Lecture 22 - Notes

Rahul Arya

April 2019

1 Overview

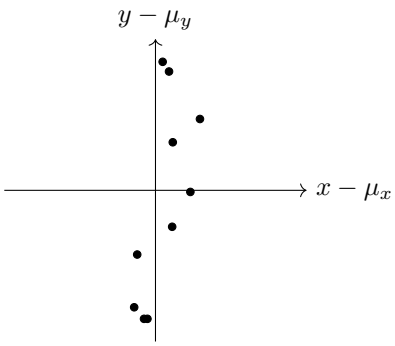
Last lecture, we began looking at statistical applications of linear algebra. In particular, given a dataset, we tasked ourselves with extracting the primary "features" from the data points, in a manner robust to noise. Last time, we saw how calculating the correlation matrix of a dataset provided some insight into the data. Now, we will look at a technique known as *principal component analysis*, which addresses a number of issues with the correlation matrix, in addition to providing a more mathematically elegant way of extracting meaning from noisy data.

2 Problems with Correlations

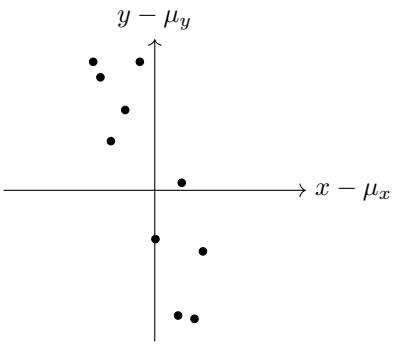
One technique we saw last time was the calculation of the correlation matrix, which expressed how any two observations correlated[1] with each other across all the data points. If the correlation was near 1, then the two observations behaved similarly, if it was near -1, the two observations behaved in opposite ways, and if it was about 0, there was no strong relationship between the two observations.

However, the correlation matrix wasn't ideal for the purposes of extracting meaning from noisy data. Consider the following two datasets, which have already been mean centered:

[1] Sorry, couldn't think of a synonym.

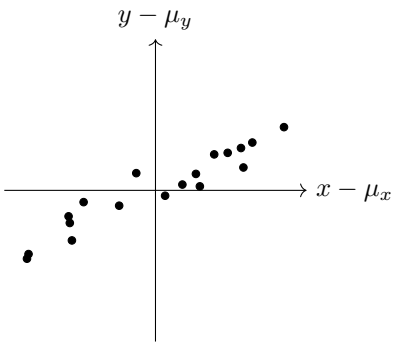


(a) Positive correlation.

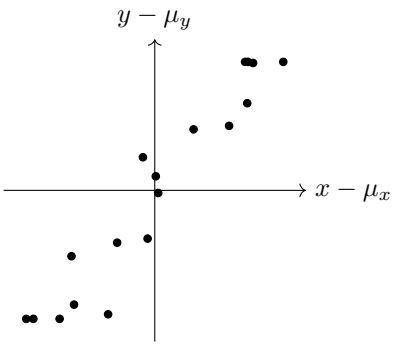


(b) Negative correlation.

Both datasets appear very similar, with y varying much more than x does. Despite that, in the first dataset, there is a strong positive correlation between x and y, while in the second, there is a strong negative correlation. So from the perspective of correlations, despite their visual similarities, these two datasets are in fact very different. Now, consider the next two datasets, again mean centered:



(a) Positive correlation.



(b) Negative correlation.

Both these datasets exhibit a strong positive correlation between their two variables - in other words, the correlation between x and y is very close to 1 in both. However, the datasets are clearly different, looking at the slope of the line of best fit passing through the points.

Clearly, the correlation matrix does not give us the full story, so something else is needed.

3 Principal Component Analysis

Let's briefly review notation. Let A be an $n \times m$ matrix of our data, with each data point forming a row of m components. We mean-center each column of A independently, to obtain the mean-centered data matrix $\widetilde{A}$. We then compute the covariance matrix S defined as

$$S = \frac{1}{n}\widetilde{A}^T\widetilde{A},$$

which represents how each pair of measurement columns from $\widetilde{A}$ align, with S_{ij} being large in magnitude if the ith and jth measurements behave similarly across all data points, and close to 0 if they are uncorrelated.

Last time, we took this intuitive understanding of covariance further by normalizing the entries of S, to obtain the correlation matrix R. Now, however, we've seen the problems with using R to try and extract trends and features of our data. Indeed, as we saw last time, in some cases it may be impossible to compute R, as it would involve dividing by zero.

