1. Preliminaries and some basic tools

**Assigned reading:** Chapters 1, 2, and 5 of the course notes. (Introduction, Concentration Inequalities, and Formulation of the Learning Problem). **Additional recommended reading:** Shalev-Shwartz and Ben-David, *Understanding Machine Learning, from Theory to Algorithms*, Chapters 1-3 and Appendix B.

**Problems to be handed in:**

1. **[Empirical cumulative distribution functions on the reals]**
   Let $F$ be a cumulative distribution function (CDF) on $\mathbb{R}$ and let $X_1, \ldots, X_n$ be $n$ independent random variables with distribution $F$. Denote the empirical distribution function by $F_n(c) = \frac{1}{n} \sum_{i \in [n]} 1\{X_i \leq c\}$ for $c \in \mathbb{R}$ and let $\Delta_n = \sup_{c \in \mathbb{R}} |F_n(c) - F(c)|$.

   (a) Use the Hoeffding inequality to get an upper bound on $\mathbb{P}\left\{ |F_n(c) - F(c)| \geq \frac{t}{\sqrt{n}} \right\}$ valid for any $c \in \mathbb{R}$ and $t \geq 0$ that depends only on $t$.

   (b) Using your answer to (a), find an upper bound on $E[|F_n(c) - F(c)|]$ valid for any $c \in \mathbb{R}$ that depends only on $n$. (Hint: For any nonnegative random variable $Y$, $E[Y] = \int_0^\infty P\{Y > c\} dc$.)

   (c) Use McDiarmid’s bound to find an upper bound on $P\left\{ |\Delta_n - E[\Delta_n]| \geq \frac{t}{\sqrt{n}} \right\}$. (A considerably stronger result is known, namely, $\mathbb{P}\{ |\Delta_n| \geq \frac{t}{\sqrt{n}} \} \leq 2e^{-2t^2}$, due to Massart, “The tight constant in the Dvoretzky-Kiefer-Wolfowitz inequality, *Annals of Probability*, vol. 18, no. 3, 1990, pp. 1269-1283.)

2. **[PAC learnability of right subintervals]**
   Consider the concept learning problem for the triple $(X, \mathcal{P}, \mathcal{C})$, where $X = [0, 1]$, $\mathcal{C} = \{[\tau, 1] : 0 \leq \tau \leq 1\}$, and $\mathcal{P}$ is the set of all probability distributions on $X$. Describe an ERM classifier $\hat{C}_n$ and show that it PAC learns $\mathcal{C}_n$ and let $C^*$ be the target concept. A random data sample is given by $Z^n = (Z_1, \ldots, Z_n)$, where $Z_i = (X_i, Y_i) = (X_i, 1\{X_i \in C^*\})$ and $C^* = [\tau^*, 1]$ is the target concept.

3. **[A converse result – aka no free lunch theorem]**
   Let $n$ be a fixed positive integer and suppose $B_1, \ldots, B_{2n}$ are Bernoulli(1/2) random variables, modeling fair coin tosses. Let $X_1, \ldots, X_n$ be selector random variables, that are each uniformly distributed over $[2n]$. Finally, assume all $3n$ of the above random variables are mutually independent. These variables have the following interpretation. A random data sample is given by $Z^n = (Z_1, \ldots, Z_n)$, where $Z_i = (X_i, B_{X_i})$. Thus, if $G = \{k : X_i = k \text{ for at least one } i \in [n]\}$, then seeing the data sample is equivalent to learning $G$ and the values of the coin tosses for indices $k \in G$. Suppose $\mathcal{A}$ is a learning algorithm that takes $Z^n$ as input and outputs a classifier function $\hat{f} : [2n] \rightarrow \{0, 1\}$.

   (a) Show that for any choice of $\mathcal{A}$ and any $k \in [2n]$, $\mathbb{P}\{ \hat{f}(k) \neq B_k \} \geq 0.25$.

   (b) Thus, by part (a), the average fraction of variables among $\{B_1, \ldots, B_{2n}\}$ that $\hat{f}$ correctly predicts is at most 75%. Find an upper bound smaller than one on the probability that $\hat{f}$ correctly predicts at least 90% of the variables. (Apply Markov’s inequality.)

   (c) Show that for any learning algorithm $\mathcal{A}$, there exists a deterministic choice $b_1, \ldots, b_{2n}$ (depending on $\mathcal{A}$), such that, given $(B_1, \ldots, B_{2n}) = (b_1, \ldots, b_{2n})$, the conditional probability that $\hat{f}$ correctly predicts at least 90% of the variables is less than or equal to 5/6. (Once $\mathcal{A}$ and the $b$’s are fixed, the randomness remaining is due to the randomness of the selector variables $X_1, \ldots, X_n$.)
(d) In this part we consider a single realizable concept learning problem \((X, \mathcal{P}, \mathcal{F})\) that simultaneously has the above problem embedded in it for all \(n \geq 1\), and show that it is not PAC learnable. Let \(X = \{0, 1\}\) and for \(m \geq 1\) let \(\mathcal{F}_m\) denote the set of all binary valued functions on \(X\) that are constant over intervals of the form \(\left(\frac{i}{m}, \frac{i+1}{m}\right]\) for some \(0 \leq i \leq m - 1\). Note that \(|\mathcal{F}_m| = 2^m\), because selecting an \(f\) in \(\mathcal{F}_m\) is equivalent to deciding its value on each of the \(m\) subintervals. Let \(\mathcal{F} = \mathcal{F}_m\). Also, let \(\mathcal{P}\) contain only a single distribution \(P\), namely, the uniform probability distribution over \(X\). Thus, the data is represented by \(Z^n = (Z_1, \ldots, Z_n)\), where \(Z_i = (X_i, f^*(X_i))\) such that the \(X_i\)'s are independent and uniformly distributed over \((0, 1)\) and \(f^*\) is the true concept, with \(f^* \in \mathcal{F}\). Show that \((X, \mathcal{P}, \mathcal{F})\) is not PAC learnable. You may give a high level description without all details.

