids?

Clustering with GMM/EM, and Performance Evaluation.

- Cluster $\boldsymbol{X}$ multiple times using GMM when number of clusters varies is in $\{2, 3, \ldots, 10\}$. An example of clustering in Python would be,

```
from sklearn.mixture import GMM
clusterer = GMM(5, 'diag')
clusterer.fit(X)
cluster_labels = clusterer.predict(X)
```

- Compute cluster purity for every choice of number of clusters, and plot number of clusters against the purity.
- Explain your observation.

Classifying with GMM/EM. In this assignment you're asked to construct a classifier through *generative* modeling, that is, every class will be modeled by a mixture of Gaussians. Guidelines:

- For every class in $\boldsymbol{X}$, fit a GMM model as in previous problem.
- For every point in the testing set, obtain the log-likelihood of that a point, and decide the label as index of the GMM with the highest log-likelihood. In Python, you can use function `clusterer.score_samples()`. In other words, let the prediction rule, given testing points $\boldsymbol{x}$, be:

$$f(\boldsymbol{x}) = \underset{y \in \{1,\ldots,N\}}{\arg\max} \log\left(p(\boldsymbol{x} \mid \bar{\Sigma}_y, \bar{\mu}_y, \bar{\theta}_y)\right) \ ,$$

  where N is the number of classes, and y-th mixture model is characterized by the parameters,

$$\bar{\Sigma}_y = \{\boldsymbol{\Sigma}_{y,1}, \ldots, \boldsymbol{\Sigma}_{y,K}\} \ , \qquad \text{(Covariances)}$$
$$\bar{\mu}_y = \{\boldsymbol{\mu}_{y,1}, \ldots, \boldsymbol{\mu}_{y,K}\} \ , \qquad \text{(Means)}$$
$$\bar{\theta}_y = \{\theta_{y,1}, \ldots, \theta_{y,K}\} \ . \qquad \text{(Mixture weights)}$$

- Report performance of this classifier on the testing set, varying number of components in each mixture in $\{2, 3, 4, 5\}$.
- (Optional) Select number of components K using validation set or cross-validation.
- (Optional) Compare to non-linear SVM.

References

[1] Y. LeCun, C. Cortes, and C. JC Burges. The mnist database of handwritten digits, 1998.