CSE 546 Homework 4A

Jake Gonzales

December 10, 2023

Collaborators: n/a

1 Conceptual Questions

A1.

(a) **True**. Here, we have $k = \text{rank}(X)$ which is the number of nonzero eigenvalues and at max is equal to d. We know that a linear combination of these vectors would fully represent the data matrix $X \in \mathbb{R}^{n \times d}$. So if we project our data onto a k-dimensional subspace using PCA, our projection will have no information loss.

(b) **False**. The columns of V are equal to the eigenvectors of $X^T X$.

(c) **False.** Choosing the k to minimize the k-means objective doesn't necessarily provide a good way to find meaningful clusters, it would be problem-dependent.

(d) **False**. The singular value decomposition is not unique.

(e) **False**. The rank of a square matrix equals the number of nonzero eigenvalues **with repetition**.

2 Think before you train

A2.

(a) We are tasked with designing a model to be a disease susceptibility predictor. We are given a rich dataset with personal demographic information (such as age, race, and sex), location information, risk factors, and whether a person has the disease or not. The information is tabular in form, so we can create a data matrix where each row represents an observation and each column represents a different parameter from our dataset. In our dataset we have both quantitative (income, age, etc.) and qualitative (race, sex, etc.) data, so we would need to develop a one hot encoding scheme for the construction of the data matrix. For example for sex we could have male = 1, female =2, or other = N/A. This also brings an issue regarding missing values in our data which we may want to perform matrix completion or data imputation by averaging or rank approximations. After we have constructed our data matrix then we would look to find a predictor y which would perform prediction in a probabilistic manner for susceptibility to the disease. We would split our dataset into training sets, validation sets, and test sets. I would propose developing two models. First we would develop a simple logistic regression model with maximum likelihood estimation. Due to the simplicity of a logistic regression model, we would have faster training, easier interpretation, and explainability of predictions. However, this requires the data to have little multicollinearity and be linearly separable. Therefore, we would develop a more complex neural network model with a certain amount of hidden layers and a softmax output. This allows us to make predictions with nonlinearities present and the softmax layer provides a probabilistic output. We would compare the two models together. We would also perform training, validation for any hyperparameter tuning, and test on the test set. Due to imperfect data and other uncertainties inherent in developing machine learning models, we would take grave precaution to informing someone of a high probability of being susceptible.

(b) One potential shortcoming is missing values in our data which was mentioned above. There could be scenarios where we have people providing answers vastly different for sex or race/ethnicity. There could be several mixtures of races/ethnicity's that could impact predictions. Looking deeper, some people from more privilege backgrounds may have fuller information on risk factors (due to more frequent doctor visits, etc.) and people from other less privileged backgrounds may not have this information so the prediction will be biased. Additionally, on the same note, risk factors can be contributed to nutritional levels, diets, and exercise which can differ based on the background of the person. There are quite a few things left to be considered. It would take some great feature engineering determine the proper ways to mitigate these biases.

(c) As mentioned above there are quite a few real-world implications. Where was the dataset collected from? What populations, demographics, differing income levels, etc. was this dataset collected? Was it evenly spread? Do some people have better or worse healthcare than other people and how do their backgrounds and income play into this? Yes there are lots of differing issues with regrading to how we collect data and use it to develop machine learning models. There certainly can be instances where our models may be unintentionally biased towards a certain group and this would be a disadvantage to the minorities.

3 k-means clustering

A3.

(a) Code submitted.

(b) Below are 10 images representing the cluster centers.

