

HOMWORK 9: LEARNING PARADIGMS

10-301/10-601 Introduction to Machine Learning (Spring 2023)

<http://www.cs.cmu.edu/~mgormley/courses/10601/>

OUT: 2023-04-21

DUE: 2023-04-27

TAs: Abhi, Ads, Hang, Markov, Neural, Poorvi, Tara

This is the final homework assignment. This assignment covers **Ensemble Methods**, ***k*-Means**, **PCA**, and **Recommender Systems**.

START HERE: Instructions

- **Collaboration Policy:** Please read the collaboration policy here: <http://www.cs.cmu.edu/~mgormley/courses/10601/syllabus.html>
- **Late Submission Policy:** See the late submission policy here: <http://www.cs.cmu.edu/~mgormley/courses/10601/syllabus.html>
- **Submitting your work:** You will use Gradescope to submit answers to all questions. Please follow instructions at the end of this PDF to correctly submit all your code to Gradescope.
 - **Written:** For written problems such as short answer, multiple choice, derivations, proofs, or plots, please use the provided template. Submissions can be handwritten onto the template, but should be labeled and clearly legible. If your writing is not legible, you will not be awarded marks. Alternatively, submissions can be written in \LaTeX . Each derivation/proof should be completed in the boxes provided. You are responsible for ensuring that your submission contains exactly the same number of pages and the same alignment as our PDF template. If you do not follow the template, your assignment may not be graded correctly by our AI assisted grader and there will be a **2% penalty** (e.g., if the homework is out of 100 points, 5 points will be deducted from your final score).

Instructions for Specific Problem Types

For “Select One” questions, please fill in the appropriate bubble completely:

Select One: Who taught this course?

- Matt Gormley
- Marie Curie
- Noam Chomsky

If you need to change your answer, you may cross out the previous answer and bubble in the new answer:

Select One: Who taught this course?

- Henry Chai
- Marie Curie
- Noam Chomsky

For “Select all that apply” questions, please fill in all appropriate squares completely:

Select all that apply: Which are scientists?

- Stephen Hawking
- Albert Einstein
- Isaac Newton
- I don't know

Again, if you need to change your answer, you may cross out the previous answer(s) and bubble in the new answer(s):

Select all that apply: Which are scientists?

- Stephen Hawking
- Albert Einstein
- Isaac Newton
- I don't know

For questions where you must fill in a blank, please make sure your final answer is fully included in the given space. You may cross out answers or parts of answers, but the final answer must still be within the given space.

Fill in the blank: What is the course number?

10-601

10-~~6~~301

Written Questions (43 points)

1 \LaTeX Bonus Point and Template Alignment (1 points)

1. (1 point) **Select one:** Did you use \LaTeX for the entire written portion of this homework?

Yes

No

2. (0 points) **Select one:** I have ensured that my final submission is aligned with the original template given to me in the handout file and that I haven't deleted or resized any items or made any other modifications which will result in a misaligned template. I understand that incorrectly responding yes to this question will result in a penalty equivalent to 2% of the points on this assignment.

Note: Failing to answer this question will not exempt you from the 2% misalignment penalty.

