Unsupervised Learning

吉建民

USTC
jianmin@ustc.edu.cn

2024 年 5 月 7 日

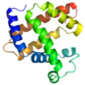
Used Materials

Disclaimer: 本课件采用了 S. Russell and P. Norvig's Artificial Intelligence –A modern approach slides, 徐林莉老师课件和其他网络课程课件，也采用了 GitHub 中开源代码，以及部分网络博客内容

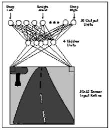
Table of Contents

Supervised learning has many successes

- Document classification
- Protein prediction
- Face recognition
- Speech recognition
- Vehicle steering etc.





However...

- Labeled data can be rare or expensive in many real applications

  - Speech
  - Medical data
  - Protein
  - ...

> Task: speech analysis
> - Switchboard dataset
> - telephone conversation transcription
> - 400 hours annotation time for each hour of speech
>
> **film** ⇒ `f ih_n uh_gl_n m`
> **be all** ⇒ `bcl b iy iy_tr ao_tr ao l_dl`

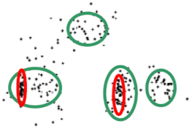
- Unlabeled data is much cheaper and abundant

Question: Can we use unlabeled data to help?

Unsupervised learning

Learning from unlabeled data (without supervision)

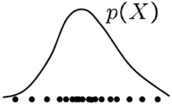
- What can we predict from unlabeled data?
  - Groups or clusters in the data



Unsupervised learning

Learning from unlabeled data (without supervision)

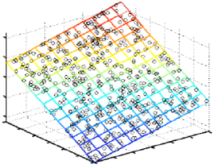
- What can we predict from unlabeled data?
    - Groups or clusters in the data
    - Density estimation (密度估计)



Unsupervised learning

Learning from unlabeled data (without supervision)

- What can we predict from unlabeled data?
  - Groups or clusters in the data
  - Density estimation (密度估计)
  - Low-dimensional structure
    - Principal Component Analysis 主元分析 (PCA) (linear)



Unsupervised learning

Learning from unlabeled data (without supervision)

- What can we predict from unlabeled data?
  - Groups or clusters in the data
  - Density estimation (密度估计)
  - Low-dimensional structure
    - Principal Component Analysis 主元分析 (PCA) (linear)
    - Manifold learning 流行学习 (non-linear)

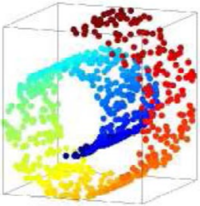
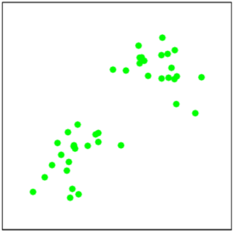


Table of Contents

Clustering

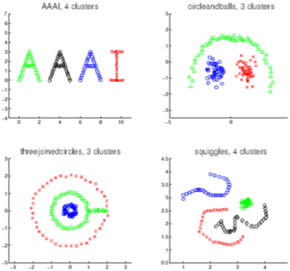


- Are there any “groups” in the data ?
- What is each group ?
- How many ?
- How to identify them?

Clustering

- Group the data objects into subsets or "clusters":
  - High similarity within clusters
  - Low similarity between clusters

- A common and important task that finds many applications in Science, Engineering, information Science, and other places
  - Group genes that perform the same function
  - Group individuals that has similar political view
  - Categorize documents of similar topics
  - Identify similar objects from pictures



Clustering

- Input: training set of input point

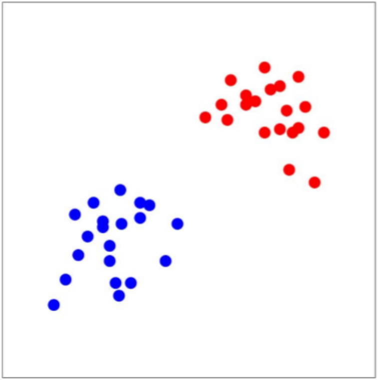
$$D_{train} = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$$

- Output: assignment of each point to a cluster

$$(\, C(1), \ldots, C(n)\,) \text{ where } C(i) \in \{\, 1, \ldots, k\,\}$$

K-means clustering

Create centers and assign points to centers to minimize sum of squared distance



K-means objective

- Each cluster is represented by a centroid μ
- Encode each point by its cluster center, pay a cost for deviation
- Loss function based on reconstruction

$$Loss_{kmeans} \sum_{j=1}^{n} \left\| \mu_{C(j)} - \mathbf{x}_j \right\|^2$$

K-means algorithm

- Goal: $\min_{\mu} \min_{C} \sum_{j=1}^{n} \left\| \mu_{C(j)} - \mathbf{x}_j \right\|^2$

- Strategy: alternating minimization
  - Step 1: if know cluster centers μ, can find best C
  - Step 2: if know cluster assignments C, can find best cluster centers

K-means algorithm

Optimize loss function $Loss(\mu, C)$

$$\min_{\mu} \min_{C} \sum_{j=1}^{n} \left\| \mu_{C(j)} - \mathbf{x}_j \right\|^2$$

