1 Naive Bayes (15 points)

(a) Suppose you are given the data for several 3-digit passwords and the output whether they open a security gate or not. The passwords are generated by a source with following distribution. Compute the parameters (priors and conditionals) of a multinomial Naive Bayes classifier that uses the digits 0, 1, 2, 3, and 4 as features assuming smoothing rule that the computed-zero probabilities are smoothed into probability 0.01 instead. You are not required to correct P(X) + P(X') > 1 which might be caused because of the smoothing rule.

Event	Password	Opens the Gate?	Probability
1	240	no	$\frac{4}{9}$
2	343	no	$\frac{4}{9}$
3	422	yes	$\frac{1}{18}$
4	031	yes	$\frac{1}{18}$

(b) Consider the password 031, will it open the gate based on your classifier?

(c) How can we show that *Logistic Regression* is the discriminative version of *Naive Bayes*?

(d) The classifier we learn by optimizing the *Naive Bayes* conditional likelihood is nevertheless not the same as what we would have learned for a *Logistic Classifier* directly. What modeling assumption makes it somewhat less generic than *Logistic Regression*?

2 Generative Model and Discriminative Model (20 points)

Let's use mixture of Gaussian and mixture of Poisson distribution to study relation between generative model and discriminative model. Suppose a dataset $\mathcal{D} = \{(\boldsymbol{x}_1, y_1), \cdots (\boldsymbol{x}_N, y_N)\}$ is generated from a mixture of 2 Gaussian distribution components. The model is:

$$y \sim \text{Bernoulli}(\pi) \tag{1}$$

$$x|y = 0 \sim \mathcal{N}(\mu_0, \Sigma_0) \tag{2}$$

$$x|y = 1 \sim \mathcal{N}(\mu_1, \Sigma_1) \tag{3}$$

(a) Please write out the log-likelihood of distribution $p(\mathcal{D}|\pi, \mu_0, \mu_1, S_0, S_1)$, and the MLE of five parameters in this distribution. For conciseness, we use(1) $t_{n,c}$ as 0/1 indicator that sample is of class c, (2) N_c as total count of samples from class c.

(b) According to Bayesian rule, after we have estimated parameters, we can infer the posterior probability of class given observation:

$$p(y = 1|\boldsymbol{x}, \pi, \mu_0, \mu_1, S_0, S_1) = \frac{p(y = 1|\pi)p(\boldsymbol{x}|y, \mu_1, S_1)}{p(y = 1|\pi)p(\boldsymbol{x}|\mu_1, S_1) + p(y = 1|\pi)p(\boldsymbol{x}|\mu_0, S_0)} \tag{4}$$

Please prove that *iff* (if and only if) $S_0 = S_1$, the posterior distribution can be simplified into generalized linear model:

$$p(y = 1|\boldsymbol{x}, \pi, \mu_0, \mu_1, S_0, S_1) = \frac{1}{1 + \exp(-\boldsymbol{\theta}^T \boldsymbol{x})} \tag{5}$$

And please write down the expression of $\boldsymbol{\theta}$ when $S_0 = S_1$.

(c) Instead of mixture of Gaussian distribution, suppose data are generated from mixture of Poisson distributions. In this case, the observation x has discrete integer value:

$$x|y = 0 \sim \text{Poisson}(\lambda_0) \tag{6}$$

$$x|y = 1 \sim \text{Poisson}(\lambda_1) \tag{7}$$

Please prove that posterior distribution $p(y|x, \pi, \lambda_0, \lambda_1)$ can be modeled with generalized linear model.

