Unsupervised Statistical Learning: Principal Components Analysis

Siva Balakrishnan
Data Mining: 36-462/36-662

March 19th, 2019

Recap: Unsupervised Learning

- In supervised learning we have (X, Y) pairs, and our goal was to predict/guess Y from X.
- In unsupervised learning we just observe $\{X_1, \ldots, X_n\}$ where $X_i \in \mathbb{R}^d$.
- We could imagine several possible tasks:
  1. **Dimension Reduction/Visualization:** Reduce the dimension of the data from d to something smaller (in a way that makes sense) so we can explore/visualize the data.
  2. **Clustering:** Group the n points into k groups (in a way that makes sense).
  3. **Density Estimation:** Estimate the underlying distribution of the data (in a way that makes sense).

Notice the goals and the metrics are much more varied.

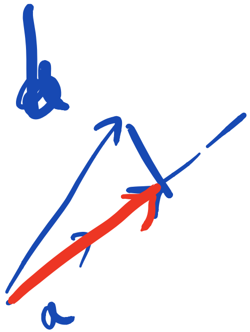
n pts in d dims,

n pts in 43 dims.

Recap: Linear Algebra Basics

- Vectors:

$$v = \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_d \end{bmatrix}.$$



- The length of a vector:

$$\|v\|_2 = \sqrt{v_1^2 + \ldots + v_d^2}.$$

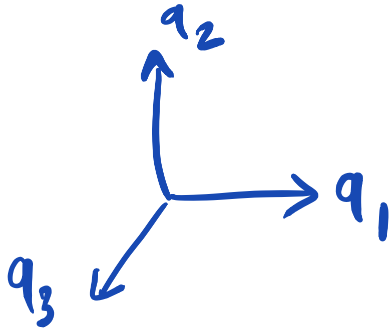
- The projection of a vector b onto a *unit* vector a:

$$\text{proj}_a(b) = \underline{(a^T b)a.}$$

$$\left(\frac{a^T b}{\|a\|} \times \frac{a}{\|a\|} \right)$$

Recap: Orthonormal Matrices

- Matrices $Q \in \underline{\mathbb{R}^{d \times d}}$:



$$Q = \begin{bmatrix} \uparrow & \uparrow & \cdots & \uparrow \\ q_1 & q_2 & \cdots & q_d \\ \downarrow & \downarrow & \cdots & \downarrow \end{bmatrix},$$

which satisfy:

$$q_i^T q_j = \begin{cases} 1 & \text{if} \quad i = j \\ 0 & \text{otherwise.} \end{cases}$$

- Orthonormal matrices satisfy:

$$\begin{aligned} Q^T Q &= I, \\ Q Q^T &= I, \\ Q^{-1} &= Q^T. \end{aligned}$$

Recap: Matrix Decompositions

- Every real, symmetric matrix M can be *diagonalized*, i.e. we can write:

$$M = U \times D \times U^T,$$

  for a *diagonal* matrix D, and an *orthonormal* matrix U.
- The columns of U are called *eigenvectors*, and each column of U has an associated diagonal entry in the matrix D are that is its associated *eigenvalue*.
- We will usually arrange things so that $|D_{11}| \geq |D_{22}| \geq \ldots$. Positive semi-definite matrices are ones for which every eigenvalue is ≥ 0.
- The eigendecomposition has many uses. Given the eigendecomposition you can easily invert the matrix, raise it to some power, compute the matrix exponential and so on.
- We will also see that it will give us crucial insight into important matrices.

Recap: Matrix Decompositions

- Every real matrix (not necessarily symmetric or even square) M can be written in terms of its *Singular Value Decomposition*:

U, V are both orthonormal

$$M = U \times \Sigma \times V^T,$$

for a *diagonal* matrix Σ with all positive entries, and two *orthnormal* matrices U, V.

- In particular, we can see that:

$$MM^T = U \times \Sigma^2 \times U^T$$
$$M^T M = V \times \Sigma^2 \times V^T.$$

So U and V are just the eigenvectors of MM^T and $M^T M$ (which are both symmetric matrices).

