CSE 258, Fall 2019: Midterm

Name:	Student ID:

Instructions

The test will start at 6:40pm. Hand in your solution at or before 7:40pm. Answers should be written directly in the spaces provided.

Do not open or start the test before instructed to do so.

Note that the final page contains some algorithms and definitions. Total marks = 26

Section 1: Regression and Ranking (6 marks)

In this section we'll consider building predictors from (features extracted from) movies in the *Marvel Cinematic Universe*. A sample of such a datasets looks like the following:

Movie	Release date	(Incomplete) list of heroes	Budget	Box office	Rating
Capt. America: Civil War	May 6, 2016	CA; IM; BP; SW; AM; SM	$250M	$1.153B	91%
Doctor Strange	Nov 4, 2016	DS	$165M	$677M	89%
Spider-Man Homecoming	Jul 7, 2017	SM; IM	$175M	$880M	92%
Thor: Ragnarok	Nov 3, 2017	T; H	$180M	$854M	93%
Black Panther	Feb 16, 2018	BP	$200M	$1.347B	97%
Infinity War	Apr 27, 2018	IM; T; H; CA; DS; SM; BP	$316M	$2.048B	85%
Ant-Man and the Wasp	Jul 6, 2018	AM	$162M	$622M	88%
Spider-Man: Far from home	Jul 2, 2019	SM	$160M	$1.132B	90%

(CA = Captain America; IM = Iron Man; BP = Black Panther; SW = Scarlet Witch; AM = Ant-Man; SM = Spider Man; DS = Doctor Strange; T = Thor; H = Hulk)

1. Suppose you want to train a predictor of the form

$$\text{Rating} = \theta_0 + \theta_1 \cdot [\text{number of heroes}] + \theta_2 \cdot [\text{budget}]$$

Write down the feature representations of the first four datapoints in the space below (1 mark):

$$\text{Rating} = \begin{bmatrix} 1 & 6 & 250 \\ 1 & 1 & 165 \\ 1 & 2 & 175 \\ 1 & 4 & 180 \end{bmatrix} \times \theta$$

2. Suppose you wanted to use the 'lists of heroes' and 'release dates' as features to predict the rating. Describe feature representations you might use for each, and write down your representations for the first two movies (2 marks):

A: list: binary encoding [δ(CA in movie), δ(IM in movie)...]
date: one-hot for year: [δ(is 2016), δ(is 2017)...]
movie1: [1 1 1 1 1 1 1 0 0 0 0 0 1 0 0 0]
2: [1 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0]

3. You train two predictors to predict the rating, that yield the following results:

Predictor 1: $\text{Rating} = 90 + 1.1 \cdot [\text{number of heroes}] - 2.8 \times 10^{-11} \cdot [\text{budget}]$

Predictor 2: $\text{Rating} = 90 + 1.3 \times 10^{-8} \cdot [\text{budget}]$

Give a potential explanation as to why the coefficient associated with 'budget' is much larger in the second predictor compared to the first (1 mark):

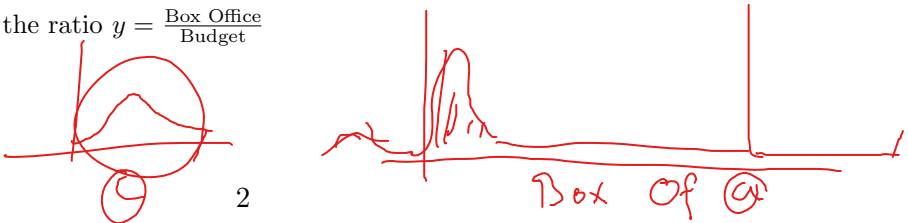
A: #heroes + budget are correlated, but #heroes is a better predictor. So when we include # heroes, budget is less informative

4. **(Critical thinking)** There are several ways we might try to predict whether a movie would be *profitable*. Three possible schemes to encode a label (y) for this task might include:

   (a) train a regressor to predict $y =$ Box Office, and compare the prediction to the Budget

   (b) train a regressor to predict $y =$ (Box Office − Budget) directly

   (c) train a regressor to predict the ratio $y = \frac{\text{Box Office}}{\text{Budget}}$



Assuming in each case we are training a regressor that minimizes the MSE with respect to the label y, which of these schemes do you think would be most effective and why (2 marks)?

