STANFORD UNIVERSITY
CS 229, Autumn 2019
Midterm Examination

Tuesday, November 5, 6:00pm-9:00pm

Question	Points
1 Short answers	/19
2 Multi-class GDA	/14
3 K-means	/11
4 Universal approximation	/21
5 L_2 regularization	/22
6 Representer theorem	/22
Total	/109

Name of Student: _____

SUNetID: _____@stanford.edu

The Stanford University Honor Code:

I attest that I have not given or received aid in this examination, and that I have done my share and taken an active part in seeing to it that others as well as myself uphold the spirit and letter of the Honor Code.

Signed: _____

Instructions

Logistics:

1. You are allowed 3 double-sided 8.5x11 inch pages of notes (handwritten or typed).
2. This exam has 26 pages in total. Please check that this is the case. If not, please let the teaching staff know immediately.
3. The use of electronic devices (except pre-approved medical devices) is **not** allowed during the entire duration of this exam.

***Additional rules for proofs in Problems 5 and 6*:**

1. **For questions asking for more than a brief explanation, we will check whether your reasoning is correct, whether you prove the desired result, and whether all your intermediary steps are valid.**
2. **As such, you must prove that multiplicative constants are non-negative when you move them through inequalities, that an argument is the global minimizer of a function if said function is convex, that inverses exist when you take them, etc.**

Please address any clarification you may need (technical or otherwise) to the teaching staff.

Formulas

- *Multivariate Gaussian probabiltity density function:*

$$\mathcal{N}(\mu, \Sigma) = \frac{1}{(2\pi)^{d/2}|\Sigma|^{1/2}} \exp\left(-\frac{1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu)\right)$$

- *Second-order multivariate Taylor expansion of f about x_0, for all $x \in domf$:*

$$f(x) = f(x_0) + (x-x_0)^\top \nabla_x f(x)\Big|_{x_0} + \frac{1}{2}(x-x_0)^\top \nabla_x^2 f(x)\Big|_{x_0}(x-x_0) + o\left(\|x-x_0\|_2^2\right)$$

- *Indicator function over an arbitrary set $\mathcal{S}$:*

$$\mathbb{1}_{\mathcal{S}} : x \mapsto \begin{cases} 1, & \text{if } x \in \mathcal{S} \\ 0, & \text{otherwise} \end{cases}$$

- *Probability mass function (pmf):*

$$p_X : x \mapsto \mathbb{P}(X = x)$$

  for a discrete random variable X, e.g.

$$p_X(x) = p^x(1-p)^{1-x}$$

  if $X \sim \text{Bernoulli}(p)$.

- *Probability density function (pdf):* f_X such that

$$\mathbb{P}[a \leq X \leq b] = \int_a^b f_X(x)dx$$

  for a continuous random variable X, e.g.

$$f_X(x) = \frac{1}{\sqrt{2\pi}} e^{-x^2/2}$$

  if $X \sim \mathcal{N}(0,1)$.

Notations

- $\mathbb{R}_+$: the set of non-negative real numbers.
- $\mathbb{S}_+^d$: the set of $d \times d$ PSD matrices.
- $[\![1, n]\!]$: $\{1, 2, \ldots, n\}$

1. **[19 points] Short answers**

   (a) **[4 points] Loss functions for different tasks**
   Consider the following loss functions:

   A. $-(y \log \hat{y} + (1-y)\log(1-\hat{y}))$
   B. $\frac{1}{2}(\hat{y}-y)^2$
   C. for $k > 2$, $-\sum_{j=1}^{k} \mathbf{1}\{y = j\} \log \hat{y}_j$
   D. $\max(0, 1 - y\hat{y})$

   For each of the given tasks below, *circle one or more* letters, corresponding to the list above, that would be most appropriate to use in a learning algorithm to solve the task.
   If you deem none of the above options adequate, please circle **None**. For each choice, you gain 1 point if and only if your answer is *entirely* correct.

   i. Estimate the number of hours that a problem set will take.

   A **B** **C** **D** **None**

   ii. Group similar students together based on what classes they've taken.

   A **B** **C** **D** **None**

   iii. Predict whether or not the average house price in SF will increase this year.

   A **B** **C** **D** **None**

   iv. Predict whether tomorrow will be a rainy, cloudy or sunny day.

