Homework 2, due Monday October 3 COMS 4771 Fall 2016

Problem 1 (Naïve Bayes; 30 points). Download the "20 Newsgroups data set" `news.mat` from Courseworks. The training feature vectors/labels and test feature vectors/labels are stored as `data/labels` and `testdata/testlabels`. Each data point corresponds to a message posted to one of 20 different newsgroups (i.e., message boards). The representation of a message is a (sparse) binary vector in $\mathcal{X} := \{0,1\}^d$ (for $d := 61188$) that indicates the words that are present in the message. If the j-th entry in the vector is 1, it means the message contains the word that is given on the j-th line of the text file `news.vocab`. The class labels are $\mathcal{Y} := \{1, 2, \ldots, 20\}$, where the mapping from classes to newsgroups is in the file `news.groups` (which we won't actually need).

In this problem, you'll develop a classifier based on a Naïve Bayes generative model. Here, we use class conditional distributions of the form $P_{\boldsymbol{\mu}}(\boldsymbol{x}) = \prod_{j=1}^{d} \mu_j^{x_j}(1-\mu_j)^{1-x_j}$ for $\boldsymbol{x} = (x_1, x_2, \ldots, x_d) \in \mathcal{X}$. Here, $\boldsymbol{\mu} = (\mu_1, \mu_2, \ldots, \mu_d) \in [0,1]^d$ is the parameter vector from the parameter space $[0,1]^d$. Since there are 20 classes, the generative model is actually parameterized by 20 such vectors, $\boldsymbol{\mu}_y = (\mu_{y,1}, \mu_{y,2}, \ldots, \mu_{y,d})$ for each $y \in \mathcal{Y}$, as well as the class prior parameters, π_y for each $y \in \mathcal{Y}$. The class prior parameters, of course, must satisfy $\pi_y \in [0,1]$ for each $y \in \mathcal{Y}$ and $\sum_{y \in \mathcal{Y}} \pi_y = 1$.

(a) Give the formula for the MLE of the parameter $\mu_{y,j}$ based on training data $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^n$. (Remember, each unlabeled point is a vector: $\boldsymbol{x}_i = (x_{i,1}, x_{i,2}, \ldots, x_{i,d}) \in \{0,1\}^d$.)

(b) MLE is not a good estimator for the class conditional parameters if the estimate turns out to be zero or one. An alternative is the following estimator based on a technique called *Laplace smoothing*: $\hat{\mu}_{y,j} := (1 + \sum_{i=1}^n \mathbb{1}\{y_i = y\} x_{i,j}) / (2 + \sum_{i=1}^n \mathbb{1}\{y_i = y\}) \in (0,1)$.

Write codes for training and testing a classifier based on the Naïve Bayes generative model described above. Use Laplace smoothing to estimate class conditional distribution parameters, and MLE for class prior parameters. You should *not* use or look at any existing implementation (e.g., such as those that may be provided as library functions). Using your codes, train and test a classifier with the data from `news.mat`. **Your codes should be easy to understand (e.g., by using sensible variable names and comments)**.

What to submit: (1) training and test error rates, (2) source code (in a separate file).

(c) Consider the *binary* classification problem, where newsgroups $\{1, 16, 20\}$ comprise the "negative class" (class 0), and newsgroups $\{17, 18, 19\}$ comprise the "positive class" (class 1). Newsgroups $\{1, 16, 20\}$ are "religious" topics, and newsgroups $\{17, 18, 19\}$ are "political" topics. Modify the data in `news.mat` to create the training and test data sets for this problem. Using these data and your codes from part (b), train and test a Naïve Bayes classifier.

What to submit: training and test error rates. Save the learned classifier for part (d)!

(d) The classifier you learn is ultimately a linear classifier, which means it has the following form:

$$\boldsymbol{x} \mapsto \begin{cases} 0 & \text{if } \alpha_0 + \sum_{j=1}^{d} \alpha_j x_j \leq 0 \\ 1 & \text{if } \alpha_0 + \sum_{j=1}^{d} \alpha_j x_j > 0 \end{cases}$$

for some real numbers $\alpha_0, \alpha_1, \ldots, \alpha_d$. Determine the values of these α_j's for your learned classifier from part (c). Then, report the vocabulary words whose indices $j \in \{1, 2, \ldots, d\}$ correspond to the 20 largest (i.e., most positive) α_j value, and also the vocabulary words whose indices $j \in \{1, 2, \ldots, d\}$ correspond to the 20 smallest (i.e., most negative) α_j value. Don't report the indices j's, but rather the actual vocabulary words (from `news.vocab`).

What to submit: two ordered list (appropriately labeled) of 20 words each.

Problem 2 (Cost-sensitive classification; 10 points)**.** Suppose you face a binary classification problem with input space $\mathcal{X} = \mathbb{R}$ and output space $\mathcal{Y} = \{0, 1\}$, where it is c times as bad to commit a "false positive" as it is to commit a "false negative" (for some real number $c \geq 1$). To make this concrete, let's say that if your classifier predicts 1 but the correct label is 0, you incur a penalty of \$$c$; if your classifier predicts 0 but the correct label is 1, you incur a penalty of \$1. (And you incur no penalty if your classifier predicts the correct label.)

Assume the distribution you care about has a class prior with $\pi_0 = 2/3$ and $\pi_1 = 1/3$, and the class conditional densities are $\mathrm{N}(0, 1)$ for class 0, and $\mathrm{N}(2, 1/4)$ for class 1. Let $f^\star \colon \mathbb{R} \to \{0, 1\}$ be the classifier with the smallest expected penalty.

(a) Assume $1 \leq c \leq 14$. Specify precisely (and with a simple expression involving c) the region in which the classifier $f^\star$ predicts 1.

(b) Now instead assume $c \geq 15$. Specify precisely the region in which the classifier $f^\star$ predicts 1.

Problem 3 (Covariance matrices; 10 points)**.** Let $\boldsymbol{X}$ be a mean-zero random vector in $\mathbb{R}^d$ (so $\mathbb{E}(\boldsymbol{X}) = \boldsymbol{0}$). Let $\boldsymbol{\Sigma} := \mathbb{E}(\boldsymbol{X}\boldsymbol{X}^\top)$ be the covariance matrix of $\boldsymbol{X}$, and suppose its eigenvalues are $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_d$. Let $\sigma > 0$ be a positive number.

(a) What are the eigenvalues of $\boldsymbol{\Sigma} + \sigma^2 \boldsymbol{I}$?

(b) What are the eigenvalues of $(\boldsymbol{\Sigma} + \sigma^2 \boldsymbol{I})^{-2}$?

In both cases, give your answers in terms of σ and the eigenvalues of $\boldsymbol{\Sigma}$.