Midterm Exam Solutions

CMU 10-601: Machine Learning (Spring 2016)

Feb. 29, 2016

Name: ______________________________

Andrew ID: ______________________________

START HERE: Instructions

- This exam has 17 pages and 5 Questions (page one is this cover page). Check to see if any pages are missing. Enter your name and Andrew ID above.
- You are allowed to use one page of notes, front and back.
- Electronic devices are not acceptable.
- Note that the questions vary in difficulty. Make sure to look over the entire exam before you start and answer the easier questions first.

Question	Point	Score
1	20	
2	20	
3	20	
4	20	
5	20	
Extra Credit	14	
Total	114	

1 Naive Bayes, Probability, and MLE [20 pts. + 2 Extra Credit]

1.1 Naive Bayes

You are given a data set of 10,000 students with their sex, height, and hair color. You are trying to build a classifier to predict the sex of a student, so you randomly split the data into a training set and a testing set. Here are the specifications of the data set:

- sex $\in$ {male,female}
- height $\in$ [0,300] centimeters
- hair $\in$ {brown, black, blond, red, green}
- 3240 men in the data set
- 6760 women in the data set

Under the assumptions necessary for Naive Bayes (not the distributional assumptions you might naturally or intuitively make about the dataset) answer each question with **T** or **F** and **provide a one sentence explanation of your answer**:

(a) [2 pts.] **T or F:** As height is a continuous valued variable, Naive Bayes is not appropriate since it cannot handle continuous valued variables.

Solution: False. Naive Bayes can handle both continuous and discrete values as long as the appropriate distributions are used for conditional probabilities. For example, Gaussian for continuous and Bernoulli for discrete

(b) [2 pts.] **T or F:** Since there is not a similar number of men and women in the dataset, Naive Bayes will have high test error.

Solution: False. Since the data was randomly split, the same proportion of male and female will be in the training and testing sets. Thus this discrepancy will not affect testing error.

(c) [2 pts.] **T or F:** $P(\texttt{height}|\texttt{sex}, \texttt{hair}) = P(\texttt{height}|\texttt{sex})$.

Solution: True. This results from the conditional independence assumption required for Naive Bayes.

(d) [2 pts.] **T or F:** $P(\texttt{height}, \texttt{hair}|\texttt{sex}) = P(\texttt{height}|\texttt{sex})P(\texttt{hair}|\texttt{sex})$.

Solution: True. This results from the conditional independence assumption required for Naive Bayes.

1.2 Maximum Likelihood Estimation (MLE)

Assume we have a random sample that is Bernoulli distributed $X_1, \ldots, X_n \sim \text{Bernoulli}(\theta)$. We are going to derive the MLE for θ. Recall that a Bernoulli random variable X takes values in $\{0, 1\}$ and has probability mass function given by

$$P(X; \theta) = \theta^X (1 - \theta)^{1-X}.$$

(a) [2 pts.] Derive the likelihood, $L(\theta; X_1, \ldots, X_n)$.

Solution:

$$\begin{aligned} L(\theta; X_1, \ldots, X_n) &= \prod_{i=1}^{n} p(X_i; \theta) \\ &= \prod_{i=1}^{n} \theta^{x_i}(1-\theta)^{1-x_i} \\ &= \theta^{\sum_{i=1}^{n} x_i}(1-\theta)^{n-\sum_{i=1}^{n} x_i}. \end{aligned}$$

Either of the final two steps are acceptable.

(b) [2 pts.] Derive the following formula for the log likelihood:

$$\ell(\theta; X_1, \ldots, X_n) = \left(\sum_{i=1}^{n} X_i\right) \log(\theta) + \left(n - \sum_{i=1}^{n} X_i\right) \log(1-\theta).$$

Solution:

$$\begin{aligned} l(\theta; X_1, \ldots, X_n) &= \log L(\theta; X_1, \ldots, X_n) \\ &= \log \left[\theta^{\sum_{i=1}^{n} x_i}(1-\theta)^{n-\sum_{i=1}^{n} x_i}\right] \\ &= (\sum_{i=1}^{n} x_i) \log(\theta) + (n - \sum_{i=1}^{n} x_i) \log(1-\theta) \end{aligned}$$

(c) **Extra Credit:** [2 pts.] Derive the following formula for the MLE: $\hat{\theta} = \frac{1}{n}(\sum_{i=1}^{n} X_i)$.

