CS 189 Introduction to Machine Learning
Spring 2022 Marvin Zhang

DIS6

Preface: This discussion is long because it is a compilation of relevant midterm and final questions from previous semesters. Work with your GSI in section to focus on the questions that will help you the most.

1 MLE, MAP, Linear Regression

(a) (Sp16 Final)

(5) [3 pts] Lasso can be interpreted as least-squares linear regression where

- ■ weights are regularized with the ℓ_1 norm
- ☐ the weights have a Gaussian prior
- ☐ weights are regularized with the ℓ_2 norm
- ☐ the solution algorithm is simpler

(b) (Sp20 Midterm A)

(d) [4 pts] Suppose we perform least-squares linear regression, but we don't assume that all weight vectors are equally reasonable; instead, we use the maximum *a posteriori* method to impose a normally-distributed prior probability on the weights. Then we are doing

- ● A: L_2 regularization
- ○ C: logistic regression
- ○ B: Lasso regression
- ● D: ridge regression

As shown in Lecture 13, the Bayesian justification for ridge regression is derived by applying MAP to the posterior probability with a Gaussian prior on the weights.

(c) (Sp19 Midterm)

(d) [3 pts] Assuming we can find algorithms to minimize them, which of the following cost functions will encourage **sparse solutions** (i.e., solutions where many components of w are zero)?

- ● $||Xw - y||_2^2 + \lambda||w||_1$
- ● $||Xw - y||_2^2 + \lambda \cdot$ (# of nonzero components of w)
- ● $||Xw - y||_2^2 + \lambda||w||_1^2$
- ○ $||Xw - y||_2^2 + \lambda||w||_2^2$

The first answer is Lasso, which we know finds sparse solutions. The second is Lasso with the penalty squared. Squaring this will leave the same isocontours and this will keep the same properties as Lasso. The third cost function penalizes solutions that are not sparse and will naturally encourage sparse solutions. The last solution is ridge regression, which shrinks weights but does not set weights to zero.

Q3. [10 pts] Maximum Likelihood Estimation

There are 5 balls in a bag. Each ball is either red or blue. Let θ (an integer) be the number of blue balls. We want to estimate θ, so we draw 4 balls **with replacement** out of the bag, replacing each one before drawing the next. We get "blue," "red," "blue," and "blue" (in that order).

(a) [5 pts] Assuming θ is fixed, what is the likelihood of getting exactly that sequence of colors (expressed as a function of θ)?

$\mathcal{L}(\theta; X) = P(X; \theta) = \left(\frac{5-\theta}{5}\right)\left(\frac{\theta}{5}\right)^3$.

(b) [3 pts] Draw a table showing (as a fraction) the likelihood of getting exactly that sequence of colors, for every value of θ from zero to 5 inclusive.

θ	$\mathcal{L}(\theta; \langle$ blue, red, blue, blue $\rangle)$
0	?
1	?
2	?
3	?
4	?
5	?

θ	$\mathcal{L}(\theta; \langle$ blue, red, blue, blue $\rangle)$
0	0
1	4 / 625
2	24 / 625
3	54 / 625
4	64 / 625
5	0

(c) [2 pts] What is the maximum likelihood estimate for θ? (Chosen among all integers; not among all real numbers.)

The maximum likelihood estimate for θ is 4.

2 MVG, Gaussian Classification

(a) (Sp20 Midterm B)

(b) [4 pts] Consider a random variable $X \sim \mathcal{N}(\mu, \Sigma) \in \mathbb{R}^d$, where the multivariate Gaussian probability density function (PDF) is axis-aligned, Σ is positive definite, and the standard deviation along coordinate axis i is σ_i. Select all that apply.

- ○ A: The d features of X are uncorrelated but not necessarily independent
- ○ B: $\Sigma = \text{diag}(\sqrt{\sigma_1}, \sqrt{\sigma_2}, \ldots, \sqrt{\sigma_d})$
- ● C: Σ has a symmetric square root $\Sigma^{1/2}$ with eigenvalues $\sigma_1, \sigma_2, \ldots, \sigma_d$
- ● D: $(X-\mu)^\top \Sigma^{-1}(X-\mu) \geq 0$

A is incorrect: It is true that because the multivariate Gaussian is axis-aligned, its components are uncorrelated. However, the uncorrelated multivariate Gaussian also implies independence here, since $p(X; \mu, \Sigma) = \prod_{i=1}^n p(x_i; \mu_i, \sigma_i^2)$ (see Option C).

B is incorrect: The eigenvectors are standard basis vectors, so Σ must be a diagonal matrix, but with elements $\text{diag}(\sigma_1^2, \ldots, \sigma_n^2)$.