4. [Bregman divergence]

Let \(\mathcal{S}\) denote a convex subset of \(\mathbb{R}^d\) with nonempty relative interior\(^1\), \(ri(\mathcal{S})\), and let \(\phi\) be a strictly convex function on \(\mathcal{S}\) that is differentiable on \(ri(\mathcal{S})\). The Bregman divergence generated by \(\phi\), is the function \(d_\phi : \mathcal{S} \times ri(\mathcal{S}) \rightarrow \mathbb{R}_+\) defined by \(d_\phi(p, q) = \phi(p) - \phi(q) - \langle p - q, \nabla \phi(q) \rangle\). It is positive if \(p \neq q\) and strictly convex in \(p\), but not necessarily convex in \(q\).

(a) Identify the Bregman divergence for the special cases (i) \(\mathcal{S} = \mathbb{R}^d\), \(\phi(p) = \|p\|^2 = \sum_i p_i^2\), and (ii) \(\mathcal{S}\) is the set of \(d\) dimensional probability vectors and \(\phi(p) = \sum_i p_i \ln p_i\), where \(0 \ln 0\) is taken to be zero (i.e. \(\phi(p)\) is the negative of the entropy of \(p\)).

(b) Let \(p_1, \ldots, p_n \in \mathcal{S}\). Their centroid, \(\overline{p}\), is defined by \(\overline{p} \triangleq \frac{1}{n} \sum_{i=1}^n p_i\). Show that if \(\overline{p} \in ri(\mathcal{S})\), then \(\overline{p}\) is the unique minimizer of the function \(q \mapsto \frac{1}{n} \sum_{i=1}^n d_\phi(p_i, q)\).

(c) Briefly explain the significance of part (b) for generalization of the K-means clustering algorithm. (See a description of the algorithm at https://en.wikipedia.org/wiki/K-means_clustering.\)\)

(d) The Kullback-Leibler (KL) divergence for probability distributions \(p\) and \(q\), defined by \(d_{KL}(p||q) = \sum_i p_i \ln \frac{p_i}{q_i}\), is an example of an \(f\) divergence, which is something of the form \(d_f(p, q) = \sum_i q_i f\left(\frac{p_i}{q_i}\right)\), corresponding to the function \(f(u) = u \ln u\). (For \(f\) divergence, the function \(f\) should be convex function over \(\mathbb{R}_+\), strictly convex at 1, with \(f(1) = 0\). You may assume \(f\) is also twice continuously differentiable.) Show that the only \(f\) divergence that is also a Bregman divergence for probability distributions, is the KL divergence, up to a constant multiple. (Hint: For Bregman divergences, \(p \mapsto d(p, q) - d(p, q')\) is a linear function over the space of probability distributions, so its Hessian must be proportional to the all ones matrix. By computation, the Hessian is a diagonal matrix, so it must be zero.)

5. [Getting started on Python]

Suppose the \(k\) nearest neighbor (KNN) classifier (not to be confused with the K-means clustering algorithm) is used for classification of the classic Iris flower data set, with uniform weighting of \(k\) nearest neighbors. Use only the first two measurements for each sample, i.e. the sepal length and width. Suppose the 150 samples are randomly, uniformly divided into 100 training samples, used to train the classifier, and 50 samples to test the classifier. The scores on the training set and on the test set are defined to be the fraction of correct classifications for those sets. The scores are random because of the random splitting of the samples into two groups. Here is what you need to compute: (a) Determine the number of neighbors \(k\) that maximizes the expected score of the trained classifier on the training sample. (b) Determine the number of neighbors \(k\) that maximizes the expected score of the trained classifier on the test data. Use 100 repetitions of the training/testing experiment to estimate these average scores. To get started, see http://nbviewer.jupyter.org/urls/courses.engr.illinois.edu/ece543/sp2019/ece543_PythonProblem1.ipynb?flush_cache=true.

---

\(^1\)The affine span of a set \(C \subset \mathbb{R}^n\) is defined by \(\text{aff}(C) = \{ \sum_{i=1}^k \lambda_i x_i : k \geq 1, x_i \in C, \sum_{i=1}^k \lambda_i = 1\}\). (Similar to convex hull but without the constraint that \(\lambda_i \geq 0\) for all \(i\).) The relative interior of a convex set \(C\) is the set of \(x \in C\) such that for some \(\epsilon > 0\), \(B(\epsilon, x) \cap \text{aff}(C) \subset C\), where \(B(\epsilon, x)\) is the radius \(\epsilon\) ball centered at \(x\).