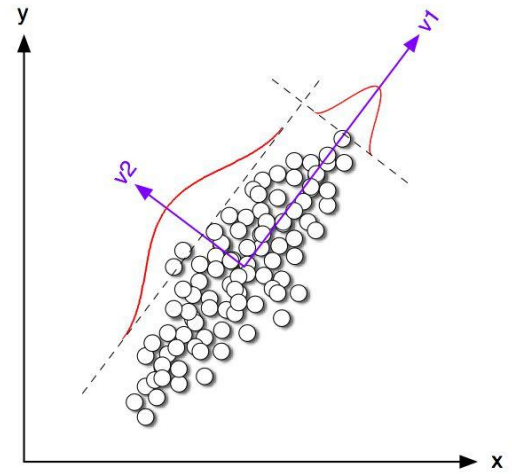
生物统计学：
生物信息中的概率统计模型

2020年秋



有关信息

- 授课教师：宁康
  - Email: ningkang@hust.edu.cn
  - Office: 华中科技大学东十一楼504室
  - Phone: 87793041, 18627968927

- 课程网页
  - http://www.microbioinformatics.org/teach/#
  - QQ群：182996651

课程安排

- 生物背景和课程简介
- 传统生物统计学及其应用
- 生物统计学和生物大数据挖掘
  - Hidden Markov Model (HMM)及其应用
    - Markov Chain
    - HMM理论
    - HMM和基因识别 (Topic I)
    - HMM和序列比对 (Topic II)
  - 进化树的概率模型 (Topic III)
  - Motif finding中的概率模型 (Topic IV)
    - EM algorithm
    - Markov Chain Monte Carlo (MCMC)
  - 基因表达数据分析 (Topic V)
    - 聚类分析-Mixture model
    - Classification-Lasso Based variable selection
  - 基因网络推断 (Topic VI)
    - Bayesian网络
    - Gaussian Graphical Model
  - 基因网络分析 (Topic VII)
    - Network clustering
    - Network Motif
    - Markov random field (MRF)
  - Dimension reduction及其应用 (Topic VIII)
- 面向生物大数据挖掘的深度学习

研究对象:
生物序列,
进化树,
生物网络,
基因表达
...

方法:
生物计算与生物统计

第9章: Dimension Reduction

- Feature selection
- PCA, LDA, CCA, SVD
- MDS, tSNE
- LASSO
- Others

Dimension Reduction

- Inputs: high dimensional

$$X_1, X_2, \cdots, X_n \in R^D$$

- Outputs: low dimensional

$$Y_1, Y_2, \cdots, Y_n \in R^d, d << D$$

- Constraints: Nearby points remain nearby, distant points remain distant.

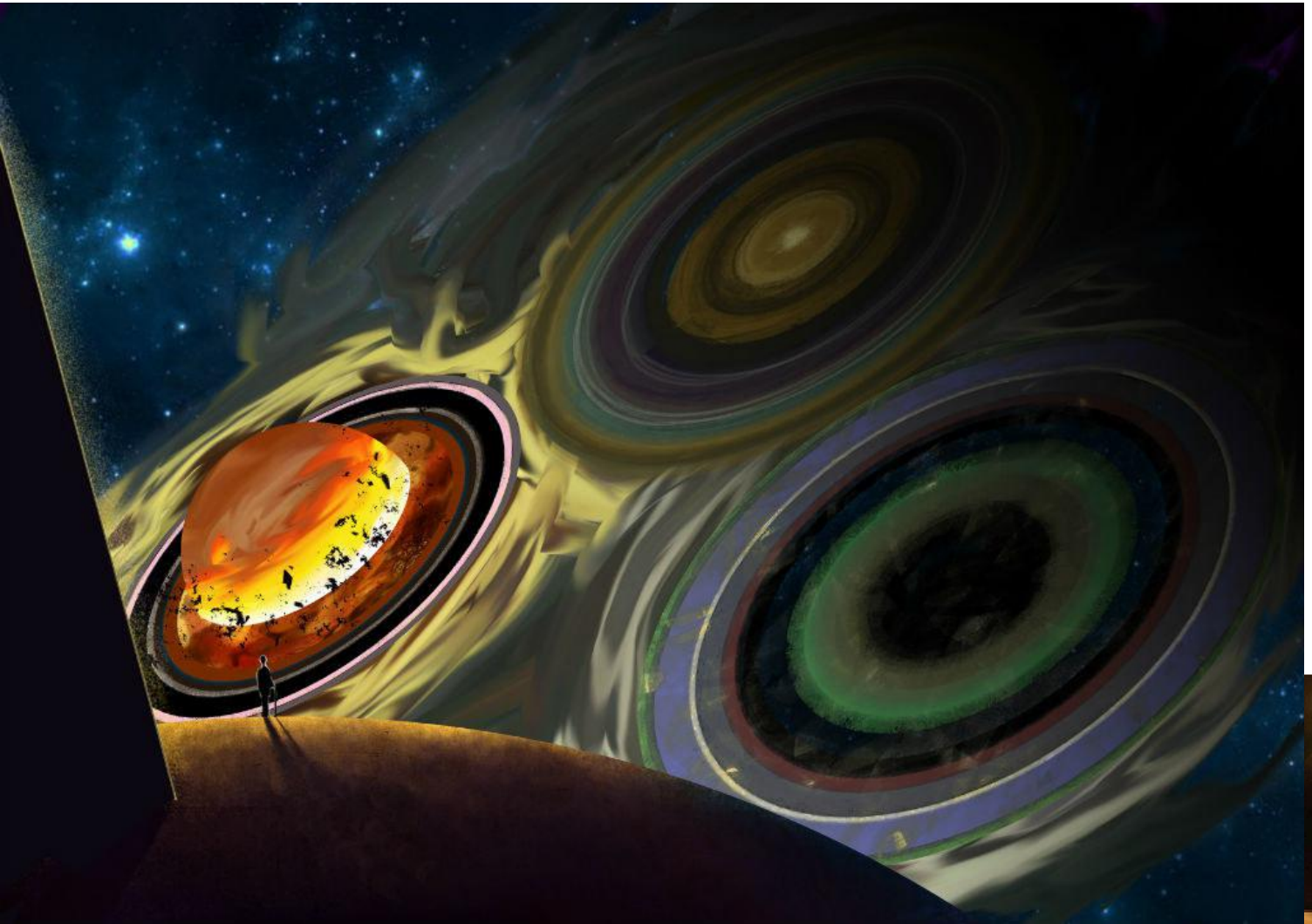
Curse of Dimensionality

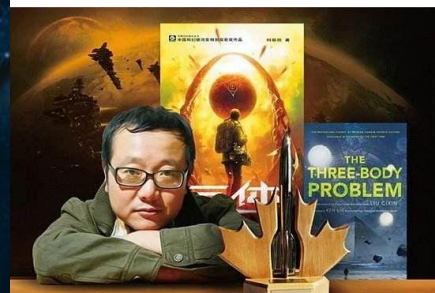
- A major problem is *the curse of dimensionality.*
- If the data x lies in high dimensional space, then an enormous amount of data is required to learn distributions or decision rules.
- Example: 50 dimensions. Each dimension has 20 levels. This gives a total of 20^{50} cells. But the no. of data samples will be far less. There will not be enough data samples to learn.

Curse of Dimensionality

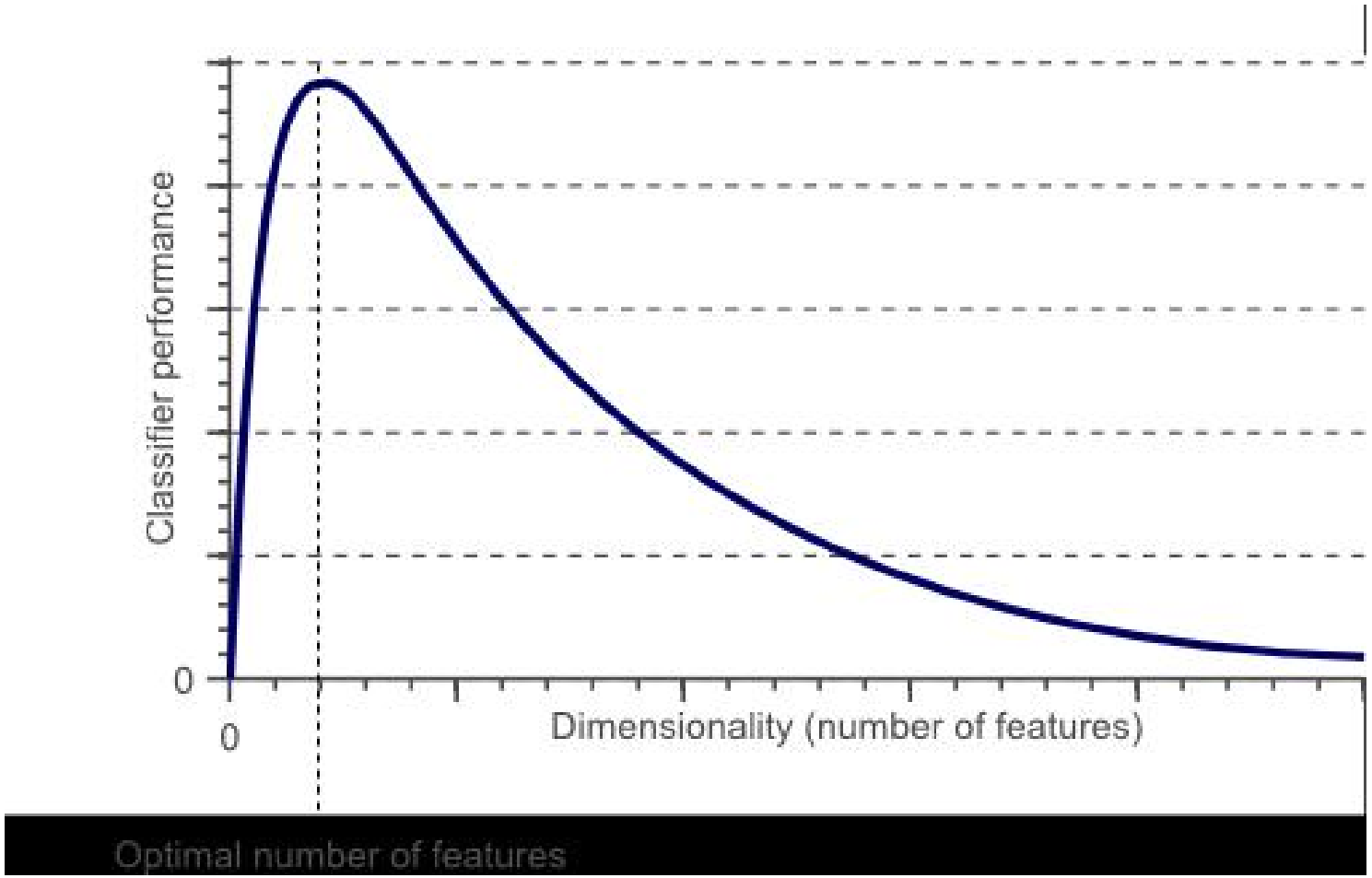
- One way to deal with dimensionality is to assume that we know the form of the probability distribution.
- For example, a Gaussian model in N dimensions has N + N(N-1)/2 parameters to estimate.
- Requires $O(N^2)$ data to learn reliably. This may be practical.

Curse of Dimensionality

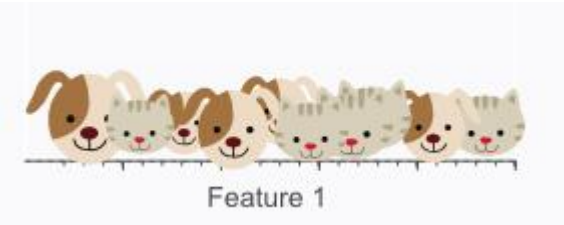


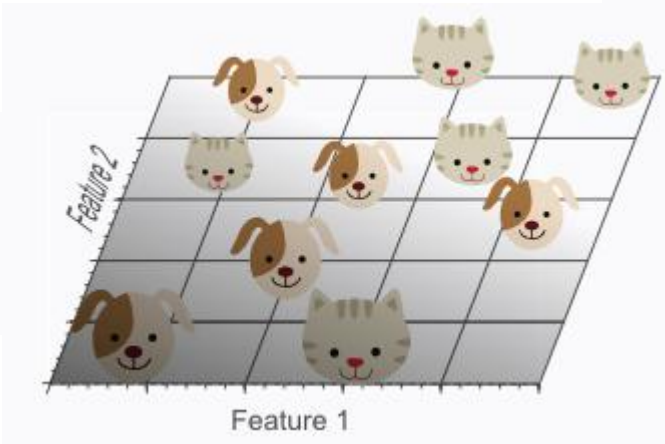


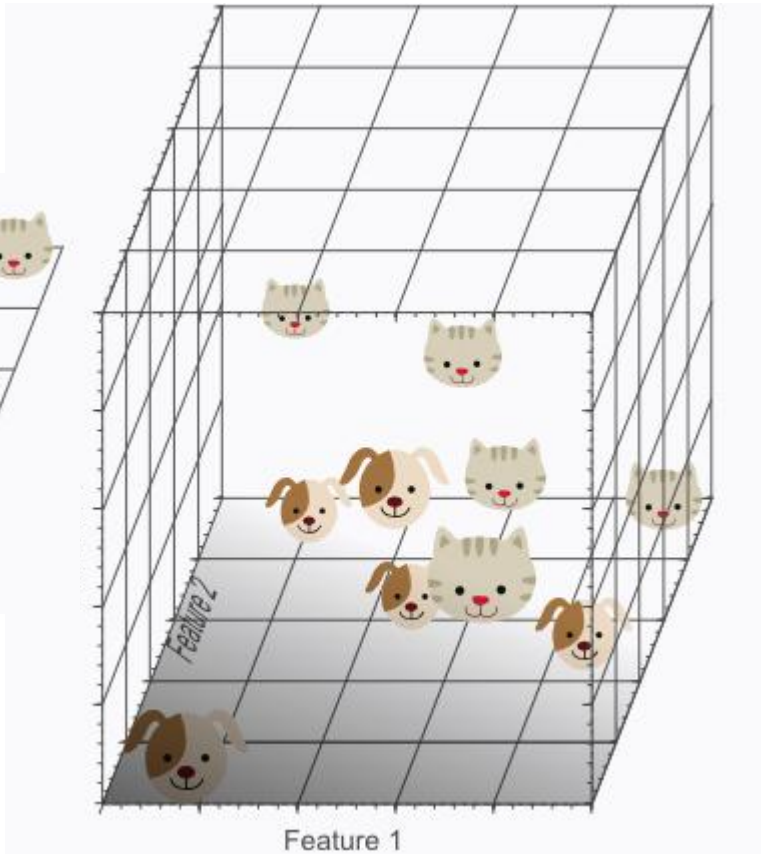
Curse of Dimensionality



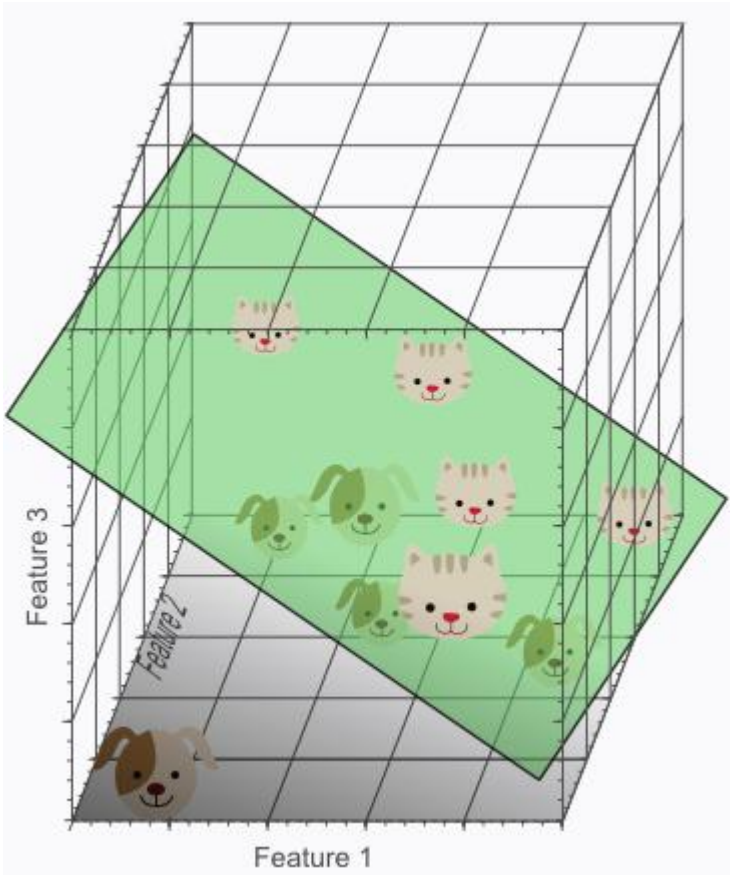
Curse of Dimensionality

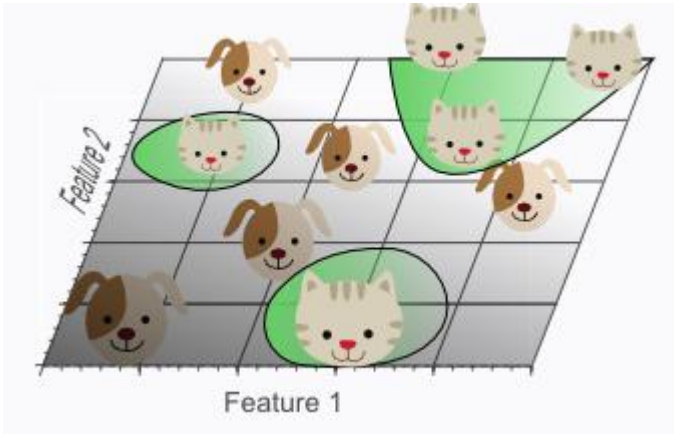


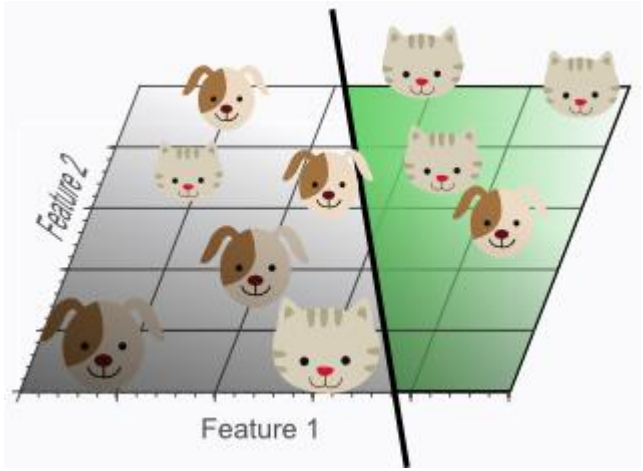




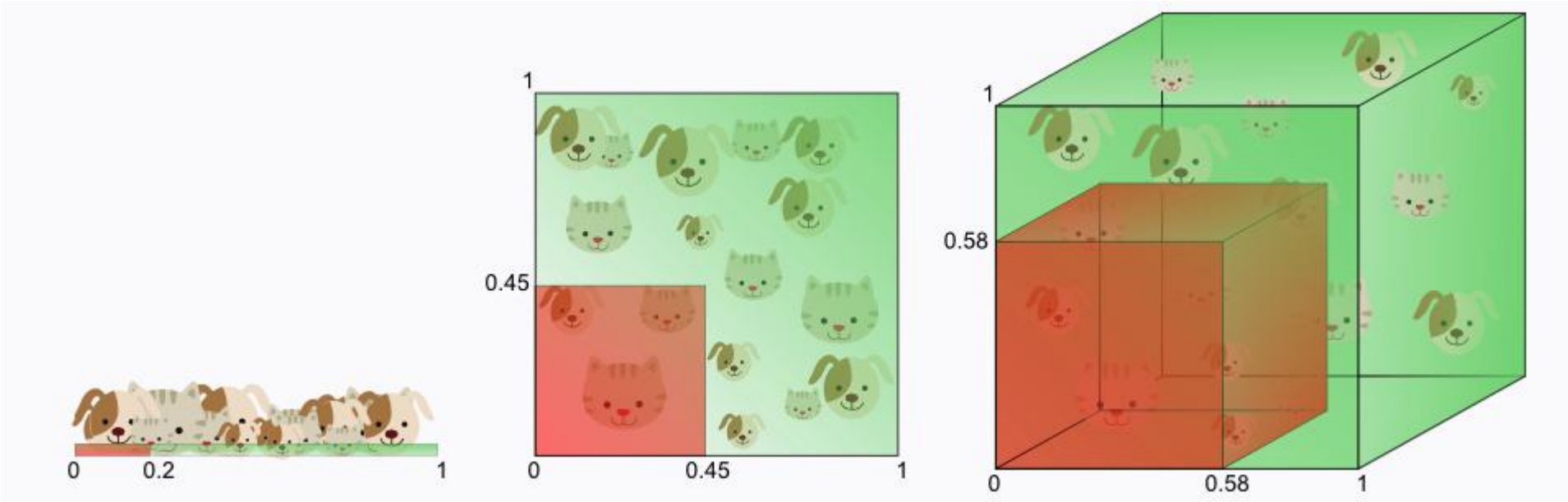
Curse of Dimensionality



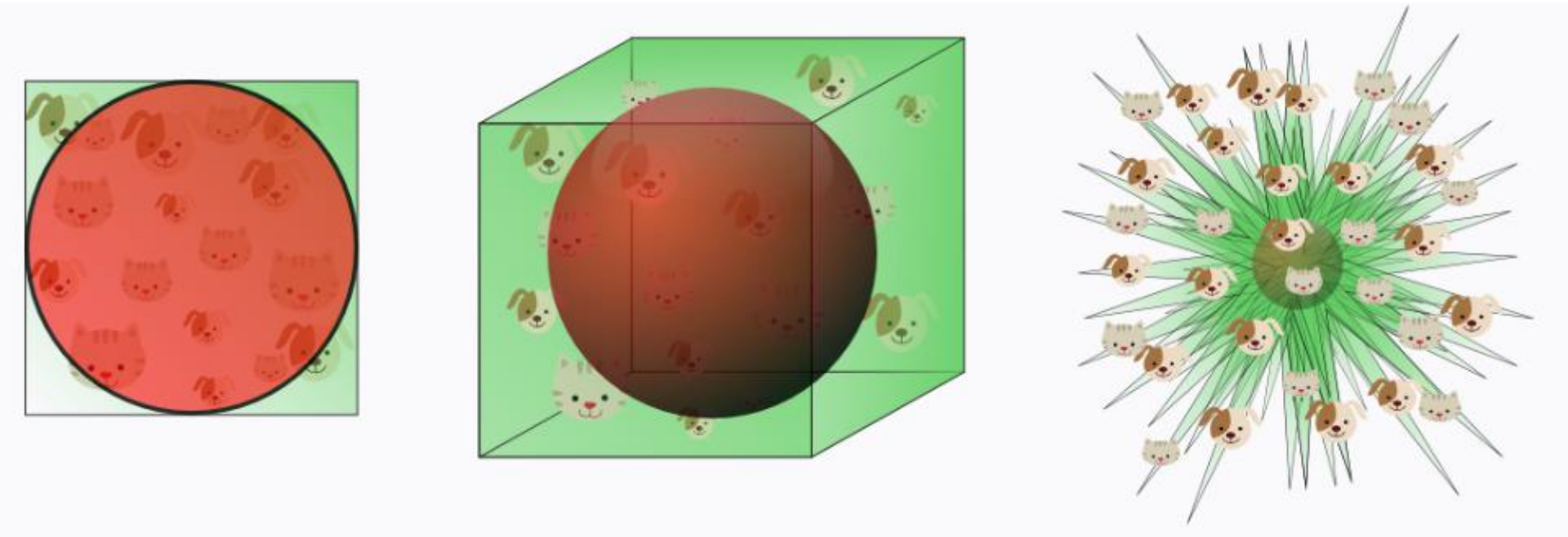




Curse of Dimensionality



Curse of Dimensionality



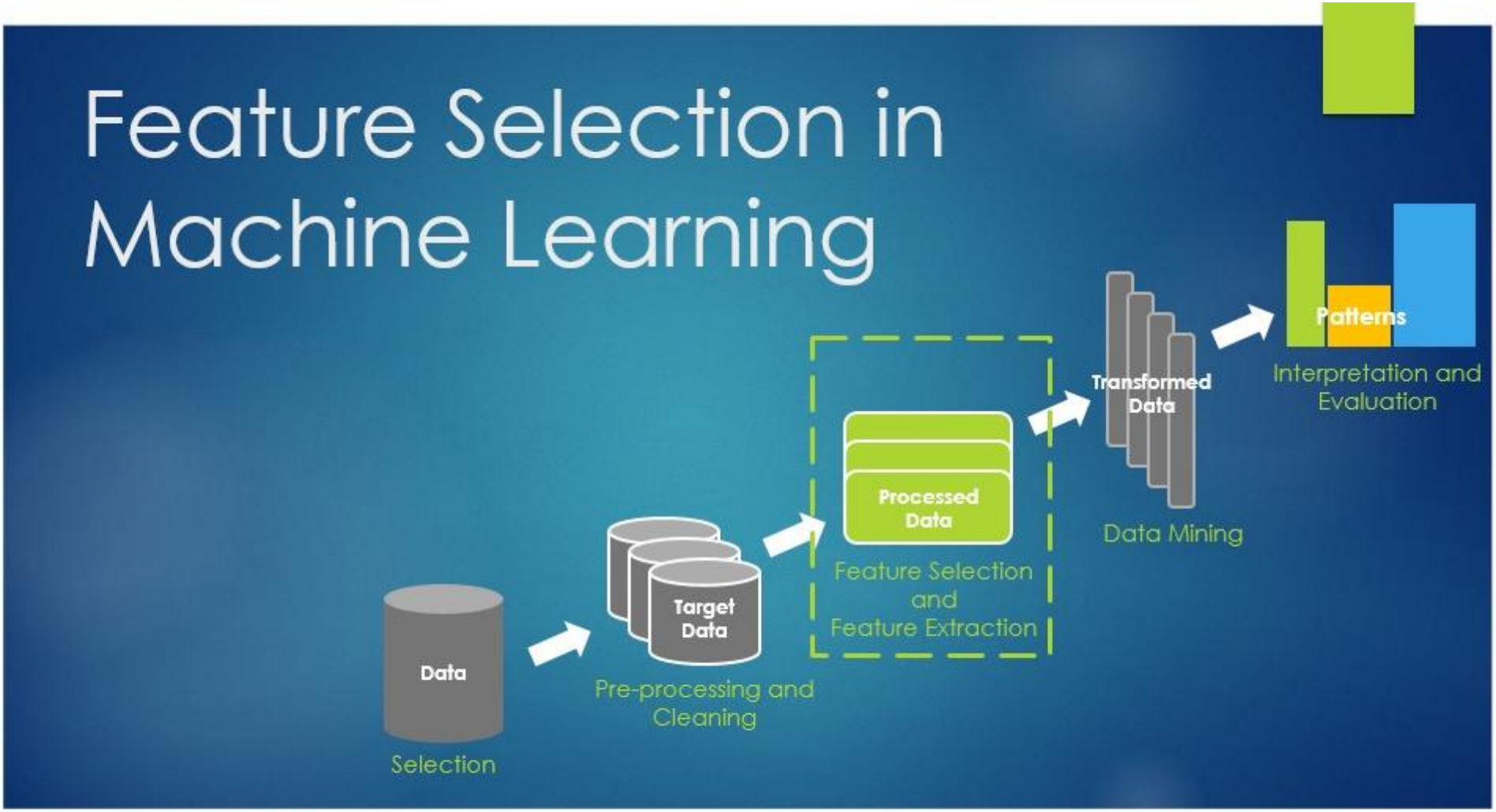
Curse of Dimensionality

Solutions:

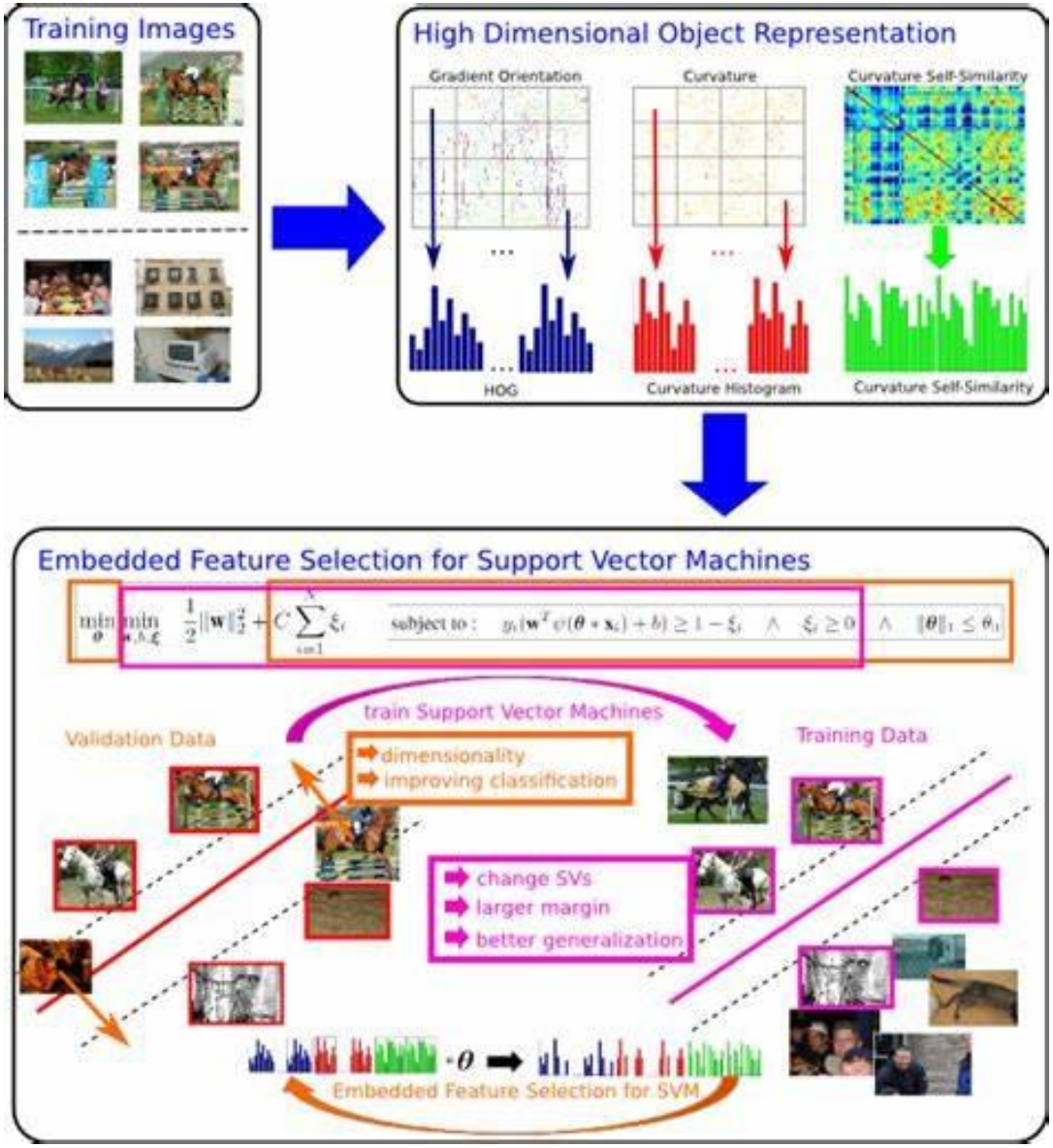
- Feature seleciton

- Projection of data onto a lower-dimensional space

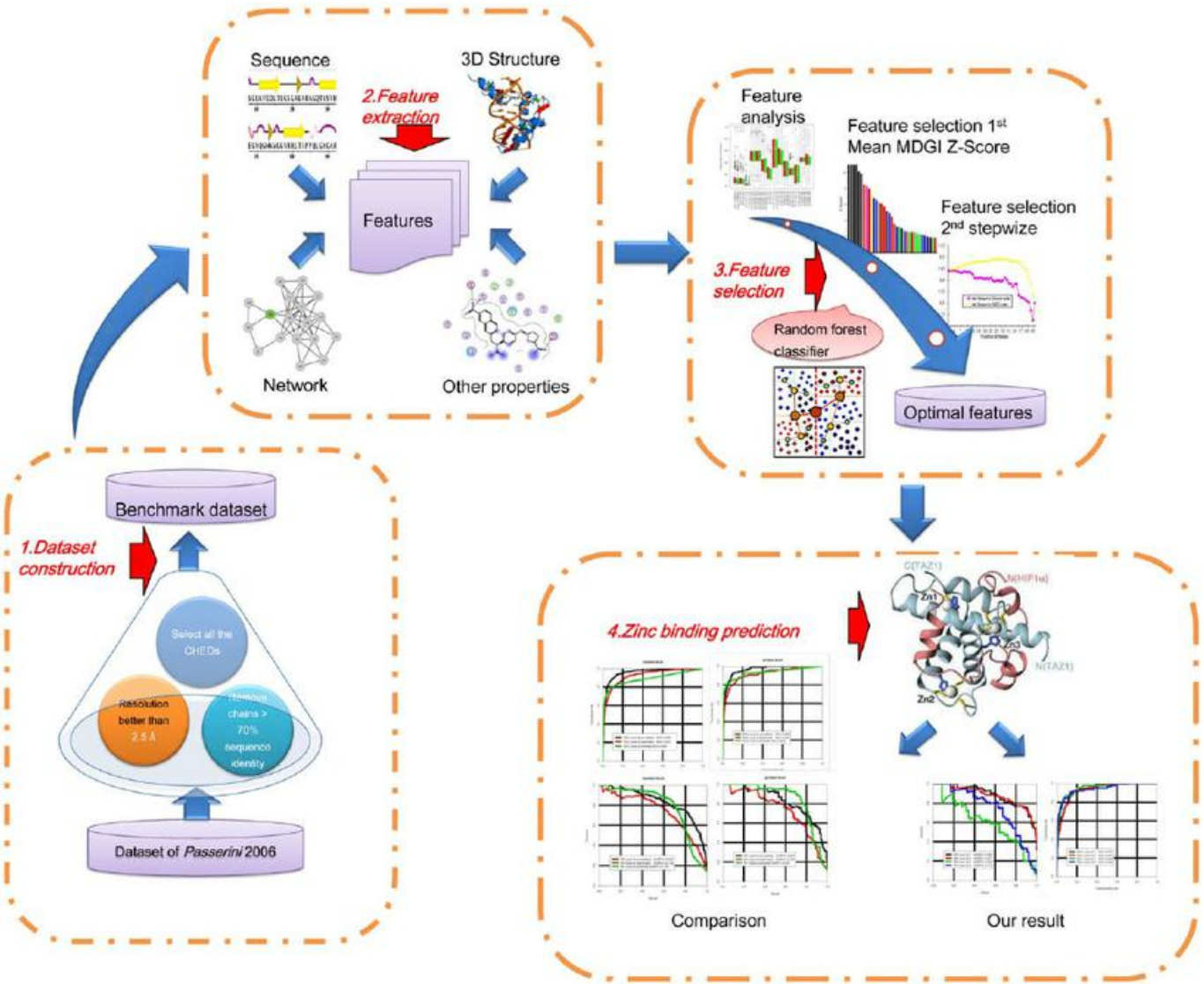
Feature seleciton



Feature seleciton

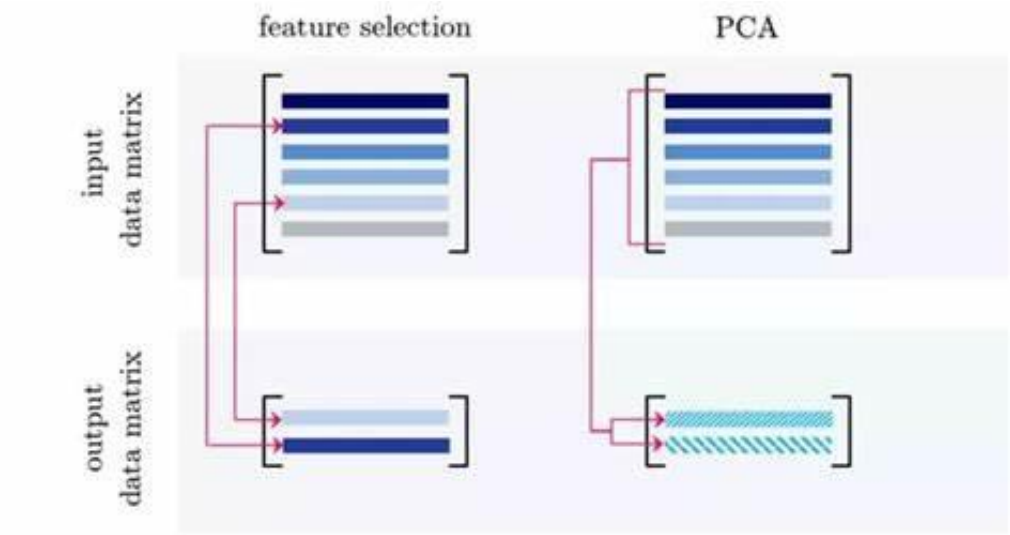


Feature seleciton



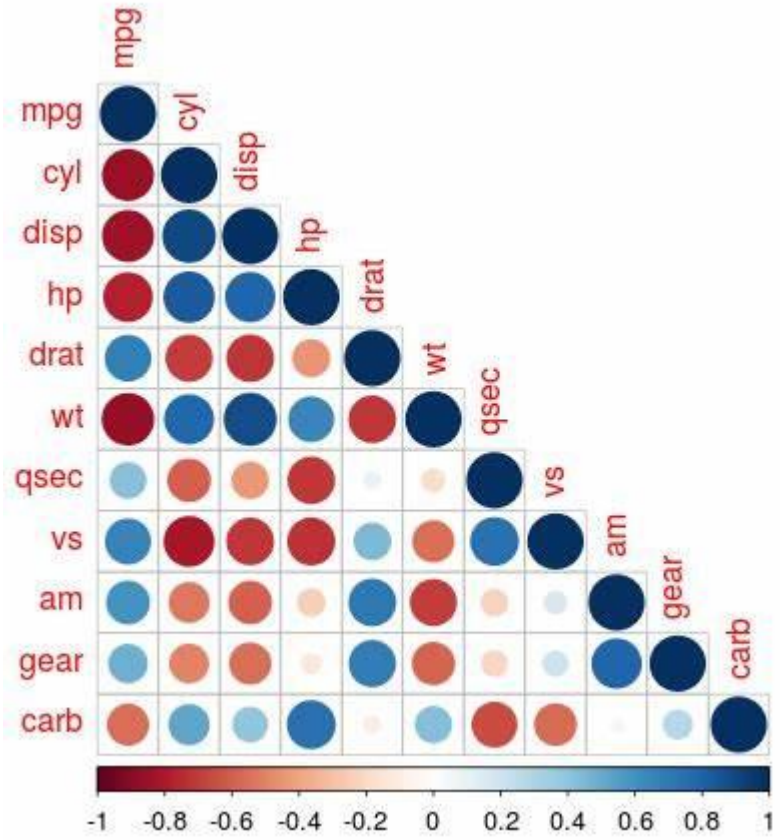
Feature seleciton

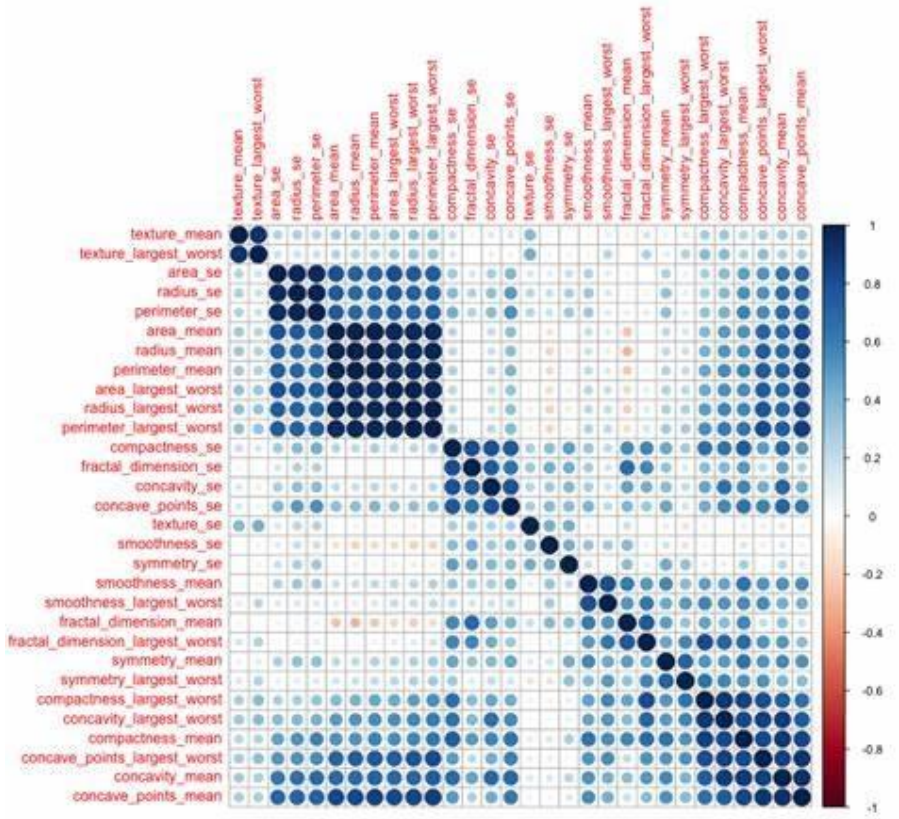
Reason 1: representativeness of features



Feature seleciton

Reason 2: reduction of redundant features





Feature seleciton

下面是特征选择/稀疏性非常有用的一些例子：

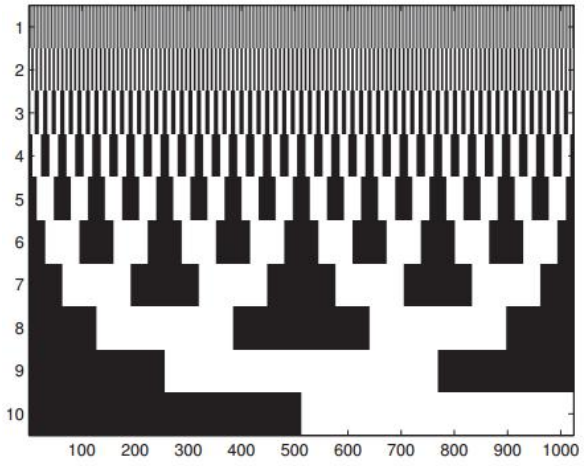
- 在许多问题中，我们的特征的数目D远大于我们的训练样本数N。这个就叫做N小D大问题，这样的问题很容易导致过拟合。所以我们就要降低特征也就是D的数目。而且在这个时代这样的现象越来越多，因为我们的传感器越来越丰富，所能够获得的信息的维度越来越高。

- 在信号处理中，常用小波基函数表示信号(图像、语音等)。为了节省时间和空间，找到信号的稀疏表示形式是有用的，这种稀疏表示形式是用少量的这种基函数表示的。这使我们能够从少量的测量中估计信号，以及压缩信号。

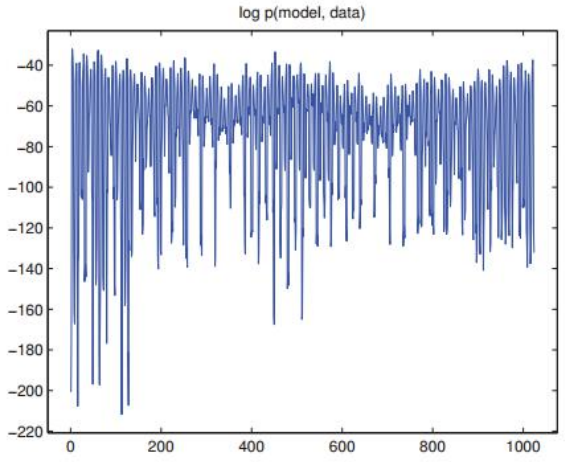
因此我们可以看到特征选择和稀疏性是当前机器学习/统计中最活跃的领域之一。

Feature seleciton

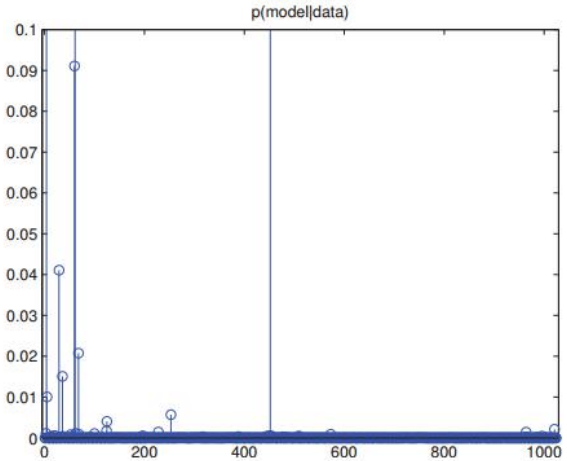
Subset selection



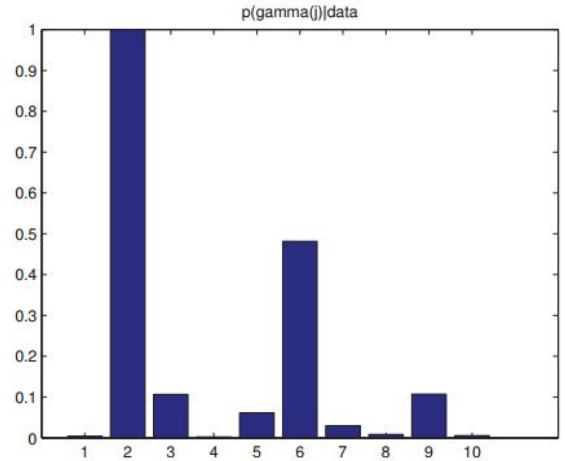
(a)



(b)



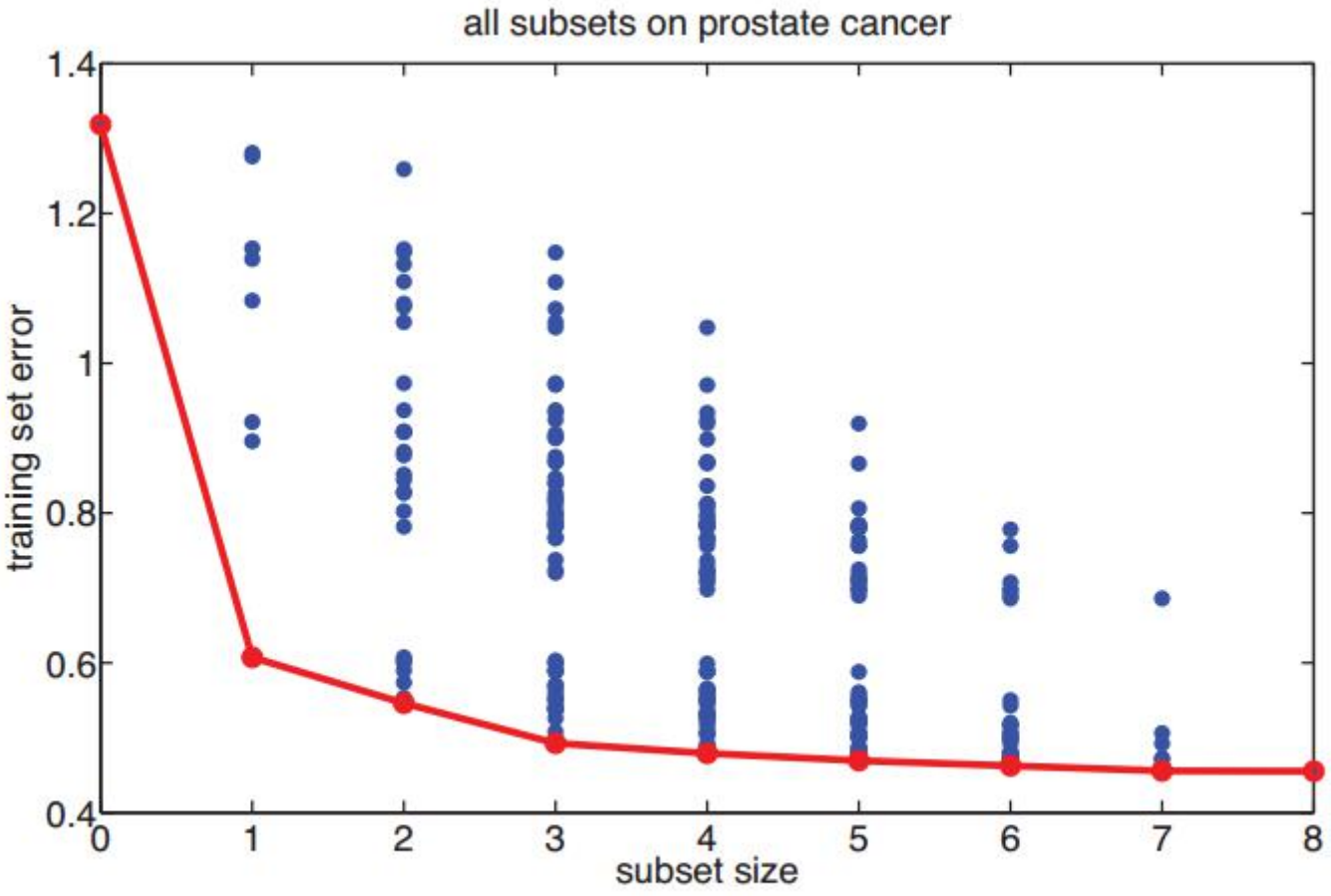
(c)



(d)

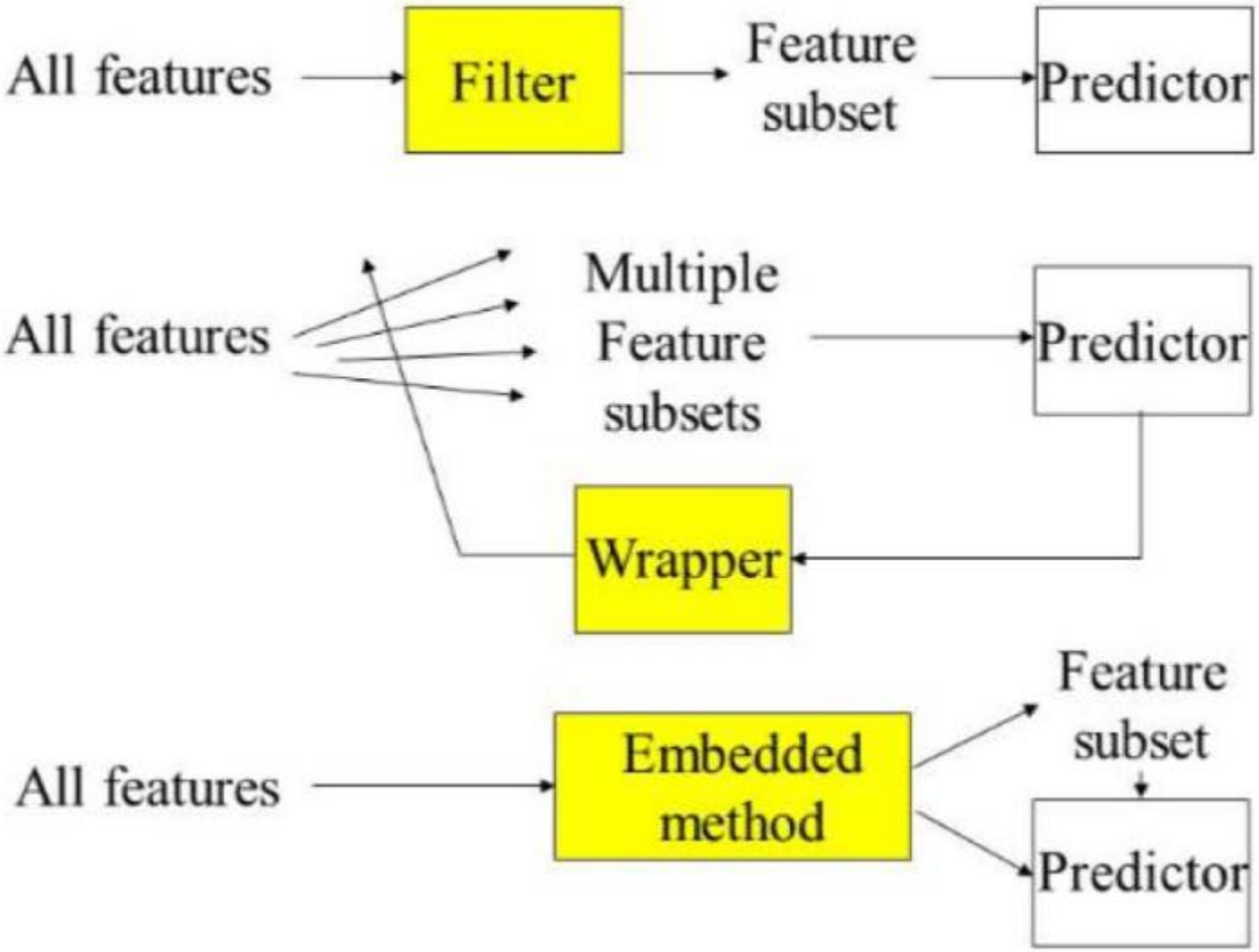
model	prob	members
4	0.447	2,
61	0.241	2, 6,
452	0.103	2, 6, 9,
60	0.091	2, 3, 6,
29	0.041	2, 5,
68	0.021	2, 6, 7,
36	0.015	2, 5, 6,
5	0.010	2, 3,

Feature seleciton



(b)

Feature seleciton



Feature seleciton

- mRMR (minimum Redundancy Maximum Relevance Feature Selection)
- Random forest

Feature seleciton

mRMR (minimum Redundancy Maximum Relevance Feature Selection)

[Frequently Asked Questions | Online Version | C/C++ Version with Source Codes | Matlab Version | Sample Data Sets | Major Publications | BIBM'07 Tutorial Slides]

- **Online Version**: You can run the program using your own data through the following form, -- you can also download the program and run on your own machines (see links below).

  * Data file (Standard CSV file format, where each row is a sample and each column is a variable/attribute/feature. MAKE SURE YOUR DATA IS SEPARATED BY COMMA, BUT NOT BLANK SPACE OR OTHER CHARACTERS!! The first row must be the feature names, and the first column must be the classes for samples. You may download a testing example data set here, which is microrray data of lung cancer (7 classes). The data has been discretized as 3-states. Note that the web-based program can only accept a data file with the maximum size 2M bytes, and maximum number of variables = 10000 -- if you have a larger data set, you should download the program and run on your own machine (see download links below).

  选择文件 未选择任何文件

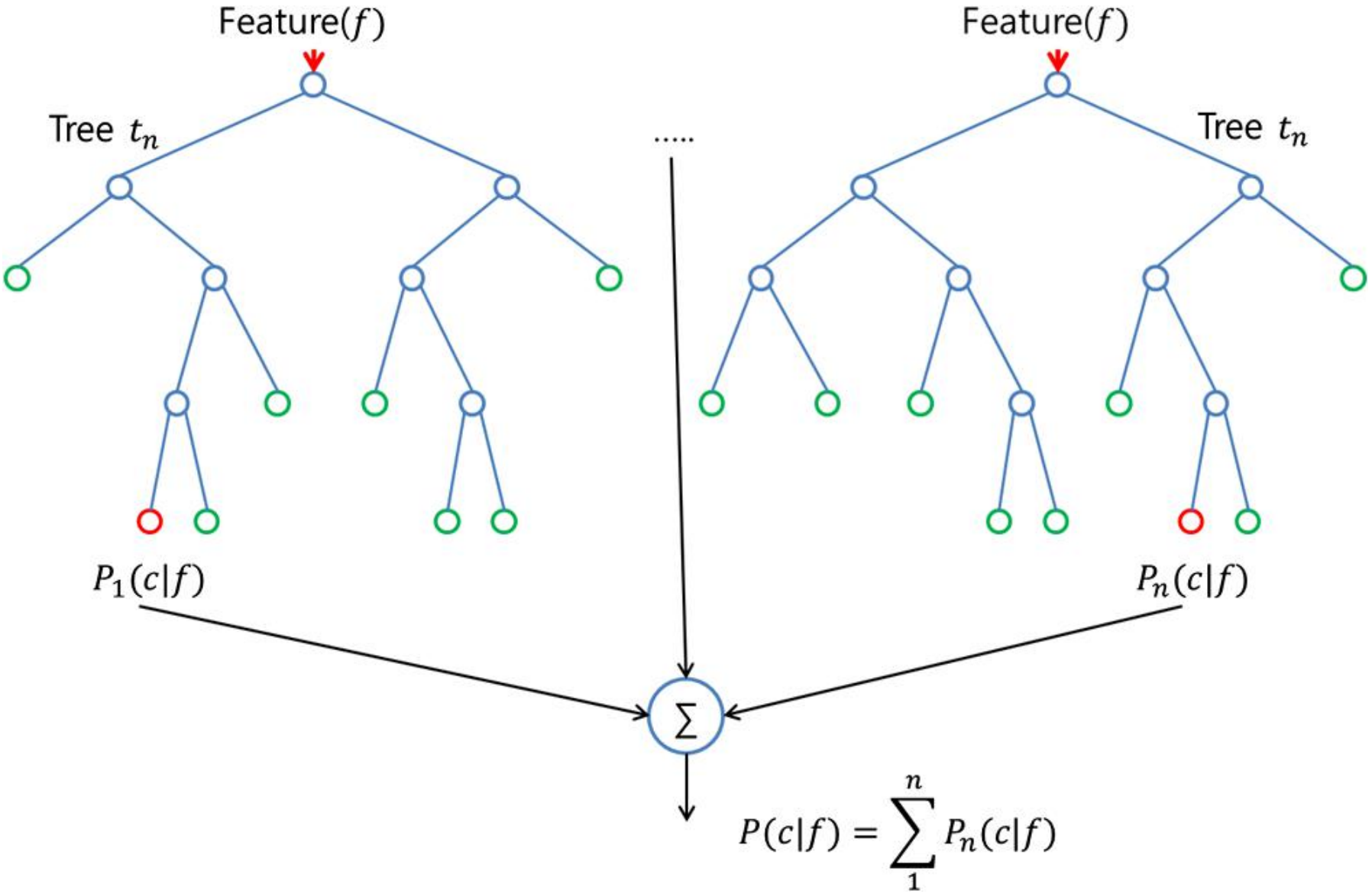
  * What is the feature selection scheme you want to use:

  MID (Mutual Information Difference) ▾

  * How many features you want to select:

  50 ▾

Random forest



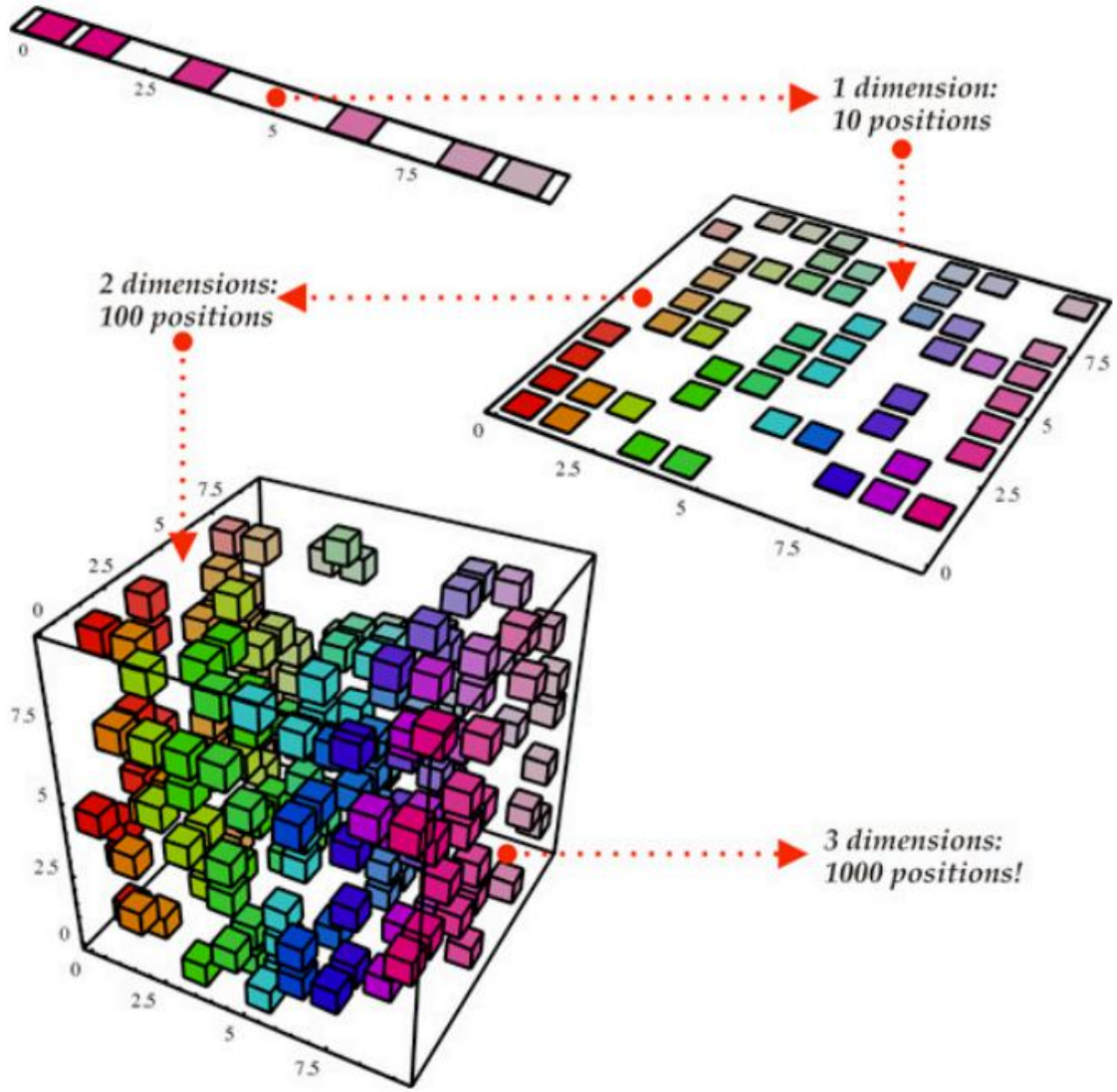
Curse of Dimensionality

Solutions:

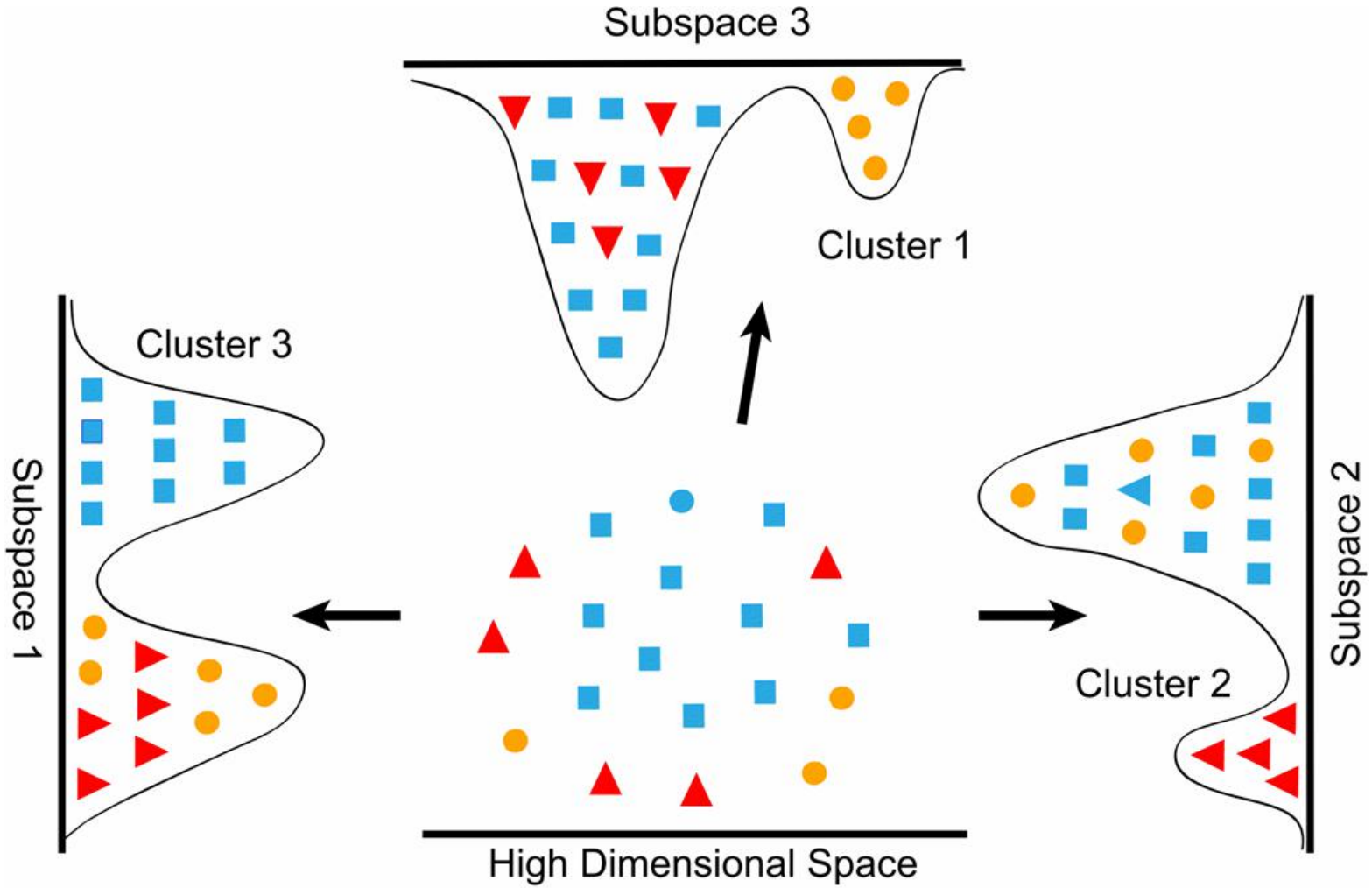
- Feature seleciton
- Projection of data onto a lower-dimensional space

Dimension reduction illusions

Dimension Reduction

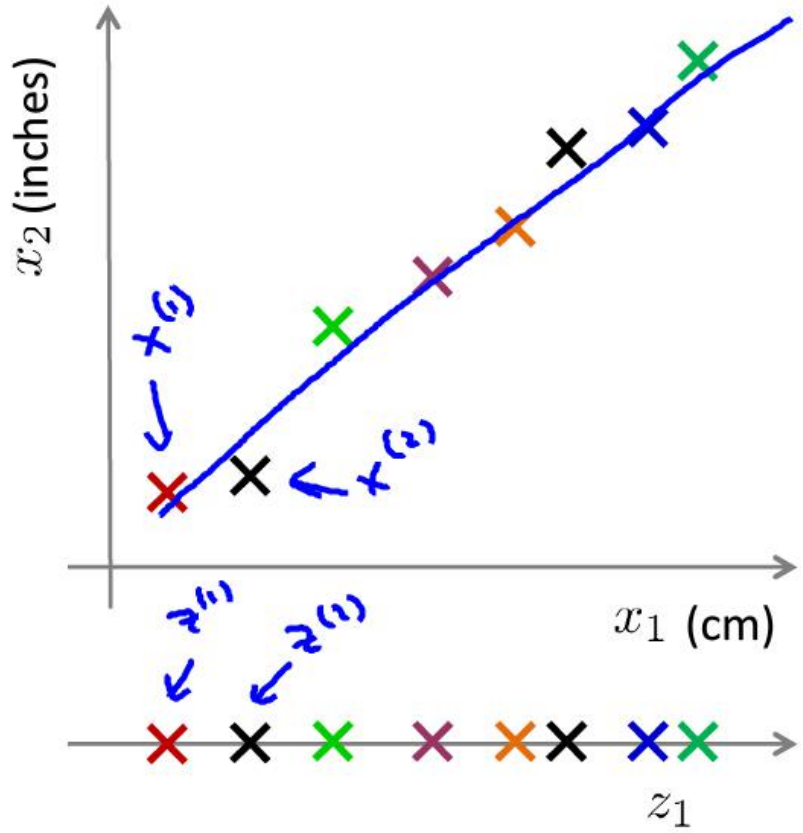


Dimension Reduction



Dimension Reduction

Data Compression



Reduce data from 2D to 1D

$x^{(1)} \in \mathbb{R}^2 \rightarrow z^{(1)} \in \mathbb{R}$

$x^{(2)} \in \mathbb{R}^2 \rightarrow z^{(2)} \in \mathbb{R}$

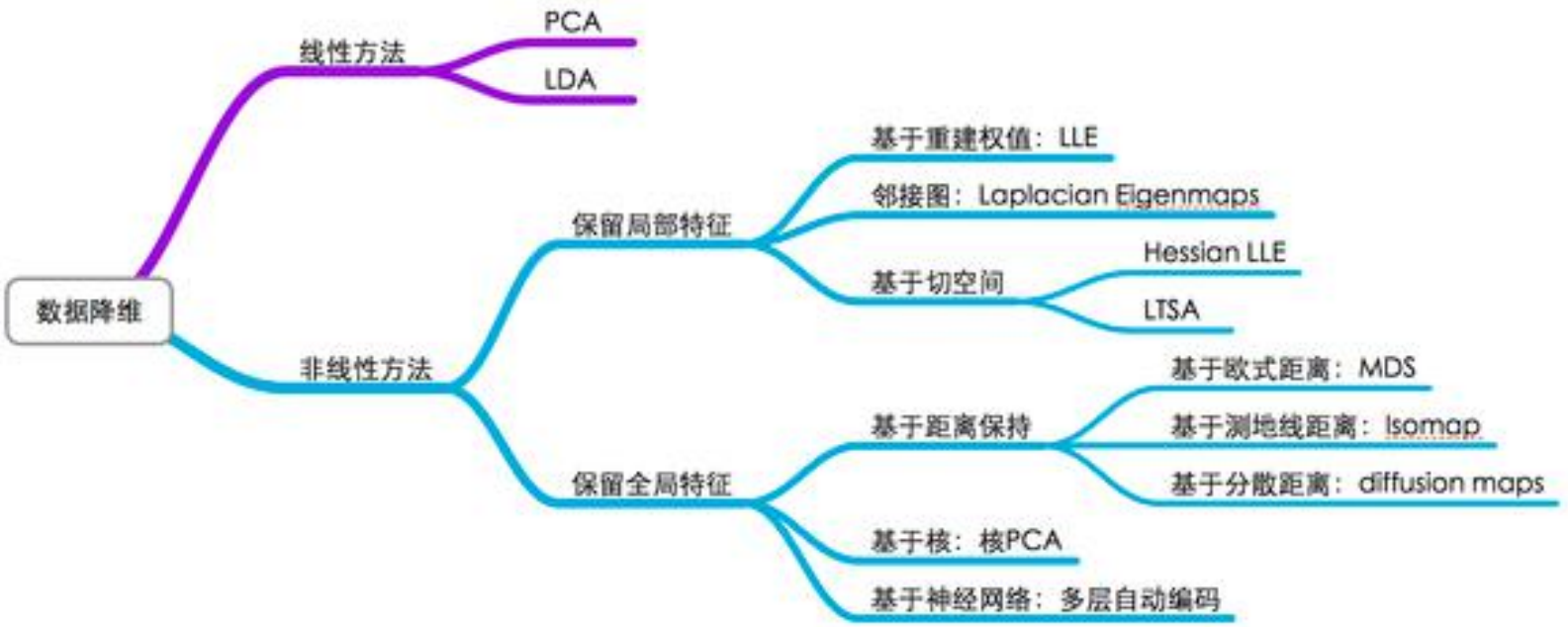
$\vdots$

$x^{(m)} \in \mathbb{R}^2 \rightarrow z^{(m)} \in \mathbb{R}$

Dimension Reduction

- One way to avoid the curse of dimensionality is by projecting the data onto a lower-dimensional space.
- Techniques for dimension reduction:
  - Principal Component Analysis (PCA)
  - Singular value decomposition (SVD)
  - Multi-dimensional Scaling (MDS).

Dimension Reduction



（1）线性降维：PCA、ICA、LDA、LFA、LPP

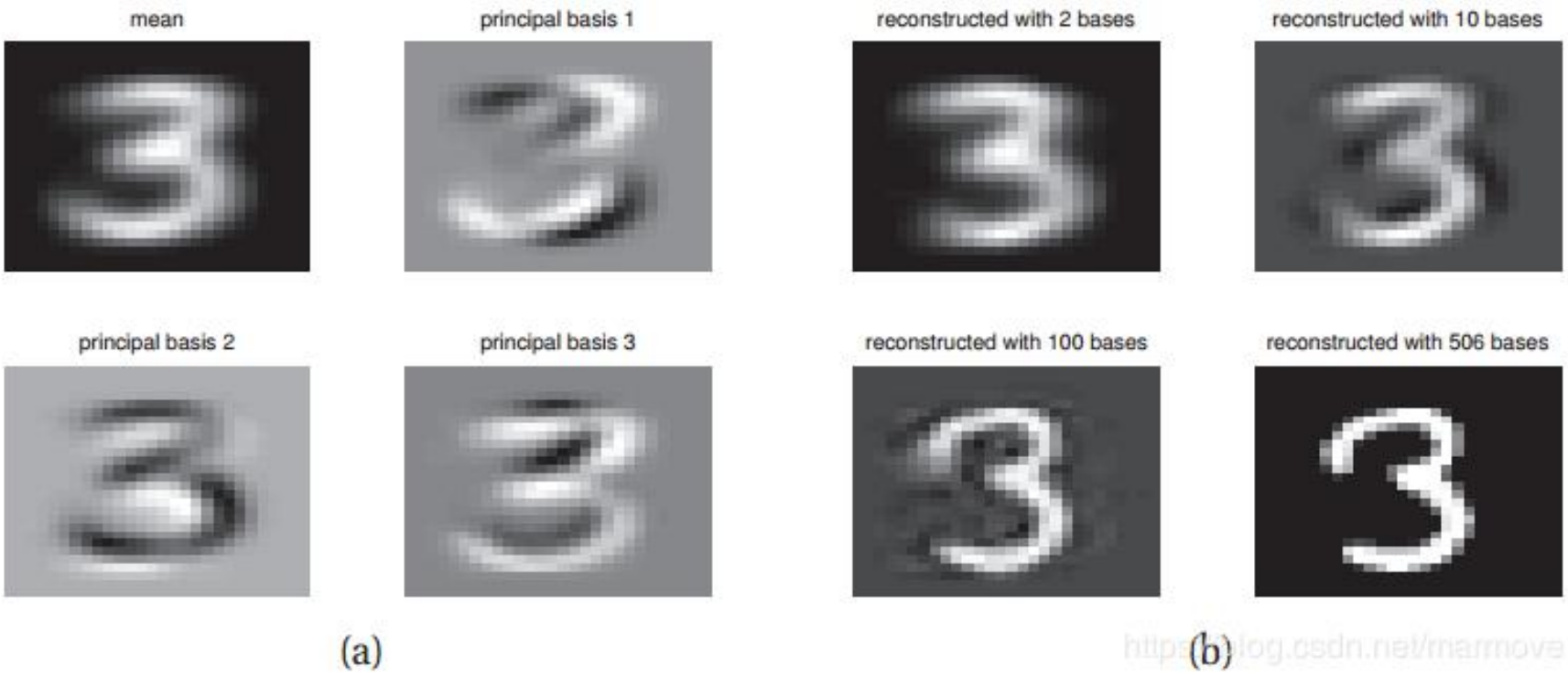
（2）非线性降维方法：

①基于核函数的方法：KPCA、KICA、KDA

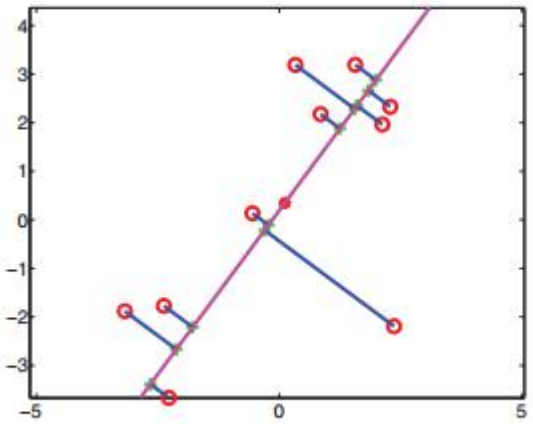
②基于特征值的方法：ISOMAP、LLE、LE、LPP、LTSA、MVU

Principle Component Analysis (PCA)

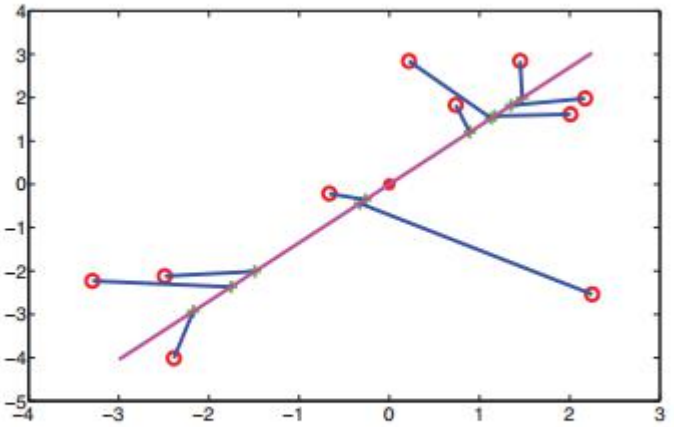
Dimension Reduction & PCA



Principal Component



(a)



(b)

Dimension Reduction & PCA

- PCA reduce dimensions of data without much loss of information.

- Used in machine learning and in signal processing and image compression (among other things).

Variance and Covariance

- Let X, Y are random variable

$$Var(X) = E(X^2) - E(X)^2$$
$$Cov(X, Y) = E[(X - EX)(Y - EY)]$$
$$= E(XY) - E(X)E(Y)$$

Multivariate Random Variable

- For multidimensional random vector

$$\vec{X} = \begin{pmatrix} X_1 \\ X_2 \\ \vdots \\ X_p \end{pmatrix}$$

Variance and Covariance

- Variance of the multivariate variable (commonly referred to as covariance)

$$Var(\vec{X}) = E[(\vec{X} - E(\vec{X})(\vec{X} - E(\vec{X})^T]$$

- For simplicity, we simply note as

$$Var(X) = E[(X - E(X))(X - E(X))^T]$$

Projection to 1D

- Let $Z = \alpha^T X$ be the projection, which is a random variable (1D)

$$
\begin{aligned}
E(Z) &= \alpha^T E(X) \\
Var(Z) &= E(Z^2) - E(Z)^2 \\
&= E[(\alpha^T X)(\alpha^T X)^T]) - [(\alpha^T E(X))(\alpha^T E(X))^T] \\
&= \alpha^T [E(XX^T) - E(X)E(X)^T]\alpha \\
&= \alpha^T Var(X)\alpha
\end{aligned}
$$

Sample Variance and Covariance

- Given the i.i.d samples, $X^{(1)}$, $X^{(2)}$, ……, $X^{(n)}$

$$\sigma_{ij} = \frac{1}{n-1}\sum_{k=1}^{n}(X_i^{(k)} - \overline{X_i})(X_j^{(k)} - \overline{X_j})$$

$$\overline{X_i} = \frac{1}{n}\sum_{k=1}^{n} X_i^{(k)}, \overline{X_j} = \frac{1}{n}\sum_{k=1}^{n} X_j^{(k)},$$

$$\hat{\Sigma} = (\sigma_{ij})_{p\times p}$$

Principle Components

- The first principle component is the projections that catches the maximum variance, we can obtain a optimization problem

$$\max_{\alpha} \alpha^T \hat{\Sigma} \alpha$$
$$s.t. \alpha^T \alpha = 1$$

- The solution is the eigenvector corresponding to the largest eigenvalue

$$\hat{\Sigma}\alpha = \lambda\alpha$$

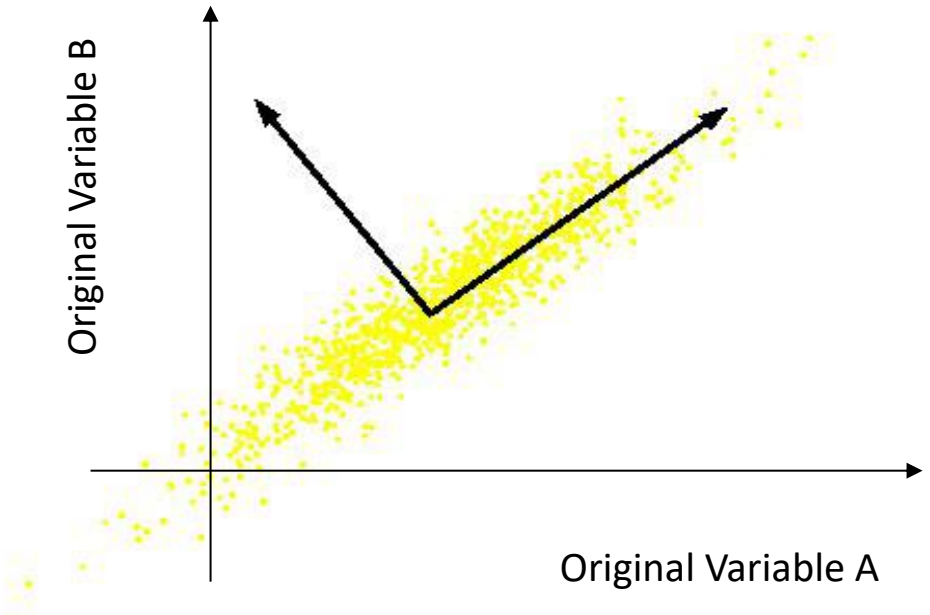
Principle Components

- Similarly, we can get the second, third principle components with extra orthogonal constraints,

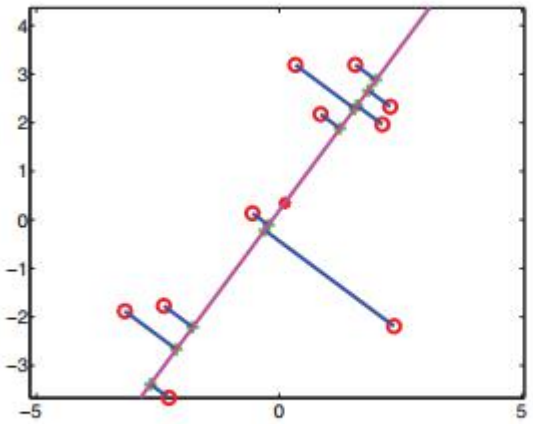
$$\hat{\Sigma}\alpha_1 = \lambda_1\alpha_1$$
$$\hat{\Sigma}\alpha_2 = \lambda_2\alpha_2$$
$$\cdots\cdots\cdots\cdots$$
$$\hat{\Sigma}\alpha_p = \lambda_p\alpha_p$$

$$\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_p$$

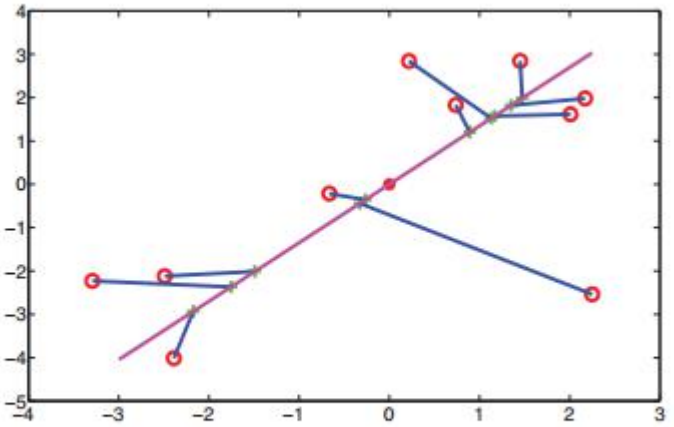
Principal Component



Principal Component



(a)



(b)

Matrix Diagonalization

- For a covariance matrix $\hat{\Sigma}$, there exist a orthogonal matrix Q, such that

$$Q^T \hat{\Sigma} Q = \begin{pmatrix} \lambda_1 & 0 & \cdots & 0 \\ 0 & \lambda_2 & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & \cdots & \lambda_p \end{pmatrix}$$

where $\lambda_1 \geq \lambda_2 \geq \cdots \lambda_p \geq 0$

Using Eigenvectors for Diagonalization

- Let $\alpha_i, i = 1, 2, \cdots, p$ be a set of orthogonal unit eigenvectors with eigenvalue λ_i

$$\hat{\Sigma}\alpha_1 = \lambda_1\alpha_1$$

$$\hat{\Sigma}\alpha_2 = \lambda_2\alpha_2$$

$$\cdots\cdots\cdots\cdots$$

$$\hat{\Sigma}\alpha_p = \lambda_p\alpha_p$$

$$
\begin{pmatrix} \alpha_1^T \\ \alpha_2^T \\ \cdots \\ \alpha_p^T \end{pmatrix} \hat{\Sigma} \left(\alpha_1, \alpha_2, \cdots, \alpha_p\right)
$$

$$
= \begin{pmatrix} \alpha_1^T \\ \alpha_2^T \\ \cdots \\ \alpha_p^T \end{pmatrix} \left(\lambda_1\alpha_1, \lambda_2\alpha_2, \cdots, \lambda_p\alpha_p\right)
$$

$$
= \begin{pmatrix} \lambda_1\alpha_1^T\alpha_1 & \lambda_2\alpha_1^T\alpha_2 & \cdots & \lambda_p\alpha_1^T\alpha_p \\ \lambda_1\alpha_2^T\alpha_1 & \lambda_2\alpha_2^T\alpha_2 & \cdots & \lambda_p\alpha_2^T\alpha_p \\ \cdots & \cdots & \cdots & \cdots \\ \lambda_1\alpha_p^T\alpha_1 & \lambda_2\alpha_p^T\alpha_2 & \cdots & \lambda_p\alpha_p^T\alpha_p \end{pmatrix}
$$

$$
= \mathrm{diag}\{\lambda_1, \lambda_2, \cdots, \lambda_p\}
$$

Principle Components

- Given p orthogonal unit eigenvectors α_i

$$\hat{\Sigma}\alpha_i = \lambda_i \alpha_i, i = 1, 2, \cdots, p$$

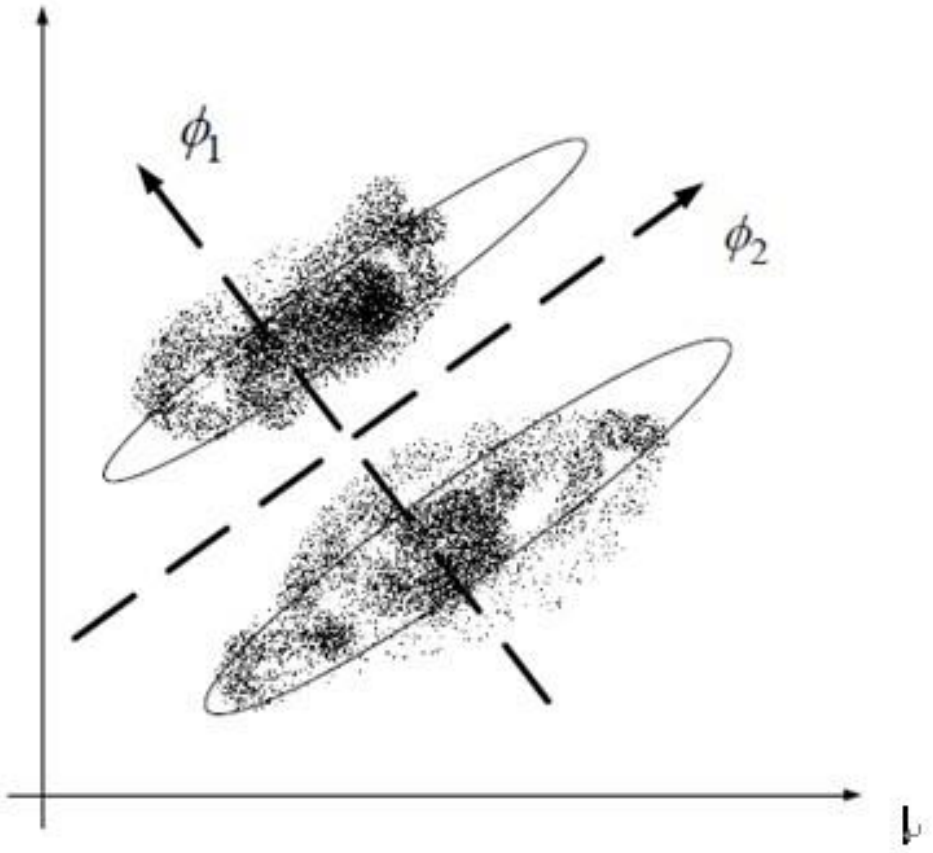
- Let $Z_i = \alpha_i^T X$, i.e $Z = Q^T X$, where

$$Z = \begin{pmatrix} Z_1 \\ Z_2 \\ \cdots \\ Z_p \end{pmatrix} \qquad Q = (\alpha_1, \alpha_2, \cdots, \alpha_p)_{p \times p}$$

PCA in R

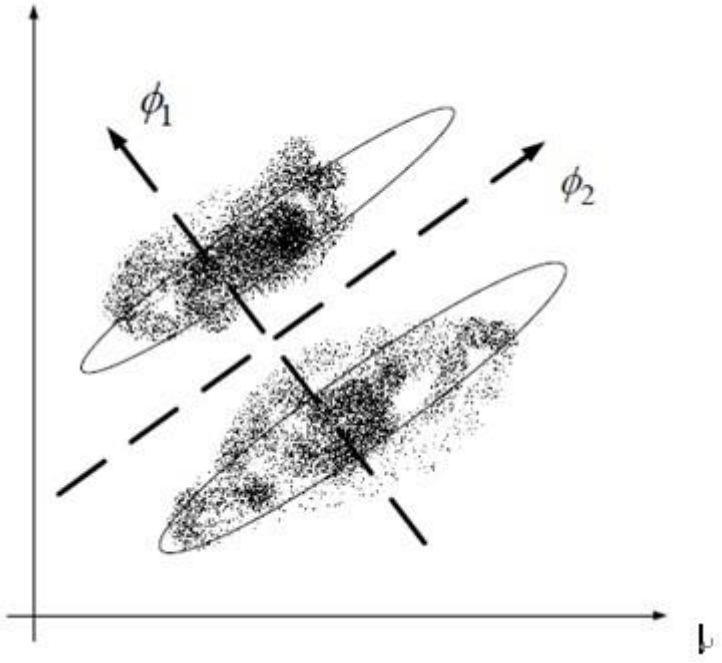
- prcomp(stats)
  - Principal Components Analysis (preferred)
- princomp(stats)
  - Principal Components Analysis
- screeplot(stats)
  - Screeplot of PCA Results
- summary(obj)
- loadings(obj)

PCA problems



PCA追求的是在降维之后能够最大化保持数据的内在信息，并通过衡量在投影方向上的数据方差的大小来衡量该方向的重要性。但是这样投影以后对数据的区分作用并不大，反而可能使得数据点揉杂在一起无法区分。

