Theory and Algorithm for Machine Learning 2020
National University of Singapore CS5339
Prof Lee Wee Sun

Homework 1 (Due date Sunday 8 March 11.59pm)

Please write the following on your homework:

- Name
- Collaborators (write none if no collaborators)
- Source, if you obtained the solution through research, e.g. through the web.

While you may collaborate, you *must write up the solution yourself*. While it is okay for the solution ideas to come from discussion, it is considered as plagiarism if the solution write-up is highly similar to your collaborator's write-up or to other sources.
You solution should be submitted to IVLE workbin. Scanned handwritten solutions are acceptable but must be legible.
Late Policy: A late penalty of 20% per day will be imposed (no submission accepted after 5 late days) unless prior permission is obtained.

---

1. **VC-dimension of Decision Trees**
   Consider using decision trees/forests to implement Boolean functions, i.e. classifiers from $\{0,1\}^d$ to $\{0,1\}$.

   (a) Argue that decision trees of height d has VC-dimension at least 2^d.

   (b) Give a good upper bound for the VC-dimension of binary decision trees with n nodes that maps from $\{0,1\}^d$ to $\{0,1\}$.
   (Hint: Consider how many bits are used to encode such decision trees in Section 18.1 in SSBD.)

   (c) Give a good upper bound for the VC-dimension of a random forest with k trees where each tree has n leaves. Here we consider a random forest as a thresholded weighted average of the k decision trees.

2. **Kernels**
   (Modified from SSBD 16.6.4) Let N be any positive integer. For every $x, x' \in \{1,\ldots,N\}$ define
   $$K(x,x') = \min\{x,x'\}.$$

   (a) Prove that K is a valid kernel; namely, find a mapping $\psi : \{1,\ldots,N\} \to H$ where H is some Hilbert space, such that
   $$\forall x, x' \in \{1,\ldots,N\}, K(x,x') = \langle \psi(x), \psi(x') \rangle.$$

In this case, let H be a feature vector of length N.
(Hint: Consider using binary vectors (where each coefficient is either 0 or 1) of length N as the output of $\psi : \{1, \ldots, N\}$. What happens when you do inner product of two such vectors?)

(b) Using the representer theorem, optimal solutions of regularized loss minimization can be represented using

$$\mathbf{w} = \sum_{i=1}^{m} \alpha_i \psi(x_i),$$

where $x_1, \ldots, x_m$ are the training inputs and $\alpha_1, \ldots, \alpha_i$ are the learned parameters. Alternatively, we can represent $\mathbf{w}$ as a vector of real numbers. Which representation is smaller when:

i. m is much smaller than N

ii. N is much smaller than m

(c) Consider the case where $N = 3$ and assume that the function is observed at points 1 and 3 with $f(1) = 0$ and $f(3) = 1$. From the representer theorem, the optimal solution of regularised loss minimization can be represented as

$$f(x) = \alpha_1 K(1, x) + \alpha_2 K(3, x).$$

i. Argue that it is not possible to represent the function

$$\begin{bmatrix} f(1) \\ f(2) \\ f(3) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

using $\alpha_1 K(1, x) + \alpha_2 K(3, x)$.

ii. On the other hand, if we allow do not restrict the kernels to be parameterized by the input data, show that the same function can be represented with $\alpha_1 K(2, x) + \alpha_2 K(3, x)$ for some value of α_1, α_2.

(d) We are interested in doing text classification. Assume that the maximum number of any particular word in each file is N and there are a total of W possible words. A reasonable similarity function for two text files is the number of co-occurring words that they have in common. To count the number of co-occuring words, we first count how many common occurrences there are for each word i, then sum the common occurrences of words for all $i = 1, \ldots, W$. For example, if file x_1 contains "the quick brown fox jumps over the lazy dog" and file x_2 contains "the lazy dog slept through the whole event the entire time" the co-occuring words are "the"(2 co-occurrences), "lazy" (1 co-occurence), and "dog" (1 co-occurrence) and the similarity $K(x_1, x_2) = 4$. Argue that this similarity function forms a valid kernel.

3. **Learn to do Forex Trading**
You are a currency trader and have K currencies, labeled $0, \ldots, K - 1$, to trade. You use a

simple strategy where each day, you either move all your money from the current currency to another currency, or keep your money in the same currency by not trading. The exchange rate for trading from currency i to j on day d is r_{ij}^d, where $r_{ii}^d = 1$ as no trade is done. If you start with \$1 in your home currency, which we assume to be currency 0, on day 1 and do a sequence of D trades obtaining a sequence $c_1, c_2, \ldots, c_{D-1}, c_D = 0$ of currencies, where the last trade on day D must be back to your home currency, then the final amount in your home currency after the trades is $r_{0c_1}^1 \ldots r_{c_{D-1}0}^D$.

(a) Assume that you know the exchange rates for each of the D days into the future r_{ij}^d. Describe a dynamic programming algorithm for computing $V(j, d)$, the optimal amount that can be obtained from \$1 in currency j at the start start of the d-th day and trading optimally for the following $D - d$ trades with the final trade on day d being constrained to be to currency 0. (Hint: Construct the equation for $V(j, d-1)$ in terms of r_{ji}^{d-1} and $V(i, d)$.)

(b) You do not know the future exchange rates r_{ij}^d. However, you would like to learn a function $f(\mathbf{x}, \theta)$ for predicting $V(0, 1)$ based on a set of features $\mathbf{x}$ that you have extracted (e.g. current exchange rates, news from the different countries, etc.), where the features are used to learn the future exchange rates $r_{ij}^d(\mathbf{x}, \theta)$ with parameters θ that is learned.

   i. Describe how you can construct a deep neural network to simulate the dynamic programming computation and hence do end-to-end training for the function $f(\mathbf{x}, \theta)$ to predict $V(0, 1)$ as a function of $\mathbf{x}$. Describe the functions used (e.g. max functions, product functions) and how they are connected together as a network.

   ii. Assume that we would like to do regression with square loss to do the learning. Given observed exchange rates and feature vector $\mathbf{x}$ every day over a long period of time, describe how to construct the training examples $(\mathbf{x}_i, y_i)$.

(c) What advantage does learning $f(\mathbf{x}, \theta)$ as described have over doing supervised learning to learn $r_{ij}^d(\mathbf{x}, \theta)$ as a function of $\mathbf{x}$, then using the dynamic programing algorithm in part (a) to compute the prediction for the optimal gain?