CM146 Final

Richard Sun

TOTAL POINTS

57 / 73

QUESTION 1

True/False 20 pts

1.1 (1) Convex, Ridge Regression, Ensemble, Perceptron 0 / 8

✓ **- 2 pts** 1. incorrect

✓ **- 2 pts** 2. incorrect

✓ **- 2 pts** 3. incorrect

✓ **- 2 pts** 4. incorrect

1.2 (5-8) PCA, K-means, Entropy, AdaBoost 6 / 8

✓ **- 2 pts** 6) false

1.3 (9-10) Dual, kernels 4 / 4

✓ **- 0 pts** Correct

QUESTION 2

Multiple Choice 18 pts

2.1 (11-14) Decision Tree, Normalization, Kernels, and NNs 11 / 12

✓ **- 1 pts** 12) a, c, d

2.2 (15) Eigenvalues 0 / 3

✓ **- 3 pts** 15) d)

2.3 (16) Leave-one-out 3 / 3

✓ **- 0 pts** Correct

QUESTION 3

3 (17) Performance Metrics 8 / 8

✓ **- 0 pts** Correct

QUESTION 4

(18) SVM 8 pts

4.1 (a) 4 / 4

✓ **- 0 pts** Correct

4.2 (b) 4 / 4

✓ **- 0 pts** Correct

QUESTION 5

5 (19) HMMs 1 / 2

✓ **- 1 pts** Used Bayes' Rule, but stops short of pulling denominator out of optimization

QUESTION 6

(20) GMMs 5 pts

6.1 (a) 2 / 2

✓ **- 0 pts** Correct

6.2 (b) 2 / 3

✓ **- 1 pts** partially incorrect steps

QUESTION 7

(21) Kernelized Logistic Regression 12 pts

7.1 (a) 4 / 4

✓ **- 0 pts** Correct

7.2 (b) 4 / 4

✓ **- 0 pts** Correct

7.3 (c) 4 / 4

✓ **- 0 pts** Correct

gradescope

CM 146 — Introduction to Machine Learning: Final

Fall 2017

Name: Richard Sun

UID: 904444918

Instructions

1. This exam is CLOSED BOOK and CLOSED NOTES.
2. The time limit for the exam is **3 hours**.
3. Mark your answers ON THE EXAM ITSELF IN THE SPACE PROVIDED. If you make a mess, clearly indicate your final answer (box it).
4. DO NOT write on the reverse side.
5. You may use scratch paper if needed.
6. For true/false questions, CIRCLE True OR False
7. For multiple-choice questions, CIRCLE ALL CORRECT CHOICES AND ONLY THE CORRECT CHOICES (in some cases, there may be more than one but always at least one correct choice) for full credit.
8. For all other questions, show the work that you did to arrive at your answer so that we can give you partial credit where appropriate.
9. If you think something about a question is open to interpretation, feel free to ask the instructor or make a note on the exam.

$x=1 \quad y=2 \quad \lambda=1$

$\theta=2$

$J(0)=(2-0)^2+\lambda 0=4$

$J(2)=(2-2)^2+\lambda 2^2=4$

True/False

1. (2 pts) A convex function always has a finite minimum value.

   (True) False

2. (2 pts) The solution to ridge regression (*i.e.*, the minimizer of $J(\boldsymbol{\theta}) = \sum_{n=1}^{N} \left(y_n - (\theta_0 + \sum_{d=1}^{D} \theta_d x_d)\right)^2 + \lambda \sum_{d=1}^{D} \theta_d^2$) is always unique for any $\lambda > 0$.

   True (False)

   $J(\theta)=(2-\theta)^2+\lambda\theta$

   $\theta=0,2$

   $J(\theta)=(y-\theta x)^2+\lambda\theta^2$

   $\frac{\partial}{\partial\theta} J(\theta)=2(y-\theta x)(-\theta)+2\lambda\theta=0$

   $-2y\theta+\theta^2 x+2\lambda\theta=0$

   $\theta(\theta x-2y+2\lambda)=0$

   $\theta=\frac{2y-2\lambda}{x}$

3. (2 pts) Consider an ensemble learning algorithm for binary classification that uses simple majority voting among 3 learned hypotheses. Suppose each of the hypotheses has training error ϵ. The error of the ensemble on the same training data can be worse than ϵ.

   True (False)

   better than random

   1 2

   1 2

   2 3

4. (2 pts) Consider two perceptron classifiers both trained on the same linearly-separable training data where one perceptron has maxIter=1000, but the other perceptron has maxIter=2000. The perceptron with maxIter=2000 might have worse training accuracy than the perceptron with maxIter=1000.