(1) Fix μ, optimize C

$$\min_{C(1),C(2),\ldots,C(n)} \sum_{j=1}^{n} \left\| \mu_{C(j)} - \mathbf{x}_j \right\|^2$$

Assign each point to the nearest cluster center

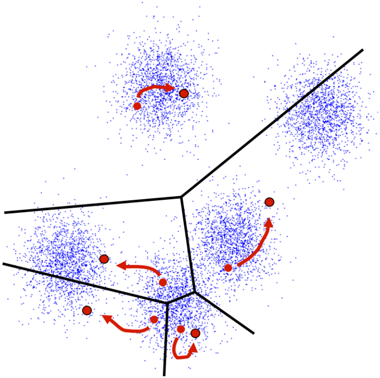
(2) Fix C, optimize μ

$$\min_{\mu(1),\mu(2),\ldots,\mu(k)} \sum_{j=1}^{n} \left\| \mu_{C(j)} - \mathbf{x}_j \right\|^2$$

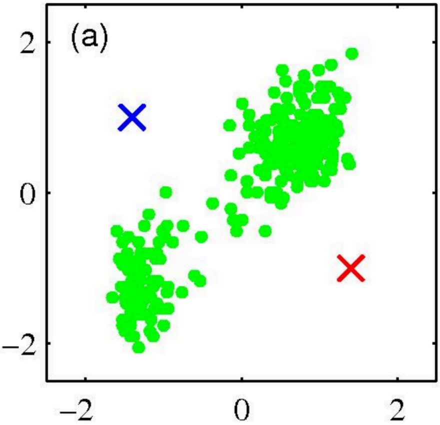
Solution: average of points in cluster i, exactly second step (re-center)

K-Means

- An iterative clustering algorithm
  - Initialize: Pick *K* random points as cluster centers
  - Alternate:
    1. Assign data points to closest cluster center
    2. Change the cluster center to the average of its assigned points
  - Stop when no points' assignments change



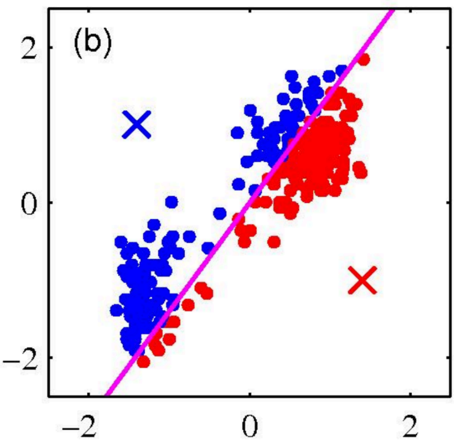
K-means clustering: Example



- Pick K random points as cluster centers (means)

Shown here for K=2

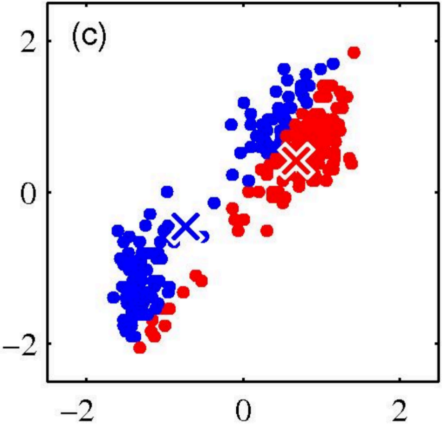
K-means clustering: Example



Iterative Step 1

- Assign data points to closest cluster center

K-means clustering: Example

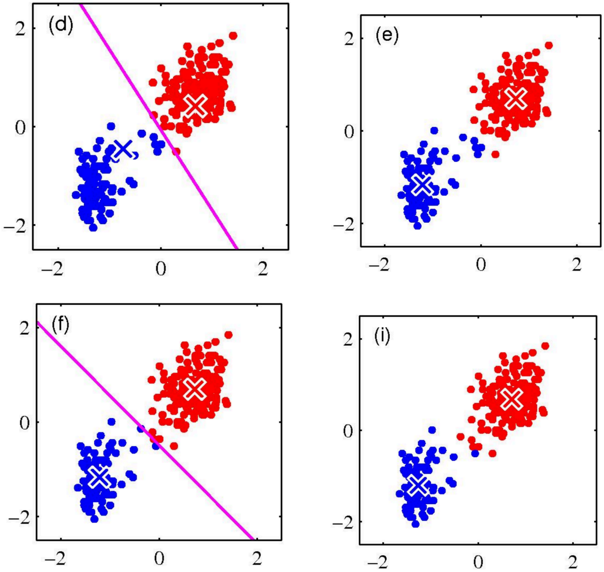


Iterative Step 2

- Change the cluster center to the average of the assigned points

K-means clustering: Example

Repeat until convergence



Properties of K-means algorithm

- Guaranteed to converge in a finite number of iterations
  - To a local minimum
  - The objective is non-convex, so coordinate descent on is not guaranteed to converge to the global minimum
- Running time per iteration: simple and efficient
  - Assign data points to closest cluster center

$$O(KN)$$

  - Change the cluster center to the average of its assigned points

$$O(N)$$

- Different initialization will lead to different results
- K-means problem is **NP-hard** （之前公式的最优解）
- Not robust to noise and outliers

