3. Binary classification


Problems to be handed in:

1. [Retracing our steps for concept learning problems]
   A concept learning problem in the agnostic case can be specified by a triplet \((X, \mathcal{P}, \mathcal{C})\) such that
   - \(X\) is the feature space
   - \(\mathcal{P} = \mathcal{P}_{X \times \{0,1\}}\) is a set of probability distributions on \(Z \equiv X \times \{0,1\}\)
   - \(\mathcal{C}\) is a set of subsets of \(X\).

   Recall that to simplify notation for analysis of ERM, we considered a class of functions \(f\) on \(Z\). For the concept learning problem with the usual 0-1 loss, each concept \(\mathcal{C}\) corresponds to a function \(\ell_C : Z \to \{0,1\}\) defined by \(\ell_C(z) = 1_{\{y \neq 1_{(x \in C)}\}}\). Let \(\mathcal{F}_C = \{\ell_C : C \in \mathcal{C}\}\). The goal of this problem is to prove the VC dimension of the set of classifiers \(\mathcal{C}\) is the same as the VC dimension of the set of functions induced in the abstract setup. That is, \(V(\mathcal{C}) = V(\mathcal{F}_C)\).

   Prove the following statements:
   
   (a) For any \(\{x_1, \ldots, x_n\} \subset X\), \(\mathcal{C}\) shatters \(\{x_1, \ldots, x_n\}\) if and only if \(\mathcal{F}_C\) shatters \(\{(x_1,0), \ldots, (x_n,0)\}\).
   (b) \(V(\mathcal{C}) \leq V(\mathcal{F}_C)\).
   (c) \(\mathcal{F}_C\) does not shatter any two point set of the form \(\{(x,0), (x,1)\}\).
   (d) \(V(\mathcal{C}) \geq V(\mathcal{F}_C)\).

2. [Illustration of the shifting algorithm used in a proof of the Sauer-Shelah lemma]

   The following matrix represents the projection of \(\mathcal{C} = \{[a,b] \subset \mathbb{R} : a \leq b\}\) to a five point subset of \(\mathbb{R}\):

   \[
   \begin{pmatrix}
   1 & 0 & 0 & 0 & 0 \\
   1 & 1 & 0 & 0 & 0 \\
   1 & 1 & 1 & 0 & 0 \\
   1 & 1 & 1 & 1 & 0 \\
   0 & 1 & 0 & 0 & 0 \\
   0 & 1 & 1 & 0 & 0 \\
   0 & 1 & 1 & 1 & 0 \\
   0 & 0 & 1 & 0 & 0 \\
   0 & 0 & 1 & 1 & 0 \\
   0 & 0 & 1 & 1 & 1 \\
   0 & 0 & 0 & 1 & 0 \\
   0 & 0 & 0 & 1 & 1 \\
   0 & 0 & 0 & 0 & 1 \\
   0 & 0 & 0 & 0 & 0
   \end{pmatrix}
   \]

   Each row of the matrix is a possible labeling of the five points. A set of \(k\) columns is shattered if the rows restricted to those columns include all \(2^k\) binary sequences of length \(k\). When the shifting algorithm is applied to a column, each 1 in the column is replaced by 0 if the resulting row is not present elsewhere in the matrix.

   (a) Which sets of columns are shattered by the given matrix? Indicate the matrix after both the first column and then the second column have been processed by the shifting algorithm, and identify which sets of columns are shattered by that matrix.

   (b) Repeat part (a) for the matrix:

   \[
   \begin{pmatrix}
   0 & 1 & 1 & 1 & 0 \\
   1 & 1 & 0 & 0 & 0 \\
   1 & 1 & 1 & 1 & 0 \\
   1 & 1 & 1 & 1 & 1 \\
   0 & 1 & 1 & 0 & 0 \\
   0 & 1 & 1 & 1 & 0 \\
   0 & 0 & 1 & 0 & 0 \\
   0 & 0 & 1 & 1 & 0 \\
   0 & 0 & 1 & 1 & 1 \\
   0 & 0 & 0 & 1 & 0 \\
   0 & 0 & 0 & 1 & 1 \\
   0 & 0 & 0 & 0 & 1 \\
   0 & 0 & 0 & 0 & 0
   \end{pmatrix}
   \]
3. **[Bayesian estimators]**

A Bayesian estimation problem of statistical decision theory can be represented by \((X, Y, U, P, \ell)\). As in the setup for model-free learning, \(X\) is a space of features, \(Y\) is a space of labels, \(U\) is a space of decision values, \(P\) is a known joint probability distribution on \(Z = X \times Y\) and \(\ell : Y \times U \rightarrow \mathbb{R}_{+}\) is a loss function. In contrast to the setup for model-free learning, only one probability distribution \(P\) is given and known, and a family of hypotheses \(\mathcal{F}\) is not needed. The Bayesian estimation problem is to find a function \(g : X \rightarrow U\) to minimize the expected loss (computed under \(P\)): \(\mathbb{E} [\ell(Y, g(X))]. \) If the function \(g\) were constrained to come from some function class \(\mathcal{F}\) it would be called an inductive bias constraint.

