CSE 546 Homework #3
Fall 2016

Janet Matsen

November 23, 2016

For this assignment, I went to office hours and developed strategies with Serena Liu and David Flemming.

My code is available at `https://github.com/JanetMatsen/Machine_Learning_CSE_546/tree/master/HW3`

1 PCA and reconstruction

1.1 Matrix Algebra Review

1.1.1 Show that $\mathbf{Tr}(AB^\intercal) = \mathbf{Tr}(B^\intercal A)$

Get the i, j^{th} element of $AB^\intercal$:
$(AB^\intercal)_{i,j} = \sum_z A_{i,z} B^\intercal_{z,j}$.
Similarly, $(B^\intercal A)_{i,j} = \sum_z B^\intercal_{i,z} A_{z,j}$.
The traces are then:

$$\mathrm{Tr}(AB^\intercal) = \sum_{i=1}^{n} \sum_{z=1}^{d} A_{i,z} B^\intercal_{z,j}$$

$$\begin{aligned}
\mathrm{Tr}(B^\intercal A) &= \sum_{j=1}^{d} \sum_{z=1}^{n} B^\intercal_{i,z} A_{z,j} \\
&= \sum_{z=1}^{n} \sum_{j=1}^{d} A_{z,j} B^\intercal_{i,z} \\
&= \sum_{i=1}^{n} \sum_{z=1}^{d} A_{i,z} B^\intercal_{z,i}
\end{aligned}$$

So $\mathrm{Tr}(AB^\intercal) = \sum_{i=1}^{n} \sum_{z=1}^{d} A_{i,z} B^\intercal_{z,j} = \mathrm{Tr}(B^\intercal A)$

1.1.2 Now we prove a few claims that helpful in the next problem. Show that ...

by SVD, $\Sigma = U^T \Lambda T$

$$\begin{aligned}
Tr(\Sigma) &= Tr(U^T \Lambda U) \\
&= Tr(\Lambda U U^T) \text{ (by previous question)} \\
&= Tr(\Lambda)
\end{aligned}$$

$$
\begin{aligned}
\Sigma &= \frac{1}{n} X^T X \\
U \Lambda U^T &= \frac{1}{n} X^T X \\
U^T U \Lambda U^T U &= U^T \frac{1}{n} X^T X U \\
\Lambda &= \frac{1}{n} U^T X^T X U \\
&= \frac{1}{n} X U U^T X^T \\
&= \frac{1}{n} X X^T \\
&= \frac{1}{n} ||X||^2
\end{aligned}
$$

1.2 PCA

I first centered the X matrix before doing PCA. In particular, I subtracted off the average of each column from each element. Any time vectors were brought back to image space, this center was added back on.

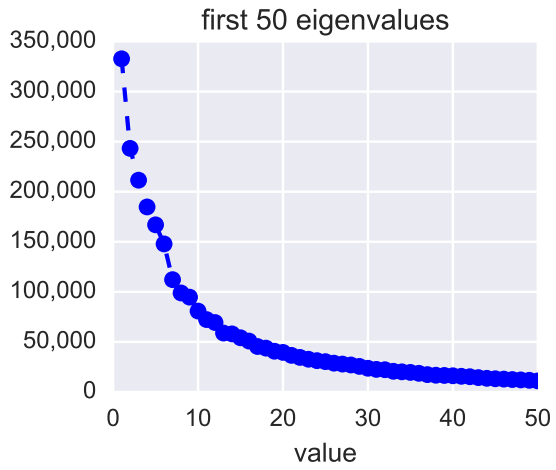
1.2.1 What are the first 50 eigenvalues?

