CS 189 Introduction to Machine Learning
Fall 2017 Final

After the exam starts, please write your student ID (or name) on every odd page. On questions 2 and 7, select *all* correct answers. On questions 3-6, justify and show all your work. On question 7, you will be graded on your top *five* out of the ten parts. You may consult only two sheets of notes. Calculators, phones, computers, and other electronic devices are not permitted. There are **31** single sided pages on the exam. **Notify a proctor immediately if a page is missing.** You may, without proof, use theorems and lemmas that were proven in the notes and/or in lecture, unless we explicitly ask for a derivation. **You have 180 minutes**: there are 7 questions with 40 total parts on this exam, worth a total of 170 points, including bonus questions. Parts 5d and 6f are optional bonus questions, and count for 20 points of extra credit out of the 170. The multivariate Gaussian PDF is included on the last page; it is useful but may not be needed.

Exam Location: 310 Soda

PRINT and SIGN Your Name: _____, _____, _____
(last) (first) (signature)

PRINT Your Student ID: _____

Person before you: _____, _____
(name) (SID)

Person behind you: _____, _____
(name) (SID)

Person to your left: _____, _____
(name) (SID)

Person to your right: _____, _____
(name) (SID)

Row (front is 1): _____ Seat (leftmost is 1): _____ (*Include empty seats/rows.*)

1 Pre-exam Questions (4 points)

(a) **(2 points)** What is your favorite fruit? Your favorite vegetable?

(b) **(2 points)** What was your favorite machine learning topic?

Do not turn this page until your instructor tells you to do so.

Extra page. If you want the work on this page to be graded, mention it on the problem's main page.

2 Potpourri: Part I (20 points)

For the following multiple choice questions, select all that apply. There is a partial credit scheme for some questions. **Bubble in your answers. Do not draw an X or just leave a check.**

(a) Which of the following is/are true? Select all that apply.

◯ In general, generative models have fewer parameters to fit.

◯ SVM is not a generative-model based classification scheme.

◯ Logistic regression is a discriminative model.

◯ Generative models are used for classification by estimating the probability that the observed data came from this model.

(b) Which of the following statements is/are true? Select all that apply.

◯ The dual objective function for SVMs is cubic.

◯ The dual solution for SVMs is usually sparse.

◯ The SVM dual optimization only depends on the training points' positions $\vec{x}_i$ through the Gram matrix consisting of pairwise inner products between training points.

◯ In a hard margin classifier with margin M, all the training points can be perturbed by less than a quarter of the margin ($M/4$) and they still will be correctly classified with the same classifier.

(c) Which of the following can be naturally kernelized? Select all that apply.

◯ SVM

◯ Ridge Regression

◯ Decision Trees

◯ k-means

◯ A 3-layer Neural Network with sigmoid activation functions starting from raw data.

(d) Which of the following is/are true? Select all that apply.

◯ In random forests you always take an arithmetic mean for the final output.

◯ You can increase diversity in random forests by allowing a different subset of features in each tree.

◯ We want to have diverse trees in random forests to reduce the "bias" of the classifier.

◯ You can increase diversity in random forests by resampling training data subsets to build each tree.

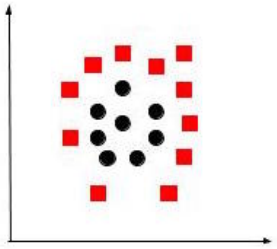
(e) Which of the following can be used to reduce overfitting in Neural Networks? Select all that apply.

- ◯ Training longer
- ◯ Injecting noise to input training data
- ◯ Adding parameter L_1 norm penalty
- ◯ Using a Gaussian prior over the weight vector
- ◯ Weight sharing

(f) Which of the following is/are true about the L_1 and the L_2 penalty? Select all that apply.

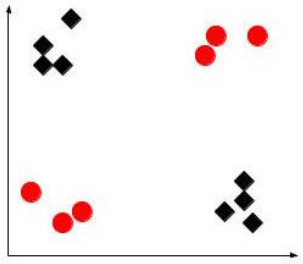
- ◯ L_1 penalty will often result in sparse solutions.
- ◯ L_2 penalty will often result in sparse solutions.
- ◯ L_2 penalty is equivalent to using an i.i.d. zero mean Gaussian prior on parameters.
- ◯ L_1 penalty is equivalent to using an i.i.d. Laplacian prior $\left(p(x) = \frac{1}{2b}\exp(-\frac{|x|}{b})\right)$ on parameters.
- ◯ Using L_1 regularization in linear regression will result in a cost function that is not convex.