PCA & LDA

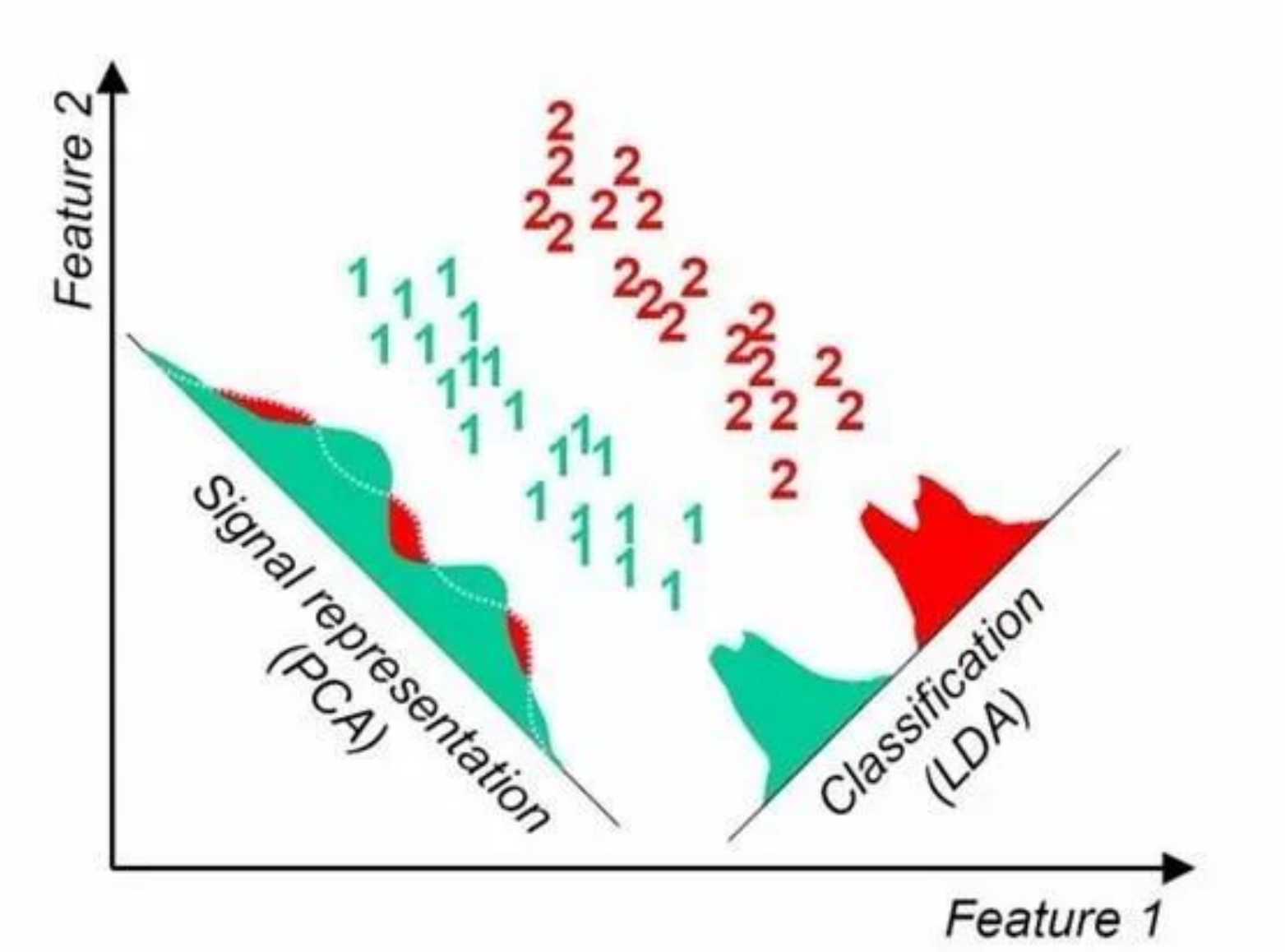


LDA：

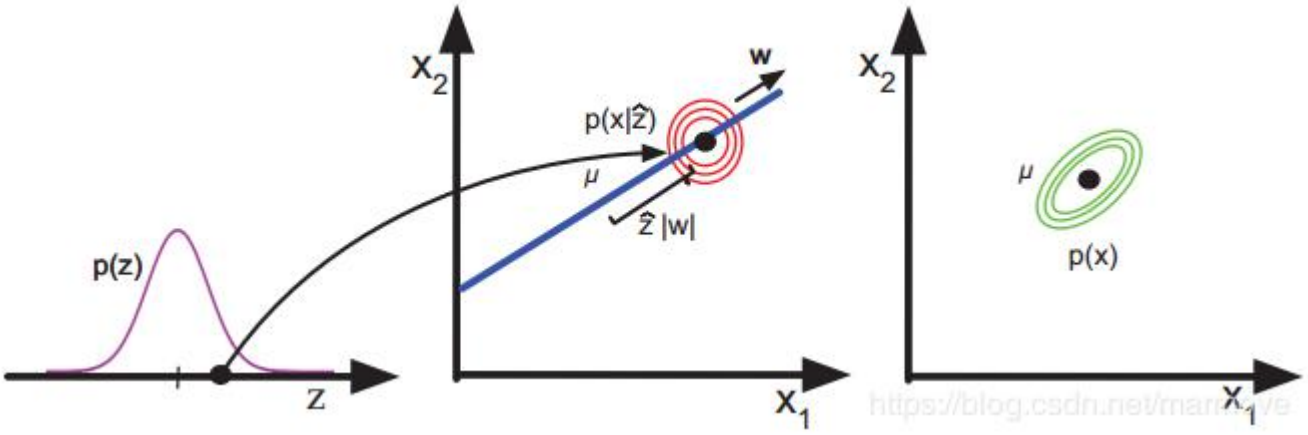
1、同类的数据点尽可能的接近（within class）

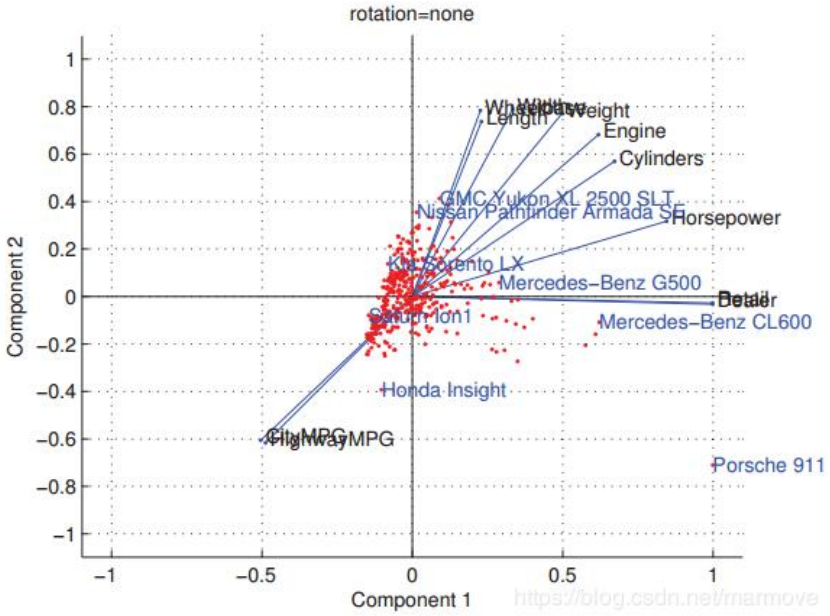
2、不同类的数据点尽可能的分开（between class）

PCA & LDA


Feature 2
Feature 1
Signal representation
(PCA)
Classification
(LDA)

factor analysis (要素分析)





PCoA

CCA

factor analysis (要素分析)

CCA：canonical correlation analysis
典型相关分析

$$\rho(X,Y)=\frac{cov(X,Y)}{\sqrt{D(X)}\sqrt{D(Y)}}$$

- 例子：我们拿到两组数据，第一组是人身高和体重的数据，第二组是对应的跑步能力和跳远能力的数据。那么我们能不能说这两组数据是相关的呢？
- 如果X是包括人身高和体重两个维度的数据，而Y是包括跑步能力和跳远能力两个维度的数据，就不能直接使用相关系数的方法。那我们能不能变通一下呢？
- CCA使用的方法是将多维的X和Y都用线性变换为1维的X’和Y’，然后再使用相关系数来看X’和Y’的相关性。
- 对于我们的CCA，它选择的投影标准是降维到1维后，两组数据的相关系数最大。

factor analysis (要素分析)

CCA： canonical correlation analysis
典型相关分析

CCA的SVD解法：

输入：各为m个的样本X和Y，X和Y的维度都大于1
输出：X,Y的相关系数ρ,X和Y的线性系数向量a和b

1. 计算X的方差S_{XX}，Y的方差S_{YY}，X和Y的协方差S_{XY}，Y和X的协方差$S_{YX} = S_{XY}^T$
2. 计算矩阵$M = S_{XX}^{-1/2} S_{XY} S_{YY}^{-1/2}$
3. 对矩阵M进行奇异值分解，得到最大的奇异值ρ，和最大奇异值对应的左右奇异向量u,v
4. 算X和Y的线性系数向量a和b,$a = S_{XX}^{-1/2} u, b = S_{YY}^{-1/2} v$

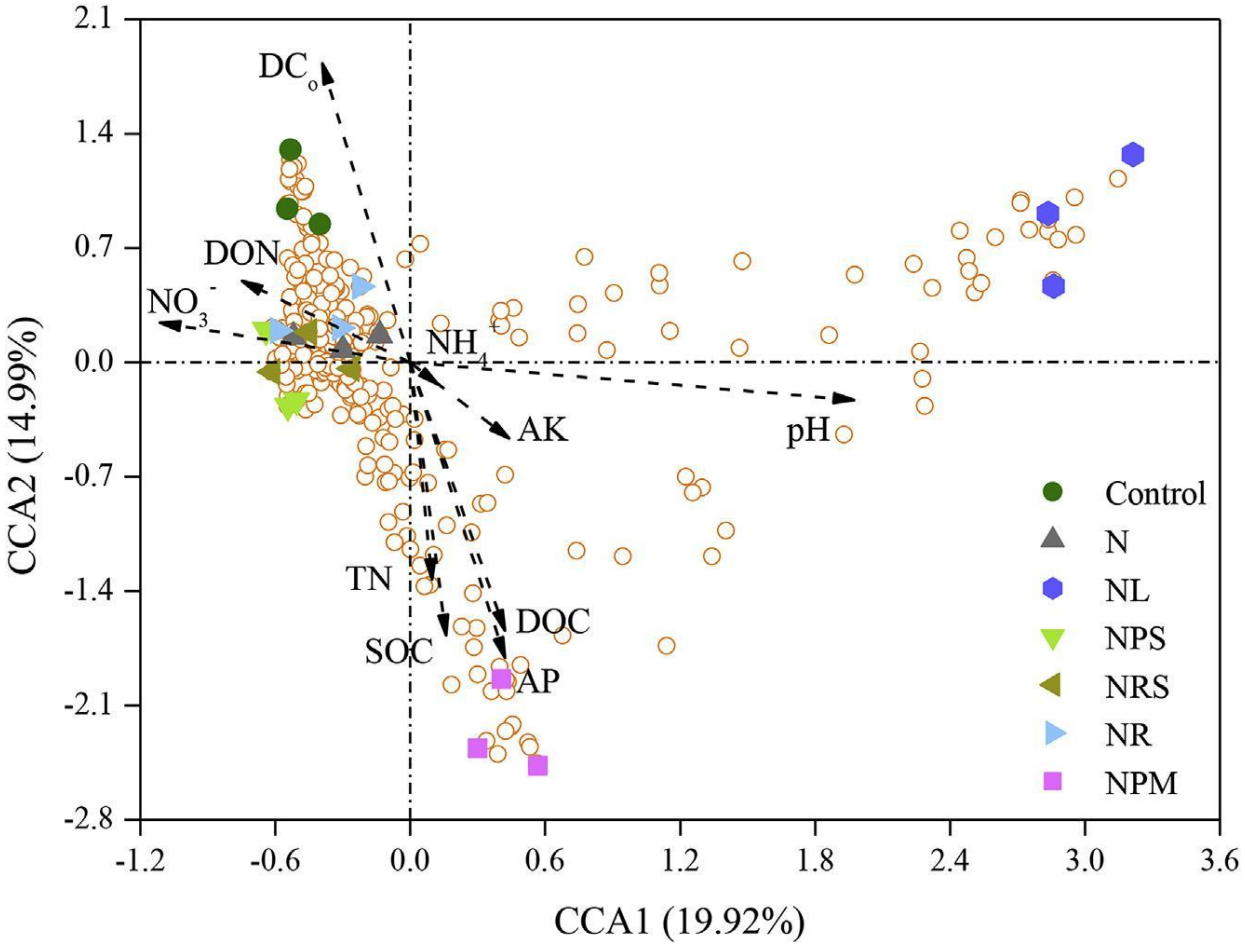
factor analysis (要素分析)

CCA：canonical correlation analysis
典型相关分析

- 我们在算法里只找了相关度最大的奇异值或者特征值
- 作为数据的相关系数，实际上我们也可以像PCA一样找出第二大奇异值，第三大奇异值，。。。得到第二相关系数和第三相关系数

factor analysis (要素分析)

CCA：canonical correlation analysis
典型相关分析



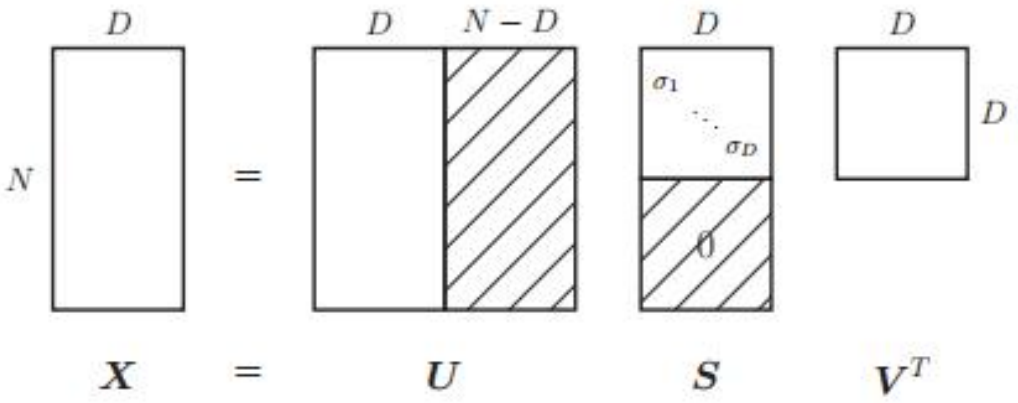
PCA, LDA & CCA

- **主成分分析PCA：**降维的原则是投影方差最大
- **线性判别分析LDA：**降维的原则是同类的投影方差小，异类间的投影方差大
- **典型相关分析CCA：**降维的原则是降维到1维后，两组数据的相关系数最大

$$\underbrace{arg\ max}_{a,b} \frac{cov(X', Y')}{\sqrt{D(X')}\sqrt{D(Y')}}$$

奇异值分解(SVD)及其应用

- 奇异值分解(SVD, Singular Value Decomposition)
- Application I: Information retrieval
- Application II: Gene expression data analysis

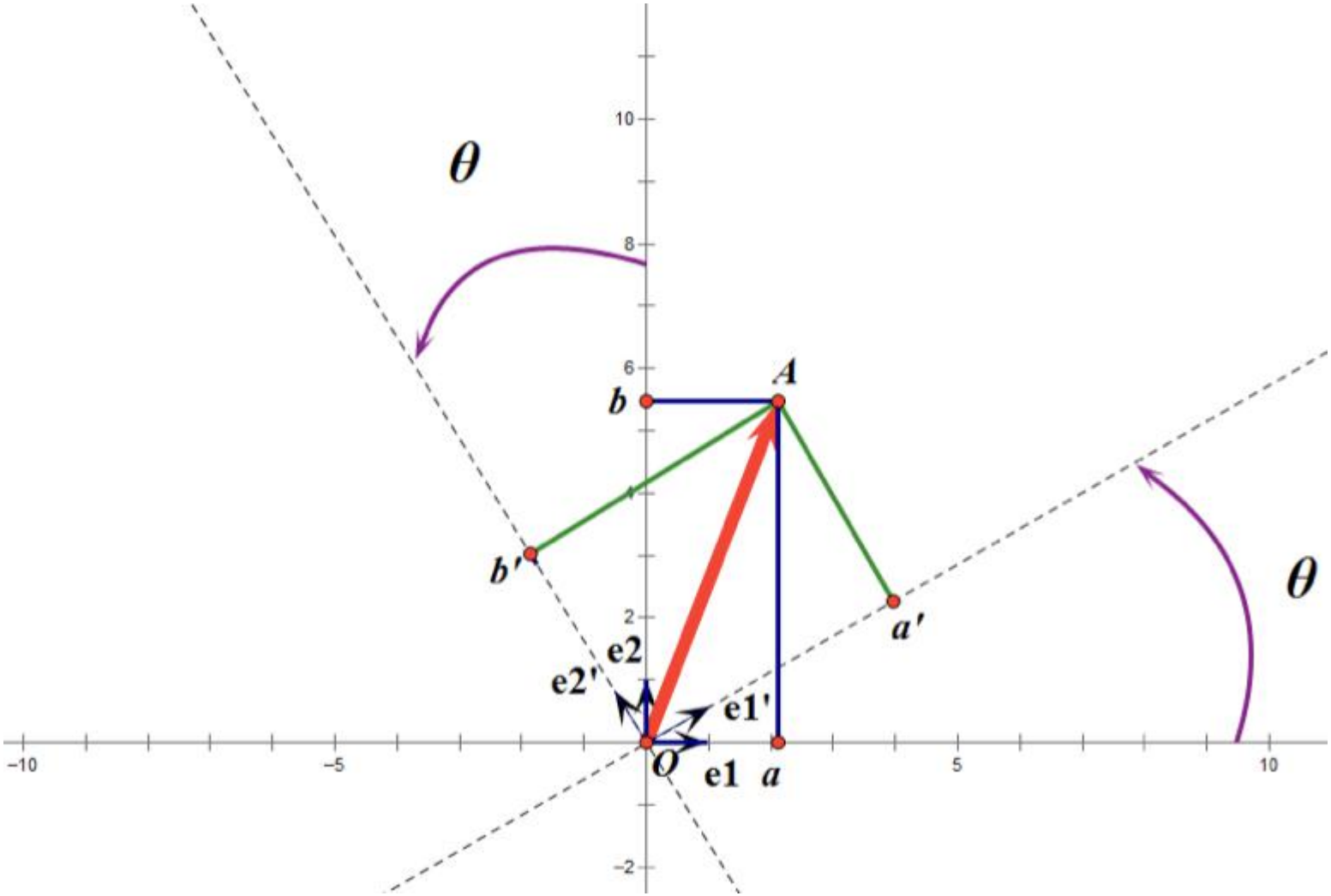


奇异值分解(SVD)

- 定理1：设A是秩为r的实值矩阵，则存在m阶正交矩阵U，n阶正交矩阵V，使得 $A=UDV^T$. 其中D为对角矩阵 $D_{m\times n}=(\mathrm{diag}\{\sigma_1,\cdots,\sigma_r,0,\cdots,0\},0)$，满足 $\sigma_1\geq\sigma_2\geq\cdots\geq\sigma_r>0$, 则

$$\mathrm{rank}(A)=r, N(A)=\mathrm{Span}\{v_{r+1},\cdots,v_n\},$$
$$R(A)=\mathrm{Span}\{u_1,\cdots,u_r\}.$$
$$A=\sum_{i=1}^{r}\sigma_i u_i v_i^T$$
$$||A||_F^2=\sigma_1^2+\cdots+\sigma_r^2$$

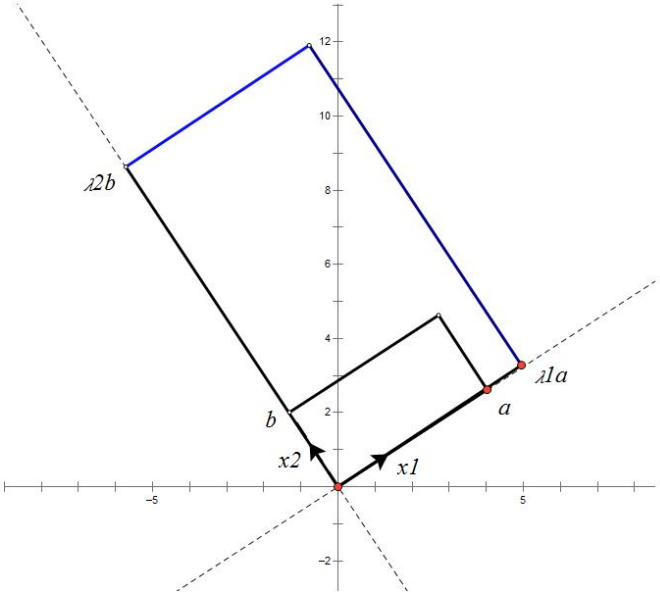
奇异值分解(SVD)



奇异值分解(SVD)

- 旋转是相对的：我们可以说向量的空间位置没有变，标准参考系向左旋转了θ角度，而如果我选择了e1’、e2’作为新的标准坐标系，那么在新坐标系中OA（原标准坐标系的表示）就变成了OA’，这样看来就好像坐标系不动。

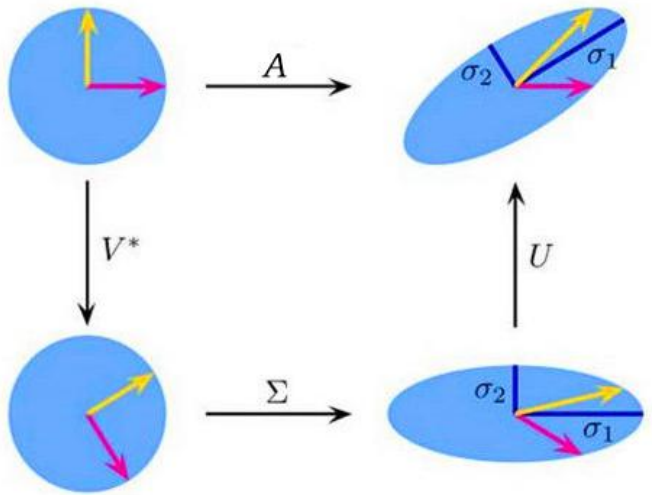
- 正交矩阵的行（列）向量都是两两正交的单位向量，正交矩阵对应的变换为正交变换，它有两种表现：旋转和反射。正交矩阵将标准正交基映射为标准正交基（从e1、e2到e1’、e2’）。



奇异值分解(SVD)

- 称 σ_i 为矩阵A的奇异值。

- 称U,V的列向量分别称为A的左、右奇异向量。



$$A = U \cdot \Sigma \cdot V^*$$

奇异值分解(SVD)

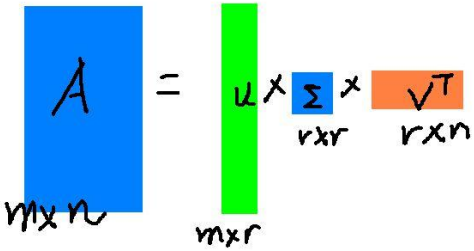
- $A^TA_{n,n}$和$AA^T_{m,m}$都是实对称矩阵，则存在n阶正交矩阵V和m阶正交矩阵U，使得

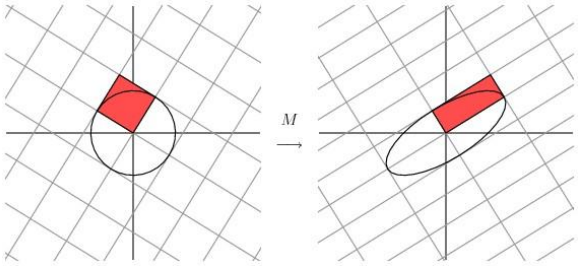
$$U(AA^T)U^T = D_1$$
$$V(A^TA)V^T = D_2$$

- (A^TA)和(AA^T)的特征值相同

$$\text{If}(A^TA)\eta = \lambda\eta, \text{then } (AA^T)A\eta = \lambda A\eta$$

- A^TA的非0 特征值的个数等于A的秩 r .





Rank, Range, and Null space

- The rank of matrix A can be calculated from SVD by the number of nonzero singular values.
- The range of matrix A is the left singular vectors of U corresponding to the non-zero singular values.
- The null space of matrix A is the right singular vectors of V corresponding to the zeroed singular values.

Rank, Range, and Null space

$$A_{m\times n} = U_{m\times m} W_{n\times n} V^T_{n\times n}$$

Range

Rank

Null Space

$$\begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & -1 \\ 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 4 & 0 & 0 & 0 & 0 \\ 0 & 3 & 0 & 0 & 0 \\ 0 & 0 & \sqrt{5} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ \sqrt{0.2} & 0 & 0 & 0 & \sqrt{0.8} \\ 0 & 0 & 0 & 1 & 0 \\ -\sqrt{0.8} & 0 & 0 & 0 & \sqrt{0.2} \end{bmatrix}^T$$

$$A_{4\times 5} = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix}_{4\times 3} \begin{bmatrix} 4 & 0 & 0 \\ 0 & 3 & 0 \\ 0 & 0 & \sqrt{5} \end{bmatrix}_{3\times 3} \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ \sqrt{0.2} & 0 & 0 \\ 0 & 0 & 0 \\ -\sqrt{0.8} & 0 & 0 \end{bmatrix}^T_{5\times 3}$$

Rank(A)=3

Condition Number

- SVD can tell How close a square matrix A is to be singular.
- The ratio of the largest singular value to the smallest singular value can tell us how close a matrix is to be singular:

$$A = U \begin{bmatrix} \sigma_1 & 0 & \cdots & 0 \\ 0 & \sigma_2 & \cdots & 0 \\ \vdots & \vdots & \cdots & \vdots \\ 0 & 0 & 0 & \sigma_k \end{bmatrix} V^T, \quad c = \frac{\sigma_1}{\sigma_k}$$

- A is singular if c is infinite.
- A is ill-conditioned if c is too large (machine dependent).

奇异值分解(SVD)

- 定理2 (Eckart and Young)： Let the SVD of A be given by $A=UDV^T$ with r = rank(A) < p = min(m, n) and define

$$A_k = \sum_{i=1}^{k} u_i \sigma_i v_i^T$$

Then

$$\min_{rank(B)=k} ||A - B||_F^2 = ||A - A_k||_F^2 = \sigma_{k+1}^2 + \cdots + \sigma_p^2$$

其中 $||\cdot||_F$为Frobenius 范数, 定义为

$$||A||_F^2 = \sum_{i=1}^{m}\sum_{j=1}^{n} |a_{ij}|^2 = \mathrm{tr}(A^*A) = \sum_{i=1}^{min(m,n)} \sigma_i^2$$

奇異値分解(SVD)

- In other words, A_k, which is constructed from the k-largest singular triplets of A, is the closest rank-k matrix to A. In fact, A_k is the best approximation to A for any unitarily invariant norm[*].

* L. MIRSKY, Symmetric gage functions and unitarilly invariant norms, Q. J. Math, 11(1960), pp. 50-59.

矩阵近似

- 前k个奇异值的贡献率

$$\rho = \frac{\sum_{j=1}^{k} \sigma_j^2}{\sum_{j=1}^{r} \sigma_j^2}$$

- 可以根据贡献率截取前k个奇异值近似矩阵

$$A \simeq \sum_{j=1}^{k} \sigma_j u_j v_j^T$$

SVD举例

$$A = \begin{pmatrix} 1 & 2 & 1 & 1 & 1 \\ 1 & 2 & 1 & 1 & 1 \\ 2 & 3 & 2 & 1 & 0 \end{pmatrix}$$

$$r(A) = 2, \sigma_1 = 5.70863, \sigma_2 = 1.188084, \sigma_3 = 0$$

$$U = (u_1, u_2, u_3)$$

$$U = \begin{pmatrix} 0.4836967 & 0.5157882 & 0.7071068 \\ 0.7294347 & -0.6840504 & 0 \\ 0.4836967 & 0.5157882 & -0.7071068 \end{pmatrix}$$

SVD举例

$$V = (v_1, v_2, v_3, v_4, v_5)$$

$$V = \begin{pmatrix} -0.4250167 & -0.28324973 & 0.00000000 & 0.85972695 & 0.000000000 \\ -0.7222558 & 0.00926002 & 0.50863102 & -0.35400522 & 0.306978282 \\ -0.4250167 & -0.28324973 & -0.80435350 & -0.30343304 & -0.008618095 \\ -0.2972391 & 0.29250975 & 0.08281395 & -0.05057217 & -0.903698656 \\ -0.1694616 & 0.86826924 & -0.29572248 & 0.20228869 & 0.298360187 \end{pmatrix}$$

$$A = \sigma_1 u_1 v_1^T + \sigma_2 u_2 v_2^T, \quad c = \frac{\sigma_1^2}{\sigma_1^2 + \sigma_2^2} = 0.958484$$

$$A \simeq \sigma_1 u_1 v_1^T$$

$$= \begin{pmatrix} 1.173575 & 1.994325 & 1.173575 & 0.8207501 & 0.4679250 \\ 1.769800 & 3.007526 & 1.769800 & 1.2377253 & 0.7056503 \\ 1.173575 & 1.994325 & 1.173575 & 0.8207501 & 0.4679250 \end{pmatrix}.$$

SVD in R

- svd()
- la.svd()

- A good demo

http://www.ats.ucla.edu/stat/r/pages/svd_demos.htm

SVD vs PCA

- If X is centering each column, then X^TX is proportional to the covariance matrix of variables g_i. So, the right singular vectors $\{v_k\}$ are the same as the principal components.
- If instead each row of X is centered, XX^T is proportional to the covariance matrix of the variables a_j. In this case, the left singular vectors $\{u_k\}$ are the same as the principal components of $\{a_j\}$.

