CIS 520, Machine Learning, Fall 2019
Homework 5
Due: Monday, October 27th, 11:59pm
Submit to Gradescope

Matthew Scharf

October 29, 2019

1 Perceptron vs. Winnow

(a) Sparse target vector u, dense feature vectors $\mathbf{x}_t$.

$$||u||_1 = ||u||_2 = k$$

Perceptron:

$$||x_t||_2 \leq d = R_2$$

So,

$$mistakes = \frac{R_2^2||u||_2^2}{\gamma^2} = \frac{d^2k^2}{\gamma^2}$$

Winnow:

$$||x_t||_\infty \leq d = 1 = R_\infty$$

So,

$$mistakes = 2ln(d)\frac{R_2^\infty||u||_1^2}{\gamma^2} = 2ln(d)\frac{k^2}{\gamma^2}$$

Choice: Winnow is a better choice because

$$2ln(d)\frac{k^2}{\gamma^2} \leq \frac{d^2k^2}{\gamma^2}$$

(b) Dense target vector u, sparse feature vectors $\mathbf{x}_t$.

Perceptron:

$$|u||_2 \leq 2\sqrt{d}, \quad ||x_t||_2 \leq k = R_2$$

So,

$$mistakes = \frac{R_2^2||u||_2^2}{\gamma^2} = \frac{k^2 4d}{\gamma^2}$$

Winnow:

$$||u||_1 = d, \quad ||x_t||_\infty \leq 1 = R_\infty$$

So,

$$mistakes = 2ln(d)\frac{R_2^\infty||u||_1^2}{\gamma^2} = 2ln(d)\frac{d^2}{\gamma^2}$$

Choice: Perceptron is a better choice because

$$\frac{k^2 4d}{\gamma^2} \leq 2ln(d)\frac{d^2}{\gamma^2}$$

due to $k << d$.

(c) If your problem has non-negative feature vectors $\mathbf{x}_t \in \mathbb{R}^d_+$, is the Winnow algorithm a meaningful choice? Why or why not?

It is not a meaningful choice. It can only produce a non-negative u. If the feature vectors are also all non-negative then the weights will only ever predict a label of $+1$. This will be useless for classification.

2 Singular Value Decomposition

1. Let $\mathbf{X}$ be a n by p matrix. Show that if $\mathbf{X}$ has a rank p (all its columns are linearly independent), and $n > p$, then using the p-dimensional pseudo-inverse $\mathbf{X}^+ = \mathbf{V}_k\Lambda_k^{-1}\mathbf{U}_k^T$ in $\hat{\mathbf{w}} = \mathbf{X}^+\mathbf{y}$ with $k = p$ solves the least squares problem $\hat{\mathbf{w}} = \arg\min_w(\mathbf{y} - X\mathbf{w})^T(\mathbf{y} - \mathbf{Xw})$.

   First, we want to find w such that $0 = \frac{d}{dw}(y - Xw)^T(y - Xw) = -2X^T(y - Xw)$ by the given identity. So, $X^T y = X^T X w$, then $(X^T X)^{-1}X^T y = w$ because X is full rank so $X^T X$ is invertible.

   Now, by the singular value decomposition theorem, let $X = U_k\Lambda_k V_k^T$.

   So,
   $$\begin{aligned} w &= ((V_k\Lambda_k U_k^T)(U_k\Lambda_k V_k^T))^{-1}X^T y \\ &= (V_k\Lambda_k^2 V_k^T)^{-1}V_k\Lambda_k U_k^T y \\ &= V_k\Lambda_k^{-2}V_k^T V_k\Lambda_k U_k^T y \\ &= V_k\Lambda_k^{-1}U_k^T y \end{aligned}$$

2. Given the eigenvectors of $\mathbf{XX}^T$ as $(\mathbf{u}_1, ..., \mathbf{u}_k)$ and corresponding eigenvalues as $(\lambda_i, ..., \lambda_k)$, give an expression for computing an eigenvector $\mathbf{v}_i$ of $\mathbf{X}^T\mathbf{X}$ in terms of $\mathbf{X}$, $\mathbf{u}_i$, and λ_i.