Eigenvalues 1, 2, 10, 30, and 50 are:

```
332719.1
243279.9
80808.5
23685.7
11035.3
```

Visually, the top 50 are:

Figure 1: First 50 eigenvalues

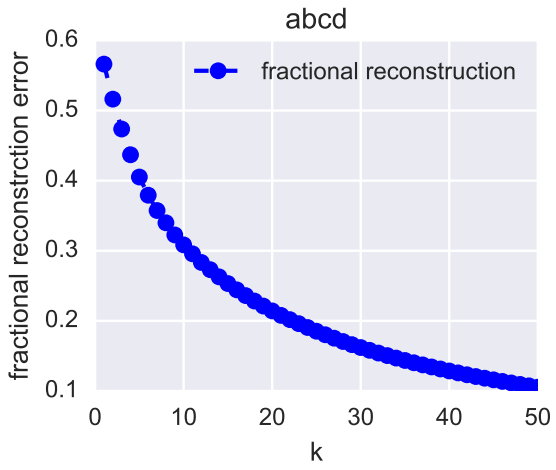


The sum of these first 50 eigenvalues is 2,827,256.7.

1.2.2 Fractional reconstruction error

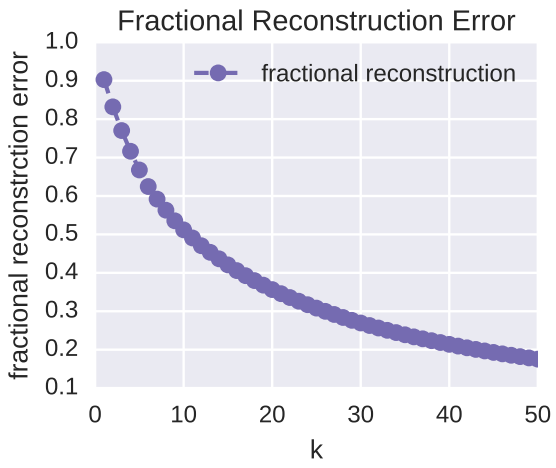
My plot before I centered each column looked like this:

Figure 2: Fractional reconstruction **before** centering the MNIST training data



After centering each column, my fractional reconstruction error improved:

Figure 3: Fractional reconstruction after centering the MNIST training data



I proceeded to use the eigenvectors for the centered MNIST data for the rest of the question.

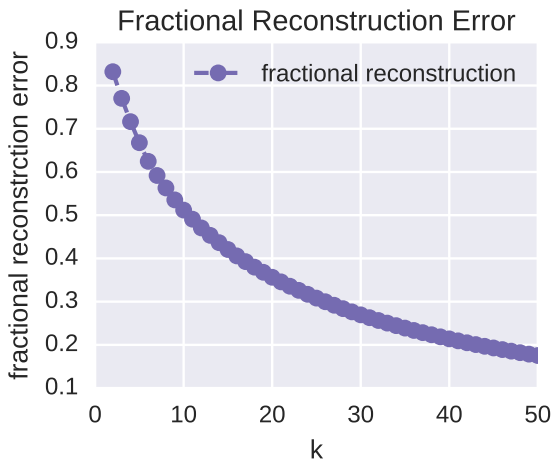
1.2.3 First eigenvalue. What does the first eigenvalue capture, and why do you think λ_1 is much larger than the other eigenvalues?

The first eigenvalue corresponds to the most important eigenvector in the MNIST data. The eigenvectors are shown later, but we can presume that the first eigenvalue captures the general notion of having image intensity at the image center. Later I saw that for my centered data, the eigenvalue also captures whether the digit is circle-like (e.g. 0, 8, 5) or stick-like (1, 7, 9).

1.2.4 fractional reconstruction error for dimensions 2 to 50

This looks very similar to the plots shown before.

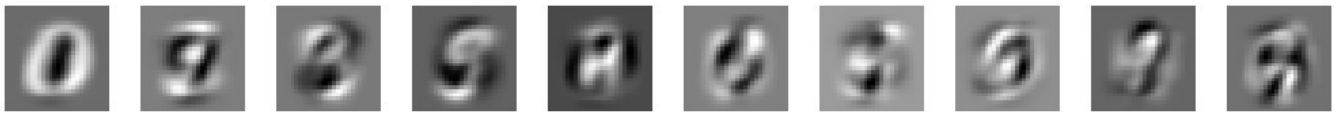
Figure 4: Fractional reconstruction after centering the MNIST training data: 2nd through 50th eigenvalues.



1.3 Visualization of the Eigen-Directions

1.3.1 Display the first 10 eigenvectors as images

Figure 5: The first 10 eigenvectors of the training MNIST data.