Solution: To find the MLE we solve $\frac{d}{d\theta}\ell(\theta; X_1, \ldots, X_n) = 0$. The derivative is given by

$$\begin{aligned} \frac{d}{d\theta}\ell(\theta; X_1, \ldots, X_n) &= \frac{d}{d\theta}\left[(\sum_{i=1}^{n} x_i) \log(\theta) + (n - \sum_{i=1}^{n} x_i) \log(1-\theta)\right] \\ &= \frac{\sum_{i=1}^{n} x_i}{\theta} - \frac{n - \sum_{i=1}^{n} x_i}{1-\theta} \end{aligned}$$

Next, we solve $\frac{d}{d\theta}\ell(\theta; X_1, \ldots, X_n) = 0$:

$$\begin{aligned} &\frac{\sum_{i=1}^{n} x_i}{\theta} - \frac{n - \sum_{i=1}^{n} x_i}{1-\theta} = 0 \\ \iff &\left(\sum_{i=1}^{n} x_i\right)(1-\theta) - \left(n - \sum_{i=1}^{n} x_i\right)\theta = 0 \\ \iff &\sum_{i=1}^{n} x_i - n\theta = 0 \\ \iff &\hat{\theta} = \frac{1}{n}(\sum_{i=1}^{n} X_i). \end{aligned}$$

1.3 MAP vs MLE

Answer each question with **T** or **F** and **provide a one sentence explanation of your answer**:

(a) [2 pts.] **T or F:** In the limit, as n (the number of samples) increases, the MAP and MLE estimates become the same.

Solution: True. As the number of examples increases, the data likelihood goes to zero very quickly, while the magnitude of the prior stays the same. In the limit, the prior plays an insignificant role in the MAP estimate and the two estimates will converge.

(b) [2 pts.] **T or F:** Naive Bayes can only be used with MAP estimates, and not MLE estimates.

Solution: False. In Naive Bayes we need to estimate the distribution of each feature X_i given the label Y. Any technique for estimating the distribution can be used, including both the MLE and the MAP estimate.

1.4 Probability

Assume we have a sample space Ω. Answer each question with **T** or **F**. **No justification is required.**

(a) [1 pts.] **T or F:** If events A, B, and C are disjoint then they are independent.

Solution: False. If they are disjoint, i.e. mutually exclusive, they are very dependent!

(b) [1 pts.] **T or F:** $P(A|B) \propto \frac{P(A)P(B|A)}{P(A|B)}$. (The sign '$\propto$' means 'is proportional to')

Solution: False. $P(A|B) = \frac{P(A)P(B|A)}{P(B)}$

(c) [1 pts.] **T or F:** $P(A \cup B) \leq P(A)$.

Solution: False. $P(A \cup B) \geq P(A)$

(d) [1 pts.] **T or F:** $P(A \cap B) \geq P(A)$.

Solution: False. $P(A \cap B) \leq P(A)$

2 Errors, Errors Everywhere [20 pts.]

2.1 True Errors

Consider a classification problem on $\mathbb{R}^d$ with distribution D and target function $c^* : \mathbb{R}^d \rightarrow \{\pm 1\}$. Let S be an iid sample drawn from the distribution D. Answer each question with **T** or **F** and **provide a one sentence explanation of your answer**:

(a) [4 pts.] **T or F:** The true error of a hypothesis h can be lower than its training error on the sample S.

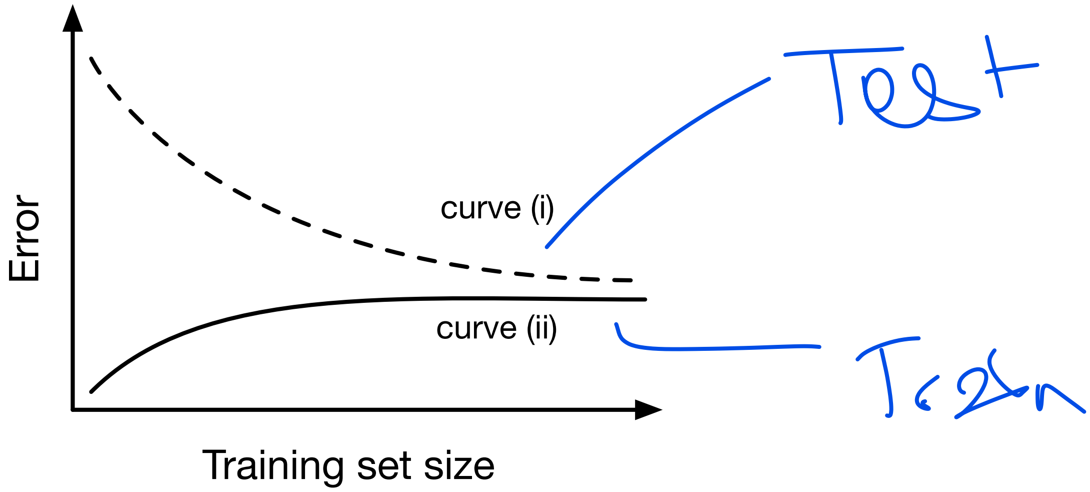
Solution: True. If the sample S happens to favor part of the space where h makes mistakes then the sample error will overestimate the true error. An extreme example is when the hypothesis h has true error 0.5, but the training sample S contains a single sample that h incorrectly classifies.