A **B** **C** **D** **None**

(b) **[10 points] SVM and kernels**
We generate 100 examples using three features that are sampled independently from a standard Gaussian distribution. For the i^{th} example, denote those features $x_1^{(i)}$, $x_2^{(i)}$ and $x_3^{(i)}$. The labels are binary, i.e. the i^{th} label, $y^{(i)}$, is either 0 or 1. Specifically, for 20 training examples $i \in [\![1, 20]\!]$:

$$y^{(i)} = \begin{cases} 1, & \text{if } \sum_{j=1}^{3} {x_j^{(i)}}^2 \geq c \\ 0, & \text{otherwise} \end{cases}$$

i. **[2 points]** Qualitatively assess the performance in terms of *separability* of using SVM in the original feature space. Briefly explain your answers.

ii. **[2 points]** Qualitatively assess the performance in terms of *generalizability* of using SVM with a 15-degree polynomial kernel. Briefly explain your answers.

iii. [**4 points**] Which kernel(s) would you choose to parameterize SVM to separate above two classes to obtain both *separability* and *generalizability*? Select all that apply and briefly explain your answers.

A 2-degree polynomial kernel
B 20-degree polynomial kernel
C 200-degree polynomial kernel
D dot product kernnel

iv. [**2 points**] Suppose we augment the features by adding five additional noisy features sampled from a standard Gaussian distribution. How would augmenting feature space as such affect the test error of your kernelized SVM? Briefly explain your answers.

(c) **[5 points] Non-exponential family distribution**
Although we have seen many examples of exponential family distributions, there exist distributions which cannot be written in the exponential family form $p(x;\eta) = b(x)\exp\big(\eta^T T(x) - a(\eta)\big)$.
Consider the following options for distributions:

(1) Uniform distribution of parameters a, b, with pdf $f(x) = \frac{1}{b-a}\mathbb{1}_{[a,b]}(x)$.

(2) Exponential distribution of parameter λ, with pdf $f(x) = \lambda e^{-\lambda x}\mathbb{1}_{\mathbb{R}_+}(x)$.

(3) Binomial distribution of parameters n, p, with pmf

$$p(x) = \binom{n}{x} p^x (1-p)^{n-x}\mathbb{1}_{[\![1,n]\!]}(x)$$

where $\binom{n}{x} = \frac{n!}{x!(n-x)!}$.

(4) Binomial distribution of parameter p and known n, with pmf being the same form as the one in (3).

(5) Multivariate Gaussian of parameters μ, Σ, with pdf

$$f(x) = \frac{1}{(2\pi)^{n/2}|\Sigma|^{1/2}} \exp\left(-\frac{1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu)\right)$$

Circle one or more of the above options corresponding to distributions that do *not* belong to the exponential family, and briefly explain why *not*. No explanation is needed for those that do belong to the exponential family. For each choice, you gain 1 point if and only if your answer is *entirely* correct, i.e. circled with explanation if the distribution is *not* in the exponential family and not circled with no explanation otherwise.

2. **[14 points] Multiclass GDA**

   In this problem we will explore a multi-class (k-class) extension of Gaussian Discriminant Analysis (GDA). Let us consider the model which has the following data generative process:

$$\begin{aligned} y &\sim \text{Categorical}(\phi) \\ x \mid y = 1 &\sim \mathcal{N}(\mu_{[1]}, \Sigma) \\ &\vdots \\ x \mid y = k &\sim \mathcal{N}(\mu_{[k]}, \Sigma), \end{aligned}$$

   where $y \in \{1, \ldots, k\}$ denotes the class label, $\phi \in \mathbb{R}^k$ are the parameters of the class prior categorical distribution such that $0 < \phi_j < 1$ for all j and $\sum_{j=1}^{k} \phi_j = 1$, $x, \mu_{[1]}, \ldots, \mu_{[k]} \in \mathbb{R}^d$ denote the input data and class means, and $\Sigma \in \mathbb{S}_+^d$ is the shared covariance matrix across all the k classes.

   Let us suppose that the parameters ϕ, $\mu_{[1]}, \ldots, \mu_{[k]}$ and Σ have been learned from the training data, and now we want to predict y for a new input x. Derive the form of the posterior distribution $p(y|x)$ for the above model. In particular, show that the posterior distribution takes the form of the softmax function:

$$p(y = j \mid x) = \frac{\exp\left\{b_j + \theta_{[j]}^T x\right\}}{\sum_{i=1}^{k} \exp\left\{b_i + \theta_{[i]}^T x\right\}}$$

   where $\theta_{[i]} \in \mathbb{R}^d$ and $b_i \in \mathbb{R}$ for all $i = 1, \ldots, k$.

