Mathematics and Interpretation of Principal Components

36-350: Data Mining

24 September 2008

1 Mathematics of Principal Components

There are several ways of deriving the principal components mathematically. The simplest one, as I mentioned last time, is by finding the projection which maximizes the variance. It might sound more plausible to look for the projection with the smallest average (mean-squared) distance between the original vectors and their projections on to the principal components; this turns out to be equivalent to maximizing the variance.

Throughout, assume that the data have been "centered", so that every feature has mean 0. If we write the standardized data in a matrix $\mathbf{X}$, where rows are objects and columns are features, then $X^T X = n\mathbf{V}$, where $\mathbf{V}$ is the covariance matrix of the data. (You should check that last statement!)

1.1 Minimizing Projection Residuals

We'll start by looking for a one-dimensional projection. That is, we have p-dimensional feature vectors, and we want to project them on to a line through the origin. We can specify the line by a unit vector along it, $\vec{w}$, and then the projection of a data vector $\vec{x_i}$ on to the line is $\vec{x_i} \cdot \vec{w}$, which is a scalar. (Sanity check: this gives us the right answer when we project on to one of the coordinate axes.) This is the distance of the projection from the origin; the actual coordinate in p-dimensional space is $(\vec{x_i} \cdot \vec{w})\vec{w}$. The mean of the projections will be zero, because the mean of the vectors $\vec{x_i}$ is zero:

$$\frac{1}{n}\sum_{i=1}^{n}(\vec{x_i} \cdot \vec{w})\vec{w} = \left(\left(\frac{1}{n}\sum_{i=1}^{n} x_i\right) \cdot \vec{w}\right)\vec{w} \tag{1}$$

If we try to use our projected or **image** vectors instead of our original vectors, there will be some error, because (in general) the images do not coincide with the original vectors. (When do they coincide?) The difference is the error or **residual** of the projection. How big is it? For any one vector, say $\vec{x_i}$, it's

$$\begin{aligned} \|\vec{x_i} - (\vec{w} \cdot \vec{x_i})\vec{w}\|^2 &= \|\vec{x_i}\|^2 - 2(\vec{w} \cdot \vec{x_i})(\vec{w} \cdot \vec{x_i}) + \|\vec{w}\|^2 & (2) \\ &= \|\vec{x_i}\|^2 - 2(\vec{w} \cdot \vec{x_i})^2 + 1 & (3) \end{aligned}$$

(This is the same trick used to compute distance matrices in the solution to the first homework; it's really just the Pythagorean theorem.) Add those residuals up across all the vectors:

$$\begin{aligned} RSS(\vec{w}) &= \sum_{i=1}^{n} \|\vec{x_i}\|^2 - 2(\vec{w}\cdot\vec{x_i})^2 + 1 && (4) \\ &= \left(n + \sum_{i=1}^{n} \|\vec{x_i}\|^2\right) - 2\sum_{i=1}^{n} (\vec{w}\cdot\vec{x_i})^2 && (5) \end{aligned}$$

The term in the big parenthesis doesn't depend on $\vec{w}$, so it doesn't matter for trying to minimize the residual sum-of-squares. To make RSS small, what we must do is make the term we subtract from it big, i.e., we want to maximize

$$\sum_{i=1}^{n} (\vec{w}\cdot\vec{x_i})^2$$

Equivalently, since n doesn't depend on $\vec{w}$, we want to maximize

$$\frac{1}{n}\sum_{i=1}^{n} (\vec{w}\cdot\vec{x_i})^2$$

which we can see is the sample mean of $(\vec{w}\cdot\vec{x_i})^2$. The mean of a square is always equal to the square of the mean plus the variance:

$$\frac{1}{n}\sum_{i=1}^{n} (\vec{w}\cdot\vec{x_i})^2 = \left(\frac{1}{n}\vec{x_i}\cdot\vec{w}\right)^2 + \mathrm{Var}\left[\vec{w}\cdot\vec{x_i}\right] \quad (6)$$

Since we've just seen that the mean of the projections is zero, minimizing the residual sum of squares turns out to be equivalent to maximizing the variance of the projections.