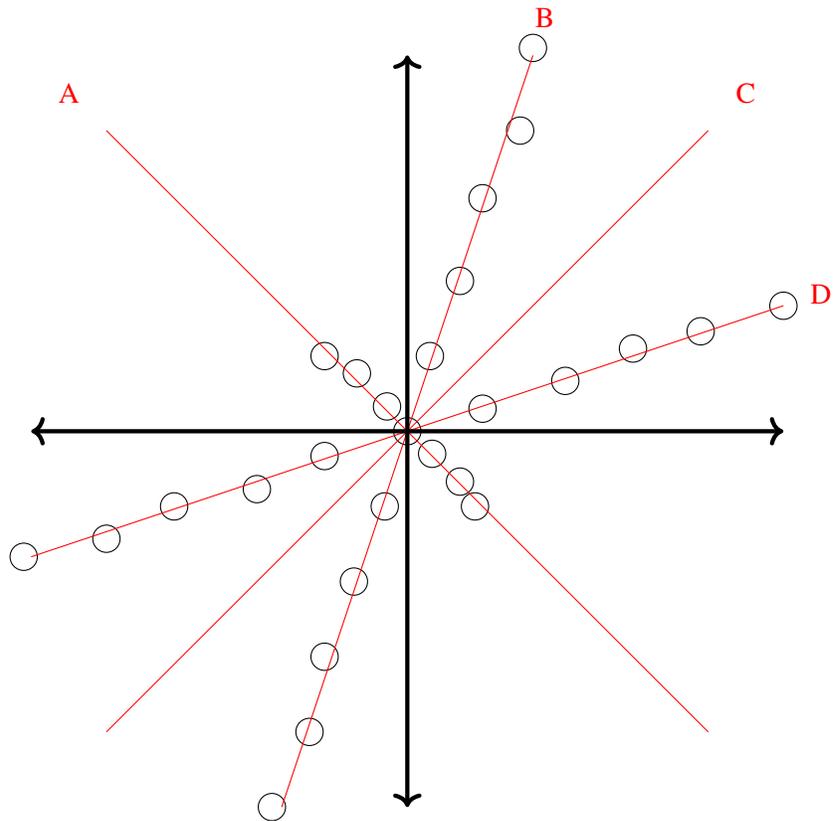
Yes

2 PCA (8 points)

Some PCA Theory

- (1 point) **Select one:** Assume we apply PCA to a matrix $\mathbf{X} \in \mathbb{R}^{n \times 2}$ and obtain two sets of PCA feature scores, $\mathbf{z}_1, \mathbf{z}_2 \in \mathbb{R}^n$, where \mathbf{z}_1 corresponds to the first principal component and \mathbf{z}_2 corresponds to the second principal component. Which is more common in the training data:
 - a point with large feature values in \mathbf{z}_1 and small feature values in \mathbf{z}_2
 - a point with small feature values in \mathbf{z}_1 and large feature values in \mathbf{z}_2
- (2 points) For the data set shown below, list the principal components from first to last.

Your Answer



- (2 points) **Select all that apply:** To get the principal components of the features, we calculate the eigenvectors of the covariance matrix, which are orthogonal, along with their corresponding eigenvalues. Which of the following are **consequences of the principal components being orthogonal to each other**?
 - The variance of the data is maximized.
 - The reconstruction error is minimized.
 - We can attribute certain variations in the data to unique principal components.
 - It ensures that our lower-dimensional data will be linearly separable.
 - None of the above

4. (1 point) **Select one:** If we wanted to perform dimensionality reduction to have just two features and train a classifier on them, we could represent our data by EITHER (a) picking any two features from the dataset, OR (b) using the first 2 principal components we obtained from PCA. Which one should we prefer and why?
- We prefer (a), because it ensures randomness in the selection and have a better chance of representing the data well.
 - We prefer (a), because PCA introduces artificial bias and does not reflect the original features.
 - We prefer (b), because PCA usually preserves higher variance than two original features.
 - We prefer (b), because PCA ensures that variance is evenly distributed across the features.

PCA in Practice

For this section, refer to the PCA demo linked [here](#). In this demonstration, we have performed PCA for you on a [simple four-feature dataset](#). The questions below have also been added to the colab notebook linked for ease of access. Run the code in the notebook, then answer the questions based on the results.

5. (1 point) **Select one:** Do you see any special relationships between any of the features? In particular, take a look at the `petal_length` feature. How would you describe its association with each of the **other features**? Select the correct statement *with appropriate justification*.
- The features are highly correlated: we observe linearly proportional relationships where increases in `petal_length` often correspond to increases in another feature
 - The features are highly correlated: we observe that the color classes can be separated with decision boundaries along the `petal_length` axis.
 - The features are uncorrelated: we observe random noise as if the features were generated from independent distributions
 - The features are uncorrelated: we observe the “default $y = x$ ” relationship between features
6. (1 point) If we wanted to find k principal components such that we preserve **at least** 95% of the variance in the data, what would be the value of k ? Hint: it is helpful here to look at the cumulative variance in the first k components, which we have calculated for you.

k

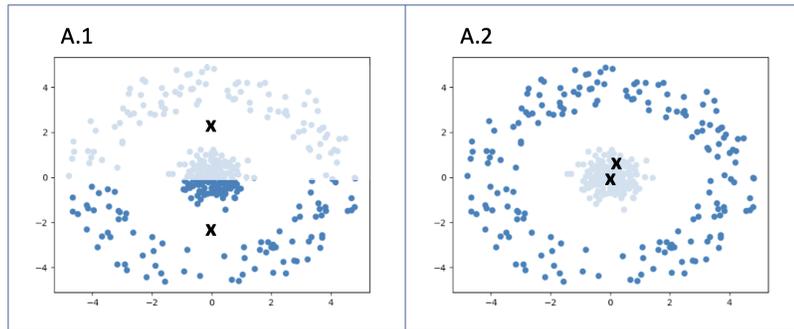
3 k -Means (12 points)

1. Consider the 2 datasets A and B. Each dataset is classified into k clusters, with centers marked X and cluster membership represented by different colors in the figure. For each dataset, exactly one clustering was generated by k -means with Euclidean distance. Select the image with clusters generated by k -means.