3 Multinomial Logistic Regression (15 points)

Multiclass logistic regression has the form

$$p(y = c|\boldsymbol{x}, \boldsymbol{W}) = \frac{\exp(\omega_{c0} + \boldsymbol{\omega}_c^T \boldsymbol{x})}{\sum_{k=1}^{C} \exp(\omega_{k0} + \boldsymbol{\omega}_k^T \boldsymbol{x})} \tag{8}$$

where $\boldsymbol{W}$ is a $(D + 1) \times C$ weight matrix. We can arbitrarily define $\boldsymbol{\omega}_c = \boldsymbol{0}$ for one of the classes, say $c = C$, since $p(y = C|\boldsymbol{x}, \boldsymbol{W}) = 1 - \sum_{c=1}^{C-1} p(y = c|\boldsymbol{x}, \boldsymbol{W})$. In this case, the model has the form

$$p(y = c|\boldsymbol{x}, \boldsymbol{W}) = \frac{\exp(\boldsymbol{\omega}_{c0} + \boldsymbol{\omega}_c^T \boldsymbol{x})}{1 + \sum_{k=1}^{C-1} \exp(\boldsymbol{\omega}_{k0} + \boldsymbol{\omega}_k^T \boldsymbol{x})} \tag{9}$$

If we do not clamp one of the vectors to some constant value, the parameters will be unidentifiable. However, suppose we do not clamp $\boldsymbol{\omega}_c = 0$, so we are using Equation (8), but we add ℓ_2 regularization by optimizing

$$\sum_{n=1}^{N} \log p(y_n|\boldsymbol{x}_n, \boldsymbol{W}) - \lambda \sum_{k=1}^{C} ||\boldsymbol{\omega}_k||_2^2 \tag{10}$$

Please prove that at the optimum we have $\sum_{k=1}^{C} \hat{\omega}_{k,j} = 0$ for $j = 1 : D$. (For unregularized terms $\hat{\omega}_{k0}$, we still need to enforce that $\omega_{0,C} = \text{const}$ to ensure identifiability of the offset)

4 Programming Questions

4.1 Practical Logistic Regression on Real Data (25 points)

In this problem you will investigate the *Spambase* dataset to predict if one email is spam email or not. The data and description can be found at https://archive.ics.uci.edu/ml/datasets/Spambase.

(a) **Load Data** Load the data into MATLAB. Randomly shuffle the rows of data. Then separate it evenly into two datasets: half for training and half for testing. Separate the rightmost column from the rest and it will be the dependent class variable you are trying to predict. The remaining columns will be used as independent variables.

(b) **Logistic Regression** Train logistic regression models on the raw and on the standardized (sometimes referred to as "normalized") data. You are required to learn the same model using 3 methods: 1) using your own **batch gradient descent** code; 2) using your own **Newton's method** code; 3) using **glmfit** function of MATLAB. Please report your training and testing results in the following table:

accuracy	raw data	standardized data
batch GD		
Newton's method		
glmfit		

Besides, for your own batch-GD and Newton's method codes, please plot the evolution of training accuracies as a function of training iterations. Please attach this figure in your report and report how many iterations are required to get similar accuracy as glmfit?

(c) **Feature Analysis** It is very common in real world problem that features be relevant to response value in some nonlinear manner. For example, the feature of *room temperature* and the response of *how likely people feel comfortable* are related closely but not linearly: If temperature is too low or too high, people won't feel comfortable. There will be more examples in the bonus problem 4.3.

For this type of features, some pre-processing such as nonlinear transform may help improving model prediction. In this problem we want to experiment one heuristic method to pick out these features and one heuristic method to transform them. We will use two statistical tools: mutual information (MI) and Pearson correlation coefficient(PCC).

- *calculate mutual information*: Firstly, for every feature, if the feature is continuous, implement a function that discretize that feature into 10 equal density bins (each bin includes approximately the same number of samples); if the feature is integer, it is considered as already discretized. Then, for every discretized features, estimate the probability of each bin or each integer value. Finally, for every feature, calculate the mutual information between label and discretized feature.
- *calculate Pearson correlation coefficient*: For every feature, calculate the PCC by definition
- *compare MI and PCC*. Please **report** the calculation result of previous 2 steps by plotting the MI-vs-FeatureID and PCC-vs-FeatureID in two subplots (e.g. subplot(1,2,1), plot(mi); subplot(1,2,2), plot(pcc)).

