CS771: Practice Set 1

Problem 1

(Prototype based Classification) Given training data $\{(\boldsymbol{x}_n, y_n)\}_{n=1}^N$ for a classification problem (assuming binary labels $y_n \in \{-1, +1\}$), show that, for the $y = \text{sign}[f(\boldsymbol{x})]$ decision rule in the prototype based classification, $f(\boldsymbol{x})$ can be written as $f(\boldsymbol{x}) = \sum_{n=1}^N \alpha_n \langle \boldsymbol{x}_n, \boldsymbol{x} \rangle + b$, where $\boldsymbol{x}$ denotes the feature vector of the test example. For this form of the decision rule, give the expressions for the α_n's and b.

Problem 2

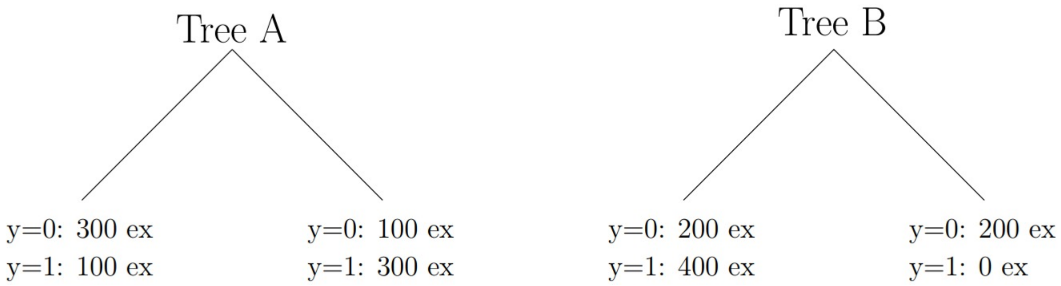
(Classes as Gaussians) Consider a model for binary classification where data in class "+1" and class "-1" is modeled using D-dimensional Gaussian distributions $\mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}_+, \boldsymbol{\Sigma})$ and $\mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}_-, \boldsymbol{\Sigma})$, respectively. For simplicity, let's assume the covariance matrix $\boldsymbol{\Sigma}$ to be the same for both the classes. Suppose we have already computed the means $\boldsymbol{\mu}_+$, $\boldsymbol{\mu}_-$ and the covariance matrix $\boldsymbol{\Sigma}$ from some training data. Now, given a test example $\boldsymbol{x}$, we wish to predict its class. To do so, we will use the following rule: assign $\boldsymbol{x}$ to the class under which it has a higher probability. Show that this rule can be written as $y = \text{sign}[f(\boldsymbol{x})]$ where $f(\boldsymbol{x}) = \boldsymbol{w}^\top \boldsymbol{x} + b$ and give the expressions of $\boldsymbol{w}$ and b. Under what condition this rule reduces to a learning with prototypes classifier?

Problem 3

(Consistent or Not?) An important notion for a classifier is that of *consistency*. A classification algorithm is said to be *consistent* if, whenever it has access to **infinite** amounts of training data, its error rate approaches the optimal error rate (a.k.a. *Bayes optimal*). Consider the noise-free setting (i.e., every training input is labeled correctly). Here, the Bayes optimal error rate is zero. Is the one-nearest-neighbor algorithm consistent in this setting? Briefly justify your answer in 100 words or less.

Problem 4

(Misclassification Rate vs Information Gain) Consider a binary classification data set consisting of 400 data points from class 0 and 400 data points from class 1. Suppose that a decision tree model $\mathcal{A}$ splits these into (300, 100) at the first leaf node and (100, 300) at the second leaf node. (Here, (n, m) denotes that n points are assigned to class 0 and m points are assigned to class 1.) Similarly, suppose that a second decision tree model $\mathcal{B}$ splits them into (200, 400) and (200, 0). (See the figure below.)



(1) Compute the training data **misclassification rate** (i.e., what fraction of training examples will be misclassified) for the two trees: are they equal or not? (2) Evaluate the **information gain** for the two trees and use these to compare the trees. (3) Do you get different answers for (1) and (2)? Does this make sense?

Problem 5

(Decision Trees for Regression) When constructing Decision Trees for classification, we prefer to split on features that divide a node such that the set of labels at the children nodes is as “pure” as possible, i.e., we want each child node to consist of examples such that one label dominates the other label(s). Entropy/information-gain can quantify the purity in the classification setting. Suggest a good criteria to choose a feature to split on if we were doing *regression* instead of classification (so the labels are real-valued instead of discrete labels in classification)? Your criteria should somehow quantify the homogeneity/diversity of the set of *real-valued* labels of the examples at each node. You may define it using equations or state it in words.

Problem 6

(How Many Calculations?) Consider learning a decision tree, given some training data where each input has D binary-valued features. Let’s assume that we will not test any feature that has been tested at one of the previous levels (but we can possibly test a feature at multiple nodes at the same level). How many information gain calculations would be needed to construct the full decision tree (i.e., assuming no pruning)? Just give the basic expression; no need to try simplifying it too much to get a more compact expression.