(g) Which of the following classifiers can be used to classify the training data *perfectly*? Methods are not kernelized unless explicitly stated so.



- ◯ Logistic regression
- ◯ Hard margin SVM
- ◯ 3-nearest neighbors
- ◯ LDA
- ◯ SVM with a quadratic kernel
- ◯ QDA

(h) Which of the following classifiers can be used to classify the data *perfectly*? Methods are not kernelized unless explicitly stated so.



◯ Logistic regression without a kernel

◯ Hard margin SVM

◯ 3-nearest neighbors

◯ LDA

◯ SVM with a quadratic kernel

◯ QDA

(i) Which of the following is/are correct about mini-batch stochastic gradient descent? Select all that apply.

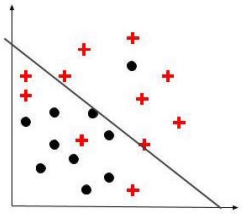
◯ One iteration of mini-batch gradient descent over a mini-batch of samples is faster to compute than one iteration of batch gradient descent

◯ Using a smaller mini-batch size always leads to faster convergence in terms of the number of iterations.

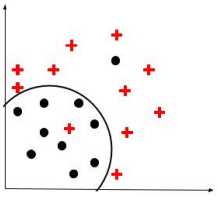
◯ A larger mini-batch size reduces the variance in the estimate of the gradient.

◯ If the mini-batch size is 1 you will process all the training samples before adjusting the parameters.

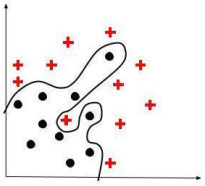
(j) For each classifier below, choose whether the classifier has high "bias," high "variance," or it is good enough for the given problem. Select all that apply.



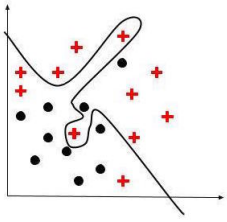
◯ High bias ◯ High variance ◯ Reasonable



◯ High bias ◯ High variance ◯ Reasonable



◯ High bias ◯ High variance ◯ Reasonable



◯ High bias ◯ High variance ◯ Reasonable

3 Love thy Neighbors (20 points)

In class (and a short homework problem), we covered the `k-Nearest-Neighbors` algorithm. To classify a test point x_i, we looked at the k closest points in the training set and took a plurality vote among their classes. In this problem, we will consider a variation where instead of classifying x_i based on the k closest points, we classify it based on the plurality vote of all points within a radius of R. This is the `R-Radius-Neighbors` algorithm.

Furthermore, recall that the k-Nearest-Neighbors algorithm (and the R-Radius-Neighbors algorithm) are slow at test time, since they required computing a lot of distances. To speed things up, we are going to use the following idea. Rather than checking all the points in the training set, we will randomly sample 1% of the training data. In particular, for every data point i, we will toss an independent, biased coin with probability of heads equal to 0.01, and include data point i in our sampled dataset if the coin comes up heads.

(a) (4 points) Let x_i be a test point that had n radius-R neighbors in the original dataset (i.e., n points within radius R). **What is that probability that it has no neighbors within radius R after the sampling procedure above?** Express your answer in terms of n. Please write your *final* answer in the box below.

(b) (5 points) Assume that we are in a binary classification setting, where the test point x_i had n^+ neighbors in the positive class and n^- neighbors in the negative class ($n = n^+ + n^-$). Let N^+ be a random variable which is the number of positive class neighbors of x_i after sampling the training data. Likewise, N^- is the number of negative class neighbors of x_i. **What is $E[N^+ - N^-]$?** Express your answer in terms of n^+ and n^-. Please write your *final* answer in the box below.

(c) (7 points) **Show that the variance** $\text{var}(N^+ - N^-) = 0.0099n$**.**

Hint: Recall that a single Bernoulli trial with probability p has variance $p(1-p)$.

