Dimension reduction 2: Principal component analysis (continued)

Ryan Tibshirani
Data Mining: 36-462/36-662

February 7 2013

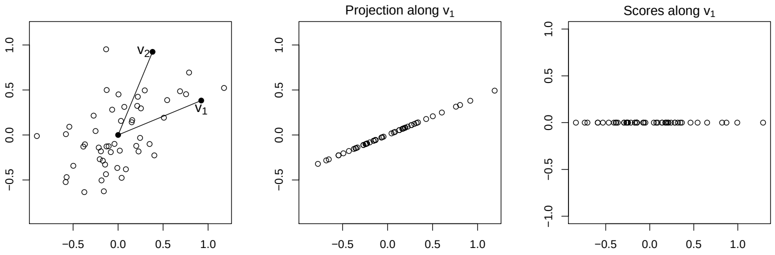
Optional reading: ISL 10.2, ESL 14.5

Reminder: projections onto unit vectors

The projection of $x \in \mathbb{R}^n$ onto a unit vector $v \in \mathbb{R}^n$ is given by $(x^T v)v \in \mathbb{R}^n$. The score from this projection is $x^T v \in \mathbb{R}$

The projections of the rows of $X \in \mathbb{R}^{n\times p}$ onto unit vector $v \in \mathbb{R}^p$ are given by rows of $Xvv^T \in \mathbb{R}^{n\times p}$. The scores are the entries of $Xv \in \mathbb{R}^n$

Example from last time: $X \in \mathbb{R}^{50\times 2}$, $v_1, v_2 \in \mathbb{R}^2$



Reminder: first principal component direction and score

Recall: given data matrix $X \in \mathbb{R}^{n \times p}$ (n observations, p features), with centered its columns

The first principal component direction of X is the unit vector $v_1 \in \mathbb{R}^p$ such that Xv_1 has the highest sample variance compared to all other unit vectors, i.e.,

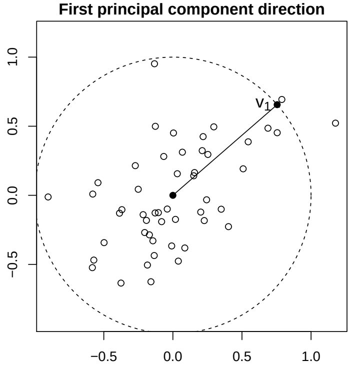
$$v_1 = \underset{\|v\|_2=1}{\operatorname{argmax}} \, (Xv)^T(Xv)$$

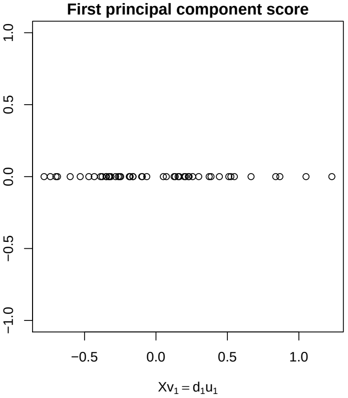
The vector $Xv_1 \in \mathbb{R}^n$ is called the first principal component score of X, and $u_1 = (Xv_1)/d_1 \in \mathbb{R}^n$ is the normalized first principal component score, where $d_1 = \sqrt{(Xv_1)^T(Xv_1)}$. The quantity d_1^2/n is the amount of variance explained by v_1

The entries of $Xv_1 = d_1 u_1$ are the scores from projecting X onto v_1, and the rows of $Xv_1 v_1^T = d_1 u_1 v_1^T$ are the projected vectors

Example: first principal component direction and score

Example from last time: $X \in \mathbb{R}^{50\times 2}$





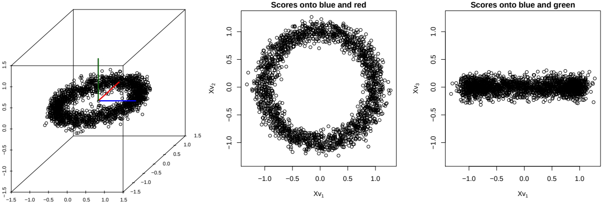
Reminder: projections onto orthonormal sets

Vectors $v_1, \ldots v_k \in \mathbb{R}^p$ are called orthonormal if each pair v_i, v_j is orthogonal, $v_i^T v_j = 0$, and each v_j has unit norm