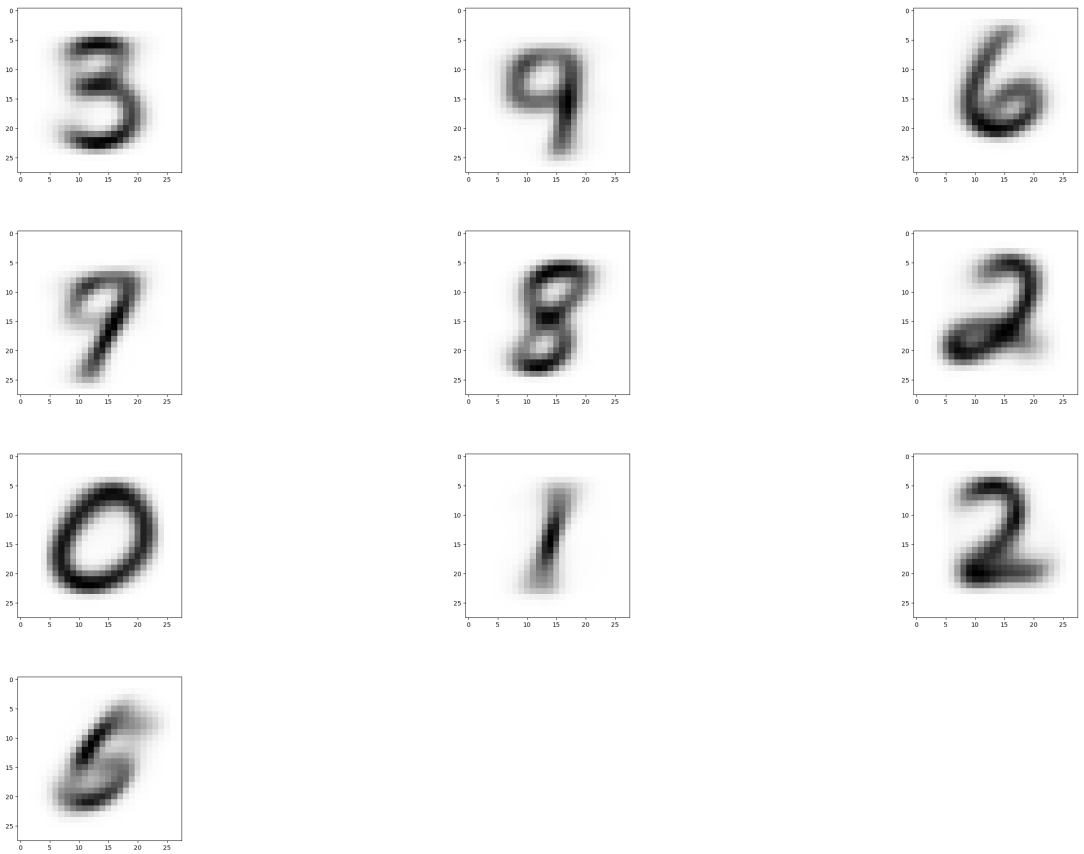

Figure 1: Cluster Centers

4 Matrix Completion and Recommendation System

A4.

(a) Our first estimator pools all the users together and, for each movie, outputs as its predictor the average user rating of that movie. We define a vector $\mu \in \mathbb{R}$ where μ_i is the average rating of the users that rated the i-th movie, which can be represented by the estimator $\widehat{R}$ as a rank-one matrix as the following:

$$\widehat{R} = \mu \mathbf{I}^T = \langle \mu_i, \mathbf{I} \rangle,$$

where we set $\mu_i = 0$ if the movie has not been rated by any users. Using this in implementation gave:

$$\mathcal{E}_{\text{test}}(\widehat{R}) = 3.088.$$

(b) Plot for MSE on training and test set v. d using a rank-d approximation $\widehat{R}^{(d)}$

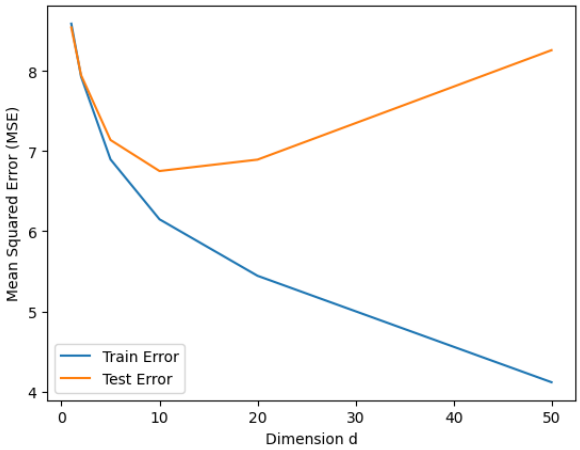

Figure 2: Rank-d Approximation

(c) Plot for MSE on training and test set v. d using alternating minimization.