(b) [4 pts.] **T or F:** Learning theory allows us to determine with 100% certainty the true error of a hypothesis to within any $\epsilon > 0$ error.

Solution: False. There is always a small chance that the sample S is not representative of the underlying distribution D, in which case the sample S may have no relationship to the true error. The sample complexity bounds discussed in class show that this is rare, but not impossible.

2.2 Training Sample Size

In this problem, we will consider the effect of training sample size n on a logistic regression classifier with d features. The classifier is trained by optimizing the conditional log-likelihood. The optimization procedure stops if the estimated parameters perfectly classify the training data or they converge. The following plot shows the general trend for how the training and testing error change as we increase the sample size $n = |S|$. Your task in this question is to analyze this plot and identify which curve corresponds to the training and test error. Specifically:



(a) [8 pts.] Which curve represents the training error? **Please provide 1–2 sentences of justification.**

Solution: It is easier to correctly classify small training datasets. For example, if the data contains just a single point, then logistic regression will always have zero training error. On the other hand, we don't ~~expect a classifier learned from few examples~~ to generalize well, so for small training sets the true error is large. Therefore, curve (ii) shows the general trend of the training error.

(b) [4 pt.] In one word, what does the gap between the two curves represent?

Solution: The gap between the two curves represents the amount of overfitting.

↳ adding data always decrease overfitting

3 Linear and Logistic Regression [20 pts. + 2 Extra Credit]

3.1 Linear regression

Given that we have an input x and we want to estimate an output y, in linear regression we assume the relationship between them is of the form $y = wx + b + \epsilon$, where w and b are real-valued parameters we estimate and ϵ represents the noise in the data. When the noise is Gaussian, maximizing the likelihood of a dataset $S = \{(x_1, y_1), \ldots, (x_n, y_n)\}$ to estimate the parameters w and b is equivalent to minimizing the squared error:

$$\arg\min_{w} \sum_{i=1}^{n} (y_i - (wx_i + b))^2.$$

Consider the dataset S plotted in Fig. 1 along with its associated regression line. For each of the altered data sets S^{new} plotted in Fig. 3, indicate which regression line (relative to the original one) in Fig. 2 corresponds to the regression line for the new data set. Write your answers in the table below.

Dataset	(a)	(b)	(c)	(d)	(e)
Regression line	(b)	(c)	(b)	(a)	(a)

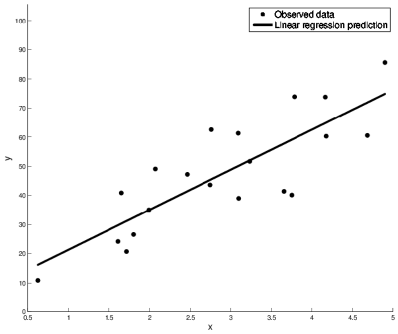
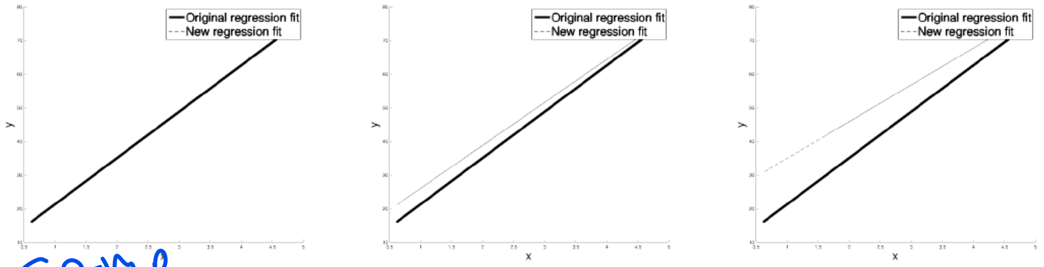