The projection of $x \in \mathbb{R}^p$ onto an othonormal set $v_1, \ldots v_k \in \mathbb{R}^p$ is $\sum_{i=j}^k (x^T v_j) v_j \in \mathbb{R}^p$. The score along v_j is $x^T v_j$

The projections of rows of $X \in \mathbb{R}^{n \times p}$ onto orthonormal columns of $V \in \mathbb{R}^{p \times k}$ are given by rows of $XVV^T \in \mathbb{R}^{n \times p}$. The scores are columns of $XV \in \mathbb{R}^{n \times k}$, i.e., the scores along v_j are given by the entries of $Xv_j \in \mathbb{R}^n$

Example from last time: $X \in \mathbb{R}^{2000 \times 3}$



Further principal component directions and scores

Given first $k-1$ principal component directions $v_1, \ldots v_{k-1} \in \mathbb{R}^p$ (these are orthonormal), the kth principal component direction $v_k \in \mathbb{R}^p$ is the unit vector such that Xv_k has the highest sample variance over all directions orthogonal to $v_1, \ldots v_{k-1}$, i.e.,

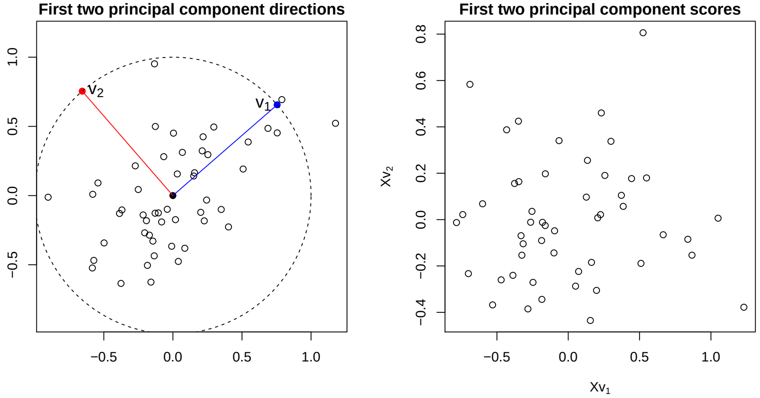
$$v_k = \underset{\substack{\|v\|_2=1 \\ v^T v_j = 0,\ j=1,\ldots k-1}}{\operatorname{argmax}} \quad (Xv)^T (Xv)$$

The vector $Xv_k \in \mathbb{R}^n$ is called the kth principal component score of X, and $u_k = (Xv_k)/d_k \in \mathbb{R}^n$ is the normalized kth principal component score, where $d_k = \sqrt{(Xv_k)^T(Xv_k)}$. The quantity d_k^2/n is the amount of variance explained by v_k

The entries of $Xv_k = d_k u_k$ are the scores from projecting X onto v_k, and the rows of $Xv_k v_k^T = d_k u_k v_k^T$ are the projected vectors

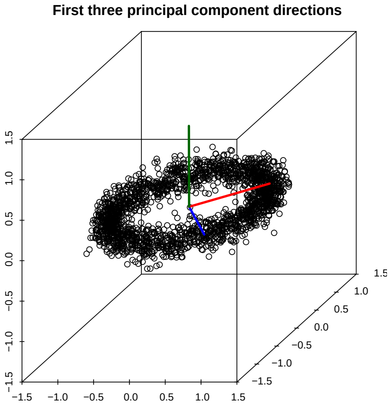
Example: second principal component direction and score

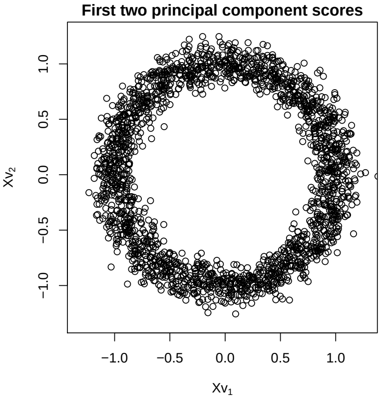
Same example as before: $X \in \mathbb{R}^{50\times 2}$