1.3.2 brief interpretation of what you think they capture

These eigenvectors capture the differences in shapes across numbers. No eigenvalue represents a single number, but rather highlight common stroke similarities across numbers, and areas where digits tend to differ. Not surprisingly, there is a lot of emphasis on whether the digit looks like a vertical stroke, and what pattern of image intensity is seen at the center of the image.

1.4 Visualization and Reconstruction

1.4.1 Visualize 5 digits

Figure 6: Reconstruction of sample MNIST digits, untransformed

1.4.2 visualize these digits, after we project them onto the top k eigenvectors

Figure 7: Reconstruction of sample MNIST digits, k = 2.

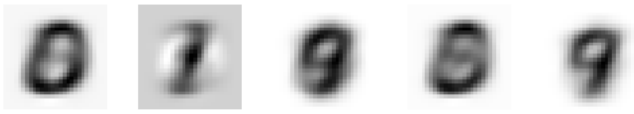


Figure 8: Reconstruction of sample MNIST digits, k = 5.

Figure 9: Reconstruction of sample MNIST digits, k = 10.

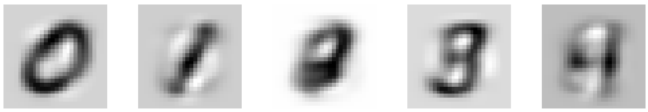


Figure 10: Reconstruction of sample MNIST digits, k = 20.

Figure 11: Reconstruction of sample MNIST digits, k = 50.

1.4.3 brief interpretation

Increasing the number of eigenvectors used to reconstruct the image leads to a more accurate representation of the original image. The first few eigenvectors give the reconstruction some notion of being a digit: they have less ink around the edges and some resemblance of a number in the center. Adding in more eigenvectors allows the more subtile differences between digits to be seen. Note that although 50 eigenvectors can do an excellent job reconstituting the original image, there are still blurry areas in most reconstructions, and the entirety of each image is less sharp than their originals.

2 state of the art on MNIST

2.1 Least Squares

In CSE 446, there was a strong emphasis on never touching the test data until all hyperparameters are validated on held-out validation data. For this reason, I did some hyperparameter tuning studies. Holding minibatch size fixed (10) and the η decay rate constant, I varied σ values in a search for one that gave the lowest validation error.

Figure 12: Hyperparameter tuning results for varying the kernel bandwidth, σ around the median distance between points. Models were trained on 20% of the training data. Only three epochs were allowed per model.

I didn't have time to run more than three epochs for this hyperparameter tuning test, so I tried a few sigma values near the minimum. I ended up using $\sigma = 1178$ I used this value on the entire test data set. Then use this sigma and the full 60,000-point training set to optimize the learning rate.

During hyperparameter exploration, it was crucial to diagnose diverging models quickly. To When searching for good values, I kept the kernel bandwidth, σ, constant, as well as the number of training points.

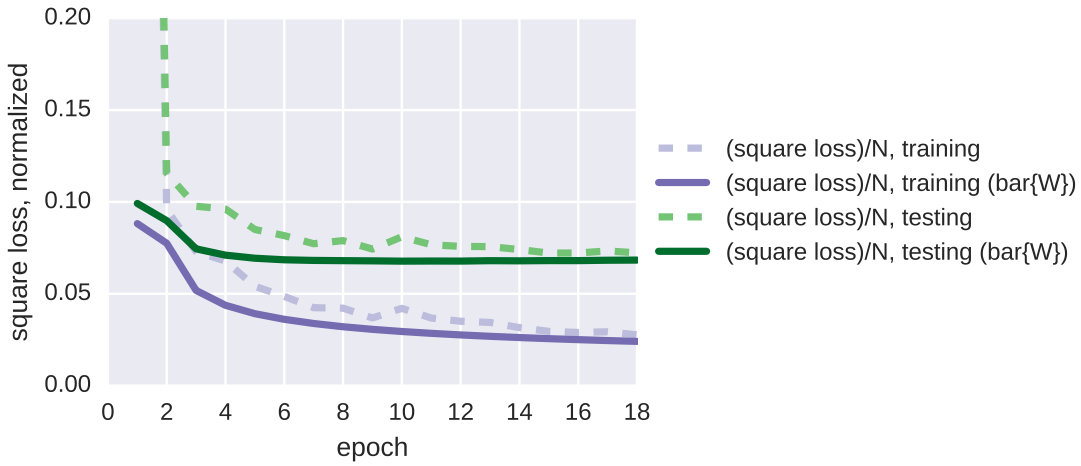
2.1.1 your learning rate (or learning rate scheme), mini-batch size, and the kernel bandwidth