K-means convergence

Objective

$$\min_{\mu}\min_{C}\sum_{i=1}^{k}\sum_{x\in C_i}|x-\mu_i|^2$$

1. Fix μ, optimize C:

$$\min_{C}\sum_{i=1}^{k}\sum_{x\in C_i}|x-\mu_i|^2 = \min_{c}\sum_{i}^{n}\left|x_i-\mu_{x_i}\right|^2$$

Step 1 of kmeans

2. Fix C, optimize μ:

$$\min_{\mu}\sum_{i=1}^{k}\sum_{x\in C_i}|x-\mu_i|^2$$

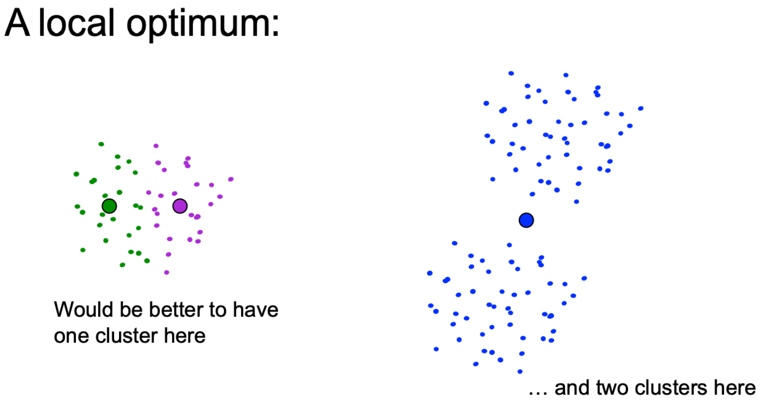
– Take partial derivative of μ_i and set to zero, we have

$$\mu_i = \frac{1}{|C_i|}\sum_{x\in C_i} x$$

Step 2 of kmeans

Kmeans takes an alternating optimization approach, each step is guaranteed to decrease the objective – thus guaranteed to converge

K-means getting stuck



K-means not able to properly cluster

Changing the features (distance function) can help

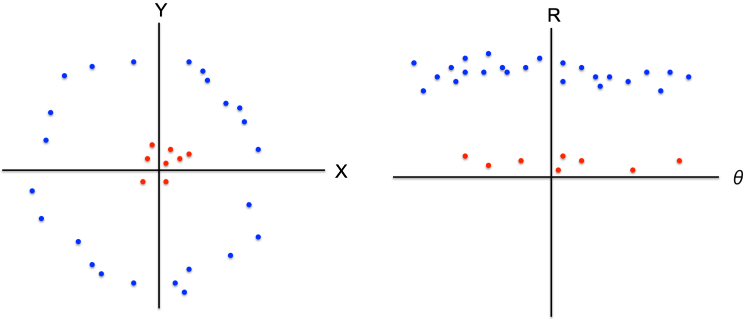


Table of Contents

Principle component analysis

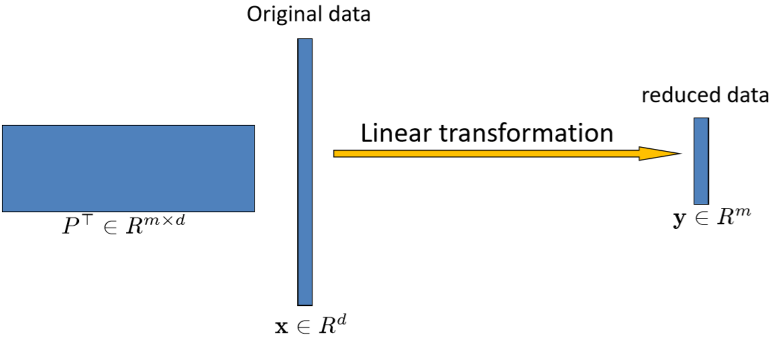
- What is dimensionality reduction?
- Why dimensionality reduction?
- Principal Component Analysis (PCA)
- Nonlinear PCA using Kernels

What is dimensionality reduction?

- Dimensionality reduction refers to the mapping of the original high-dimensional data onto a lower-dimensional space.
  - Criterion for dimensionality reduction can be different based on different problem settings.
    - Unsupervised setting: minimize the information loss
      最近重构性：样本点到这个超平面的距离都足够近
    - Supervised setting: maximize the class discrimination
      最大可分性：样本点在这个超平面上的投影能尽可能分开
    - 对样本进行中心化处理以后，两者等价
- Given a set of data points of d dimension variables $\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\}$
- Compute the linear transformation (projection)

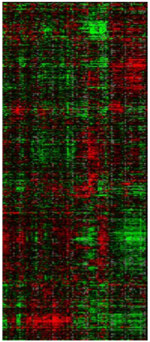
$$P \in R^{d\times m}: \ \mathbf{x} \in R^d \rightarrow \mathbf{y} = P^\top \mathbf{x} \in R^m \quad (m << d)$$

What is dimensionality reduction?



$$P \in R^{d\times m}: \ \mathbf{x} \in R^d \to \mathbf{y} = P^\top \mathbf{x} \in R^m$$

High-dimensional data



Gene expression

Face images

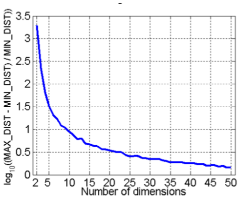
Handwritten digits

Why dimensionality reduction?