(Of course in general we don't want to project on to just one vector, but on to multiple principal components. If those components are orthogonal and have the unit vectors $\vec{w_1}, \vec{w_2}, \ldots \vec{w_k}$, then the image of x_i is its projection into the space spanned by these vectors,

$$\sum_{j=1}^{k} (\vec{x_i}\cdot\vec{w_j})\vec{w_j}$$

The mean of the projection on to each component is still zero. If we go through the same algebra for the residual sum of squares, it turns out that the cross-terms between different components all cancel out, and we are left with trying to maximize the sum of the variances of the projections on to the components. (Exercise: Do this algebra.)

1.2 Maximizing Variance

Accordingly, let's maximize the variance! Writing out all the summations grows tedious, so let's do our algebra in matrix form. If we stack our n data vectors into an $n \times p$ matrix, $\mathbf{X}$, then the projections are given by $\mathbf{Xw}$, which is an $n \times 1$ matrix. The variance is

$$\begin{aligned}
\sigma^2_{\vec{w}} &= \frac{1}{n}\sum_i \left(\vec{x_i} \cdot \vec{w}\right)^2 && (7)\\
&= \frac{1}{n}\left(\mathbf{Xw}\right)^T\left(\mathbf{Xw}\right) && (8)\\
&= \frac{1}{n}\mathbf{w}^T\mathbf{X}^T\mathbf{Xw} && (9)\\
&= \mathbf{w}^T\frac{\mathbf{X}^T\mathbf{X}}{n}\mathbf{w} && (10)\\
&= \mathbf{w}^T\mathbf{Vw} && (11)
\end{aligned}$$

We want to chose a unit vector $\vec{w}$ so as to maximize $\sigma^2_{\vec{w}}$. To do this, we need to make sure that we only look at unit vectors — we need to **constrain** the maximization. The constraint is that $\vec{w} \cdot \vec{w} = 1$, or $\mathbf{w}^T w = 1$. This needs a brief excursion into constrained optimization.

We start with a function $f(w)$ that we want to maximize. (Here, that function is $\mathbf{w}^T V \mathbf{w}$.) We also have an equality constraint, $g(w) = c$. (Here, $g(w) = \mathbf{w}^T\mathbf{w}$ and $c = 1$.) We re-arrange the constraint equation so its right-hand side is zero, $g(w) - c = 0$. We now add an extra variable to the problem, the **Lagrange multiplier** λ, and consider $u(w, \lambda) = f(w) + \lambda(g(w) - c)$. This is our new objective function, so we differentiate with respect to both arguments and set the derivatives equal to zero:

$$\begin{aligned}
\frac{\partial u}{\partial w} &= 0 = \frac{\partial f}{\partial w} + \lambda\frac{\partial g}{\partial w} && (12)\\
\frac{\partial u}{\partial \lambda} &= 0 = g(w) - c && (13)
\end{aligned}$$

That is, maximizing with respect to λ gives us back our constraint equation, $g(w) = c$. At the same time, when we have the constraint satisfied, our new objective function is the same as the old one. (If we had more than one constraint, we would just need more Lagrange multipliers.)[1]

For our projection problem,

$$\begin{aligned}
u &= \mathbf{w}^T\mathbf{Vw} - \lambda(\mathbf{w}^T\mathbf{w} - 1) && (14)\\
\frac{\partial u}{\partial \mathbf{w}} &= 2\mathbf{Vw} - 2\lambda\mathbf{w} = 0 && (15)\\
\mathbf{Vw} &= \lambda\mathbf{w} && (16)
\end{aligned}$$

[1] To learn more about Lagrange multipliers, read Boas (1983) or (more compactly) Klein (2001).

Thus, desired vector $\mathbf{w}$ is an **eigenvector** of the covariance matrix $\mathbf{V}$, and the maximizing vector will be the one associated with the largest **eigenvalue** λ. This is good news, because finding eigenvectors is something which can be done comparatively rapidly (see *Principles of Data Mining* p. 81), and because eigenvectors have many nice mathematical properties.