After calculating MI and PCC, do the following experiment

- Select 20 features of highest mutual information values. Use *standardized data and glmfit function* to learn logistic regression model. Please **report** the ID of selected features and your training/testing accuracy.

- Among these 20 features, can you observe some features of low $|PCC|$? Choose 3 features of lowest $|PCC|$, design your discretization method (e.g. discretizing into M equal density bins or M equal width bins) and your resolution (i.e. how many bins to divide each feature into), and then create binary representations for these 3 selected features. (For example, we binarize one continuous feature x in this way: we divide the feature range into $M = 3$ bins: $x < 0, 0 \leq x \leq 1.5, 1.5 < x$; for every observation of x, we decide which bin is it from, and create a 3×1 indicator vector to represent and replace raw value of x; if $x < 0$, then x is from bin 1 and new feature is $[1, 0, 0]$, else-if $x > 1.5$, then x is from bin 3 and new feature is $[0, 0, 1]$, else, new feature is $[0, 1, 0]$. For more details and motivation of binary representation, please refer to 4.3 the bonus problem.) Train logistic regression model using *glmfit* and compare with results before binarizing 3 features. Make sure other 17 features are standardized. Please **report** (1) the ID of these 3 selected features, (2) how you discretize features and (3) your training/testing accuracy.

4.2 Generative Model and Discriminative Model (25 points)

In this problem, we use experiments on synthetic data to implement results in *Generative Model and Discriminative Model* section in algorithm part. In file toyGMM.mat you are provided a labeled training dataset $\mathcal{D}$ as plotted in figure (1). Samples are generated from mixture of 3 Gaussian components with parameters $\{\pi_i, \mu_i, S_i\}_{i=1,2,3}$. In the data file the samples from 3 components are recorded $\{$(dataTr.x1, dataTe.x1), (dataTr.x2, dataTe.x2), (dataTr.x3, dataTe.x3)$\}$. The component id is the label of samples.

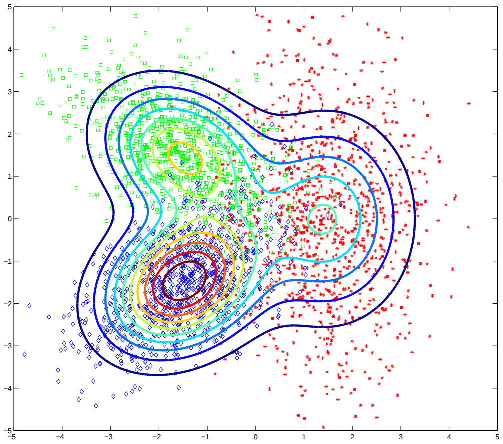

Figure 1: Mixture of 3 Gaussian distribution components: ground-truth probabilities and labeled samples.

For the training dataset, please to the following experiments:

(a) Write code to calculate MLE of parameters, assuming that covariances S_1, S_2, S_3 of three

Gaussian components are independent, print the value of parameters in the report, and report the testing accuracy.

(b) Write code to calculate MLE of parameters, assuming that $S_1 = S_2 = S_3$, print the value of parameters in the report, and report the testing accuracy.

(c) Train a multinomial logistic regression model, and report the testing accuracy.

(d) Feed parameters in above problems (a), (b), (c) into function plotContour.m to plot the contour of distribution, and the separating hyperplane. Please attach the figure, and analyze the results.

(e) Randomly select $\{1\%, 5\%, 10\%, 25\%, 50\%, 100\%\}$ of training data. For each subset train above 3 models and plot the testing accuracy w.r.t. training data sizes for three models. Make sure the ratio of each class is unchanged in selecting subsets (this is called *stratified* down-sampling).

We provided two code files: *trainEvalModels.m* and *plotBoarder.m*. You can either start with these code or start coding from scratch. If you use these two files, your work load will be very light.