Example: principal component analysis in $\mathbb{R}^3$

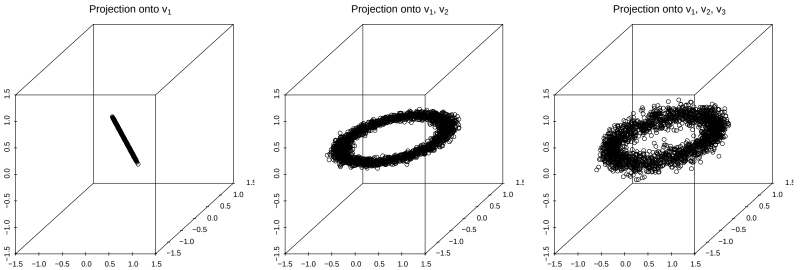
Example from last time: $X \in \mathbb{R}^{2000\times 3}$. Shown are the first three principal component directions $v_1, v_2, v_3 \in \mathbb{R}^3$, and the scores from projecting onto the first two directions





Example: projecting onto principal component directions

Same example. What happens if replace X by its projection onto v_1? Onto v_1, v_2? Onto v_1, v_2, v_3?



The third plot looks exactly the same as the original data. Is this a coincidence? No! (Why not?)

Example: principal component analysis in $\mathbb{R}^{12}$

Example: data from 2012 Cadillac Championship, professional golf tournament. Here $X \in \mathbb{R}^{72 \times 12}$, 72 golfers with 12 features:

```
eagles
birdies
pars
bogeys
double.bogeys
driving.accuracy
driving.distance
strokes.gained.putting
putts.per.round
putts.per.gir
greens.in.reg
sand.saves
```

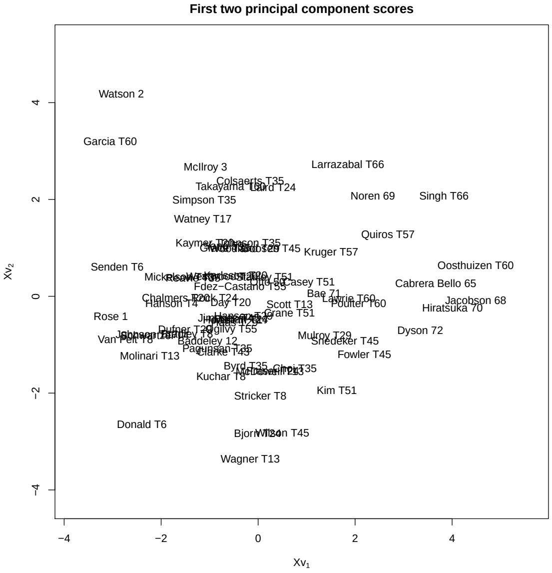
These are average measurements over the 4 day tournament

The first two principal component directions $v_1, v_2 \in \mathbb{R}^{12}$ are:

	PC1	PC2
eagles	-0.139	0.208
birdies	-0.463	0.185
pars	0.168	-0.582
bogeys	0.303	0.420
double.bogeys	0.062	0.181
driving.accuracy	-0.128	-0.241
driving.distance	-0.036	0.430
strokes.gained.putting	-0.438	-0.091
putts.per.round	0.325	0.026
putts.per.gir	0.491	-0.158
greens.in.reg	-0.171	-0.099
sand.saves	-0.238	-0.296

For each direction, look at the signs ... what do you notice here?

Scores from projecting onto $v_1, v_2 \in \mathbb{R}^{12}$:



Dimension reduction via the principal component scores

As we've seen in the examples, dimension reduction via principal component analysis can be achieved by taking the first k principal component scores $Xv_1, \ldots Xv_k \in \mathbb{R}^n$

We can think of $Xv_1, \ldots Xv_k$ as our new feature vectors, which is a big savings if $k \ll p$ (e.g. $k = 2$ or 3)

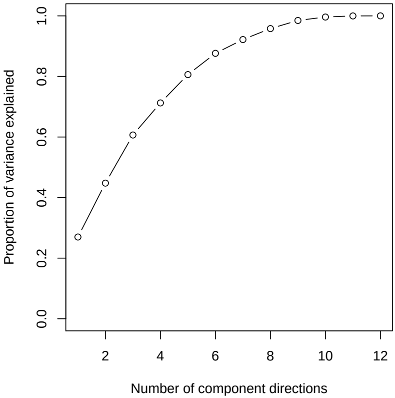
An important question: how good are these features at capturing the structure of our old features? Broken up into two questions:

1. How good are they, for a fixed k?
2. What exactly do we gain by increasing k?

Recall that the second question can be addressed by looking at the proportion of variance explained as a function of k

Example: proportion of variance explained

For the golf data set:



Approximation by projection

As for the first question, think about approximating X by $XV_kV_k^T$, the projection of X onto the first k principal component directions

An important alternate characterization of the principal component directions: given centered $X \in \mathbb{R}^{n\times p}$, if $V_k = [v_1 \ \ldots v_k] \in \mathbb{R}^{p\times k}$ is the matrix whose columns contain the first k principal component directions of X, then

$$XV_kV_k^T = \underset{\text{rank}(A)=k}{\text{argmin}} \; \|X - A\|_F^2 = \underset{\text{rank}(A)=k}{\text{argmin}} \; \sum_{i=1}^n \sum_{j=1}^p (X_{ij} - A_{ij})^2$$

In other words, $XV_kV_k^T$ is the best rank k approximation to X

(Aside: the above problem is nonconvex, and would be very hard to solve in general!)

