Dimensionality Reduction: Principal Components Analysis

cs4821-cs5831-s24

Some slides adapted from P. Smyth; A. Moore, D. Klein Han, Kamber, Pei; Tan, Steinbach, Kumar; L. Kaebling; R. Tibshirani; T. Taylor; and L. Hannah

Types of Dimensionality Reduction Methods

Example Methods

	How lower-dimensional space is built?	
What machine learning/data mining method is considered?	Extract, Unsupervised Ex. PCA	Select, Unsupervised Ex. EM Clustering
	Extract, Supervised Ex. LDA	Select, Supervised Ex. Many Feature selection

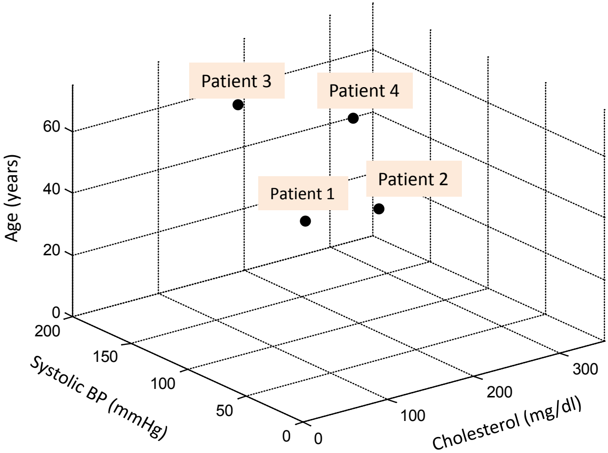
Let’s look at the unsupervised technique of **PCA**
Principal Components Analysis

Principal Components Analysis

- Principal components allows for a large set of features to be summarized with a smaller number of representative features that explain most of the variability in the original data
- The directions of the principal components are those in which the original data is highly variable
- PCA, principal components analysis, is the process to compute the principal components
- PCA is an unsupervised method, does not require a class label for the data
    - Note, PCA can be run on supervised data sets, the target/class variable is usually not included in the analysis

Vectors in p-dimensional space

- Let $x \in \mathbb{R}^p$ be a sample data measurement



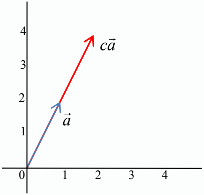
Multiplication by a scalar

- Consider a vector $\vec{x} \in \mathbb{R}^p = (x_1, x_2, \ldots, x_p)$ and a scalar a
- Define $a\vec{x} = (ax_1, ax_2, \ldots, ax_p)$
- *When you multiply a vector by a scalar, you “stretch” it in the same or opposite direction depending on whether the scalar is positive or negative*

$\vec{a} = (1,2)$

$c = 2$

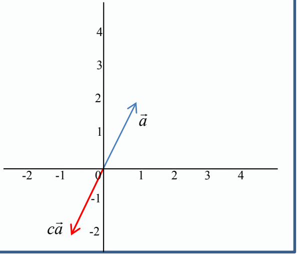
$\vec{c}a = (2,4)$



$\vec{a} = (1,2)$

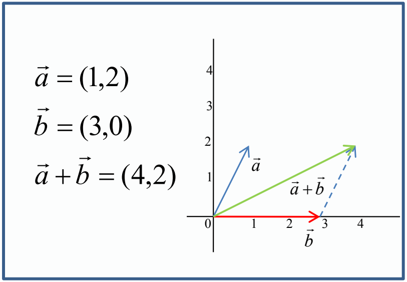
$c = -1$

$\vec{c}a = (-1,-2)$



Addition

- Consider a vector $\vec{x} \in \mathbb{R}^p = (x_1, x_2, \ldots, x_p)$ and $\vec{y} = (y_1, y_2, \ldots, y_p)$
- Define: $\vec{x} + \vec{y} = (x_1 + y_1, x_2 + y_2, \ldots, x_p + y_p)$



