CS 1675 Introduction to Machine Learning
Lecture 22

Dimensionality reduction II

Milos Hauskrecht
milos@cs.pitt.edu
5329 Sennott Square

Dimensionality reduction

Problem: Is there a lower dimensional representation of the data that captures well its characteristics?

- **Assume:**
  - We have data $D = \{\mathbf{x_1}, \mathbf{x_2}, .., \mathbf{x_N}\}$ such that
    $\mathbf{x}_i = (x_i^1, x_i^2, .., x_i^d)$
  - Assume the dimension d of the data point $\boldsymbol{x}$ is very large
- **Our goal:**
- find a lower dimensional representation d' of the data
  - where every $\mathbf{x}_i$ is replaced with a new $\mathbf{x}_i'$
- **Why we want to do this?**
  - **Many methods of analysis are sensitive to the dimensionality *d***

Task-specific feature selection

Assume: **Classification problem**:

- **x** – input vector, y - output

Objective: Find a subset of inputs/features that gives/preserves most of the data prediction capabilities

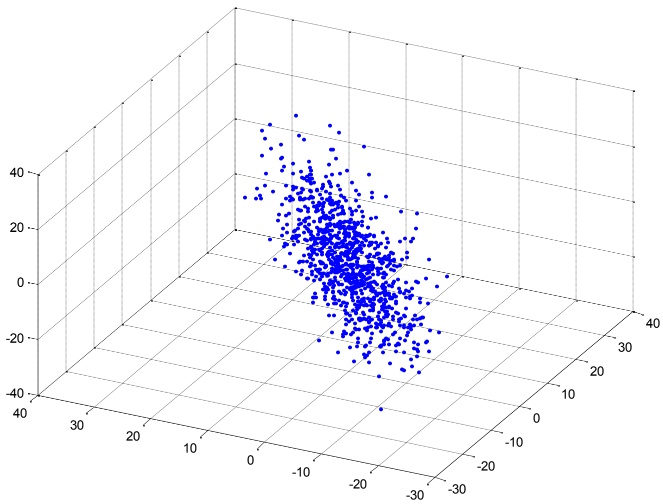
Selection approaches:

- **Filtering approaches**
  - Filter out features with small predictive potential
  - Done before classification; typically uses univariate analysis
- **Wrapper approaches**
  - Select features that directly optimize the accuracy of the multivariate classifier
- **Embedded methods**
  - Feature selection and learning closely tied in the method
  - Regularization methods, decision tree methods

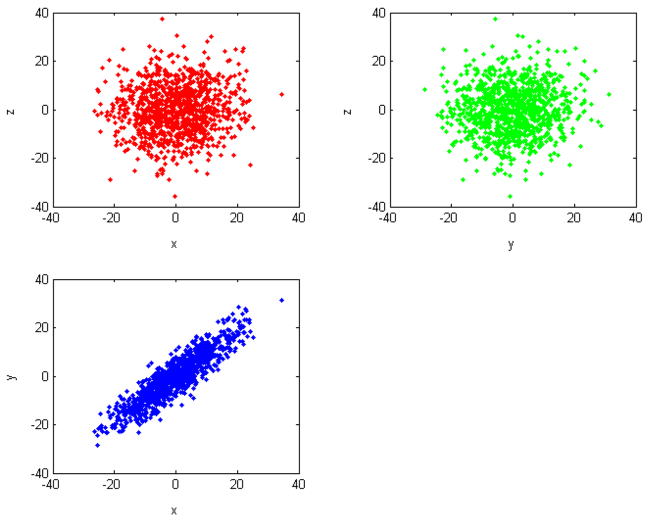
Principal component analysis (PCA)

- **Unsupervised dimensionality reduction method**
- **Objective:** We want to replace a high dimensional input with a small set of features (obtained by combining inputs)
  - Different from the feature subset selection !!!
- **PCA:**
  - A linear transformation of *d* dimensional input *x* to M dimensional feature vector *z* such that $M < d$ under which the retained variance is maximal.
  - Equivalently it is the linear projection for which the sum of squares reconstruction cost is minimized.

