**Homework 2** **CSCE 633**
Due: 11.59pm on October 20, 2017

---

**Instructions for homework submission**
Please submit on eCampus a **single pdf** file containing your solutions.
a) Please typewrite in Latex the answers to the math problems. If this is not possible, please handwrite your solution *very clearly*, scan it and merge it to the final pdf file. Make sure that your solution is visible after scanning. Non-visible solutions will not be graded: we wouldn't like our TA to have to guess what you are writing :)
b) Please write a brief report for the experimental problems. At the end of the pdf file, please include your code. The code has to be directly converted instead of scanned (i.e. the text in the code must be selectable).
c) Please start early :)

**Question 1: Decision Tree**
(a) Suppose we have 80 observations representing 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 (i) the intermediate computations, (ii) the predictor variable/feature that you select for the split in each node of the tree based on information gain criteria, and (iii) draw the resulting decision tree.

| 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) 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. 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 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: $\phi_G(p_1, \ldots, p_K) = \sum_{k=1}^{K} p_k(1 - p_k)$ and the entropy is defined as $\phi_E(p_1, \ldots, p_K) = \sum_{k=1}^{K} p_k \log p_k$.

(c) **Classifying benign vs malignant tumors:** We would like to classify if a tumor is benign or malign based on its attributes. We use data from the Breast Cancer Wisconsin Data Set of the UCI Machine Learning Repository: `https://archive.ics.uci.edu/ml/datasets/breast+cancer+wisconsin+(original)`.

Inside "Homework 2" folder on Piazza you can find one file containing the data (named "hw2_question1.csv") for our experiments. The rows of these files refer to the data samples, while the columns denote the features (columns 1-9) and the outcome variable (column 10), as described bellow:

1. Clump Thickness: discrete values $\{1, 10\}$
2. Uniformity of Cell Size: discrete values $\{1, 10\}$
3. Uniformity of Cell Shape: discrete values $\{1, 10\}$
4. Marginal Adhesion: discrete values $\{1, 10\}$
5. Single Epithelial Cell Size: discrete values $\{1, 10\}$
6. Bare Nuclei: discrete values $\{1, 10\}$
7. Bland Chromatin: discrete values $\{1, 10\}$
8. Normal Nucleoli: discrete values $\{1, 10\}$
9. Mitoses: discrete values $\{1, 10\}$
10. Class: 2 for benign, 4 for malignant (this is the **outcome** variable)

(c.i) Compute the number of samples belonging to the benign and the number of samples belonging to the malignant case. What do you observe? Are the two classes equally represented in the data? Separate the data into a train (2/3 of the data) and a test (1/3 of the data) set. Make sure that both classes are represented with the same proportion in both sets.

(c.ii) *Implement* two decision trees using the training samples. The splitting criterion for the first one should be the entropy, while for the second one should be the gini index. Plot the accuracy on the train and test data while the number of nodes in the tree increases for both splitting criteria. Do you observe any differences in practice?

(c.ii) **Bonus:** *Implement* pre-pruning using a lower threshold on the values of the splitting criterion for each branch. Experiment with different thresholds and report results both in the train and test set.

**Question 2: Kernel Ridge Regression**

In this problem, we will derive kernel ridge regression, a nonlinear extension of linear ridge regression. Given a set of training data $\{(\mathbf{x_1}, y_1), \ldots, (\mathbf{x_N}, y_N)\}$ where $\mathbf{x_n} \in \mathcal{R}^D$, linear ridge regression learns the weight vector $w$ (assuming the bias term is absorbed into $w$) by optimizing the following objective function:

$$J(\mathbf{w}) = (\mathbf{y} - \mathbf{X}\mathbf{w})^T(\mathbf{y} - \mathbf{X}\mathbf{w}) + \lambda\|\mathbf{w}\|_2^2$$

where $\lambda$ is the regularization coefficient.
Assume that we apply a nonlinear feature mapping to each sample $\mathbf{x_n} \rightarrow \mathbf{\Phi}_i = \boldsymbol{\phi}(\mathbf{x_n}) \in \mathcal{R}^T$, where $T \gg D$. Define $\Phi \in \mathcal{R}^{N \times T}$ as a matrix containing all $\boldsymbol{\Phi_n}$.