$\vec{a} = (1,2)$

$\vec{b} = (3,0)$

$\vec{a} + \vec{b} = (4,2)$

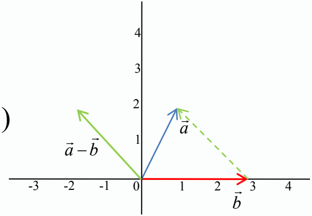
Subtraction

- Consider a vector $\vec{x} \in \mathbb{R}^p = (x_1, x_2, \ldots, x_p)$ and $\vec{y} = (y_1, y_2, \ldots, y_p)$
- Define: $\vec{x} - \vec{y} = (x_1 - y_1, x_2 - y_2, \ldots, x_p - y_p)$

$$\vec{a} = (1,2)$$

$$\vec{b} = (3,0)$$

$$\vec{a} - \vec{b} = (-2,2)$$

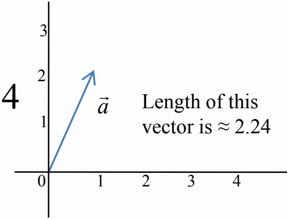


Euclidean length of L2-norm

- Consider a vector $\vec{x} \in \mathbb{R}^p = (x_1, x_2, \ldots, x_p)$
- Define the L2-norm as $\|\vec{x}\|_2 = \sqrt{x_1^2 + x_2^2 + \cdots + x_p^2}$
  - If the norm is written without a subscript, it is usually assumed to be the L2 norm

$$\vec{a} = (1,2)$$

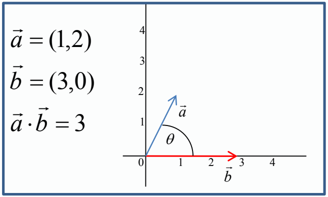
$$\left\|\vec{a}\right\|_2 = \sqrt{5} \approx 2.24$$



Length of this vector is ≈ 2.24

Dot product

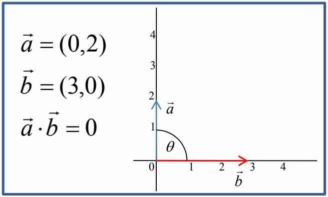
- Consider a vector $\vec{x} \in \mathbb{R}^p = (x_1, x_2, \ldots, x_p)$ and $\vec{y} = (y_1, y_2, \ldots, y_p)$
- Define: $\vec{x} \cdot \vec{y} = x_1 y_1 + x_2 y_2 + \cdots x_p y_p = \sum_{i=1}^{n} x_i y_i$
  - The law of cosines says that $\vec{x} \cdot \vec{y} = \|\vec{x}\|_2 \|\vec{y}\|_2 \cos\theta$, where θ is the angle between $\vec{x}$ and $\vec{y}$



$\vec{a} = (1,2)$

$\vec{b} = (3,0)$

$\vec{a} \cdot \vec{b} = 3$



$\vec{a} = (0,2)$

$\vec{b} = (3,0)$

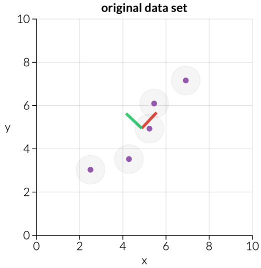
$\vec{a} \cdot \vec{b} = 0$

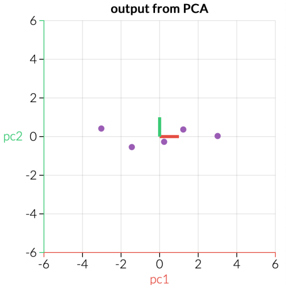
Basic Idea of PCA

- Given set of p “old” or original variables
- We can create a set of p “new” variables, where the new variables are linear combinations of the old.
    - the new variables are ordered by importance
    - the new variables are in directions orthogonal to one another
- So, we can select the best $k < p$ “new” variables to maintain the most valuable parts of all of the variables

Basic Idea of PCA

- We look at two aspects of our data, the “direction” and “magnitude” (how important is it)
- For some 2D data, the red direction is the most important, followed by the green direction; why?
- We can transform our data to align with the directions (linear combinations of the original variables)





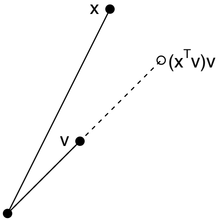
Example from: `http://setosa.io/ev/principal-component-analysis/`

Walk Through PCA Idea

1. For data, X; for each column, subtract the mean of the column for each entry
2. Decide if you want to standardize. If the importance of features is independent of the variance of the features, then divide each entry in a column by that column's standard deviation. Let's call this new data Z
3. Calculate $\frac{1}{n-1}Z^T Z$ - the covariance matrix of Z
4. Find new basis (set of directions) for data that maximizes variance.
   a. Eigendecomposition
   b. Singular Value Decomposition