4.3 Practical Logistic Regression on Toy Data (Bonus problem: 25 points)

Logistic regression method is an example of *generalized linear model* (GLM), and suitable to model linearly separable classes. However, in real world problems the data are mostly nonlinear. For example, in e-commerce companies, if data scientists want to predict the probability one customer clicks an advertisement based on the demographic data, it is not reasonable to use Zip code number or customer age in logistic regression directly, because the numerical value of zip code does not convey useful information and the probability does not necessarily change monotonically with age. Although there are advanced non-linear classifiers such as SVM using nonlinear kernel, random forests, deep neural networks, provided limited computational resource, it not always practical to implement them on large scale data.

(a) One solution in practice is to discretize features and represent them as multiple boolean features, that's, one raw feature is described with one multi-dimensional indicator vector.

Take the datasets in figure 2 for example: we used blue and red colors to represent two classes, and obviously none of they are linearly separable. However, if we discretize the region of $[-2, 2] \times [-2, 2]$ into 16 equal size rectangle grids, we can represent each sample with a 16-dimensional indicator vector, and intuitively we can expect a better classification accuracy.

Please load four datasets $\{\{\text{data1}.xTr, yTr, xTe, yTe\}, \cdots \{\text{data4}.xTr, yTr, xTe, yTe\}\}$ in figure (2) from file toySpiral.mat and do the following experiments:

- (**discretization**) Train logistic regression model on discretized representation of 2-dimensional samples. Please try equally dividing the space into $\{2^2, 4^2, 8^2, 16^2\}$ grids, and use cross-validation to choose the optimal discretization resolution. Please report the {training, heldout and testing} accuracies for these 4 models. You may assume every raw feature of all samples are within $[2, 2]$.

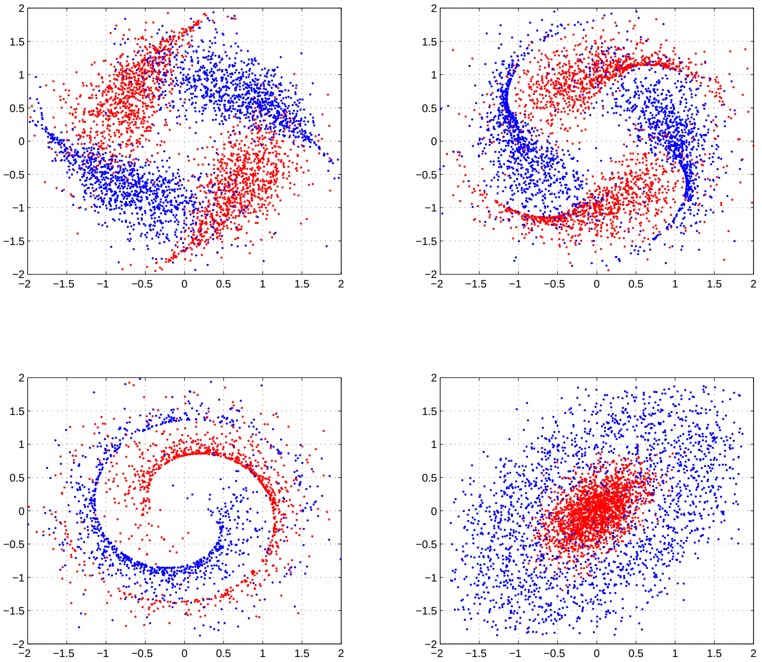

Figure 2: Synthetic datasets that are not linearly separable.

- (**$\boldsymbol{\ell}$2 norm regularization**) Divide the raw feature space into 8^2 girds, and train logistic regression with $\boldsymbol{\ell}_2$ regularization. Please try coefficient $\lambda \in \{1, 0.1, 0.01, 0.001\}$ and report the {training, heldout and testing} accuracies for these 4 models.
- (**visualization**) For dataset 2 and for dataset 3, please draw three 3D-bar plots for parameters of 1)best unregularized, and 2) best ℓ_2 regularized logistic regression under 8×8 discretization. For 2) and 3) choose the one with best coefficient. You may use matlab command **bar3** to draw the figures.