$\mathbf{V}$ is a $p \times p$ matrix, so it will have p different eigenvectors. $\mathbf{V}$ is a covariance matrix, so it is symmetric, and linear algebra tells us that the eigenvectors must be orthogonal to one another. The second principal component, remember, is the direction with the most variance which is orthogonal to the first principal component. Thus, the second principal compoent will be the eigenvector of $\mathbf{V}$ corresponding to the second largest eigenvalue, and so on. Because it is orthogonal to the first eigenvector, their projections will be uncorrelated. In fact, projections on to all the principal components are uncorrelated with each other. If we use k principal components, our weight matrix $\mathbf{w}$ will be a $p \times k$ matrix, where each column will be a different eigenvector of the covariance matrix $\mathbf{V}$. The eigenvalues will give the share of the total variance described by each component.

Back to the residuals The fraction of the total variance accounted for by the first k principal components is the R^2 of the projection (just like with a regression). The relative magnitude of the residuals then is $1 - R^2$.

2 Interpreting a PCA Plot

Last time, I drew a projection plot where, in addition to the data, I projected unit vectors along each of the original features (Figure 1). This helped us guess at what the principal components meant, and also told us how changing the attribute values will change the projections.

We can also do this in reverse. If we take a projected point, we can estimate its attribute values by looking at its position along the arrows. (That is, we can find the image of the projected point from the arrows.) This estimate will be good if R^2 is large. Similarly, the angles between the arrows give us an estimate of the correlation between features. If the angle is θ, then the correlation is roughly $\cos\theta$. This is exact when $R^2 = 1$, and gets worse as R^2 gets smaller.

2.1 A Recipe

We can now pull everything together to give a short recipe for how to interpret a PCA plot.

To begin with, find the first two principal components of your data. (I say "two" only because that's what you can plot; see below.) It's generally a good idea to standardized all the features first, but not strictly necessary.

Coordinates Using the arrows, summarize what each coordinate (h_1 and h_2) means. For the cars data, h_1 indicates something like "overall size" and h_2 something like "sporty".

Correlations For many datasets, the arrows cluster into groups of highly correlated attributes. Describe these attributes. Also determine the overall level of correlation (given by the R^2 value).

Clusters Clusters indicate a preference for particular combinations of attribute values. Summarize each cluster by its prototypical member. For the cars data, the vans form a cluster.

Funnels Funnels are wide at one end and narrow at the other. They happen when one dimension affects the variance of another, orthogonal dimension. Thus, even though the components are uncorrelated (because they are perpendicular) they still affect each other. (They are uncorrelated but not *independent*.) The cars data has a funnel, showing that small cars are similar in sportiness, while large cars are more varied.

Voids Voids are areas inside the range of the data which are unusually unpopulated. A **permutation plot** is a good way to spot voids. (Randomly permute the data in each column, and see if any new areas become occupied.) For the cars data, there is a void of sporty cars which are very small or very large. This suggests that such cars are undesirable or difficult to make.

Projections on to the first two or three principal components can be visualized; however they may not be enough to really give a good summary of the data. Usually, to get an R^2 of 1, you need to use all p principal components.[2] How many principal components you should use depends on your data, and how big an R^2 you need. In some fields, you can get better than 80% of the variance described with just two or three components. A sometimes-useful device is to plot $1 - R^2$ versus the number of components, and keep extending the curve it until it flattens out.

3 PCA Cautions

Trying to guess at what the components might mean is a good idea, but like many god ideas it's easy to go overboard. Specifically, once you attach an idea in your mind to a component, and especially once you attach a *name* to it, it's very easy to forget that those are names and ideas you made up; to **reify** them, as you might reify clusters. Sometimes the components actually do measure real variables, but sometimes they just reflect patterns of covariance which have many different causes. If I did a PCA of the same features but for, say, 2007 cars, I might well get a similar first component, but the second component would probably be rather different, since SUVs are now common but don't really fit along the sports car/mini-van axis.

[2] The exception is when some of your features are linear combinations of the others, so that you don't really have p *different* features.