   (a) **[6 points]** Provide expressions for b_i and $\theta_{[i]}$ (for any i, since they are all symmetric) in terms of $\phi, \mu_{[1]}, \ldots, \mu_{[k]}$ and Σ.

(b) **[3 points]** Suppose an input data point $x \in \mathbb{R}^d$ lies on the decision boundary between class j_1 and class j_2, where $(j_1, j_2) \in [\![1, k]\!]^2$ and $j_1 \neq j_2$. Write the condition x satisfies in terms of $\theta_{[j_1]}$, b_{j_1}, $\theta_{[j_2]}$, b_{j_2} and $\theta_{[j]}$, b_j for all $j \neq j_1, j_2$. As such, is the decision boundary between two classes affine?

(c) **[5 points]** Now suppose the covariance matrix for the prior of the first class becomes $c\Sigma$ with $c > 1$, and further suppose $\phi_j > 0$ for all $j \in [\![1, k]\!]$. Write the new decision boundary between class 1 and class 2 in terms of Σ, c,$\mu_{[1]}$,$\mu_{[2]}$, ϕ_1 and ϕ_2. Compared to when the covariance matrix was the same Σ across all classes, does the current posterior $p(y = 1|x)$ increase, decrease or stay unchanged for x such that $\|x-\mu_{[1]}\|_2 \approx 0$? What about for x such that $\|x-\mu_{[1]}\|_2$ is very large?

3. **[11 points] K-means**

Recall the k-means clustering algorithm on a set of data points $X = \{x^{(1)}, \ldots, x^{(n)}\}$:

- Initialize the cluster centroids $\mu_1, \mu_2, \ldots, \mu_k \in \mathbb{R}^d$ to be k randomly chosen points from X.
- Repeat until convergence: {

  For every i, set
  $$c^{(i)} := \arg\min_j \|x^{(i)} - \mu_j\|^2.$$

  For each j, set
  $$\mu_j := \frac{\sum_{i=1}^n \mathbf{1}\{c^{(i)} = j\} x^{(i)}}{\sum_{i=1}^n \mathbf{1}\{c^{(i)} = j\}}.$$

  }

Also, recall in the notes that we defined a distortion function as follows:

$$J(c, \mu) = \sum_{i=1}^n \|x^{(i)} - \mu_{c^{(i)}}\|^2$$

(a) **[3 points]** k-means is not guaranteed to converge to the global minimum. Provide an example of non-global convergence for $k = 2$. Specify (1) a set $X = \{x^{(i)} \in \mathbb{R}\}$ (i.e. for $d = 1$), (2) a set of converged cluster centroids and their cost J_1, and (3) a different set of converged cluster centroids and their cost $J_2 \neq J_1$.

(b) **[4 points]** Give a brief explanation why k-means will always converge to a locally optimal clustering.

(c) [**4 points**] Because k-means converges to local optima, a standard technique is to run the algorithm many times and pick the clusters of the run that yielded the lowest J. It turns out that with better initialization, we can greatly improve the performance of k-means, having it run in fewer iterations and converge to lower J. The k-means++ algorithm presents an initialization technique that generally performs much better than random initialization:

i. Choose an initial centroid $\mu_1 \in \mathbb{R}^d$ uniformly at random from X.

ii. With $D(x)$ defined as the shortest distance from a data point x to its closest centroid that has already been chosen, choose the next centroid μ_i by randomly sampling $x' \in X$ with probability

$$\frac{D(x')^2}{\sum_{x \in X} D(x)^2}$$

iii. Repeat step (ii) until all k centroids have been chosen.

iv. Proceed with regular k-means using these centroids.

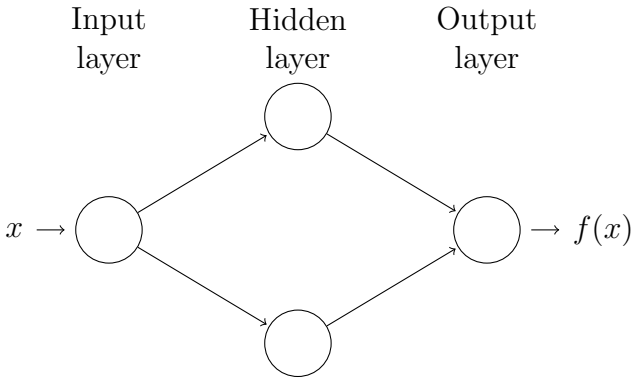
In 2-3 sentences, provide the intuition for why k-means++ will generally perform better than regular k-means.

