1: True or False. (TA:- Abhinav Maurya)

State whether true (with a brief reason) or false (with a contradictory example). Credit will be
granted only if your reasons/counterexamples are correct.

(a) If the VC Dimension of a set of classification hypotheses is $\infty$, then the set of classifiers can
achieve 100% training accuracy on any dataset.

True
If the VC Dimension of a set of classification hypotheses is $\infty$, it can shatter some dataset $D$ at any
given dataset size. Finding a mapping from a given dataset $D'$ to $D$ allows 100% training accuracy
on any dataset.

[2 points]

(b) There is no actual set of classification hypotheses useful in practical machine learning that has
VC Dimension $\infty$.
False
Kernelized SVM with an arbitrary number of support vectors has VC Dim $\infty$ 

\[2 \text{ points}\]

(c) VC Dimension of the set of all decision trees (defined on a given set of real-valued features) is finite.

False
The set of axis-aligned rectangles can shatter any dataset of arbitrary size consisting of distinct points.

\[2 \text{ points}\]

(d) Since the true risk is bounded by the empirical risk, it is a good idea to minimize the training error as much as possible.

False
Reducing the empirical risk by increasing the complexity of the classifier makes the PAC bound looser.

\[2 \text{ points}\]

(e) PAC learning bounds help you estimate the number of samples needed to reduce the discrepancy between the true risk and empirical risk to an arbitrary constant with high probability.

True
From the construction of PAC learning bounds

\[2 \text{ points}\]

(f) If the VC Dimension of a set of classification hypotheses is $K$, then no algorithm can have a mistake bound that is strictly less than $K$.

True
\[VCDim(H) \leq Opt(H)\] where $Opt(H)$ indicates the mistake bound. See recitation slides.

\[2 \text{ points}\]

(g) SVMs with a gaussian kernel have VC Dimension equal to $n + 1$ where $n$ is the number of support vectors.

True
The SVM prediction function is linear in $n$ features where the features are evaluated as the kernel function with respect to each support vector.
(h) As the degree of the polynomial kernel increases, the VC Dimension of the set of classification hypotheses increases.

True  
Consider the expansion of the polynomial kernel of a particular degree. As the degree increases, the number of features available increases as well.

(i) VC Dimensions of the sets of classification hypotheses induced by logistic regression and linear SVM (learnt on the same set of features) are different.

False  
They both induce linear separators. Since the set of classification hypotheses is the same, the VC Dim is the same.

(j) VC Dimension depends on the dataset we use for shattering.

False  
If you cannot shatter any dataset of size \((n + 1)\) but can shatter some dataset of size \(n\), then VC Dim is \(n\). It is independent of the dataset because it is one less than the smallest dataset size at which it is not possible to shatter any possible dataset.
2: PAC learning for conjunctions of boolean literals. (TA:- Ying Yang)

Consider a function that takes $n$-bit binary inputs ($\mathbf{x} = (x_1, x_2, \cdots, x_n)$, $x_i \in \{0, 1\}$ ) and output binary responses. This function is a list of conjunctions of boolean literals, and the list can include $x_i$ or $\neg x_i$, or neither of them, but not both of them, for $i = 1, \cdots, n$. One example of such a function is

$$h(\mathbf{x}) = x_1 \land \neg x_2 \land x_3.$$ 

We are given data that can be perfectly explained by at least one such function. Suppose we are also given an algorithm that learns a function, which has zero training error on finite data samples, how many training examples do we need to guarantee, with probability at least 95%, that the true error rate of our learned function is $< 5\%$? Use $n = 10$ in your computation.

[5 points]

Solution

There are three possibilities for each $x_i$,

1. $x_i$ is in the conjunction list
2. $\neg x_i$ is in the conjunction list
3. neither $x_i$ nor $\neg x_i$ is in the conjunction list

Therefore, there are $3^n$ distinct combinations for all $n$ inputs, and $|H| = 3^n$. Now let’s apply the inequality of PAC learning for a finite $H$. To make sure with probability $\ge 1 - \delta$, the true error of our learned function $\le \epsilon$, we need

$$m \ge \frac{1}{\epsilon} [\ln |H| + \ln(1/\delta)]$$

$\delta = 1 - 0.95 = 0.05, \epsilon = 0.05, |H| = 3^n = 3^{10}$, therefore

$$m \ge \frac{1}{0.05} [\ln 3^{10} + \ln(1/0.05)] = \frac{1}{0.05} [10 \ln 3 + \ln(1/0.05)] = 279.6$$

Therefore $m \ge 280$.

Note: Among the $3^n$ functions in $H$, there is one, where none of the inputs is included, (i.e, for each $x_i$, neither $x_i$ nor $\neg x_i$ is included), and there are $2n$ functions having no conjunctions (e.g. $h(\mathbf{x}) = x_i, or h(\mathbf{x}) = \neg x_i$). If you did not count these functions, you are also correct. Notice that compared with the total number $3^n$ (exponential), these $2n + 1$ functions are only a small portion (polynomial).
3: VC-dimensions of binary classifiers. (TA: Ying Yang)

Write the VC-dimensions of the following families of binary classifiers. Explain your results with examples that can or cannot be shattered.