My hyperparameter explorations led me to a kernel bandwithd σ of (median σ)/2 = 1177.66. I used an initial learning rate of 0.00277, which was set automatically by my method . η was

scaled by $1/\text{epoch}^{0.5}$. My mini-batch size was 10.

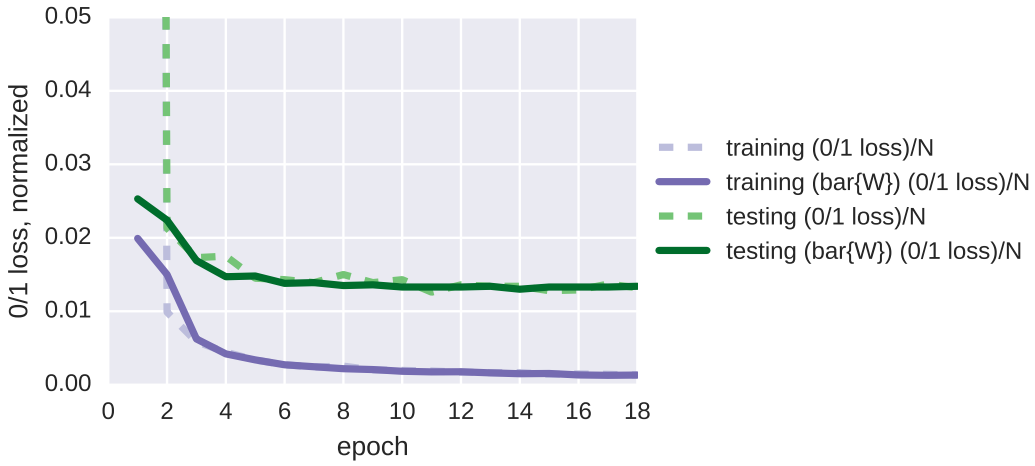
2.1.2 plot square loss after every epoch

Figure 13: Square loss on the train and test data during model fitting.



2.1.3 0/1 loss plots

Figure 14: 0/1 loss on the train and test data during model fitting.



2.1.4 final training squared loss and 0/1 loss, total number of mistakes

Metric	Value
squared loss	0.024
0/1 loss	0.00132
number of mistakes	79 (out of 60,000)

Table 1: Model fit on training data

2.1.5 final test squared loss and 0/1 loss, total number of mistakes

Metric	Value
(squared loss)/N	0.068
(0/1 loss)/N	0.0132
number of mistakes	132 (out of 10,000)

Table 2: Model fit on test data

3 SVMs: Hinge loss and mistake bounds

3.1 Argue that the function $\ell((x, y), w) = \max\{0, 1 - yw^T x\}$ is convex (as a function of w.)

Wikipedia says the maximum of two convex functions is convex.
Straight lines are convex, so 0 is convex.
To show that $1 - yw^T x$ is convex, we need to show that the definition holds.
The definition of convexity is
$\forall x_1, x_2 \in X, \forall t \in [0, 1] : f(tx_1 + (1 - t)x_2) \leq tf(x_1) + (1 - t)f(x_2)$.
Let $f((x, y), w_1) = 1 - yw^T x$.
f is convex if $tf((x, y), w_1) + (1 - t)f((x, y), w_2) \geq f((x, y), tw_i + (1 - t)w_2)$.
Show this:

$$\begin{aligned} tf((x, y), w_1) + (1 - t)f((x, y), w_2) &\geq t(1 - yw_1^T x) + (1 - t)(1 - yw_2^T x) \\ &\geq t - tyw_1^T x + 1 - yw_2^T x - t - tyw_2^T x \\ &\geq 1 - tyw_1^T x - yw_2^T x - tyw_2^T x \\ &\geq 1 - y(tw_1^T x + w_2^T x + tw_2^T x) \\ &\geq 1 - y(tw_1^T x + (1 - t)w_2^T x) \\ &\geq f((x, y), tw_1 + (1 - t)w_2) \end{aligned}$$

Both 0 and $1 - yw^T x$, so ℓ is convex.