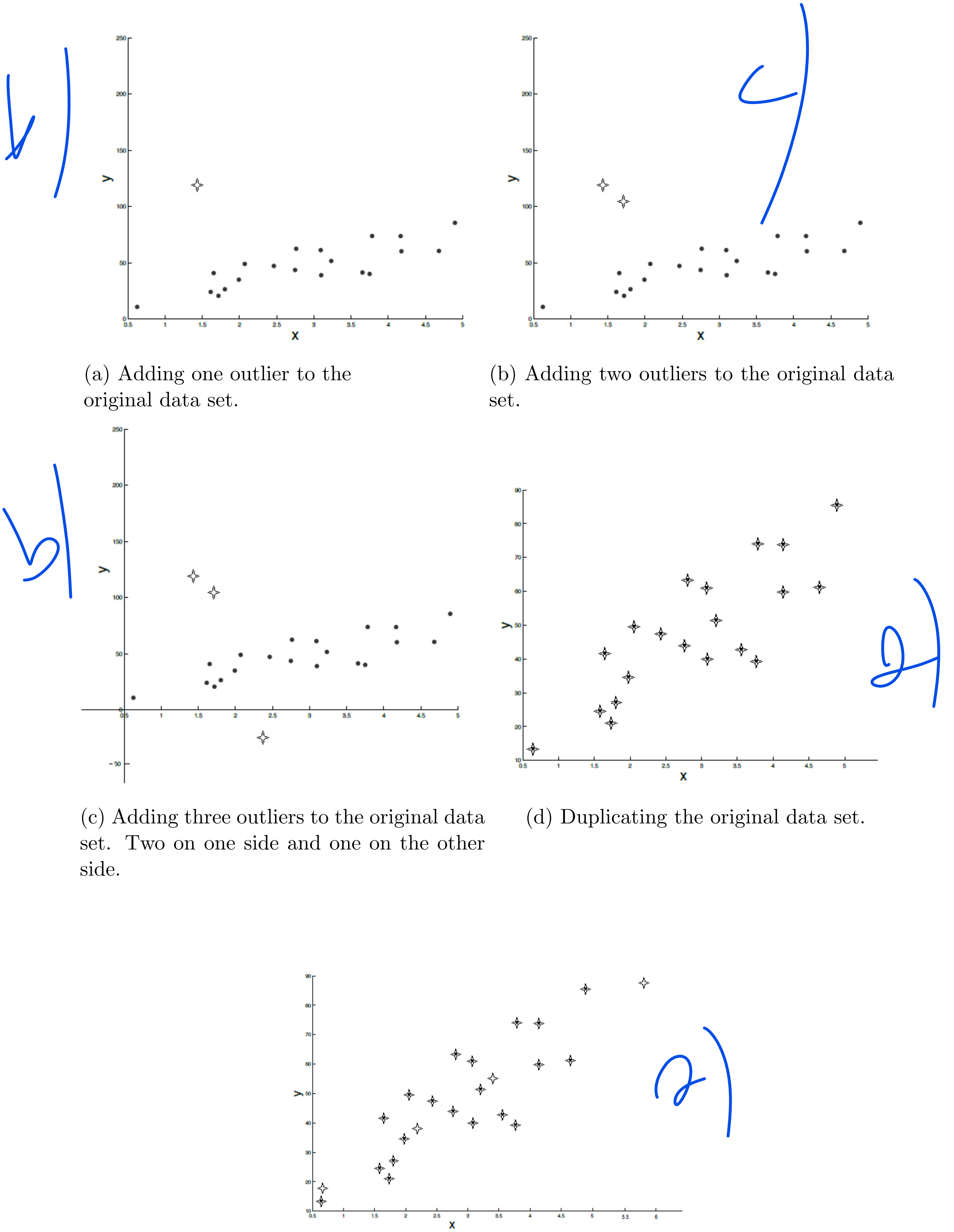
Figure 1: An observed data set and its associated regression line.



same

(a) Old and new regression lines. (b) Old and new regression lines. (c) Old and new regression lines.

Figure 2: New regression lines for altered data sets S^{new}.



(a) Adding one outlier to the original data set.

(b) Adding two outliers to the original data set.

(c) Adding three outliers to the original data set. Two on one side and one on the other side.

(d) Duplicating the original data set.

(e) Duplicating the original data set and adding four points that lie on the trajectory of the original regression line.

Figure 3: New data set S^{new}.

3.2 Logistic regression

Given a training set $\{(x_i, y_i), i = 1, \dots, n\}$ where $x_i \in \mathbb{R}^d$ is a feature vector and $y_i \in \{0, 1\}$ is a binary label, we want to find the parameters $\hat{w}$ that maximize the likelihood for the training set, assuming a parametric model of the form

$$p(y = 1|x; w) = \frac{1}{1 + \exp(-w^T x)}.$$

The conditional log likelihood of the training set is

$$\ell(w) = \sum_{i=1}^{n} y_i \log p(y_i, |x_i; w) + (1 - y_i) \log(1 - p(y_i, |x_i; w)),$$

and the gradient is

$$\nabla \ell(w) = \sum_{i=1}^{n} (y_i - p(y_i|x_i; w))x_i.$$

(a) [5 pts.] Is it possible to get a closed form for the parameters $\hat{w}$ that maximize the conditional log likelihood? How would you compute $\hat{w}$ in practice?