   True (False)

9. (2 pts) For a constrained optimization problem, we can always obtain the solution to the primal by solving the dual instead.

True False

10. (2 pts) For a valid kernel function k, $k(\boldsymbol{x}, \boldsymbol{x}) \geq 0$ for all $\boldsymbol{x}$.

True False

15. (3 pts) Let $\lambda_1 > \lambda_2 > \ldots > \lambda_d$ be the eigenvalues of the sample covariance matrix $\boldsymbol{C}$. The solution to the optimization problem $max_{\boldsymbol{x}} \boldsymbol{x}^T \boldsymbol{C} \boldsymbol{x}$.

(a) λ_1
(b) λ_d
(c) 0
(d) ∞

Short answers

17. (8 pts) **Performance metrics**
Consider a linear hypothesis that we use to make predictions for a binary classification problem where the two classes are denoted $\{0, 1\}$ and $h_{\boldsymbol{w},b} = \text{SIGN}(\boldsymbol{w}^T\boldsymbol{x} + b)$ models the probability that $\boldsymbol{x}$ has label 1. We assume that class 1 represents positives and class 0 represents negatives. What happens to the following as we increase b ? (Choose all that apply)

(a) recall can ... increase / decrease / stay the same

(b) the number of positives can ... increase / decrease / stay the same

(c) specificity can ... increase / decrease / stay the same

(d) precision can ... increase / decrease / stay the same

(b) (4 pts) What is the effect of increasing C on the following quantities? ↑ cost of slack

i. The margin

Decrease

As $C \to \infty$, $\xi_n \to 0 \Rightarrow$ hard margin SVM

ii. The number of support vectors

Decrease

Points inside the margin but classified correctly no longer need support vectors.

20. (5 pts) **Gaussian Mixture Models**

We now consider clustering 1D data using a **Gaussian Mixture Model**. We assume the number of components of the mixture (equivalently the number of clusters) $K = 2$. You are given three instances: $(x_1, x_2, x_3) = (1, 10, 20)$ where each $x_n \in \mathbb{R}, n \in \{1, 2, 3\}$. We use the EM algorithm to maximize the likelihood. Suppose the output of the E-step is the following matrix:

$$\Gamma = \begin{pmatrix} 1 & 0 \\ 0.4 & 0.6 \\ 0 & 1 \end{pmatrix}$$

Here entry (n, k) of this matrix $\gamma_{n,k}$ is the posterior probability that instance n belongs to mixture component k. For this question, you can leave your final answer in the form $\frac{a}{b}$.

(a) (2 pts) Show the M-step update for the mixing weights π_1, π_2.

$$\pi_k = \frac{\sum_n \gamma_{nk}}{\sum_k \sum_n \gamma_{nk}} \qquad \sum_k \sum_n \gamma_{nk} = 3$$

$$\pi_1 = \frac{1 + 0.4 + 0}{3} = \frac{1.4}{3}$$

$$\pi_2 = \frac{0 + 0.6 + 1}{3} = \frac{1.6}{3}$$

21. (12 pts) **Kernelized logistic regression**

In this problem, we explore how logistic regression can be kernelized.

We are given a set of N training examples, $\{(\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_N, y_N)\}$ where $\boldsymbol{x}_n \in \mathbb{R}^D$, $y_n \in \{0, 1\}$. We learn a logistic regression model $h_{\boldsymbol{\theta}}(\boldsymbol{x}) = \sigma(\boldsymbol{\theta}^T \boldsymbol{x})$ using gradient descent where $\sigma(x) = \frac{1}{1+e^{-x}}$ is the sigmoid function.

In iteration t of gradient descent, we update $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} - \eta \sum_n \epsilon_n \boldsymbol{x}_n$ where $\epsilon_n = h_{\boldsymbol{\theta}}(\boldsymbol{x}_n) - y_n$ is the error for the n^{th} training sample, and η is the step size or learning rate.

We map $\boldsymbol{x}$ to $\boldsymbol{\phi}(\boldsymbol{x})$ and we would like to learn a logistic regression model $\sigma(\boldsymbol{\theta}^T \boldsymbol{\phi}(\boldsymbol{x}))$ while only working with the inner products $\boldsymbol{\phi}^T(\boldsymbol{x})\boldsymbol{\phi}(\boldsymbol{x}')$.

