Chapter 30

Class 18: Principal Components Analysis

Schedule

30.1 Introduction to Principal Components Analysis [15 mins]

In a previous assignment you explored, in a graphical manner, the relationship between the eigenvectors of the covariance matrix and the distribution of the data. For instance, you looked at the daily temperature values in Boston versus Sao Paolo and the daily temperatures in Boston versus Washington D.C.

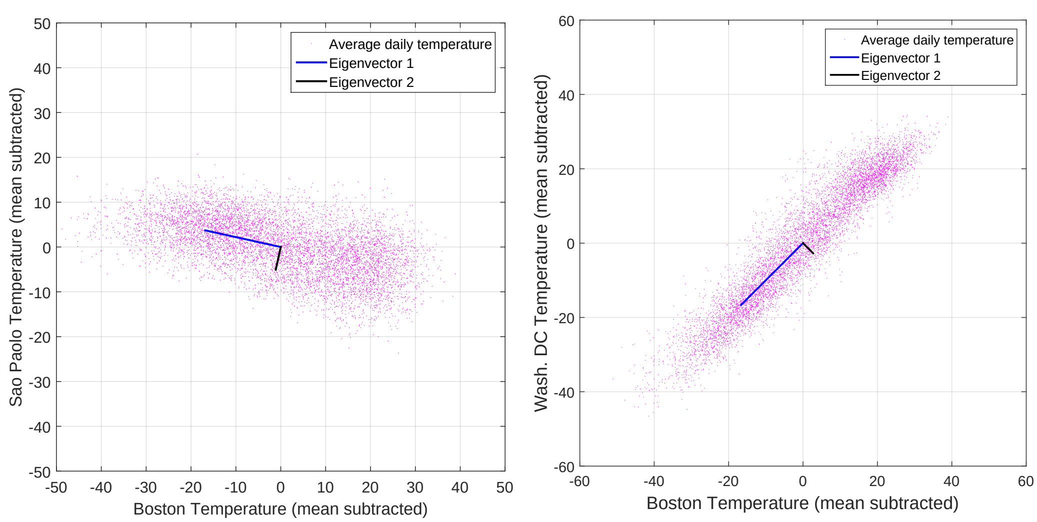

Figure 30.1: Centered average daily temperatures of Boston vs Sao Paolo (left) and Boston vs Washington DC, with the eigenvectors of the covariance matrix.

From visually inspecting these figures we saw that eigenvector 1, which corresponded to the larger of the two eigenvalues, seemed to be pointing in the direction where the data exhibited the most variability (i.e., the data was most spread out along this direction). You also looked at this for a 3D dataset consisting of the temperatures from Boston, Sao Paolo, and Washington DC.

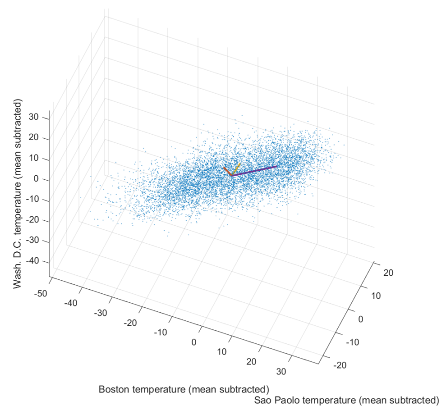

Figure 30.2: Temperatures and eigenvectors for Boston, Sao Paolo, and Washington DC

In this 3D dataset, we see the same phenomenon: that the principal eigenvector points along the direction of maximum variation in the data. It turns out that this phenomenon will hold no matter the dimensionality of the data (it works for 4D datasets, 10D datasets, and even datasets with 1,000s of dimensions)! This fact provides the basis for principal components analysis (PCA).

Exercise 30.1

Discuss why it makes sense to project data onto directions in which there is most variation in the data for both data compression and face recognition.

In PCA, instead of working with the data in its original form, we express it in a basis given by eigenvectors of the covariance matrix that have the largest eigenvalues. If that sounds confusing, don't worry! We are going to spend the next few classes breaking this concept down. We can understand the concepts of using this basis through two key properties.