Scaling the features

We always center the columns of X before computing the principal component directions. Another common pre-processing step is to scale the columns of X, i.e., to divide each feature by its sample variance, so that each feature in our new X has a sample variance of one

Why? Look at the first principal component score of the golf data, without scaling:

```
         eagles               birdies                pars
         -0.001                 0.007               0.007
         bogeys          double.bogeys    driving.accuracy
         -0.015                 0.002               0.071
driving.distance strokes.gained.putting    putts.per.round
         -0.122                 0.015              -0.016
   putts.per.gir          greens.in.reg          sand.saves
         -0.001                -0.004               0.990
```

And note that the golf features have sample variance:

```
          eagles                 birdies                pars
           0.033                   0.685               0.965
          bogeys           double.bogeys    driving.accuracy
           0.561                   0.095              59.837
driving.distance strokes.gained.putting      putts.per.round
         100.702                   0.739               1.263
   putts.per.gir           greens.in.reg          sand.saves
           0.006                  54.162             423.474
```

But sometimes scaling is not appropriate (e.g., when you know the variables are all on the same scale to begin with)

Computing principal component directions

There are various ways to compute principal component directions. We'll consider computation via the singular value decomposition (SVD) of X:

$$\underset{n \times p}{X} = \underset{n \times p}{U} \; \underset{p \times p}{D} \; \underset{p \times p}{V^T}$$

Here $D = \mathrm{diag}(d_1, \ldots d_p)$ is diagonal with $d_1 \geq \ldots \geq d_p \geq 0$, and U, V both have orthonormal columns. This gives us everything:

- columns of V, $v_1, \ldots v_p \in \mathbb{R}^p$, are the principal component directions
- columns of U, $u_1, \ldots u_p \in \mathbb{R}^n$, are the normalized principal component scores
- squaring the jth diagonal element of D and dividing by n, d_j^2/n, gives the variance explained by v_j

(Don't forget that we must first center the columns of X!)

Note that

$$XV = UDV^TV = UD$$

because $V^TV = I$. This means that

$$Xv_j = d_ju_j, \quad j = 1, \ldots p$$

two ways of representing principal component scores, as expected

Note also that

$$X^TX = VD^2V^T$$

and so $v_1, \ldots v_p$ are eigenvectors of X^TX. (Check?)

These two facts suggest another way of computing the principal component directions and scores

Recap: principal component analysis

We reviewed the principal component directions $v_1, \ldots v_p \in \mathbb{R}^p$ and scores $Xv_1, \ldots Xv_p \in \mathbb{R}^n$ of a centered matrix $X \in \mathbb{R}^{n \times p}$

The matrix $XV_k \in \mathbb{R}^{n \times k}$ (where V_k contains the first k principal component directions) can be thought of as a reduced dimension version of X

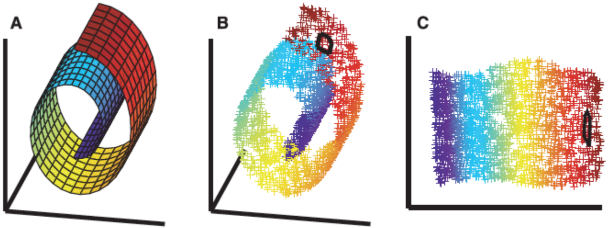
The matrix $XV_kV_k^T \in \mathbb{R}^{n \times p}$ (projecting X onto its first k principal component directions) can be thought of as an approximation to X in the original feature space. For a fixed k this approximation is the best we can do across rank k matrices (measured by Frobenius distance to X)

Scaling the variables can be crucial, if they are on different numeric scales

Computation can be done via the singular value decomposition

Next time: nonlinear dimension reduction

The famous "swiss roll" data set ...



(From Roweis et al. (2000), "Nonlinear dimensionality reduction by locally linear embedding")