C is correct: Because Σ is a diagonal matrix, expanding out the probability density function gives us option C.

D is correct: Σ is positive definite, so Σ^{-1} is also positive definite. Thus, $(X-\mu)^\top \Sigma^{-1}(X-\mu)$ must be nonnegative.

(b) (Sp19 Midterm)

(i) [3 pts] Which of the following apply to **linear discriminant analysis**?

- ● You calculate the sample mean for each class
- ○ You calculate the sample covariance matrix using the mean of all the data points
- ● It approximates the Bayes decision rule
- ○ The model produced by LDA is never the same as the model produced by QDA

Top left: Calculating the sample mean within each class is part of LDA by definition

Bottom left: You calculate the sample covariance using the mean for each class, not the mean of all the data points

Top right: LDA finds what the Bayes decision rule would be under the assumption the class conditionals have normal distributions, parameterized by the sample means and covariance

Bottom right: QDA can produce the same covariance for each class as LDA

(c) (Sp19 Midterm)

(s) [3 pts] Suppose you have a sample in which each point has d features and comes from class C or class D. The class conditional distributions are $(X_i|y_i = C) \sim N(\mu_C, \sigma_C^2)$ and $(X_i|y_i = D) \sim N(\mu_D, \sigma_D^2)$ for unknown values $\mu_C, \mu_D \in \mathbb{R}^d$ and $\sigma_C^2, \sigma_D^2 \in \mathbb{R}$. The class priors are π_C and π_D. We use 0-1 loss.

- ● If $\pi_C = \pi_D$ and $\sigma_C = \sigma_D$, then the Bayes decision rule assigns a test point z to the class whose mean is closest to z.
- ● If $\pi_C = \pi_D$, then the Bayes decision rule is $r^*(z) = \text{argmin}_{A \in \{C,D\}} \left(|z - \mu_A|^2/(2\sigma_A^2) + d \ln \sigma_A \right)$
- ● If $\sigma_C = \sigma_D$, then the Bayes decision boundary is always linear.
- ○ If $\sigma_C = \sigma_D$, then QDA will always produce a linear decision boundary when you fit it to your sample.

4 Watermelons

1. (6 points) Finn lives on a watermelon farm and wants to classify whether a watermelon is sweet (labelled as $y = 1$) or not sweet (labelled as $y = 0$). Finn observes a d-dimensional feature vector $x \in \mathbb{R}^d$ associated with the appearance and the smell of each watermelon. Before observing a watermelon, Finn's general prior is that a watermelon is sweet with probability $p(y = 1) = \pi_1$ and not sweet with probability $p(y = 0) = 1 - \pi_1$.

   Finn is a watermelon expert and knows that the conditional probability distribution of the watermelon features $p(x|y = k)$ for class k (where $k = 0, 1$) is a d-dimensional Gaussian distribution $N(\mu_k, \mathbf{\Sigma})$ with mean $\mu_k \in \mathbb{R}^d$ and covariance $\Sigma \in \mathbb{R}^{d\times d}$. Note that the same covariance matrix Σ is shared between the two classes.

$$f(x|y = k) = \frac{1}{\sqrt{(2\pi)^d|\Sigma|}} \exp\{-\frac{1}{2}(x - \mu_k)^T\Sigma^{-1}(x - \mu_k)\} \tag{8}$$

   Can you help Finn find out how likely a given watermelon is sweet? Write down a simplified expression of $p(y = 1|x)$ as a function of x, μ_0, μ_1, Σ, and π_1. The expression should be in the form of $s(w^T x + b)$ where $s(z) = \frac{1}{1+e^{-z}}$ is the sigmoid function is the sigmoid function. Write down what w and b should be.

   Solution: By Bayes' Rule,

$$p(y = 1|x) = \frac{p(x|y = 1)p(y = 1)}{p(x)} = \frac{p(x|y = 1)p(y = 1)}{p(x|y = 0)p(y = 0) + p(x|y = 1)p(y = 1)}.$$

   Therefore,

$$p(y = 1|x) = \frac{\pi_1 f_1(x)}{\pi_1 f_1(x) + \pi_0 f_0(x)} = \frac{1}{1 + \frac{\pi_0 f_0(x)}{\pi_1 f_1(x)}}.$$

   When substituting in the Gaussian probability density function for $f(x)$ and writing π_1 as $e^{\log \pi_1}$,

$$\pi_k f_k(x) = \frac{1}{\sqrt{(2\pi)^d|\Sigma|}} \exp\{-\frac{1}{2}(x - \mu_k)^T\Sigma^{-1}(x - \mu_k) + \log \pi_k\} = \frac{1}{\sqrt{(2\pi)^d|\Sigma|}} \exp(q_k(x)),$$

   where $q_k(x) = -\frac{1}{2}(x - \mu_k)^T\Sigma^{-1}(x - \mu_k) + \log \pi_k$.