- *Property 1*: the principal eigenvectors of the covariance matrix will maximize the variance of the data when the data is projected onto these vectors (we can think of vectors that capture large variation in the data as representing important properties of the data).
- *Property 2*: the principal eigenvectors of the covariance matrix will allow us, in a particular sense, to optimally compress our data. That is, we will be able to recover the original data with the highest possible accuracy from the projections of the data onto the principal eigenvectors.

The power of PCA lies in its ability to achieve both of these properties simultaneously. For this reason, the principal components of a dataset will act as keys to unlocking the secrets lurking in the data!

30.2 Initial Explorations of PCA

There are two paths forward; you should pursue them both, but you can choose with your partner/table which one to pursue first. If you want to see what this looks like with some toy data by graphing in MATLAB, go to the next section. If you want to think through this conceptually first from an application standpoint, then skip to section after that.

Then do the other section!

30.2.1 Seeing PCA Dimensionality Reduction in Dimensions [25 minutes]

Download and work through this notebook. Remember that when you download a Live Editor notebook from Google Drive, it looks like a bunch of files. Just click download symbol in the top right corner. It will show you a simple example of data in 2 dimensions and what it looks like if you use PCA to represent it as a single value along the eigenvector with the largest eigenvalue.

Experimenting with PCA for Data Compression in 2 dimensions

QEA1 - Olin College

In general, PCA is conducted on data that is mean-centered (i.e., the data has had the mean of each variable subtracted out). We use this mean-centered data to create the covariance matrix in previous examples.

Let's think about a simple example data set D.

```
D = [-1 3; 1 4; 3 4; 7 5; 10 9]
```

Plot D on the x-y plane. Use plot(x-vals, y-vals, 'x','LineWidth',2) to plot these 4 data points as the x symbol that is easy to see.

```
figure;
% your code goes here.

xlabel('x'); ylabel('y'); axis square; axis ([-3 12 -3 12])
```

Now let's subtract the mean from each column of data. Here's a little trick to do it for each column all at once.

First, find the mean of the columns... what size should this be?

Think, then remove the ; to check

```
mean_D = mean(D,1); % mean of D along first dimension
```

If you change the number above to 2, it will take the mean the other way (giving a vector of size 5).

Make a mean-subtracted version of D. Matlab kindly assumes that you mean to subtract the appropriate mean from each value.

```
D_mean_sub = D - mean_D
```

Now, we can plot the mean-subtracted D on the x-y plane.

```
figure;
plot(D_mean_sub(:,1),D_mean_sub(:,2),'x','LineWidth',2);
xlabel('x'); ylabel('y'); axis square;  axis([-6 8 -6 8]);
```

What's different about these two plots?

Find the principal components

The principal components (p1 and p2) are the eigenvectors of the covariance matrix of the mean-centered D (D_mean_sub). Compute p1 and p2.

```
%cov_mat =
```

```
%[eigVec,eigVal]=
```

p1 should be the eigenvector with the larger eigenvalue.

```
%p1 =
%p2 =
```

This code will plot p1 and p2 on the mean-centered data.

```
figure;
plot(D_mean_sub(:,1),D_mean_sub(:,2),'x','LineWidth',2)
xlabel('x'); ylabel('y'); axis square; axis([-6 8 -6 8]);
hold on
quiver(0,0,p1(1),p1(2),"off",'LineWidth',2);
quiver(0,0,p2(1),p2(2),"off",'LineWidth',2);
hold off
```

Project data onto the principal eigenvector

Now we will compute the projection of your data onto the eigenvector which corresponds to the largest eignevalue, which we have set to be p1.

Projection of D_mean_sub onto p1

First, we will find how far along p1 each datapoint in D_mean_sub lands when projected onto p1. We'll call this alpha1.

```
alpha1 = D_mean_sub * p1
```

You may have noticed that this seems a lot like solving for $c = V^{-1}x$, It is! Because the eigenvectors are orthonormal is the same as $c = V^T x$. Since we're just looking at one principal component (eigenvector), this is like doing $c_1 = {v_1}^T x$ or $c_1 = x^T v_1$, by taking the transpose of both sides. We're calling things alpha1 here instead of c1 to avoid confusion with the c in covariance, but it's the same thing...just a coefficient along p1.