   We know that $X = U_k\Lambda_k V_k^T$ where the columns of unitary U_k are the eigenvectors of XX^T, the columns of unitary V_k are the eigenvectors of $X^T X$, and the square of diagonal Λ_k is the eigenvalues of both.

   Then, $\Lambda_k^{-1}U_k^T X = V_k^T$.

   So, $X^T U_k\Lambda_k^{-1} = V_k$.

   This means that for $i \in \{1, \ldots, k\}$,
   $$v_i = \frac{X^T u_i}{\lambda_i}$$

3. Let $\mathbf{X}$ be a n by p matrix. Under what conditions (in terms of the relationship between n and p) would the above calculation be an efficient way to find the largest eigenvectors of $\mathbf{X}^T\mathbf{X}$?

   $X^T X$ is pxp and XX^T is nxn so if $p >> n$ then finding the eigenvectors of XX^T (aka U_k) would be much faster and so the above calculation would be an efficient way to find the eigenvectors of $X^T X$ (aka V_k).

3 Principal Component Analysis

3.1 Part 1: Comparing Principal Components

1. Report the eigenvectors and eigenvalues here. This section is auto graded, checking the outputs of your PCA function.

   Eigenvectors: $\begin{bmatrix} -0.707 \\ 0.707 \end{bmatrix}, \begin{bmatrix} -0.707 \\ -0.707 \end{bmatrix}$

   Eigenvalues: 1.653, 0.358

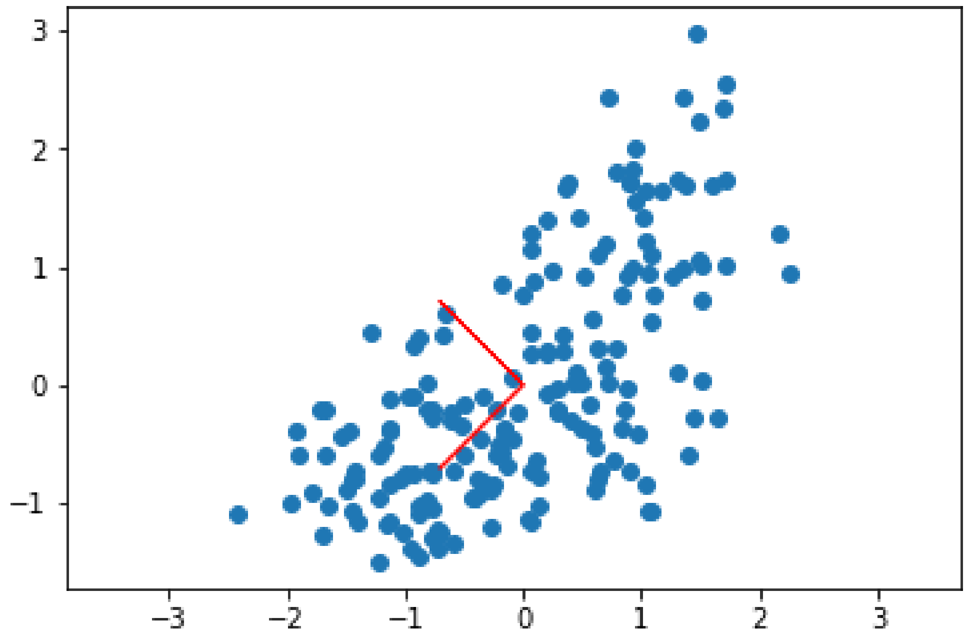
2. The two principal components are orthogonal (they have to be).

3.2 Part 2: Plotting Principal Components in Original Space

1. Please describe how the principal components relate to the points.

   The first principal component is the direction of maximum variance and the second principal component is the only vector orthogonal to the first in 2 dimensional space.

