COMS 4771 HW2 (Fall 2022)

Due: Sun Oct 23, 2022 at 11:59pm

This homework is to be done **alone**. No late homeworks are allowed. To receive credit, a typesetted copy of the homework pdf must be uploaded to Gradescope by the due date. You must show your work to receive full credit. Discussing possible approaches for solutions for homework questions is encouraged on the course discussion board and with your peers, but you must write your own individual solutions and **not** share your written work/code. You must cite all resources (including online material, books, articles, help taken from/given to specific individuals, etc.) you used to complete your work.

1 Learning DNFs with kernel perceptron

Suppose that we have $S = \{(x^{(i)}, y^{(i)})\}_{i=1}^n$ with $x^{(i)} \in \{0,1\}^d$ and $y^{(i)} \in \{-1, 1\}$. Let $\varphi : \{0,1\}^d \to \{0,1\}$ be a "target function" which "labels" the points. Additionally, assume that φ is a DNF formula (i.e. φ is a disjunction of conjunctions, a boolean "OR" of a collection of boolean "AND"s). "Labelling" the points simply means $\mathbf{1}[y^{(i)} = 1] = \varphi(x^{(i)})$.

For example, let $\varphi(x) = (x_1 \wedge x_2) \vee (x_1 \wedge \bar{x}_2 \wedge x_3)$ (where x_i denotes the ith entry of x), $x^{(i)} = \begin{pmatrix} 1 & 0 & 1 \end{pmatrix}^\mathsf{T}$, and $x^{(j)} = \begin{pmatrix} 1 & 0 & 0 \end{pmatrix}^\mathsf{T}$. Then, we would have $\varphi(x^{(i)}) = 1$ and $\varphi(x^{(j)}) = 0$, and thus $y^{(i)} = 1$ and $y^{(j)} = -1$.

(i) Give an example target function φ (make sure it is a DNF formula) and set S such that the data is not linearly separable.

Part (i) clearly shows that running the perceptron algorithm on S cannot work in general since the data does not need to be linearly separable. However, we can try to use a feature transformation and the kernel trick to linearize the data and thus run the kernelized version of the perceptron algorithm on these datasets.

Consider the feature transformation $\phi : \{0,1\}^d \to \{0,1\}^{3^d}$ which maps a vector x to the vector of all the conjunctions of its entries or of their negations. For example, if $d = 2$ then $\phi(x) = \begin{pmatrix} 1 & x_1 & x_2 & \bar{x}_1 & \bar{x}_2 & x_1 \wedge x_2 & x_1 \wedge \bar{x}_2 & \bar{x}_1 \wedge x_2 & \bar{x}_1 \wedge \bar{x}_2 \end{pmatrix}^\mathsf{T}$ (note that 1 can be viewed as the empty conjunction, i.e. the conjunction of zero literals).

Let $K : \{0,1\}^d \times \{0,1\}^d \to \mathbb{R}$ be the kernel function associated with ϕ (i.e. for $a, b \in \{0,1\}^d$: $K(a,b) = \phi(a) \cdot \phi(b)$). Note that the naive approach of calculating $K(a,b)$ (simply calculating $\phi(a)$ and $\phi(b)$ and taking the dot product) takes time $\Theta(3^d)$.

Also let $w^* \in \{0,1\}^{3^d}$ be such that $w_1^* = -0.5$ (this is the entry which corresponds to the empty conjunction, i.e. $\forall x \in \{0,1\}^d : \phi(x)_1 = 1$) and $\forall i > 1 : w_i^* = 1$ iff the ith conjunction is one of the conjunctions of φ. For example, in the above case where $d = 2$ and $\phi(x) = \begin{pmatrix} 1 & x_1 & x_2 & \overline{x_1} & \overline{x_2} & x_1 \wedge x_2 & x_1 \wedge \overline{x_2} & \overline{x_1} \wedge x_2 & \overline{x_1} \wedge \overline{x_2} \end{pmatrix}^\mathsf{T}$ and letting $\varphi(x) = (x_1 \wedge x_2) \vee$

$(\overline{x_1})$, we would have:

$$w^* = \begin{pmatrix} -0.5 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \end{pmatrix}^\mathsf{T}$$

(ii) Find a way to compute $K(a, b)$ in $O(d)$ time.

(iii) Show that w^* linearly separates $\phi(S)$ (where $\phi(S)$ is shorthand for $\{(\phi(x^{(i)}), y^{(i)})\}_{i=1}^n$) and find a lower bound for the margin γ with which it separates the data. Remember that $\gamma = \min_{(\phi(x^{(i)}),y^{(i)})\in\phi(S)} y_i\left(\frac{w^*}{\|w^*\|} \cdot \phi(x^{(i)})\right)$. Your lower bound should depend on s, the number of conjunctions in φ.

(iv) Find an upper bound on the radius R of the dataset $\phi(S)$. Remember that

$$R = \max_{(\phi(x^{(i)}),y^{(i)})\in\phi(S)} \|\phi(x^{(i)})\|.$$

(v) Use parts (ii), (iii), and (iv) to show that we can run kernel perceptron efficiently on this transformed space in which our data is linearly separable (show that each iteration only takes $O(nd)$ time per point) but that unfortunately the mistake bound is very bad (show that it is $O(s2^d)$).