(a) Show that the expected loss is minimized by \(g^*(x) = \arg \min_{u \in U} \mathbb{E} [\ell(Y, u)|X = x]\). (Hint: Use the tower property of conditional expectations, conditioning on \(X\) to compute \(\mathbb{E} [\ell(Y, g(X))].\)

(b) Give a simplified description of \(g^*\) in case \(Y = U = \mathbb{R}^d\) for some \(d \geq 1\) and \(\ell(s, y) = \|s - y\|^2\). (Assume \(\mathbb{E} [\|Y\|^2] < \infty\).)

(c) Give a simplified description of \(g^*\) in case \(Y\) has finite cardinality, \(Y = U\), and \(\ell\) is the 0-1 loss function: \(\ell(y, u) = 1_{\{y \neq u\}}\).

4. **[VC dimension of combined classifiers using hard thresholding]**

Let \(\mathcal{G}\) denote the set of interval classifiers \(g : \mathbb{R} \rightarrow \{1, -1\}\). Each \(g \in \mathcal{G}\) has the form \(g(x) = \text{sgn}((x - a)(b - x))\) for \(a \leq b \in \mathbb{R}\), where \(\text{sgn}(u) = 1_{\{u \geq 0\}} - 1_{\{u < 0\}}\).

(a) What is the VC dimension, \(V(\mathcal{G})\), and what is the resulting upper bound on the maximum Rademacher complexity for a sample of size \(n\): \(R_n(\mathcal{G}(x^n))\) for samples \(\{x_1, \ldots, x_n\} \subset \mathbb{R}\), for \(n \geq 1\) (obtained from the finite class lemma, Sauer-Shelah lemma, and \((\frac{n}{\sqrt{d}}) \leq (n + 1)^d\) ?

(b) Let \(\mathcal{G}_1\) be the set of classifiers of the form \(g(x) = \text{sgn} \left( \sum_{i=1}^{N} c_i g_i(x) \right)\), where \(N \geq 1\), \(g_i \in \mathcal{G}\) for \(i \in [N]\), and \((c_1, \ldots, c_N)\) is a probability vector. Thus, \(g\) can be the result of comparing a convex combination of arbitrarily many simple interval classifiers to the threshold 0. In short, \(\mathcal{G}_1 = \text{sgn}(\text{conv}(\mathcal{G}))\). Identify the VC dimension of \(\mathcal{G}_1\) and the Rademacher average for \(n\) sample points, \(R_n(\mathcal{G}_1(x^n))\) (with notation as in part (a)).

5. **[Using a weak learning algorithm to find good combinations of simple classifiers]**

In this problem you will derive the update equations used by the popular and effective AdaBoost, or adaptive boosting, algorithm. Let \(\mathcal{G}\) be a set of base classifiers mapping \(X\) to the label set \(\{-1, 1\}\). (Think of them as being very simple.) Focus on a classifier of the form \(\text{sgn}(f)\) where \(f(x) = \sum_{j=1}^{t} w_j g_j(x)\), for some \(t \geq 1\), \(g_j \in \mathcal{G}\) for \(j \in [t]\), and coefficients \(w_1, \ldots, w_t\). We also consider surrogate loss for the exponential penalty function, \(\varphi(x) = e^x\). If \(f\) is used on a fresh sample \((x, y)\), the 0-1 loss is given by \(\ell(y, f(x)) = 1_{\{y \neq \text{sgn}(f(x))\}}\) and the surrogate loss is given by \(\ell_{\varphi}(x, y) = \exp(-y f(x))\).

Let \(z^n = ((x_1, y_1), \ldots, (x_n, y_n))\) be a set of labeled data points, and suppose \(D^n = (D_1, \ldots, D_n)\) is a vector of positive weights for the \(n\) data points. Then \((z^n, D^n)\) represents a weighted sample set. Suppose there is a learning algorithm \(WL\) (for “weak learner”) that maps weighted sample sets to base classifiers: \(WL : Z^n \times (0, \infty)^n \rightarrow \mathcal{G}\). Specifically, suppose the weak learner is a weighted ERM algorithm for the base class \(\mathcal{G}\) and 0-1 loss:

\[
WL(z^n, D^n) = \arg \min_{g \in \mathcal{G}} \frac{1}{n} \sum_{i=1}^{n} D_i 1_{\{y_i \neq g(x_i)\}}.
\]