(d) (4 points) Say that x_i is an element of the test set with a positive label. We say that x_i will be classified correctly *with high probability* if $E[N^+ - N^-]$ is at least two standard deviations above 0. Mathematically, we write

$$E[N^+ - N^-] - 2\sqrt{\text{var}(N^+ - N^-)} > 0.$$

In a huge data set, say that some x_i has $n = 10^4$ neighbors within radius R. Assume that the correctly label for x_i is the positive label. **How large does n^+ need to be in order for x_i to be classified correctly *with high probability* after sampling the training data?** Your answer should be an integer. You may assume that $\sqrt{0.0099} = 0.1$; recall that $n^+ + n^- = n$. Please write your *final* answer in the box below.

Extra page. If you want the work on this page to be graded, mention it on the problem's main page.

4 The k-Meaning of Life (32 points)

Recall that in k-means clustering we are attempting to minimize an objective defined as follows:

$$\min_{C_1,C_2,\ldots,C_k} \sum_{i=1}^{k} \sum_{x_j \in C_i} \|x_j - \mu_i\|_2^2, \text{ where}$$

$$\mu_i = \text{argmin}_{\mu_i \in \mathbb{R}^d} \sum_{x_j \in C_i} \|x_j - \mu_i\|_2^2 = \frac{1}{|C_i|} \sum_{x_j \in C_i} x_j, \;\; i = 1, 2, \ldots, k.$$

The samples are $\{x_1, \ldots, x_n\}$, where $x_j \in \mathbb{R}^d$, and C_i is the set of samples assigned to cluster i. Each sample is assigned to exactly one cluster, and clusters are non-empty.

(a) (3 points) **What is the minimum value of the objective when $k = n$ (the number of clusters equals the number of samples)?** Please write your *final* answer in the box below.

(b) (5 points) **Prove that the best clustering on $k+1$ clusters is at least as good as the best clustering on k clusters (in terms of the objective function).** We assume $k \leq n-1$.

(c) (8 points) Suppose we add a regularization term to the above objective. That is, the objective now becomes

$$\sum_{i=1}^{k} \left(\lambda \|\mu_i\|_2^2 + \sum_{x_j \in C_i} \|x_j - \mu_i\|_2^2 \right).$$

Show that the optimum of

$$\min_{\mu_i \in \mathbb{R}^d} \lambda \|\mu_i\|_2^2 + \sum_{x_j \in C_i} \|x_j - \mu_i\|_2^2$$

is obtained at $\mu_i = \frac{1}{|C_i|+\lambda} \sum_{x_j \in C_i} x_j$.

(d) (10 points) Now we add a feature (lifting) map $\phi : \mathbb{R}^d \to \mathbb{R}^p$ with $p \gg d$, with the inner product in $\mathbb{R}^p$ expressed by a kernel function k, as

$$k(x,y) = \langle \phi(x), \phi(y) \rangle.$$

The objective of regularized k-means now becomes:

$$\sum_{i=1}^{k} \left(\lambda \|\mu_i\|_2^2 + \sum_{x_j \in C_i} \|\phi(x_j) - \mu_i\|_2^2 \right).$$

As in the previous part, the representative of the ith cluster is $\mu_i = \frac{1}{|C_i|+\lambda} \sum_{x_j \in C_i} \phi(x_j)$. **Show that $\|\phi(x_j) - \mu_i\|_2^2$ can be expressed only in terms of kernel functions k, input data $x_j, j = 1, \ldots, n$, and partitions C_i, as:**

$$\|\phi(x_j) - \mu_i\|_2^2 = k(x_j,x_j) - \frac{2}{|C_i|+\lambda} \sum_{x_s \in C_i} k(x_s,x_j) + \frac{1}{(|C_i|+\lambda)^2} \sum_{x_s \in C_i} \sum_{x_t \in C_i} k(x_s,x_t).$$

(e) (6 points) Based on the above part, we can design a regularized kernel k-means algorithm which avoids computation of μ_j in a higher dimensional space. We use c to denote the number of clusters. **Fill in the underlined portion of this algorithm.** No need to explain; you will only be evaluated on your answer.