(a) (1 point) Dataset A

Select one:

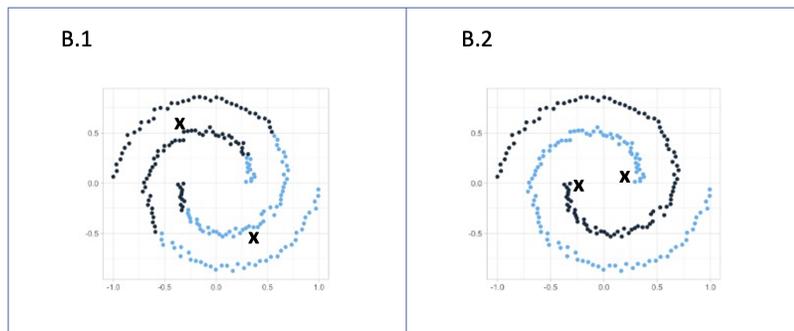
- A.1
 A.2



(b) (1 point) Dataset B

Select one:

- B.1
 B.2



2. Consider a dataset \mathcal{D} with 5 points as shown below. Perform a k -means clustering on this dataset with $k = 2$ using the Euclidean distance as the distance function. Remember that in the k -means algorithm, one iteration consists of following two steps: first, we assign each data point to its nearest cluster center; second, we recompute each center as the average of the data points assigned to it. Initially, the 2 cluster centers are chosen randomly as $\mu_0 = (5.3, 3.5)$, $\mu_1 = (5.1, 4.2)$. Parts (a) through (d) refer only to the first iteration of k -means clustering performed on \mathcal{D} .

$$\mathcal{D} = \begin{bmatrix} 5.5 & 3.1 \\ 5.1 & 4.8 \\ 6.6 & 3.0 \\ 5.5 & 4.6 \\ 6.8 & 3.8 \end{bmatrix}$$

(a) (1 point) **Select one:** Which of the following points will be the new center for cluster 0?

- (5.7 , 4.1)
- (5.6 , 4.8)
- (6.3 , 3.3)
- (6.7 , 3.4)

(b) (1 point) **Select one:** Which of the following points will be the new center for cluster 1?

- (6.1 , 3.8)
- (5.5 , 4.6)
- (5.4 , 4.7)
- (5.3 , 4.7)

(c) (1 point) How many points will belong to cluster 0, using the new centers?

Answer

(d) (1 point) How many points will belong to cluster 1, using the new centers?

Answer

3. Recall that in k -means clustering we attempt to find k cluster centers $\mathbf{c}_1, \dots, \mathbf{c}_k$ such that the total distance between each point and the nearest cluster center is minimized. We thus solve

$$\operatorname{argmin}_{\mathbf{c}_1, \dots, \mathbf{c}_k} \sum_{i=1}^N \min_{j \in \{1, \dots, k\}} \|\mathbf{x}^{(i)} - \mathbf{c}_j\|_2^2$$

where n is the number of data points. Instead of holding the number of clusters k fixed, your friend John tries to also minimize the objective over k , solving

$$\operatorname{argmin}_k \operatorname{argmin}_{\mathbf{c}_1, \dots, \mathbf{c}_k} \sum_{i=1}^N \min_{j \in \{1, \dots, k\}} \|\mathbf{x}^{(i)} - \mathbf{c}_j\|_2^2$$

You found this idea to be a bad one.

(a) (1 point) What is the minimum possible value of the objective function when minimizing over k ?

Answer

- (b) (1 point) What is a value of k for which we achieve the minimum possible value of the objective function when $N = 100$?