Data Fitting Problem

$$y = ax^2 + bx + c$$

$$\underbrace{\begin{bmatrix} x_1^2 & x_1 & 1 \\ x_2^2 & x_2 & 1 \\ \vdots & \vdots & \vdots \\ x_N^2 & x_N & 1 \end{bmatrix}}_{S} \underbrace{\begin{bmatrix} a \\ b \\ c \end{bmatrix}}_{\mathbf{a}} = \underbrace{\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_M \end{bmatrix}}_{\mathbf{y}}$$

$$\mathbf{a} = V \cdot [\mathrm{diag}(1/\sigma_i)] \cdot (U^{\mathsf{T}}\ \mathbf{y})$$

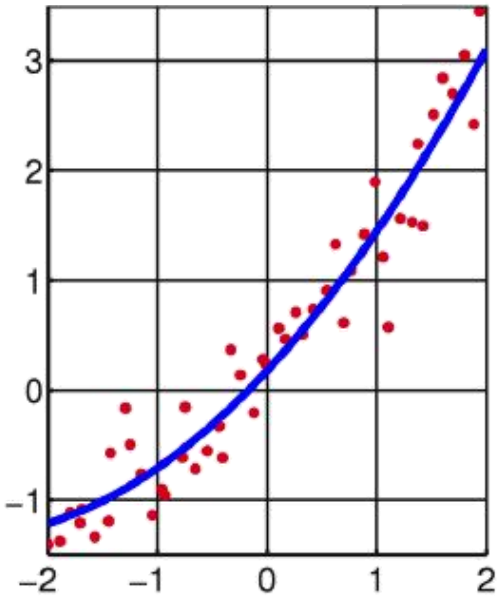
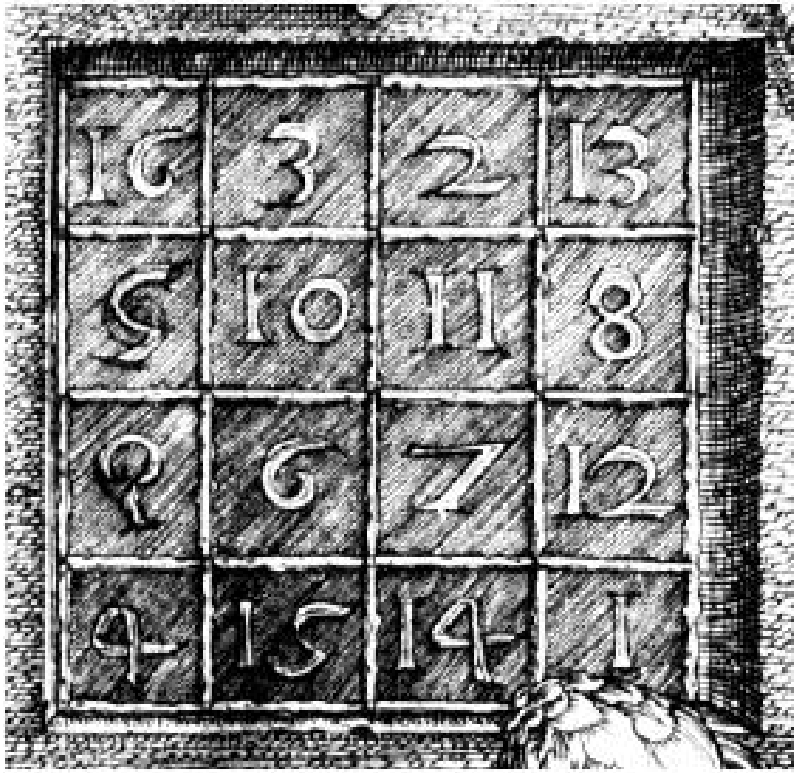
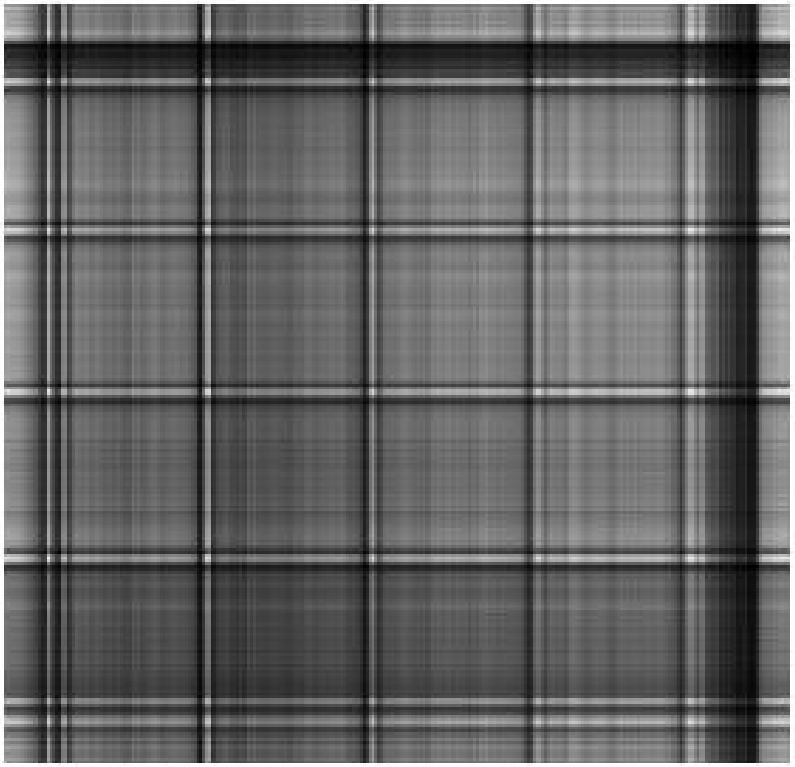


Image Processing

[U,W,V]=svd(A)

NewImg=U(:,1)*W(1,1)*V(:,1)’

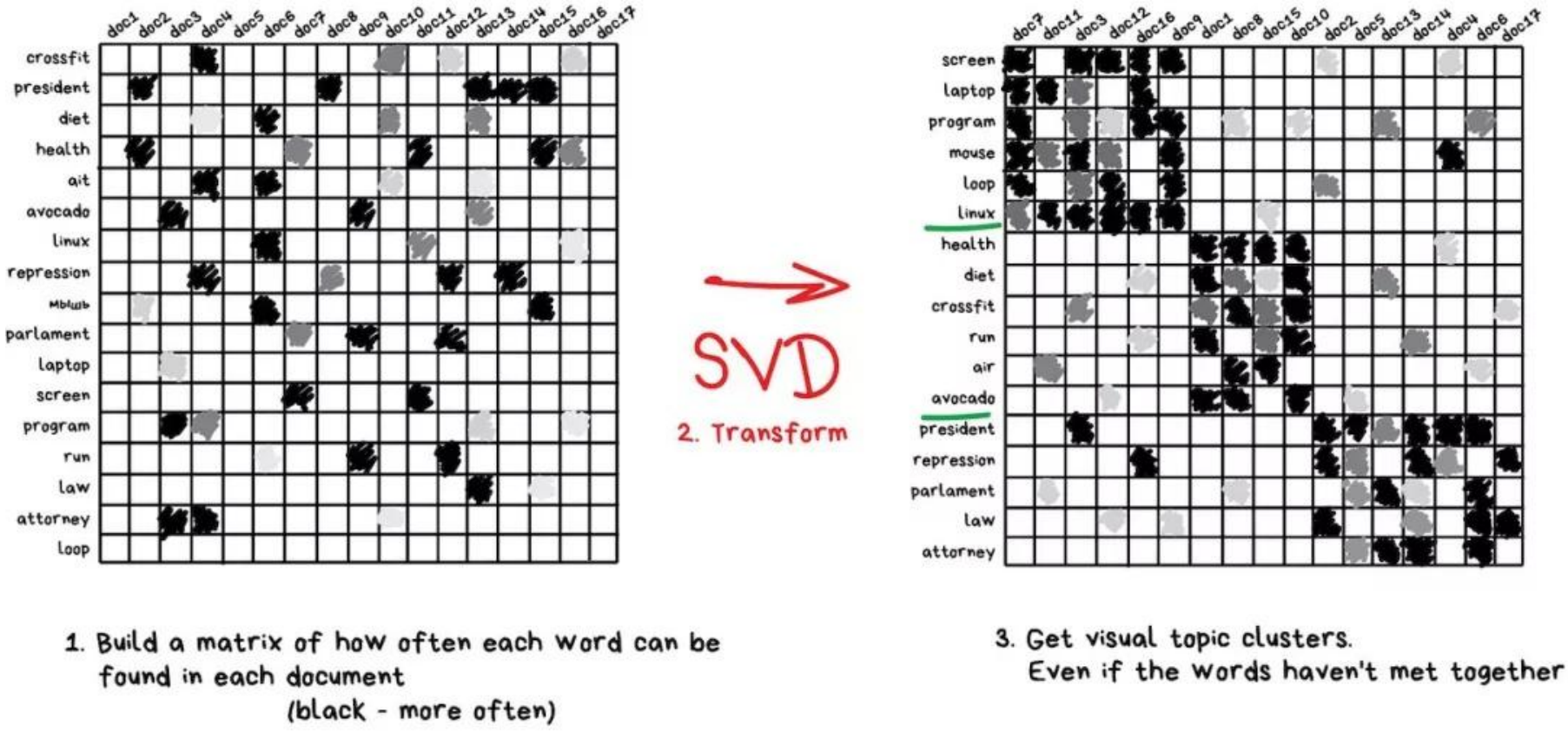




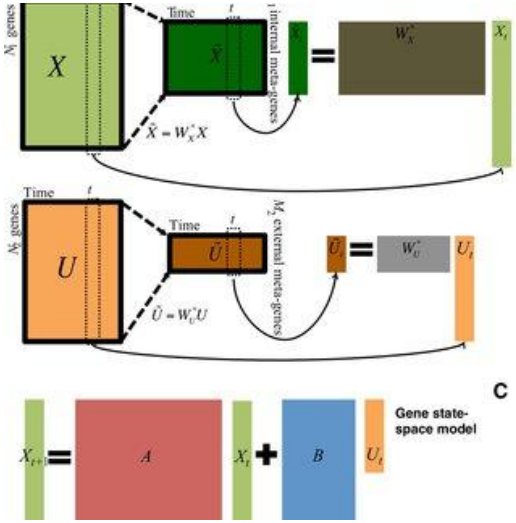
Digital Signal Processing (DSP)

- SVD is used as a method for noise reduction.
- Let a matrix A represent the noisy signal:
  - compute the SVD,
  - and then discard small singular values of A.
- It can be shown that the small singular values mainly represent the noise, and thus the rank-k matrix A_k represents a filtered signal with less noise.

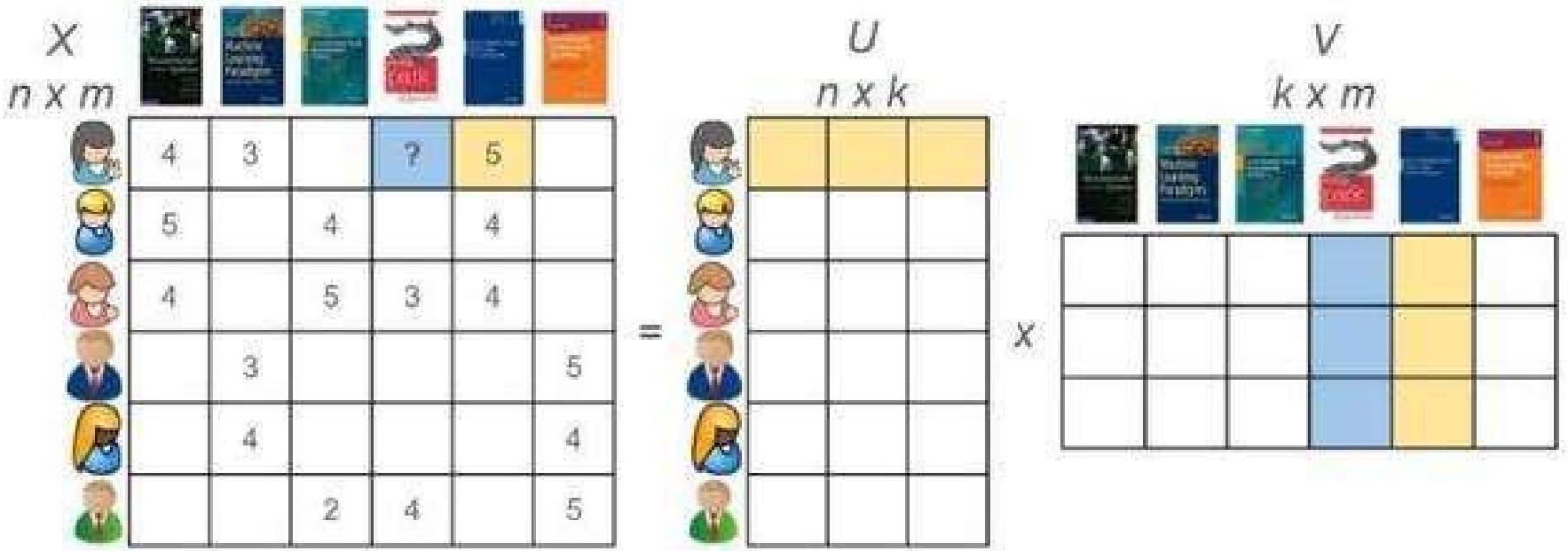
奇异值分解(SVD)


SEPARATE DOCUMENTS BY TOPIC
doc1 doc2 doc3 doc4 doc5 doc6 doc7 doc8 doc9 doc10 doc11 doc12 doc13 doc14 doc15 doc16 doc17
crossfit
president
diet
health
air
avocado
linux
repression
мышь
parlament
laptop
screen
program
run
law
attorney
loop
SVD
2. Transform
doc7 doc11 doc3 doc12 doc16 doc9 doc1 doc8 doc15 doc10 doc2 doc5 doc13 doc14 doc4 doc6 doc17
screen
laptop
program
mouse
loop
linux
health
diet
crossfit
run
air
avocado
president
repression
parlament
law
attorney
1. Build a matrix of how often each word can be found in each document
(black - more often)
3. Get visual topic clusters.
Even if the words haven't met together
LATENT SEMANTIC ANALYSIS (LSA)

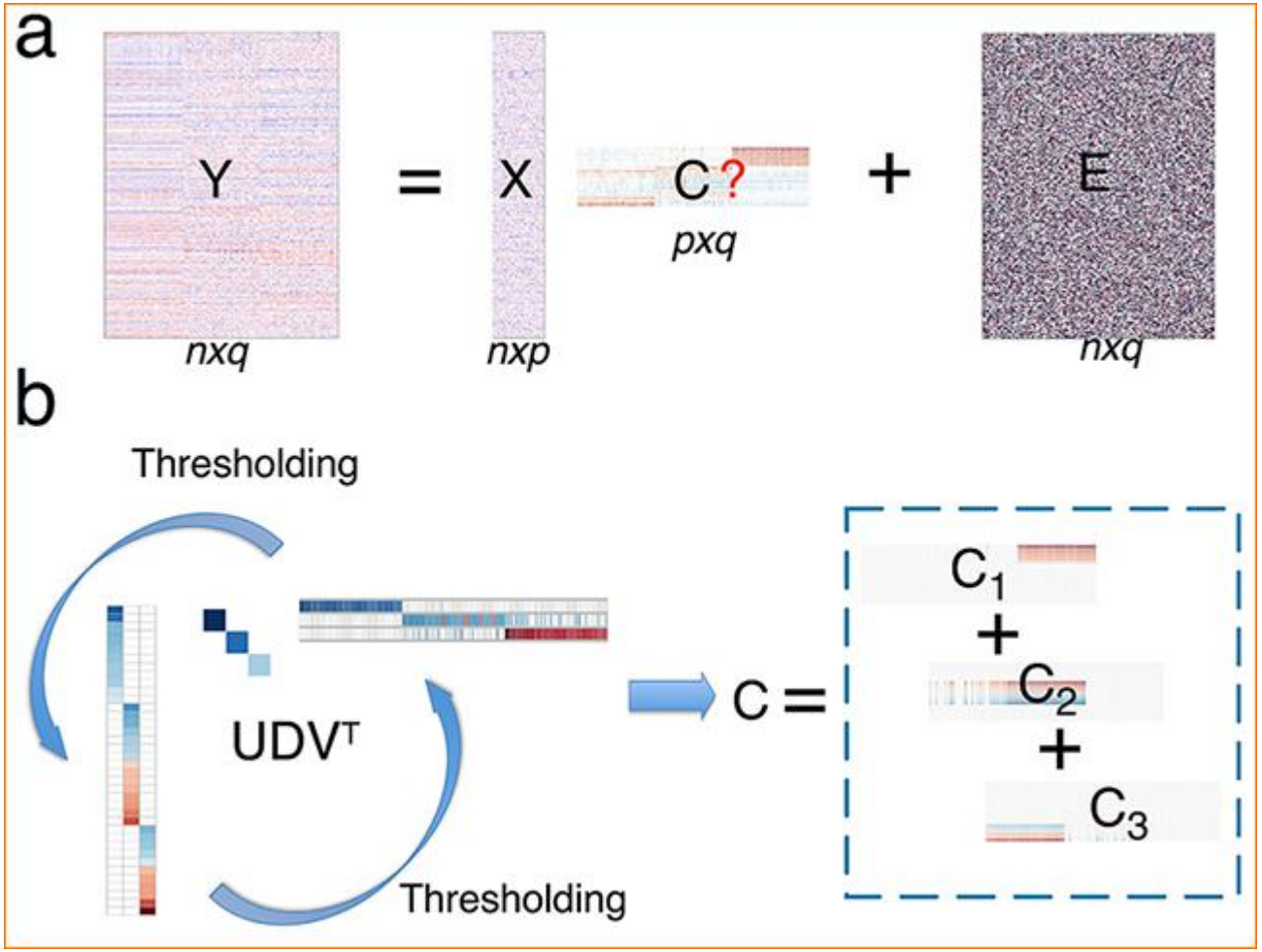
奇异值分解(SVD)

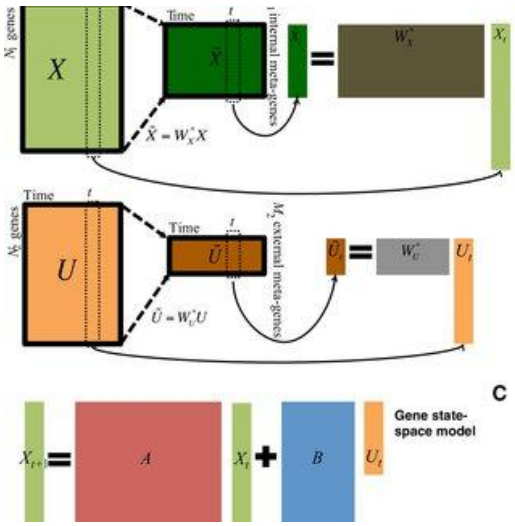


奇异值分解(SVD)

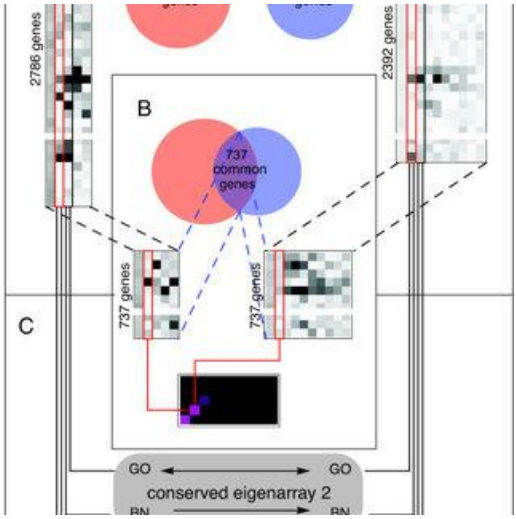


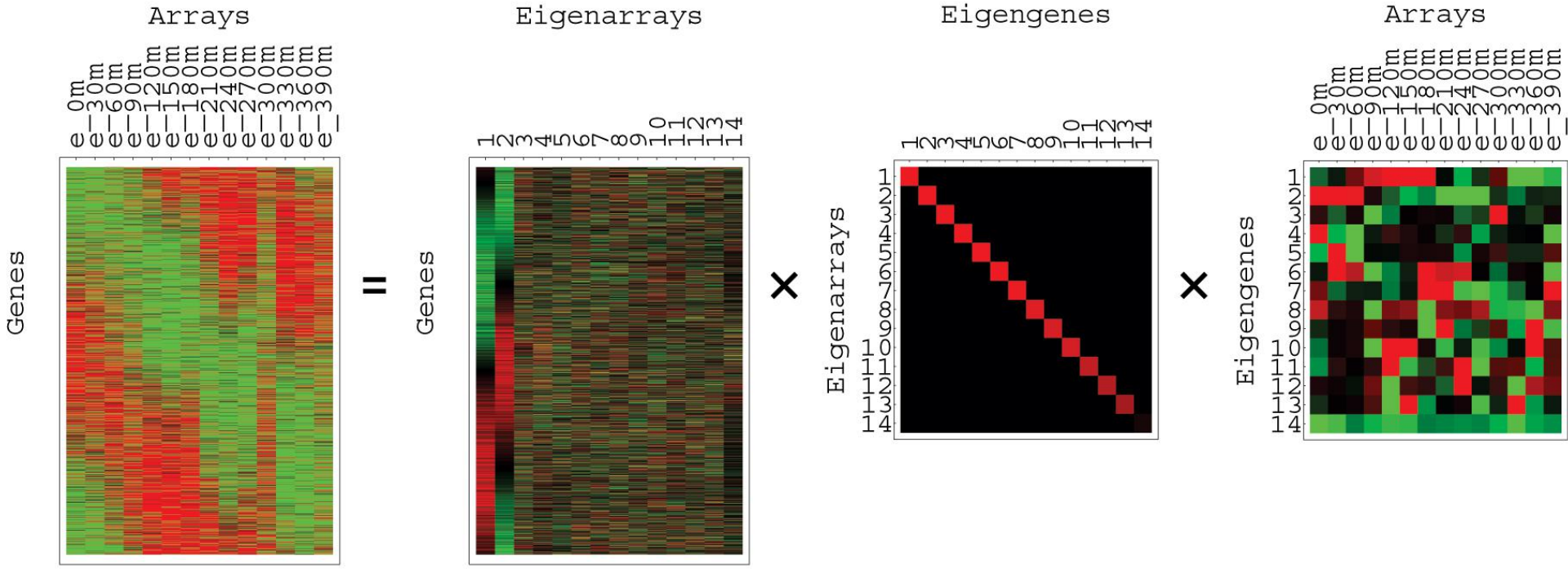
奇异值分解(SVD)


a
Y
=
X
C?
pxq
+
E
nxq
nxp
nxq
b
Thresholding
UDV^T
Thresholding
C =
C1
+
C2
+
C3


X
Time
internal meta-genes
U
external meta-genes
Gene state-space model
A
B
C

奇异值分解(SVD)


2786 genes
2392 genes
B
737 common genes
737 genes
737 genes
C
GO
GO
conserved eigenarray 2


Arrays
Eigenarrays
Eigengenes
Arrays
Genes
Genes
Eigenarrays
Eigengenes

Applications

- Information retrieval
  - LSI: Latent semantic indexing
  - SVD applied to term document matrix
  - compute best rank k approximation
  - eigenvectors correspond to linguistic concepts
- Gene expression data analysis
  - SVD useful preprocessing step
  - grouping genes by transcriptional response, grouping assays by expression profile

Latent Semantic Indexing (LSI)
by
Singular Value Decomposition

Michael W. Berry, Susan T. Dumais, Gavin W. O'Brien. Using Linear Algebra for Intelligent Information Retrieval. SIAM Review 37(4): 573-595, 1995.

Information Retrieval

- Q : "Light waves."
- D_1: "Particle and wave models of light."
- D_2: "Surfing on the waves under star lights."
- D_3: "Electro-magnetic models for fotons."

	particle	wave	model	light	surfing	star	electro-magnetic	foton	REL	MATCH
D_1	x	x	x	x					R	M
D_2		x		x	x	x				M
D_3			x				x	x	R	

Motivation for LSI

- To find and fit a useful model of the relationships between terms and documents.
- To find out what terms "really" are implied by a query .
- LSI allow the user to search for concepts rather than specific words.
- LSI can retrieve documents related to a user's query even when the query and the documents do not share any common terms.

How LSI Works?

- Uses multidimensional vector space to place all documents and terms.
- Each dimension in that space corresponds to a concept existing in the collection.
- Thus underlying topics of the document is encoded in a vector.
- Common related terms in a document and query will pull document and query vector close to each other.

LSI Method

- Create a rank-*k* approximation to *A*
- k < r_A or k = r_A ,
- $A_k = U_k S_k V^T_k$

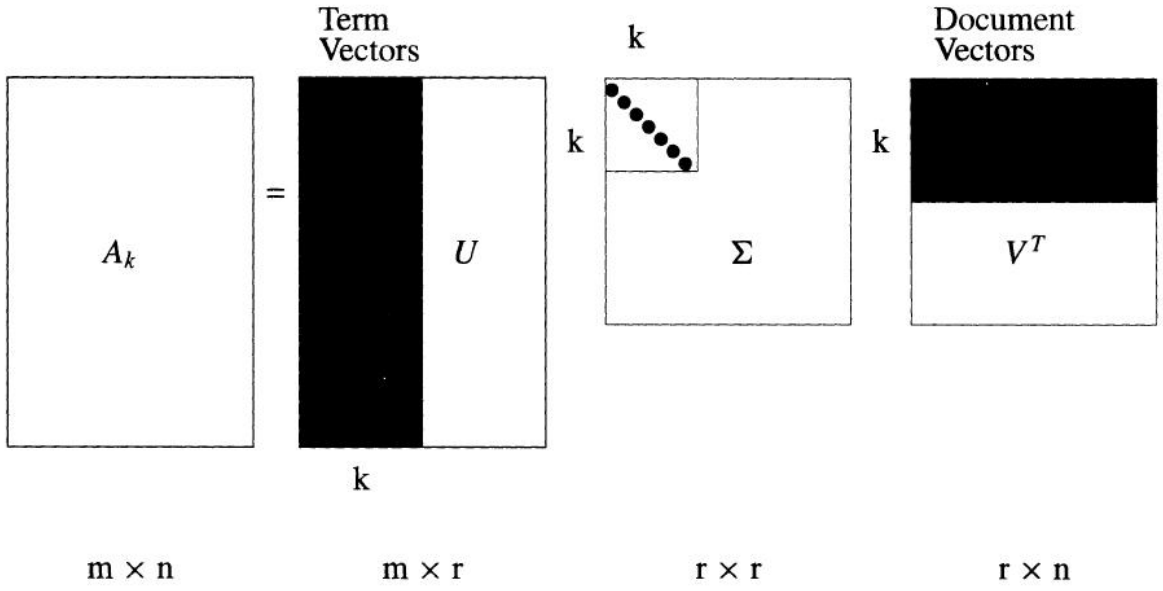


FIG. 1. *Mathematical representation of the matrix* A_k.

LSI Method

- Using truncated SVD, underlying *latent* structure is represented in reduced-k dimensional space.

- *Noise* in word usage is eliminated

LSI-Procedure

- Obtain term-document matrix.
- Compute the SVD.
- Truncate-SVD into reduced-k LSI space.
  - k-dimensional semantic structure
  - Similarity on reduced-space
    - Term-term
    - Term-document
    - Document-document

Query processing

- Map the query to reduced k-space

  $$\boldsymbol{q'} = q^T U_k S^{-1}{}_k$$

- Retrieve documents or terms within a proximity.
  - Cosine
  - Best m

Updating

- Folding-in

$$\boldsymbol{d'} = d^T U_k S^{-1}_k$$

similar to query projection

- SVD re-computation

Folding-In (Documents)

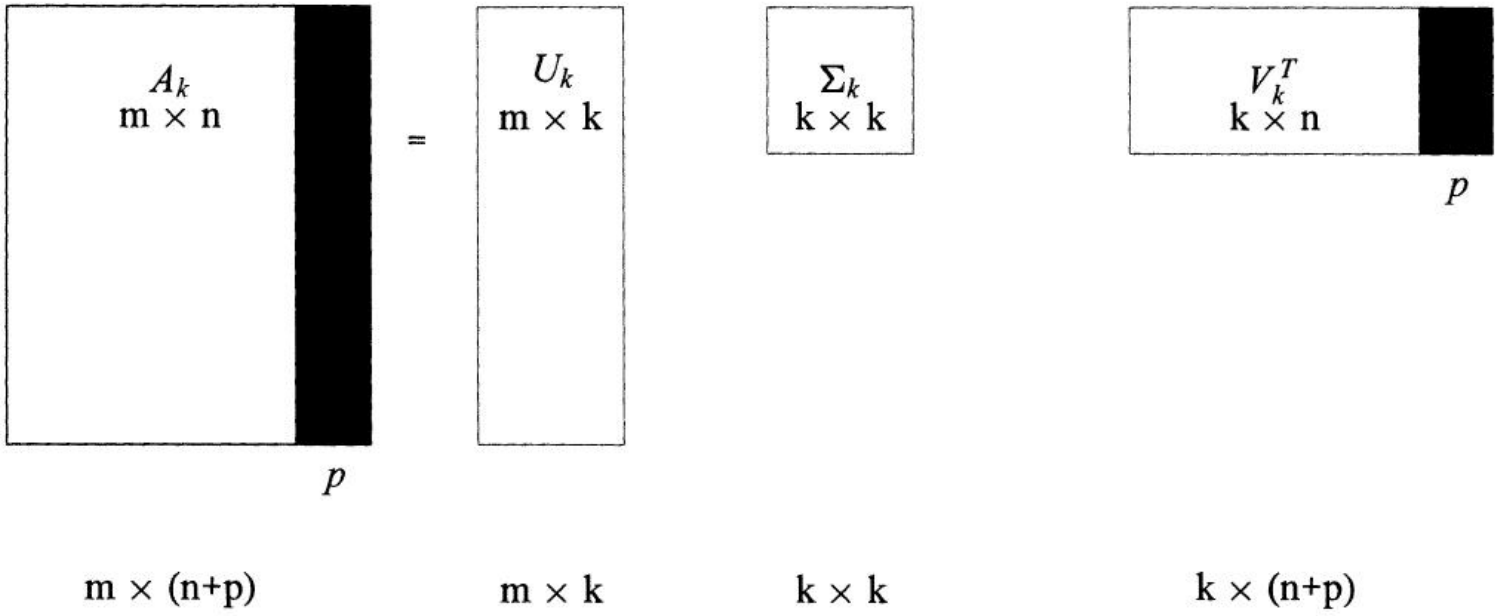


FIG. 2. *Mathematical representation of folding-in p documents.*

Folding-In (Terms)