The Covariance Matrix

- We have talked about matrices abstractly so far. Let us now think about a particular important matrix. Remember, all we have is a data matrix $X \in \mathbb{R}^{n \times d}$.

$$X = \begin{bmatrix} \leftarrow x_1 \rightarrow \\ \leftarrow x_2 \rightarrow \\ \vdots \\ \leftarrow x_n \rightarrow \end{bmatrix}$$

- We will assume throughout the rest of the lecture that we have centered the matrix X so it has columns with mean 0 (so the mean of the data is the 0 vector).
- One thing that we can compute is the covariance matrix:

$$\widehat{\Sigma} = \frac{X^T X}{n}. \qquad \hat{\Sigma} \in \mathbb{R}^{d \times d}$$

The covariance matrix can also be written as:

$$\widehat{\Sigma} = \frac{1}{n} \sum_{i=1}^{n} x_i x_i^T.$$

where $x_i \in \mathbb{R}^d$ is the i-th data sample (the i-th row of X represented as a column vector).

The Covariance Matrix

$$\left(\frac{X^T X}{n}\right)^T = \frac{X^T X}{n}.$$

- The covariance matrix is symmetric (and real) and so has an eigendecomposition.
- It is also a positive semi-definite matrix. $\}$ all its eigenvalues are positive.
- Finally, observe that for any vector v we can compute:

v is some unit vector.

$$\text{salar} \leftarrow v^T \widehat{\Sigma} v = v^T \left[\frac{1}{n} \sum_{i=1}^{n} x_i x_i^T \right] v$$

$$= \frac{1}{n} \sum_{i=1}^{n} (v^T x_i)^2.$$

$\} \to$ variance of data in direction v.

This is just the *variance* of the data projected onto the direction v.

$$\hookrightarrow \frac{v^T \hat{\Sigma} v}{\|v\|^2}$$

$x_1, \ldots, x_n.$

$v^T x_1, \ldots, v^T x_n.$

$$\text{Variance} = \frac{1}{n} \sum_{i=1}^{n} \left[(v^T x_i)^2 \right] - \left(\frac{1}{n} \sum_{i=1}^{n} v^T x_i \right)^2$$

Back to Unsupervised Learning: What is Dimension Reduction?

Dimension reduction: the task of transforming our data set to one with fewer features. We want this transformation to preserve the main structure that is present in the feature space

A new feature can be one of the old features, or it can be a some linear or nonlinear combination of old features.

It is often the first step in an analysis, to be followed by, e.g., visualization, clustering, regression, classification

Linear dimension reduction

We're going to start with linear dimension reduction.

This means: looking for linear subspaces around which our data seem to concentrate.

Specifically, we'll be looking for subspaces which contain a large amount of the variance in the data. This is PCA.

- We hope that dimensions which contain lots of the variance are also interesting...

PCA Examples

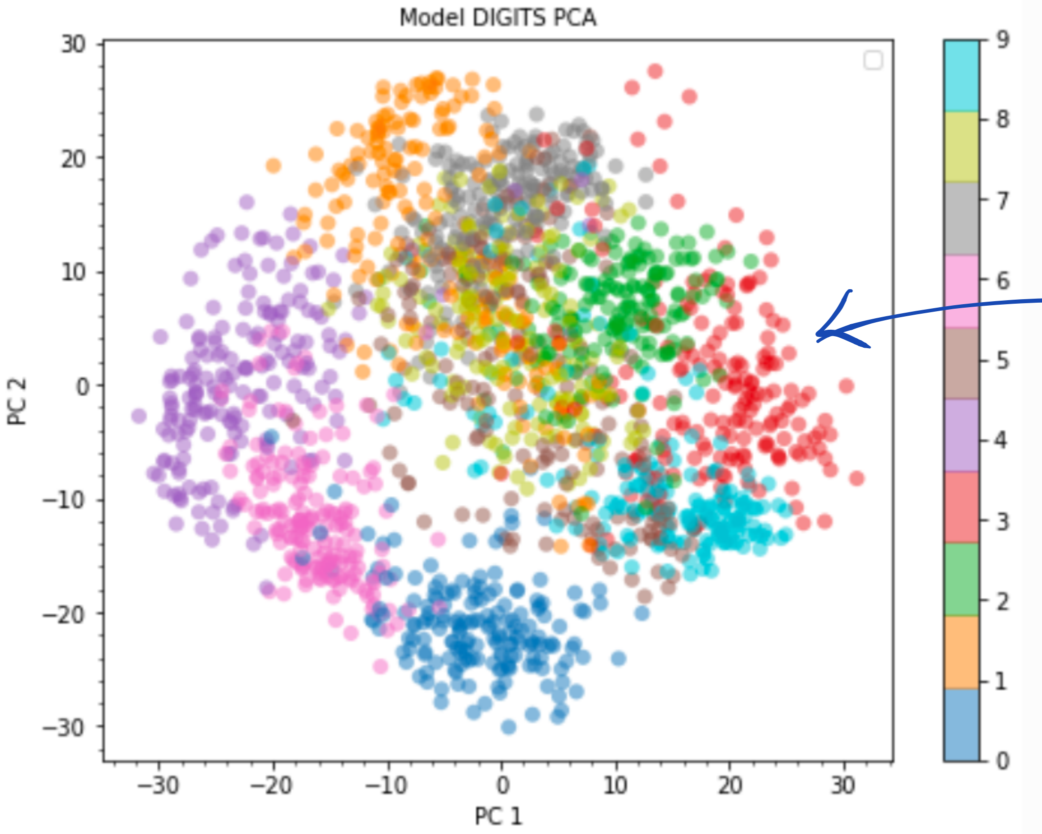
- Just to convince you that PCA is actually an interesting method here are a couple of examples: Suppose we took the MNIST digits dataset (a dataset of handwritten digits). Here is a small sample:

data pt

We want to understand/visualize the data but it is 800-dimensional and there are 50,000 points.

PCA on MNIST

Suppose we found two "interesting directions" and projected the data onto those two and plotted them.

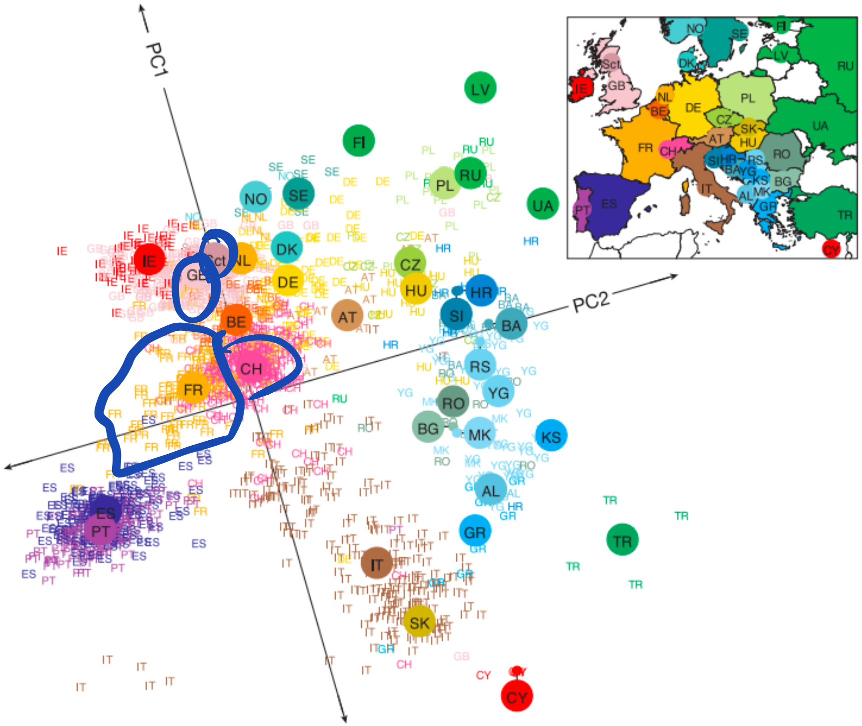


$X \in \mathbb{R}^{n \times d}$

$\tilde{X} \in \mathbb{R}^{n \times 2}$

Genes Mirror Geography

- This is data from about 3000 Europeans – for each of them we measure 0.5 million DNA sites. So our data matrix has 3000 points each in 0.5 million dimensions.