- Most machine learning and data mining techniques may not be effective for high-dimensional data
  - Curse of Dimensionality
  - Query accuracy and efficiency degrade rapidly as the dimension increases.
- The intrinsic dimension may be small.
  - For example, the number of genes responsible for a certain type of disease may be small.

Curse of Dimensionality (维数灾难)

- When dimensionality increases, data becomes increasingly sparse in the space that it occupies
- Definitions of density and distance between points, which is critical for clustering and outlier detection, become less meaningful
- If $\mathrm{N}_1 = 100$ represents a dense sample for a single input problem, then $N_{10} = 100^{10}$ is the sample size required for the same sampling density with dimension 10.
- The proportion of a hypersphere (超球面) with radius *r* and dimension *d*, to that of a hyercube (超立方体) with sides of length $2r$ and dimension *d* converges to 0 as *d* goes to infinity —nearly all of the high-dimensional space is "far away" from the center



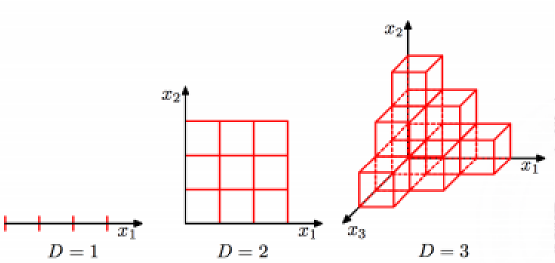
- Randomly generate 500 points
- Compute difference between max and min distance between any pair of points

High dimensional spaces are empty

The volume of an hypercube with an edge length of $r = 0.1$ is 0.1^p $\rightarrow$ when p grows, it quickly becomes so small that the probability to capture points from your database becomes very very small...

Points in high dimensional spaces are isolated

To overcome this limitation, you need a number of sample which grows exponentially with p...



Lost in space

Let's consider a hypersphere of radius r inscribed in a hypercube with sides of length $2r$. Then take the ratio of the volume (体积) of the hypersphere to the hypercube. We observe the following trends.

- in 2 dimensions:

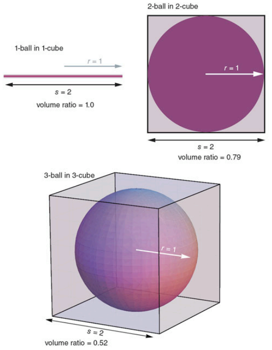
$$\frac{V(S_2(r))}{V(H_2(2r))} = \frac{\pi r^2}{4r^2} = 78.5\%$$

- in 3 dimensions:

$$\frac{V(S_3(r))}{V(H_3(2r))} = \frac{\frac{4}{3}\pi r^3}{8r^3} = 52.4\%$$

- when the dimensionality d increases asymptotically

$$\lim_{d\to\infty} \frac{V(S_d(r))}{V(H_d(2r))} = \lim_{d\to\infty} \frac{\pi^{d/2}}{2^d\Gamma(\frac{d}{2}+1)} \to 0$$



Why dimensionality reduction?

- Visualization: projection of high-dimensional data onto 2D or 3D.
- Data compression: efficient storage and retrieval
- Noise removal: positive effect on query accuracy.

Application of feature reduction

- Face recognition
- Handwritten digit recognition
- Text mining
- Image retrieval
- Microarray data analysis
- Protein classification
- $\ldots$

What is Principal Component Analysis?

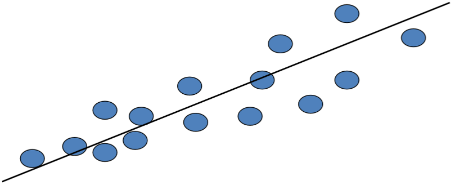
- Principal component analysis (PCA)
  - Reduce the dimensionality of a data set by finding a new set of variables, smaller than the original set of variables
  - Retains most of the sample's information.
  - Useful for the compression and classification of data.
- By information we mean the variation present in the sample, given by the correlations between the original variables.
  - The new variables, called principal components (PCs), are uncorrelated, and are ordered by the fraction of the total information each retains.

Principal components (PCs)

Given n points in a d dimensional space, for large d, how does one project on to a low dimensional space while preserving broad trends in the data and allowing it to be visualized?