Principles of Linear Projection

- Let $x \in \mathbb{R}^p$ be a sample data measurement
- Let $v \in \mathbb{R}^p$ be a vector where $||v||_2^2 = v^T v = 1$, that is v has a unit norm.
- The projection of x onto the direction of v is $(x^T v)v$
  think of as $c \cdot v$ where c is the "score' ' or coefficient $c = x^T v$



Principles of Linear Projection

- Let $X \in \mathbb{R}^{n \times p}$ be a data set. Each sample (row), $x_i \in \mathbb{R}^p$ can be projected onto a direction v.
- The entries of $Xv \in \mathbb{R}^n$ are scores
- The rows of $Xvv^T \in \mathbb{R}^{n \times p}$ are the projected vectors

$$Xv = \begin{pmatrix} x_1^T v \\ x_2^T v \\ \dots \\ x_n^T v \end{pmatrix} \qquad Xvv^T = \begin{pmatrix} x_1^T vv^T \\ x_2^T vv^T \\ \dots \\ x_n^T vv^T \end{pmatrix}$$

Review of Orthonormal Vectors

- Vectors $v_1, v_2 \in \mathbb{R}^p$ are orthogonal if $v_1^T v_2 = 0$, that is $v_1 \cdot v_2 = 0$
- The set of vectors $v_1, \ldots, v_k \in \mathbb{R}^p$ are orthogonal if $v_i^T v_j = 0$ for any i, j, where $i \neq j$
- Vectors $v_1, \ldots, v_k \in \mathbb{R}^p$ are orthonormal if the vectors are orthogonal and each v_j has unit form, $||v||_2^2 = v^T v = 1$

- The projection of $x \in \mathbb{R}^p$ onto the orthonormal vectors $v_1, \ldots, v_k \in \mathbb{R}^p$ is $\sum_{j=1}^k (x^T v_j) v_j$.
  - the score along the jth direction is $x^T v_j$

Review of Orthonormal Vectors

- The collection of orthonormal vectors $v_1, \ldots, v_k \in \mathbb{R}^p$ is the matrix $V \in \mathbb{R}^{p \times k}$, where each v_j is a column
- Project the rows of the data matrix $X \in \mathbb{R}^{n \times p}$ onto the columns of V
    - the scores are given by $XV \in \mathbb{R}^{n \times k}$, where the jth column Xv_j are the scores of projecting X onto v_j.
    - the projections are the rows of $XVV^T \in \mathbb{R}^{n \times p}$

$$Xv_j = \begin{pmatrix} x_1^T v_j \\ x_2^T v_j \\ \cdots \\ x_n^T v_j \end{pmatrix}$$

Review of Statistics (vector notation)

- Let $x \in \mathbb{R}^n$ be a vector of observations
- Sample mean: $\bar{x} = \frac{1}{n} x^T \mathbf{1} \in \mathbb{R}$, where $\mathbf{1} \in \mathbb{R}^n$ is the vector of 1s
- Sample variance: $\frac{1}{n}(x - \bar{x}\mathbf{1})^T (x - \bar{x}\mathbf{1}) \in \mathbb{R}$
- Let $X \in \mathbb{R}^{n \times p}$, be a data matrix of n samples of p observations
- Sample mean vector: $\bar{X} = \frac{1}{n} X^T \mathbf{1} \in \mathbb{R}^p$
- Sample covariance matrix: $\frac{1}{n}(X - \mathbf{1}\bar{X}^T)^T (X - \mathbf{1}\bar{X}^T) \in \mathbb{R}^{p \times p}$

Center Data

It is necessary to center data before running PCA

- To center $x \in \mathbb{R}^n$, replace it with $\tilde{x} = x - \bar{x}\mathbf{1} \in \mathbb{R}^n$
    - $\tilde{x}$ has sample mean 0 and sample variance is same as before
- To center (column-center) $X \in \mathbb{R}^{n \times p}$, replace it with $\tilde{X} = X - \mathbf{1}\bar{X}^T \in \mathbb{R}^{n \times p}$
    - each column of $\tilde{X}$ has sample mean zero, but sample covariance remains the same as before