   Therefore, the ratio can be written as

$$\frac{\pi_0 f_0(x)}{\pi_1 f_1(x)} = e^{q_0(x)-q_1(x)},$$

   and hence

$$p(y = 1|x) = \frac{1}{1 + e^{q_0(x)-q_1(x)}} = s(q_1(x) - q_0(x)).$$

   Now let's look at the expression $q_1(\mathbf{x}) - q_0(\mathbf{x})$

$$q_1(\mathbf{x}) - q_0(\mathbf{x}) = \log \frac{\pi_1}{1 - \pi_1} - \frac{1}{2}(\mathbf{x} - \boldsymbol{\mu}_1)^\top \mathbf{\Sigma}^{-1}(\mathbf{x} - \boldsymbol{\mu}_1) + \frac{1}{2}(\mathbf{x} - \boldsymbol{\mu}_0)^\top \mathbf{\Sigma}^{-1}(\mathbf{x} - \boldsymbol{\mu}_0)$$

$$= \log \frac{\pi_1}{1-\pi_1} + \mathbf{x}^\top \boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0) - \frac{1}{2}\boldsymbol{\mu}_1^\top \boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_1 + \frac{1}{2}\boldsymbol{\mu}_0^\top \boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_0$$

Notice that we can write it out as:

$$q_1(\mathbf{x}) - q_0(\mathbf{x}) = \log \frac{\pi_1}{1-\pi_1} + \frac{1}{2}\boldsymbol{\mu}_0^\top \boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_0 - \frac{1}{2}\boldsymbol{\mu}_1^\top \boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_1 + (\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0)^\top \boldsymbol{\Sigma}^{-1}\mathbf{x} = w_0 + \mathbf{w}^\top \mathbf{x}$$

In other words,

$$p(y = 1 \mid x) = s(w_0 + w^T x),$$

where

$$w_0 = \log \frac{\pi_1}{1-\pi_1} + \frac{1}{2}\boldsymbol{\mu}_0^\top \boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_0 - \frac{1}{2}\boldsymbol{\mu}_1^\top \boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_1,$$

and

$$w^T = (\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0)^\top \boldsymbol{\Sigma}^{-1}.$$

3 Logistic Regression, Optimization

(a) (Sp19 Midterm)

(e) [3 pts] Which of the following statements about **logistic regression** are correct?

- ● The cost function of logistic regression is convex
- ○ The cost function of logistic regression is concave
- ○ Logistic regression uses the squared error as the loss function
- ○ Logistic regression assumes that each class's points are generated from a Gaussian distribution

(b) (Sp20 Midterm B)

(d) [4 pts] For classification problems with two features ($d = 2$, test point $z \in \mathbb{R}^2$), which of the following methods have posterior probability distributions of the form $P(Y|X = z) = s(Az_1^2 + Bz_2^2 + Cz_1z_2 + Dz_1 + Ez_2 + F)$ where s is the logistic function $s(\gamma) = \frac{1}{1+e^{-\gamma}}$ and $A, B, C, D, E, F \in \mathbb{R}$ can all be nonzero?

- ○ A: Logistic regression with linear features
- ● C: Logistic regression with quadratic features
- ● B: Linear discriminant analysis (LDA) with quadratic features
- ● D: Quadratic discriminant analysis (QDA) with linear features

All but A, as standard logistic regression does not have the potential for A, B, C to be nonzero.

(c) (Sp21 Midterm)

(c) [4 pts] Recall the logistic function $s(\gamma)$ and its derivative $s'(\gamma) = \frac{\mathrm{d}}{\mathrm{d}\gamma}s(\gamma)$. Let γ^* be the value of γ that maximizes $s'(\gamma)$.

- ○ A: $\gamma^* = 0.25$
- ○ C: $s'(\gamma^*) = 0.5$
- ● B: $s(\gamma^*) = 0.5$
- ● D: $s'(\gamma^*) = 0.25$

A glance at the logistic curve and its symmetry makes it clear that the slope is maximized when $\gamma^* = 0$ and $s(\gamma^*) = 0.5$. Recall that $s' = s(1 - s)$; hence $s'(\gamma^*) = 0.25$.