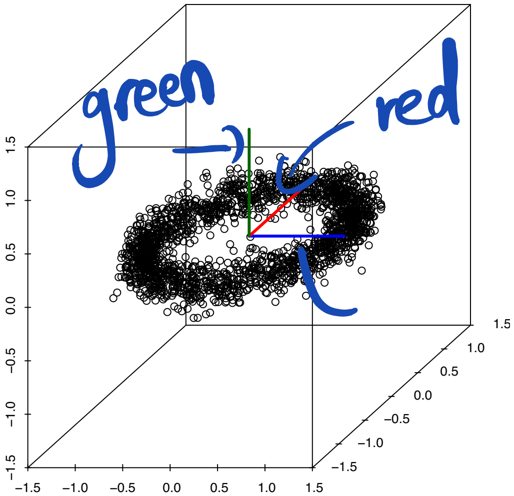


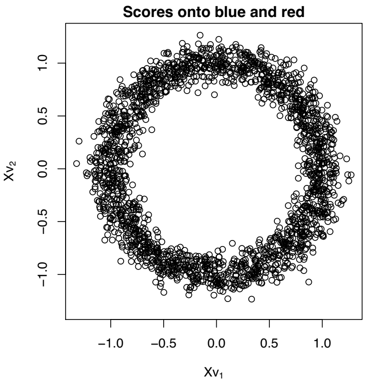
What are these interesting directions?

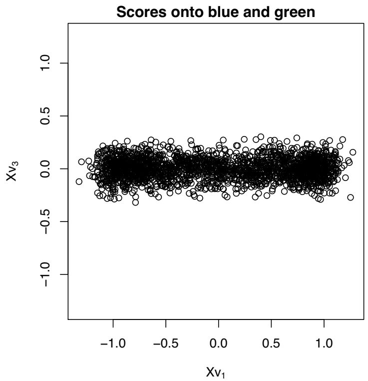
Example: Projections onto Orthonormal Vectors

Example: $X \in \mathbb{R}^{2000 \times 3}$, and $v_1, v_2, v_3 \in \mathbb{R}^3$ are the unit vectors parallel to the coordinate axes

green – nothing interesting







Not all linear projections are equal! What makes a good one?

Principal component analysis

v_1

The first principal component direction of X is the unit vector $v_1 \in \mathbb{R}^p$ that maximizes the sample variance of $Xv_1 \in \mathbb{R}^n$ when compared to all other unit vectors.

As we saw earlier the variance in direction v is just given by $v^T \widehat{\Sigma} v$. Hence the first principal component direction $v_1 \in \mathbb{R}^p$ is

$$v_1 = \underset{\|v\|_2=1}{\operatorname{argmax}} \; \underbrace{v^T \widehat{\Sigma} v}$$

variance in dir v.

We will call the variance in the direction v_1 as the amount of variance explained by v_1:

$$d_1^2 = v_1^T \widehat{\Sigma} v_1.$$

Xv_1

The vector $Xv_1 \in \mathbb{R}^n$ is called the first principal component score of X.

How do we think about this in terms of Eigenvectors and Eigenvalues?

- The top principal component is just the top eigenvector (i.e. with largest eigenvalue) of $\widehat{\Sigma}$.

→ v_1 is just leading EV of $\hat{\Sigma}$.

→ prove on HW.

- The ~~proportion~~ amount of variance explained is just the associated top eigenvalue.

Suppose v_1 is top EV of $\hat{\Sigma}$

$$v_1^T \hat{\Sigma} v_1 = v_1^T (EV) v_1 = (EV) v_1^T v_1 = EV.$$

$$\hat{\Sigma} v_1 = (EV) v_1$$

Further principal component directions and scores

Suppose want to define v_2.

Given the $k-1$ principal component directions $v_1, \ldots v_{k-1} \in \mathbb{R}^p$ (note that these are orthonormal), we define the kth principal component direction $v_k \in \mathbb{R}^p$ to be

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

maximize variance but $\perp$ to first one.
↳ perp.

The vector $Xv_k \in \mathbb{R}^n$ is called the kth principal component score of X.

The amount of variance explained by the k-th PC is:

$$d_k^2 = v_k^T \widehat{\Sigma} v_k.$$

2^{nd} PC is just 2^{nd} Evector of $\hat{\Sigma}$,
variance expl. is just 2^{nd} Evalue

How do we think about the PC scores?