Solution: There is no closed form expression for maximizing the conditional log likelihood. One has to consider iterative optimization methods, such as gradient descent, to compute $\hat{w}$.

(b) [5 pts.] What is the form of the classifier output by logistic regression?

Solution: Given x, we predict $\hat{y} = 1$, if $p(y = 1|x) \geq p(y = 0|x)$. This is reduces to $\hat{y} = 1$, if $w^T x \geq 0$, which is a linear classifier.

(c) [2 pts.] **Extra Credit:** Consider the case with binary features, i.e, $x \in \{0, 1\}^d \subset \mathbb{R}^d$, where feature x_1 is rare and happens to appear in the training set with only label 1. What is $\hat{w}_1$? Is the gradient ever zero for any finite w? Why is it important to include a regularization term to control the norm of $\hat{w}$?

Solution: If a binary feature was active for only label 1 in the training set then, by maximizing the conditional log likelihood, we will make the weight associated to that feature be infinite. This is because, when this feature is observed in the training set, we will want to predict predict 1 irrespective of everything else. This is an undesired behaviour from the point of view of generalization performance, as most likely we do not believe this rare feature to have that much information about class 1. Most likely, it is spurious co-occurrence. Controlling the norm of the weight vector will prevent these pathological cases.

4 SVM, Perceptron and Kernels [20 pts. + 4 Extra Credit]

4.1 True or False

Answer each of the following questions with **T** or **F** and **provide a one line justification**.

(a) [2 pts.] Consider two datasets $D^{(1)}$ and $D^{(2)}$ where $D^{(1)} = \{(x_1^{(1)}, y_1^{(1)}), ..., (x_n^{(1)}, y_n^{(1)})\}$ and $D^{(2)} = \{(x_1^{(2)}, y_1^{(2)}), ..., (x_m^{(2)}, y_m^{(2)})\}$ such that $x_i^{(1)} \in \mathbb{R}^{d_1}$, $x_i^{(2)} \in \mathbb{R}^{d_2}$. Suppose $d_1 > d_2$ and $n > m$. Then the maximum number of mistakes a perceptron algorithm will make is higher on dataset $D^{(1)}$ than on dataset $D^{(2)}$.

Solution: False. The maximum number of mistakes made by a perceptron is dependent on the margin and radius of the training data, not its dimension or size. The maximum mistake a perceptron will make is $(\frac{R}{\gamma})^2$.

(b) [2 pts.] Suppose $\phi(\mathbf{x})$ is an arbitrary feature mapping from input $\mathbf{x} \in \mathcal{X}$ to $\phi(\mathbf{x}) \in \mathbb{R}^N$ and let $K(\mathbf{x}, \mathbf{z}) = \phi(\mathbf{x}) \cdot \phi(\mathbf{z})$. Then $K(\mathbf{x}, \mathbf{z})$ will always be a valid kernel function.

Solution: True. K is a kernel if it is an inner product after applying some feature transformation.

(c) [2 pts.] Given the same training data, in which the points are linearly separable, the margin of the decision boundary produced by SVM will always be greater than or equal to the margin of the decision boundary produced by Perceptron.

Solution: True. SVM explicitly maximizes margin; Perceptron does not differentiate between decision boundaries as long as they lie within the margin of the training data.

4.2 Multiple Choice

(a) [3 pt.] If the data is linearly separable, SVM minimizes $\|w\|^2$ subject to the constraints $\forall i, y_i w \cdot x_i \geq 1$. In the linearly separable case, which of the following may happen to the decision boundary if one of the training samples is removed? **Circle all that apply.**

- Shifts toward the point removed Yes
- Shifts away from the point removed No
- Does not change Yes

(b) [3 pt.] Recall that when the data are not linearly separable, SVM minimizes $\|w\|^2 + C\sum_i \xi_i$ subject to the constraint that $\forall i,\ y_i w \cdot x_i \geq 1 - \xi_i$ and $\xi_i \geq 0$. Which of the following may happen to the size of the margin if the tradeoff parameter C is increased? **Circle all that apply.**

- Increases No
- Decreases Yes

=> less regularization

stricter -> lot less point pass the margin

decrease the margin

Also: $\frac{1}{\|w\|} + C$ } increases $\frac{1}{\|w\|}$

↳ $d = \frac{1}{\|w\|}$

↳ d decreases

cuz $d = \frac{1}{\|w\|}$

- Remains the same Yes

Proof of part (b):

Let $w_0^* = \text{argmin } \|w\|^2 + c_0 \sum_i \xi_i$ and let $w_1^* = \text{argmin } \|w\|^2 + c_1 \sum_i \xi_i$. Let $c_1 > c_0$. We need $\|w_1^*\|^2 \geq \|w_0^*\|^2$. Define $\xi_{i(0)}$ to be the slack variables under w_0^* and $\xi_{i(1)}$ to be the slack variables under w_1^*.