2. Paste the graph here of the plot of the given points (with both axis in same scale) as well as the lines representing the principal components in original space, with x1 in the x axis and x2 in the y axis.



3.3 Part 3: Plotting Data Projected onto Component Space

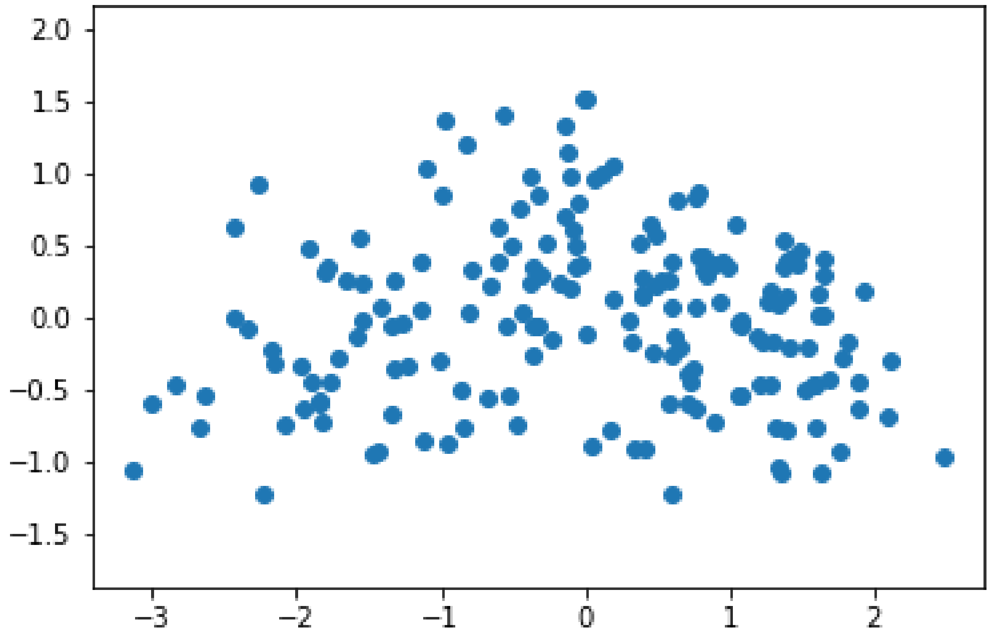
1. Explain how the graph of points on principal component space relates to the graph of points on original space above.

   It is the original space rotated 135^o counter-clockwise.

2. Explain the difference in distribution of points projected on the first component vs. projected on the second.

The distribution of points projected on the first component has maximal variance and so has greater variance than the distribution projected on the second.