(d) (Sp19 Midterm)

(r) [3 pts] Let $L_i(w)$ be the loss corresponding to a sample point X_i with label y_i. The update rule for **stochastic gradient descent** with step size ϵ is

- ○ $w_{\text{new}} \leftarrow w - \epsilon \nabla_{X_i} L_i(w)$
- ● $w_{\text{new}} \leftarrow w - \epsilon \nabla_w L_i(w)$
- ○ $w_{\text{new}} \leftarrow w - \epsilon \sum_{i=1}^{n} \nabla_{X_i} L_i(w)$
- ○ $w_{\text{new}} \leftarrow w - \epsilon \sum_{i=1}^{n} \nabla_w L_i(w)$

Any option taking the gradient w.r.t. X_i is an incorrect update rule for the weights w, so that leaves only the solutions which use ∇_w. Any option which sums over multiple gradients is a batch method, which leaves only one option, the solution.

Q5. [10 pts] Logistic Regression with One Feature

We are given another sample in which each point has only one feature. Consider a binary classification problem in which sample values $x \in \mathbb{R}$ are drawn randomly from two different class distributions. The first class, with label $y = 0$, has its mean to the left of the mean of the second class, with label $y = 1$. We will use a modified version of logistic regression to classify these data points. We model the posterior probability at a test point $z \in \mathbb{R}$ as

$$P(y = 1|z) = s(z - \alpha),$$

where $\alpha \in \mathbb{R}$ is the sole parameter we are trying to learn and $s(\gamma) = 1/(1 + e^{-\gamma})$ is the logistic function. The decision boundary is $z = \alpha$ (because $s(z) = \frac{1}{2}$ there).

We will learn the parameter α by performing gradient descent on the logistic loss function (a.k.a. cross-entropy). That is, for a data point x with label $y \in \{0, 1\}$, we find the α that minimizes

$$J(\alpha) = -y \ln s(x - \alpha) - (1 - y) \ln(1 - s(x - \alpha)).$$

(a) [5 pts] Derive the stochastic gradient descent update for J with step size $\epsilon > 0$, given a sample value x and a label y. Hint: feel free to use s as an abbreviation for $s(x - \alpha)$.

By the chain rule,

$$\frac{d}{d\alpha} s(x - \alpha) = -s(1 - s).$$

Hence,

$$\begin{aligned} J'(\alpha) &= \frac{ys(1 - s)}{s} - \frac{(1 - y)s(1 - s)}{1 - s} \\ &= y(1 - s) - (1 - y)s \\ &= y - s. \end{aligned}$$

So the stochastic gradient descent update rule is

$$\alpha^{(t+1)} \leftarrow \alpha^{(t)} + \epsilon(s(x - \alpha) - y).$$

(b) [3 pts] Is $J(\alpha)$ convex over $\alpha \in \mathbb{R}$? Justify your answer.

Continuing from the last part,

$$J''(\alpha) = \frac{d}{d\alpha}(y - s) = s(1 - s).$$

As the logistic function is always in the range $(0, 1)$, $s(1 - s)$ is always positive, so $J(\alpha)$ is convex. (Moreover, it's strictly convex and hence admits at most one solution.)

(c) [2 pts] Now we consider multiple sample points. As $d = 1$, we are given an $n \times 1$ design matrix X and a vector $y \in \mathbb{R}^n$ of labels. Consider batch gradient descent on the cost function $\sum_{i=1}^{n} J(\alpha; X_i, y_i)$. There are circumstances in which this cost function does not have a minimum over $\alpha \in \mathbb{R}$ at all. What is an example of such a circumstance?

If all the sample points are of only one class, then $J(\alpha)$ is either monotonically increasing or monotonically decreasing over $\alpha \in \mathbb{R}$.

4 SVM

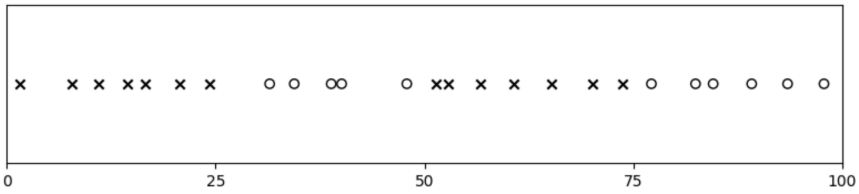
(a) (Sp16 Final)

(14) [3 pts] Which of the following can help to reduce overfitting in an SVM classifier?

- ■ Use of slack variables
- ☐ High-degree polynomial features
- ☐ Normalizing the data
- ☐ Setting a very low learning rate

(b) (Sp19 Midterm)

(k) [3 pts] Suppose you are given the one-dimensional data $\{x_1, x_2, \ldots, x_{25}\}$ illustrated below and you have only a **hard-margin support vector machine** (with a fictitious dimension) at your disposal. Which of the following modifications can give you 100% training accuracy?