A: (a) and (b) have large outliers: MSE may not work well. (c) should not have significant outliers, so is more suitable w/ MSE

Section 2: Classification and Diagnostics (11 marks)

Diagnose **one** potential problem with each of the following experimental pipelines (there could be more than one problem, but it is sufficient to identify a single issue), and suggest a potential correction:

5. You train a binary classifier based on words in a document to distinguish positive versus negative sentiment. You use a 1,000 word dictionary (i.e., 1,001 features including the offset). You collect 2,000 samples, and withhold half for testing. Around half of the labels are positive. Your method has 98% accuracy on the training set but is no better than random on the test set (2 marks).

Problem:	Overfitting
Solution:	increase regularization strength λ

6. Using a large dataset, you train a content filter (a classifier) to detect R-rated (~~i.e.~~, adult) novels among a corpus of 100,000 books, based on their descriptions. Your classifier has approximately 98% accuracy on both the training and test sets, but fails to identify any R-rated novels (2 marks).

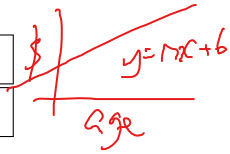
Problem:	label imbalance → trivial classifier
Solution:	use a balanced classifier, and evaluate BER

7. You train a regressor of the form

$$\text{income} = \theta_0 \cdot [\text{age}] + \theta_1 \cdot [\text{has college education}] + \theta_2 \cdot [\text{works in CS}],$$

using a dataset with 10,000 income and demographic measurements. However your predictor has $R^2 < 0$ on both the training *and* the test set (2 marks).

Problem:	We forgot offset term $\theta_0 \times 1$
Solution:	include offset



Design an appropriate measure of "success" for each of the following situations. Your measure could be a known metric (accuracy, BER, etc.), a combination of metrics, or a new metric that you design specifically for the task.

8. You want to filter spam e-mails (i.e., build a classifier that identifies spam). It is okay to let a few spam e-mails through the filter, but the number of non-spam e-mails mistakenly filtered should be close to zero (1 mark).

A: very low FP, somewhat low FN. "weighted" BER ϵ(FPR) + (1-ϵ)FNR, ϵ close to 1

9. You want to train a *regressor* to predict sentiment scores (e.g. ratings). However your dataset consists of 95% female users and 5% male users, and you want your regressor to perform about equally well for both groups (1 mark).

A: "balanced" MSE: (MSE for female) + (MSE for male)

10. You want to build a search engine to find songs based on partially-remembered lyrics (i.e., a user enters some lyrics in the search bar, and you return a ranked list of results via a UI). You know there is exactly one relevant result for each query (i.e., the user is searching for one specific song) (1 mark).

A: UI returns K results, optimize prec@K

11. **(Critical thinking)** We introduced logistic regression as a means of training regressors on binary data, by mapping binary labels ($y \in \{0, 1\}$) to continuous values (in the range $[0, 1]$) by using a predictor of the form $y \simeq \sigma(X \cdot \theta)$. A more crude solution might consist of using ordinary regression (i.e., $y \simeq X \cdot \theta$) directly on the binary labels, and minimizing the Mean Squared Error. (In either case, the final classification is made by checking whether the output is > 0.5, i.e., $\sigma(X \cdot \theta) > 0.5$ for Logistic Regression,[1] or $X \cdot \theta > 0.5$ for our trivial version). Comment on why the MSE might be a poor choice of error measurement here, and why a classifier trained in this way would probably perform poorly (2 marks):

A: if we predict 4 (positive) when the label is 1 (positive), we are penalized more than predicting 0 (negative) when the label is 1 (positive)

Clustering / Communities (3 marks)

Suppose you have a dataset from a social restaurant review network (such as *Yelp*) which contains both home addresses of different users, as well as their social networks. That is, you have 2-d data representing their latitude and longitude coordinates, as well as an adjacency matrix representing users' social connections.