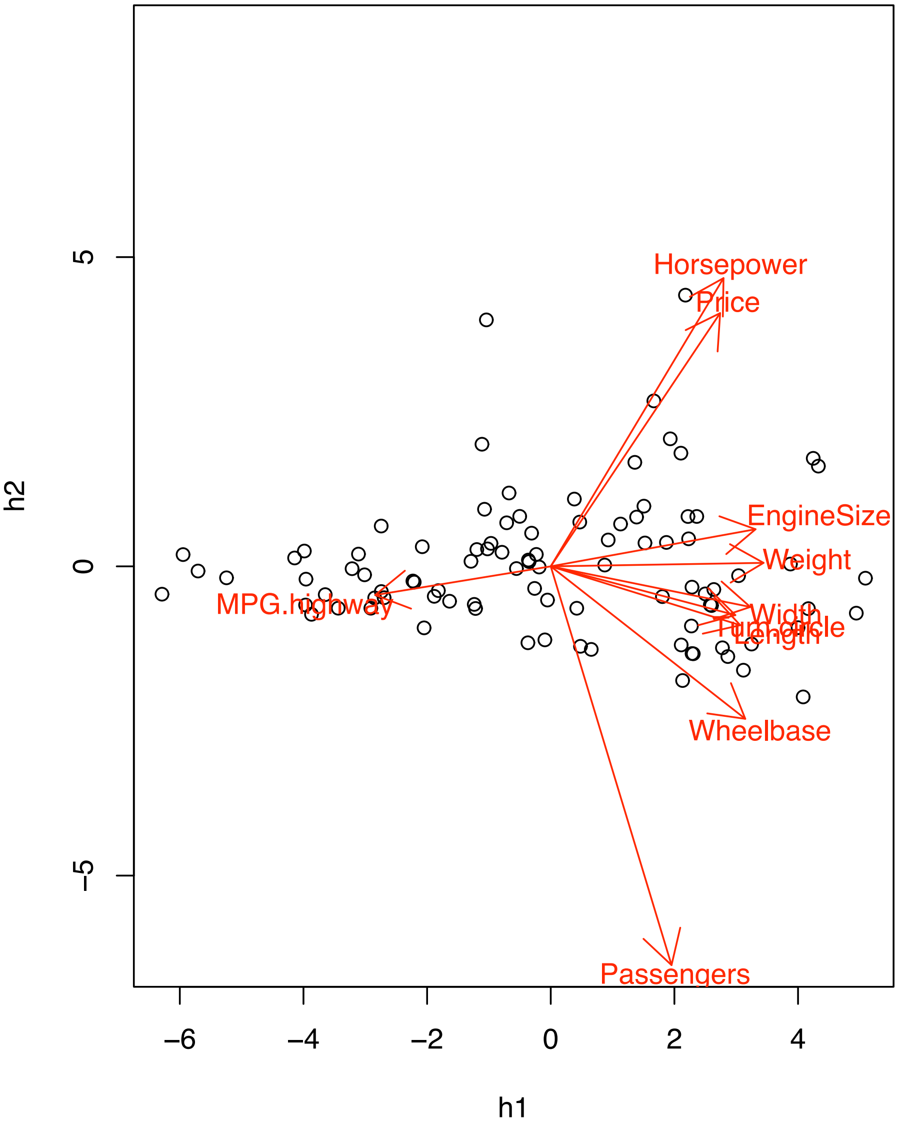

Figure 1: Projection of the cars data-vectors on to the first two principal components, and the loads of the original features. (Reproduced from last time for convenience.)

A more important example comes from population genetics. Starting thirty years ago, L. L. Cavalli-Sforza and collaborators began a huge project of mapping human genetic variation — of determining the frequencies of different genes in different populations throughout the world. (Cavalli-Sforza *et al.* (1994) is the main summary; Cavalli-Sforza has also written several excellent popularizations.) For each point in space, there are a very large number of features, which are the frequencies of the various genes among the people living there. Plotted over space, this gives a map of that gene's frequency. What they noticed (unsurprisingly) is that many genes had similar, but not identical, maps. This led them to use PCA, reducing the huge number of features (genes) to a few components. Results look like Figure 2. They interpreted these components, very reasonably, as signs of large population movements. The first principal component for Europe and the Near East, for example, was supposed to show the expansion of agriculture out of the Fertile Crescent. The third, centered in steppes just north of the Caucasus, was supposed to reflect the expansion of Indo-European speakers towards the end of the Bronze Age. Similar stories were told of other components elsewhere.