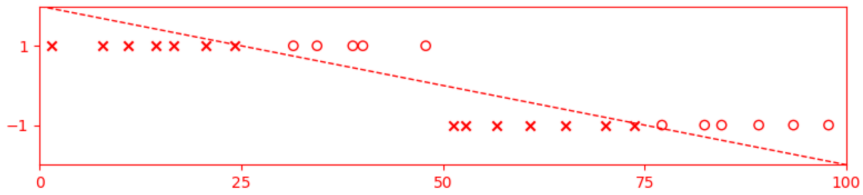
- ○ Centering the data
- ○ Add a feature x_i^2
- ● Add a feature that is 1 if $x \leq 50$, or -1 if $x > 50$
- ● Add two features, x_i^2 and x_i^3

○ The performance of SVM is shift invariant, so centering the data won't affect the result;

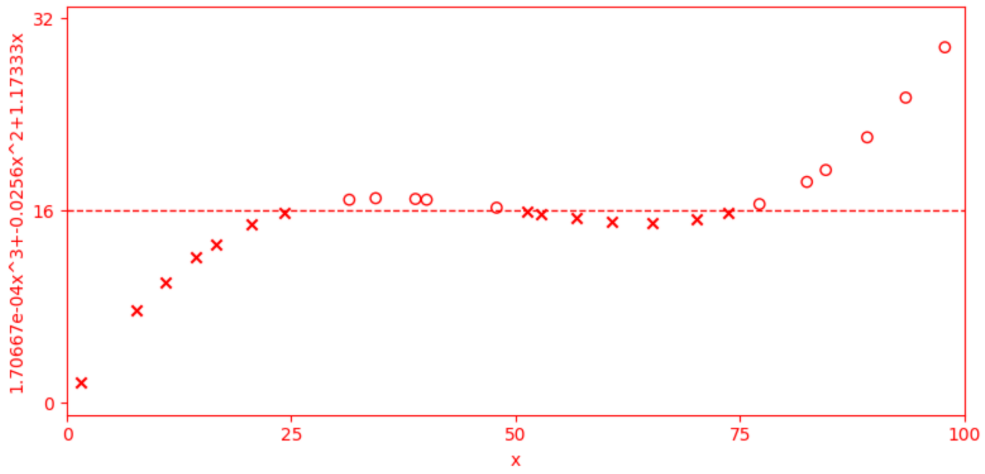
○ A line can separate a quadratic function into at most 3 segments and is not sufficient;

● See image below;

● A line can separate a cubic function into 4 segments. See image below.



Adding "1 if $x_i \leq 50$..." feature



Adding x_i^2 and x_i^3

(c) (Sp21 Midterm)

(b) [4 pts] Which of the following changes would commonly cause an SVM's margin $1/\|w\|$ to shrink?

- ● A: Soft margin SVM: increasing the value of C
- ○ C: Soft margin SVM: decreasing the value of C
- ● B: Hard margin SVM: adding a sample point that violates the margin
- ○ D: Hard margin SVM: adding a new feature to each sample point

The greater the value of C is, the higher the penalty for violating the margin. The soft margin shrinks to compensate.

If you add a sample point that violates the margin, a hard margin always shrinks.

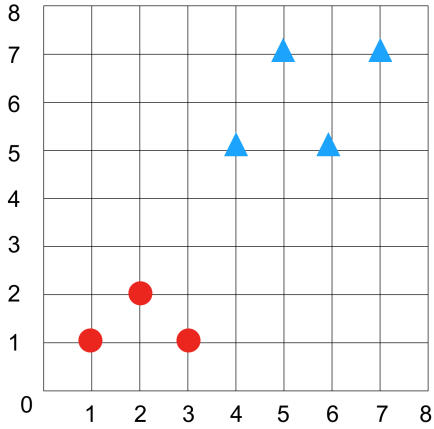
If you add a feature, the old solution can still be used (by setting the weight associated with the new feature to zero). Although the new feature might enable a new solution with a wider margin, the optimal solution can't be worse than the old solution.

Q2. [20 pts] Hard-Margin Support Vector Machines

Recall that a **maximum margin classifier**, also known as a hard-margin support vector machine (SVM), takes n training points $X_1, X_2, \ldots, X_n \in \mathbb{R}^d$ with labels $y_1, y_2, \ldots, y_n \in \{+1, -1\}$, and finds parameters $w \in \mathbb{R}^d$ and $\alpha \in \mathbb{R}$ that satisfy a certain objective function subject to the constraints

$$y_i(X_i \cdot w + \alpha) \geq 1, \quad \forall i \in \{1, \ldots, n\}.$$

For parts (a) and (b), consider the following training points. Circles are classified as positive examples with label +1 and triangles are classified as negative examples with label −1.