We first show that for any $\|w'\|^2 < \|w_0^*\|^2$, $\sum_i \xi_i' > \sum_i \xi_{i(0)}$ where ξ_i' are the slack variables under w'.

By contradiction, assume $\|w'\|^2 < \|w_0^*\|^2$ and $\sum_i \xi_i' \leq \sum_i \xi_{i(0)}$. Then, $\|w'\|^2 + c_0 \sum_i \xi_i' < \|w_0^*\|^2 + c_0 \sum_i \xi_{i(0)}$ and $w_0^* \neq \text{argmin } \|w\|^2 + c_0 \sum_i \xi_i$.

Thus $\forall \|w'\|^2 \leq \|w_0^*\|^2$, $\sum_i \xi_i' \geq \sum_i \xi_{i(0)}$.

Next, we show that if $w_0^* = \text{argmin } \|w\|^2 + c_0 \sum_i \xi_i$ and $w_1^* = \text{argmin } \|w\|^2 + c_1 \sum_i \xi_i$, then $\|w_1^*\|^2 \geq \|w_0^*\|^2$.

By contradiction, assume $\|w_1^*\|^2 < \|w_0^*\|^2$. Since $w_0^* = \text{argmin } \|w\|^2 + c_0 \sum_i \xi_i$:

$$\|w_0^*\|^2 + c_0 \sum_i \xi_{i(0)} \leq \|w_1^*\|^2 + c_0 \sum_i \xi_{i(1)}$$

Since $c_1 > c_0$ and $\sum_i \xi_{i(1)} > \sum_i \xi_{i(0)}$, then

$$\|w_0^*\|^2 + c_1 \sum_i \xi_{i(0)} < \|w_1^*\|^2 + c_1 \sum_i \xi_{i(1)}$$

But,

$$w_1^* = \text{argmin } \|w\|^2 + c_1 \sum_i \xi_i$$

This yields a contradiction. Thus $\|w_1^*\|^2 \geq \|w_0^*\|^2$.

4.3 Analysis

(a) [4 pts.] In one or two sentences, describe the benefit of using the Kernel trick.

Solution: Allows mapping features into higher dimensional space but avoids the extra computational costs of mapping into higher dimensional feature space explicitly.

(b) [4 pt.] The concept of margin is essential in both SVM and Perceptron. Describe why a large margin separator is desirable for classification.
Solution: Separators with large margin will have low generalization errors with high probability.

(c) [4 pts.] **Extra Credit:** Consider the dataset in Fig. 4. Under the SVM formulation in section 4.2(a),

(1) Draw the decision boundary on the graph.

(2) What is the size of the margin?

(3) Circle all the support vectors on the graph.

Solution: $x_2 - 2.5 = 0$. The size of margin is ~~0.5~~. Support vectors are x_2, x_3, x_6, x_7.

its 1

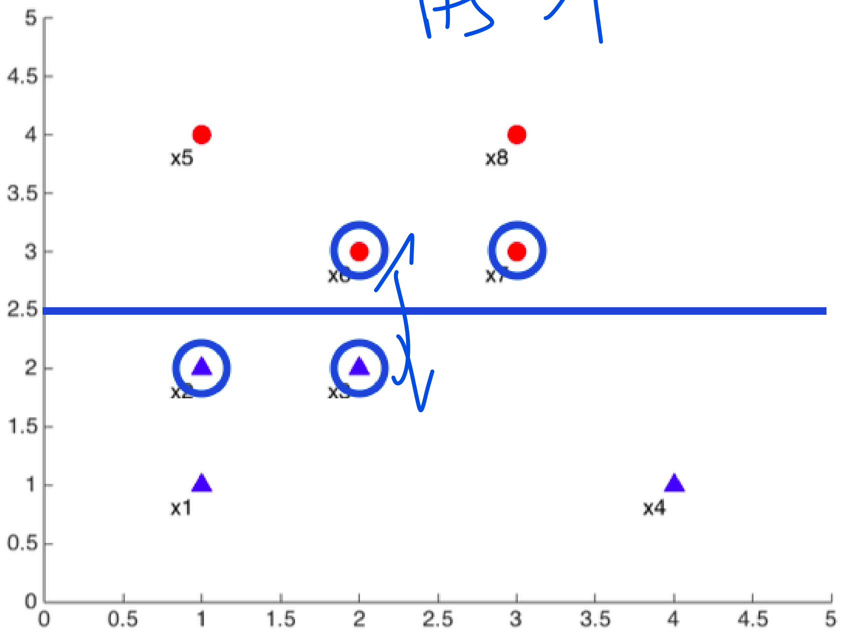


Figure 4: SVM toy dataset

5 Learning Theory [20 pts.]

5.1 True or False

Answer each of the following questions with **T** or **F** and **provide a one line justification.**

(a) [3 pts.] **T or F**: It is possible to label 4 points in $\mathbb{R}^2$ in all possible 2^4 ways via linear separators in $\mathbb{R}^2$.