Notes:

- Remember that the coefficient for ℓ_2 regularization is defined for single sample. You are optimizing $\frac{1}{N}\sum_{n=1}^{N}(y_n \log \sigma(\boldsymbol{x}_n, \omega) + (1 - y_n) \log(1 - \sigma(\boldsymbol{x}_n, \omega)) - \lambda ||\boldsymbol{\omega}||_2^2$, not $\sum_{n=1}^{N}(y_n \log \sigma(\boldsymbol{x}_n, \omega) + (1 - y_n) \log(1 - \sigma(\boldsymbol{x}_n, \omega)) - \lambda ||\boldsymbol{\omega}||_2^2$
- For learning both unregularized and ℓ_2 regularized logistic regression models, please **use your own batch gradient descent code.**

(b) Let's design a special regualrization for the discretized representations to learn a classifier whose prediction change smoothly on the 2D raw feature space. We know that the binary representation of samples from two neighboring grids, e.g. grid d_1 and grid d_2, differ only at two bits: $[x_{d1}, x_{d,2}] = [0, 1]$ and $[x_{d1}, x_{d,2}] = [1, 0]$. Thus if ω_{d_1} and ω_{d_2} do not differ significantly, predictions on samples from grid d_1 and d_2 won't change drastically.

According to above discussion, let's design new regularized logistic regression problem:

$$\mathcal{L}_D = \frac{1}{N}\sum_{n=1}^{N} y_i \log \sigma(\omega_0 + \boldsymbol{\omega} \cdot \boldsymbol{x}_n) + (1 - y_i) \log(1 - \sigma(\omega_0 + \boldsymbol{\omega} \cdot \boldsymbol{x}_n)) \tag{11}$$

$$- \frac{1}{2}\lambda \sum_{(i,j)\in S} (\omega_i - \omega_j)^2 \tag{12}$$

where S is the collection of pairs of neighboring dimensions (grids). For simplicity, we only consider the 4-neighborhood of grids, and do not consider the diagonal neighbors.

- Please discretize the 2D space into 8^2 grids, index every grid and create the set S.
- Please train logistic regresison model (11) **using your batch gradient descent code**. You need to derive the derivative of $\mathcal{L}$ w.r.t. $\{\omega_0, \boldsymbol{\omega}\}$. Please try to learn four models using $\lambda \in \{1, 0.1, 0.01, 0.001\}$ and use cross validation to select the best one.
- Please draw 3D bar plot of parameters on dataset 2 and dataset 3, and compare with parameters without pairwise regularization over $\boldsymbol{\omega}$.

Submission Instruction: You need to provide the followings:

- Provide your answers in PDF file, named as `CSCI567_hw2_spring16_yourUSCID.pdf`. You need to submit the homework in both hard copy (at CS567 Homework lockers by 4 pm of the deadline date) and electronic version as pdf file on Blackboard. If you choose handwriting instead of typing all the answers, 40% points will be deducted.
- Submit ALL the code and report via Blackboard. The only acceptable language is MATLAB. For your program, you MUST include the main function called `CSCI567_hw1_spring16.m` in the root of your folder. After running this main file, your program should be able to generate all of the results needed for the programming assignment, either as plots or console outputs. You can have multiple files (i.e your subfunctions), however, the only requirement is that once we unzip your folder and execute your main file, your program should execute correctly. Please double-check your program before submitting. You should only submit one `.zip` file. No other formats are allowed except `.zip` file. Also, please name it as `[lastname]_[firstname]_hw2_spring16.zip`.

Collaboration: You may collaborate. However, collaboration has to be limited to discussion only and you need to write your own solution and submit separately. You also need to list with whom you have discussed.