1. \( f : \mathbb{R} \to \{0, 1\}, f(x) = \begin{cases} 1 & \text{if } x \in [a, b] \\ 0 & \text{otherwise} \end{cases} \) where \( a \) and \( b \) are any real constants, such that \( a < b \).

2. The functions in 1 and the functions that flip the outputs of the functions in 1.

3. \( f : \mathbb{R}^2 \to \{0, 1\}, f(x) = \begin{cases} 1 & \text{if } ||x||^2 < c \\ 0 & \text{otherwise} \end{cases} \) where \( x \in \mathbb{R}^2, c > 0 \in \mathbb{R} \)

\([9 \text{ points}]\)

Solution

1. \( VC(H) = 2 \), see Figure 1

2. Here, \( H = H_1 \cup H_2 \), where \( H_1 \) is the function family in 3.1, \( H_2 \) is the following function family:
\[ f : \mathbb{R} \to \{0, 1\}, f(x) = \begin{cases} 0 & \text{if } x \in [a, b] \\ 1 & \text{otherwise} \end{cases} \]. In this function family, when you classify a set of points with labelling, you can either classify points inside the interval \([a, b]\) as positive, points outside as negative, or classify points inside the interval \([a, b]\) as negative, points outside as positive. \( VC(H) = 3 \), see Figure 2

3. The decision boundary of this family is a circle centered at the origin on a two dimensional space. When distance of a point to the origin is smaller than \( \sqrt{c} \), we classify it as positive. \( VC(H) = 1 \), see Figure 3
**VC(H) ≥ 2**

For all of the four labellings of two points on the real line, we can classify them using at least one function from the family.

![Diagram showing four labellings of two points](image)

---

**VC(H) < 3**

Consider ANY three points on the real line

![Diagram showing three points](image)

there always exists a labelling that can not be perfectly classified, such as

Because we can only label points within the interval as positive, and we can not find an interval to include the positive points and leave the middle negative point out.

---

**Figure 1: 3.1.**

---

**VC(H) ≥ 3**

For all of the eight labellings of three points on the real line, we can perfectly classify them using at least one function from the family H

![Diagram showing eight labellings of three points](image)

Classify as + if inside the interval

Classify as - if inside the interval

---

**VC(H) < 4**

Consider ANY four points on the real line

![Diagram showing four points](image)

there always exists a labelling for the four points that can not be perfectly classified, such as

Because we can not find an interval such that all points inside it have one label, and all points outside it have the other label.

---

**Figure 2: 3.2.** Note that for some cases on the left, we can use more than one function to classify them, but only is shown.
For a one-point set, we can classify the each of the two labelling perfectly.

Consider ANY two points on the plane $x_1$ and $x_2$, assume $||x_1||^2 \leq ||x_2||^2$, we label $x_2$ as positive, $x_1$ as negative.

Then we can not find a circle centered at the origin and only include $x_2$.

Figure 3: 3.3.
4: Learning theory of SVMs with quadratic Kernels. (TA:- Ying Yang)

Given a family of support vector machines with a quadratic kernel \( k(x_1, x_2) = (x_1^T x_2)^2 \). The inputs \( x \in \mathbb{R}^n \), and the output is binary.

1. What is the VC-dimension of this family? 

   \[4 \text{ points}\]

2. Now we are given data that can be perfectly classified by one SVM in this family. If we are trying to train an SVM in this family, how many training examples do we need to guarantee that the true error rate of the trained SVM is < 5% with probability at least 95%? Use \( n = 10 \) in your computation.

   \[2 \text{ points}\]

Solution

1. The kernel \( k(x_1, x_2) = (x_1^T x_2)^2 \) is essentially the inner product of a transform of the input \( \phi(\cdot) \). That is, \( k(x_1, x_2) = \phi(x_1)^T \phi(x_2) \), where \( x_1, x_2 \in \mathbb{R}^n \). Let \( x_1^{(j)} \) denote the jth element of \( x_1 \). Then we have

   \[
k(x_1, x_2) = (x_1^T x_2)^2
   = (x_1^{(1)} x_2^{(1)} + x_1^{(2)} x_2^{(2)} + \cdots + x_1^{(n)} x_2^{(n)})^2
   = \sum_{i=1}^{n} (x_1^{(i)})^2 (x_2^{(i)})^2 + 2 \sum_{i \leq j} x_1^{(i)} x_2^{(j)} x_1^{(j)} x_2^{(i)}
   = \sum_{i=1}^{n} (x_1^{(i)})^2 (x_2^{(i)})^2 + 2 \sum_{i \leq j} x_1^{(i)} x_2^{(j)} x_1^{(j)} x_2^{(i)}
   = \phi(x_1)^T \phi(x_2)
\]

   \( \phi(x) = \left( (x^{(1)})^2, (x^{(2)})^2, \cdots (x^{(n)})^2, \sqrt{2} x^{(1)} x^{(2)}, \sqrt{2} x^{(1)} x^{(3)} \cdots, \sqrt{2} x^{(n-1)} x^{(n)} \right)^T \)

   The decision boundary of SVM with such a kernel is \( w^T \phi(x) + b = 0 \), which is a linear classifier in the dimension of \( \phi(x) \) \( (n + n(n - 1)/2 = n(n + 1)/2) \). We know that the VC dimension of linear classifiers in the \( d \)-dimensional space is \( d + 1 \), therefore the VC dimension here is \( n(n + 1)/2 + 1 \).