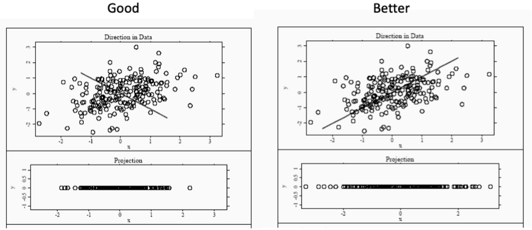
Geometric picture of principal components

- Given n points in a d dimensional space, for large d, how does one project on to a 1 dimensional space



- Choose a line that fits the data so the points are spread out well along the line

Let us see it on a figure



PCA 希望降维后信息损失最小，可以理解为投影后的数据尽可能的分开，这种分散程度可以用方差来表示 (μ 为均值)：

$$Var(a) = \frac{1}{n} \sum_{i=1}^{n} (a_i - \mu)^2$$

对数据进行中心化后，即 $\mu = 0$：

$$Var(a) = \frac{1}{n} \sum_{i=1}^{n} a_i^2$$

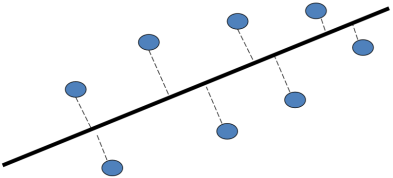
Geometric picture of principal components

对数据进行中心化:

$$\bar{\mathbf{x}} = \frac{1}{n}\sum_{i=1}^{n} x_i,$$
$$\mathbf{x}_i' = \mathbf{x}_i - \bar{\mathbf{x}}, \quad 1 \le i \le n.$$

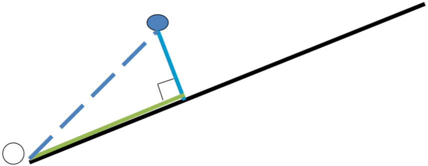
对于中心化以后的数据，即 $\bar{\mathrm{x}}' = 0$，以下说法等价: Find a line that

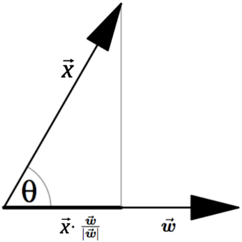
- maximize the variance of the projected data
- maximize the sum of squares of data samples' projections on that line
- minimize the sum of squares of distances to the line



Algebraic Interpretation — 1D

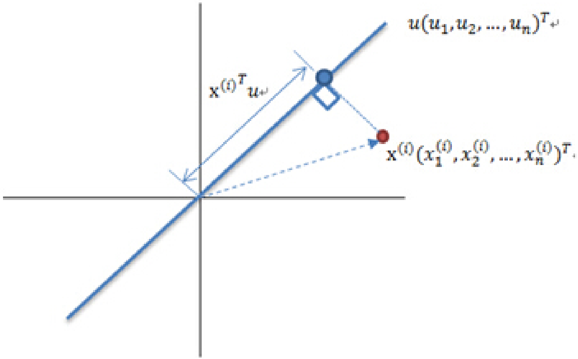
- Minimizing sum of squares of distances to the line is the same as maximizing the sum of squares of the projections on that line, thanks to Pythagoras (毕达哥拉斯).





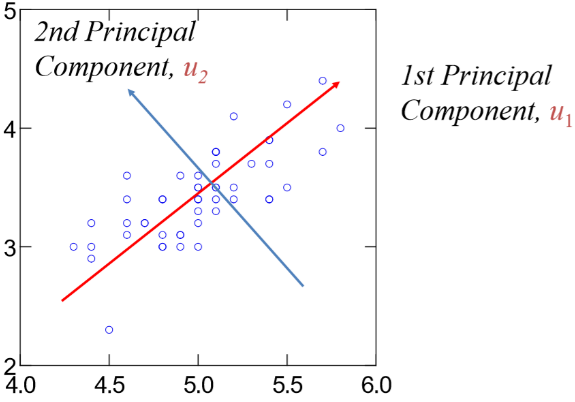
投影长度为: $\mathbf{x}^\top \frac{\mathbf{w}}{\|\mathbf{w}\|}$

Algebraic Interpretation — 1D



投影长度为: $\mathbf{x}^\top \mathbf{u} = \mathbf{u}^\top \mathbf{x}$ subject to $\mathbf{u}^\top \mathbf{u} = 1$

Geometric picture of principal components



Geometric picture of principal components

- the 1^{st} PC $\mathbf{u}_1$ is a minimum distance fit to a line in X space
- the 2^{nd} PC $\mathbf{u}_2$ is a minimum distance fit to a line in the plane perpendicular (垂直于) to the 1^{st} PC

PCs are a series of linear least squares fits to a sample, each orthogonal (垂直于) to all the previous.

