1 Density Estimation

(a) (**10 points**) Suppose we have N i.i.d samples $x_1, x_2, \cdots, x_n$. We will practice the maximum likelihood estimation techniques to estimate the parameters in each of the following cases:

- We assume that all samples can only take value between 0 and 1, and they are generated from the Beta distribution with parameter α unknown and $\beta = 1$. Please show how to derive the maximum likelihood estimator of α.
- We assume that all samples are generated from Normal distribution $\mathcal{N}(\theta, \theta)$. Please show how to derive the maximum likelihood estimator of θ.

(b) (**10 points**) Suppose random variable X is distributed according to density function $f(x)$ and the kernel density estimation is in the form of $\hat{f}(x) = \frac{1}{n}\sum_{i=1}^{n}\frac{1}{h}K(\frac{x-X_i}{h})$. Show the bias of the kernel density estimation method by the following steps:

- Show $\mathbf{E}_{X_1,\cdots,X_n}[\hat{f}(x)] = \frac{1}{h}\int K(\frac{x-t}{h})f(t)dt$.
- Use Taylor's theorem around x on the density $f(x-hz)$ with $z = \frac{x-t}{h}$.
- Compute the bias $\mathbf{E}[\hat{f}(x)] - f(x)$.

2 Nearest Neighbor

(a) (**15 points**) Suppose we have the locations (coordinates) of 10 USC students during class time, and we know their majors, as follows:
Mathematics: $\{(10, 49), (-12, 38), (-9, 47)\}$
Electrical Engineering: $\{(29, 19), (32, 31), (37, 38)\}$
Computer Science: $\{(8, 9), (30, -28), (-18, -19), (-21, 12)\}$

- (**5 points**) Normalize/standardize the coordinates above along x and y axes, such that it has a zero-mean and standard deviation equal to 1.
- (**10 points**) Suppose we have a student whose coordinate is at $(9, 18)$ with unknown major. Using K-Nearest Neighbor with L_2 distance metric, predict the student's major if we are using $K = 1$ (**2 points**) and if we are using $K = 3$ (**2 points**). Similarly, what are the student's major predictions if we use L_1 distance metric with $K = 1$ (**2 points**) and with $K = 3$ (**2 points**). Please compare the results between these 4 different predictions (**2 points**). Do not forget to normalize/standardize the coordinate of the student with unknown major using the mean and standard deviation of the students with known major. Provide intermediate computations of how do you arrive at your predictions.

(b) (**10 points**) Suppose now we want to derive a probabilistic K-Nearest Neighbor for classification of an unlabeled data point $\mathbf{x}$, which is D-dimensional. We have a (multi-dimensional) D-sphere with center at $\mathbf{x}$, allowing its radius to grow until it precisely contains K labeled data points, irrespective of their class. At this size, the volume of the sphere is V. Let there be a total of N labeled data points in the entire space (both inside and outside of the sphere), with N_c data points labeled as class c, such that $\sum_c N_c = N$. Also, a subset of the

K data points inside of the sphere belongs to class c, there are K_c of them in total. We model estimated density associated with each class as $p(\mathbf{x} \mid Y = c) = \frac{K_c}{N_c V}$ and the class prior as $p(Y = c) = \frac{N_c}{N}$.[1]

- (**5 points**) Using the fact that $\sum_c K_c = K$, derive the formula for unconditional density $p(\mathbf{x})$.
- (**5 points**) Using Bayes rule, derive the formula for the posterior probability of class membership $p(Y = c \mid \mathbf{x})$.

3 Decision Tree

(a) (**10 points**) Suppose we did 80 observations resulting in the following tabulated data with binary features, such as temperature, humidity, and sky condition, and we observe the occurrence of the rainy day, denoted by total rainy days per total observations (*#Rainy/#Observations*) for each combination of features. Using this data, we want to grow a decision tree which maximizes information gain, to predict the future occurrence rainy days. Please provide the intermediate computations (**4 points**), the predictor variable/feature that you select for the split in each node of the tree based on information gain criteria (**4 points**), and draw the resulting decision tree (**2 points**).

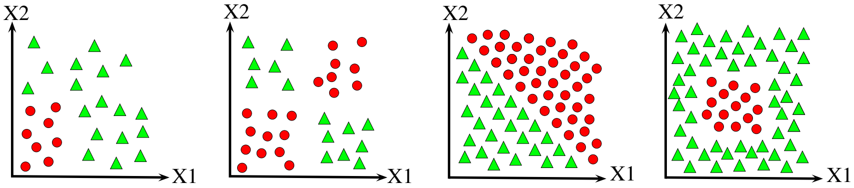
Temperature	Humidity	Sky Condition	#Rainy/#Observations
Hot	High	Cloudy	9/10
Hot	High	Clear	5/10
Hot	Low	Cloudy	6/10
Hot	Low	Clear	3/10
Cool	High	Cloudy	7/10
Cool	High	Clear	2/10
Cool	Low	Cloudy	3/10
Cool	Low	Clear	1/10

(b) (**10 points**) In training decision trees, the ultimate goal is to minimize the classification error. However, the classifaction error is not a smooth function; thus, several surrogate loss functions have been proposed. Two of the most common loss functions are the *Gini index* and *Cross-entropy*, see [MLaPP, Section 16.2.2.2] or [ESL, Section 9.2.3] for the definitions. Prove that, for any discrete probability distribution p with K classes, the value of the Gini index is less than or equal to the corresponding value of the cross-entropy. This implies that the Gini index is a better approximation of the misclassification error.