3. Paste the plot of the given points (with both axis in same scale) in principal component space.



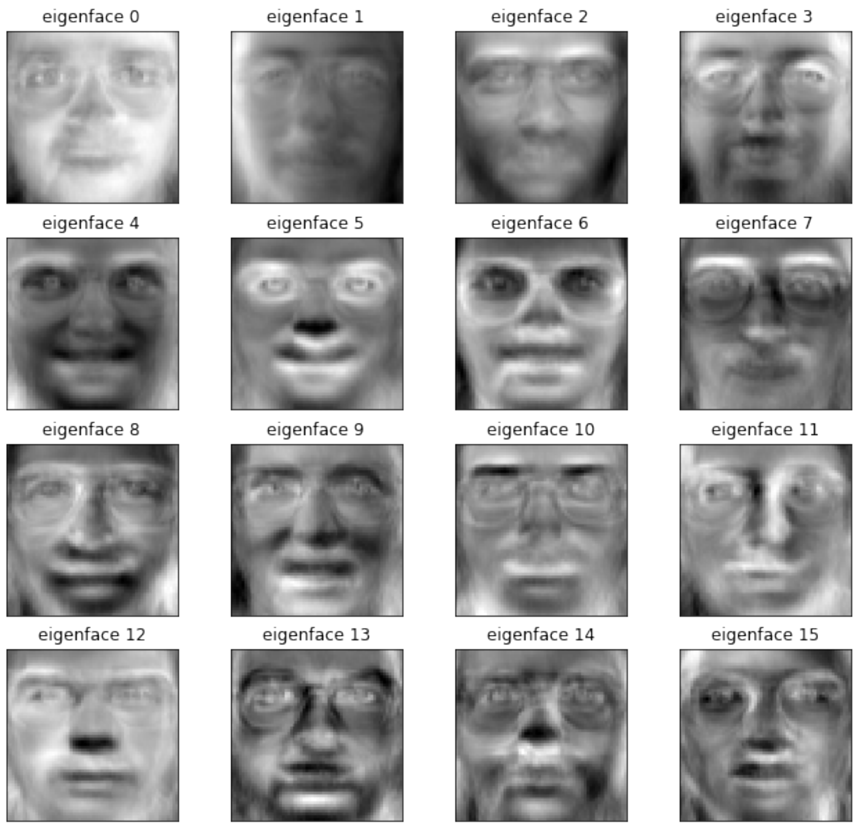
4 Principal Component Analysis on Faces

Now we will perform PCA on images of faces and see how reducing the latent dimensions of our images affects the images reconstructions. First, start by uncommenting the code below to retrieve the faces dataset.

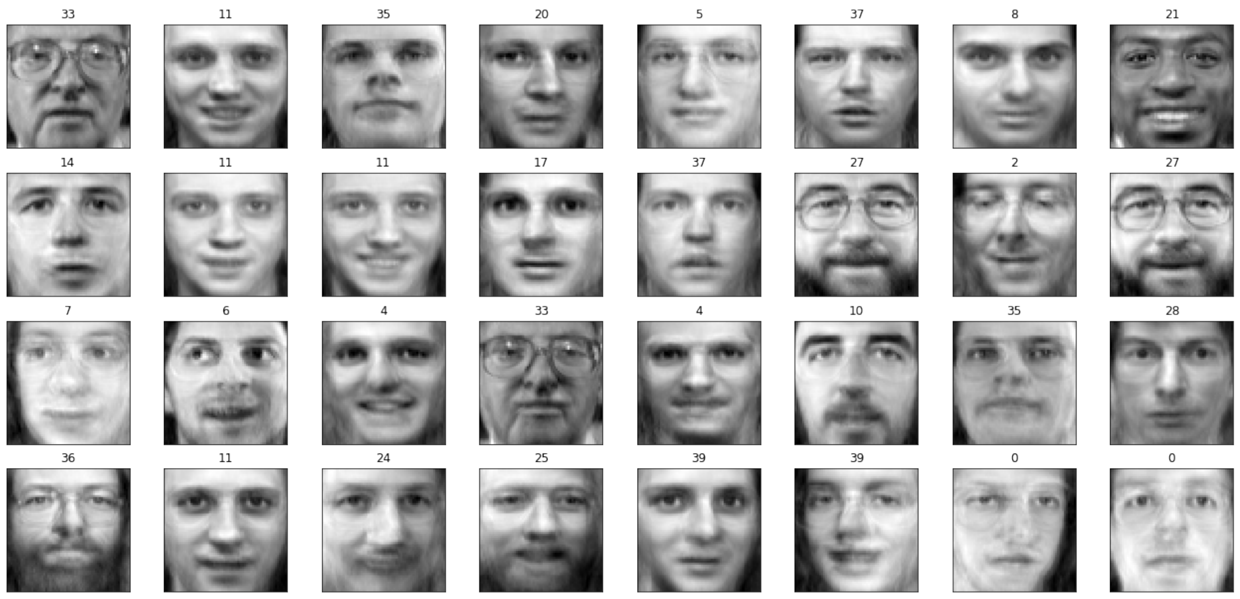
4.1 Part 1: PCA with SVD and Resulting Eigenfaces

1. This section is auto graded, checking the outputs of your PCA function.
2. Please describe what the eigenfaces look like. What do you expect to observe with the eigenfaces associated with lower eigenvalues?