Algorithm 1 Kernel K-means

Require: Data matrix $X \in \mathbb{R}^{n \times d}$; Number of clusters c, regularization λ.
Ensure: Cluster id $i(j)$ for each sample x_j.
function KERNEL-K-MEANS(X, c)
Randomly initialize $i(j)$ to be an integer in $1, 2, \ldots, k$ for each x_j.
while not converged **do**
for i=1,...,c **do**
Set $C_i = \{j \in \{1, 2, \ldots, n\} : i(j) = i\}$.
end for
for j=1,...,n **do**
Set $i(j) = \text{argmin}_i$ **Fill in the objective function here composed of kernel evaluations only.**
end for
end while
Return C_i for $i = 1, 2, \ldots, c$.
end function

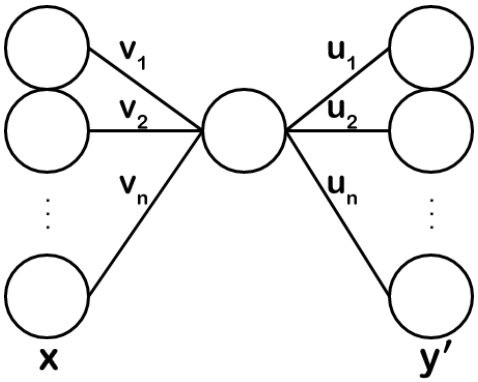
Extra page. If you want the work on this page to be graded, mention it on the problem's main page.

5 Neural networks and autoencoders (27 points)

To gain a better understanding of what neural networks learn, let's consider a simple network with two dense layers, linear activations, and no biases. This should remind you also of the simple linear autoencoder you saw in class, although here we will not be interested in making the input equal to the output. We will also restrict to the case of one neuron in the hidden layer. Because we use a linear activation (no nonlinearity between each layer), the output of the network can be given by

$$y' = uv^\top x$$

where the input $x \in \mathbb{R}^d$, the output $y' \in \mathbb{R}^d$, and the linear layers are given by $u \in \mathbb{R}^{d\times 1}$ and $v^\top \in \mathbb{R}^{1\times d}$. An example of the network is shown in the diagram below:



Given n samples, form an input matrix $X \in \mathbb{R}^{d\times n}$ by stacking samples in its columns, and the corresponding output matrix $Y \in \mathbb{R}^{d\times n}$ that is generated by stacking up the outputs as columns, but with noise. Assume $n \geq d$. We can express the squared error loss function as

$$L(u,v) = ||Y - uv^\top X||_F^2.$$

In this problem, we will assume that our input is whitened, i.e. $XX^\top = I$, and so the derivatives are given by

$$\frac{1}{2}\frac{\partial L}{\partial u} = -(Y - uv^\top X)(v^\top X)^\top = -YX^\top v + \|v\|_2^2 u, \text{ and}$$
$$\frac{1}{2}\frac{\partial L}{\partial v^\top} = -u^\top (Y - uv^\top X)X^\top = -u^\top YX^\top + \|u\|_2^2 v^\top.$$

In the following parts, you will:

- Characterize all points where the loss function $L(u,v)$ has derivative zero.
- Show that the optimal u and v that minimizes $L(u,v)$ is related to a low-rank approximation of $YX^\top$.

Let us use the notation $Z \triangleq YX^\top$ for convenience.

(a) (10 points) To start, let's identify a set of stationary points in the loss. We will represent the matrix of interest via its SVD as

$$Z \triangleq YX^\top = U\Sigma V^\top = \sum_{i=1}^{d} \sigma_i u_i v_i^\top$$

Assume that there are d nonzero singular values so that all the matrices $U, \Sigma, V^\top \in \mathbb{R}^{d\times d}$, with $\sigma_1 > \sigma_2 > \ldots > \sigma_d > 0$. The matrix Σ is thus a diagonal matrix of singular values.

Show that the choice $u' = u_i$ and $v' = \sigma_i v_i$ is a stationary point of the function L (i.e. having zero derivative with respect to both its arguments), for every choice of the index $i = 1, 2, \ldots, d$.