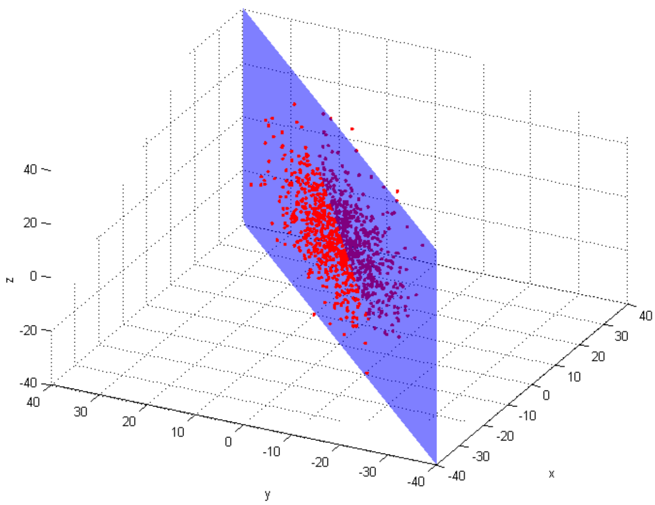
PCA


40
20
0
-20
-40
40
30
20
10
0
-10
-20
-30
-30
-20
-10
0
10
20
30
40

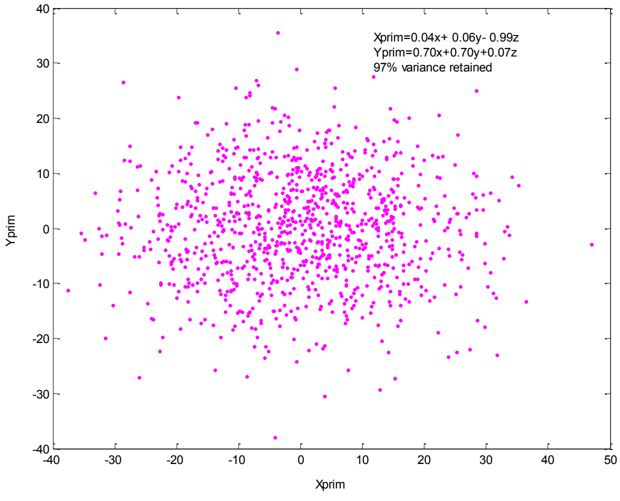
PCA


z
x
z
y
y
x
40
20
0
-20
-40
-40
-20
0
20
40

PCA


40
20
0
-20
-40
z
40
30
20
10
0
-10
-20
-30
-40
y
40
30
20
10
0
-10
-20
-30
-40
x

PCA


Xprim=0.04x+ 0.06y- 0.99z
Yprim=0.70x+0.70y+0.07z
97% variance retained
40
30
20
10
0
-10
-20
-30
-40
Yprim
-40
-30
-20
-10
0
10
20
30
40
50
Xprim

Principal component analysis (PCA)

- **PCA:**
  - linear transformation of a d dimensional input $\mathbf{x}$ to M dimensional vector $\mathbf{z}$ such that $M < d$ under which the retained variance is maximal.
  - Task independent
- **Fact:**
  - A vector $\mathbf{x}$ can be represented using a set of orthonormal vectors $\mathbf{u}$

$$\mathbf{x} = \sum_{i=1}^{d} z_i \mathbf{u}_i$$

  - Leads to transformation of coordinates (from $\mathbf{x}$ to $\mathbf{z}$ using $\mathbf{u}$'s)

$$z_i = \mathbf{u}_i^T \mathbf{x}$$

PCA

- **Idea:** replace d coordinates with M of z_i coordinates to represent x. We want to find the subset M of basis vectors.

$$\tilde{\mathbf{x}} = \sum_{i=1}^{M} z_i \mathbf{u}_i + \sum_{i=M+1}^{d} b_i \mathbf{u}_i$$

b_i - constant and fixed

- **How to choose the best set of basis vectors?**
  - We want the subset that gives the best approximation of data x in the dataset on average (we use least squares fit)