Instead, we will stick with S, which can always be computed, and see what further properties can be obtained from it. Last time, we claimed that we should diagonalize S, to obtain

$$S = P\Lambda P^T.$$

We will call the columns of P our *principal components*, and assert that they can be used to obtain the key properties of our dataset. First, notice that as S is a real symmetric matrix, the real spectral theorem tells us that such a diagonalization will always exist, with the columns of P being orthogonal. Moreover, recall that we showed in a previous lecture that all the eigenvalues of $\widetilde{A}^T\widetilde{A}$ will be nonnegative, so all the entries of Λ will be nonnegative as well. Without loss of generality, we can order our eigenvectors such that the eigenvalues in Λ are in descending order from left to right. Consequently, writing

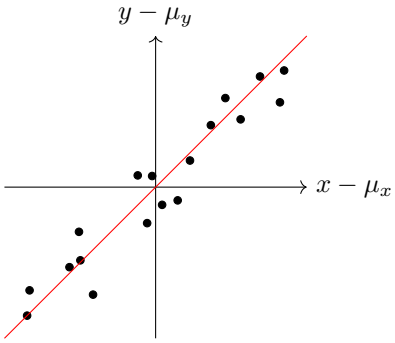
$$P = \begin{bmatrix} | & & | \\ \vec{p}_1 & \cdots & \vec{p}_m \\ | & & | \end{bmatrix},$$

we have that the eigenvalues corresponding to $\vec{p}_1, \vec{p}_2, \ldots$ will be in descending order.

4 Maximizing Variance

Now, what's so important about these principal component vectors? We claim that the $\vec{p}_i$ describe the "most important" directions of our dataset, in descending order of importance - in particular, we claim that $\vec{p}_1$ is the direction that contains the "most" information about our data.

What does it mean for a direction to be important? Intuitively speaking, we should expect our data to be "most aligned" along such a direction, as shown below, with the red line representing what we might imagine to be an "important direction":



Notice that the data varies greatly along this line, but does not vary as much away from it. In other words, we will define the "most important" direction to be the direction along which our data points, when projected onto this direction, have the *maximum variance*.

Let's prove that our first principal component $\vec{p}_1$ is indeed along this direction. Consider some arbitrary direction, represented by a unit vector $\vec{v}$. Observe that the variance of our dataset does not change with translation, so we can work entirely with our mean-centered data matrix $\widetilde{A}$. Let each data point within $\widetilde{A}$ be $\vec{a}_i$, so we have

$$\widetilde{A} = \begin{bmatrix} - & \vec{a}_1^T & - \\ - & \vec{a}_2^T & - \\ & \vdots & \\ - & \vec{a}_n^T & - \end{bmatrix}.$$

We need to first project each of the $\vec{a}_i$ onto $\vec{v}$. Since $\vec{v}$ is of unit magnitude, each scalar projection will simply be

$$\text{proj}_{\vec{v}}(\vec{a}_i) = \vec{a}_i^T \vec{v},$$

so we can stack our scalar projections in a vector

$$\begin{bmatrix} \text{proj}_{\vec{v}}(\vec{a}_1) \\ \text{proj}_{\vec{v}}(\vec{a}_2) \\ \vdots \\ \text{proj}_{\vec{v}}(\vec{a}_n) \end{bmatrix} = \widetilde{A}\vec{v}.$$

By definition, observe that the variance of these projections is

$$\frac{1}{n}\sqrt{(\text{proj}_{\vec{v}}(\vec{a}_1))^2 + (\text{proj}_{\vec{v}}(\vec{a}_2))^2 + \ldots + (\text{proj}_{\vec{v}}(\vec{a}_n))^2} = \frac{1}{n}\left\|\widetilde{A}\vec{v}\right\|.$$

Now, our problem reduces to determining the unit vector $\vec{v}$ that maximizes this quantity. Recall that the $\vec{p}_i$ formed an orthonormal basis for our m-dimensional space, so we may write

$$\vec{v} = \alpha_1\vec{p}_1 + \alpha_2\vec{p}_2 + \ldots + \alpha_n\vec{p}_m,$$

where

$$1 = \|\vec{v}\| = \sqrt{\alpha_1^2 + \alpha_2^2 + \ldots + \alpha_m^2}$$

for suitable chosen constants α_i.

Now, it should be clear that maximizing $\frac{1}{n}\left\|\widetilde{A}\vec{v}\right\|$ is the same thing as maximizing $\left\|\widetilde{A}\vec{v}\right\|^2 = \vec{v}^T\widetilde{A}^T\widetilde{A}\vec{v}$, since they are both always nonnegative. Since the $\vec{p}_i$ are eigenvectors, we can write

$$\begin{aligned}\vec{v}^T\widetilde{A}^T\widetilde{A}\vec{v} &= (\alpha_1\vec{p}_1 + \alpha_2\vec{p}_2 + \ldots + \alpha_n\vec{p}_n)^T\widetilde{A}^T\widetilde{A}(\alpha_1\vec{p}_1 + \alpha_2\vec{p}_2 + \ldots + \alpha_m\vec{p}_m)\\ &= (\alpha_1\vec{p}_1 + \alpha_2\vec{p}_2 + \ldots + \alpha_n\vec{p}_m)^T(\alpha_1\lambda_1\vec{p}_1 + \alpha_2\lambda_2\vec{p}_2 + \ldots + \alpha_m\lambda_m\vec{p}_m),\end{aligned}$$

where the λ_i are the eigenvalues associated with the $\vec{p}_i$. Taking advantage of the fact that the $\vec{p}_i$ are all orthogonal to one another, we can continue our above calculation to see that

$$\begin{aligned}\vec{v}^T\widetilde{A}^T\widetilde{A}\vec{v} &= (\alpha_1\vec{p}_1 + \alpha_2\vec{p}_2 + \ldots + \alpha_n\vec{p}_m)^T(\alpha_1\lambda_1\vec{p}_1 + \alpha_2\lambda_2\vec{p}_2 + \ldots + \alpha_m\lambda_m\vec{p}_m)\\ &= \alpha_1^2\lambda_1 + \alpha_2^2\lambda_2 + \ldots + \alpha_n^2\lambda_m.\end{aligned}$$

