ISYE 6740, Spring 2021, Homework 4

100 points + 10 bonus points

Prof. Yao Xie

1. Implementing EM for MNIST dataset (60 points).

Implement the EM algorithm for fitting a Gaussian mixture model for the MNIST handwritten digits dataset. For this question, we reduce the dataset to be only two cases, of digits "2" and "6" only. Thus, you will fit GMM with $C = 2$. Use the data file data.mat or data.dat. True label of the data are also provided in label.mat and label.dat.

The matrix images is of size 784-by-1990, i.e., there are totally 1990 images, and each column of the matrix corresponds to one image of size 28-by-28 pixels (the image is vectorized; the original image can be recovered by map the vector into a matrix).

First use PCA to reduce the dimensionality of the data before applying to EM. We will put all "6" and "2" digits together, to project the original data into 5-dimensional vectors.

Now implement EM algorithm for the projected data (with 5-dimensions).

(a) (5 points) Select one image of "2" and one image of "6", and visualize the two images.

(b) (10 points) Write down detailed expression of the E-step and M-step in the EM algorithm (hint: when computing τ_k^i, you can drop the $(2\pi)^{n/2}$ factor from the numerator and denominator expression, since it will be canceled out; this can help avoid some numerical issues in computation).

(c) (15 points) Implement EM algorithm yourself. Use the following initialization

- initialization for mean: random Gaussian vector with zero mean
- initialization for covariance: generate two Gaussian random matrix of size n-by-n: S_1 and S_2, and initialize the covariance matrix for the two components are $\Sigma_1 = S_1 S_1^T + I_n$, and $\Sigma_2 = S_2 S_2^T + I_n$, where I_n is an identity matrix of size n-by-n.

Plot the log-likelihood function versus the number of iterations to show your algorithm is converging.

(d) (15 points) Report, the fitted GMM model when EM has terminated in your algorithms as follows. Make sure to report the weights for each component, and the mean of each component (you can reformat the vector to make them into 28-by-28 matrices and show images). Ideally, you should be able to see these means corresponds to "average" images. You can report the two 784-by-784 covariance matrices by visualizing their intensities (e.g., using a gray scaled image or heat map).

(e) (15 points) Use the τ_k^i to infer the labels of the images, and compare with the true labels. Report the mis-classification rate for digits "2" and "6" respectively. Perform K-means clustering with $K = 2$ (you may call a package or use the code from your previous homework). Find out the mis-classification rate for digits "2" and "6" respectively, and compare with GMM. Which one achieves the better performance?

2. Optimization (40 points).

Consider a simplified logistic regression problem. Given m training samples (x^i, y^i), $i = 1, \ldots, m$. The data $x^i \in \mathbb{R}$ (note that we only have one feature for each sample), and $y^i \in \{0, 1\}$. To fit a logistic regression model for classification, we solve the following optimization problem, where $\theta \in \mathbb{R}$ is a parameter we aim to find:

$$\max_{\theta} \ell(\theta), \tag{1}$$

where the log-likelihood function

$$\ell(\theta) = \sum_{i=1}^{m} \left\{ -\log(1 + \exp\{-\theta x^i\}) + (y^i - 1)\theta x^i \right\}.$$

1. (10 points) Show step-by-step mathematical derivation for the gradient of the cost function $\ell(\theta)$ in (1).

2. (10 points) Write a pseudo-code for performing **gradient descent** to find the optimizer θ^*. This is essentially what the training procedure does.

3. (10 points) Write the pseudo-code for performing the **stochastic gradient descent** algorithm to solve the training of logistic regression problem (1). Please explain the difference between gradient descent and stochastic gradient descent for training logistic regression.

4. (10 points) We will **show that the training problem in basic logistic regression problem is concave.** Derive the Hessian matrix of $\ell(\theta)$ and based on this, show the training problem (1) is concave (note that in this case, since we only have one feature, the Hessian matrix is just a scalar). Explain why the problem can be solved efficiently and gradient descent will achieve a unique global optimizer, as we discussed in class.

3. De-bias review system using EM (bonus: 10 points).

In this question, we will develop an algorithm to remove individual reviewer's bias from their score. Consider the following problem. There are P papers submitted to a machine learning conference. Each of R reviewers reads each paper, and gives it a score indicating how good he/she thought that paper was. We let $x^{(pr)}$ denote the score that reviewer r gave to paper p. A high score means the reviewer liked the paper, and represents a recommendation from that reviewer that it be accepted for the conference. A low score means the reviewer did not like the paper.

