

- Due noon, March 24th, via Gradescope.
- No extensions, and you cannot use late hours.
- No collaboration, you must answer these questions all on your own.
- You are free to ask clarification questions on Piazza.
- This exam is designed to take 3-5 hours to do.
- Suggestion: Write the solutions by hand (no LaTeX) and submit a scanned copy. It will save you time.
- All questions have short answers. If your answer is very long, you're probably over-thinking it.

1 Multiple Choice Questions (35 points)

Just specify the correct answer. No need to justify.

Lasso (2 points)

Question A: (True or False) Every L1-constrained regression problem is equivalent to some L1-regularized regression problem.

Recall that an L1-constrained regression problem is:

$$\arg \min_{w,b} \sum_{(x,y) \in S} (y - (w^T x + b))^2, \quad \text{s.t.} \quad \|w\|_1 \leq C,$$

and that an L1-regularized regression problem is:

$$\arg \min_{w,b} \lambda \|w\|_1 + \sum_{(x,y) \in S} (y - (w^T x + b))^2.$$

Bagging & Bootstrap Sampling (3 points)

Question B: (Multiple Choice) Suppose we generate M bootstraps S'_1, \dots, S'_M of a training set $S = \{(x_i, y_i)\}_{i=1}^N$. For any specific $(x, y) \in S$, what is the probability that (x, y) does not appear in ANY of S'_1, \dots, S'_M ?

(Recall that a bootstrap is a dataset with the same size as S generated by sampling uniformly with replacement from S .)

Option A:

$$\left(1 - \frac{1}{N}\right)^N M$$

Option B:

$$\left(1 - \frac{1}{N}\right)^{NM}$$

Option C:

$$\left(1 - \frac{1}{NM}\right)^{NM}$$

Option D:

$$\left(1 - \left(1 - \frac{1}{NM}\right)^N\right)^M$$

Convolutional Filters (4 points)

Consider the three convolutional filters below:

$$K_1 = \begin{bmatrix} 0.0043 & 0.0144 & 0.0214 & 0.0144 & 0.0043 \\ 0.0144 & 0.0478 & 0.0712 & 0.0478 & 0.0144 \\ 0.0214 & 0.0712 & 0.1062 & 0.0712 & 0.0214 \\ 0.0144 & 0.0478 & 0.0712 & 0.0478 & 0.0144 \\ 0.0043 & 0.0144 & 0.0214 & 0.0144 & 0.0043 \end{bmatrix}$$

$$K_2 = \begin{bmatrix} 0.0068 & 0.0225 & 0.0335 & 0.0225 & 0.0068 \\ 0.0225 & 0.0746 & 0.1112 & 0.0746 & 0.0225 \\ 0.0335 & 0.1112 & 0.1660 & 0.1112 & 0.0335 \\ 0.0225 & 0.0746 & 0.1112 & 0.0746 & 0.0225 \\ 0.0068 & 0.0225 & 0.0335 & 0.0225 & 0.0068 \end{bmatrix}$$

$$K_3 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.8 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Question C: (Multiple Choice) If we apply K_1 , K_2 , and K_3 to Figure 1, which filter corresponds to which image in Figure 2?



Figure 1: Original Image. All pixel intensities are clipped between 0 and 1 (black=0 and white=1).



Figure 2: Convolved Images (hint: the grey background is intentional)

Multiclass SVMs (3 points)

Consider the following 4-class multiclass SVM objective:

$$\begin{aligned} \arg \min_{w, \xi} \quad & \frac{1}{2} \|w\|^2 + \frac{C}{N} \sum_{i=1}^N \xi_i \\ \text{s.t.} \quad & \forall i, \forall y' \in \{1, 2, 3, 4\} : w_{y_i}^T x_i - w_{y'}^T x_i \geq \mathbf{1}_{[y_i \neq y']} - \xi_i \end{aligned}$$

where:

$$w = \begin{bmatrix} w_1 \\ w_2 \\ w_3 \\ w_4 \end{bmatrix},$$

and predictions are made via $\arg \max_y w_y^T x$. In other words, each ξ_i can be defined as:

$$\xi_i = \max_{y' \in \{1, 2, 3, 4\}} \{ \mathbf{1}_{[y_i \neq y']} - (w_{y_i}^T x_i - w_{y'}^T x_i) \}. \quad (1)$$