4. **[21 points] Universal approximation for neural networks**

   In this problem, we develop an intuition for the flexibility of neural networks.

   (a) **[7 points]** Define a neural network with one hidden layer consisting of two hidden neurons to approximate the unit pulse function

$$f(x) = \begin{cases} 1 & -\frac{1}{2} \leq x < \frac{1}{2} \\ 0 & \text{otherwise} \end{cases}, x \in \mathbb{R}$$



   Using:

   - the sign function $g(z) = \mathbb{1}\{z \geq 0\}$ as your activation function in the hidden layer,
   - and the identity function as the activation function in the output layer,

   fully specify the dimensions and explicitly write the values of $W^{[i]}$ and $b^{[i]}$ for $i \in \{1, 2\}$, where $W^{[i]}$ and $b^{[i]}$ are the i^{th} layer's weight matrix and bias vector, respectively, such that the output of this neural network is exactly f.

(b) **[2 points]** Let ϕ_{ϵ,h,x_0} be the pulse function centered around x_0, of width 2ϵ and height h. Write the expression of ϕ_{ϵ,h,x_0} as a piecewise function.

(c) **[7 points]** Explicitly write the values of the weight matrices and bias vectors of the network shown in (a) such that the output is now **not** f, but ϕ_{ϵ,h,x_0}. Use the same activation functions as in (a). Express these values in terms of ϵ, h, and x_0.

(d) **[5 points]** Describe how the output of the generalized pulse function in part (c) can be used to approximate some arbitrary function $f : \mathbb{R} \to \mathbb{R}$ over a finite interval $[x_{\min}, x_{\max}]$ using a neural network. Draw a graph comparing $f(x)$ with the output of the neural network. Draw the structure of the neural network. No formal proof is required: a straightforward description in words, equations, and diagrams would suffice.

5. **[23 points] L_2 regularization under quadratic approximation**

   Consider a twice-differentiable and convex objective function, $J : \mathbb{R}^d \to \mathbb{R}$. Since J is twice-differentiable, you may assume the following result:

   $$\forall\ \theta \in \mathbb{R}^d,\ \forall\ (i,j) \in [\![1,d]\!]^2, \quad \frac{\partial^2 J(\theta)}{\partial \theta_i \partial \theta_j} = \frac{\partial^2 J(\theta)}{\partial \theta_j \partial \theta_i}$$

   As you have seen in class, a common technique to prevent overfitting is to add a regularization term to this objective that "shrinks" the weights. Suppose that we are applying L_2 regularization.

   (a) **[1 points]** Let $\theta \in \mathbb{R}^d$. Write the regularized objective, $\tilde{J}$, evaluated at θ, as a function of J and θ, denoting the regularization parameter by $c \geq 0$, and assuming a multiplicative factor of $\frac{1}{2}$ (for convenience).

   (b) **[2 points]** Derive the update rule if we were to use gradient descent for the *regularized* objective $\tilde{J}$ with a learning rate $0 < \alpha < 1/c$. Derive the multiplicative term in front of θ as function of α and c. Does θ decrease in this update rule?

(c) Note that the above is for one single step. Let's try to characterize what happens over the entire course of training. Recall that the goal in the *unregularized* case is to find the weights that minimize the objective J, i.e. find θ^* such that:

$$\theta^* = \arg\min_{\theta} J(\theta)$$

which is a different minimizer than

$$\hat{\theta} = \arg\min_{\theta} \tilde{J}(\theta)$$

Let's try to characterize the change from θ^* to $\hat{\theta}$ over the entire course of training.

i. **[7 points]** Let $H_{\theta^*} := \nabla_\theta{}^2 J(\theta)\Big|_{\theta=\theta^*}$.

Under the second order Taylor approximation of the *regularized* objective $\tilde{J}$ about θ^*, solve for $\hat{\theta}$ as a function of H_{θ^*}, θ^* and c.

ii. **[7 points]** Let $(o_1, \ldots, o_d)$ denote the orthonormal basis of eigenvectors of H_{θ^*}. Find the decomposition of $\hat{\theta}$ on $(o_1, \ldots, o_d)$, i.e. explicitly find the α_i's such that:

$$\hat{\theta} = \sum_{i=1}^{d} \alpha_i o_i$$

The expression of α_i's should **not** contain $\hat{\theta}$.
Hint 1: Any symmetric matrix S can be expressed as $O^\top \Lambda O$, where O is an orthonormal matrix and Λ is a diagonal matrix whose diagonal values are the eigenvalues of S (spectral theorem).
Hint 2: If O is orthonormal, then $O^\top = O^{-1}$.

iii. **[4 points]** Let λ_i denote the i^{th} eigenvalue for $i \in [\![1, d]\!]$. Give an interpretation of the above result for when $\lambda_i \gg c$ and when $\lambda_i \ll c$.

iv. **[2 points]** For which objective function J (seen in class) is the above approximation exact and why?