(1)

The learner is considered to be weak because the base class \(\mathcal{G}\) is typically very simple. The idea of AdaBoost is to use such a weak learner to select the weights \(w_j\)’s and classifiers \(g_j\)’s to make the combined classifier \(\text{sgn}(f)\) strong.
(a) Show that (1) is equivalent to WL finding \( g \in \mathcal{G} \) to minimize the weighted empirical surrogate loss for hypothesis \( w g \) for a fixed \( w > 0 \). That is, show that (1) is equivalent to:

\[
WL(z^n, D^n) = \arg \min_{g \in \mathcal{G}} \frac{1}{n} \sum_{i=1}^{n} D_i \exp(-y_i w g(x_i)).
\]

(Hint: Remember that functions in \( \mathcal{G} \) are \( \{-1, 1\} \)-valued.)

(b) Suppose \((X, Y)\) represents a random labeled sample with values in \( X \times \{-1, 1\} \), and suppose \( g \) is a classifier; \( g : X \rightarrow \{-1, 1\} \). Multiplying \( g \) by a scaler weight \( w \) yields the weighted classifier \( w g \), and the expected surrogate loss of the weighted classifier for the exponential penalty function is \( \mathbb{E} [e^{-Y(w g(X))}] \). Find the weight \( w \) that minimizes the expected surrogate loss for \( w g \). Express your answer in terms of \( \epsilon \), where \( \epsilon \) is the expected 0-1 loss for \( g \) : \( \epsilon = \mathbb{P} \{ Y \neq g(X) \} \). Also, show that if \( \epsilon \leq \frac{1}{2} - \gamma \) for some \( \gamma > 0 \), then \( \min_w \mathbb{E} [e^{-Y(w g(X))}] = 2 \sqrt{\epsilon (1 - \epsilon)} \leq e^{-2 \gamma^2} \).

(c) The empirical surrogate loss for \( f \) is given by

\[
A_{n, \varphi}(f) = \frac{1}{n} \sum_{i=1}^{n} \exp \left( -y_i \sum_{j=1}^{t} w_j g_j(x_i) \right).
\]

AdaBoost adds new classifiers to \( f \) one at a time, so to examine one step of the algorithm, suppose \( w_1, \ldots, w_{t-1} \) and \( g_1, \ldots, g_{t-1} \) are already given. The idea is to use a greedy approach: select \( w_t \) and \( g_t \) to minimize \( A_{n, \varphi}(f) \). Show that such minimization can be decomposed into two steps:

Step one: Find \( g_t \in \mathcal{G} \) to minimize the weighted empirical probability of error, \( \epsilon_t \), using weights

\[
D_i^{(t)} = \exp \left( -y_i \sum_{j=1}^{t-1} w_j g_j(x_i) \right).
\]

(For \( t = 1 \), \( D_i^{(1)} = 1 \). For the definition of weighted empirical probability of error, the weights need to be normalized to sum to one.)

Step two: Find \( w_t \).

Explain why the weights \( D_1^{(t)}, \ldots, D_n^{(t)} \) given in (2) are appropriate for the first step, and then describe the choice of \( w_t \). (You can assume \( w_t > 0 \), which will be true if \( \epsilon_t < 1/2 \), meaning that \( g_t \) does better than random guessing.)

(d) Let \( f^{(t)} \) denote the classifier \( f \) constructed by the algorithm after \( t \) terms have been found for \( f \).

Show that \( A_{n, \varphi}(f^{(t)}) = A_{n, \varphi}(f^{(t-1)}) 2 \sqrt{\epsilon_t (1 - \epsilon_t)} \) for \( t \geq 1 \).

Hence, if the WL has the guarantee \( \epsilon_t \leq \frac{1}{2} - \gamma \) for all \( t \) (i.e. WL can find a classifier with weighted empirical 0-1 loss less than or equal to \( \frac{1}{2} - \gamma \) for any weighted data sample), then by induction on \( t \), \( A_{n, \varphi}(f^{(t)}) \) decreases at an rate \( e^{-2t \gamma^2} \). That is, the empirical surrogate risk, and hence also the 0-1 risk, converges to zero. However, as seen in problem 4, the VC dimension of the space of all combined classifiers is infinite, so that overfitting may occur for large \( t \).

6. [A view of AdaBoost in SK learn]

The python programming problem for this problem set is explained within the .ipynb file. You can see a static version at http://nbviewer.jupyter.org/url/courses.engr.illinois.edu/ece543/ece543_PythonProblem3.ipynb?flush_cache=true and download the ipynb file from the static version or directly from https://courses.engr.illinois.edu/ece543/ece543_PythonProblem3.ipynb.