Error for data entry $\mathbf{x}^n$ $\quad \mathbf{x}^n - \tilde{\mathbf{x}}^n = \sum_{i=M+1}^{d} (z_i^n - b_i)\mathbf{u}_i$

Reconstruction error

$$E_M = \frac{1}{2}\sum_{n=1}^{N} \left\| \mathbf{x}^n - \tilde{\mathbf{x}}^n \right\| = \frac{1}{2}\sum_{n=1}^{N} \sum_{i=M+1}^{d} (z_i^n - b_i)^2$$

PCA

- **Differentiate the error function** with regard to all b_i and set equal to 0 we get:

$$b_i = \frac{1}{N}\sum_{n=1}^{N} z_i^n = \mathbf{u}_i^T \overline{\mathbf{x}} \qquad \overline{\mathbf{x}} = \frac{1}{N}\sum_{n=1}^{N} \mathbf{x}^n$$

- Then we can rewrite:

$$E_M = \frac{1}{2}\sum_{i=M+1}^{d} \mathbf{u}_i^T \mathbf{\Sigma} \mathbf{u}_i \qquad \mathbf{\Sigma} = \sum_{n=1}^{N} (\mathbf{x}^n - \overline{\mathbf{x}})(\mathbf{x}^n - \overline{\mathbf{x}})^T$$

- The error function is optimized when basis vectors satisfy:

$$\mathbf{\Sigma}\mathbf{u}_i = \lambda_i \mathbf{u}_i \qquad E_M = \frac{1}{2}\sum_{i=M+1}^{d} \lambda_i$$

Eigenvectors

- If A is a square matrix, a non-zero vector **v** is an eigenvector of A if there is a scalar λ (eigenvalue) such that

$$Av = \lambda v$$

- Example: $\begin{pmatrix} 2 & 3 \\ 2 & 1 \end{pmatrix}\begin{pmatrix} 3 \\ 2 \end{pmatrix} = \begin{pmatrix} 12 \\ 8 \end{pmatrix} = 4\begin{pmatrix} 3 \\ 2 \end{pmatrix}$

- If we think of the squared matrix as a transformation matrix, then multiply it with the eigenvector do not change its direction.

PCA

- The error function

$$E_M = \frac{1}{2} \sum_{i=M+1}^{d} \mathbf{u}_i^T \mathbf{\Sigma} \mathbf{u}_i \qquad \mathbf{\Sigma} = \sum_{n=1}^{N} (\mathbf{x}^n - \bar{\mathbf{x}})(\mathbf{x}^n - \bar{\mathbf{x}})^T$$

- is optimized when basis vectors satisfy:

$$\mathbf{\Sigma} \mathbf{u}_i = \lambda_i \mathbf{u}_i \qquad E_M = \frac{1}{2} \sum_{i=M+1}^{d} \lambda_i$$

- Eigenvectors: $\mathbf{u}_i$ are called **principal components**
- **Solution: Select the best *M* basis vectors**: that is, basis vectors with the largest eigenvalues
- Or equivalently discard basis vectors with *d-M* smallest eigenvalues

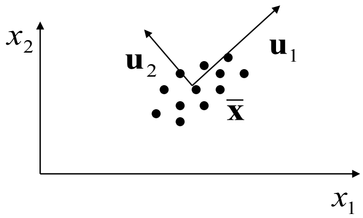
PCA algorithm

PCA steps: transform an $N \times d$ matrix X into an $N \times m$ matrix Y:

- Centralize the data (subtract the mean).
- Calculate the $d \times d$ covariance matrix: $C = \frac{1}{N-1} X^T X$
- $C_{i,j} = \frac{1}{N-1} \sum_{q=1}^{N} X_{q,i} . X_{q,j}$
  - $C_{i,i}$ (diagonal) is the variance of variable i.
  - $C_{i,j}$ (off-diagonal) is the covariance between variables i and j.
- Calculate the eigenvectors of the covariance matrix (orthonormal).
- Select *m* eigenvectors that correspond to the largest *m* eigenvalues to be the new basis.