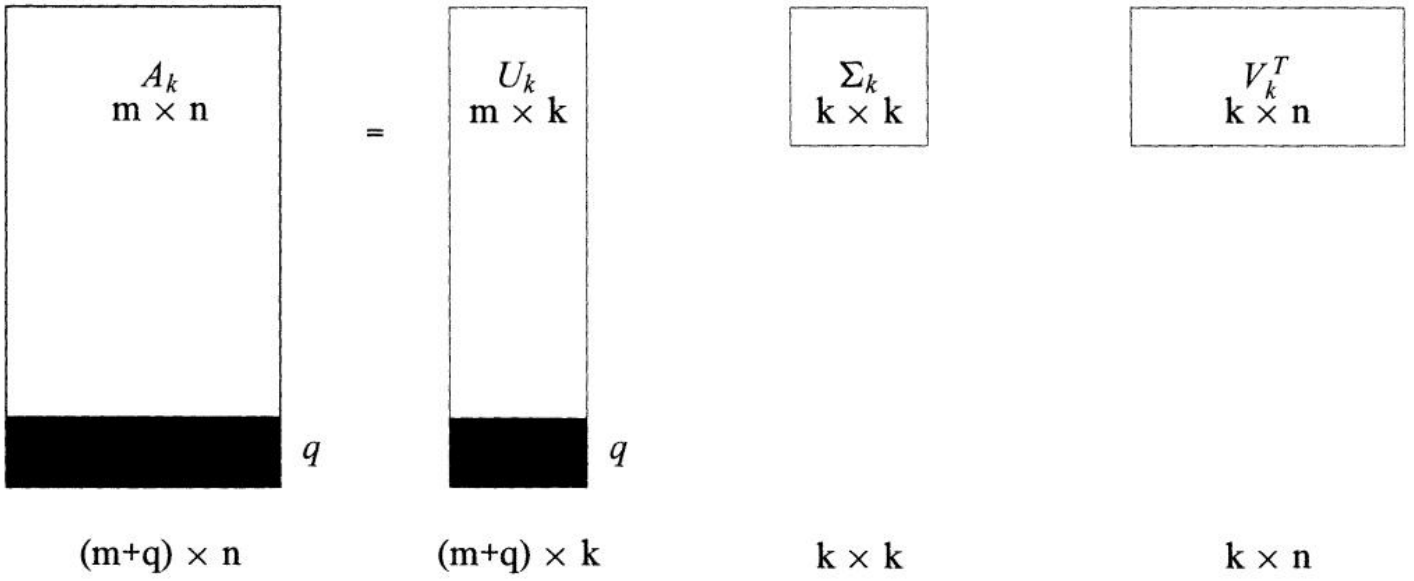


FIG. 3. *Mathematical representation of folding-in q terms.*

Label	Titles
B1	A Course on Integral Equations
B2	Attractors for Semigroups and Evolution Equations
B3	Automatic Differentiation of Algorithms: Theory, Implementation, and Application
B4	Geometrical Aspects of Partial Differential Equations
B5	Ideals, Varieties, and Algorithms - An Introduction to Computational Algebraic Geometry and Commutative Algebra
B6	Introduction to Hamiltonian Dynamical Systems and the N-Body Problem
B7	Knapsack Problems: Algorithms and Computer Implementations
B8	Methods of Solving Singular Systems of Ordinary Differential Equations
B9	Nonlinear Systems
B10	Ordinary Differential Equations
B11	Oscillation Theory for Neutral Differential Equations with Delay
B12	Oscillation Theory of Delay Differential Equations
B13	Pseudodifferential Operators and Nonlinear Partial Differential Equations
B14	Sinc Methods for Quadrature and Differential Equations
B15	Stability of Stochastic Differential Equations with Respect to Semi-Martingales
B16	The Boundary Integral Approach to Static and Dynamic Contact Problems
B17	The Double Mellin-Barnes Type Integrals and Their Applications to Convolution Theory

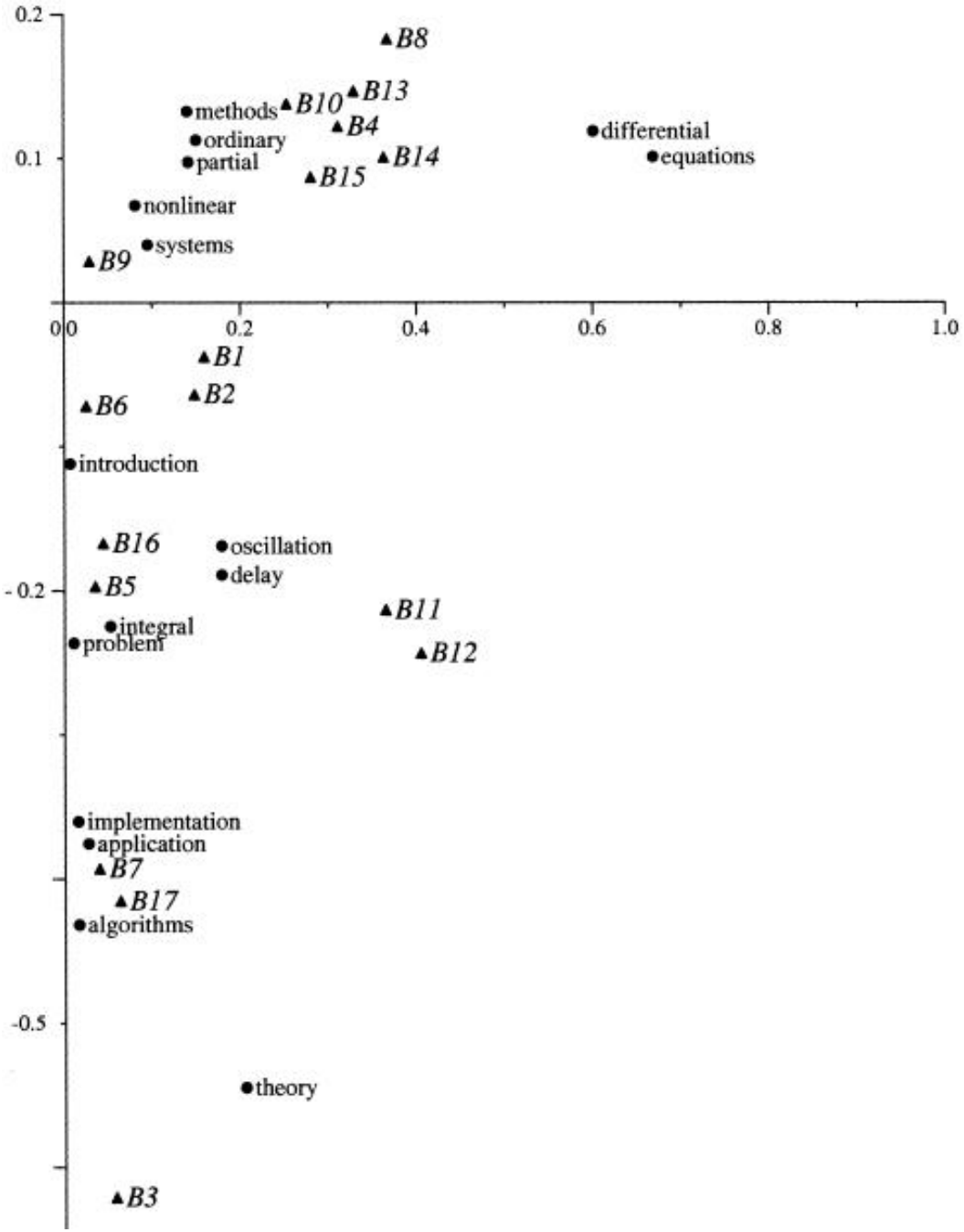
Example:SIAM Review Titles

Example

Terms	Documents																
	B1	B2	B3	B4	B5	B6	B7	B8	B9	B10	B11	B12	B13	B14	B15	B16	B17
algorithms	0	0	1	0	1	0	1	0	0	0	0	0	0	0	0	0	0
application	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1
delay	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0
differential	0	0	0	1	0	0	0	1	0	1	1	1	1	1	1	0	0
equations	1	1	0	1	0	0	0	1	0	1	1	1	1	1	1	0	0
implementation	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0
integral	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1
introduction	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0
methods	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0
nonlinear	0	0	0	0	0	0	0	0	1	0	0	0	1	0	0	0	0
ordinary	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0
oscillation	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0
partial	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0
problem	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	1	0
systems	0	0	0	0	0	1	0	1	1	0	0	0	0	0	0	0	0
theory	0	0	1	0	0	0	0	0	0	0	1	1	0	0	0	0	1

Mapping Method

- Compute the truncated SVD (k=2)
- Terms plane
  - x-coordinates : The first column of U_2 multiplied by the first singular value, for the
  - y-corrdinates: second column of U_2 multiplied by the second singular value.
- Documents plane:
  - x-coordinates : The first column of V_2 scaled by the first singular value, for the
  - y-corrdinates: second column of V_2 scaled by the second singular value.



Mapping Results

Query

- Query: “application and theory”

  $\boldsymbol{q}^T$ = [0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 1].

$$\begin{pmatrix} 0.0511 & -0.3337 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{pmatrix}^T \begin{pmatrix} 0.0159 & -0.4317 \\ 0.0266 & -0.3756 \\ 0.1785 & -0.1692 \\ 0.6014 & 0.1187 \\ 0.6691 & 0.1209 \\ 0.0148 & -0.3603 \\ 0.0520 & -0.2248 \\ 0.0066 & -0.1120 \\ 0.1503 & 0.1127 \\ 0.0813 & 0.0672 \\ 0.1503 & 0.1127 \\ 0.1785 & -0.1692 \\ 0.1415 & 0.0974 \\ 0.0105 & -0.2363 \\ 0.0952 & 0.0399 \\ 0.2051 & -0.5448 \end{pmatrix} \begin{pmatrix} 4.5314 & 0 \\ 0 & 2.7582 \end{pmatrix}^{-1}$$

FIG. 5. *Derived coordinates for the query of application theory.*

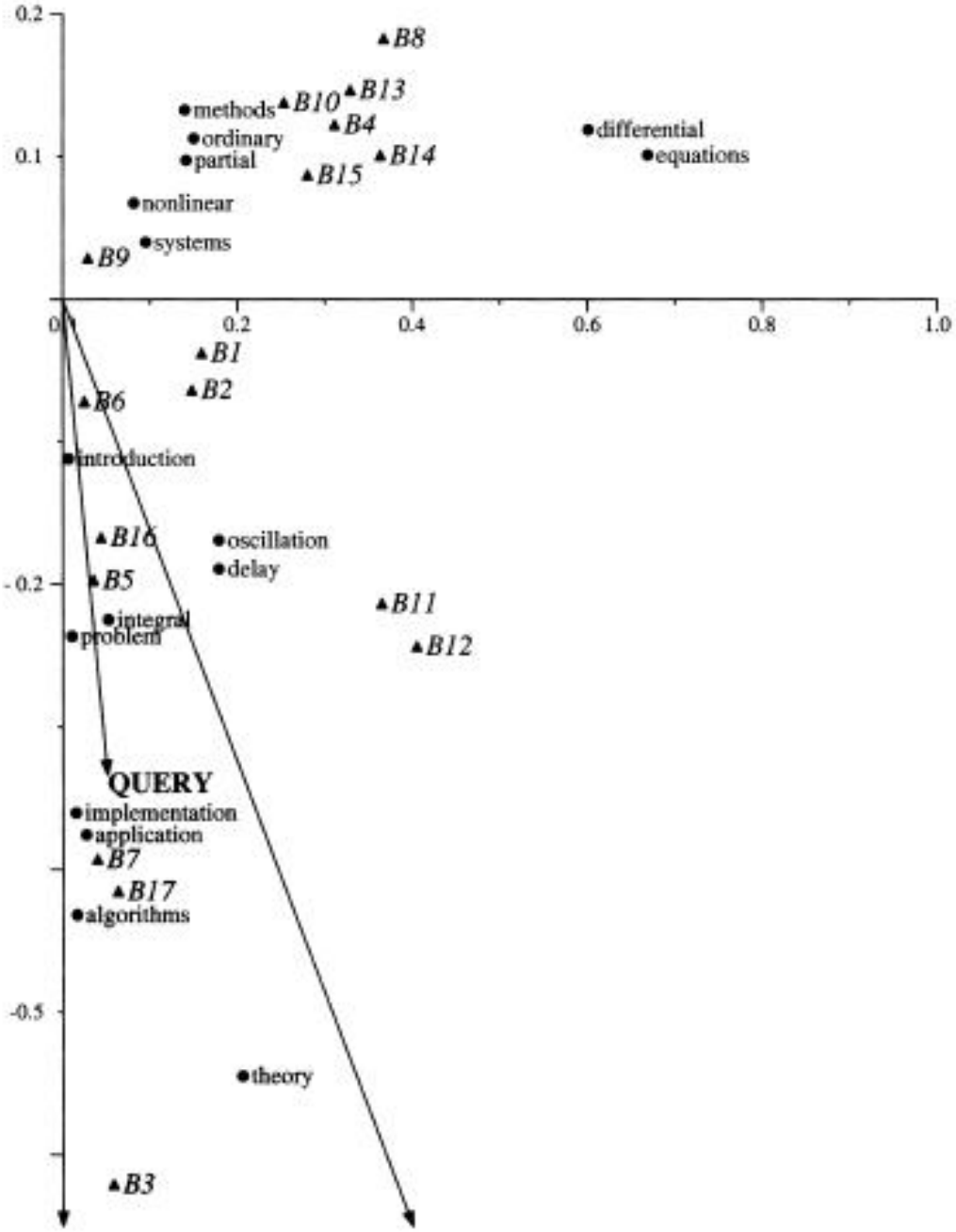


FIG. 6. *A two-dimensional plot of terms and documents along with the query application theory.*

Comparison with Lexical Matching

- Query of “Application and theory”; k=2
- Using a cosine threshold of 0.9 returns 6 books: B_3, B_5, B_6, B_7, B_{16}, and B_{17}.
- Using a cosine threshold 0.55 also returns B_{11} and B_{12} (somewhat related)
- But lexical matching only returns 4 books: B_3, B_{11}, B_{12}, B_{17}

Gene Expression Data Analysis Using SVD

Wall, Michael E., Andreas Rechtsteiner, Luis M. Rocha."Singular value decomposition and principal component analysis". in *A Practical Approach to Microarray Data Analysis*. D.P. Berrar, W. Dubitzky, M. Granzow, eds. pp. 91-109, Kluwer: Norwell, MA (2003). LANL LA-UR-02-4001.

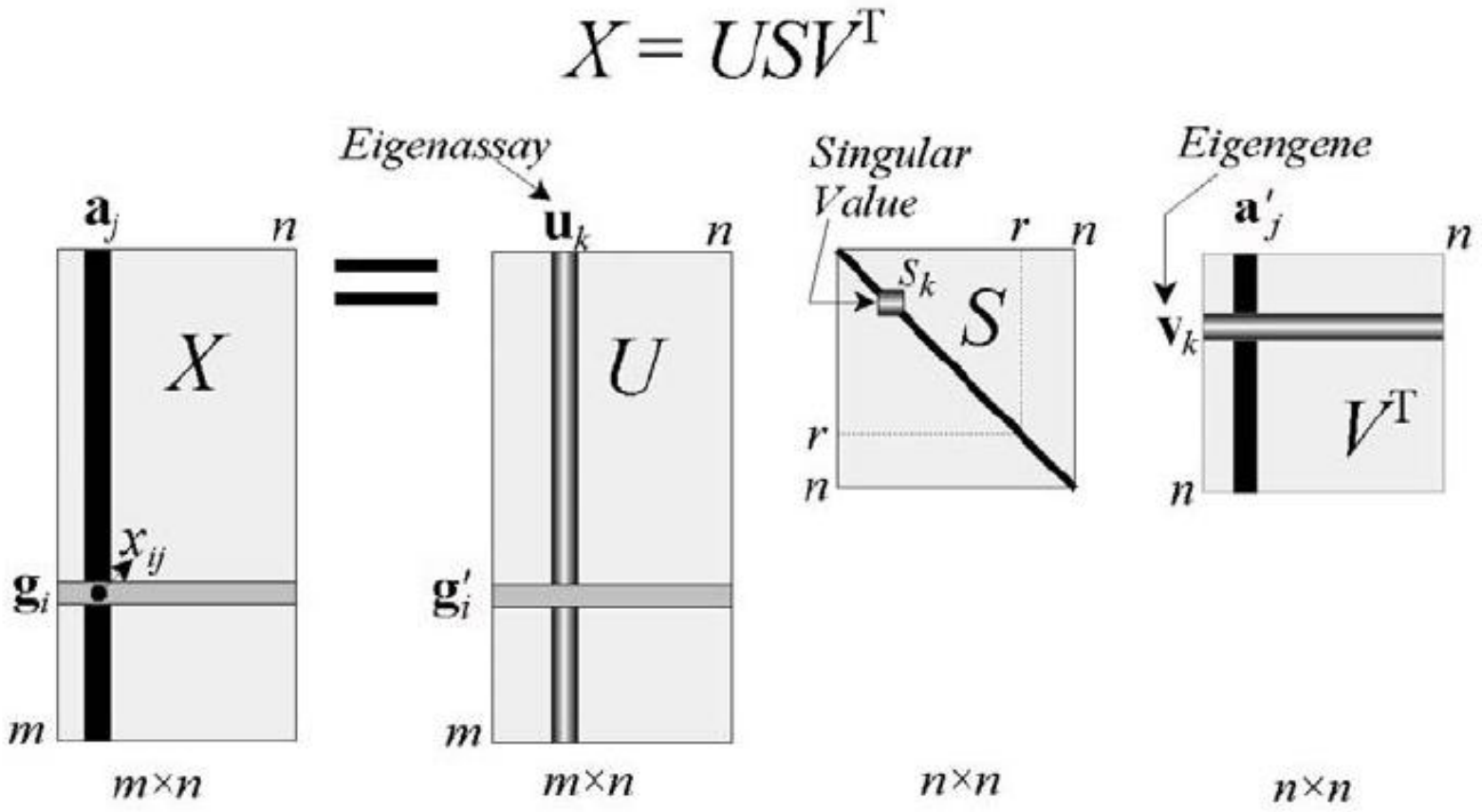


Figure 5.1. Graphical depiction of SVD of a matrix X, annotated with notations adopted in this chapter.

Left Singular Vector and Right Singular Vector

- The right singular vector span the space of the gene transcriptional responses $\{g_i\}$;

- The left singular vectors span the space of the assay expression profiles $\{\mathbf{a}_j\}$

Eigen-vectors

- Genes: linear combination of eigen-genes $\{v_k\}$

$$\vec{g}_i = \sum_{k=1}^{r} u_{ik} s_k \vec{v}_k, \quad i = 1, \cdots, m.$$

- Assays: linear combination of eigen-assays $\{u_k\}$

$$\vec{a}_j = \sum_{k=1}^{r} v_{jk} s_k \vec{u}_k$$

Dimension Reduction

- Using the idea of truncated-SVD, we can represent expression profiles as,

$$\vec{g'}_i = \sum_{k=1}^{d} u_{ik} s_k \vec{v}_k, \quad i = 1, \cdots, m.$$

where d<r=rank(X).

- In another word, we treat the last several singular values only contributing to the noise.

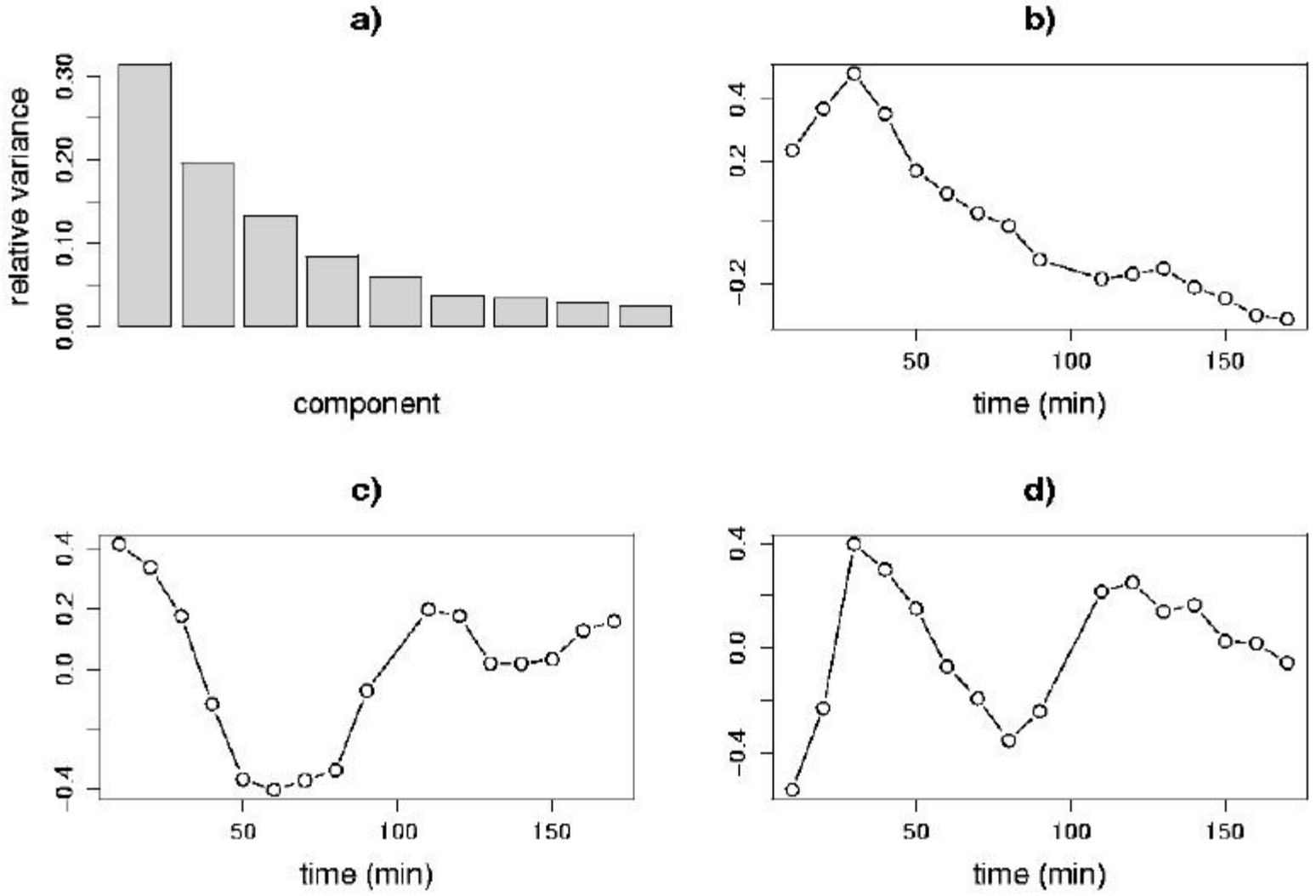


Figure 5.2. Visualization of the SVD of cell cycle data. Plots of relative variance (a); and the first (b), second (c) and third (d) eigengenes are shown. The methods of visualization employed in each panel are described in section 2.1. These data inspired our choice of the sine and exponential patterns for the synthetic data of section 2.1.

Synthetic Time Series Data

- 2000 x 14
- 1600 noisy genes
- 200 genes with noisy sine patterns

$$a \sin(\frac{2\pi t}{140} z), \quad a \sim U(1.5, 3)$$

- 200 genes with exponential patterns

$$b \exp(-\frac{t}{100}), \quad b \sim U(4, 8)$$

Synthetic Data

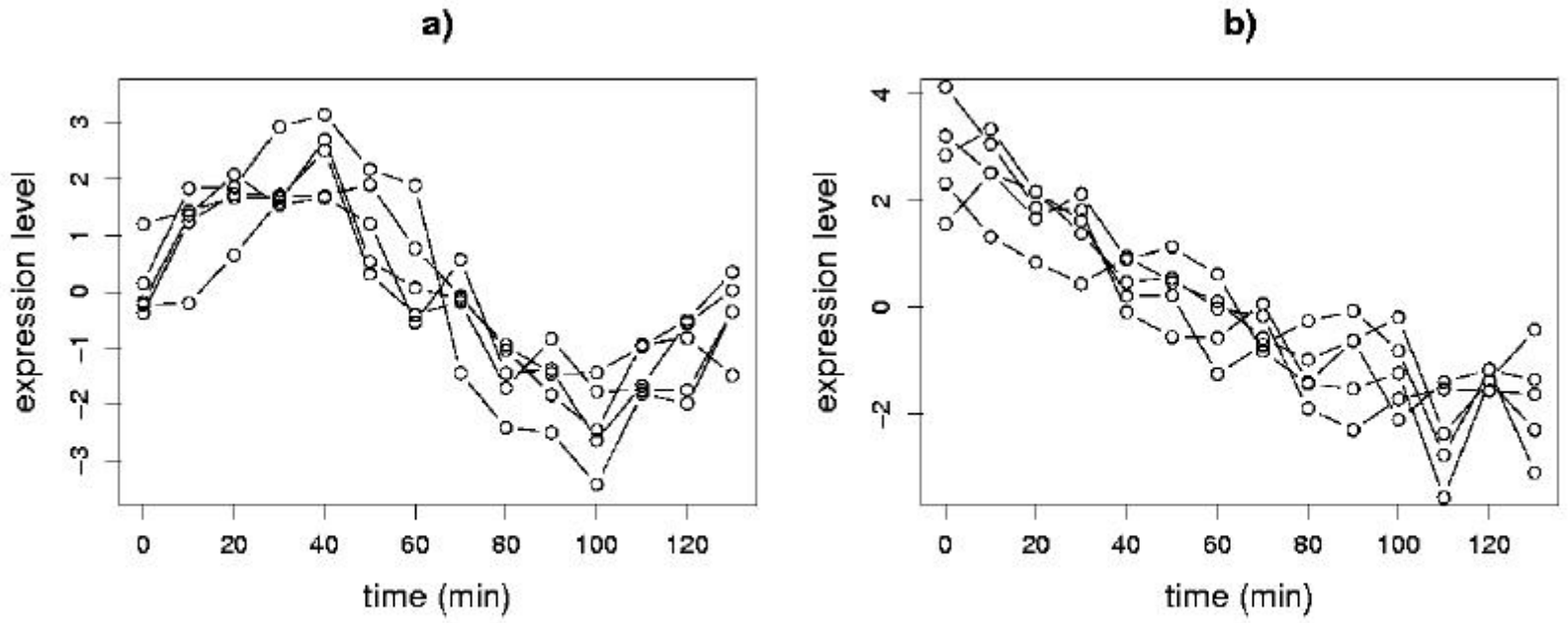


Figure 5.3. Gene transcriptional responses from the synthetic data set. Overlays of a) five noisy sine wave genes and b) five noisy exponential genes.

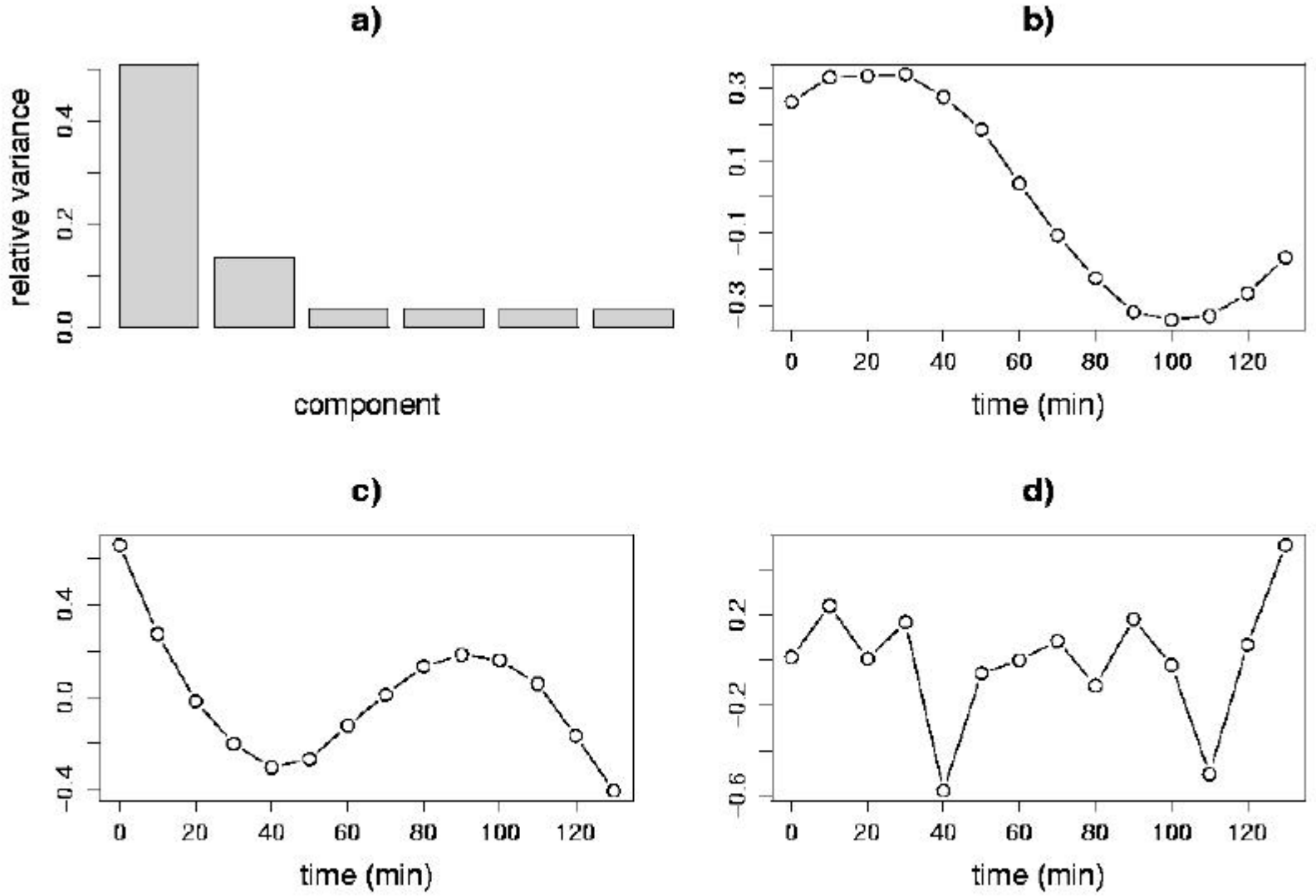


Figure 5.4. Visualization of the SVD of the synthetic data matrix. a) Singular value spectrum in a relative variance plot. The first two singular values account for 64% of the variance. The first (b), second (c), and third (d) eigengenes are plotted vs. time (assays) in the remaining panels. The third eigengene lacks the obvious cyclic structure of the first and second.