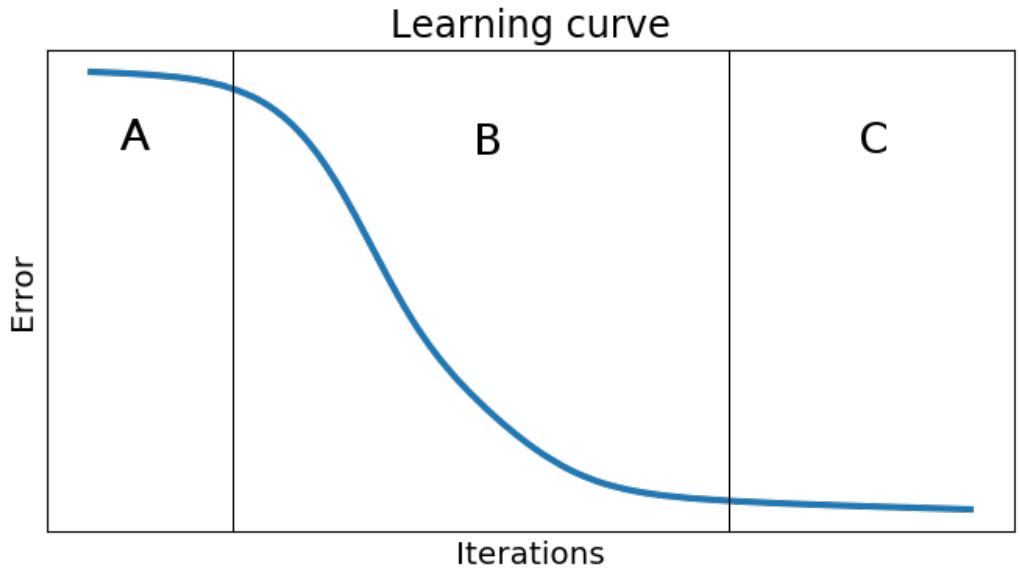
[What is true but won't be shown in this problem is that these are the *only* stationary points of the loss function.]

(b) (12 points) Among all the stationary points from the previous part, let us now compute the minimum. By evaluating the loss function L explicitly at all the stationary points $(u', v') = (u_i, \sigma_i v_i)$ for all $i \in \{1, 2, \ldots, d\}$, **show that the optimal argument is given by** $(u^*, v^*) = (u_1, \sigma_1 v_1)$**.**

Hint: You may use the fact that the loss function can be written as

$$L(u, v) = \mathsf{trace}(Y^\top Y) - 2\mathsf{trace}(YX^\top vu^\top) + \|u\|_2^2 \mathsf{trace}(vv^\top),$$

where trace denotes the sum of diagonal entries of a square matrix.



(c) (5 points: BONUS) While we didn't show it in this problem, all stationary points except for the global minimum are saddle points, i.e., there is some descent direction for these points, and they are not local minima. With this in mind, answer the following question.

You are given the following learning curve when running gradient descent on this problem with the "correct" step-size, guaranteeing that you won't diverge. At the end, u and v converge to the optimal solution. **Indicate which of the choices below describes what is happening in regions (A), (B), and (C) of the plot.** Each answer will be used at most once, and only one choice describes each region. Please bubble in your *final* answer in the corresponding blanks.

1. u, v has essentially converged to the global maximum $\arg\max_{u'',v''} L(u'', v'')$.
2. u, v has essentially converged to the global minimum $\arg\min_{u'',v''} L(u'', v'')$.
3. u, v is near a saddle point.
4. u, v is not close to any stationary point.

A. ◯ 1 ◯ 2 ◯ 3 ◯ 4
B. ◯ 1 ◯ 2 ◯ 3 ◯ 4
C. ◯ 1 ◯ 2 ◯ 3 ◯ 4

Extra page. If you want the work on this page to be graded, mention it on the problem's main page.

6 Dimensionality Reduction and Classification (57 points)

In this problem, you will explore different methods to transform data onto a lower-dimensional space. A good dimensionality reduction procedure is one that allows us to simultaneously preserve the structure in the dataset that we want to discover, while affording some form of compression that can then be used to speed up computations downstream.