3.2 What range of hinge loss?

When a point is correctly classified, the sign of $y_i w^T x_i$ is always positive.
This means that $1 - yw^T x$ is largest when $yw^T x = 0$.
Te max argument also binds us to positive values.
Thus the hinge loss for correctly classified points is thus in the range $[0, 1]$.

3.3 Let M(w) be the number of mistakes made by w on our dataset (in termsof the classification loss). Show that:

$$\frac{1}{n} M(w) \leq \frac{1}{n} \sum_{i=1}^{n} \max\{0, 1 - y_i w^T x_i\}$$

Define a new function l such that $l(z) = 0 for z > 0$, and $l(z) = 1 for z < 0$.
Let $h(z) = max\{0, 1 - z\}$.
Note that $l(z) \leq h(z)$.
When we make a mistake, this corresponds to $y_i w^T x_i$ being negative, and thus $l(y_i w^T x_i = 1$.
This means $M(w) = \sum_i l(y_i w^T x_i) \leq \sum_i h(y_i w^T x_i) = \sum_{i=1}^{n} max\{0, 1 - y_i w^T x_i\}$.
Thus $\frac{1}{n} M(w) = \frac{1}{n} \sum_{i=1}^{n} max\{0, 1 - y_i w^T x_i\}$.

4 Fitting an SVM classifier by hand

4.1 Write down a vector that is parallel to the optimal vector $\hat{w}$.

point	x value	class	mapped value
1	0	-1	$[1, 0, 0]^T$
2	$\sqrt{2}$	1	$[1, 2, 2]^T$

$\hat{w}$ is parallel to the point connecting the two support vectors. This vector is $[1, 2, 2] - [1, 0, 0] = [0, 2, 2]$. The decision boundary is orthogonal to this vector.

4.2 What is the value of the margin that is achieved by this $\hat{w}$?

The distance between these two points is twice the value of the margin. The distance between the two points is $\sqrt{(1-1)^2 + (2-0)^2 + (2-0)^2} = \sqrt{0+4+4} = \sqrt{8}$. So the margin is $\sqrt{8}/2 = \sqrt{2}$.

4.3 Solve for $\hat{w}$, using the fact the margin is equal to $1/||\hat{w}||$.

$$
\begin{aligned}
margin &= \frac{1}{||\hat{w}||} \\
&= \frac{1}{\sqrt{w_1^2 + w_2^2 + w_3^2}} \\
&\text{(dropped the hat notation for ease)} \\
margin^2 &= \frac{1}{w_1^2 + w_2^2 + w_3^2} \\
w_1^2 + w_2^2 + w_3^2 &= \frac{1}{margin^2} \\
&= \frac{1}{\sqrt{2}^2} = \frac{1}{2}
\end{aligned}
$$

so we know that:

$$w_1^2 + w_2^2 + w_3^2 = \frac{1}{2}$$

And we know that $\hat{w}$ is parallel to $[0, 2, 2]$ so $\hat{w} = a[0, 2, 2]$, or $\hat{w} = [0a, 2a, 2a]$.

$$\begin{aligned} a^2 \cdot 0^2 + a^2 \cdot 2^2 + a^2 \cdot 2^2 &= 1/2 \\ (a^2 \cdot 4)2 &= 1/2 \\ 8a^2 &= 1/2 \\ a^2 &= 1/16 \\ a &= 1/4 \end{aligned}$$

(we don't want -1/4 or sign would be wrong given our 2 support vectors)

plug a back into first line

$$\begin{aligned} \hat{w} &= [0, 2/4, 2/4] \\ &= [0, \frac{1}{2}, \frac{1}{2}] \end{aligned}$$