(a) [3 pts] Which points are the support vectors? Write it as $\begin{bmatrix}\text{horizontal}\\ \text{vertical}\end{bmatrix}$. E.g., the bottom right circle is $\begin{bmatrix}3\\1\end{bmatrix}$.

The support vectors are the points $\begin{bmatrix}2\\2\end{bmatrix}$ and $\begin{bmatrix}4\\5\end{bmatrix}$.

(b) [4 pts] If we add the sample point $x = \begin{bmatrix}5\\1\end{bmatrix}$ with label −1 (triangle) to the training set, which points are the support vectors?

The support vectors are the points $\begin{bmatrix}3\\1\end{bmatrix}$, $\begin{bmatrix}4\\5\end{bmatrix}$, and $\begin{bmatrix}5\\1\end{bmatrix}$.

For parts (c)–(f), forget about the figure above, but assume that there is at least one sample point in each class and that the sample points are linearly separable.

(c) [2 pts] Describe the geometric relationship between w and the decision boundary.

The weight vector w (called the *normal vector*) is orthogonal to the decision boundary.

(d) [2 pts] Describe the relationship between w and the margin. (For the purposes of this question, the margin is just a number.)

The margin (the distance from the decision boundary to the nearest sample point) is $1/\|w\|$.

(e) [4 pts] Knowing what you know about the hard-margin SVM objective function, explain why for the optimal (w, α), there must be at least one sample point for which $X_i \cdot w + \alpha = 1$ and one sample point for which $X_i \cdot w + \alpha = -1$.

The objective is to minimize $\|w\|^2$ (or equivalently, $\|w\|$). If every sample point has $y_i(X_i \cdot w + \alpha) > 1$, we can simply scale w to make it smaller until there is a point such that $y_i(X_i \cdot w + \alpha) = 1$, thereby improving the "solution."

If we have a positive sample point for which $X_i \cdot w + \alpha = 1$ but every negative sample point has $X_i \cdot w + \alpha < -1$, we can make α a little greater so that every sample point has $y_i(X_i \cdot w + \alpha) > 1$. Then we can shrink w some more. So any such "solution" cannot be optimal. (The symmetric argument applies if a negative sample point touches the slab but not positive sample point does.)

(f) [5 pts] If we add new features to the sample points (while retaining all the original features), can the optimal $\|w_{\text{new}}\|$ in the enlarged SVM be greater than the optimal $\|w_{\text{old}}\|$ in the original SVM? Can it be smaller? Can it be the same? Explain why! (Most of the points will be for your explanation.)

It can be smaller, or it can be the same, but it cannot be greater.

If w_{old} and α are an optimal solution of the original SVM, when we add features we can create a w_{new} that has the same values as w_{old}, with zeros added for the new features. Then w_{new} and α satisfy all the constraints of the enlarged SVM. These might not be the optimal solution, but the optimal solution of the enlarged SVM cannot have $\|w_{\text{new}}\|$ greater than $\|w_{\text{old}}\|$.

$\|w_{\text{new}}\|$ can be smaller, because the new features can put an arbitrarily large amount of space between the classes, making the margin arbitrarily large.

$\|w_{\text{new}}\|$ will be the same as $\|w_{\text{old}}\|$ if the new features are all zeros in all the sample points.

5 Featurization, Kernels

(a) (Sp19 Midterm)

(q) [3 pts] You are given four sample points $X_1 = [-1, -1]^\top$, $X_2 = [-1, 1]^\top$, $X_3 = [1, -1]^\top$, and $X_4 = [1, 1]^\top$. Each of them is in class C or class D. For what feature representations are the lifted points $\Phi(X_i)$ *guaranteed* to be **linearly separable** (with no point lying exactly on the decision boundary) for every possible class labeling?

- ○ $\Phi(x) = [x_1, x_2, 1]$
- ○ $\Phi(x) = [x_1^2, x_2^2, x_1, x_2, 1]$
- ○ $\Phi(x) = [x_1, x_2, x_1^2 + x_2^2, 1]$
- ● $\Phi(x) = [x_1^2, x_2^2, x_1x_2, x_1, x_2, 1]$

Consider the labels $y_1 = 1, y_2 = 0, y_3 = 0, y_4 = 1$. No linear classifier (with or without bias) can classify these samples. Also, $x_1^2 = x_2^2 = 1$ for all X_i, and $x_1^2 + x_2^2 = 2$ for all X_i.

The last option is a quadratic kernel, which can learn e.g. an elliptical decision boundary in the original sample space to perfectly classify the above labeling.