We will work with a matrix $S \in \mathbb{R}^{s \times d}$ that transforms data points in d-dimensions to s-dimensions, i.e., for a point $x \in \mathbb{R}^d$, its transformation is given by the vector $Sx \in \mathbb{R}^s$. Assume that our dataset consists of n data points $x^{(1)}, x^{(2)}, x^{(3)}, \ldots, x^{(n)}$, each in d dimensions; half the data are from class C and the other half from class D. The labels are given by $y^{(i)} = 1$ if data point i has label C, and -1 otherwise. For concreteness, you should think of each data point as an image in high-dimensions, and we have half the images labeled $C =$ "*cats*" and the other half labeled $D =$ "*dogs*".

We have the following **three** assumptions on our data:

- For some $\varepsilon \geq 0$, the mean of data points in each of the classes is given by

$$\mu_C = (-\varepsilon, 0, 0, \ldots, 0) \in \mathbb{R}^d, \text{ and}$$
$$\mu_D = (\varepsilon, 0, 0, \ldots, 0) \in \mathbb{R}^d.$$

- We have the empirical covariances

$$\Sigma_C = \Sigma_D = \mathsf{diag}\left(\frac{1}{d^2}, \frac{1}{(d-1)^2}, \ldots, \frac{1}{2^2}, 1\right),$$

  where $\mathsf{diag}(v)$ denotes a $d \times d$ diagonal matrix having $[\mathsf{diag}(v)]_{ii} = v_i$.

- The maximum Euclidean distance of any point from its class mean is at most 5, i.e.,

$$\|x^{(i)} - \mu_C\|_2 \leq 5 \text{ for all } i \text{ labeled "} C = cat\text{"}$$
$$\|x^{(i)} - \mu_D\|_2 \leq 5 \text{ for all } i \text{ labeled "} D = dog\text{"}.$$

It may be helpful to draw yourself a schematic to visualize the data.

In parts (a) and (b), we will explore some standard data-dependent transformation algorithms that you have studied. In parts (c), (d), and (e), you will study some classification approaches in the original d-dimensional space. In the bonus part (f), you will be introduced to a data-independent transformation procedure, and you will see that classification can now be performed in a lower dimensional space, more quickly than before.

(a) (5 points) First, let us set the transformation matrix S such that we perform PCA on the *entire dataset* (including both classes C and D together), but ignoring labels. In effect, we want to transform the data onto the top s eigenvectors of the covariance matrix Σ of the entire dataset.

Show that if $\varepsilon = 0$, the ith row of S is given by $S_i^\top = e_{d-i+1}^\top$, where $e_i \in \mathbb{R}^d$ is the ith standard basis vector in d-dimensions (e.g. $e_1 = (1,0,0,\ldots,0) \in \mathbb{R}^d$).

(b) (6 points) Let us now bring back the labels as one-hot vectors in $\mathbb{R}^2$, i.e., $\ell^{(i)} = (1,0)$ if $y^{(i)} = 1$, and $\ell^{(i)} = (0,1)$ otherwise. For any fixed $\varepsilon > 0$ (no longer equal to zero), let us now perform CCA for predicting ℓ from x. This is run on the dataset by transforming all the data $\{x^{(i)}\}_{i=1}^n$ onto one vector by computing $Sx^{(i)}$, but $S \in \mathbb{R}^{1\times d}$ is now a row vector. **Which vector S would CCA choose, if we constrained the vector to be unit norm?** Please write your *final* answer in the box below.

Hint: Recall that LDA is a useful technique when the class covariances are the same. Also recall from homework that we established a precise connection between LDA and CCA.

(c) (8 points) **What is the minimum value of ε such that a 2-nearest neighbour algorithm can be guaranteed to incur zero error when a test data point is drawn from one of the training examples $x^{(1)}, x^{(2)}, \ldots, x^{(n)}$?**. Please write your *final* answer in the box below.

Hint: Recall all three assumptions on our data.

$\varepsilon =$ _____

(d) (13 points) Let $\varepsilon = 20$. **Show that there must exist a hyperplane separator having margin (distance between the hyperplane and the closest point of any one class) at least** 15. Proof by picture is fine.

With $\varepsilon = 20$, **which of the following hyperplanes could be a max-margin separator of the two classes? Bubble in and justify your answer.** (Here, x_i denotes the ith coordinate of the d-dimensional space, e.g. $i = 1$ corresponds to the first coordinate.)