6. **[22 points] Representer theorem**

In the supervised learning setting, we have input data $x \in \mathbb{R}^n$ and label $y \in \mathcal{Y}$.

(a) Specify the set $\mathcal{Y}$ in the case of:

i. **[1 points]** linear regression

ii. **[1 points]** binary classification

iii. **[1 points]** multi-class (k-class) classification

(b) *(Representer theorem)*

For each of the above problems, we constructed a loss function $\ell : \mathbb{R} \times \mathcal{Y} \to \mathbb{R}$ with $\ell(\theta^T x, y)$ measuring the loss we suffer for predicting $\theta^T x$. For example, linear regression uses the squared residual for the loss: $\ell(\theta^\top x, y) = \frac{1}{2}(\theta^\top x - y)^2$. Let $\Omega : \mathbb{R} \to \mathbb{R}$ be a non-decreasing function. Consider the set of training examples $\{(x^{(i)}, y^{(i)}) \mid i \in [\![1, n]\!]\}$. Let $\tilde{J} : \mathbb{R}^d \to \mathbb{R}$ be the regularized objective across n examples:

$$\tilde{J} : \theta \mapsto \sum_{i=1}^{n} \ell(\theta^\top x^{(i)}, y^{(i)}) + \Omega\left(\|\theta\|_2\right)$$

Let $\theta \in \mathbb{R}^d$. In this question, we would like to show that there exists $\alpha \in \mathbb{R}^n$ such that:

$$\tilde{J}(\hat{\theta}) \leq \tilde{J}(\theta) \tag{5}$$

where

$$\hat{\theta} := \sum_{i=1}^{n} \alpha_i x^{(i)}$$

i. **[6 points]** Recall that for any subspace $\mathcal{S}$ of $\mathbb{R}^d$ (i.e. $S \subset \mathbb{R}^d$) we can define its orthogonal complement $\mathcal{S}^\perp = \left\{u \in \mathbb{R}^d \mid \forall v \in \mathcal{S}, u^T v = 0\right\}$. Also recall that the orthogonal decomposition theorem states that any vector $x \in \mathbb{R}^d$ can be uniquely written as:

$$x = x_{\mathcal{S}} + x_{\mathcal{S}^\perp}$$

where $x_{\mathcal{S}}$ and $x_{\mathcal{S}^\perp}$ are the projections of x onto $\mathcal{S}$ and $\mathcal{S}^\perp$, respectively.

Let $\theta \in \mathbb{R}^d$ and $\mathcal{S}_x := \text{span}\left\{x^{(i)} \mid i \in [\![1, n]\!]\right\}$. Denote $\theta_{\mathcal{S}}$ and $\theta_{\mathcal{S}^\perp}$ the projections of θ onto $\mathcal{S}_x$ and $\mathcal{S}_x^\perp$, respectively. Write the orthogonal decomposition of θ and express $\theta_{\mathcal{S}}$ and $\theta_{\mathcal{S}^\perp}$ in bases of $\mathcal{S}_x$ and $\mathcal{S}_x^\perp$, respectively. The values of the coefficients don't need to be explicitly written, but please do introduce relevant notation and specify the dimensions of each basis in terms of any of the following values: $\dim S_x$, n and d.

ii. **[5 points]** Show that $\|\theta\|_2 \geq \|\theta_{\mathcal{S}}\|_2$.

iii. [**8 points**] Now prove the theorem.

The implications of this theorem are far-reaching. In particular, if $\tilde{J}$ can be minimized, then by virtue of the representer theorem, its minimizer admits the following representation:

$$\theta^* = \sum_{i=1}^{n} \alpha_i x^{(i)}$$

and we can replace any occurrence of $\theta^\top x$ with $\theta^\top x = \sum_{i=1}^{n} \alpha_i x^\top x^{(i)}$ which is handy because the kernel trick $\phi(x)^\top \phi(x^{(i)}) := K(x, x^{(i)})$ can then be applied for some high-dimensional mapping ϕ, and we can directly solve for α instead.

Congratulations! You've reached the end of the midterm.