(b) (Sp20 Final)

(10) [4 pts] Which of the following statement(s) about **kernels** are true?

- ● A: The dimension of the lifted feature vectors $\Phi(\cdot)$, whose inner products the kernel function computes, can be infinite.
- ○ B: For any desired lifting $\Phi(x)$, we can design a kernel function $k(x, z)$ that will evaluate $\Phi(x)^\top\Phi(z)$ more quickly than explicitly computing $\Phi(x)$ and $\Phi(z)$.
- ● C: The kernel trick, when it is applicable, speeds up a learning algorithm if the number of sample points is substantially less than the dimension of the (lifted) feature space.
- ● D: If the raw feature vectors x, y are of dimension 2, then $k(x, y) = x_1^2y_1^2 + x_2^2y_2^2$ is a valid kernel.

A is correct; consider the Gaussian kernel from lecture. B is wrong; most liftings don't lead to super-fast kernels. Just some special ones do. C is correct, straight from lecture. Though in this case, the dual algorithm is faster than the primal whether you use a fancy kernel or not. D is correct because $k(x, y)$ is inner product of $\Phi(x) = [x_1^2 \quad x_2^2]^\top$ and $\Phi(y) = [y_1^2 \quad y_2^2]^\top$.

(c) (Sp16 Final)

(7) [3 pts] The kernel trick

- ☐ can be applied to every classification algorithm
- ☐ is commonly used for dimensionality reduction
- ☐ changes ridge regression so we solve a $d \times d$ linear system instead of an $n \times n$ system, given n sample points with d features
- ■ exploits the fact that in many learning algorithms, the weights can be written as a linear combination of input points

Q4. [10 pts] Kernels

(1) [2 pts] What is the primary motivation for using the kernel trick in machine learning algorithms?

If we want to map sample points to a very high-dimensional feature space, the kernel trick can save us from having to compute those features explicitly, thereby saving a lot of time.

(Alternative solution: the kernel trick enables the use of infinite-dimensional feature spaces.)

(2) [4 pts] Prove that for every design matrix $X \in \mathbb{R}^{n \times d}$, the corresponding kernel matrix is positive semidefinite.

For every vector $\mathbf{z} \in \mathbb{R}^n$,

$$\mathbf{z}^\top K \mathbf{z} = \mathbf{z}^\top X X^\top \mathbf{z} = |X^\top \mathbf{z}|^2,$$

which is clearly nonnegative.

(3) [2 pts] Suppose that a regression algorithm contains the following line of code.

$$\mathbf{w} \leftarrow \mathbf{w} + X^\top M X X^\top \mathbf{u}$$

Here, $X \in \mathbb{R}^{n \times d}$ is the design matrix, $\mathbf{w} \in \mathbb{R}^d$ is the weight vector, $M \in \mathbb{R}^{n \times n}$ is a matrix unrelated to X, and $\mathbf{u} \in \mathbb{R}^n$ is a vector unrelated to X. We want to derive a dual version of the algorithm in which we express the weights $\mathbf{w}$ as a linear combination of samples X_i (rows of X) and a dual weight vector $\mathbf{a}$ contains the coefficients of that linear combination. Rewrite the line of code in its dual form so that it updates $\mathbf{a}$ correctly (and so that $\mathbf{w}$ does not appear).

$$\mathbf{a} \leftarrow \mathbf{a} + M X X^\top \mathbf{u}$$

(4) [2 pts] Can this line of code for updating $\mathbf{a}$ be kernelized? If so, show how. If not, explain why.

Yes:

$$\mathbf{a} \leftarrow \mathbf{a} + M K \mathbf{u}$$

6 Bias-Variance Decomposition, Decision Theory

(a) (Sp16 Final)

(20) [3 pts] How does the bias-variance decomposition of a ridge regression estimator compare with that of ordinary least squares regression? (Select one.)

- ☐ Ridge has larger bias, larger variance
- ☐ Ridge has smaller bias, larger variance
- ■ Ridge has larger bias, smaller variance
- ☐ Ridge has smaller bias, smaller variance

(b) (Sp19 Midterm)

(j) [3 pts] Which of the following are reasons why you might adjust your model in ways that increase the bias?

- ○ You observe high training error and high validation error
- ● You observe low training error and high validation error
- ● You have few data points
- ○ Your data are not linearly separable

Top left: High training and validation error is a sign of underfitting; higher bias leads to greater underfitting hence you would not want the bias to increase

Bottom left: With few data points, noise in the data has a larger effect on the model. Methods to reduce overfitting to the noise would increase the bias

Top right: Low training error and high validation error is a sign of overfitting; methods to decrease overfitting increase bias

Bottom right: If the data are not linearly separable, you need a more complex decision boundary. Higher bias would reduce the complexity of the decision boundary.