Hint: Recall all three assumptions on our data.

- $x_1 = 0$.

  ◯ Yes ◯ No

  Justification:

- $x_2 = 0$.

  ◯ Yes ◯ No

  Justification:

- $x_1 = 10$.

  ◯ Yes ◯ No

  Justification:

(e) (10 points) With $\varepsilon = 20$, **which of the following methods would find a perfect (not necessarily max-margin) linear separator between the points? Again, bubble in and justify your answer.** Remember to recall all the assumptions on our data.

Notice that any linear separator can be written as $\widehat{w}^\top x + \widehat{b} = 0$. We set $\widehat{b} = 0$, and $\widehat{w}$ is obtained via

- $\widehat{w} = \arg\min_w \|w\|_2^2$ s.t. $y^{(i)} w^\top x^{(i)} \geq 1$ for all $i = 1, 2, \ldots, n$.

  ◯ Yes ◯ No

  Justification:

- Let the optimal value of the previous problem be denoted by

$$p^* = \min_w \|w\|_2^2 \text{ s.t. } y^{(i)} w^\top x^{(i)} \geq 1 \text{ for all } i = 1, 2, \ldots, n, \text{ and define}$$

$$\widehat{w} = \arg \min_{w : \|w\|_2^2 \geq p^*} \|w\|_2^2 + \sum_{i=1}^{n} \max(0, 1 - y^{(i)} w^\top x^{(i)}).$$

  ◯ Yes ◯ No

  Justification:

(f) (15 points: BONUS) In parts (a) and (b), we saw two data-dependent dimensionality reduction techniques. Let us now choose (in a *data-independent* manner) the entries of the transformation matrix $S \in \mathbb{R}^{s \times d}$ i.i.d. as

$$S_{ij} \sim \mathcal{N}(0, 1/s).$$

By a miracle of probability, it turns out that provided we choose $s \geq \frac{3 \log n}{\delta^2}$, we can guarantee that all n points $\{x^{(1)}, x^{(2)}, \ldots, x^{(n)}\}$ can be transformed onto s dimensions such that

$$(1-\delta)\|x^{(i)} - x^{(j)}\|_2 \leq \|Sx^{(i)} - Sx^{(j)}\|_2 \leq (1+\delta)\|x^{(i)} - x^{(j)}\|_2$$

for all pairs $i \neq j$ simultaneously with probability greater than $1 - 1/n$.

Let us now set $s = 12 \log n$. Use the fact above to show that the resulting dataset $Sx^{(1)}, Sx^{(2)}, \ldots, Sx^{(n)}$ is still linearly separable in s-dimensional space with probability greater than $1 - 1/n$.

Is the 2 nearest neighbor algorithm still guaranteed to work after the random transformations?

Hint: Think about the value of δ that setting $s = 12 \log n$ corresponds to.

Extra page. If you want the work on this page to be graded, mention it on the problem's main page.

7 Potpourri: Part II (10 points)

For the following multiple choice questions, select all that apply. There is a partial credit scheme for some questions. You will be graded on your best *five* out of the ten questions in this section, so feel free to skip things you're unsure about.

(a) Which of the following statements is/are true about SVMs? Select all that apply.

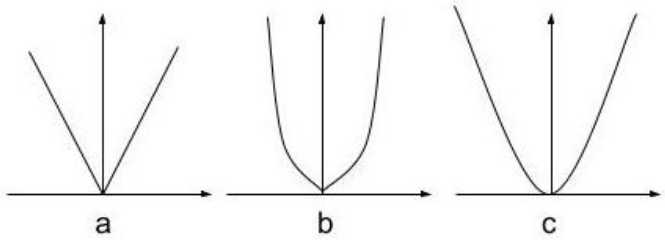
◯ We find at least one support vector from each class.

◯ Soft margin SVM effectively optimizes hinge loss plus ridge regularization.

◯ If a training point has a positive slack, then it is a support vector.

◯ Hinge loss is convex.

(b) Which of the following loss functions tends to result in a sparse solution? Select all that apply.



◯ a ◯ b ◯ c ◯ None

(c) Which of the following is true about decision trees? Select all that apply.