Answer: $\hat{w} = [0, \frac{1}{2}, \frac{1}{2}]$

4.4 Solve for $\hat{w}_0$ using your value for w and Equations 1 to 3

Equation 2 becomes:

$$\begin{aligned} (-1) * ([w_1, w_2, w_3][1, 0, 0]^T + w_0) &\geq 1 \\ -w_1 - w_0 &\geq 1 \\ 0 - w_0 &\geq 1 \\ w_0 &\leq -1 \end{aligned}$$

Equation 3 becomes:

$$\begin{aligned} (+1) * ([w_1, w_2, w_3][1, 2, 2]^T + w_0) &\geq 1 \\ w_1 + 2w_2 + 2w_3 + w_0 &\geq 1 \\ 0 + 2(1/2) + 2(1/2) + w_0 &\geq 1 \\ 2(2(1/2)) + w_0 &\geq 1 \\ 2 + w_0 &\geq 1 \\ w_0 &\geq -1 \end{aligned}$$

The only solution is then $w_0 = -1$

4.5 Write down the form of the discriminant function $f(x) = \hat{w}_0 + \hat{w}^T\phi(x)$ as an explicit function of x. Plot the 2 points in the dataset, along with f(x) in a 2d plot.

Above we found that $\hat{w} = [0, \frac{1}{2}, \frac{1}{2}]$, and that $\hat{w}_0 = -1$.
Then the discriminant function is:

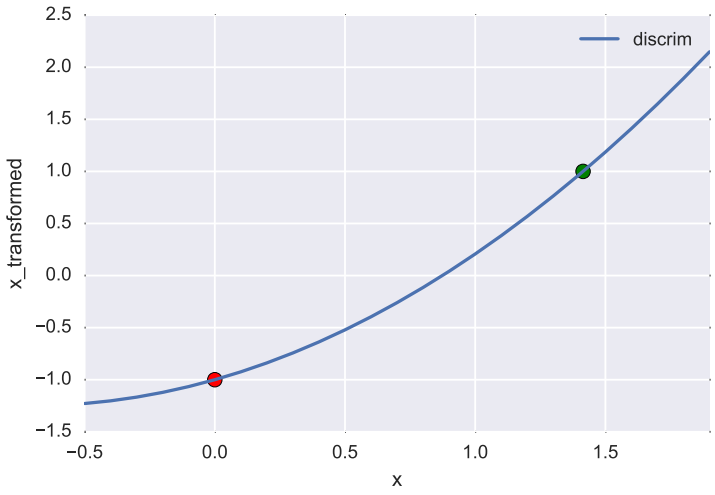
$$\begin{aligned} f(x) &= \hat{w}_0 + \hat{w}^T\phi(x) \\ &= -1 + [0, \frac{1}{2}, \frac{1}{2}][1, \sqrt{2}x, x^2]^T \\ &= -1 + 0 + \frac{1}{2}\sqrt{2}x + \frac{1}{2}x^2 \\ &= -1 + \frac{1}{2}(\sqrt{2}x + x^2) \end{aligned}$$

For point x_1, $x = 0$, so the discriminant function produces $-1 + \frac{1}{2}(\sqrt{2} \cdot 0 + 0^2) = -1$
For point x_2, $x = \sqrt{2}$, so the discriminant function produces

$$\begin{aligned} -1 + \frac{1}{2}(\sqrt{2}\sqrt{2} + \sqrt{2}^2) &= -1 + \frac{1}{2}(2 + 2) \\ &= -1 + \frac{1}{2}(4) \\ &= -1 + 2 \\ &= 1 \end{aligned}$$

This result is plotted below. The x-axis is the input x for each data point. The y-value is that data point transformed by the discriminant function. The green dot is our example x_2 (positive) point and the red dot is the x_1 (negative) point. The blue line is the value of the discriminant function for intermediate values.

Figure 15: SVM by hand.



This kernel SVM problem allows us to have a nonlinear classification rule for input feature x values.

5 K-Means

5.1 Run the algorithm

I used the 50-PCA-component projections for this question. My clusters were initialized by sampling 50 random points without replacement.