(c) (Sp20 Midterm A)

(e) [4 pts] Which of the following statements regarding ROC curves are true?

2

- [x] A: the ROC curve is monotonically increasing
- [ ] B: for a logistic regression classifier, the ROC curve's horizontal axis is the posterior probability used as a threshold for the decision rule
- [ ] C: the ROC curve is concave
- [x] D: if the ROC curve passes through $(0, 1)$, the classifier is always correct (on the test data used to make the ROC curve)

The axes of an ROC curve do not correspond to the "knob" we're turning when we plot the curve.

Always predicting positive will give us 100

Does not have to be concave, just needs to be increasing.

Since it's increasing, the curve is a horizontal line at y=1. So, we have no false positives nor false negatives.

Q4. [20 pts] Finding Bias, Variance, and Risk

For $z \in \mathbb{R}$, you are trying to estimate a true function $g(z) = 2z^2$ with **least-squares regression**, where the regression function is a line $h(z) = wz$ that goes through the origin and $w \in \mathbb{R}$. Each sample point $x \in \mathbb{R}$ is drawn from the **uniform distribution on** $[-1, 1]$ and has a corresponding label $y = g(x) \in \mathbb{R}$. There is no noise in the labels. We train the model with **just one sample point**! Call it x, and assume $x \neq 0$. We want to apply the bias-variance decomposition to this model.

(a) [3 pts] In one sentence, why do we expect the bias to be large?

Because a line is not a good fit for a parabola.

(b) [6 pts] What is the bias of your model $h(z)$ as a function of a test point $z \in \mathbb{R}$? (*Hint: start by working out the value of the least-squares weight w.*) Your final bias should not include an x; work out the expectation.

The least-squares solution is $w = X^+ y$, where X is the 1×1 matrix $[x]$. Hence $X^+ = (X^\top X)^{-1} X^\top = \frac{x}{x^2} = \frac{1}{x}$. Then

$$\begin{aligned} w &= X^+ y = \frac{1}{x}(2x^2) = 2x, \\ h(z) &= wz = 2xz, \text{ and} \\ \text{bias}(z) &= \mathbb{E}[h(z)] - g(z) = \mathbb{E}[2xz] - 2z^2 = 2z\,\mathbb{E}[x] - 2z^2 = 2z \int_{-1}^{1} x \frac{1}{2}\,\mathrm{d}x - 2z^2 = -2z^2. \end{aligned}$$

(Note: it's pretty obvious that $\mathbb{E}[x] = 0$ for a uniformly distributed $x \in [-1, 1]$; the integral is not required.)

(c) [6 pts] What is the variance of your model $h(z)$ as a function of a test point $z \in \mathbb{R}$? Your final variance should not include an x; work out the expectation.

$$\text{Var}(h(z)) = \text{Var}(2xz) = \mathbb{E}[4x^2 z^2] - \mathbb{E}[2xz]^2 = \int_{-1}^{1} 4x^2 z^2 \frac{1}{2}\,\mathrm{d}x - 4z^2\,\mathbb{E}[x]^2 = \frac{2}{3}x^3 z^2 \Big|_{-1}^{1} - 0 = \frac{4}{3}z^2.$$

An alternative answer is

$$\text{Var}(h(z)) = \text{Var}(2xz) = 4z^2\,\text{Var}(x) = 4z^2\,\mathbb{E}[(x - \mathbb{E}[x])^2] = 4z^2 \int_{-1}^{1} x^2 \frac{1}{2}\,\mathrm{d}x = \frac{2}{3}z^2\, x^3 \Big|_{-1}^{1} = \frac{4}{3}z^2.$$

(d) [5 pts] Let $R(h, z)$ be the risk (expected loss) for a fixed, arbitrary test point $z \in \mathbb{R}$ with the noise-free label $g(z)$ (where the expectation is taken over the distribution of values of (x, y)). What is the mathematical relationship between the risk $R(h, z)$, the bias of $h(z)$ at z, and the variance of $h(z)$ at z? What are the values (as numbers) of these three quantities for $z = 1$?

From class, recall the bias-variance decomposition: $R(h, z) = \text{bias}(z)^2 + \text{Var}(h(z))$. (There is no irreducible error because there is no noise in z's label.)

For $z = 1$, we have $\text{bias}(z) = -2$, $\text{Var}(h(z)) = \frac{4}{3}$, and therefore $R(h, z) = (-2)^2 + \frac{4}{3} = \frac{16}{3}$.