Principal Component Scores

Suppose we computed the SVD of X:

$$X = U \times \widetilde{D} \times V^T,$$

principal comp scores

principal components

amount of variance explained.

where V is the collection of eigenvectors of the covariance matrix, and U are the eigenvectors of XX^T. So

$$Xv_1 = u_1\widetilde{d}_{11},$$

$$\vdots$$

$$Xv_k = u_k\widetilde{d}_{kk}.$$

$Xv_1 = \widetilde{d}_{11} \times u_1.$

So the PC scores are just given by the U matrix in the SVD of X. Furthermore, if we wanted the projection of X onto the principal component v_k we would use:

$$Xv_kv_k^T \in \mathbb{R}^{n\times p}.$$

$$ {}_n\overset{d}{X} = {}_n\overset{k}{U}{}_k\,\overset{k}{\tilde{D}}{}_k\,\underbrace{V^T}_{d} $$

mean 0.

each column is n-dimensional.

pricipal components

$$ \frac{X^TX}{n} = VDV^T $$

$$ XV = U\tilde{D} $$

$$ \tilde{X} = {}_n\overset{\times 2}{U}{}_2\,\overset{2\times}{\tilde{D}}{}_2\,\overset{d}{V^T} $$

Suppose we wanted to plot/reconstruct the original data in $\mathbb{R}^d$ (but on a 2-d subspace)

we would use $\tilde{X}$

$$n \; \overset{d}{X} \longrightarrow \begin{cases} v_1 \leftarrow d \times 1 \\ v_2 \leftarrow d \times 1. \end{cases}$$

$$\to \underbrace{Xv_1}_{n \times 1} \;\; \& \;\; \underbrace{Xv_2}_{n \times 1}.$$

$$n \; \underbrace{\overset{2}{\left(Xv_1 , \; Xv_2 \right)}}_{\text{this is the embedding we plotted earlier}}$$

Properties and representations

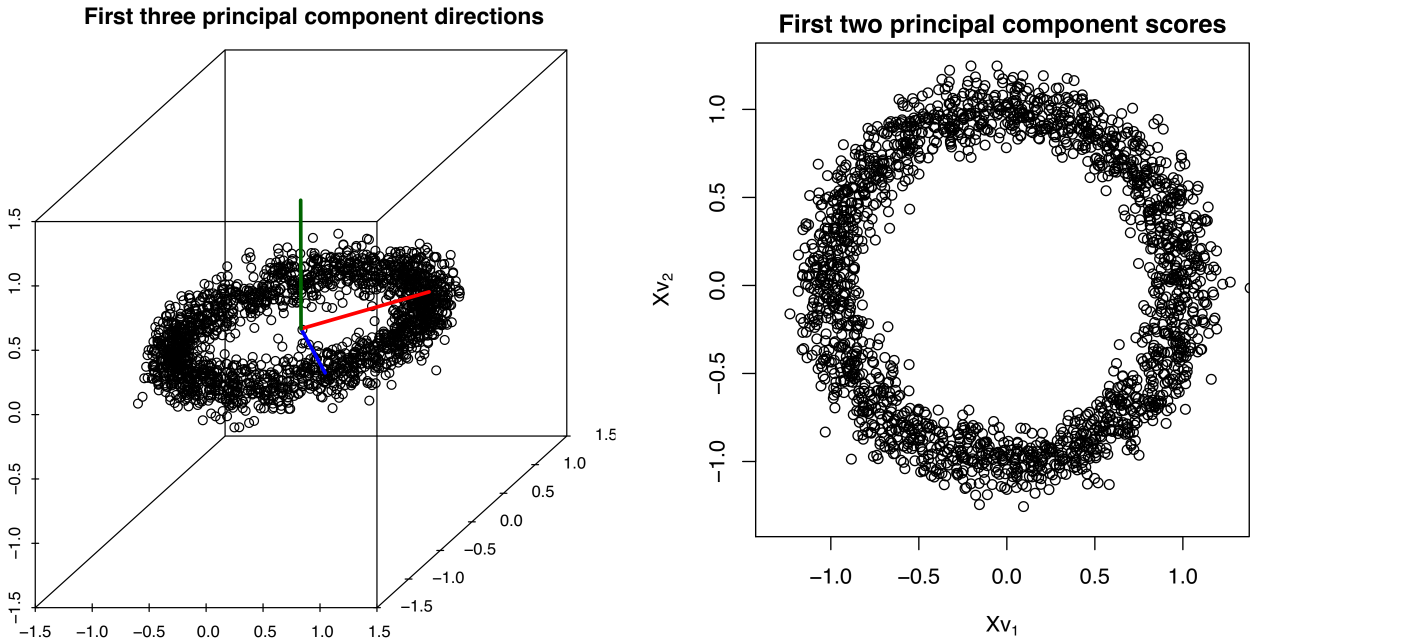
- For the kth principal component direction $v_k \in \mathbb{R}^p$ and score $u_k \in \mathbb{R}^n$, the entries of $Xv_k = d_k u_k$ are the scores from projecting X onto v_k, and the rows of $Xv_k v_k^T = d_k u_k v_k^T$ are the projected vectors