Principal Component Analysis

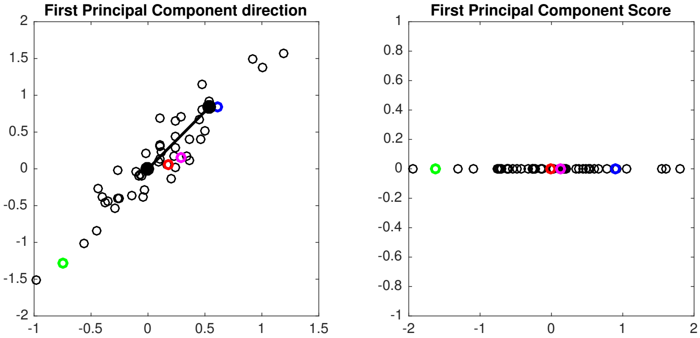
- Let $X \in \mathbb{R}^{n \times p}$ be a centered data matrix.

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

$$v_1 = \underset{||v||_2=1}{\arg\max} (Xv)^T (Xv)$$

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

Example: First PCA direction and score



PCA, beyond first direction

- We have successfully explained the variance of X along v_1, now need to look at variance in a different direction, an orthonormal direction.

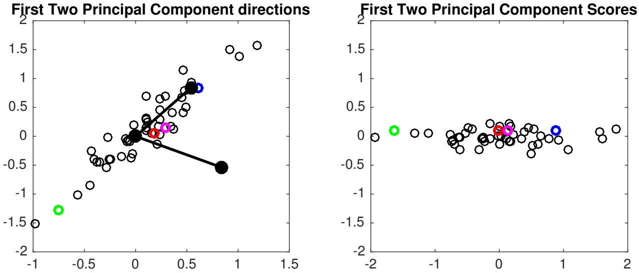
- The second principal component direction of X is the unit vector $v_2 \in \mathbb{R}^p$, with $v_2^T v_1 = 0$ that maximizes the sample variance of $Xv_2 \in R^n$ compared to all other unit vectors orthogonal to v_1

$$v_2 = \underset{||v||_2=1, v^T v_1 = 0}{\arg\max} (Xv)^T (Xv)$$

- The second principal component score is the vector $Xv_2 \in \mathbb{R}^n$
  - normalized principal component score is $u_2 = (Xv_2)/d_2 \in \mathbb{R}^n$
  - the amount of variance explained by v_2 is d_2^2/n where $d_2 = \sqrt{(Xv_2)^T (Xv_2)}$

Example: First Two PCA direction and score

Using the same example data as before.



PCA, in general

- Given $k-1$ principal component directions $v_1, \ldots, v_{k-1} \in \mathbb{R}^p$
- The kth principal component direction of X is the unit vector $v_k \in \mathbb{R}^p$,

$$v_k = \underset{||v||_2=1, v^T v_j=0 \; for \; j=1,\ldots,k-1}{\arg\max} (Xv)^T(Xv)$$

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

PCA in R

There are two main functions that can be used for principal components analysis in R.

- the function `princomp` in the `base` package, computes the score and directions via an eigendecomposition of $X^T X$.

```r
pc = princomp(x)
dirs = pc$loadings  # directions
scores = pc$scores  # scores
```

- the function `prcomp` in the `base` package, computes the scores and directions via singluar value decomposition of X.

```r
pc = prcomp(x)
dirs = pc$rotation
scores = pc$x
```

PCA in Matlab

The main function for performing PCA in matlab is pca.

```
[coeffs, scores] = pca(x)
dirs = coeffs    % directions
scores = scores  % scores
```

PCA in Python

```
from sklearn.decomposition import PCA
pca = PCA(n_components=k)
pca.fit(X)
print(pca.components_)        # directions
Xnew = pca.transform(X)       # scores
```

Dimension Reduction with PCA

Dimension reduction is performed via PCA by taking the first k principal component scores $Xv_1, \ldots, Xv_k \in \mathbb{R}^n$.

Then, $Xv_1, \ldots, Xv_k$ can be thought of as the new feature vectors, with a savings when $k << p$

The question is then, how good are the features at capturing the information of the original data?

Approximation of Data

Think about approximating X by $XV_kV_k^T$, the projection of X onto the first k principal component directions.