Scatter Plots

- Projection of data into SVD subspaces and visualization with scatter plots can reveal structures in the data that may be used for classification.
- Two gene “coordinates” for scatter plots
  - Projection
  - Correlation

Scatter Plots

- Projection of g_i on eigen-gene v_k

$$q_{ik} = \vec{g_i} \cdot \vec{v}_k, XV = US \Rightarrow q_{ik} = (US)_{ik}$$

- Correlation between g_i and eigen-gene v_k

$$r_{ik} = \mathrm{PCC}(\vec{g}_i, \vec{v}_k)$$

$$\mathrm{PCC}(X, Y) = \frac{\sum_i (X_i - \overline{X})(Y_i - \overline{Y})}{\sqrt{\sum_i (X_i - \overline{X})^2 \sum_i (Y_i - \overline{Y})^2}}$$

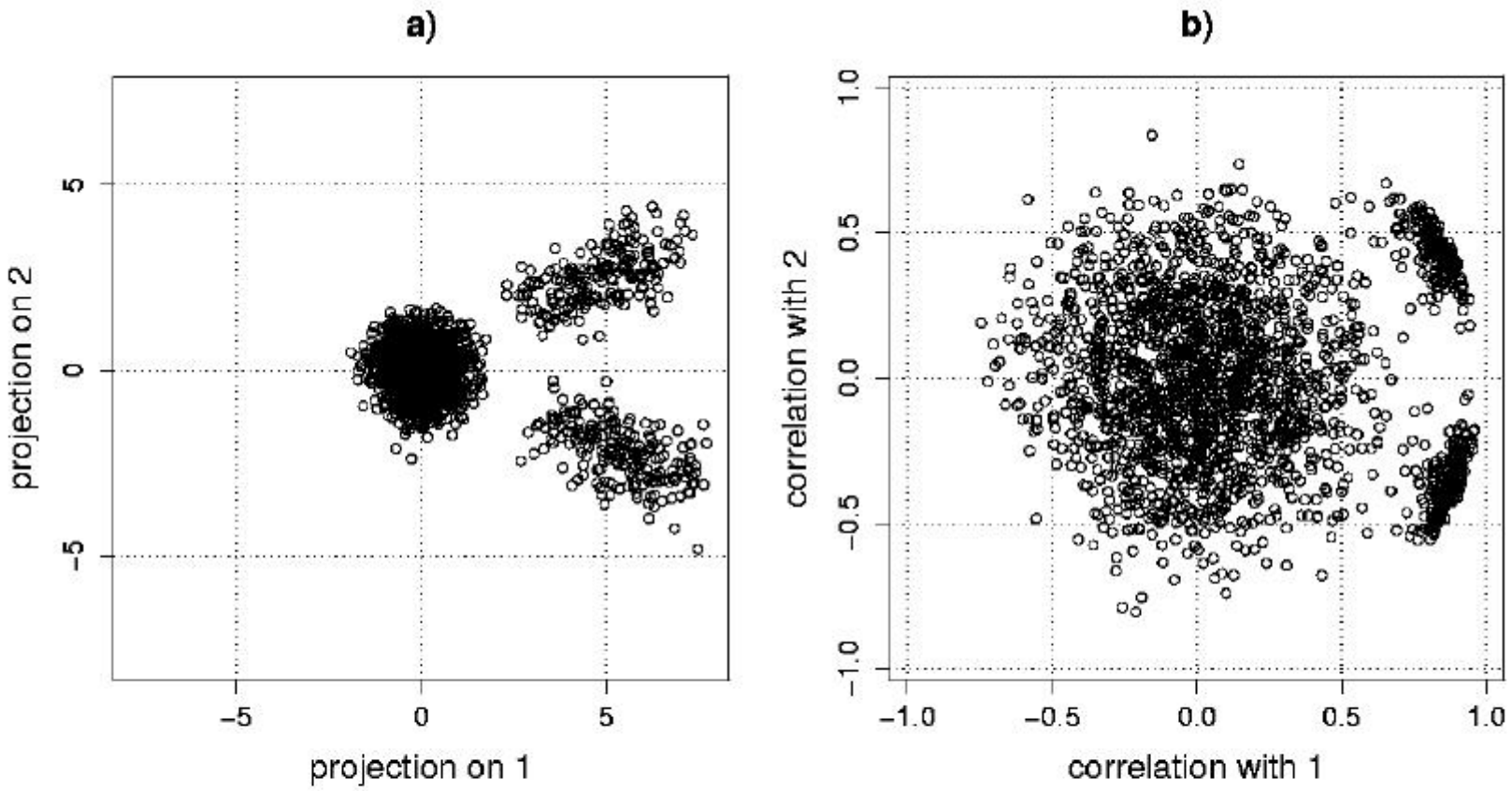


Figure 5.5. SVD scatter plots. Genes from our synthetic example data set are displayed in a) a projection scatter plot; and b) a correlation scatter plot. The bottom right cluster corresponds to sine wave genes, and the top right cluster corresponds to exponential decay genes. The cluster of genes around the origin corresponds to the noise-only genes.

More Applications

- Netflix prize, big data, SVD and R

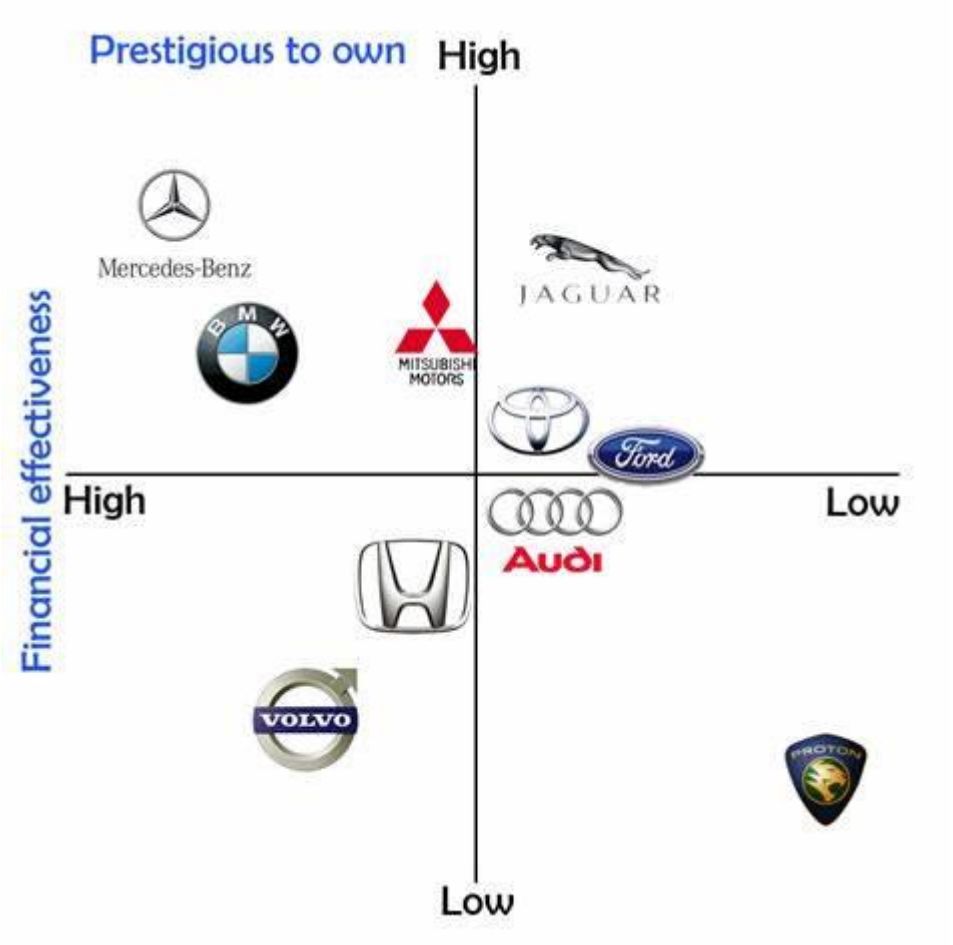
  http://blog.revolutionanalytics.com/2011/05/the-neflix-prize-big-data-svd-and-r.html

Multi-dimensional Scaling (MDS)

部分Slide来自

www.cs.haifa.ac.il/~rita/uml_course/lectures/PCA_MDS.pdf

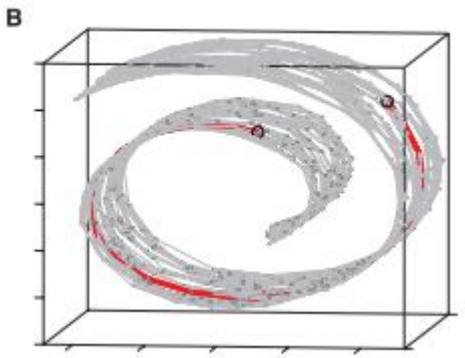
MDS (多维尺度分析)


Prestigious to own
High
Low
High
Low
Financial effectiveness
Mercedes-Benz
JAGUAR
MITSUBISHI MOTORS
Ford
Audi
VOLVO
PROTON

MDS (多维尺度分析)

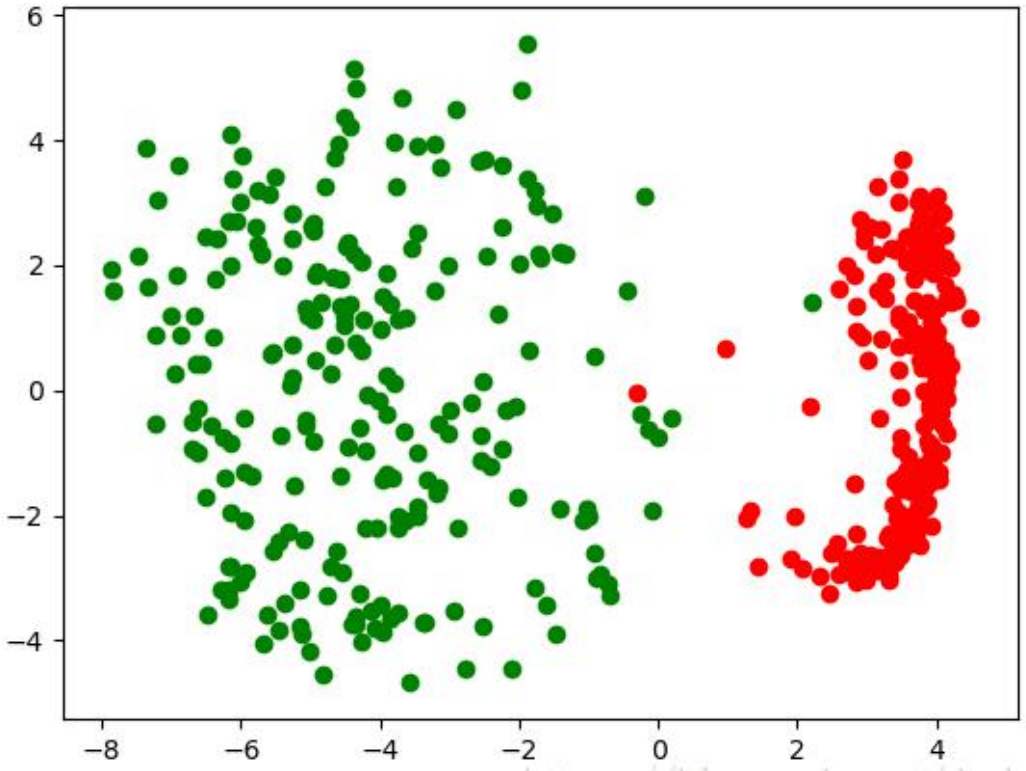
- MDS attempts to preserve pairwise distances.

- Attempts to construct a configuration of *n* points in Euclidian space by using the information about the **distances between the *n* patterns (Not necessary n points in the high dimensional space)** .



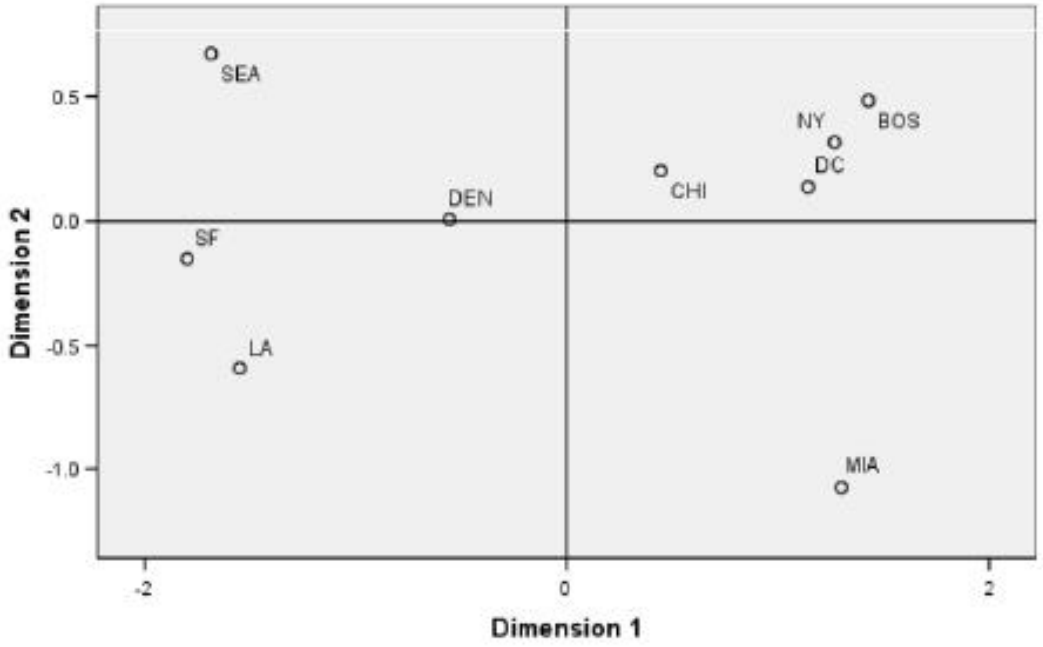
MDS (多维尺度分析)

其主要思想是构造低维空间的内积矩阵，**使得该内积矩阵中所表达的任意两点之间的距离与高维空间的相应两点距离相等**，然后通过对该内积矩阵进行正交特征值分解，析出两个矩阵相乘(即矩阵与矩阵的转置进行相乘)的形式，获得最终的变换矩阵。



Example: Distances between US Cities

	BOS	CHI	DC	DEN	LA	MIA	NY	SEA	SF
BOS	0	963	429	1,949	2,979	1,504	206	2,976	3,095
CHI	963	0	671	996	2,054	1,329	802	2,013	2,142
DC	429	671	0	1,616	2,631	1,075	233	2,684	2,799
DEN	1,949	996	1,616	0	1,059	2,037	1,771	1,307	1,235
LA	2,979	2,054	2,631	1,059	0	2,687	2,786	1,131	379
MIA	1,504	1,329	1,075	2,037	2,687	0	1,308	3,273	3,053
NY	206	802	233	1,771	2,786	1,308	0	2,815	2,934
SEA	2,976	2,013	2,684	1,307	1,131	3,273	2,815	0	808
SF	3,095	2,142	2,799	1,235	379	3,053	2,934	808	0



Classical MDS (I)

- Given n points $X_1, \cdots, X_n \in R^p$, the pairwise dissimilarities

$$X_i = \begin{pmatrix} X_{i1} \\ X_{i2} \\ \cdots \\ X_{ip} \end{pmatrix} \qquad X = \begin{pmatrix} X_{11} & X_{21} & \cdots & X_{n1} \\ X_{12} & X_{22} & \cdots & X_{n2} \\ \cdots & \cdots & \cdots & \cdots \\ X_{1p} & X_{2p} & \cdots & X_{np} \end{pmatrix}_{p \times n}$$

$$\begin{aligned} \delta_{ij}^2 &= (X_i - X_j)^T (X_i - X_j) \\ &= X_i^T X_i + X_j^T X_j - 2 X_i^T X_j \\ &= l_i^2 + l_j^2 - 2 X_i^T X_j \end{aligned}$$

- Where l_i is the length of ith point.

Classical MDS (II)

- Let

$$D = (\delta_i j^2), K = X^T X$$
$$k = diag\{l_1^2, l_2^2, \cdots, l_n^2\}$$

- Then

$$D = kee^T + ee^T k - 2K$$

$$e = \begin{pmatrix} 1 \\ 1 \\ \cdots \\ 1 \end{pmatrix}_{n \times 1}, \quad ee^T = \begin{pmatrix} 1 & 1 & \cdots & 1 \\ 1 & 1 & \cdots & 1 \\ \cdots & \cdots & \cdots & \cdots \\ 1 & 1 & \cdots & 1 \end{pmatrix}_{n \times n}$$

Classical MDS (III)

- Centralization

$$\overline{X} = \frac{1}{n}\sum_{i=1}^{n} X_i = \frac{1}{n} e^T X$$

$$\tilde{X} = X - e\overline{X} = X - \frac{1}{n} X e e^T$$

- Let

$$H = I_n - \frac{1}{n} e e^T$$

$$\tilde{X} = XH$$

Classical MDS (IV)

- We have

$$H = \frac{1}{n}\begin{pmatrix} n-1 & -1 & \cdots & -1 \\ -1 & n-1 & \cdots & -1 \\ \cdots & \cdots & \cdots & \cdots \\ -1 & -1 & \cdots & n-1 \end{pmatrix}$$

$$\begin{aligned} H^2 &= (I_n - \frac{1}{n}ee^T)(I_n - \frac{1}{n}ee^T) \\ &= I_n - \frac{2}{n}ee^T + \frac{1}{n^2}e(e^Te)e^T \\ &= H \end{aligned} \qquad H = H^T$$

Classical MDS (V)

- Let

$$\tilde{K} = \tilde{X}^T \tilde{X}$$

$$\tilde{K} = (XH)^T (XH) = H^T X^T X H = HKH$$

- Let

$$B = -\frac{1}{2} HDH$$

Classical MDS (VI)

- Then

$$\begin{aligned} B &= -\frac{1}{2}HDH \\ &= -\frac{1}{2}H(kee^T + ee^Tk - 2K)H^T \\ &= HKH^T - \frac{1}{2}Hkee^TH^T - \frac{1}{2}Hee^TkH^T \\ &= HKH^T = \tilde{K} \end{aligned}$$

- Here

$$He = (0, 0, \cdots, 0)^T$$

Classical MDS (VII)

- Finally, we have

$$B_{n\times n} = \tilde{K}_{n\times n} = \tilde{X}^T\tilde{X} \geq 0$$

- Then we can have a SVD of B (p<n)

$$B = V\Lambda V^T$$

$$\Lambda = diag\{\lambda_1, \lambda_2, \cdots, \lambda_p, 0, \cdots, 0\}$$

$$\lambda_1 \geq \lambda_2 \geq \cdots \lambda_p > 0$$

where V is a matrix whose columns are the eigenvectors of *B*

Classical MDS (VIII)

- With p-eignvectors, we can reconstruct B

$$(\hat{X}_p)_{p\times n} = diag\{\sqrt{\lambda_1}, \cdots, \sqrt{\lambda_p}\}(v_1, \cdots, v_p)^T$$

$$B = (\hat{X}_p)^T \hat{X}_p$$

- With k-eignvectors, we can reduce the dimension to k-dimensional space

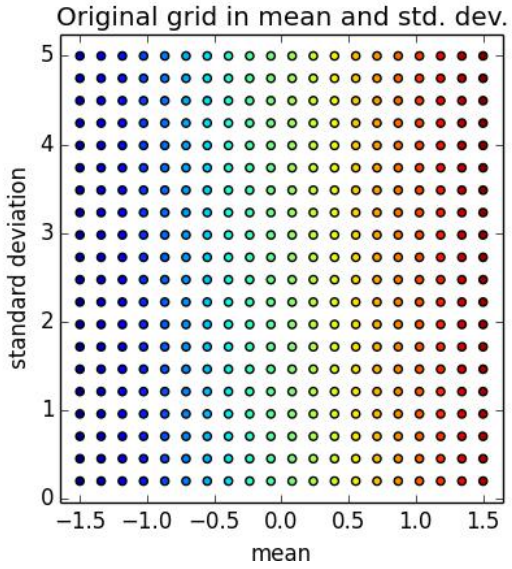
$$Y_{k\times n} = diag\{\sqrt{\lambda_1}, \cdots, \sqrt{\lambda_k}\}(v_1, \cdots, v_k)^T$$

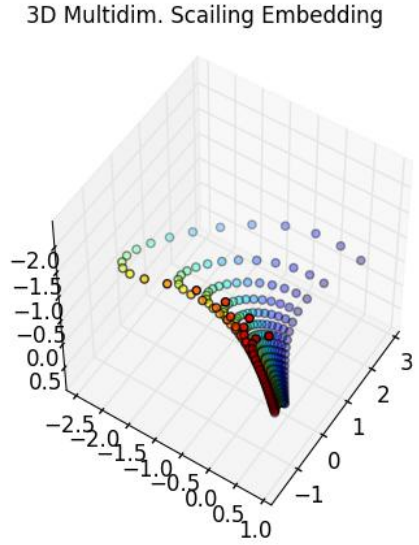
Classical MDS (IX)

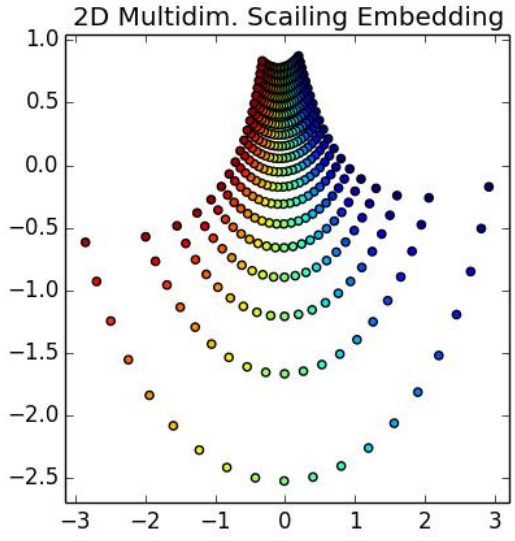
- Optimality property

Theorem 1. *Let X denote a configuration of points in $\mathbb{R}^p$, with interpoint distances $\delta_{ij}^2 = (\mathbf{x}_i - \mathbf{x}_j)^T(\mathbf{x}_i - \mathbf{x}_j)$. Let L be a $p \times p$ rotation matrix and set $L = (L_1, L_2)$, where L_1 is $p \times k$ for $k < p$. Let $\hat{X} = XL_1$, the projection of X onto a k-dimensional subspace of $\mathbb{R}^p$, and let $\hat{d}_{ij}^2 = (\hat{\mathbf{x}}_i - \hat{\mathbf{x}}_j)^T(\hat{\mathbf{x}}_i - \hat{\mathbf{x}}_j)$. Amongst all projections $\hat{X} = XL_1$, the quantity $\phi = \sum_{i,j}(\delta_{ij}^2 - \hat{d}_{ij}^2)$ is minimized when X is projected onto its principal coordinates in k dimensions. For all i, j we have $\hat{d}_{ij} \leq \delta_{ij}$. The value of ϕ for the principal coordinate projection is $\phi = 2n(\lambda_{k+1} + \cdots + \lambda_p)$.*

MDS (多维尺度分析)


Original grid in mean and std. dev.
standard deviation
mean
0
1
2
3
4
5
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5


3D Multidim. Scailing Embedding
−2.0
−1.5
−1.0
−0.5
0.0
0.5
−2.5
−2.0
−1.5
−1.0
−0.5
0.0
0.5
1.0
3
2
1
0
−1


2D Multidim. Scailing Embedding
1.0
0.5
0.0
−0.5
−1.0
−1.5
−2.0
−2.5
−3
−2
−1
0
1
2
3

Relation to PCA

	PCA	**MDS**
Spectral Decomposition	Covariance matrix (D x D)	Gram matrix (n x n)
Eigenvalues	Matrices share nonzero eigenvalues up to constant factor	
Results	Same	
Computation	$O((n+d)D^2)$	$O((D+d)n^2)$

Non-metric MDS

- Transform pairwise distances: $\delta_{ij} \rightarrow g(\delta_{ij})$
  - Transformation: nonlinear, but monotonic.
  - Preserves rank order of distances.

- Find vectors y_i such that $\|y_i - y_j\| \approx g(\delta_{ij})$

$$Cost = \min_{y} \sum_{ij} \left(g(\delta_{ij}) - \|y_i - y_j\| \right)^2$$

Non-metric MDS

- Possible objective function:

$$Cost = \sum_{i\,j} \left(\frac{\| \mathbf{x}_i - \mathbf{x}_j \| - \| \mathbf{y}_i - \mathbf{y}_j \|}{\| \mathbf{x}_i - \mathbf{x}_j \|} \right)^2$$

Non-metric MDS

- Strengths
  - Relaxes distance constraints.
  - Yields nonlinear embeddings.
- Weaknesses
  - Highly nonlinear, iterative optimization with local minima.
  - Unclear how to choose distance
  - transformation.

MDS Implementation

- MDS in R
  - isoMDS(MASS)
    - Kruskal's Non-metric Multidimensional Scaling
  - cmdscale(stats)
    - Classical (Metric) Multidimensional Scaling
  - sammon(MASS)
    - Sammon's Non-Linear Mapping
- Various software and resources about MDS

  http://www.granular.com/MDS/

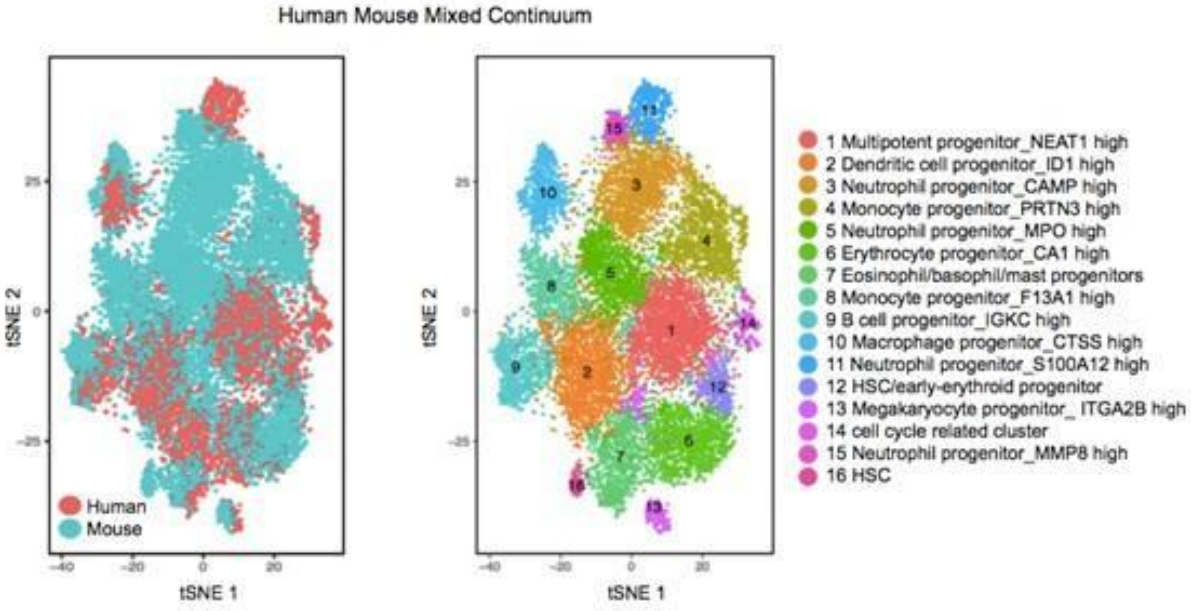
Other Methods

- Nonlinear dimension reduction
  - Kernel PCA
  - Locally linear embedding (LLE)
  - Isomap
  - t-SNE
  - LASSO

- A good tutorial： Ali Ghodsi. Dimensionality Reduction, A Short Tutorial.
  http://www.math.uwaterloo.ca/~aghodsib/courses/s09stat946/readings/tutorial_stat890.pdf
- More materials can be found at his webpage
  http://www.math.uwaterloo.ca/~aghodsib/courses/s09stat946/

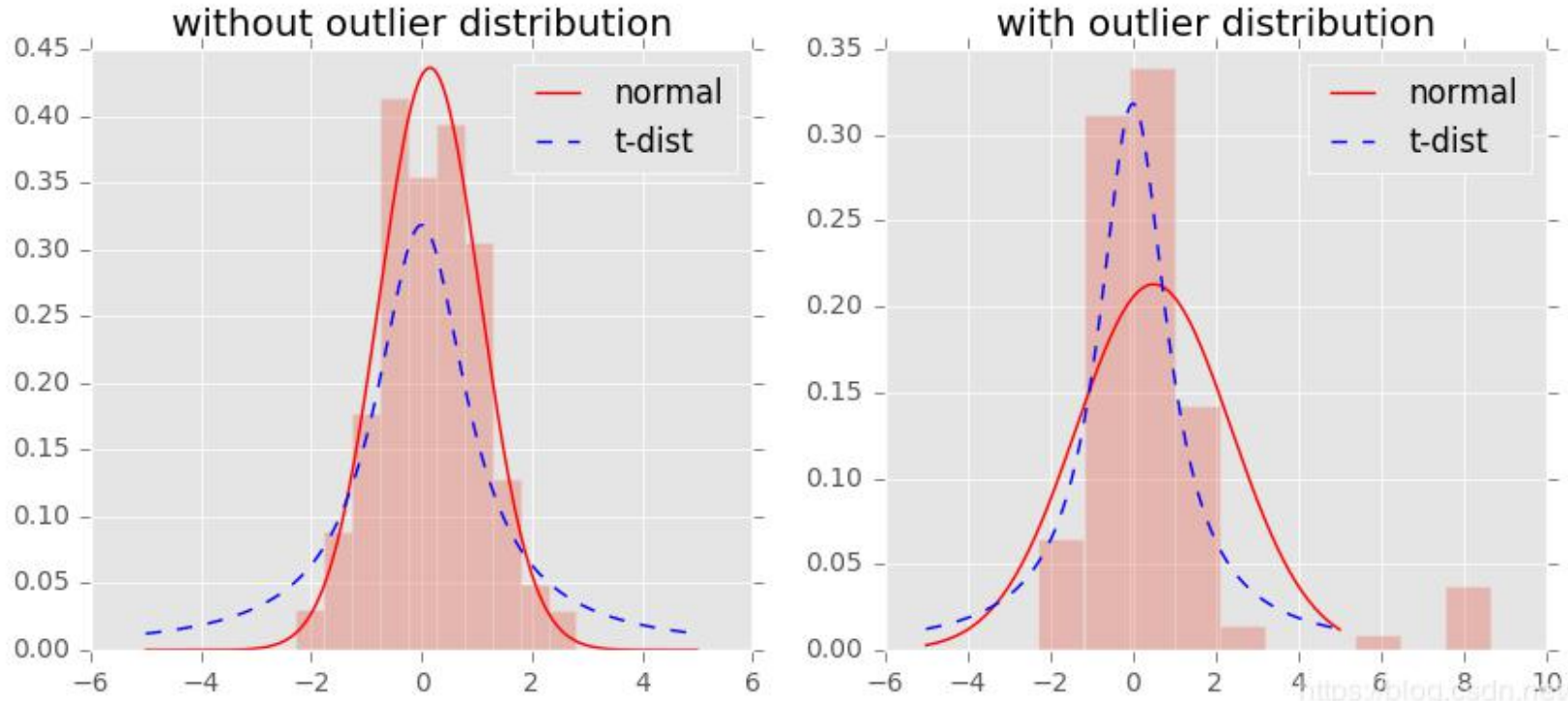
t-SNE(t-distributed stochastic neighbor embedding)

- SNE是通过仿射(affinitie)变换将数据点映射到概率分布上，主要包括两个步骤：
  - SNE构建一个高维对象之间的概率分布，使得相似的对象有更高的概率被选择，而不相似的对象有较低的概率被选择。
  - SNE在低维空间里在构建这些点的概率分布，使得这两个概率分布之间尽可能的相似。



t-SNE(t-distributed stochastic neighbor embedding)