- The directions v_k and normalized scores u_k are only unique up to sign flips

- Concise representation: let the columns of $V \in \mathbb{R}^{p \times p}$ be the directions.
  1. Scores: columns of $XV \in \mathbb{R}^{n \times p}$.
  2. Projections onto V_k (first k columns of V): rows of $XV_k V_k^T \in \mathbb{R}^{n \times p}$

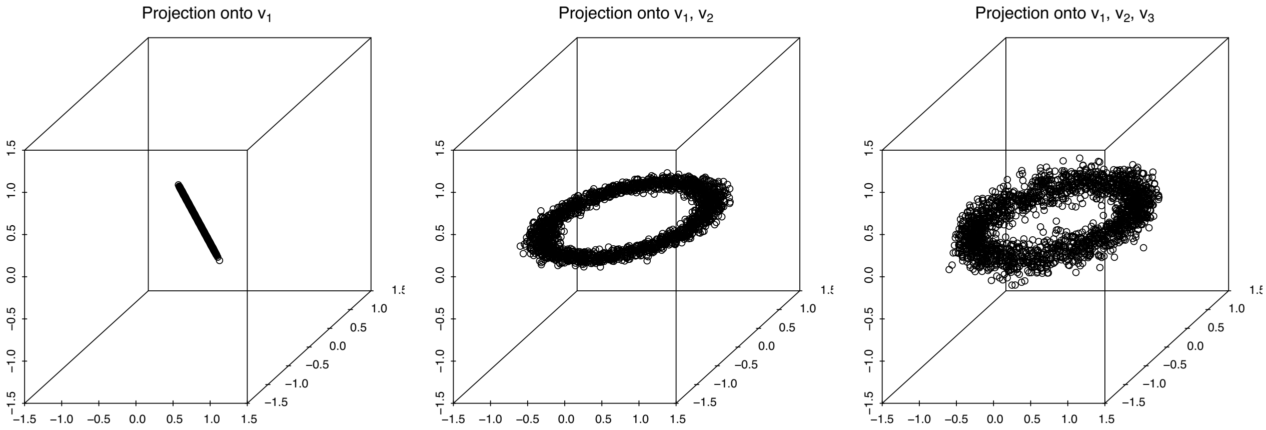
Example: principal component analysis in $\mathbb{R}^3$

Example: $X \in \mathbb{R}^{2000\times 3}$. Shown are the 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: $X \in \mathbb{R}^{2000\times 3}$, $v_1, v_2, \ldots v_3 \in \mathbb{R}^3$. 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...

Proportion of variance explained

Recall that we said: d_k^2 is the amount of variance explained by the kth principal component direction v_k

sum all eigenvalues of $\hat{\Sigma}$.

