Lecture 8: Mathematics and Interpretation of Principal Components

36-350: Data Mining

October 2, 2006

Mathematics of Principal Components

There are several ways of deriving the principal components mathematically. The simplest one is by finding the projection which maximizes the variance. We could also do it by minimizing the information lost to projection, but that is more algebraically demanding.

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!)

Maximizing Variance

We want to project our p-dimensional feature vectors onto a one-dimensional line, which runs 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. (Check: this gives us the right answer when we project on to one of the coordinate axes.) 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 mean of the projections will be zero, because the mean of the $\vec{x_i}$ is zero. The variance is

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

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:

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

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.)

For our projection problem,

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

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, which will 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 — that is, it 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.

Minimizing Information Loss

When we project our p dimensional feature vectors on to a k dimensional surface, $\mathbf{X} \mapsto \mathbf{X}\mathbf{w}$, we lose some information, because multiple vectors can project to the same point. If we have a k-dimensional vector $\mathbf{h}$, we can try to undo the projection, getting $\mathbf{h}\mathbf{w}^T$ as the **image** in the original feature space. The difference between $\mathbf{X}$ and $\mathbf{h}\mathbf{w}^T$ is the **residual**. If we try to minimize the sum of the squared residuals for each point, we find that the projection which achieves this is PCA. 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 error is measured by $1 - R^2$.

Interpreting the arrows

Last time, I drew a projection plot where, in addition to the data, I projected unit vectors along each of the original features. This helped us figure out 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.

Interpreting a PCA Plot

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) is measuring. For the cars data, h_1 measures "size" and h_2 measures "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).

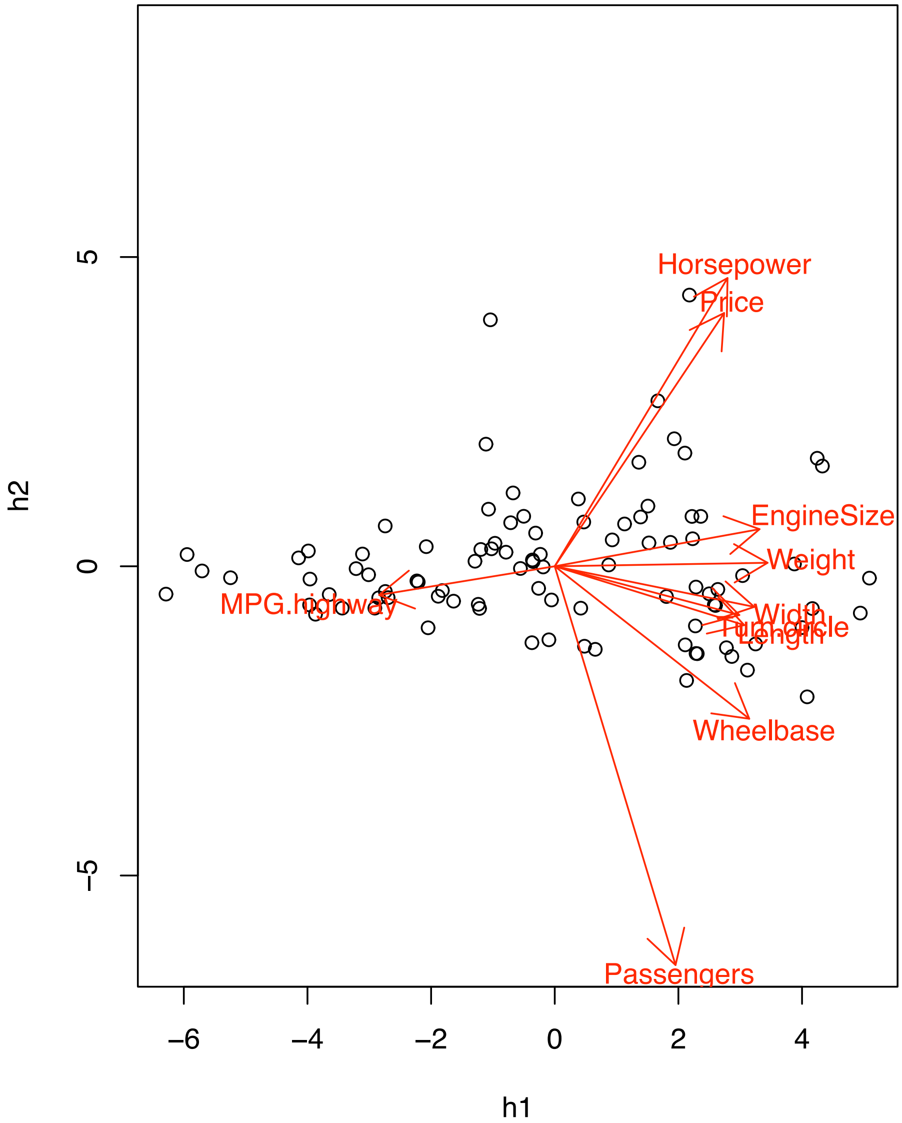
Clusters Clusters indicate a preference for particular combinations of attribute values. Summarize each cluster by its prototypical member. For the cars data, the vans form a cluster.

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 dimensions are uncorrelated (because they are perpendicular) they still affect each other. 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.[1] 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. (For instance, Congressional voting is *at best* two-dimensional.) A useful device is to plot $1 - R^2$ versus the number of components, and keep extending the curve it until it flattens out.

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


Horsepower
Price
EngineSize
Weight
Width
Turn.circle
Length
Wheelbase
Passengers
MPG.highway
h2
h1
5
0
−5
−6
−4
−2
0
2
4