Suppose for training data (x_i, y_i) we have $y_i = 1$ and:

$$\begin{aligned} w_1^T x_i &= 1.0, \\ w_2^T x_i &= 1.5, \\ w_3^T x_i &= 0.1, \\ w_4^T x_i &= -0.8. \end{aligned}$$

Question D: (Multiple Choice) Which $\hat{y} \in \{1, 2, 3, 4\}$ is the maximizer of (1)?

Feature Maps (3 points)

Consider the following 3-class multiclass feature map:

$$\phi(x, y) = \begin{bmatrix} \mathbf{1}_{[y=1]} x \\ \mathbf{1}_{[y=2]} x \\ \mathbf{1}_{[y=3]} x \\ -\mathbf{1}_{[y \in \{1, 2\}]} x \end{bmatrix}, \quad (2)$$

where predictions are made via $\arg \max_{y \in \{1, 2, 3\}} w^T \phi(x, y)$.

Question E: (Multiple Choice) What is the effect of using (2) versus a “conventional” multiclass feature map of:

$$\phi(x, y) = \begin{bmatrix} \mathbf{1}_{[y=1]} x \\ \mathbf{1}_{[y=2]} x \\ \mathbf{1}_{[y=3]} x \end{bmatrix}. \quad (3)$$

Option A: Compared to (3), in (2) the model scores for Class 1 & 2 are encouraged to be correlated.

Option B: Compared to (3), in (2) the model scores for Class 1 & 2 are encouraged to be anti-correlated.

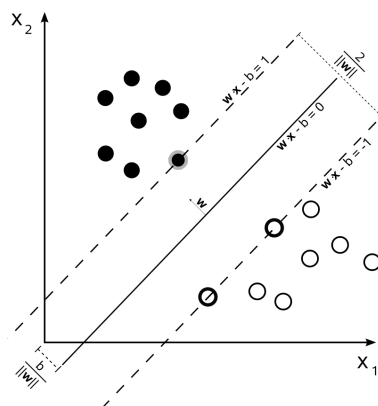
Option C: Either A or B are possible depending on the specific training data.

(Recall that the model score for a class y' is just $w^T \phi(x, y')$.)

Hard-Margin SVMs (5 points)

A hard margin SVM can be written as:

$$\begin{aligned} & \arg \min_{w,b} \frac{1}{2} \|w\|^2 \\ & \text{s.t.} \\ & \forall (x, y) \in S : y(w^T x - b) \geq 1 \end{aligned}$$



The figure to the right depicts a visualization of the hard-margin SVM for a 2-dimensional example (image source Wikipedia). The data points that are on the margin (for which the inequality constraint in the training optimization problem is a tight equality) are known as “support vectors”.

Question F: (True or False) If the data is not linearly separable, then a hard-margin SVM returns $w = 0$.

Question G: (Multiple Choice) Assume the data is linearly separable. What happens when we remove a data point that is NOT a support vector from the training set, and then retrain the model?

Option A: The solution w and b do not change from before.

Option B: The solution changes, and the training error is reduced.

Option C: The solution changes, and the training error is increased.

AdaBoost (2 points)

Question H: (True or False) Each iteration of AdaBoost is guaranteed to reduce the training 0/1 Loss of the aggregate model.

Recall that the aggregate model of AdaBoost at iteration T is:

$$f_T(x) = \sum_{t=1}^T \alpha_t h_t(x) \in \mathbb{R},$$

and predictions are made via the sign of $f_T(x)$: $h_T(x) = \text{sign}(f_T(x))$.

Tensor Model Training (2 points)

Question I: (True or False) Every L2-regularized tensor latent-factor regression problem can be optimized via alternating closed-form optimization (i.e., converges to a local optimum).