PCA

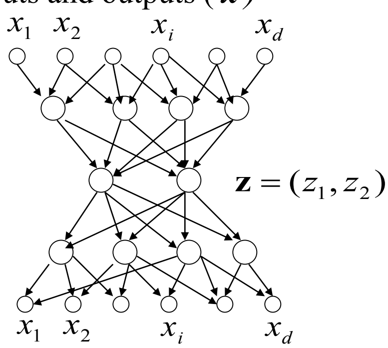
- Once eigenvectors $\mathbf{u}_i$ with largest eigenvalues are identified, they are used to transform the original *d*-dimensional data to *M* dimensions



- To find the "true" dimensionality of the data *d'* we can just look at eigenvalues that contribute the most (small eigenvalues are disregarded)
- **Problem:** PCA is a linear method. The "true" dimensionality can be overestimated. There can be non-linear correlations.
- **Modifications for nonlinearities:** kernel PCA

Dimensionality reduction with neural nets

- **PCA** is limited to linear dimensionality reduction
- To do non-linear reductions we can use neural nets
- **Auto-associative (or auto-encoder) network:** a neural network with the same inputs and outputs ($\boldsymbol{x}$)



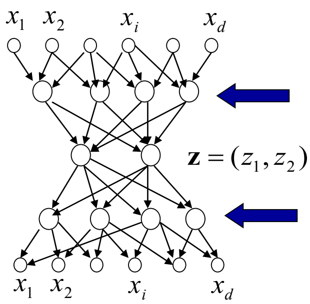
- The middle layer corresponds to the reduced dimensions

Dimensionality reduction with neural nets

- **Error criterion:**

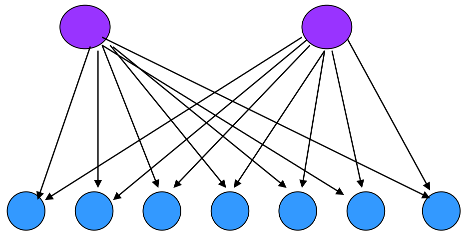
$$E = \frac{1}{2} \sum_{n=1}^{N} \sum_{i=1}^{d} \left(y_i(x^n) - x^n \right)^2$$

- Error measure tries to recover the original data through limited number of dimensions in the middle layer
- **Non-linearities** modeled through intermediate layers between the middle layer and input/output
- If no intermediate layers are used the model replicates PCA optimization through learning



Latent variable models

Latent variables (s): Dimensionality k



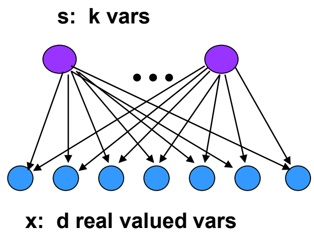
Observed variables x: real valued vars
Dimensionality d

Examples

Model:

$$x = \mathbf{W}\mathbf{s}$$

$$\mathbf{W} = \begin{pmatrix} w_{11} & w_{12} & .. & w_{1k} \\ w_{21} & & & \\ & .. & & \\ w_{d1} & .. & .. & w_{dk} \end{pmatrix}$$



Factor analysis:

- **Decomposes** signal into multiple Gaussian sources

Cooperative vector quantizer:

- **Decomposes** signal into binary sources

Multidimensional scaling

- Find a lower dimensional space projection such that the distances among data points are preserved
- Used in visualization – d-diminensional data transformed to 3D or 2D
- **Dissimilarities before projection** $\delta_{i,j} = \|x_i - x_j\|$
- **Objective:** Optimize points and their coordinates by fitting the dissimilarities afterwards

$$\min_{\{x_1, x_2, \cdots x_n\}} \sum_{i<j} (\|x_i' - x_j'\| - \delta_{ij})^2$$