Answer

4. Consider the following brute-force algorithm for minimizing the k -means objective: Iterate through each possible assignment of the points to k clusters, $\mathbf{z} = [z^{(1)}, \dots, z^{(N)}]$. For each assignment $\mathbf{z} \in \{1, \dots, k\}^N$, you evaluate the following objective function:

$$J(\mathbf{z}) = \operatorname{argmin}_{\mathbf{c}_1, \dots, \mathbf{c}_k} \sum_{i=1}^N \|\mathbf{x}^{(i)} - \mathbf{c}_{z^{(i)}}\|_2^2$$

At the end, you pick the assignment \mathbf{z} that had lowest $J(\mathbf{z})$.

- (a) (1 point) Suppose we have N points and k clusters. For how many possible assignments \mathbf{z} does the brute force algorithm have to evaluate $J(\mathbf{z})$?

Answer

- (b) (1 point) Suppose $N = 1000$, $k = 10$, and it takes us 0.01 seconds to evaluate $J(\mathbf{z})$ for a single assignment \mathbf{z} . How many seconds will the brute force algorithm take to check all assignments?

Answer

5. Initializing the centers has a big impact on the performance of the k -means clustering algorithm. Usually, we randomly initialize k cluster centers. However, there are other methods, namely, furthest point initialization and k -means++ initialization.

- (a) (1 point) **Select one:** Clustering at convergence generated by furthest point initialization is sensitive to outliers. Which of the following statements is correct about this phenomenon?

- Although outliers will not be selected in the first several iterations, they will temporarily be chosen as centers during training.
- Outliers will slow convergence, but will never be centers at convergence time.
- Outliers are likely to be selected as centers in the first few iterations.

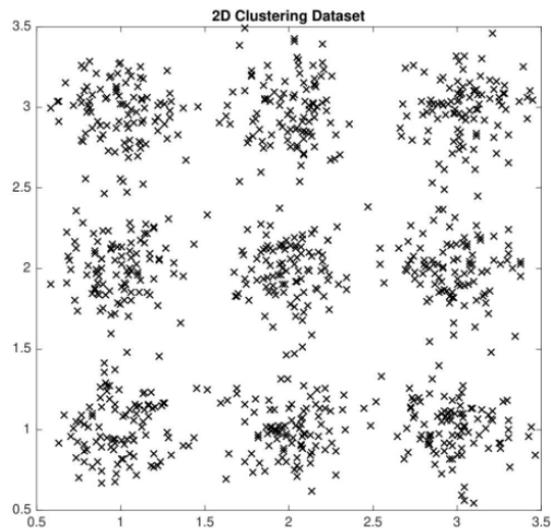


Figure 1: 2D Dataset

- (b) (1 point) **Select one:** Using the dataset in Figure 1 above, compared to random initialization, using k -means++ initialization is _____.
- more likely to choose one sample from each cluster because centers are chosen with probability proportional to squared distance from existing centers.
 - less likely to choose one sample from each cluster because the formula does not account for the number of clusters which may be found and thus won't be calibrated to correctly choose one point from each cluster.
 - equally likely to choose one sample from each cluster because as the number of points grows large, k -means++ asymptotically approaches random initialization.

4 Ensemble Methods (14 points)

- (1 point) **True or False:** In a random forest, it is generally better for the trees to be highly correlated, as this reduces variability.
 - True
 - False
- (2 points) **Select all that apply:** Which of the following is true about OOB error?
 - OOB error is calculated on a held-out dataset separate from the dataset used to generate bootstrap samples
 - OOB error is the aggregated value of the errors of subsets of the ensemble on samples those subsets were not trained on
 - Cross-validation error is the same as OOB error
 - OOB error is a valid method of estimating true error
 - None of the above
- (2 points) **Select all that apply:** Which of the following are hyperparameters that can be tuned in a random forest?
 - Number of trees trained
 - Number of points used to train each decision tree
 - Size of feature subsets used to train each decision tree
 - Which features are used for splits in each decision tree
 - None of the above
- In this question, we will now derive an error bound for random forests in the case of **binary classification**. Given a random forest of B trees $\{h_i(x)\}_{i=1}^B$ and a sample (x, y) drawn from some data distribution \mathcal{D} , define the classification margin as:

$$m(x, y) = \frac{1}{B} \left(\sum_{i=1}^B \mathbb{I}[h_i(x) = y] - \sum_{i=1}^B \mathbb{I}[h_i(x) \neq y] \right)$$

In words, the margin $m(x, y)$ is the difference between the average vote for the correct label and the average vote for the incorrect label.

- (1 point) **Fill in the blank:** For any example (x, y) , the example is classified incorrectly if and only if $m(x, y) \leq \underline{\hspace{1cm}}$. Assume majority vote ties are classified incorrectly.