Recall that a L2-regularized tensor regression problem can be written as:

$$\arg \min_{U,V,W} \frac{1}{2} (\|U\|_F^2 + \|V\|_F^2 + \|W\|_F^2) + \frac{1}{2} \sum_{(a,b,c) \in S} (Y_{a,b,c} - \langle u_a, v_b, w_c \rangle)^2,$$

where u_a , v_b , and w_c correspond to the a -th column of U , the b -th column of V , and the c -th column of W , respectively. The three-way dot product is defined as:

$$\langle u_a, v_b, w_c \rangle = \sum_{k=1}^K u_{a,k} v_{b,k} w_{c,k}.$$

Alternating closed-form optimization implies that U (likewise V or W) can be solved optimally in closed-form if one holds the other two matrices V and W fixed.

Bias-Variance Decomposition (2 points)

Let $f_S \in F_S$ denote a set of linear regression models trained on training set S using different learning algorithms (e.g., each f_S is trained using a different regularization strength). Let \hat{f}_S denote the $f_S \in F_S$ that has the lowest bias in the bias-variance decomposition.

Question J: (True or False) The minimum bias learning algorithm is guaranteed to have the smallest test squared loss. In other words, the test error of \hat{f}_S is guaranteed to be lower than the test error of any other f_S :

$$\forall f_S \in F_S : E_{(x,y) \sim P(x,y)} E_S \left[(y - \hat{f}_S(x))^2 \right] \leq E_{(x,y) \sim P(x,y)} E_S [(y - f_S(x))^2],$$

where $P(x, y)$ denotes the test distribution.

HMM EM Learning (3 points)

Question K: (Multiple Choice) During EM training in the unsupervised setting, what best describes what happens when you initialize $P(y^j | y^{j-1})$ and $P(x^j | y^j)$ to both be the uniform distribution?

Option A: Nothing special, you just converge to some local optimum.

Option B: You converge to a degenerate local optimum where all the hidden states learn the same thing.

Option C: You get a divide-by-0 error and training crashes.

Recall that an HMM model is specified by:

$$P(x, y) = P(\text{End} | y^M) \prod_{j=1}^M P(x^j | y^j) P(y^j | y^{j-1}),$$

the unsupervised learning problem is specified by

$$\arg \max_{\Theta} \prod_{x \in S} P(x) = \arg \max_{\Theta} \prod_{x \in S} \sum_{y'} P(x, y'),$$

where S denotes the training set and Θ denotes all the HMM model parameters. The EM algorithm alternates between inferring the distribution of sequences y' for each training example x and using that inferred distribution to estimate better model parameters Θ .

Non-Negative Matrix Factorization (2 points)

Consider the matrix factorization problem of minimizing squared reconstruction error:

$$\arg \min_{U \in \mathbb{R}^{N \times K}, V \in \mathbb{R}^{M \times K}} \|Y - UV^T\|_{Fro}^2,$$

where $Y \in \mathbb{R}^{N \times M}$ is the matrix we wish to encode in a low-rank model, and U and V are our model components.

Question L: (True or False) Suppose Y is non-negative. If we enforce U and V to also be non-negative, then we typically need a larger latent dimension K to achieve the same squared reconstruction error compared to allowing U and V to take negative values.

Decision Trees (2 points)

Question M: (True or False) Given infinite training data (sampled i.i.d. from the test distribution), decision trees can essentially learn arbitrary classifier functions. (Ignore the distinction between countably vs uncountably infinite, this is a high-level conceptual question.)

Overfitting (2 points)

Question N: (True or False) Given infinite training data (sampled i.i.d. from the test distribution), it is essentially impossible to overfit. (Ignore the distinction between countably vs uncountably infinite, this is a high-level conceptual question.)