There are ways to get a better mistake bound in this same kernel space, but then the running time becomes very bad (exponential). It is open whether there are ways to get both polynomial mistake bound and running time.

2 Training error vs. Test error

In this problem, we will study why we expect training error to be smaller than test error.

Suppose $(x_1, y_1), \ldots, (x_n, y_n), (x, y)$ are i.i.d. samples taking values in $\mathbb{R}^d \times \mathbb{R}$. Let $\mathcal{R}$ denote the squared error of a (linear) classifier, defined as

$$\mathcal{R}(w) := \mathbb{E}\big[(w \cdot x - y)^2\big]$$

for any $w \in \mathbb{R}^d$, and let $\hat{\mathcal{R}}$ denote the training error based on $(x_1, y_1), \ldots, (x_n, y_n)$, so

$$\hat{\mathcal{R}}(w) := \frac{1}{n}\sum_{i=1}^{n}(w \cdot x_i - y_i)^2$$

for any $w \in \mathbb{R}^d$.

Let $\hat{w}$ denote the squared training error minimizing decision boundary based on samples $(x_1, y_1), \ldots, (x_n, y_n)$, so $\hat{\mathcal{R}}(\hat{w}) \le \hat{\mathcal{R}}(w)$ for all $w \in \mathbb{R}^d$. (You may assume that $\hat{w}$ is unique.)

Prove that

$$\mathbb{E}\Big[\hat{\mathcal{R}}(\hat{w})\Big] \le \mathbb{E}\Big[\mathcal{R}(\hat{w})\Big],$$

where the expectation on both sides is taken with respect to $(x_1, y_1), \ldots, (x_n, y_n)$.

Hint: Let $(\tilde{x}_1, \tilde{y}_1), \ldots, (\tilde{x}_n, \tilde{y}_n)$ be another i.i.d. random sample, independent of $(x_1, y_1), \ldots, (x_n, y_n)$, but having the same distribution as (x, y). Then

$$\mathcal{R}(w) = \mathbb{E}\Big[\underbrace{\frac{1}{n}\sum_{i=1}^{n}(w \cdot \tilde{x}_i - \tilde{y}_i)^2}_{=:\tilde{\mathcal{R}}(w)}\Big], \quad w \in \mathbb{R}^d.$$

Now let $\tilde{w}$ denote the squared training error minimizing decision boundary based on samples $(\tilde{x}_1, \tilde{y}_1), \ldots, (\tilde{x}_n, \tilde{y}_n)$ (again, assume uniqueness). How do $\mathbb{E}[\hat{\mathcal{R}}(\hat{w})]$, $\mathbb{E}[\tilde{\mathcal{R}}(\tilde{w})]$, and $\mathbb{E}[\hat{\mathcal{R}}(\hat{w})]$ compare?

3 Making data linearly separable by feature space mapping

Consider the infinite dimensional feature space mapping

$$\begin{aligned}\Phi_\sigma &: \mathbb{R} \to \mathbb{R}^\infty \\ &x \mapsto \Big(\mathbf{1}\big[|\alpha - x| < \sigma\big] \cdot \exp\big(-1/(1-(|\alpha - x|/\sigma)^2)\big)\Big)_{\alpha \in \mathbb{R}}.\end{aligned}$$

(It may be helpful to sketch the function $f(\alpha) := \mathbf{1}\big[|\alpha| < 1\big] \cdot \exp\big(-1/(1-\alpha^2)\big)$ for understanding the mapping and answering the questions below.)

(i) Show that for any n distinct points $x_1, \ldots, x_n$, there exists a $\sigma > 0$ such that the mapping Φ_σ can linearly separate *any* binary labeling of the n points.

(ii) Show that one can efficiently compute the dot products in this feature space, by giving an analytical formula for $\Phi_\sigma(x) \cdot \Phi_\sigma(x')$ for arbitrary points x and x'.

(iii) Given an input space X and a feature space mapping ϕ that maps elements from X to a (possibly infinite dimensional) inner product space V, let $K : X \times X \to \mathbb{R}$ be a kernel function that can efficiently compute the inner products in V, that is, for any $x, x' \in X$, $K(x, x') = \phi(x) \cdot \phi(x')$.

Consider a binary classification algorithm that predicts the label of an unseen instance according to the class with the closest average. Formally, given a training set $S = (x_1, y_1), \ldots, (x_m, y_m)$, for each $y \in \{\pm 1\}$ define

$$c_y := \frac{1}{m_y} \sum_{i : y_i = y} \phi(x_i),$$

where $m_y = |\{i : y_i = y\}|$. Assume that m_+ and m_- are nonzero. Then, the algorithm outputs the following decision rule:

$$h(x) := \begin{cases} +1 & \|\phi(x) - c_+\| \leq \|\phi(x) - c_-\| \\ -1 & \text{otherwise.} \end{cases}$$

(a) Let $w := c_+ - c_-$ and let $b = \frac{1}{2}(\|c_-\|^2 - \|c_+\|^2)$. Show that

$$h(x) = \text{sign}(\langle w, \phi(x) \rangle + b).$$

(b) Show that $h(x)$ can be expressed via $K(\cdot, \cdot)$, without accessing individual entries of $\phi(x)$ or w, thus showing that $h(x)$ is efficiently computable.