The variable alpha1 is the representation of each datapoint along p1.

```
figure;
plot(alpha1,zeros(length(alpha1),1),'.','MarkerSize',20);
xlabel('p1');axis([-8 8 0 1]);
```

Here, we have represented our data D_mean_sub as a single value along p1. Each point tells us how far to go along p1 to get as close as possible to the initial data point in the x-y plane. Now we want to see what this looks like in our regular x-y plane.

Represent our reduced dimensionalty data on the x-y plane

We now want to show the values of alpha1 on the x-y plane, so we need to use our eigenvector to transform it back to the x-y basis.

What dimensions should B have?

Discuss then remove the ; to check.

```
B = alpha1 * p1';
```

Plot the original data D and the reduced data B.

```
figure;
plot(D_mean_sub(:,1),D_mean_sub(:,2),'x','LineWidth',2)
xlabel('x'); ylabel('y'); axis square; axis([-6 8 -6 8]);
hold on
quiver(0,0,p1(1),p1(2),"off",'LineWidth',2);
quiver(0,0,p2(1),p2(2),"off",'LineWidth',2);
plot(B(:,1),B(:,2),'o','LineWidth',2)
hold off
legend({'Original','p1','p2','Reduced dim'},'Location','northwest');
```

Discuss the questions below at your table.

Can you recreate D perfectly from B?

...

What would have happened if you had created B using only information about the values along p2 instead of p1?

You can try it if you want to see what it looks like. Just copy the code from above and replace p1 with p2.

How might you quantify how well you can represent D in this reduced dimensionality form?

Discuss conceptually and try it if you have extra time.

```

```

If you received a new piece of data, how would you go about representing it as a linear combination of p1 and p2?

...

What would have happened if you had created B using some random vector? Think about different vectors you might project onto. Why is p1 usually the optimal vector to choose?

You can try it if you want to see what it looks like.

30.2.2 Thinking Through a PCA Application Conceptually [30 mins]

Now we are going to think through this PCA stuff conceptually as we apply the fundamental principles of PCA to movie reviews. In our next class, we'll dive deeper into the math.

Exercise 30.2

Consider a dataset consisting of ratings from m users of n movies. Let's assume that the ratings are numerical and are on a scale of 1 to 5 (5 being the best). Consider some collection of movies and a particular population of users (could be college students, QEA professors, or just the general population).

1. Draw the data matrix $\mathbf{A}$ and label the rows and columns (e.g., with movies or users). To make this concrete, identify (say) four different movies, and five hypothetical users.
2. Imagine that you wanted to calculate the movie-movie covariance matrix for your data matrix. What would that matrix look like? What does an element of that matrix tell you?
3. Imagine that you wanted to calculate the user-user covariance matrix. What might that look like? What does an element of *that* matrix tell you?
4. Let's say that all five users had similar ratings for movies C and D, but the ratings were all over the map for movies A and B. In a qualitative sense, make a prediction as to what the first principal component would look like for the movie-movie analysis of this dataset. That is, what would the direction be that maximizes the variance of the data projected onto that direction? What does this vector *mean* about movies or about users? Try this thought exercise for different possibilities of the data.
5. What might the second principal component look like? No numbers... just guess at which dimensions would be positive, negative, or close to 0 for your principal components.
6. Repeat the previous two questions for the user-user case. What would happen if one user gave every movie a rating of 3?

Summary: PCA at a High Level

- In many applications, being able to reduce dimensionality of data is extremely helpful as it can lead to efficient representations and reduced computational complexity.
- Given some data, PCA enables us to express multidimensional data as a linear combination of orthonormal vectors, starting with the vector in the direction with most variation in the data. The next vector will be in the direction with most variation of all directions orthogonal to the first, and so on.
- So, if we want to work with a lower-dimensional representation of our data, we can focus on those directions that contain the most variation.
- The eigenvectors of the covariance matrix of the data are the principal component vectors. The eigenvector corresponding to the largest eigenvalue lies in the direction with most variation in the data set. The eigenvector corresponding to the second largest eigenvalue lies in the direction with the next most variation, of all directions orthogonal to the first eigenvector, etc. We will explore the mathematics of PCA in the next class.