(a) (4 pts) Assume we initialize $\boldsymbol{\theta}$ to zero in the gradient descent algorithm, *i.e.*, $\boldsymbol{\theta} \leftarrow \mathbf{0}$. Show that at the end of every iteration of gradient descent, $\boldsymbol{\theta}$ is always a linear combination of the training samples: $\boldsymbol{\theta} = \sum_{n=1}^{N} \alpha_n \boldsymbol{\phi}(\boldsymbol{x}_n)$.

$t = 0$: $\theta = \sum_{n=1}^{N} \alpha_n \phi(x_n) = 0$, $\alpha_n = 0$ $n = 1 \ldots N$

$t > 0$ Suppose $\theta_{t-1} = \sum_n \beta_n \phi(x_n)$.

$$\theta_t = \theta_{t-1} - \eta \sum_n \epsilon_n \phi(x_n)$$

$$= \sum_n \beta_n \phi(x_n) - \eta \sum_n \epsilon_n \phi(x_n)$$

$$= \sum_n (\beta_n - \eta \epsilon_n) \phi(x_n) \qquad \alpha_n = \beta_n - \eta \epsilon_n \quad n = 1 \ldots N$$

By induction, θ is always a linear combination of the training samples.

(c) (4 pts) The final step in kernelization is to show that we do not need to explicitly store $\boldsymbol{\theta}$. Instead from part (a), we can implicitly update $\boldsymbol{\theta}$ by updating α_n. Show how α_n is intialized and how it is updated.

As shown in part (a),

Initialize $\alpha_n = 0 \quad n = 1 \dots N$

Update $\alpha_n \leftarrow \alpha_n - \eta \epsilon_n \quad n = 1 \dots N$

(α_n from the previous iteration)

From part (b), $\epsilon_n = h_\theta(x_n) - y_n$ only depends on inner products.

Richard Sun
904444918

Q	Problem	Points	Score
1-10	True/False	20	
11-15	Multiple choice	15	
16-19	Short answers	23	
20	Kernelized logistic regression	12	
Total		70	

5. (2 pts) A non-invertible covariance matrix does not permit PCA.

PSD

True (False)

6. (2 pts) For fixed prototypes, finding the cluster assignment that minimizes the K-means objective function is a convex problem.

(True) False

7. (2 pts) The entropy of a discrete probability distribution is maximized for a uniform distribution.

(True) False

8. (2 pts) In AdaBoost, the weight associated with each weak learner is never less than zero.

(True) False

Multiple choice

11. (3 pts) Suppose we have a binary decision tree trained using the ID3 algorithm with maximum depth, k, for a D-dimensional feature space with N training examples. The worst case cost of classifying an unseen datapoint is:

    (a) O(D)
    (b) O(log N)
    (c) O(kD)
    (d) O(k)

12. (3 pts) For which of the following algorithms can the results change on normalizing the features?

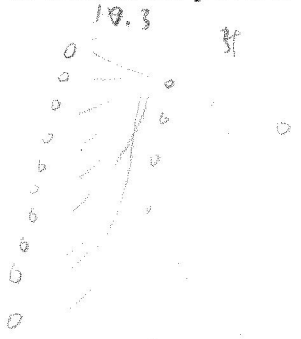
    (a) K-Nearest Neighbors
    (b) Decision trees
    (c) Neural networks
    (d) PCA

13. (3 pts) Given two kernel functions $k_1(\boldsymbol{u}, \boldsymbol{v})$ and $k_2(\boldsymbol{u}, \boldsymbol{v})$ that take as input two vectors $\boldsymbol{u}, \boldsymbol{v} \in \mathbb{R}^2$, which of the following are valid kernel functions ?

    (a) $k_1(\boldsymbol{u}, \boldsymbol{v}) + k_2(u, v)$
    (b) $k_1(\boldsymbol{u}, \boldsymbol{v}) - k_2(\boldsymbol{u}, \boldsymbol{v})$
    (c) $-k_1(\boldsymbol{u}, \boldsymbol{v})$
    (d) $\boldsymbol{u}^T \boldsymbol{A} \boldsymbol{v}$, $\boldsymbol{A} = \begin{pmatrix} 0.1 & 0 \\ 0 & 3 \end{pmatrix}$

14. (3 pts) Given a 3-layer neural network consisting of an input layer with 9 input units, a hidden layer with 3 units and an output layer with a single unit. Assume that the units in a given layer are connected to all units in the previous layer. The number of parameters in this network is:

    (a) 12
    (b) 30
    (c) 34
    (d) 44



16. (3 pts) Suppose you are running a learning experiment on a new algorithm for binary classification. You have a data set consisting of 100 positive and 100 negative examples. You plan to use leave-one-out cross-validation and compare your algorithm to a baseline function: a simple majority function. What is the average cross-validation accuracy of the baseline?