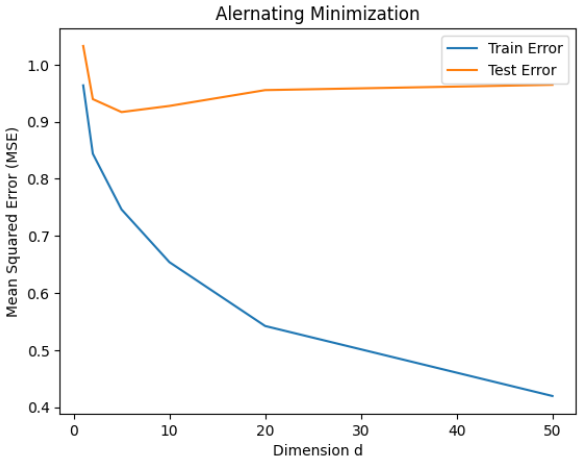

Figure 3: Alternating Minimization

5 Image Classification on CIFAR-10

A5.

(a) **Fully-connected output, 1 fully-connected hidden layer:** for this network our hyperparameter was M, which is the size of the hidden layer. For this we used a stochastic gradient descent optimizer and for learning rates searched over [0.0010, 0.0120, 0.0230, 0.0340, 0.0450, 0.0560, 0.0670, 0.0780, 0.0890, 0.100]. This was altered slightly throughout the parameter search process but the upper and lower bounds are the same. For M we searched over [300, 310, 320, $\cdots$, 520]. Below is a list of the hyperparameter configurations and the accuracy achieved.

lr	M	Acc
0.0700	450	0.50079
0.0900	490	0.51335
0.0560	510	0.50366

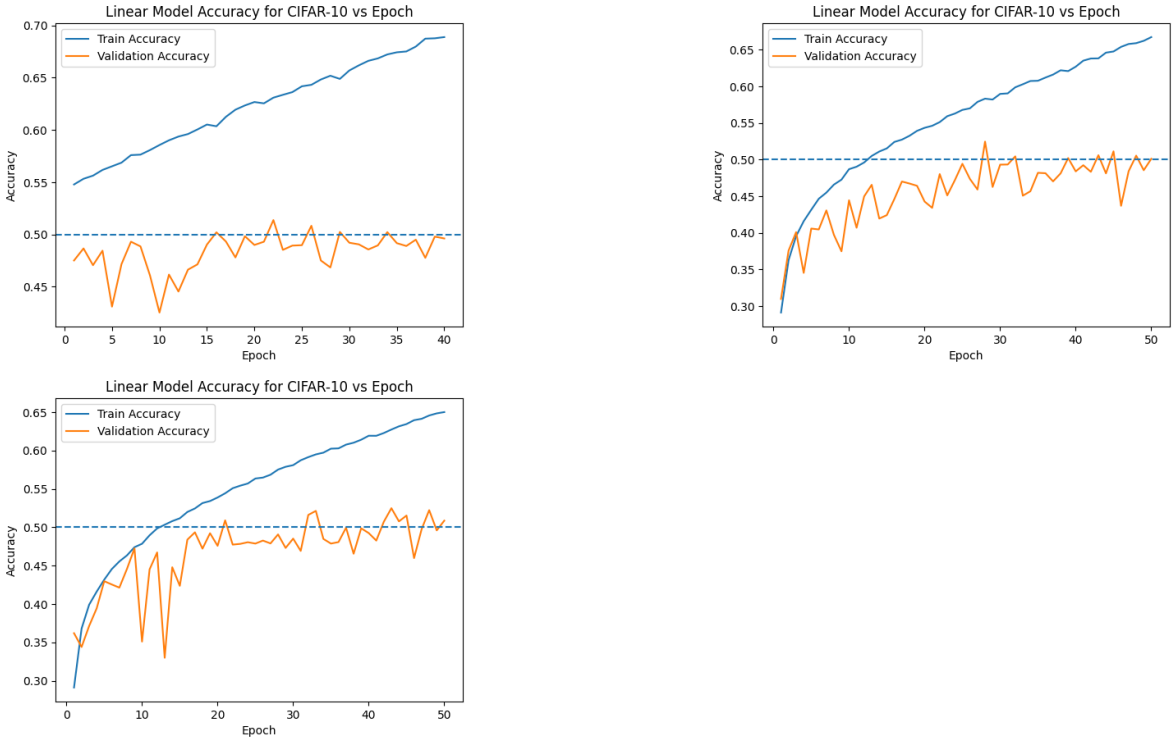

Figure 4: Accuracy v. Epochs for Part A Model

(b) **Convolutional layer with max-pool and fully-connected output:** for this convolutional neural network (CNN) our hyperparameters were M, k, N and the learning rate. I was able to do a parameter search using SGD and within two searches achieved the required model accuracy of 65%. Below are the ranges that were searched over for each hyperparameter:

$$\text{lr: [1e-3, 5e-3, 1e-2, 5e-2, 1e-1]}$$
$$\text{M} = [300, 310, 320, \cdots, 520]$$
$$\text{N} = [4, 5, 6, \cdots, 12]$$
$$\text{k} = [3, 4, 5, \cdots, 8].$$

The hyperparameter configurations and the accuracy achieved are listed below:

lr	M	N	k	Acc
0.1	460	2	3	0.5934
0.01	360	2	5	0.6559
0.05	480	4	5	0.6875

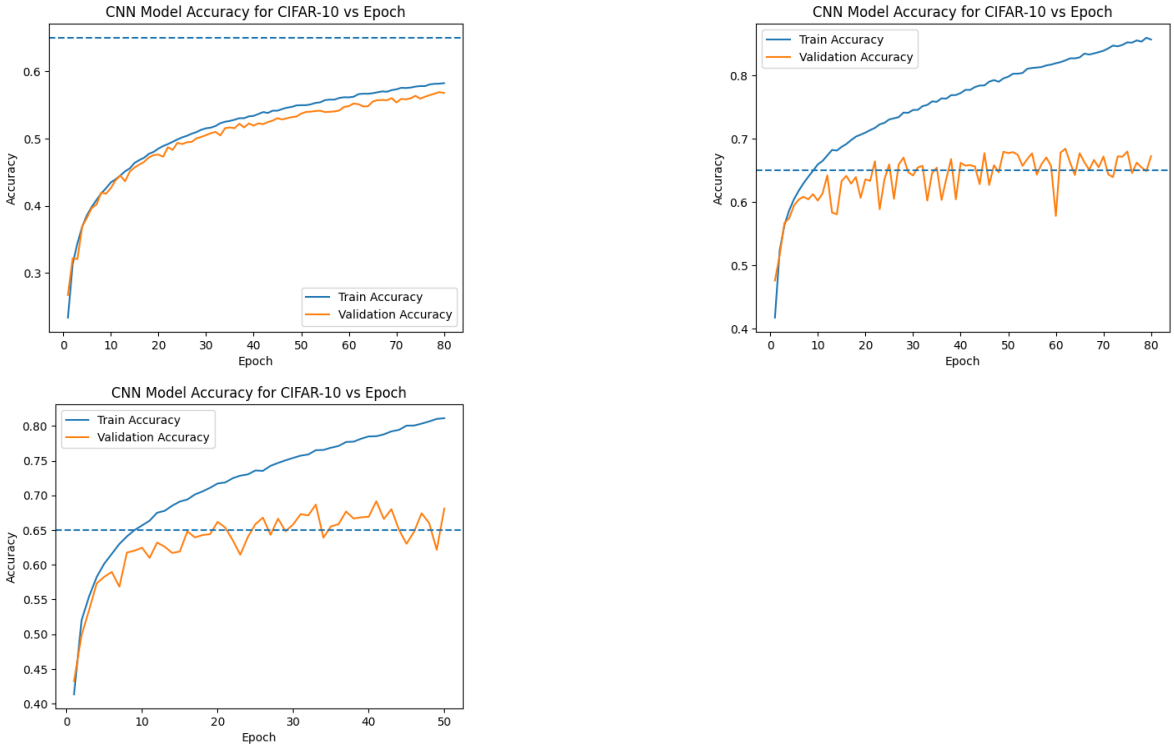

Figure 5: Accuracy v. Epochs for CNN Model

6 Administrative

A6.

(a) If you mean training included probably 80 hours, honestly no idea. Actually coding or working on it probably about 20 hours.