CS 7641 CSE/ISYE 6740 Homework 3

Yao Xie

Deadline: Monday March 5, 11:55pm

- Submit your answers as an electronic copy on T-square.
- No unapproved extension of deadline is allowed. Zero credit will be assigned for late submissions. Email request for late submission may not be replied.
- For typed answers with LaTeX (recommended) or word processors, extra credits will be given. If you handwrite, try to be clear as much as possible. No credit may be given to unreadable handwriting.
- Explicitly mention your collaborators if any.

1 EM application [25 points].

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 rs 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. In this part, we will 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 updates to 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: In a recent machine learning conference, 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.

2 Basic optimization [10 points]

1. Show that log-sum-exp $f(x) = \log(e^{x_1} + \cdots + e^{x_n})$ is convex.

2. (Jensen's inequality.) Use the definition of a concave function f, to show that

$$f\left(\sum_{i=1}^{m} \alpha_i x_i\right) \geq \sum_{i=1}^{m} \alpha_i f(x_i)$$

   where $\sum_{i=1}^{m} \alpha_i = 1$, $\alpha_i \geq 0$.

3 Bayes Classifier [25 points]

1. Bayes Classifier With General Loss Function In class, we talked about the popular 0-1 loss function in which $L(a, b) = 1$ for $a \neq b$ and 0 otherwise, which means all wrong predictions cause equal loss. Yet, in many other cases including cancer detection, the asymmetric loss is often preferred (misdiagnosing cancer as no-cancer is much worse). In this problem, we assume to have such an asymmetric loss function where $L(a, a) = L(b, b) = 0$ and $L(a, b) = p, L(b, a) = q, p \neq q$. Write down the the Bayes classifier $f : X \to Y$ for binary classification $Y \in \{-1, +1\}$. Simplify the classification rule as much as you can. [20 pts]

2. Gaussian Class Conditional distribution

   (a) Suppose the class conditional distribution is a Gaussian. Based on the general loss function in problem 1, write the Bayes classifier as $f(X) = \text{sign}(h(X))$ and simplify h as much as possible. What is the geometric shape of the decision boundary? [10 pts]

(b) Repeat (a) but assume the two Gaussians have identical covariance matrices. What is the geometric shape of the decision boundary? [10 pts]

(c) Repeat (a) but assume now that the two Gaussians have covariance matrix which is equal to the identity matrix. What is the geometric shape of the decision boundary? [10 pts]

4 Bayes and KNN classifier [20 points]

In this programming assignment, you are going to apply the Bayes Classifier to handwritten digits classification problem. Here, we use the binary 0/1 loss for binary classification, i.e., you will calculate the miss-classification rate as a performance metric.

To ease your implementation, we selected two categories from USPS dataset in usps-2cls.mat and provide a skeleton code containing data splitting part, SplitData.m, and the main routine main.m. SplitData.m is designed to split the dataset into training set and testing set according to a predefined ratio.

1. Your first task is implementing the classifier by assuming the covariance matrix is

   (a) Full matrix, i.e., $\Sigma_1 = \Sigma_2 = \Sigma$ and Σ is a dense matrix, i.e., all entries can be non-zero;

   (b) Diagonal matrix, $\Sigma_1 = \Sigma_2 = D$, where D is a diagonal matrix and

   (c) Spherical (the diagonal has a constant value), $\Sigma_1 = \Sigma_2 = \sigma^2 I$.

   The first step you will need to do, is to estimate the mean parameter and the covariance parameters using the training data. You can use maximum likelihood estimate to estimate the parameters in each case. Then report the misclassification rate (error rate) over the training set and over the testing set averaged over the 100 random train/test splits for the each of the three covariance matrix cases by using different value of p. Explain and compare the performance of each classifier.

   The file you need to edit is ModelFull.m, ModelDiagonal.m and ModelSpherical.m provided with this homework. After implementing these methods, you should evaluate your algorithm on the given set. Repeat 100 times: split the dataset into two parts randomly, use p portion for training and the other $1 - p$ portion for testing. Let p change from 0.1, 0.2, 0.5, 0.8, 0.9.

2. Now repeat the classification again using K-nearest neighbors, for $K = 5, 10, 15, 30$. Repeat 100 times: split the dataset into two parts randomly, use p portion for training and the other $1 - p$ portion for testing. Let p change from 0.1, 0.2, 0.5, 0.8, 0.9. Report the training error and testing error for each case.

5 Logistic regression [20 points]

Repeat the binary classification task above for the same data usps-2cls.mat using logistic regression. Repeat 100 times: split the dataset into two parts randomly, use p portion for training and the other $1 - p$ portion for testing. Let p change from 0.1, 0.2, 0.5, 0.8, 0.9. Report the training error and testing error for each case. You may use the matlab built-in function for logistic regression (see class demo code as an example).