Algebraic derivation of PCs

- Given a sample of n observations on a vector of d variables

$$\{\,\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\,\} \in R^d$$

- First project the data onto a one-dimensional space with a d -dimensional vector $\mathbf{u}_1$ (where $\mathbf{u}_1^\top \mathbf{u}_1 = 1$):

$$\left\{\mathbf{u}_1^\top \mathbf{x}_1, \mathbf{u}_1^\top \mathbf{x}_2, \cdots, \mathbf{u}_1^\top \mathbf{x}_n\right\}$$

- Find $\mathbf{u}_1$ to maximize the variance the projected data:

$$\frac{1}{n}\sum_{i=1}^{n}\left(\mathbf{u}_1^\top \mathbf{x}_i - \mathbf{u}_1^\top \bar{\mathbf{x}}\right)^2 = \mathbf{u}_1^\top S \mathbf{u}_1$$

Where $\bar{\mathbf{x}} = \frac{1}{n}\sum_{i=1}^{n} \mathbf{x}_i$ and $S = \frac{1}{n}\sum_{i=1}^{n} (\mathbf{x}_i - \bar{\mathbf{x}})(\mathbf{x}_i - \bar{\mathbf{x}})^\top$

Algebraic derivation of PCs

- To solve $\max_{\mathbf{u}_1} \mathbf{u}_1^\top S \mathbf{u}_1$ subject to $\mathbf{u}_1^\top \mathbf{u}_1 = 1$
- Let λ_1 be a Lagrangian multiplier (拉格朗日乘子)

$$L = \mathbf{u}_1^\top S \mathbf{u}_1 + \lambda_1 \left(1 - \mathbf{u}_1^\top \mathbf{u}_1\right)$$

$$\frac{\partial L}{\partial \mathbf{u}_1} = S\mathbf{u}_1 - \lambda_1 \mathbf{u}_1 = 0$$

$$S\mathbf{u}_1 = \lambda_1 \mathbf{u}_1$$

$\Rightarrow \mathbf{u}_1$ is an eigenvector (特征向量)

$$\mathbf{u}_1^\top S \mathbf{u}_1 = \lambda_1$$

$\Rightarrow \mathbf{u}_1$ corresponds to the eigenvector with the largest eigenvalue λ_1

- 即，$\max_{\mathbf{u}_1} \mathbf{u}_1^\top S \mathbf{u}_1$ subject to $\mathbf{u}_1^\top \mathbf{u}_1 = 1$ 的结果就是矩阵 S 的最大特征值
  - 矩阵 S 特征值计算方法：构造特征多项式 $|S - \lambda I| = 0$ (I 为单位矩阵)，特征值为线性方程组的解

Algebraic derivation of PCs

- To find the second component $\mathbf{u}_2$
- Solve the following

$$\max_{\mathbf{u}_2} \mathbf{u}_2^\top S \mathbf{u}_2 \quad \text{subject to} \quad \mathbf{u}_2^\top \mathbf{u}_2 = 1 \;\&\; \mathbf{u}_1^\top \mathbf{u}_2 = 0$$

  - $\mathbf{u}_2$ is the eigenvector with the second largest eigenvalue λ_2

...

Algebraic derivation of PCs

- Main steps for computing PCs
  - Calculate the covariance matrix S

$$S = \frac{1}{n}\sum_{i=1}^{n}(\mathbf{x}_i - \bar{\mathbf{x}})(\mathbf{x}_i - \bar{\mathbf{x}})^\top$$

or first center the data: $\{\mathbf{x}'_1, \mathbf{x}'_2, \ldots, \mathbf{x}'_n\}$ and $\bar{\mathbf{x}}' = 0$

$$\text{let } X = [\mathbf{x}'_1, \mathbf{x}'^\top_2, \ldots, \mathbf{x}'_n] \in R^{d\times n}; \text{ then } S = \frac{1}{n}XX^\top$$

  - Find the first m eigenvectors $\{\mathbf{u}_i\}_{i=1}^m$
  - Form the projection matrix

$$P = [\mathbf{u}_1 \ \mathbf{u}_2 \ \cdots \ \mathbf{u}_m] \in R^{d\times m}$$

  - A new test point can be projected as:

$$\mathbf{x}_{new} \in R^d \rightarrow P^\top \mathbf{x}_{new} \in R^m$$

Algebraic derivation of PCs

$$\mathbf{y} = P^\top \mathbf{x} \in R^m$$

- Getting the old data back?
  - If P is a square matrix (方阵), we can recover $\mathbf{x}$ by

    $$\mathbf{x} = \left(P^\top\right)^{-1} \mathbf{y} = P\mathbf{y} = PP^\top \mathbf{x}$$

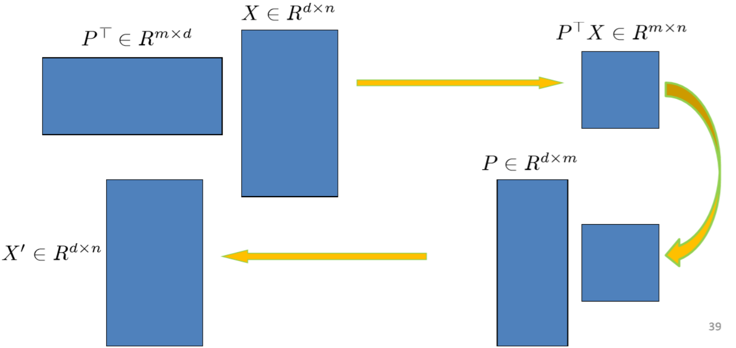
    注： $\mathbf{u}_i^\top \mathbf{u}_i = 1$ and $\mathbf{u}_i^\top \mathbf{u}_j = 0$ for $i \neq j$, then $P^\top P = I_m$ (where $m = n$) and $(P^\top)^{-1} = P$
  - Here P is not full ($m << d$), but we can still recover $\mathbf{x}$ by $\mathbf{x} = P\mathbf{y} = PP^\top \mathbf{x}$, and lose some information
- Objective:
  - Lose least amount of information