Recall, $X \in \mathbb{R}^{n\times p}$ is centered data, and $V_k = [v_1 \dots v_k] \in \mathbb{R}^{n\times k}$ is the matrix whose columns contain the first k principal component directions of X then,

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

That is, $XV_kV_k^T$ is the best rank k approximation to X.

Proportion of Variance Explained

Recall that d_k^2/n is the amount of variance explained by the kth principal component direction v_k

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

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

If the proportion is large for a small value of k, this means the main structure of X can be explained by a small number of directions.

Computation of PCA directions

There are two main methods for computing PCA: *eigendecomposition* and *SVD*, we will briefly discuss SVD.

The singular value decomposition (SVD) of X:

$$\begin{array}{ccccc} X & = & U & D & V^T \\ n \times p & & n \times p & p \times p & p \times p \end{array}$$

The matrix $D = diag(d_1, \ldots, d_p)$ is the diagonal with $d_1 \geq \ldots \geq d_p \geq 0$ and U, V have orthonormal columns, where:

- 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 given the variance explained by v_j

PCA Example: Iris Data

Iris data set is $[150 \times 4]$, 4 features are sepal length, sepal width, petal length, and petal width.

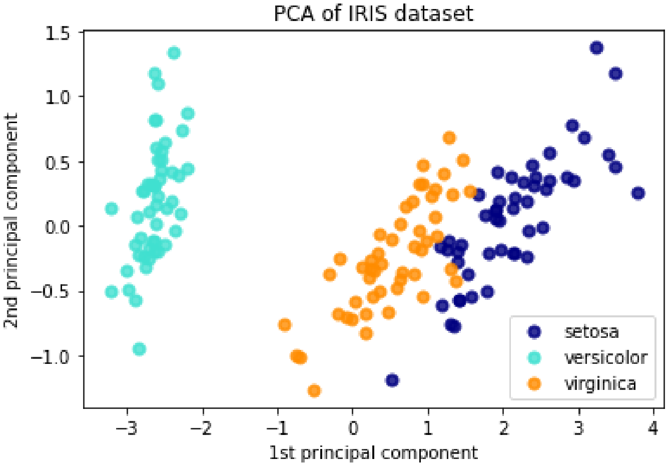
```
pc = prcomp(iris[,1:4])
dirs = pc$rotation
scores = pc$x

dirs
```

```
##                      PC1         PC2         PC3        PC4
## Sepal.Length  0.36138659 -0.65658877  0.58202985  0.3154872
## Sepal.Width  -0.08452251 -0.73016143 -0.59791083 -0.3197231
## Petal.Length  0.85667061  0.17337266 -0.07623608 -0.4798390
## Petal.Width   0.35828920  0.07548102 -0.54583143  0.7536574
```

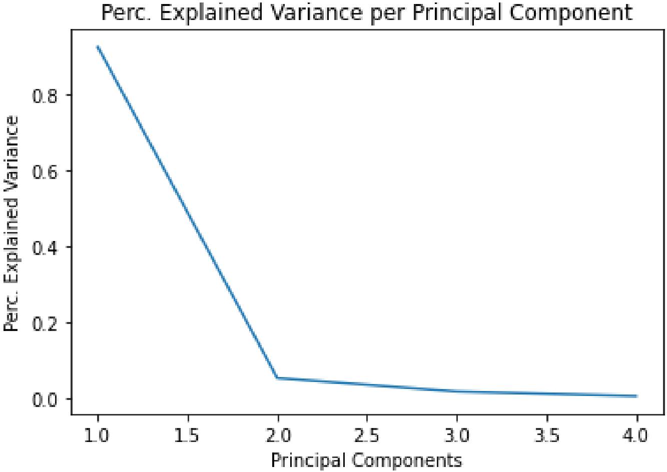
PCA Example: Iris Data

```
from sklearn import datasets
iris = datasets.load_iris()
X = iris.data

pca = PCA(n_components=4)
pca.fit(X)
Xnew = pca.transform(X)

print(pca.components_)
[[ 0.36138659 -0.08452251  0.85667061  0.3582892 ]
 [ 0.65658877  0.73016143 -0.17337266 -0.07548102]
 [-0.58202985  0.59791083  0.07623608  0.54583143]
 [-0.31548719  0.3197231   0.47983899 -0.75365743]]
```

PCA Example: Iris Data



PCA Example: Iris Data


Perc. Explained Variance per Principal Component
Perc. Explained Variance
Principal Components