12. Among the following algorithms, which of them would potentially be good choices to build feature representations? Assume your goal is to predict which restaurant a user will visit next. For any you are unsure about, provide a brief explanation (3 marks):

    (a) PCA / (b) K-means / (c) Hierarchical Clustering / (d) Graph Cuts / (e) Clique Percolation / (f) Connected Components

Useful:	K-means / Hierarchical – addresses are clustered clique perc – networks are "cliquey"
Not useful:	PCA – variance about equal in all dirs Graph cuts – network is not "adversarial" Connect. comp – will only be 1 giant component

[1] σ is the sigmoid function, $\sigma(x) = \frac{1}{1+e^{-x}}$.

Recommender Systems (6 marks)

13. Suppose a user listens to a sequence of songs on an online streaming service. The only feedback they can provide is 'thumbs up'/'thumbs down,' (which can be 1, -1, or missing) as well as implicit feedback in the form of finishing or skipping a track. Each time a track finishes (or is skipped), you must select a new track that you expect the user to like. E.g. sequences for two users might look like:

User 1 song sequence	Track completed?	Thumbs up/down?
Two Minutes to Midnight	0	-1
Aces High	0	-1
El Dorado	1	?
Infinite Dreams	1	1
Flight of Icarus	1	?

User 2 sequence	Completed?	up/down?
American Girl	1	1
El Dorado	0	?
Drops of Jupiter	0	-1
Highway Don't Care	1	?

Describe what algorithms you would use to select the next song, and what comparisons you would make (e.g. if using a set similarity metric, what sets would be used as inputs?) (3 marks):

A: U_j = set of users who liked song j;

$Sim(i,j) = Jaccard(U_i, U_j)$

① user u likes or plays to end: song k

next song = $\arg\max_j \sum_{k' \text{ user liked}} Sim(j,k') + \lambda Sim(j,k)$

② u dislikes k $\arg\max_j \sum_{k' \text{ user liked}} Sim(j,k') - \lambda Sim(j,k)$

14. **(Design thinking)** Suppose you want to build a system to recommend running routes to users based on historical data about their exercises (e.g. GPS and heartrate data extracted from a smartwatch, and other metadata). Describe what features you would use, what algorithms you would select, how you would measure performance (etc.) in order to build a recommendation pipeline from this data (3 marks).

A: Task: build a regressor to estimate av. HR for a GPS route

features: - length
- Σ in elevation
- gH, weather,

user features: age, gender, av. HR from previous route

Test time: user specifies location and target HR, recommend nearby route closest to that HR

R^2:

$$1 - \frac{MSE(f)}{Var(y)}$$

Precision:

$$\frac{|\{\text{relevant documents}\} \cap \{\text{retrieved documents}\}|}{|\{\text{retrieved documents}\}|}$$

Recall:

$$\frac{|\{\text{relevant documents}\} \cap \{\text{retrieved documents}\}|}{|\{\text{relevant documents}\}|}$$

Balanced Error Rate (BER):

$$\frac{1}{2}(\text{False Positive Rate} + \text{False Negative Rate})$$

F-score:

$$2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}$$

Jaccard similarity:

$$\text{Sim}(A, B) = \frac{|A \cap B|}{|A \cup B|}$$

Cosine similarity:

$$\text{Sim}(A, B) = \frac{A \cdot B}{\|A\|\|B\|}$$

Algorithm 1 K-means

Initialize every cluster to contain a random set of points
while cluster assignments change between iterations **do**
  Assign each X_i to its nearest centroid
  Update each centroid to be the mean of points assigned to it

Algorithm 2 Clique percolation with parameter k

Initially, all k-cliques in the graph are communities
while there are two communities that have a $(k-1)$-clique in common **do**
  merge both communities into a single community

Algorithm 3 Ratio cut

Choose communities $c \in C$ that minimize $\frac{1}{2}\sum_{c \in C} \frac{\overbrace{cut(c, \bar{c})}^{\text{edges in cut}}}{\underbrace{|c|}_{\text{size of community}}}$

Algorithm 4 Hierarchical clustering

Initially, every point is assigned to its own cluster
while there is more than one cluster **do**
  Compute the center of each cluster
  Combine the two clusters with the nearest centers

Algorithm 5 K-means

Initialize every cluster to contain a random set of points
while cluster assignments change between iterations **do**
  Assign each X_i to its nearest centroid
  Update each centroid to be the mean of points assigned to it

Please write any addition answers/corrections/comments on the front page.