2 Naive Bayes (20 points)

We consider the following Naive Bayes model:

$$P(\text{Happy?}, \text{Grade}, \text{Year}) = P(\text{Happy?})P(\text{Grade}|\text{Happy?})P(\text{Year}|\text{Happy?}).$$

In other words, the y is the Happy? variable, and the two x 's are the Grade and Year variables. We assume that all variables take two values, $\text{Happy?} \in \{\text{Yes}, \text{No}\}$, $\text{Grade} \in \{A, C\}$, and $\text{Year} \in \{\text{Freshman}, \text{Senior}\}$.

Consider the following training data:

Grade	Year	Happy?
A	Senior	Yes
A	Senior	Yes
A	Senior	No
A	Freshman	Yes
C	Freshman	No
C	Freshman	No
C	Senior	No
C	Senior	Yes

Question 1: (8 points) Fit the model parameters of the Naive Bayes using maximum likelihood with uniform unit pseudocounts. For instance, the maximum likelihood estimate for $P(\text{Grade}|\text{Happy?})$ is:

$$P(\text{Grade} = A|\text{Happy?} = \text{Yes}) = \frac{1 + \sum_{(x,y)} 1_{[x_{\text{Grade}}=A \wedge y=\text{Yes}]}}{2 + \sum_{(x,y)} 1_{[y=\text{Yes}]}}.$$

(Check Lecture 12, Slide 38. Notice that we are using $\lambda = 2$ here.) Write out the final probability tables, i.e. $P(\text{Grade}|\text{Happy?})$, $P(\text{Year}|\text{Happy?})$ and $P(\text{Happy?})$.

Question 2: (5 points) Compute $P(\text{Year} = \text{Freshman}, \text{Grade} = C, \text{Happy?} = \text{No})$ using the trained model from Question 1.

Question 3: (7 points) Write out the pseudocode for drawing a sample from any trained model. Assume you have repeated access to a function `random()` that returns a uniform random number in $[0, 1]$.

3 Data Transformations (15 points)

We consider the problem of linear regression, where we predict real values given input features x via $w^T x$ using a linear model w (ignoring the bias term). Suppose we want to transform the data points x to a new representation via a transformation matrix: $\tilde{x} = Ax$. For instance, A can be a rescaling of the dimensions of x :

$$A = \begin{bmatrix} a_1 & 0 & \dots & 0 \\ 0 & a_2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & a_D \end{bmatrix}, \quad (4)$$

where each $a_d > 0$ scales each dimension.

Question 1: (5 points) What is the relationship between w and \tilde{w} ? In other words, write w as a function of \tilde{w} and A such that $w^T x = \tilde{w}^T \tilde{x}$ holds for all x . Assume that A is a square, invertible matrix.

Consider the ridge regression learning objective on the transformed data:

$$\arg \min_{\tilde{w}} \frac{\lambda}{2} \|\tilde{w}\|^2 + \sum_i (y_i - \tilde{w}^T \tilde{x}_i)^2, \quad (5)$$

Question 2: (5 points) Rewrite (5) using w , x , and A . In other words, what is the optimization problem that yields the w that corresponds to the \tilde{w} learned in (5) (with the correspondence established in Question 1)? Assume that A is a square, invertible matrix.

Question 3: (5 points) Interpret your answers to Question 1 and Question 2 when A is a rescaling transformation such as (4). In other words, how is your answer to Question 2 different from standard ridge regression for w :

$$\arg \min_w \frac{\lambda}{2} \|w\|^2 + \sum_i (y_i - w^T x_i)^2.$$

4 Latent Markov Embedding (15 points)

This problem deals with the song embedding model from [Playlist Prediction via Metric Embedding](#). The paper describes two models, a dual-point model and a single-point model. In the dual-point model, the transition probability from song s to song s' is:

$$P(s'|s) = \frac{e^{-\|U(s')-V(s)\|_2^2}}{Z(s)}. \quad (6)$$

The probability of a playlist $p = \langle p^{[1]}, \dots, p^{[M_p]} \rangle$ is:

$$P(p) = \prod_{i=1}^{M_p} P(p^{[i]}|p^{[i-1]}).$$

Note that there is a special “start” song, which you can ignore for the questions below. The total probability of a dataset S of playlists is then:

$$P(S) = \prod_{p \in S} P(p) = \prod_{p \in S} \prod_{i=1}^{M_p} \frac{e^{-\|U(p^{[i]})-V(p^{[i-1]})\|_2^2}}{Z(p^{[i-1]})}, \quad \text{with: } Z(s) = \sum_{s'} e^{-\|U(s')-V(s)\|_2^2}. \quad (7)$$