Answer

- (b) (2 points) Observe that $P_{(x,y)\sim\mathcal{D}}(m(x,y) \leq c)$, where c is your answer to part (a), corresponds to the generalization error of the ensemble. Additionally, for an ensemble $\{h_i(x)\}_{i=1}^B$, define the strength of the ensemble as $s = \mathbb{E}_{(x,y)\sim\mathcal{D}}[m(x,y)]$.

Assume $s > 0$. Write a bound for the generalization error in the form $P(m(x,y) < c) \leq d$, where d is an expression in terms of s and $\text{Var}(m(x,y))$.

Hint: You should use Chebyshev's inequality, which states that for any random variable X with finite expectation and variance and any constant $a > 0$, we have

$$P(|X - \mathbb{E}[X]| \geq a) \leq \frac{\text{Var}(X)}{a^2}$$

Generalization error bound

- (c) (1 point) **Select one:** Through some additional manipulation, it is possible to show that $\text{Var}(m(x,y)) \leq \bar{\rho}(1 - s^2)$, where $\bar{\rho}$ is the mean correlation between trees in the ensemble. Substitute this into your bound from part (b). Which of the following describes how the error bound is affected by s and $\bar{\rho}$?
- The error bound gets smaller as $\bar{\rho}$ increases and s increases.
 - The error bound gets smaller as $\bar{\rho}$ increases and s decreases.
 - The error bound gets smaller as $\bar{\rho}$ decreases and s increases.
 - The error bound gets smaller as $\bar{\rho}$ decreases and s decreases.

5. (1 point) **True or False:** Consider some training point $(x^{(i)}, y^{(i)})$ to the AdaBoost algorithm. If for all t , the weak learner h_t learned during training at time t correctly classifies $h_t(x^{(i)}) = y^{(i)}$, there will eventually be a finite time t such that the weight assigned to $x^{(i)}$ in the training distribution \mathcal{D}_t reaches exactly 0.

True

False

6. (1 point) **True or False:** If the ensemble learned by AdaBoost reaches perfect training accuracy, all weak learners created in subsequent iterations will be identical (i.e., they will produce the same output on any input). Assume we are using deterministically selected weak learners.

True

False

7. Assume we use a deterministic training procedure for weak learners. Suppose for some iteration t' of AdaBoost we find that the weak classifier learned by the algorithm at time t' has error $\epsilon_{t'} = 0.5$ of the weak learner $h_{t'}$ on the training distribution weighted by $\mathcal{D}_{t'}$.

(a) (1 point) What weight $\alpha_{t'}$ will AdaBoost assign to the classifier $h_{t'}$ from above?

$\epsilon_{t'}$

(b) (1 point) **Select all that apply:** In which of the following cases will $\mathcal{D}_{t'+1}(i) > \mathcal{D}_{t'}(i)$ (in other words, in which of the following cases will the weight of training sample $(x^{(i)}, y^{(i)})$ *strictly* increase from time step t' to $t' + 1$)?

$h_{t'}(x^{(i)}) = y^{(i)}$ ($h_{t'}$ classifies $x^{(i)}$ correctly)

$h_{t'}(x^{(i)}) \neq y^{(i)}$ ($h_{t'}$ classifies $x^{(i)}$ incorrectly)

None of the above.

(c) (1 point) **Select all that apply:** Which of the following are true about the next iteration of the AdaBoost algorithm?

The errors $\epsilon_{t'+1}$ and $\epsilon_{t'}$ are equivalent

The weak learners $h_{t'+1}$ and $h_{t'}$ will be equivalent (i.e., they will have the same output for every input)

