Mehryar Mohri
Foundations of Machine Learning 2020
Courant Institute of Mathematical Sciences
Homework assignment 2
Oct 08, 2020
Due: Oct 28, 2020 [before class starts].

**A. Rademacher complexity**

1. Non-negativity of empirical Rademacher complexity: Show that for any hypothesis set $\mathcal{H}$ and sample $S$, we have $\widehat{\mathfrak{R}}_S(\mathcal{H}) \geq 0$.

2. Empirical Rademacher complexity of products: let $\mathcal{H}_1$ and $\mathcal{H}_2$ be two hypothesis sets of functions mappiing from the input space $\mathcal{X}$ to $\{0, 1\}$. Let $\mathcal{H} = \{h_1 h_2 \colon h_1 \in \mathcal{H}_2, h_2 \in \mathcal{H}_2\}$. Show that the following inequality holds:

$$\widehat{\mathfrak{R}}_S(\mathcal{H}) \leq \widehat{\mathfrak{R}}_S(\mathcal{H}_1) + \widehat{\mathfrak{R}}_S(\mathcal{H}_2).$$

   *Hint*: you could use Talagrand's contraction lemma.

**B. VC-dimension of neural networks**

Let $C$ be a concept class over $\mathbb{R}^r$ with VC-dimension $d$. A $C$-neural network with one intermediate layer is a concept defined over $\mathbb{R}^n$ that can be represented by a directed acyclic graph such as that of Figure 1, in which the input nodes are those at the bottom and in which each other node is labeled with a concept $c \in C$.

The output of the neural network for a given input vector $(x_1, \ldots, x_n)$ is obtained as follows. First, each of the $n$ input nodes is labeled with the corresponding value $x_i \in \mathbb{R}$. Next, the value at a node $u$ in the higher layer and labeled with $c$ is obtained by applying $c$ to the values of the input nodes admitting an edge ending in $u$. Note that since $c$ takes values in $\{0, 1\}$, the value at $u$ is in $\{0, 1\}$. The value at the top or output node is obtained similarly by applying the corresponding concept to the values of the nodes admitting an edge to the output node.

1. Let $H$ denote the set of all neural networks defined as above with $k \geq 2$ internal nodes. Show that the growth function $\Pi_H(m)$ can be upper bounded in terms of the product of the growth functions of the hypothesis sets defined at each intermediate layer.

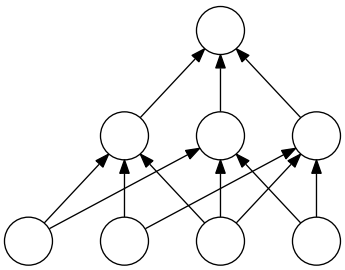

Figure 1: A neural network with one intermediate layer.

2. Use that to upper bound the VC-dimension of the $C$-neural networks (*Hint*: you can use the implication $m = 2x\log_2(xy) \Rightarrow m > x\log_2(ym)$ valid for $m \geq 1$, and $x, y > 0$ with $xy > 4$).

3. Let $C$ be the family of concept classes defined by threshold functions $C = \{\text{sgn}(\sum_{j=1}^{r} w_j x_j) \colon \mathbf{w} \in \mathbb{R}^r\}$. Give an upper bound on the VC-dimension of $H$ in terms of $k$ and $r$.

**C. Support Vector Machines (SVMs)**

1. Download and install the `libsvm` software library from:

   `http://www.csie.ntu.edu.tw/~cjlin/libsvm/`

2. Download the `abalone` data set:

   `http://archive.ics.uci.edu/ml/datasets/Abalone`

   Use the `libsvm` scaling tool to scale the features of all the data. Use the first 3133 examples for training, the last 1044 for testing. The scaling parameters should be computed only on the training data and then applied to the test data.

3. Download the `abalone` data set:

   `http://archive.ics.uci.edu/ml/datasets/Abalone`

   Use the `libsvm` scaling tool to scale the features of all the data. Use the first 3133 examples for training, the last 1044 for testing. The scaling parameters should be computed only on the training data and then applied to the test data.

4. Consider the binary classification that consists of distinguishing classes 1 through 9 from the rest. Use SVMs combined with polynomial kernels to tackle this binary classification problem.

   To do that, randomly split the training data into ten equal-sized disjoint sets. For each value of the polynomial degree, $d = 1, 2, 3, 4$, plot the average cross-validation error plus or minus one standard deviation as a function of $C$ (let other parameters of polynomial kernels in `libsvm` be equal to their default values), varying $C$ in powers of 2, starting from a small value $C = 2^{-k}$ to $C = 2^k$, for some value of $k$. $k$ should be chosen so that you see a significant variation in training error, starting from a very high training error to a low training error. Expect longer training times with `libsvm` as the value of $C$ increases.

5. Let $(C^*, d^*)$ be the best pair found previously. Fix $C$ to be $C^*$. Plot the ten-fold cross-validation error and the test errors for the hypotheses obtained as a function of $d$. Plot the average number of support vectors obtained as a function of $d$. How many of the support vectors lie on the margin hyperplanes?

6. In class, we gave two types of argument in favor of the SVMs algorithm: one based on the sparsity of the support vectors, another based on the notion of margin. Suppose that instead of maximizing the margin, we choose instead to maximize sparsity by minimizing the norm $p$ of the vector $\boldsymbol{\alpha}$ that defines the weight vector $\mathbf{w}$, for some $p \geq 1$. For simplicity, fix $p = 2$. This gives the following optimization problem for a kernel function $K$:

$$\min_{\alpha, b} \frac{1}{2} \sum_{i=1}^{m} \alpha_i^2 + C \sum_{i=1}^{m} \xi_i \tag{1}$$

$$\text{subject to } y_i \Big( \sum_{j=1}^{m} \alpha_j y_j K(x_i, x_j) + b \Big) \geq 1 - \xi_i, i \in [1, m]$$

$$\xi_i, \alpha_i \geq 0, i \in [1, m].$$

   (a) Show that modulo the non-negativity constraint on $\boldsymbol{\alpha}$, the problem coincides with an instance of the primal optimization problem of SVMs (indicate exactly how to view it as such).

   (b) Is the positive-definiteness of the kernel function $K$ needed to ensure that this is a convex optimization problem? Justify your response.

(c) Derive the dual optimization of problem of (1).

(d) Suppose we omit the non-negativity constraint on $\boldsymbol{\alpha}$. Use `libsvm` to solve the problem. Plot the ten-fold cross-validation training and test errors for the hypotheses obtained based on the solution $\boldsymbol{\alpha}$ as a function of $d$, for the best value of $C$ measured on the validation set.