(a) 0.50
(b) 1.00
(c) 0.00
(d) Not enough information

$\frac{TP+TN}{P+N}$

Leave one positive: 99 P, 100 N $\rightarrow$ N

always wrong

Leave one negative: 100 P, 99 N $\rightarrow$ P

18. (8 pts) **SVM**
Recall the soft-margin SVM in the primal:

$$\begin{aligned} \arg\min_{\boldsymbol{w},b,\{\xi_n\}} \frac{1}{2}\|\boldsymbol{w}\|^2 + C\sum_{n=1}^{N}\xi_n \\ y_n(\boldsymbol{w}^T\boldsymbol{x}_n + b) &\geq 1-\xi_n \quad n\in\{1,\ldots,N\} \\ \xi_n &\geq 0 \quad n\in\{1,\ldots,N\} \end{aligned}$$

(a) (4 pts) Suppose you are given the solution to this problem but only for $(\boldsymbol{w}^*, b^*)$. Instances 1 and 2 are the support vectors. Compute the optimal values of the slack variables from these values.

constant

$$\arg\min_{\xi_1,\xi_2} \frac{1}{2}\|w^*\|^2 + C\sum_{n=1}^{2}\xi_n \qquad y_n(w^{*T}x_n + b^*) \geq 1-\xi_n$$

$$\xi_n \geq 0$$

$$= \arg\min_{\xi_1,\xi_2} C\xi_1 + C\xi_2$$

$$\min \xi_n \text{ s.t. } y_n(w^{*T}x_n + b^*) \geq 1-\xi_n,\ \xi_n \geq 0$$

$$\xi_n \geq 1 - y_n(w^{*T}x_n + b^*) \geq 0 \qquad 1 - y_n(w^{*T}x_n + b^*) > 0 \quad n=1,2$$

because they are support vectors

$$\Rightarrow \xi_n = 1 - y_n(w^{*T}x_n + b^*) \qquad \text{(otherwise } \xi_n = 0\text{)}$$

$$\xi_n = 1 - y_n(w^{*T}x_n + b^*) \quad n=1,2$$

$P(x|y) = \frac{P(y|x)P(x)}{P(y)}$

19. (2 pts) **Hidden Markov Model**
We want to compute the sequence of hidden states $x_{1:T}$ that has maximum posterior probability given the observations from T time points: $y_{1:T}$. Specifically we want to compute

$$\arg\max_{x_{1:T}} P(x_{1:T}|y_{1:T})$$

How does the solution to this problem compare to the solution to the most probable path problem discussed in class ? Justify.

This problem is the most probable path problem given observations $y_{1:T}$. So, the solution is the path that has the most probable state transitions and most probable emissions.

most probable path

$$\arg\max_{x_{1:T}} P(x_{1:T}|y_{1:T}) = \arg\max_{x_{1:T}} \frac{p(x_{1:T})\, p(y_{1:T}|x_{1:T})}{p(y_{1:T})}$$

(b) (3 pts) Show the M-step update for the means μ_1, μ_2.

$$\mu_k = \frac{\sum_n \gamma_{nk} x_n}{\sum_k \sum_n \gamma_{nk}}$$

$$\mu_1 = \frac{1(1) + 0.4(10) + 0(20)}{3} = \frac{5}{3}$$

$$\mu_2 = \frac{0(1) + 0.6(10) + 1(20)}{3} = \frac{26}{3}$$

(b) (4 pts) Using the above result, show how we can write $h_\theta(\boldsymbol{x})$ to make a prediction on a new input $\boldsymbol{\phi}(\boldsymbol{x})$ by only using inner products of the form $\boldsymbol{\phi}(\boldsymbol{x})^T\boldsymbol{\phi}(\boldsymbol{x}')$.

$$h_\theta(x) = \sigma\left(\theta^T \phi(x)\right)$$

$$= \sigma\left[\left(\sum_n \alpha_n \phi(x_n)\right)^T \phi(x)\right]$$

$$= \sigma\left[\left(\sum_n \alpha_n \phi(x_n)^T\right) \phi(x)\right]$$

$$= \sigma\left(\sum_n \alpha_n \phi(x_n)^T \phi(x)\right)$$

(Blank page provided for your work)