Solution: F. The VC dimension of linear separator in $\mathbb{R}^2$ is 3, hence it cannot shatter a set of size 4 in all possible ways.

(b) [3 pts.] **T or F**: To show that the VC-dimension of a concept class H (containing functions from X to $\{0, 1\}$) is d, it is sufficient to show that there exists a subset of X with size d that can be labeled by H in all possible 2^d ways.

Solution: F. This is only a necessary condition. We also need to show that no subset of X with size $d + 1$ can be shattered by H.

(c) [3 pts.] **T or F**: The VC dimension of a finite concept class H is upper bounded by $\lceil \log_2 |H| \rceil$.

Solution: T. For any finite set S, if H shatters S, then H at least needs to have $2^{|S|}$ elements, which implies $|S| \leq \lceil \log_2 |H| \rceil$.

(d) [3 pts.] **T or F**: The VC dimension of a concept class with infinite size is also infinite.

Solution: F. Consider all the half-spaces in $\mathbb{R}^2$, which has infinite cardinality but the VC dimension is 3.

(e) [3 pts.] **T or F**: For every pair of classes, H_1, H_2, if $H_1 \subseteq H_2$ and $H_1 \neq H_2$, then $\text{VCdim}(H_1) < \text{VCdim}(H_2)$ (note that this is a strict inequality).

Solution: F. Let H_1 be the collection of all the half-spaces in $\mathbb{R}^2$ with finite slopes and let H_2 be the collection of all the half-spaces in $\mathbb{R}^2$. Clearly $H_1 \subseteq H_2$ and $H_1 \neq H_2$, but $VC(H_1) = VC(H_2) = 3$.

(f) [3 pts.] **T or F**: Given a realizable concept class and a set of training instances, a consistent learner will output a concept that achieves 0 error on the training instances.

Solution: T. Since the concept class is realizable, then the consistent learner can output the oracle labeler by definition, which is guaranteed to achieve 0 error on the training set.

5.2 VC dimension

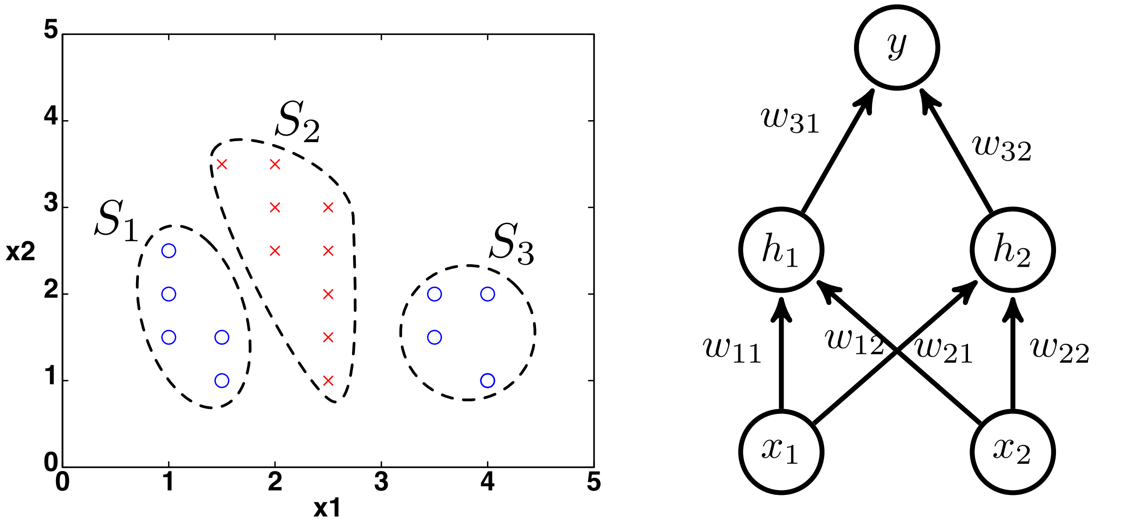
Briefly explain **in 2–3 sentences** the importance of sample complexity and VC dimension in learning with generalization guarantees.

Solution: Sample complexity guarantees quantify how many training samples we need to see from the underlying data distribution D in order to guarantee that uniformly for all hypotheses in the class of functions under consideration we have that their empirical error rates are close to their true errors. This is important because we care abut finding a hypothesis of small true error, but we can only optimize over a fixed training sample. VC bounds are one kind of sample complexity guarantee, where the bound depends on the VC-dimension of the hypothesis class, and they are particularly useful when the class of functions is infinite.