None of the above

5 Recommender Systems (8 points)

- (2 points) **Select all that apply:** In which of the following situations will a collaborative filtering system be a more appropriate learning algorithm than a linear or logistic regression model?
 - You manage an online bookstore, and you have book ratings and sales data from many users. For each user, you want to recommend other books she will enjoy, based on her own ratings and the ratings of other users.
 - You manage an online bookstore, and you have book ratings and sales data from many users. You want to learn to predict the expected sales volume (number of books sold) as a function of the average rating of a book.
 - You run an online news aggregator, and for every user, you know some subset of articles that the user likes and some different subset that the user dislikes. You want to use this to find other articles that a given user likes.
 - You've written a piece of software that downloads news articles from many news websites. In your system, you also keep track of which articles you personally like and which ones you dislike, and the system also stores away features of these articles (e.g., word counts, name of author). Using this information, you want to build a system to try to find additional new articles that you personally will like.
 - None of the above
- (2 points) **Select all that apply:** What is the basic intuition behind matrix factorization?
 - That content filtering and collaborative filtering are just two different factorizations of the same rating matrix.
 - That factoring user and item matrices can partition the users and items into clusters that can be treated identically, which can reduce computation when making recommendations by retaining only representative users or items in each cluster.
 - That computing a user-user or item-item correlation is more efficient when first factoring matrices, even when including the cost of factoring matrices.
 - That users and items can be well described in a shared low dimensional space that can be computed from the rating matrices.
 - None of the above
- Neural the Narwhal decides to set up a friend-recommendation system for all the students in 10-301/601, of which there are $N = 10,301,601$. Ideally, Neural would store the full $N \times N$ matrix M , where M_{ij} is 1 if student i and j are friends and 0 if student i and j are nemeses, or null if students i and j have never met. Assume that these are the only possible relationships between 2 people and that all relationships are symmetric (so it cannot be the case that student i thinks student j is their friend while j thinks i is their nemesis). Unfortunately, storing M in its entirety would take over 10 TB of storage, which Neural cannot afford on a TA salary. Neural instead uses the following procedure to approximate M as UU^T for some low rank $N \times d$ matrix U , where each row of U corresponds to a student.

```

1: Given learning rate  $\eta$ , ground truth relationships  $M$ 
2: Randomly initialize user embedding matrix  $U \in \mathbb{R}^{N \times d}$ 
3: while not converged do
4:   Sample  $i \sim \text{Uniform}(1, N), j \sim \text{Uniform}(1, N)$ 
5:    $\hat{M}_{ij} \leftarrow \sigma(\vec{u}_i^T \vec{u}_j)$  //  $\sigma$  is sigmoid function
6:    $\mathcal{L}(\vec{u}_i, \vec{u}_j) \leftarrow -M_{ij} \log(\hat{M}_{ij}) - (1 - M_{ij}) \log(1 - \hat{M}_{ij})$  // Compute logistic loss
7:    $\vec{g}_i \leftarrow \nabla_{\vec{u}_i} \mathcal{L}(\vec{u}_i, \vec{u}_j)$  // Compute and perform gradient updates
8:    $\vec{g}_j \leftarrow \nabla_{\vec{u}_j} \mathcal{L}(\vec{u}_i, \vec{u}_j)$ 
9:    $\vec{u}_i \leftarrow \vec{u}_i - \eta \cdot \vec{g}_i$ 
10:   $\vec{u}_j \leftarrow \vec{u}_j - \eta \cdot \vec{g}_j$ 

```

(a) (1 point) Explain why line 4 of the algorithm is incorrect.

Fixed line of code

(b) (1 point) **Select one:** Based on the loss function given on line 6, derive an expression for $\nabla_{\vec{u}_j} \mathcal{L}(\vec{u}_i, \vec{u}_j)$ in terms of \vec{u}_i , \vec{u}_j , and M_{ij} . Note that σ denotes the sigmoid function and \log is natural log.

- $\vec{u}_i M_{ij} \sigma(\vec{u}_i^T \vec{u}_j)$
- $\vec{u}_i (M_{ij} + \sigma(\vec{u}_i^T \vec{u}_j))$
- $\vec{u}_i (-M_{ij} + \sigma(\vec{u}_i^T \vec{u}_j))$
- None of the above.

(c) (2 points) **Select all that apply:** Why is it appropriate here to factorize M as the product of a single matrix by itself UU^T rather than as the product of two distinct matrices VW^T ?

- UU^T enforces that our approximation is symmetric, while VW^T is not necessarily symmetric.
- SGD is guaranteed to converge faster and to a better optimum for U because we have fewer parameters to learn.
- We can parallelize training for U , whereas we could not parallelize training for V and W .
- We wish to model relationships between objects of a single type, not between objects of two different types.
- None of the above.

6 Collaboration Questions

After you have completed all other components of this assignment, report your answers to these questions regarding the collaboration policy. Details of the policy can be found [here](#).

1. Did you receive any help whatsoever from anyone in solving this assignment? If so, include full details.
2. Did you give any help whatsoever to anyone in solving this assignment? If so, include full details.
3. Did you find or come across code that implements any part of this assignment? If so, include full details.

Your Answer