Optimality property of PCA

Reconstruction

Dimension reduction

$$X \in R^{d\times n} \rightarrow Y = P^\top X \in R^{m\times n} \rightarrow X' = PP^\top X \in R^{d\times n}$$



Optimality property of PCA

Main theoretical result:

The matrix P consisting of the first m eigenvectors of the covariance matrix S solves the following min problem:

$$\begin{aligned}
\arg\min_{P \in R^{d\times m}} \|X - X'\|^2 &= \arg\min_{P \in R^{d\times m}} \|X - PP^\top X\|^2 \\
&= \arg\max_{P \in R^{d\times m}} trace(X^\top PP^\top X) \\
&= \arg\max_{P \in R^{d\times m}} trace(P^\top XX^\top P) \\
&= \arg\max_{P \in R^{d\times m}} trace(P^\top SP) \\
\text{subject to} \quad & P^\top P = I_m
\end{aligned}$$

Reconstruction error

Notice that, for a matrix A $m \times n$ and B $n \times m$,

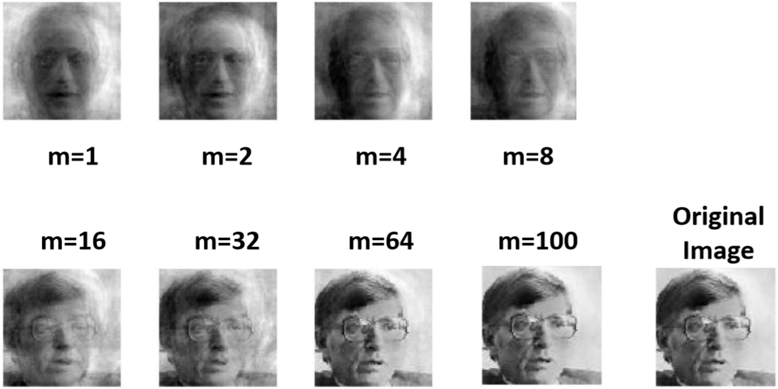
$trace(AB) = trace(BA) = \sum_{i=1}^{m} \sum_{j=1}^{n} a_{ij} b_{ji}$

$\arg\min_P \sum_{i=1}^{d} \sum_{j=1}^{n} (x_{ij} - x'_{ij})^2$ is equivalent to $\arg\max_P \sum_{i=1}^{d} \sum_{j=1}^{n} x_{ij} x'_{ij}$,

as $\sum_{i=1}^{d} \sum_{j=1}^{n} {x'_{ij}}^2 = trace((PP^T X)^T PP^T X) = trace(X^T PP^T X)$

PCA projection minimizes the reconstruction error among all linear projections of size m.

PCA for image compression



Nonlinear PCA using Kernels

Rewrite PCA in terms of dot products

- Assume the data has been centered, i.e., $\sum_i \mathrm{x}_i = 0$
- The covariance matrix S can be written as $S = \frac{1}{n}\sum_i \mathbf{x}_i \mathbf{x}_i^\top$
- If $\mathbf{u}$ is an eigenvector of S corresponding to nonzero eigenvalue

$$S\mathbf{u} = \frac{1}{n}\sum_i \mathbf{x}_i \mathbf{x}_i^\top \mathbf{u} = \lambda \mathbf{u} \;\Rightarrow\; \mathbf{u} = \frac{1}{n\lambda}\sum_i \left(\mathbf{x}_i^\top \mathbf{u}\right)\mathbf{x}_i$$

- Eigenvectors of S lie in the space spanned by all data points

Kernel methods:

- denote the representation of $\mathbf{x}$ as $\varphi(\mathbf{x})$
- define the kernel function $k : \mathbf{X} \times \mathbf{X} \to \mathbb{R}$ by $k(\mathbf{x}_i, \mathbf{x}_j) = \varphi(\mathbf{x}_i)^\top \varphi(\mathbf{x}_j)$
- define the kernel matrix K: $K_{ij} = k(\mathbf{x}_i, \mathbf{x}_j)$

Nonlinear PCA using Kernels

$$S\mathbf{u} = \frac{1}{n}\sum_i \mathbf{x}_i\mathbf{x}_i^\top \mathbf{u} = \lambda\mathbf{u} \;\Rightarrow\; \mathbf{u} = \frac{1}{n\lambda}\sum_i \left(\mathbf{x}_i^\top \mathbf{u}\right)\mathbf{x}_i$$