5.1.1 Run the K-means algorithms with K = 16. Describe and plot.

Figure 16: k = 16. Squared reconstruction error vs iteration number. When dividing by N=60,000, the final reconstruction error is 1.77e+06

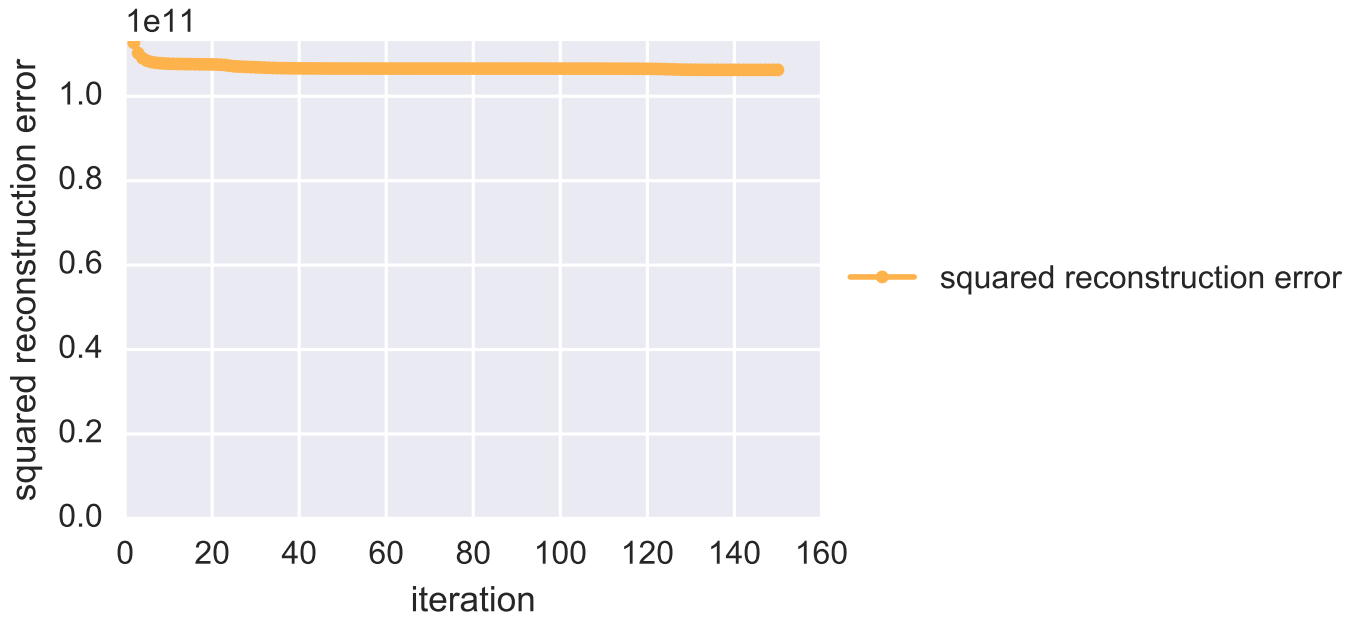


Figure 17: k = 16. Number of assignments per cluster.

Figure 18: k = 16. Centers in order of number of points assigned.

5.1.2 Provide a brief interpretation for the k = 16 case.

With only 16 clusters, you get somewhat generic representations of each numbers, with some duplications. Note that the 6th cluster looks like a mix of 8, 5, and 3. I surmised that the points assigned to this cluster would be mostly threes and fives. The counts of labels for the points assigned to this center are:

Table 3: True labels for cluster 6

Label	Count
8	1702
3	1059
5	995
2	165
9	111
0	99
1	26
6	24
4	17
7	13

Similarly, the centers that "look" very pure are dominated by the label that the PCA reconstruction illustrates.