◯ Decision trees are less interpretable as compared to other classifiers.

◯ Decision trees are not prone to overfitting.

◯ Decision trees are sensitive to the choice of feature ordering.

◯ As compared to other classifiers, it is easier to deal with missing features in training points.

(d) Which of the following is/are true about AdaBoost? Select all that apply.

◯ The classifier is a linearly weighted combination of different classifiers.

◯ AdaBoost stops if it finds a classifier with zero training error.

◯ AdaBoost at step t reweights the training data so the boosted classifier found at step t is not better than a random guess.

◯ We can only use AdaBoost ideas on decision trees.

(e) Which of the following is/are true about GANs? Select all that apply.

◯ GAN is a supervised learning algorithm.

◯ In GAN we have two neural networks competing with each other.

◯ During GAN training, the generator's effective loss function changes.

◯ In GAN training, the gradient backpropagates through the generator into the discriminator.

(f) Assume that we are computing weights for a regression problem without any regularization and the weight vector is $w = [102, 431, 95]^T$. We tried different norm penalties to reduce variance and the resulting weight vectors are $w_1 = [4.35, 8, 1.65]^T$, $w_2 = [0, 1.87, 2.34]^T$, $w_3 = [0, 15.87, 23.7]^T$, and $w_4 = [10.31, 18, 10.65]^T$. Which of the following could be true? λ is the regularization hyperparameter.

◯ $w_1 = [4.35, 8, 1.65]^T$ can be the result of applying L_2 (ridge) regularization.

◯ The weights $w_3 = [0, 15.87, 23.7]^T$ result in a predictor that is more sensitive to perturbations of the input x as compared to the weights $w_2 = [0, 1.87, 2.34]^T$.

◯ $w_3 = [0, 15.87, 23.7]^T$ could be the result of applying L_1 regularization

◯ We applied a smaller λ to compute $w_1 = [4.35, 8, 1.65]^T$ compared to the λ used to compute $w_4 = [10.31, 18, 10.65]^T$

(g) Which of the following is/are true about LDA and QDA? Select all that apply.

◯ In general, QDA is more prone to overfitting than LDA.

◯ QDA and LDA are generative approaches for classification.

◯ In QDA the decision boundaries are always hyperplanes.

◯ LDA has fewer parameters to learn than QDA.

(h) Which is/are true about k-nearest neighbors (kNN)? Select all that apply.

◯ 1NN would have zero training error on the training set itself.

◯ Increasing k in kNN tends to result in less overfitting.

◯ You can use different distance measures for kNN.

◯ kNN always learns linear decision boundaries.

◯ kNN cannot be used with dimensionality reduction.

(i) Which of the following function classes can be represented exactly by a neural network with one hidden layer with threshold activation functions (i. e. step function) and a final output that is linear? Select all that apply.

◯ Polynomials of degree two

◯ Piecewise linear functions

◯ All continuous functions

◯ Piecewise constant functions

(j) Assume that k_1 is a valid kernel taking $\mathbb{R}^d \times \mathbb{R}^d \mapsto \mathbb{R}$. Which of the following is/are necessarily a valid kernel for all valid choices of k_1 and k_2? Select all that apply.

◯ $k(x,z) = k_1(x,z) + C$ where $C > 0$.

◯ $k(x,z) = k_1(f(x), f(z))$ where $f : \mathbb{R}^d \mapsto \mathbb{R}^d$.

◯ $k(x,z) = h(k_1(x,z))$ where $h : \mathbb{R} \mapsto \mathbb{R}$ is a degree 2 polynomial.

◯ $k(x,z) = k_1(f(x), g(z))$ where $f : \mathbb{R}^d \mapsto \mathbb{R}^d$ and $g : \mathbb{R}^d \to \mathbb{R}^d$ are different functions.

The PDF of a multivariate n-dimensional Gaussian with mean μ and covariance matrix Σ is given by

$$f(x) = \frac{1}{(2\pi)^{n/2}\sqrt{\det(\Sigma)}} \exp\left(-\frac{1}{2}(x-\mu)^\top \Sigma^{-1}(x-\mu)\right).$$

Doodle page! Draw us something if you want or give us suggestions or complaints. You can also use this page to report anything suspicious that you might have noticed.