The covariance matrix can be written in matrix form

$$S = \frac{1}{n}XX^T, \text{ where } \mathrm{X} = [\mathrm{x}_1, \mathrm{x}_2, \dots, \mathrm{x}_n].$$

$$\mathbf{u} = \sum_i \alpha_i \mathbf{x}_i = X\boldsymbol{\alpha} \qquad S\mathbf{u} = \frac{1}{n}XX^T X\boldsymbol{\alpha} = \lambda X\boldsymbol{\alpha}$$

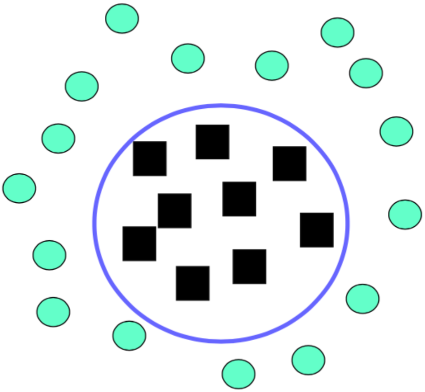
$$\frac{1}{n}(X^T X)(X^T X)\boldsymbol{\alpha} = \lambda (X^T X)\boldsymbol{\alpha}$$

Any benefits?

$$\frac{1}{n}(X^T X)\boldsymbol{\alpha} = \lambda\boldsymbol{\alpha}$$

$$\frac{1}{n}K\boldsymbol{\alpha} = \lambda\boldsymbol{\alpha}$$

Nonlinear PCA

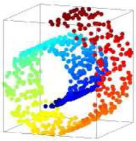


Linear projections will not detect the pattern.

Comments on PCA

- Linear dimensionality reduction method
- Can be kernelized
- Many nonlinear dimensionality reduction methods (Isomap, graph Laplacian eigenmap, and locally linear embedding/LLE) can be described as kernel PCA with a special kernel

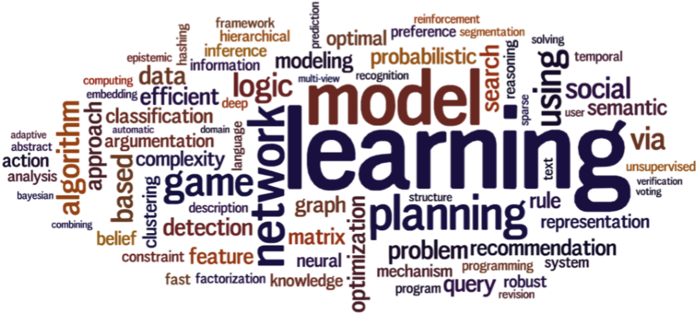
- Non-convex optimization problem
- But easy to solve $\cdots$



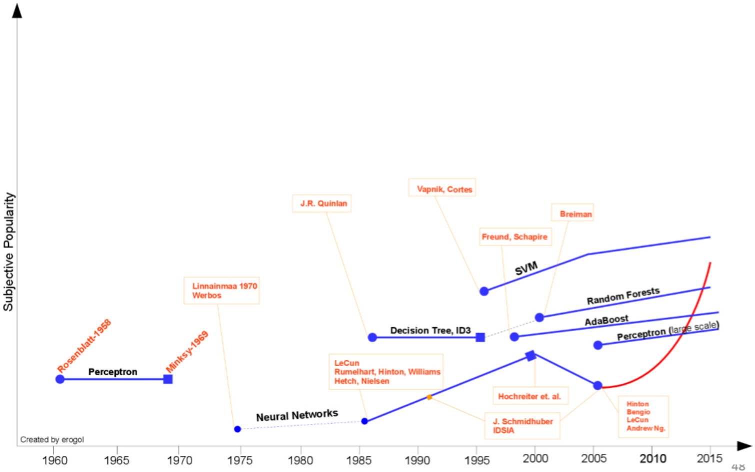
Want to Learn More?

- Machine Learning: a Probabilistic Perspective, K. Murphy
- Pattern Classification, R. Duda, P. Hart, and D. Stork. Standard pattern recognition textbook. Limited to classification problems. Matlab code. http://rii.ricoh.com/~stork/DHS.html
- Pattern recognition and machine learning. C. Bishop
- The Elements of statistical Learning: Data Mining, Inference, and Prediction. T. Hastie, R. Tibshirani, J. Friedman, Standard statistics textbook. Includes all the standard machine learning methods for classification, regression, clustering. R code. http://www-stat-class.stanford.edu/~tibs/ElemStatLearn/
- Introduction to Data Mining, P.-N. Tan, M. Steinbach, V. Kumar. AddisonWesley, 2006
- Principles of Data Mining, D. Hand, H. Mannila, and P. Smyth. MIT Press, 2001
- 统计学习方法，李航

Machine Learning in AI



Machine Learning History



Summary

- Supervised learning
  - Learning Decision Trees
  - K Nearst Neighbor Classfier
  - Linear Predictions
  - Support Vector Machines
- Unsupervised learning
  - Clustering
  - Principle Component Analysis

作业

- K-means 算法是否一定会收敛？如果是，给出证明过程；如果不是，给出说明。