We imagine that each paper has some "intrinsic" true value that we denote by μ_p, where a large value means it's a good paper. Each reviewer is trying to estimate, based on reading the paper, what μ_p is; the score reported $x^{(pr)}$ is then reviewer r's guess of μ_p.

However, some reviewers are just generally inclined to think all papers are good and tend to give all papers high scores; other reviewers may be particularly nasty and tend to give low scores to everything. (Similarly, different reviewers may have different amounts of variance in the way they review papers, making some reviewers more consistent/reliable than others.) We let ν_r denote the "bias" of reviewer r. A reviewer with bias ν_r is one whose scores generally tend to be ν_r higher than they should be.

All sorts of different random factors influence the reviewing process, and hence we will use a model that incorporates several sources of noise. Specifically, we assume that reviewers's scores are generated by a random process given as follows:

$$y^{(pr)} \sim \mathcal{N}(\mu_p, \sigma_p^2)$$

$$z^{(pr)} \sim \mathcal{N}(\nu_r, \tau_r^2)$$

$$x^{(pr)} | y^{(pr)}, z^{(pr)} \sim \mathcal{N}(y^{(pr)} + z^{(pr)}, \sigma^2).$$

The variables $y^{(pr)}$ and $z^{(pr)}$ are independent; the variables (x, y, z) for different paper-reviewer pairs are also jointly independent. Also, we only ever observe the $x^{(pr)}$s; thus, the $y^{(pr)}$s and $z^{(pr)}$s are all latent random variables.

We would like to estimate the parameters μ_p, σ_p^2, ν_r, τ_r^2. If we obtain good estimates of the papers "intrinsic values" μ_p, these can then be used to make acceptance/rejection decisions for the conference.

We will estimate the parameters by maximizing the marginal likelihood of the data $\{x^{(pr)}; p = 1, \ldots, P, r = 1, \ldots, R\}$. This problem has latent variables $y^{(pr)}$s and $z^{(pr)}$s, and the maximum likelihood problem cannot be solved in closed form. So, we will use EM.

Your task is to derive the EM update equations. For simplicity, you need to treat only $\{\mu_p, \sigma_p^2; p = 1 \ldots, P\}$ and $\{\nu_r, \tau_r^2; r = 1...R\}$ as parameters, i.e. treat σ^2 (the conditional variance of $x^{(pr)}$ given $y^{(pr)}$ and $z^{(pr)}$) as a fixed, known constant.

1. Derive the E-step:

   (a) The joint distribution $p(y^{(pr)}, z^{(pr)}, x^{(pr)})$ has the form of a multivariate Gaussian density. Find its associated mean vector and covariance matrix in terms of the parameters μ_p, σ_p^2, ν_r, τ_r^2 and σ^2. [Hint: Recognize that $x^{(pr)}$ can be written as $x^{(pr)} = y^{(pr)} + z^{(pr)} + \epsilon^{(pr)}$, where $\epsilon^{(pr)} \sim \mathcal{N}(0, \sigma^2)$ is independent Gaussian noise.

(b) Derive an expression for $Q_{pr}(\theta'|\theta) = \mathbb{E}[\log p(y^{(pr)}, z^{(pr)}, x^{(pr)})|x^{(pr)}, \theta]$ using the conditional distribution $p(y^{(pr)}, z^{(pr)}|x^{(pr)})$ (E-step) (Hint, you may use the rules for conditioning on subsets of jointly Gaussian random variables.)

2. Derive the M-step to update the parameters μ_p, σ_p^2, ν_r, and τ_r^2. [Hint: It may help to express an approximation to the likelihood in terms of an expectation with respect to $(y^{(pr)}, z^{(pr)})$ drawn from a distribution with density $Q_{pr}(y^{(pr)}, z^{(pr)})$.]

Remark: John Platt (whose SMO algorithm you've seen) implemented a method quite similar to this one to estimate the papers' true scores. (There, the problem was a bit more complicated because not all reviewers reviewed every paper, but the essential ideas are the same.) Because the model tried to estimate and correct for reviewers' biases, its estimates of the paper's value were significantly more useful for making accept/reject decisions than the reviewers' raw scores for a paper.