CS4786/5786: Machine Learning for Data Science, Spring 2015
2/21/2015: Assignment 1: PCA, CCA and Random Projections

Instructions Due at 11:59pm Monday March 7 on CMS. Submit what you have at least once by an hour before that deadline, even if you haven't quite added all the finishing touches — CMS allows resubmissions up to, but not after, the deadline. If there is an emergency such that you need an extension, contact the professor.

You may work in groups of one up to four. Each group of two or more people must create a group on CMS well before the deadline (there is both an invitation step and an accept process; make sure both sides of the handshake occur), and submits 1 submission per group. You may choose different groups for different assignments. The choice of the number "four" is intended to reflect the idea of allowing collaboration, but requiring that all group members be able to fit "all together at the whiteboard", and thus all be participating equally at all times. (Admittedly, it will be a tight squeeze around a laptop, but please try.) Please ensure that each member of the group can individually defend or explain your group's submission equally well.

You will submit both a writeup and some datafiles you create. The writeup can be handwritten or typeset, but please make sure it is easily readable either way.

Keep an eye on the course webpage for any announcements or updates.

Academic integrity policy We distinguish between "merely" violating the rules for a given assignment and violating *academic integrity*. To violate the latter is to commit *fraud* by claiming credit for someone else's work. For this assignment, an example of the former would be getting an answer from person X who is not in your CMS-declared group but stating in your homework that X was the source of that particular answer. You would cross the line into fraud if you did not mention X. The worst-case outcome for the former is a grade penalty; the worst-case scenario in the latter is academic-integrity hearing procedures.

The way to avoid violating academic integrity is to always document any portions of work you submit that are due to or influenced by other sources, even if those sources weren't permitted by the rules.[1]

---

*Q1 (***The Picture Mishap).**
The back story:
You took some photos of smiley faces a while ago and knowing about the PCA algorithm, compressed these photos. The original photos were of size 105×105 grey-scale images represented as 11025 dimensional vectors. The compressed photos were 20 dimensional each along with the projection matrix W of size 11025×20 and the mean vector μ of size 11025×1.

[1] We make an exception for sources that can be taken for granted in the instructional setting, namely, the course materials. To minimize documentation effort, we also do not expect you to credit the course staff for ideas you get from them, although it's nice to do so anyway.

Accidentally one night, you deleted all but one of the compressed version of the photos and all you have now is the projection matrix W, the mean vector μ and the compressed 20 dimensional vector of just the first image say $\mathbf{y}_1$. Fortunately for you, one of your friends has printed copies of these photos. However as luck would have it, these photos were all stained with an identical stain on all of them. To make things worse, your friend takes a picture of all these photos at an angle (in this assignment for my ease a 90 degree angle but in general this is some unknown angle of rotation) and emails these photos over to you. So now you have these 28 images all rotated and all with identical stain on them. Can you use your knowledge of PCA to reconstruct your favorite smiley photos?

Pedagogical nugget: The aim of this problem is to learn *translation* and *rotation invariance* of PCA, and how we can use these property to do some cool stuff. Sample real-life setting: when you get data measurements from different sources that didn't check with each other ahead of time and so some of the features might have been reordered. We shall build up to our finale by walking you through various steps.

1. (warmup to make sure you have the right tools available and get used to them; **nothing to turn in for this part**) Your first task is to generate 2 dimensional gaussian distributed random variables. Generate 1000 gaussian distributed, 2-dimensional random variables X, such that the first dimension of X is normal distributed with variance 1 and the second dimension of X is independently drawn, normal distributed with variance 2. Scatter-plot these points. They should look like an elongated ellipse (if they don't look elongated look at range of X and Y axis, they have to be the same). Now duplicate X and call it say X_{dup}. First, we shall rotate all the points in X_{dup} by a 45 degree angle. Do this by multiplying X_{dup} with rotation matrix
$$R = \begin{bmatrix} \cos(\pi/4) & -\sin(\pi/4) \\ \sin(\pi/4) & \cos(\pi/4) \end{bmatrix} = \begin{bmatrix} 1/\sqrt{2} & -1/\sqrt{2} \\ 1/\sqrt{2} & 1/\sqrt{2} \end{bmatrix}$$
Now translate X_{dup} by adding to every point in X_{dup} (i.e. every row) the vector $(1, 1)$. Now scatter-plot in the same figure you plotted X's the points Xd (with different color). You will now see another elongated ellipse at a 45 degree angle with center roughly at $(1, 1)$.