Definitions: For a K-valued discrete random variable with probability mass function $p_i, i = 1, \ldots, K$ the Gini index is defined as: $\sum_{k=1}^{K} p_k(1 - p_k)$ and the cross-entropy is defined as $-\sum_{k=1}^{K} p_k \log p_k$.

[1] This problem is a rephrase of some materials from ”Pattern Recognition and Machine Learning” book by Christopher M. Bishop. However, even without having/reading the book, this problem should be fairly easy to solve, just by following the description in this problem set.

(c) (**5 points**) Suppose we have continuous attributes X_1 and X_2, and we have collected a labeled dataset having these attributes with two class labels $Y = c$ (depicted as red circles) and $Y = t$ (depicted as green triangles). We want to use decision trees for classification on this dataset. The test for split at each internal node is using inequalities of the form $X_i > s$, and each node is evaluated as either $true$ or $false$. s is called a split point, which is any real number chosen by the learning algorithm. In this decision tree, each attribute is allowed to be tested any number of times on any path in the tree. Consider the four cases depicted below (a), (b), (c), and (d). Which of these dataset can be classified correctly with a tree depth less than 6? (*hint*: draw the decision boundaries on each cases)



(a) (b) (c) (d)

4 Naive Bayes

(a) (**15 points**) Suppose a training set with N examples $(X_1, Y_1), (X_2, Y_2), \cdots, (X_N, Y_N)$ is given, where $X_i = (x_{i1}, \cdots, x_{iD}) \in \mathbb{R}^D$ is a D-dimensional feature vector, and $Y_i \in \{1, 2, \cdots, K\}$ is its corresponding label. Given the following assumptions:

- The label variable Y follows a categorical distribution, with parameter $p_k = P(Y = k)$, for $k = 1, 2, \cdots, K$.
- For each x_j of the D features, we have that $P(x_j|Y = y_k)$ follows a Gaussian distribution of the form $\mathcal{N}(\mu_{jk}, \sigma_{jk})$.
- Naive Bayes assumption: for all $j' \neq j$, x_j and $x_{j'}$ are conditionally independent given Y.

Please provide the maximum likelihood estimation for the parameters of the Naive Bayes with Gaussian assumption. In other words, you need to provide the estimates for p_k, μ_{jk}, and σ_{jk}, for $j = 1, \ldots, D$ and $k \in \{1, 2, \cdots, K\}$.

(b) (**15 points**) As a TA, you want to know if one student understand the knowledge covered in Machine Learning class or not, given his/her answers of a quiz with D multiple choice problems.

A binary label variable Y is used to indicate whether the student understand enough machine learning knowledge($Y = 1$) or not($Y = 0$), where Y follows a Bernoulli distribution with parameter $\pi = P(Y = 1)$. For each problem, a binary feature variable X_j denotes whether the answer to question j is correct($X_j = 1$) or not($X_j = 0$), where each feature X_j follows a Bernoulli distribution given the label $P(X_j|Y = y_k) = \theta_{jk}^{X_j}\theta_{jk}^{1-X_j}$.

Given a D-dimensional binary vector $\mathbf{X} = \{X_1, \ldots, X_D\}$ and a label variable Y, you want to know whether the student is good not not under Naive Bayes assumption.

Please show that $P(Y|\mathbf{X})$ has the form as follows:

$$P(Y = 1|X) = \frac{1}{1 + \exp(-w_0 + \mathbf{w}^\top \mathbf{X})}$$

Specifically, you need to find the explicit form of w_0 and $\mathbf{w}$ in terms of π and θ_{jk}, for $j = 1, \ldots, D$ and $k \in \{0, 1\}$.

5 Programming

5.1 Density Estimation

In this problem, you will write a MATLAB program to fit a density estimator using 1) `Gaussian kernel`, 2) `Epanechnikov kernel`, and 3) `histogram`.

You can get newest version of Matlab from http://itservices.usc.edu/matlab/.

The file `hw1progde.mat` contains two sets of data points, `x_tr` and `x_te`, which are i.i.d generated from the same unknown univariate probability density of the unit interval, i.e., between 0 and 1.

Fit the estimators at a variety of values of the bandwidth h on `x_tr` (At least 5 different h for each estimator). Using `x_te` to estimate the integrated squared bias and the integrated variance of each estimator.

You should approximate the integrals by summing over 50 evenly spaced points in the unit interval, and estimate the variance by dividing the test data `x_te` into subsets of 500 points and calculating kernel density estimates for each subset.