Unfortunately, as Novembre and Stephens (2008) showed, spatial patterns like this are what one should expect to get when doing PCA of any kind of spatial data with local correlations, because that essentially amounts to taking a Fourier transform, and picking out the low-frequency components.[3] They simulated genetic diffusion processes, without any migration or population expansion, and got results that looked very like the real maps (Figure 3). This doesn't mean that the stories of the maps *must be* wrong, but it does undercut the principal components as evidence for those stories.

References

Boas, Mary L. (1983). *Mathematical Methods in the Physical Sciences*. New York: Wiley, 2nd edn.

Cavalli-Sforza, L. L., P. Menozzi and A. Piazza (1994). *The History and Geography of Human Genes*. Princeton: Princeton University Press.

Klein, Dan (2001). "Lagrange Multipliers without Permanent Scarring." Online tutorial. URL `http://dbpubs.stanford.edu:8091/~klein/lagrange-multipliers.pdf`.

Novembre, John and Matthew Stephens (2008). "Interpreting principal component analyses of spatial population genetic variation." *Nature Genetics*, **40**: 646–649. doi:10.1038/ng.139.

[3] Remember that PCA re-writes the original vectors as a weighted sum of new, orthogonal vectors, just as Fourier transforms do. When there is a lot of spatial correlation, values at nearby points are similar, so the low-frequency modes will have a lot of amplitude, i.e., carry a lot of the variance. So first principal components will tend to be similar to the low-frequency Fourier modes.

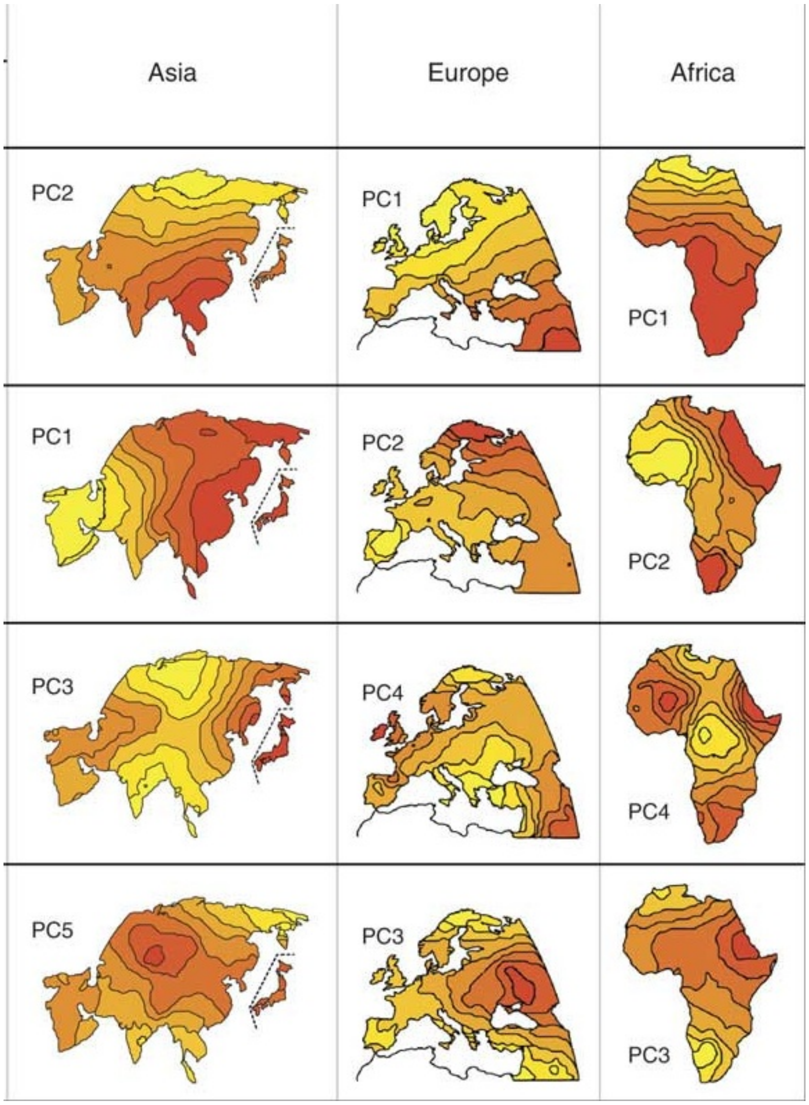

Figure 2: Principal components of genetic variation in the old world, according to Cavalli-Sforza *et al.* (1994), as re-drawn by Novembre and Stephens (2008).