   Run PCA on X and X_{dup} to get two corresponding projection matrices W and W_{dup}. Recall that the columns of these two matrices are the principal directions, i.e., the eigenvectors of the corresponding covariance matrices. Then, apply the projections W and W_{dup} to their corresponding data matrices X and X_{dup} to yield Y and Y_{dup}.

   - Scatter plot Y and Y_{dup}. What do you see? Compare just the magnitude of first column of Y with that of Y_{dup} and similarly for the second column, what do you see?
   - Look at W and W_{dup}, these will be different, also look at $\text{Cov}(X)$ and $\text{Cov}(X_{\text{dup}})$ the covariance matrices corresponding to the two views, these will also be different.

   When you examine Y and Y_{dup}, you should see that the absolute values of corresponding entries are the same!

This part of the question, is more for you guys to get warmed up, nothing to turn in for this part.

2. Let's now generalize what we've just observed. A rotation matrix is a square matrix whose transpose is its inverse. (Intuition: a clockwise rotation can be undone via counterclockwise rotation by the same angular amount.) As usual, let our $n \times d$-dimensional data vectors be denoted by $\mathbf{x}_1, \ldots, \mathbf{x}_n$ (example: the rows of your X matrix above) and let R be a $d \times d$ rotation matrix.

   For simplicity, you may assume that the $\mathbf{x}_t$s have been centered at 0. Let $\mathbf{x}'_t = R\mathbf{x}_t + \mathbf{v}$ where $\mathbf{v}$ is some fixed translation. (example: the rows in your Xd matrix above), forming a second dataset.

   Now, for any K we pick, let us use PCA on each of the two data sets to obtain K-dimensional projections $\mathbf{y}_1, \ldots, \mathbf{y}_n$ and $\mathbf{y}'_1, \ldots, \mathbf{y}'_n$, respectively.

   (a) Write down a relationship between the two PCA projection matrices W and W' in terms of the rotation matrix R and the translation vector $\mathbf{v}$. Explain mathematically how you arrived at this answer.

   (b) Explain why for any $t \in \{1, \ldots, n\}$, the entries of $\mathbf{y}_t$ and $\mathbf{y}'_t$ are the same up to sign.

   Hint: to explain why the signs in corresponding entries might be flipped, consider the following question: if a vector $\mathbf{b}$ is an eigenvector of matrix C, must it follow that -$\mathbf{b}$ is?

3. Now lets get back to our story of the "The Picture Mishap".
   Question: You have the projection matrix W given to you in the file `W.csv`, the mean vector μ given in file `Mu.csv` and the first compressed face $\mathbf{y}_1$, a 20 dimensional compressed vector given to you in `Y1.csv`. You are also given the bunch of stained rotated images your friend emails you. To make it simpler for the assignment, we have already vectorized these images and these are provided to you in the file `Xbad.csv` where each row of this file is a vectorized version of each of the 28 images. The first image in this file corresponds to the first compressed image $\mathbf{y}_1$ you posses. **Your goal is to reconstruct the matrix X of original images (vectorized) and submit the file** `X.csv`. That is `X.csv` has 28 rows, each row corresponding to each of the 28 images. Each row has 11025 entries corresponding to the vectorized version of the reconstructed images. You don't need put these back in image format, if you want to for your own curiosity, you can. The code snippet to take the vectorized version of reconstructed image t and convert it into a 105×105 image is

```
For m = 1 to 105
   For n = 1 to 105
      I(n,m,t) = X(t,(m-1)*105+ n);
   End
End
```

But remember you dont need to do this for the submission, this is only for your curiosity.