2. We use the inequality in PAC learning when we know the VC dimension,

   \[
m \geq \frac{1}{\epsilon} (4 \log_2(2/\delta) + 8 \text{VC}(H) \log_2(13/\epsilon))
\]

   \( \epsilon = 0.05, \delta = 1 - 0.95 = 0.05, n = 10, \text{VC}(H) = 10 \star 11/2 + 1 = 56 \)

   \[
m \geq \frac{1}{0.05} (4 \log_2(2/0.05) + 8 \star 56 \log_2(13/0.05)) = 72306.17
\]

   \[
m \geq 72307
\]
Some students may have trouble understanding how a kernel corresponds to a mapping \( \phi(x) \) of input \( x \) to a higher dimensional space. Some students are also confused about how we got to the dual problem of SVM. Here is some insight.

Let’s consider the linear SVM with no slack variables. Let \( x \) be the input, \( y \) be the label, \( x_i, y_i \) be the data points. The original problem (primal problem) is

$$\min \frac{1}{2} w^T w$$

s.t. \( y_i(w^T x_i + b) \geq 1 \)

To solve it, we can introduce Lagrange multipliers \( \alpha_i \geq 0 \). The Lagrange function is

$$L(w, b, \alpha_i) = \frac{1}{2} w^T w - \sum_i \alpha_i (y_i(w^T x_i + b) - 1)$$

To get the best \( w, b \), we need the KKT conditions to hold,

$$\frac{\partial L}{\partial w} = w - \sum_i \alpha_i y_i x_i = 0$$

$$\frac{\partial L}{\partial b} = \sum_i \alpha_i y_i = 0$$

$$\frac{\partial L}{\partial \alpha_i} = y_i(w^T x_i + b) - 1 = 0 \text{ and } \alpha_i > 0$$

or \( y_i(w^T x_i + b) - 1 > 0 \) and \( \alpha_i = 0 \)

Note that \( \alpha_i > 0 \) corresponds to the support vectors. For non-support vectors, \( \alpha_i = 0 \). Now we consider the dual of this optimization problem

$$\max_{\alpha_i} \min_{w,b} L(w, b, \alpha_i)$$

To obtain \( \max_{w,b} L(w, b, \alpha_i) \), we plug in \( w \) that satisfy the KKT conditions,

$$w = \sum_i \alpha_i y_i x_i$$

Then

$$\min_{w,b} L(w, b, \alpha_i) = \frac{1}{2} w^T w - \sum_i \alpha_i (y_i(w^T x_i + b) - 1)$$

$$= \frac{1}{2} w^T w - \sum_i \alpha_i (y_i(w^T x_i + b)) + \sum_i \alpha_i$$

$$= \sum_i \alpha_i + \frac{1}{2} w^T w - \sum_i \alpha_i (y_i(w^T x_i + b))$$

$$= \sum_i \alpha_i + \frac{1}{2} \left( \sum_i \alpha_i y_i x_i \right)^T \left( \sum_i \alpha_i y_i x_i \right) - \sum_i \alpha_i y_i \left( \sum_j \alpha_j y_j x_j^T \right) x_i - \left( \sum_i \alpha_i y_i \right) b$$

$$= \sum_i \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j x_i^T x_j$$
where $\sum_i \alpha_iy_i = 0$, with the constraint $\alpha_i \geq 0$, the dual is written as

$$
\max_{\alpha_i} \left( \sum_i \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i\alpha_jy_iy_jx_ix_j \right)
$$

s.t. \quad \sum_i \alpha_iy_i = 0

$$
\alpha_i \geq 0
$$

The solution to the dual problem is also the solution to the primal problem. So sometimes we solve the dual problem instead of the primal problem.

When introducing the kernels, we just replace $x_i^T x_j$ with $\phi(x_i)^T \phi(x_j) = k(x_j, x_j)$, this is the same as replacing $x_i$'s with $\phi(x_i)$s. You can verify that all the derivations above remains the same when you replace $x_i$'s with $\phi(x_i)$s.

The decision boundary without kernel was $w^T x + b = 0$, or $\sum_i \alpha_iy_i x_i^T x + b = 0$ (using $w = \sum_i \alpha_iy_i x_i$). With kernels, the boundary is $w^T \phi(x) + b = 0$, or $\sum_i \alpha_iy_i \phi(x_i)^T \phi(x) + b = \sum_i \alpha_iy_i k(x_i, x) + b$ (using $w = \sum_i \alpha_iy_i \phi(x_i)$).

So each kernel $k$ corresponds to a transform of the input $\phi(\cdot)$. Often, the kernel itself is much easier to compute than the transform, so we can use the dual problem to solve the SVM efficiently.

**Total: 40**