(a) Express the above criterion function in terms of the non-linear transform $\boldsymbol{\phi}$ and show that the weights $\mathbf{w}^*$ that minimize the criterion function can be written as

$$\mathbf{w}^* = \mathbf{\Phi}^T(\mathbf{\Phi}\mathbf{\Phi}^T + \lambda \boldsymbol{I_N})^{-1}\mathbf{y}$$

where $\mathbf{y} = [y_1, \ldots, y_N]^T$ and $\mathbf{X} = \begin{bmatrix} - \mathbf{x_1}^T - \\ \vdots \\ - \mathbf{x_N}^T - \end{bmatrix}$

*Hint:* You may use the following identity for matrices. For any matrix $P \in \mathcal{R}^{p \times p}$, $B \in \mathcal{R}^{q \times p}$, $R \in \mathcal{R}^{q \times q}$ and assume the matrix inversion is valid, we have

$$(\mathbf{P}^{-1} + \mathbf{B}^T\mathbf{R}^{-1}\mathbf{B})^{-1}\mathbf{B}^T\mathbf{R}^{-1} = \mathbf{P}\mathbf{B}^T(\mathbf{B}\mathbf{P}\mathbf{B}^T + \mathbf{R})^{-1}$$

(b) Given a testing sample $\boldsymbol{\phi}(\mathbf{x})$, show that the prediction $y = \mathbf{w}^{*T}\boldsymbol{\phi}(\mathbf{x})$ can be written as:

$$y = \boldsymbol{y}^T(\boldsymbol{K} + \lambda \boldsymbol{I}_N)^{-1}\kappa(\mathbf{x})$$

where $\mathbf{K} \in \mathcal{R}^{N \times N}$ is a kernel matrix defined as $K_{ij} = \boldsymbol{\Phi_i}^T\boldsymbol{\Phi_j}$, $\kappa(\mathbf{x}) \in \mathcal{R}^N$ is a vector with n$^{th}$ element $(\kappa(\mathbf{x}))_n = \boldsymbol{\phi}^T(\mathbf{x_n})\boldsymbol{\phi}(\mathbf{x})$. Now you can see that $y$ only depends on the dot product (or kernel value) of $\boldsymbol{\Phi_i}$.

(c) **Bonus:** Compare the computational complexity between linear ridge regression and kernel ridge regression.

**Question 3: Support Vector Machines**

We will use the Phishing Websites Data Set from UCI's machine learning data repository: `https://archive.ics.uci.edu/ml/datasets/Phishing+Websites`. The dataset is for a binary classification problem to detect phishing websites.

Inside "Homework 2" folder on Piazza you can find a file containing the data (named "hw2_question3.csv") for our experiments. The rows of these files refer to the data samples, while the columns denote the features (columns 1-30) and the binary outcome variable (column 31).

(a) **Data pre-processing:** All the features in the datasets are categorical. You need to preprocess the training and test data to make features with multiple values to features taking values only zero or one. If a feature $f_i$ have value $\{-1, 0, 1\}$, we create three new features $f_{i,-1}$, $f_{i,0}$, and $f_{i,1}$. Only one of them can have value 1 and $f_{i,x} = 1$ if and only if $f_i = x$. For example, we transform the original feature with value 1 into $[0, 0, 1]$. In the given dataset, the features 2, 7, 8, 14, 15, 16, 26, 29 (index starting from 1) take three different values $\{-1, 0, 1\}$. You need to transform each above feature into three 0/1 features. For all the following experiments randomly separate the data into train (2/3) and test (1/3) set.

(b) **Use linear SVM in LIBSVM:** LIBSVM is widely used toolbox for SVM and has Matlab interface. Download LIBSVM from `https://www.csie.ntu.edu.tw/~cjlin/libsvm/` and install it according to the README file provided with the download. Experiment with different values of misclassification cost $C$, applying 3-fold cross validation on the train set and reporting the cross validation accuracy and average training time. Report the results on the test set using the best $C$ that was found through cross-validation.

(c) **Use kernel SVM in LIBSVM:** LIBSVM supports a number of kernel types. Here you need to experiment with the polynomial kernel and RBF (Radial Basis Function) kernel and their different parameters. Based on the cross validation results of Polynomial and RBF kernel, which kernel type and kernel parameters will you choose?

(d) **Bonus: Implement linear SVM:** Implement the training and testing parts of a linear support vector machine. In your implementation, you can use publicly available quadratic programming functions (e.g. quadprog in Matlab) to solve the dual quadratic problem.