Plot the integrated squared bias, the integrated variance, and the integrated squared error (the sum of the first two terms) as a function of h. Estimate the optimal bandwidth based from your plot. For the bandwidth you choose, plot the density estimate. Compare the relative performance of the three methods based on your results.

5.2 Classification

In this assignment, you will experiment with three classification algorithms on real-world datasets. You will use MATLAB's functions to experiment with Decision Tree, but you need to implement Naive Bayes and K-Nearest Neighbor (K-NN) algorithms by yourself. You are NOT allowed to use any related MATLAB toolbox functions for Naive Bayes and K-NN (e.g. `knnclassify`, `knnsearch`). Below, we describe the steps that you need to take to accomplish this programming assignment.

Dataset: We have pre-processed the *Tic-Tac-Toe Endgame Dataset* (for all classification algorithms) and the *Nursery Dataset* (ONLY for Naive Bayes) from UCI's machine learning data repository. The training/validation/test sets are provided along with the assignment in Blackboard as `hw1ttt_train.data`, `hw1ttt_valid.data`, and `hw1ttt_test.data` for the *Tic-Tac-Toe Endgame Dataset*, and `hw1nursery_train.data`, `hw1nursery_valid.data`, and `hw1nursery_test.data` for the *Nursery Dataset*. For description of the datasets, please refer to https://archive.ics.uci.edu/ml/datasets/Tic-Tac-Toe+Endgame and https://archive.ics.uci.edu/ml/datasets/Nursery.

Please follow the steps below:

(a) Data Inspection The first step in every data analysis experiment is to inspect the datasets and make sure that the data has the appropriate format. You will find that the features in the provided dataset are categorical. However, some of the algorithms require the features to be real-valued numbers. To convert a categorical feature with K categories to real-valued number, you can create K new binary features. The ith binary feature indicates whether the original feature belongs to the ith category or not.

(b) Implement Naive Bayes Please fill in the function `naive_bayes` in `naive_bayes.m` file (in Blackboard). The inputs of this function are training data and new data (either validation or testing data). The function needs to output the accuracy on both training and new data (either validation or testing). Note that some feature values might exist in the validation/testing data, but do not exist in the training data. In that case, please set the probability of that feature value to a small value, for example, 0.1. Since you will use Naive Bayes to operate on 2 datasets later on, it is strongly recommended that you write an implementation of the algorithm which is general-purpose, as less dependent on the dataset (size, etc.) as possible.

(c) Implement *K*-NN Please fill in the function `knn_classify` in `knn_classify.m` file (in Blackboard). The inputs of this function are training data, new data (either validation or testing data) and K. The function needs to output the accuracy on both training and new data (either validation or testing).

(d) Performance Comparison Compare the three algorithms (K-NN, Naive Bayes, and Decision Tree) on the provided dataset.

K-NN: Consider $K = 1, 3, 5, \cdots, 15$. For each K, report the training, validation and test accuracy. When computing the training accuracy of K-NN, we use leave-one-out strategy, i.e. classifying each training point using the remaining training points. Note that we use this strategy only for K-NN in this assignment. Operate ONLY on the *Tic-Tac-Toe Endgame Dataset*.

Decision Tree: Train decision trees using function `ClassificationTree.fit` or `fitctree` in Matlab. Report the training, validation and test accuracy for different split criterions (*Gini index* and *cross-entropy*, using the SplitCriterion attribute), by setting the minimum number of leaf node observations to $1, 2, \cdots, 10$ (using the MinLeaf attribute). So basically, you need to report the results for $2 \times 10 = 20$ different cases. When training decision trees, please turn off pruning using the `Prune` attribute. Operate ONLY on the *Tic-Tac-Toe Endgame Dataset*.

Naive Bayes: Report and compare the training, validation, and test accuracy, both on the *Tic-Tac-Toe Endgame Dataset* and the *Nursery Dataset*. If there is a significant difference between the two datasets' classification accuracy, could you guess why?

(e) Decision Boundary In this step, you need to apply K-NN on the `hw1boundary.mat` dataset (provided in Blackboard) which is a binary classification dataset with only two features. You need to run K-NN with $K = 1, 5, 15, 25$ and examine the decision boundary. A simple way to visualize the decision boundary, is to draw 10000 data points on a uniform 100×100 grid in the square $(x, y) \in [0, 1] \times [0, 1]$ and classify them using the K-NN classifier. Then, plot the data points with different markers corresponding to different classes. Repeat this process for all k and discuss the smoothness of the decision boundaries as K increases.

Submission Instruction: You need to provide the followings:

- Provide your answers to problems 1-4, 5.1, 5.2(d), and 5.2(e) in PDF file, named as `CSCI567_hw1_fall15.pdf`. You need to submit the homework in both hard copy (at CS Front Desk with a box labeled as CSCI567-homework by 5pm of the deadline date) and electronic version as pdf file on Blackboard. If you choose handwriting instead of typing all the answers, you will get 40% points 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_fall15.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 this programming assignment, either as plots or console outputs. You can have multiple files (i.e your sub-functions), 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]_hw1_fall15.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.