We wish to choose our α_i to maximize the above quantity, while keeping the sum of their squares equal to 1. For notational convenience, let $\beta_i = \alpha_i^2$. From the previous equation, and since $\vec{v}$ is a unit vector, we therefore have

$$\begin{aligned}\beta_1 + \beta_2 + \ldots + \beta_m &= 1\\ \vec{v}^TA^TA\vec{v} &= \beta_1\lambda_1 + \beta_2\lambda_2 + \ldots + \beta_m\lambda_m.\end{aligned}$$

Recall that we chose our diagonalization, using the properties of S, such that

$$\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_m \geq 0.$$

Since all the β_i are nonnegative (since they are the squares of real numbers), it is obvious (and can be easily shown using a simple exchange argument) that $\vec{v}^T\widetilde{A}^T\widetilde{A}\vec{v}$ has a maximum value of λ_1, that can be achieved by setting $\beta_1 = \alpha_1 = 1$ and the other $\beta_i = \alpha_i = 0$. Thus, we see that our optimal value

$$\vec{v} = 1\vec{p}_1 + 0\vec{p}_2 + \ldots + 0\vec{p}_m = \vec{p}_1,$$

as expected. Thus, we have shown that the first principal component (i.e. the one with the largest eigenvalue) corresponds to the direction that maximizes the variance of the data, when projected onto it.

As it turns out, we can go one step further. Often, simply looking at the projection of a dataset onto a single direction is not sufficient, even if this direction is the first principal component. It would be nice to obtain a k-dimensional subspace that maximizes the variance of the data points projected onto it. Although we will not prove it here (as it will be a homework problem), it turns out that this subspace is exactly the subspace formed by taking linear combinations of the first k principal components.

5 Zero Covariance

One important use for these principal components is known as *feature extraction*. That is to say, given an m-dimensional dataset where m is large, we may wish to extract k scalar features that represent the most interesting aspects of our dataset. In the previous section, we saw that PCA allowed us to pick k orthogonal scalar features that each maximized the variance of the data points projected onto each of the k principal components. However, having more features is of little use if they convey no additional information - that is to say, if a feature are strongly correlated with some others, then it is less useful, since we can use these other features to form a good estimate of the new feature.

As it turns out, the features obtained by projecting each datapoint onto the principal component directions are highly uncorrelated - in fact, the correlation between the data projected onto any pair of principal components is zero! Let's try to show this. Let $\hat{A}$ be the mean-centered data matrix expressed in the principal component basis, so we have

$$\widetilde{A} = \hat{A}P \implies \hat{A} = \widetilde{A}P^T.$$

Now, we can compute the covariance matrix $\hat{S}$ in the standard manner, to obtain

$$\begin{aligned}
\hat{S} &= \frac{1}{n}\hat{A}^T\hat{A} \\
&= \frac{1}{n}(\widetilde{A}P^T)^T(\widetilde{A}P^T) \\
&= \frac{1}{n}P\widetilde{A}^T\widetilde{A}P^T \\
&= \frac{1}{n}PSP^T
\end{aligned}$$

Since, by their construction, our principal components are the eigenvectors of S, we have that

$$\hat{S} = \frac{1}{n}PSP^T = \frac{1}{n}\Lambda,$$

which is clearly a diagonal matrix. Thus, the covariances of any two features in the principal component basis are 0, so their correlations are therefore also 0.

6 Relationship to the SVD

Finally, we will look at how the principal components of a matrix relate to its SVD - some sort of relationship may have already been hinted at in our earlier calculations, since diagonalizing $\widetilde{A}^T\widetilde{A}$ is a key step both in computing the principal components and the SVD. Let's formalize this connection, by considering the SVD

$$\widetilde{A} = U\Sigma V^T.$$

Observe that our covariance matrix

$$\begin{aligned} S &= \frac{1}{n} V\Sigma^T U^T U\Sigma V^T \\ &= \frac{1}{n} V\Sigma^T \Sigma V^T \\ &= V\left(\frac{1}{n}\Sigma^T\Sigma\right)V^T, \end{aligned}$$

which is a diagonalization of the covariance matrix. Thus, it is clear that the columns of V are our principal components, and we can relate the eigenvalues of the principal components to the singular values as follows:

$$\lambda_i = \frac{1}{n}\sigma_i^2$$

for $1 \leq i \leq m$. This process can clearly be reversed, starting from the principal components to obtain the singular value decomposition of a mean-centered data matrix. Thus, we have obtained a straightforward relationship between the SVD and PCA.