In the single-point model, there is only one vector X representing each song (in contrast to the two vectors U and V used in the dual-point model). The transition probability from song s to song s' is:

$$P(s'|s) = \frac{e^{-\|X(s')-X(s)\|_2^2}}{Z(s)}. \quad (8)$$

This results in the total probability of the dataset being:

$$P(S) = \prod_{p \in S} P(p) = \prod_{p \in S} \prod_{i=1}^{M_p} \frac{e^{-\|X(p^{[i]})-X(p^{[i-1]})\|_2^2}}{Z(p^{[i-1]})}, \quad \text{with: } Z(s) = \sum_{s'} e^{-\|X(s')-X(s)\|_2^2}. \quad (9)$$

For the following problems, assume that the optimal choices for U , V , and X are selected (i.e., the ones that maximize $P(S)$).

Question 1: (8 points) Show that the data likelihood $P(S)$ for the dual-point model (7) is never less than the data likelihood for the single-point model (9).

Question 2: (7 points) If (6) is equal to (8) for every pair of songs s and s' , what does that imply about the relationship between U , V , and X ?

(Hint: the answers to the two questions are related.)

5 Neural Net Backprop Gradient Derivation (15 points)

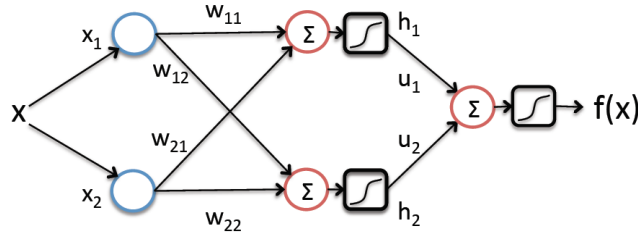


Figure 3: Illustration of Simple Neural Network.

In this question, we will consider the following neural network depicted in Figure 3. There are six parameters in the model, u_1 , u_2 , w_{11} , w_{21} , w_{12} , and w_{22} .

The network takes in a 2-dimensional input x and outputs a real value $f(x) \in [0, 1]$. The final output $f(x)$ is a weighted combination of two hidden node activations with a sigmoid transfer function:

$$f(x) = \sigma \left(\sum_{i=1}^2 u_i h_i(x) \right), \quad (10)$$

where:

$$\sigma(s) = \frac{e^s}{1 + e^s}, \quad \text{and} \quad h_i(x) = \sigma \left(\sum_{j=1}^2 w_{ji} x_j \right).$$

Question 1: (5 points) For a given training data point (x, y) , compute the stochastic gradient of the squared-loss of (x, y) w.r.t. w_{11} :

$$\frac{\partial}{\partial w_{11}} L(y, f(x)) \equiv \frac{\partial}{\partial w_{11}} (y - f(x))^2.$$

Hint: write the formula using the chain rule and use the following definition of the derivative of $\sigma(s)$:

$$\frac{\partial}{\partial s} \sigma(s) = \sigma(s)(1 - \sigma(s)).$$

Question 2: (5 points) Find $\frac{\partial}{\partial w_{11}} L(y, f(x))$ where:

$$(x, y) = ((0.1, 0.5), 0.75)$$

$$(u_1, u_2) = (0.5, -0.1)$$

$$(w_{11}, w_{12}, w_{21}, w_{22}) = (0.25, 0.1, 0.05, -0.25)$$

Question 3: (5 points) Use your derivations from the previous two questions to identify which term in the gradient derivation results in the vanishing gradient problem, and briefly explain how the problem is exacerbated in neural networks with more layers.