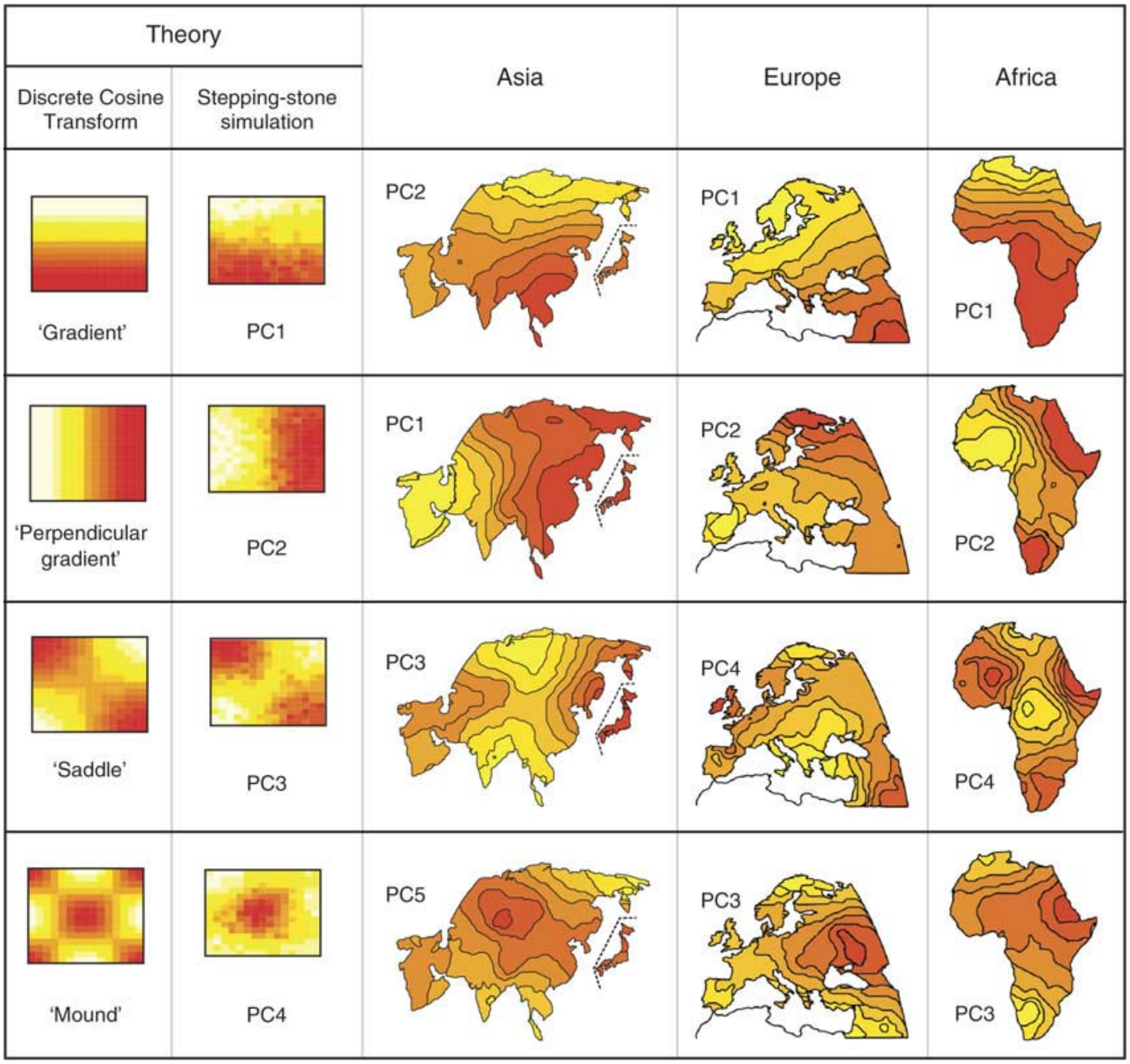

Figure 3: How the PCA patterns can arise as numerical artifacts (far left column) or through simple genetic diffusion (next column). From Novembre and Stephens (2008).