5.1.3 K-means algorithms with K = 250. Plot.

Figure 19: k = 250. Reconstruction error after each step. When dividing by N=60,000, the final reconstruction error is 1.05e+06

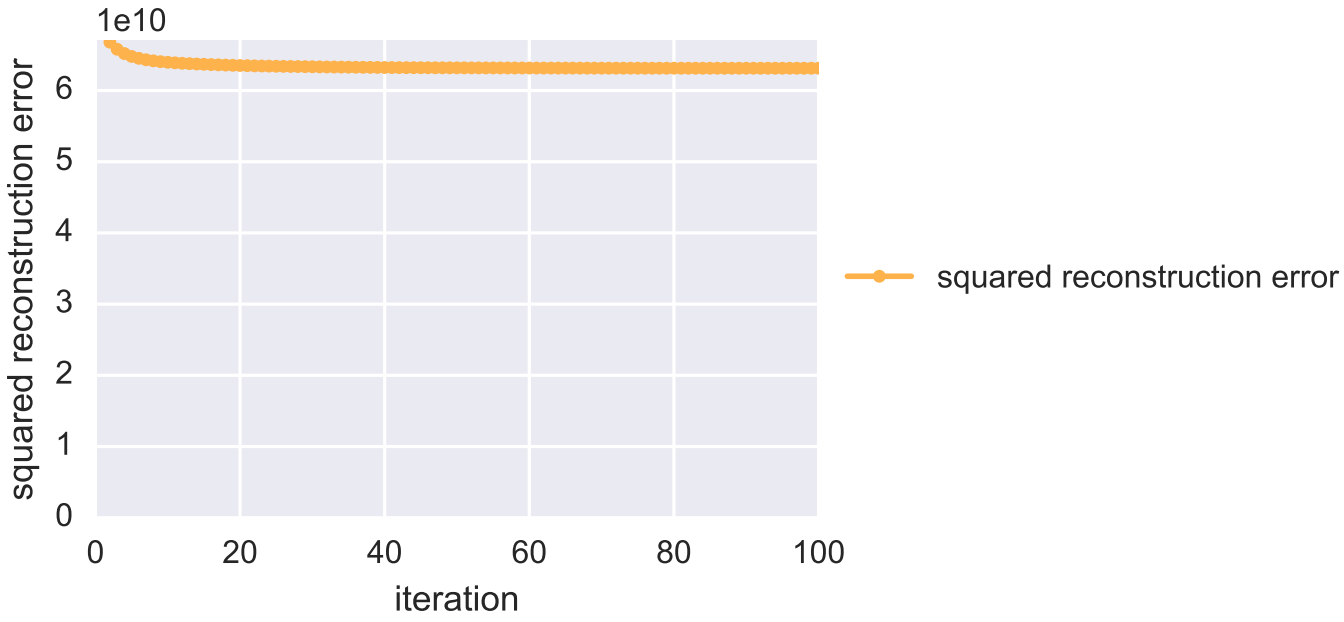


Figure 20: k = 250. Number of assignments for each center in descending order.

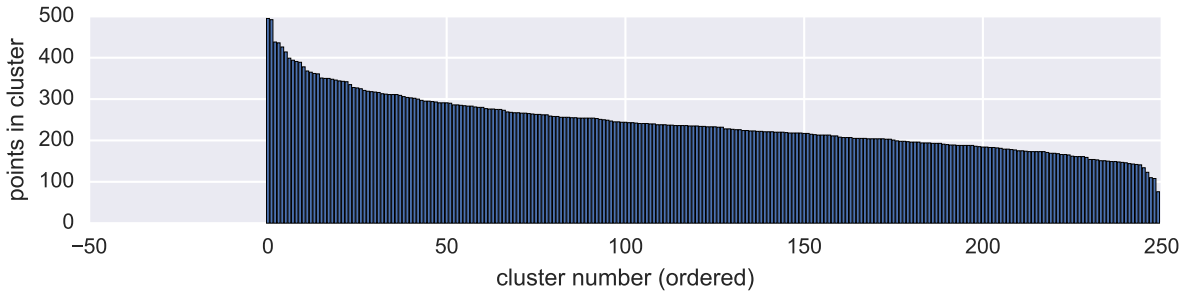


Figure 21: k = 250. Sixteen random centers.

As expected, with 250 centers, typical centers look more like pure numbers than blends.

5.2 Classification with K-means

5.2.1 For K = 16, what are your training and test 0/1 losses?

0/1 loss	3.18 e+03
(0/1 loss)/N	3.18 e-01

5.2.2 For K = 250, what are your training and test 0/1 losses?

0/1 loss	8.03 e+02
(0/1 loss)/N	8.03 e-02