   Each one is representative of a different aspect of the face, the importance of which (defined by that facial aspect's description of variance in the data set) decreases with each successive eigenface.
3. Please insert your eigenfaces output here.



4.2 Part 2: Reconstructing Faces



1. Paste in the reconstructed faces plot. Compare the reconstructed images to the original images. How are they similar and how are they different?

They are similar in that they retain much of the defining characteristic of the original face. However, the faces also lose much of their clarity, and more subtle features. Also, all the faces have a touch of average elements like light outlines of glasses.

2. What do you expect to see from the reconstructed images as the number of principal components chosen for PCA increases? Please explain why.

   As the number of principal components increases, the reconstructed faces will become closer and closer to the original faces. This is because PCA is an optimal lower dimensional representation of the original pixel space and so as the dimension of that representation increases, it will more and more resemble the original space.

4.3 Part 3: Variance Explanation

1. How do you expect (based on theory; please be precise!) the plot of variance explained as the number of components to relate to the eigenvalues of the corresponding components?

   The change in cumulative variance when a component is added should correspond directly to the eigenvalue of that component. In other words, the eigenvalue of a component is a measure of the variance it describes.

2. What is the relation between reconstruction error and the variance explained?

   They are inversely related. As the variance explained increases, more of the original distribution is accurately described, and so the reconstruction error will fall and vice versa.