6 Extra Credit: Neural Networks [6 pts.]

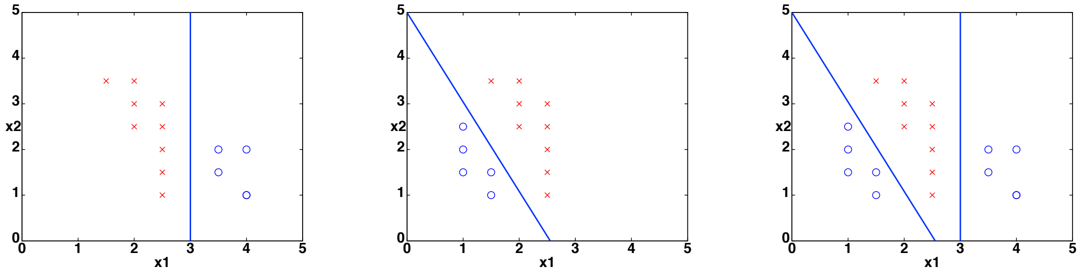
In this problem we will use a neural network to classify the crosses (×) from the circles (○) in the simple dataset shown in Figure 5a. Even though the crosses and circles are not linearly separable, we can break the examples into three groups, S_1, S_2, and S_3 (shown in Figure 5a) so that S_1 is linearly separable from S_2 and S_2 is linearly separable from S_3. We will exploit this fact to design weights for the neural network shown in Figure 5b in order to correctly classify this training set. For all nodes, we will use the threshold activation function

$$\phi(z) = \begin{cases} 1 & z > 0 \\ 0 & z \le 0. \end{cases}$$



(a) The dataset with groups S_1, S_2, and S_3. (b) The neural network architecture

Figure 5



(a) Set S2 and S3 (b) Set S1 and S2 (c) Set S1, S2 and S3

Figure 6: NN classification.

(a) First we will set the parameters w_{11}, w_{12} and b_1 of the neuron labeled h_1 so that its output $h_1(x) = \phi(w_{11}x_1 + w_{12}x_2 + b_1)$ forms a linear separator between the sets S_2 and S_3.

(1) [1 pt.] On Fig. 6a, draw a linear decision boundary that separates S_2 and S_3.

(2) [1 pt.] Write down the corresponding weights w_{11}, w_{12}, and b_1 so that $h_1(x) = 0$ for all points in S_3 and $h_1(x) = 1$ for all points in S_2.

Solution: $w_{11} = -1, w_{12} = 0, b_1 = 3$. With these parameters, we have $w_{11}x_1 + w_{22}x_2 + b_1 > 0$ if and only if $-x_1 > -3$, which is equivalent to $x_1 < 3$.

(b) Next we set the parameters w_{21}, w_{22} and b_2 of the neuron labeled h_2 so that its output $h_2(x) = \phi(w_{21}x_1 + w_{22}x_2 + b_2)$ forms a linear separator between the sets S_1 and S_2.

(1) [1 pt.] On Fig. 6b, draw a linear decision boundary that separates S_1 and S_2.

(2) [1 pt.] Write down the corresponding weights w_{21}, w_{22}, and b_2 so that $h_2(x) = 0$ for all points in S_1 and $h_2(x) = 1$ for all points in S_2.

Solution: The provided line has a slope of -2 and crosses the x_2 axis at the value 5. From this, the equation for the region above the line (those points for which $h_2(x) = 1$)) is given by $x_2 \geq -2x_1 + 5$ or, equivalently, $x_2 + 2x_1 - 5 \geq 0$. Therefore, $w_{21} = 2, w_{22} = 1, b_2 = -5$.

(c) Now we have two classifiers h_1 (to classify S_2 from S_3) and h_2 (to classify S_1 from S_2). We will set the weights of the final neuron of the neural network based on the results from h_1 and h_2 to classify the crosses from the circles. Let $h_3(x) = \phi\big(w_{31}h_1(x) + w_{32}h_2(x) + b_3\big)$.

(1) [1 pt.] Compute w_{31}, w_{32}, b_3 such that $h_3(x)$ correctly classifies the entire dataset.

Solution: Consider the weights $w_{31} = w_{32} = 1$ and $b_3 = -1.5$. With these weights, $h_3(x) = 1$ if $h_1(x) + h_2(x) \geq 1.5$. For points in S_1 and S_3 either h_1 or h_2 is zero, so they will be classified as 0, as required. For points in S_2, both h_1 and h_2 output 1, so the point is classified as 1. This rule has zero training error.

(2) [1 pt.] Draw your decision boundary in Fig. 6c.

Use this page for scratch work