Two facts:

- The total sample variance of X is $\sum_{j=1}^p d_j^2$
- The total sample variance of $XV_kV_k^T$ is $\sum_{j=1}^k d_j^2$ (amount of variance explained by $v_1 \ldots v_k$)

v_1 explains d_1^2

v_2 explains d_2^2

Hence the proportion of variance explained by the first k principal component directions $v_1, \ldots v_k$ is

$$\frac{\sum_{j=1}^k d_j^2}{\sum_{j=1}^p d_j^2}$$

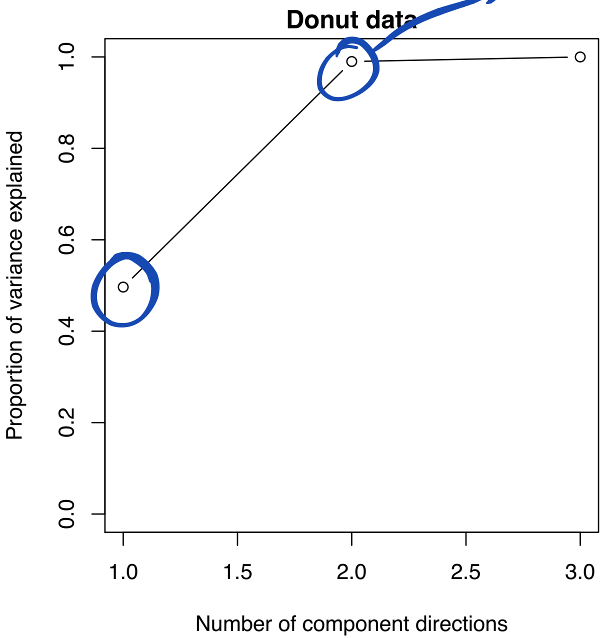
$$\frac{\text{var exp by } k}{\text{total variance}}.$$

If this is high for a small value of k, then it means that the main structure in X can be explained by a small number of directions

Example: proportion of variance explained

Example: proportion of variance explained as a function of k, for the donut data

first 2 explain most of variance



$$\hat{\Sigma} = V \underset{\sim}{D} V^T.$$

$$= \begin{bmatrix} v_1 & \cdots & v_d \\ & & \\ & & \end{bmatrix} \begin{bmatrix} d_{11} & & \\ & \ddots & \\ & & \ddots \end{bmatrix}$$

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 $Xv_1, \ldots Xv_k \in \mathbb{R}^n$

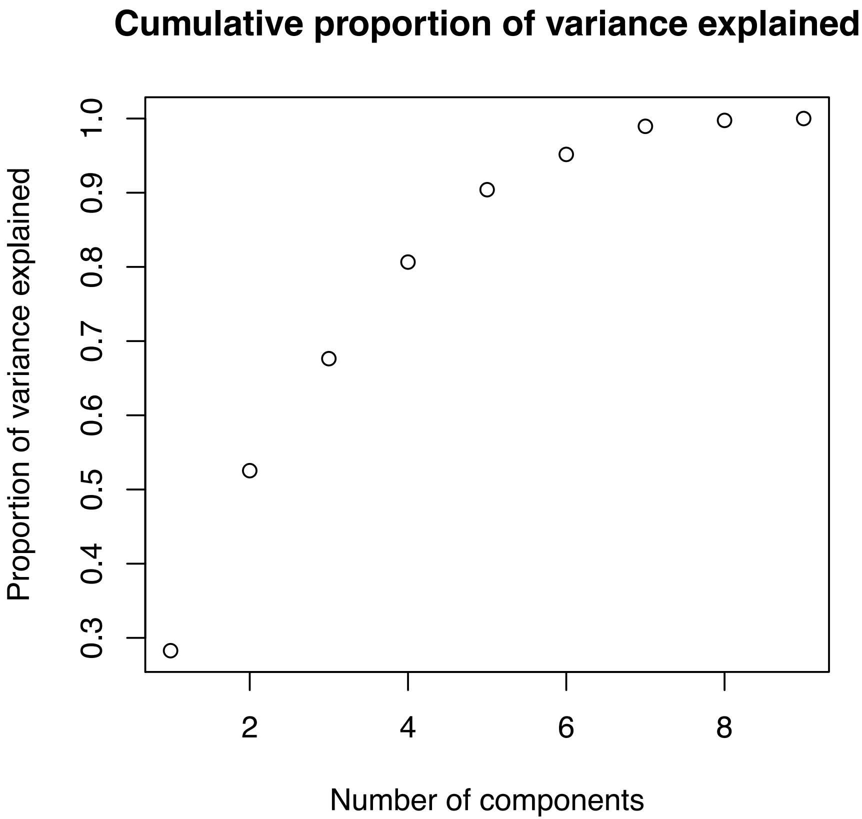
We can think of $Xv_1, \ldots Xv_k$ as our new feature vectors, which is a big savings if $k \ll p$ (e.g. $k = 2$ or 3)