3. Line plots:

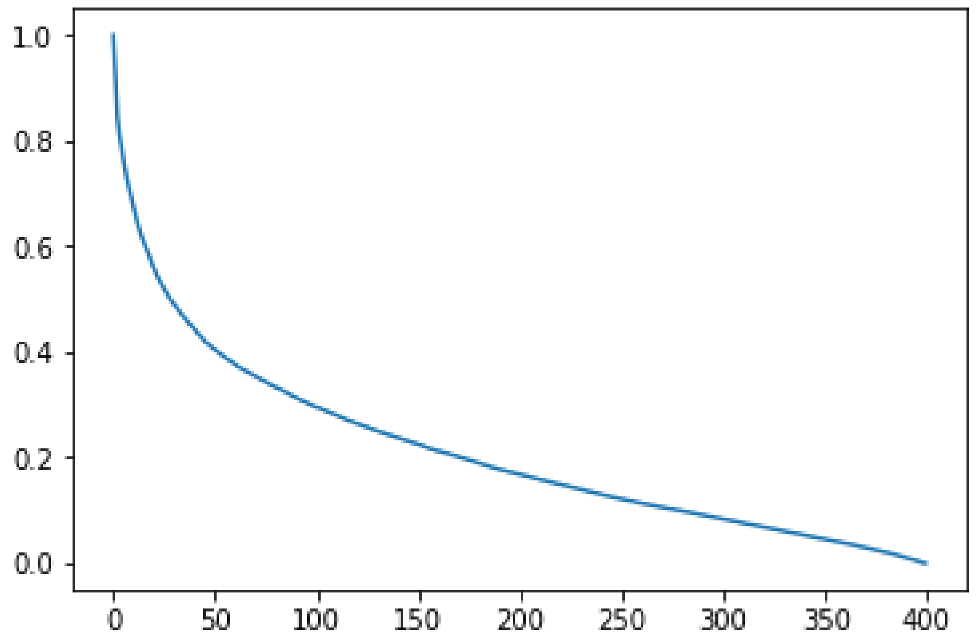


Figure 1: Reconstruction Error

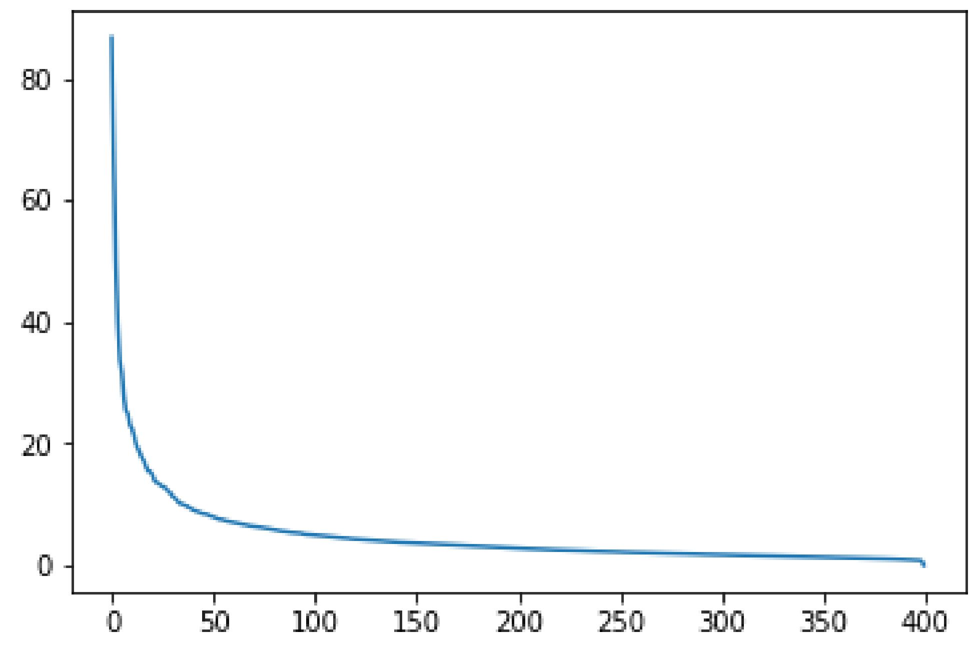


Figure 2: Singular Values

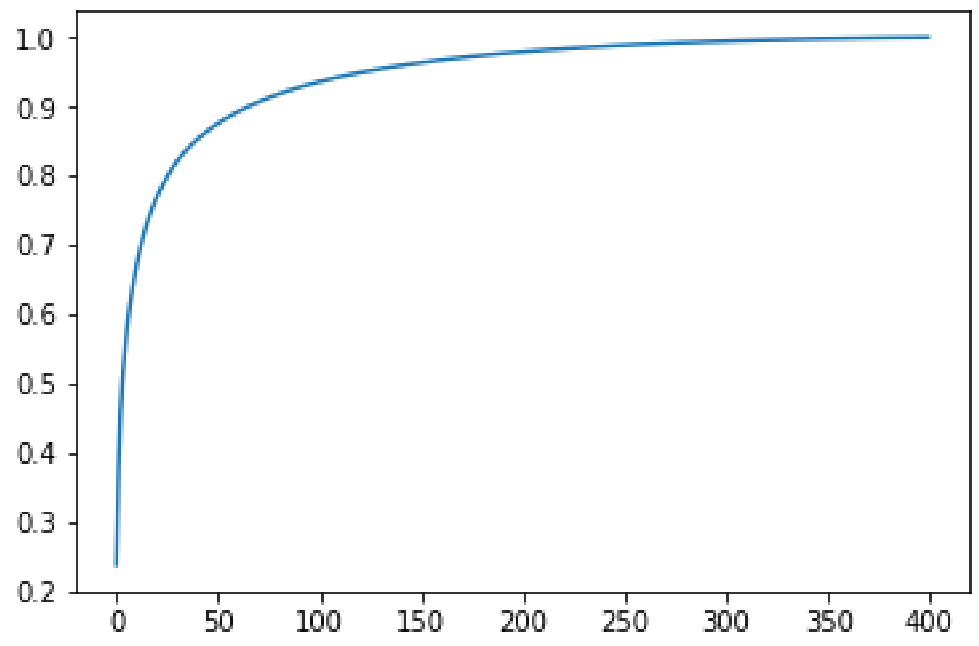


Figure 3: Normalized Cumulative Singular Values