How to try out the above:
Remember that any two low-dimensional projections produced via PCA of the *same* data that is only rotated and translated will have the same absolute values for the entries of the $\mathbf{y}$'s but might have their signs flipped on coordinates. This is annoying, but can be fixed because, luckily, there exists that first image of *the same smiley* for which you have the original low dimensional projection and also have the same image in stained and rotated formal. Take your projection $\mathbf{y}_1$ and the 20 dimensional projection $\tilde{\mathbf{y}}_1$ corresponding to this face got by performing PCA on the stained images. Now, if on any coordinate, the sign of $\tilde{\mathbf{y}}_1$ and $\mathbf{y}_1$ are different, then for this coordinate flip the sign of *all* the $\tilde{\mathbf{y}}_t$'s. Finally, take the *new* $\tilde{\mathbf{y}}_t$'s and perform image reconstruction.

*Q2 (***Uncovering Secrets (Multiple-view CCA)).** The goal of this question is to get a better understanding of CCA and use it to uncover shared secrets among three friends.

Story: Three friends Alice(A), Bob(B) and Carol(C) all listen to 1000 songs on youtube. They each provide 10 dimensional vectors describing each of the 1000 songs given to you in files `XA.csv`, `XB.csv` and `XC.csv`. Only these vectors carry secret information about likes and dislikes encoded in them which can be uncovered only by the right linear projection technique. Your goal in this problem is to extract the joint rating of all three Alice, Bob and Carol. Your second goal is to extract the rating of songs that Alice and Bob secretly like that Carol is unaware of. The information in the three sets of 10 dimensional vectors are all encoded linearly.

1. Use CCA to extract a one dimensional projection that retains the information that is common to all three views. That is, you will get a 1 dimensional projection of (any one of the views) which provide a rating/ like dislike that all three of A, B and C share. **Submit to us this one dimensional vector in `YABC.csv`. Explain your solution.**

2. Next use CCA to extract a one dimensional projection that pulls out information shared between Alice and Bob but not Carol.
   Hint: If you do CCA between Alice and Bob, the information will contain not only shared information between Alice and Bob alone but also the information shared between all three. Figure out a way to remove information common between all three from the information shared between Alice and Bob. **Submit to us this one dimensional vector in `YAB.csv`. Explain your solution.**

As a way for you to check your answers we are also providing you with label files `LabelABC.csv` and `LabelAB.csv`. If you threshold $YABC$ at 0 and use this to predict the labels $LabelABC$ it should have a good accuracy (need not be 100%). Warning: after thresholding either > 0 or < 0 could indicate a label of 1. If you get very low accuracy in predicting the labels, thats a good sign since you can simply flip labels. Similarly threshold YAB at 0 and compare labels with $LabelAB$.

Hint: The common information between A, B and C can be linearly projected finally only to a single dimension. Similarly, the secret rating between A and B not shared with C again,

in the end, can be linearly projected to one dimension. Another hint that might come in handy: if the top eigen vectors represent common information then what do the bottom ones for a given view represent?

*Q3 (***Random Projection Vs PCA).** Your goal in this question is to generate two data sets consisting of 100, 1000-dimensional points. For each point $\mathbf{x}_t$ in the two data sets, ensure that their norm (distance to **0**) is exactly 1. We shall perform PCA and random projections on both the data sets to $K = 20$ dimensions. The data sets should be such that on the first one PCA outperforms random projection by a large margin and in the second, random projection outperforms PCA.

To evaluate our projections we shall use the following metric on how well the projections preserve average inter point distances:

$$\mathrm{Err}(\mathbf{y}_1, \ldots, \mathbf{y}_n) = \frac{1}{n(n-1)/2} \sum_{i=1}^{n} \sum_{j=i+1}^{n} \left| \|\mathbf{y}_j - \mathbf{y}_i\|_2 - \|\mathbf{x}_j - \mathbf{x}_i\|_2 \right|$$

You shall pick $K = 20$ and perform PCA and random projections on both the data sets. Your task in this problem is to create the data sets such that

- On the first data set, Err of PCA is much smaller compared to that of random projection.
- On the second data set, the Err of Random Projection is much smaller compared to that of PCA.

Submit your two datasets as csv files `PcaBeatsRp.csv` and `RpBeatsPCA.csv` in the same format we've used in the files we supplied you (so, they should be plain-text files with 100 lines, each with 1000 comma-separated numbers in it). Also, in your assignment writeup, explain how you generated the two data sets and the rationale behind this choice. Your rationale should explain how you used the properties of what RP and PCA produce to guide your thinking.