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, glass data



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 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

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

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

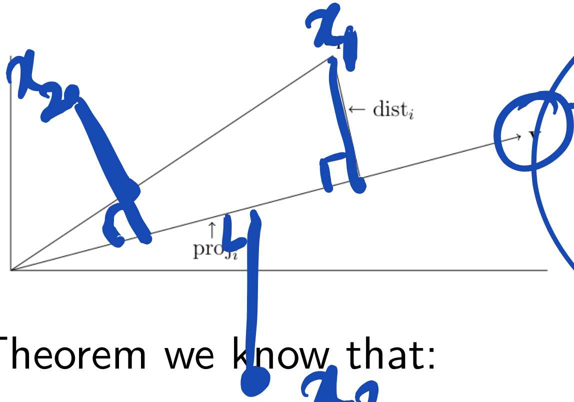
(Aside: the above problem is nonconvex, and would be very hard to solve in general!)

Understanding the Alternate Characterization

max. variance

- We will not spend too much time on this but here is how to think about the alternate characterization.

minimize $\sum_{i=1}^{n} (\text{dist}_i)^2$



top ~~princ.~~ component

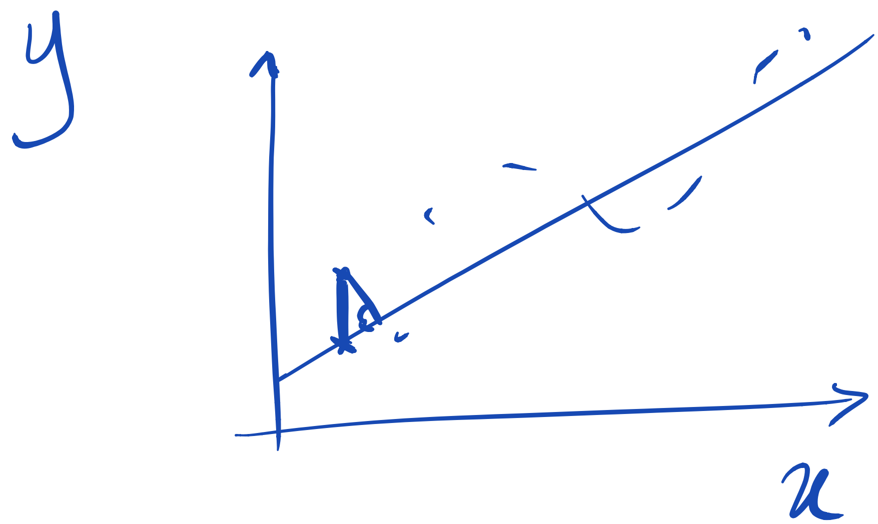
$\text{Var}_i = (\text{proj}_i)^2$

- By Pythagoras' Theorem we know that:

$$(\text{dist}_i)^2 + (\text{proj}_i)^2 = \underbrace{\|x_i\|_2^2}_{\text{const.}}$$

- So we conclude that:

max $\sum_i (\text{proj}_i)^2$ is eqvt to min $\sum_i (\text{dist}_i)^2$


y
x

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.

Computing principal component directions

This is just a repeat of things you have already seen. There are two ways to compute the principal components.

Eigenvalue Decomposition: We write $\frac{X^T X}{n} = VDV^T$, where the columns of V are the eigenvectors and D is the diagonal matrix of eigenvalues. Then

- The columns of V, v_j are the principal component directions.
- The eigenvalues are the amounts of variation explained.
- We can compute the scores Xv_j.

Computing principal component directions: SVD

The other alternative is to compute the SVD of X.

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

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 principal component scores
- Squaring the jth diagonal element of D and dividing by n gives the variance explained by v_j

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

Summary

- Two ways to think about PCA:
  1. k orthogonal directions of maximum variance.
  2. k dimensional subspace that is "closest" to the data.
- Two (closely related) ways to compute the principal components:
  1. Using an eigendecomposition on the covariance matrix.
  2. Using SVD on the data matrix X.