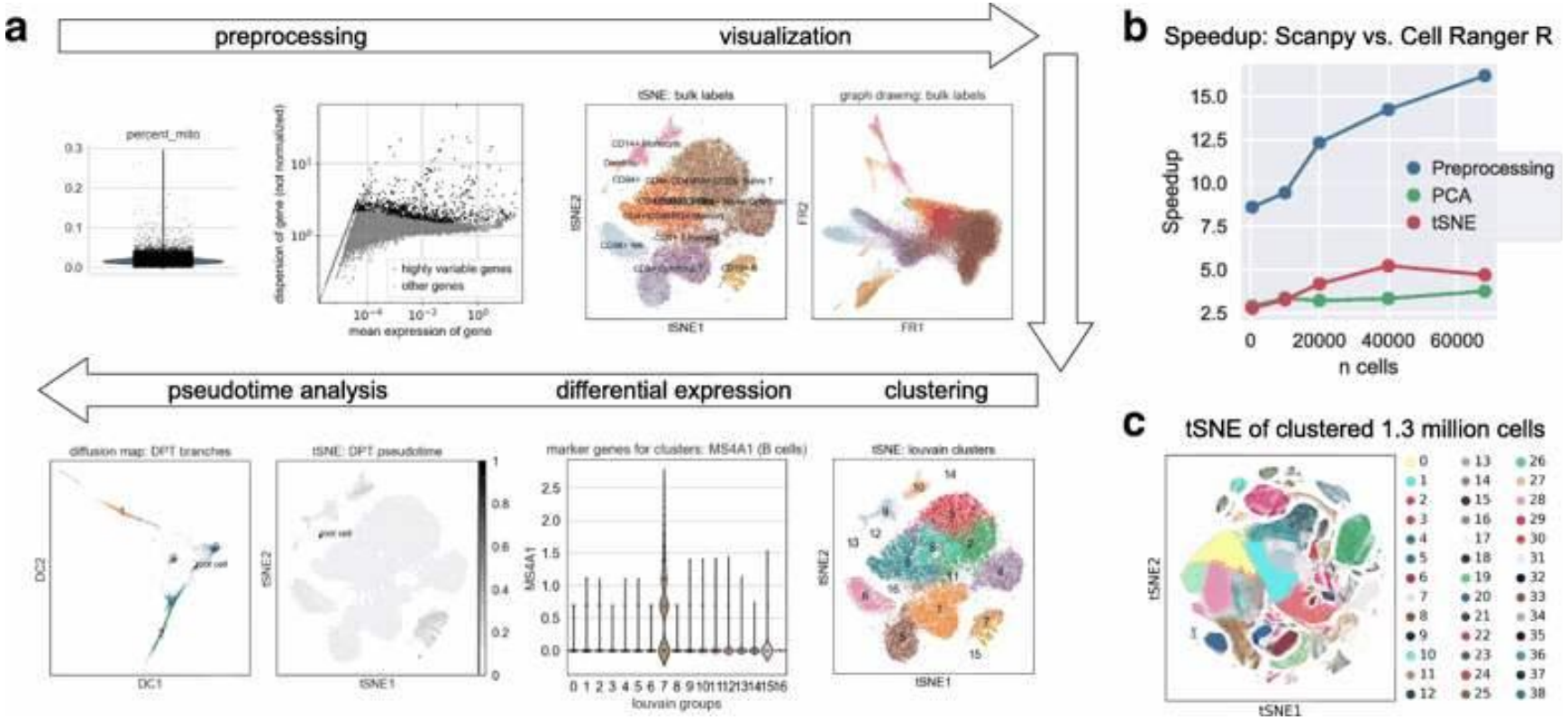
- 对称SNE实际上在高维度下 另外一种减轻”拥挤问题”的方法：在高维空间下，在高维空间下我们使用高斯分布将距离转换为概率分布，在低维空间下，我们使用更加偏重长尾分布的方式来将距离转换为概率分布，使得高维度下中低等的距离在映射后能够有一个较大的距离。



t-SNE(t-distributed stochastic neighbor embedding)

t-SNE(t-distributed stochastic neighbor embedding)

Single-cell applications



t-SNE(t-distributed stochastic neighbor embedding)

- 主要不足有四个:
  - 主要用于可视化，很难用于其他目的。比如测试集合降维，因为他没有显式的预估部分，不能在测试集合直接降维；比如降维到10维，因为t分布偏重长尾，1个自由度的t分布很难保存好局部特征，可能需要设置成更高的自由度
  - t-SNE倾向于保存局部特征，对于本征维数(intrinsic dimensionality)本身就很高的数据集，是不可能完整的映射到2-3维的空间
  - t-SNE没有唯一最优解，且没有预估部分。如果想要做预估，可以考虑降维之后，再构建一个回归方程之类的模型去做。但是要注意，t-sne中距离本身是没有意义，都是概率分布问题
  - 训练太慢。

LASSO

- Lasso是一种数据降维方法
- 不仅适用于线性情况，也适用于非线性情况
- Lasso是基于惩罚方法对样本数据进行变量选择
  - 通过对原本的系数进行压缩，将原本很小的系数直接压缩至0
  - 将这部分系数所对应的变量视为非显著性变量
  - 将不显著的变量直接舍弃

Lasso Model

(least absolute shrinkage and selection operator)

- Lasso: Least Absolute Shrinkage and Selection Operator
- Minimize

$$\min_{\beta} \sum_{i=1}^{n} \frac{1}{2}(y_i - \sum_{j=1}^{p} \beta_j x_{ij})^2 + \lambda \sum_{j=1}^{p} |\beta_j|$$

- Equivalent to minimizing sum of squares with constraint (Lagarangian function)

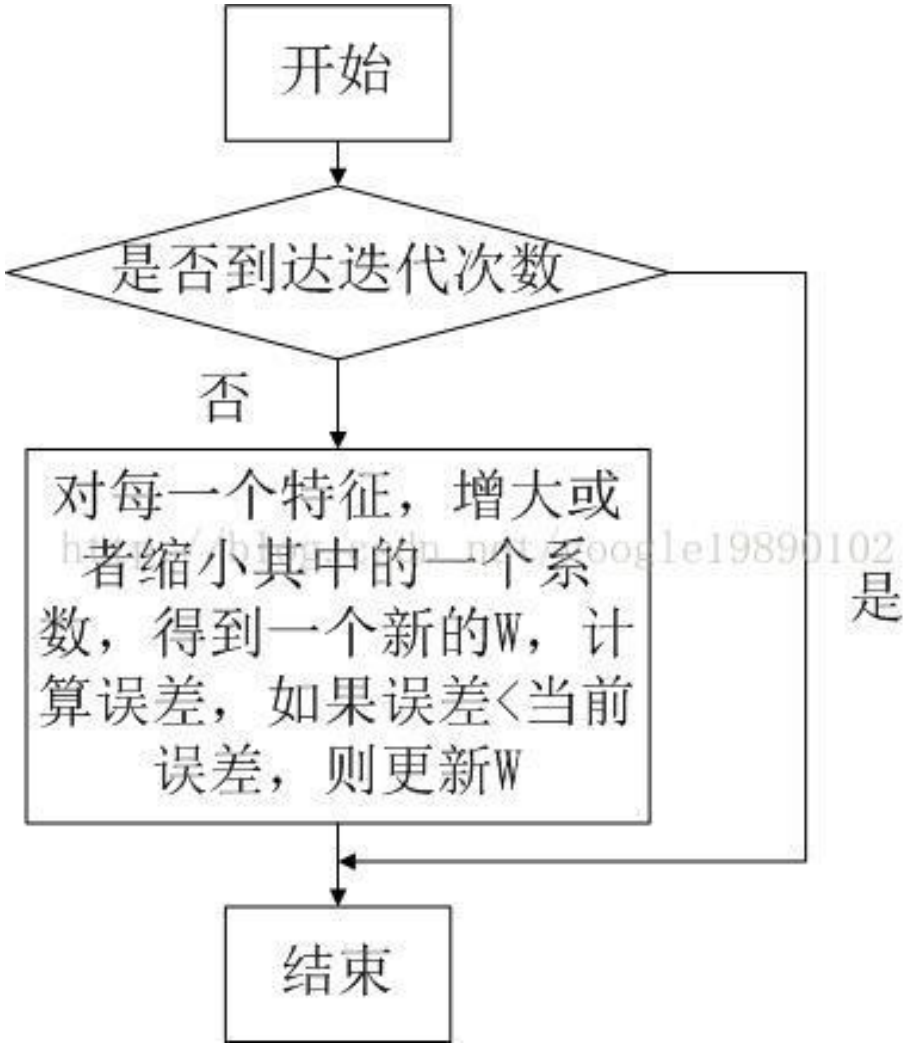
$$\sum_{j=1}^{p} |\beta_j| \leq s$$

Lasso Explanation

- The bound "s" is a tuning parameter. When "s" is large enough, the constraint has no effect and the solution is just the usual multiple linear least squares regression of y on x_1, x_2, ...x_p.
- However when for smaller values of s (s>=0) the solutions are shrunken versions of the least squares estimates. Often,some of the coefficients b_j are zero. Choosing "s" is like choosing the number of predictors to use in a regression model, and cross-validation is a good tool for estimating the best value for "s".

http://statweb.stanford.edu/~tibs/lasso/simple.html

Algorithms for Lasso



Algorithms for Lasso

- Standard convex optimizer
- Least angle regression (LAR) - Efron et al 2004- computes
- Entire path of solutions. State-of-the-Art until 2008
- Pathwise coordinate descent---New

Ridge Regression

- Minimize

$$\min_{\beta} \frac{1}{2} \sum_{i=1}^{n} (y_i - \sum_{j=1}^{p} \beta_j x_{ij})^2 + \lambda \sum_{j=1}^{p} |\beta_j^2|$$

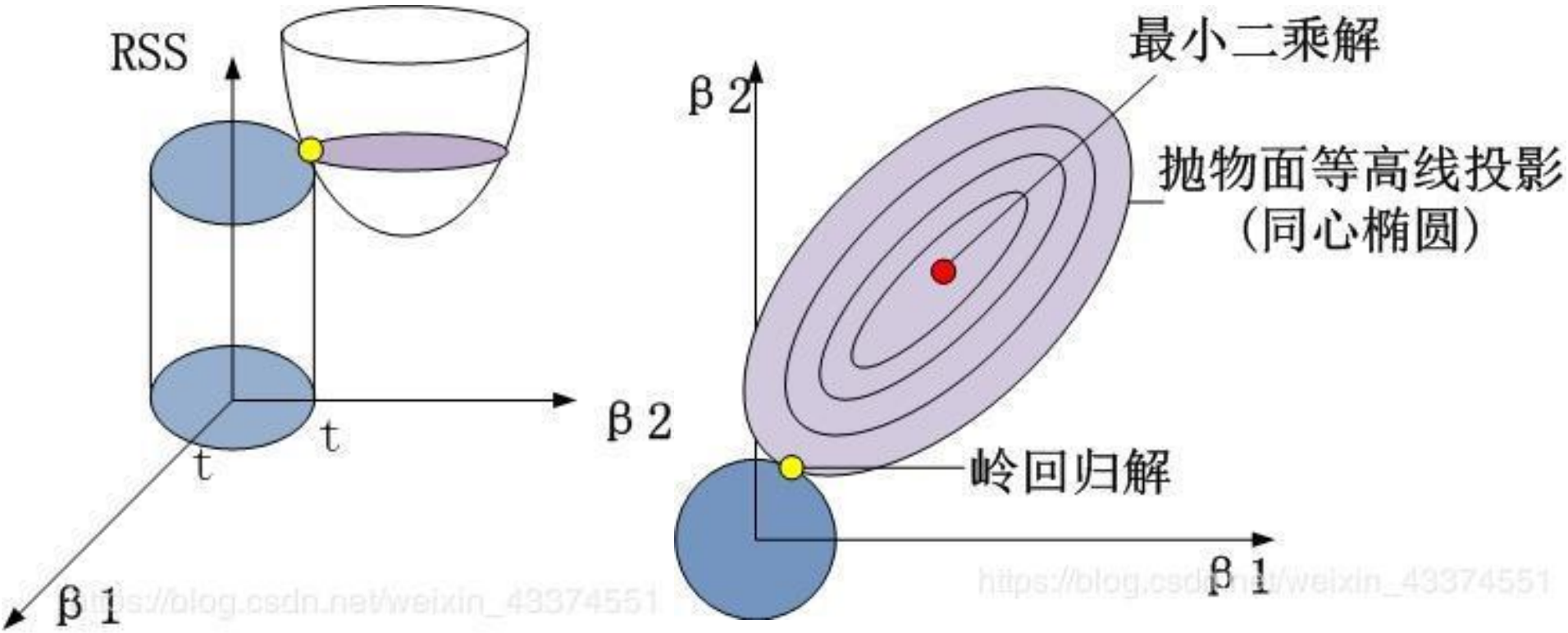
- Equivalent to minimizing sum of squares with constraint

$$\sum_{j=1}^{p} |\beta_j|^2 \leq s$$

- Close-form solution

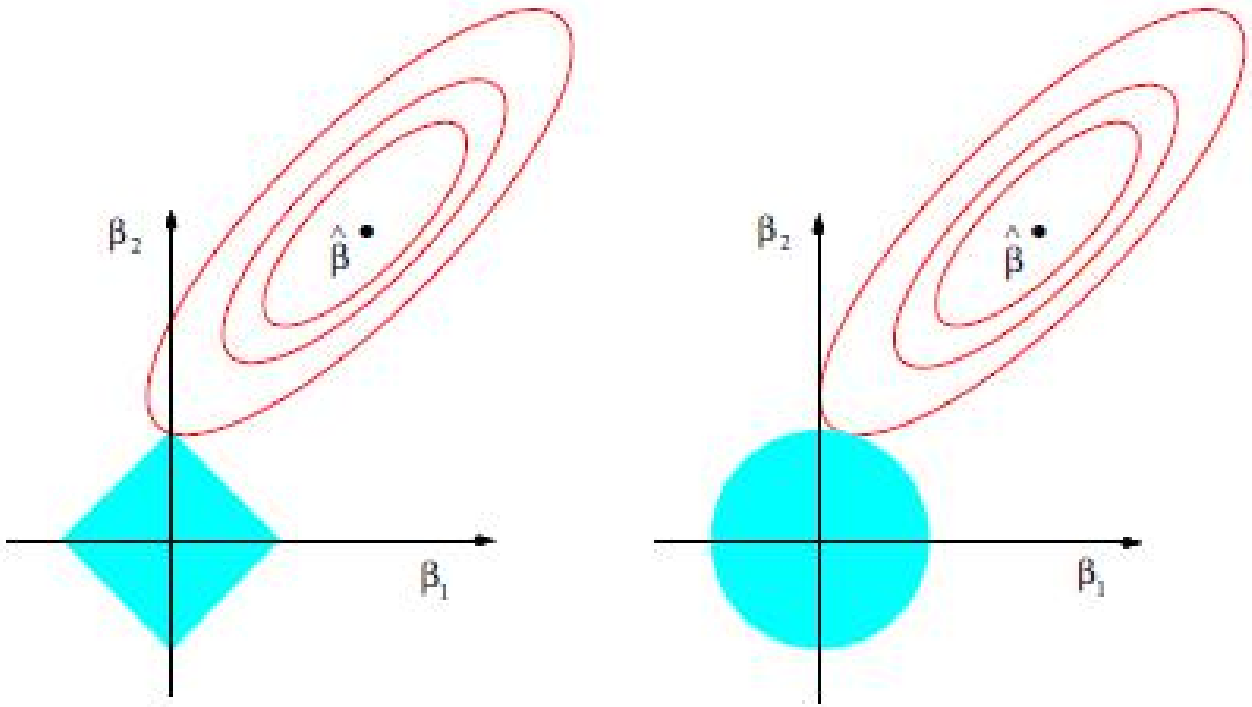
$$\beta^{\text{ridge}} = (X^T X + \lambda I)^{-1} X^T Y$$

Ridge Regression



Ridge方法

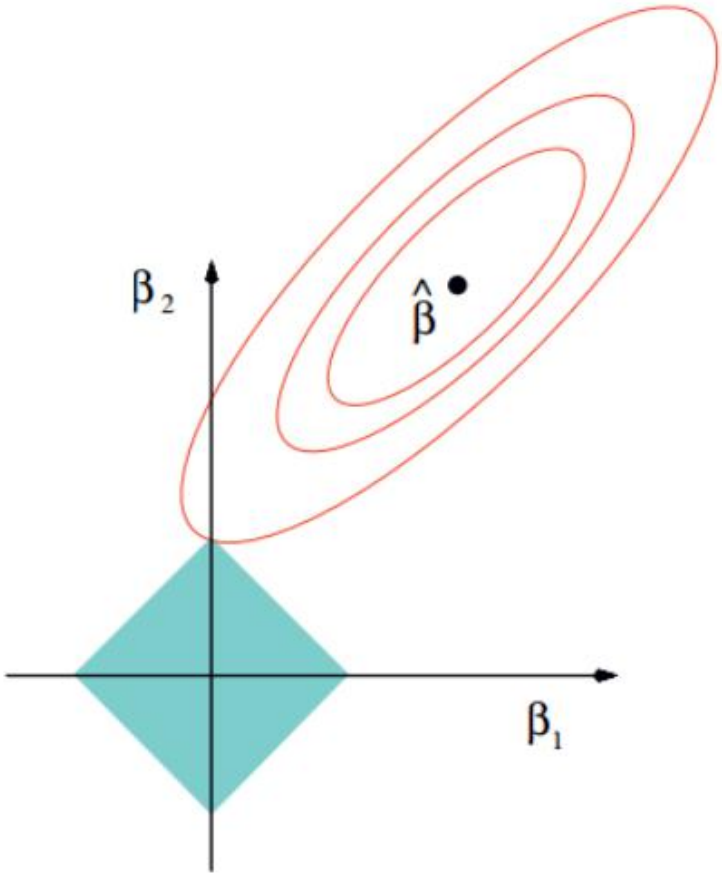
Lasso and Ridge Regression

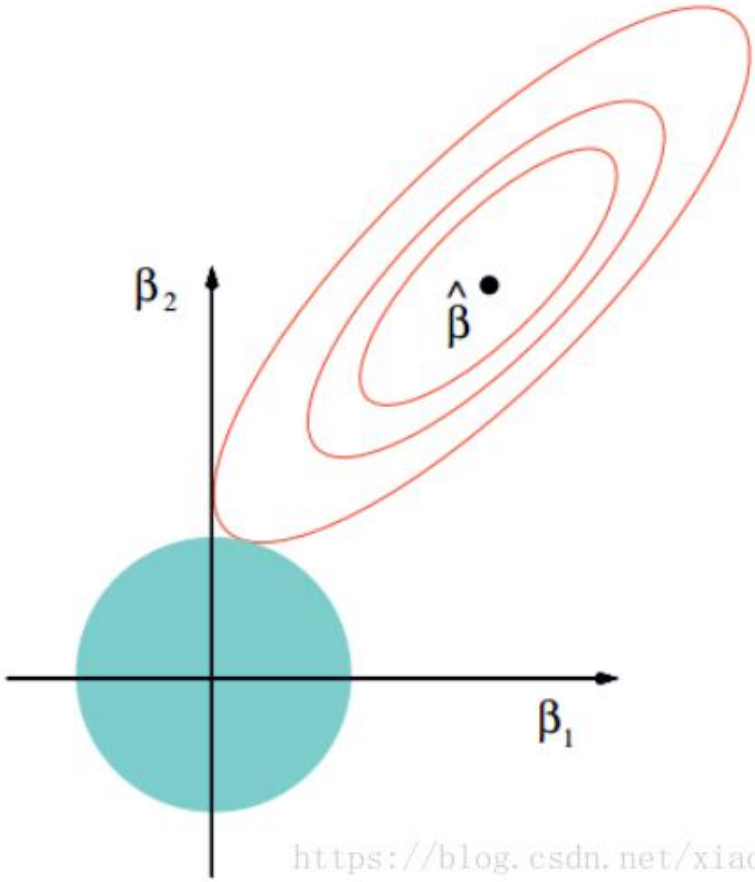


左图：只要不是特殊情况下与正方形的边相切，一定是与某个顶点优先相交，那必然存在横纵坐标轴中的一个系数为0，起到对变量的筛选的作用。

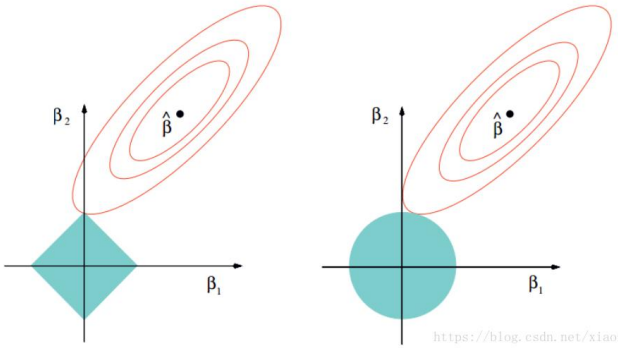
右图：这个圆的限制下，点可以是圆上的任意一点，所以q＝2的时候也叫做**岭回归**，岭回归是起不到压缩变量的作用的，在这个图里也是可以看出来的。

LASSO

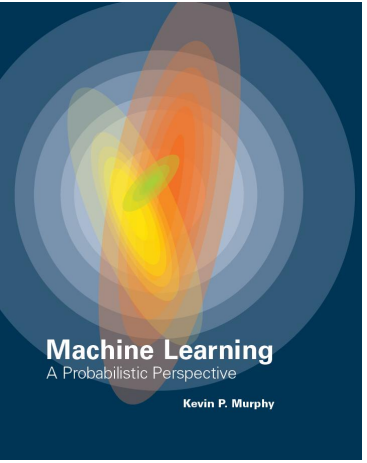




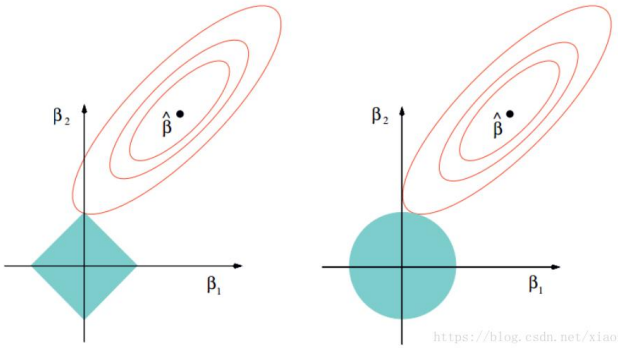
LASSO

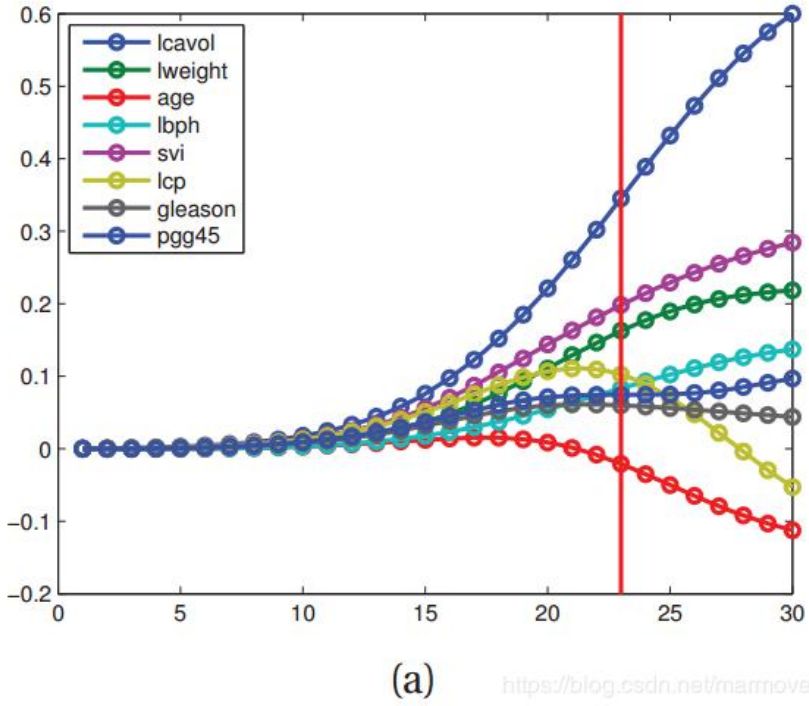


- 以二维数据空间为例，说明lasso和Ridge两种方法的差异，左图对应于Lasso方法，右图对应于Ridge方法。
- 两个图是对应于两种方法的等高线与约束域。
- 红色的椭圆代表的是随着λ的变化所得到的残差平方和，β^为椭圆的中心点，为对应普通线性模型的最小二乘估计。
- 左右两个图的区别在于约束域，即对应的蓝色区域。
- 等高线和约束域的切点就是目标函数的最优解，Ridge方法对应的约束域是圆，其切点只会存在于圆周上，不会与坐标轴相切，则在任一维度上的取值都不为0，因此没有稀疏；对于Lasso方法,其约束域是正方形，会存在与坐标轴的切点，使得部分维度特征权重为0，因此很容易产生稀疏的结果。
- 所以，Lasso方法可以达到变量选择的效果，将不显著的变量系数压缩至0，而Ridge方法虽然也对原本的系数进行了一定程度的压缩，但是任一系数都不会压缩至0，最终模型保留了所有的变量。

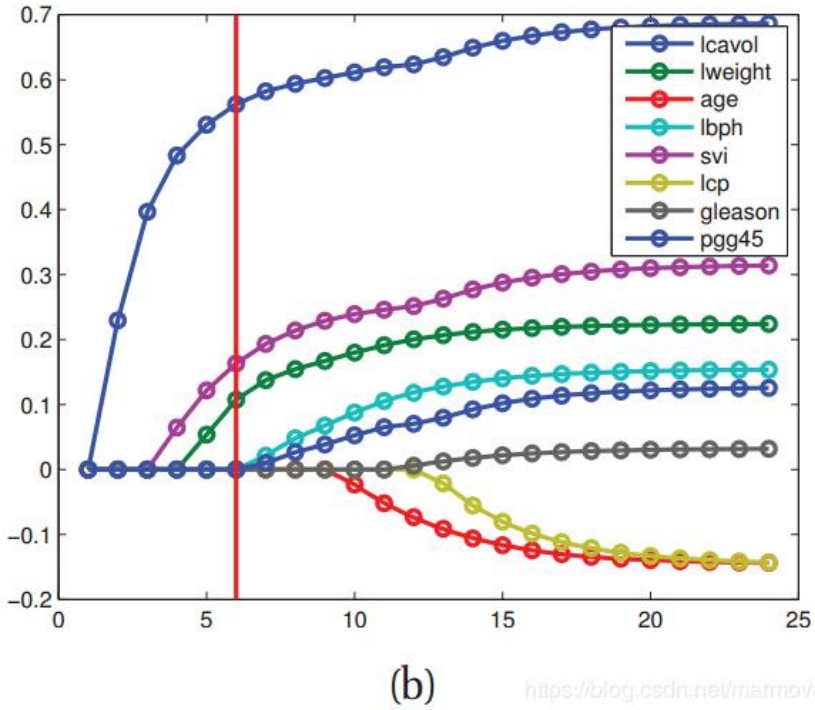


LASSO





(a)

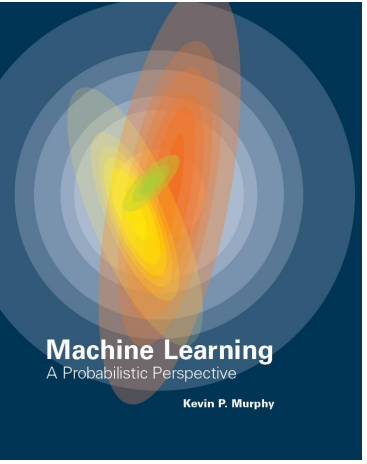


(b)

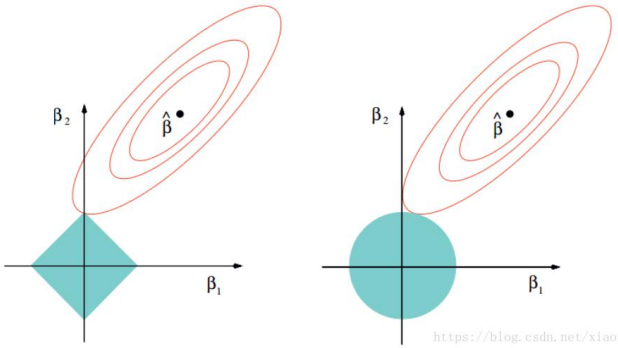
Ridge方法

https://blog.csdn.net/marmove/article/details/85260241

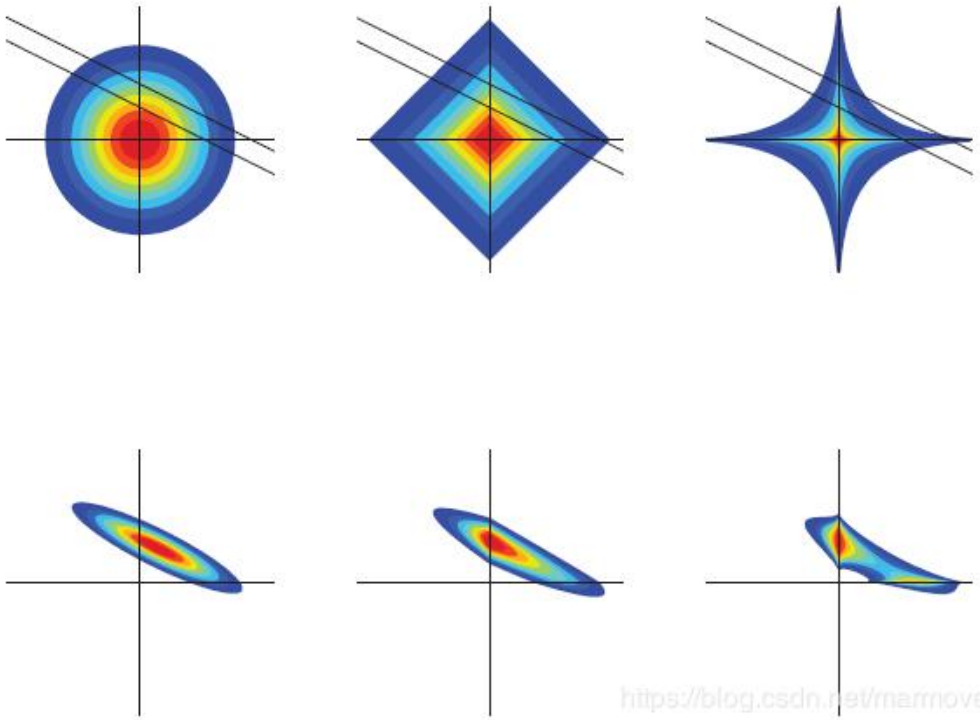
LASSO方法



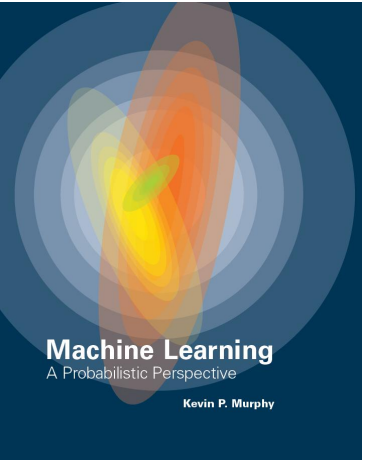
LASSO



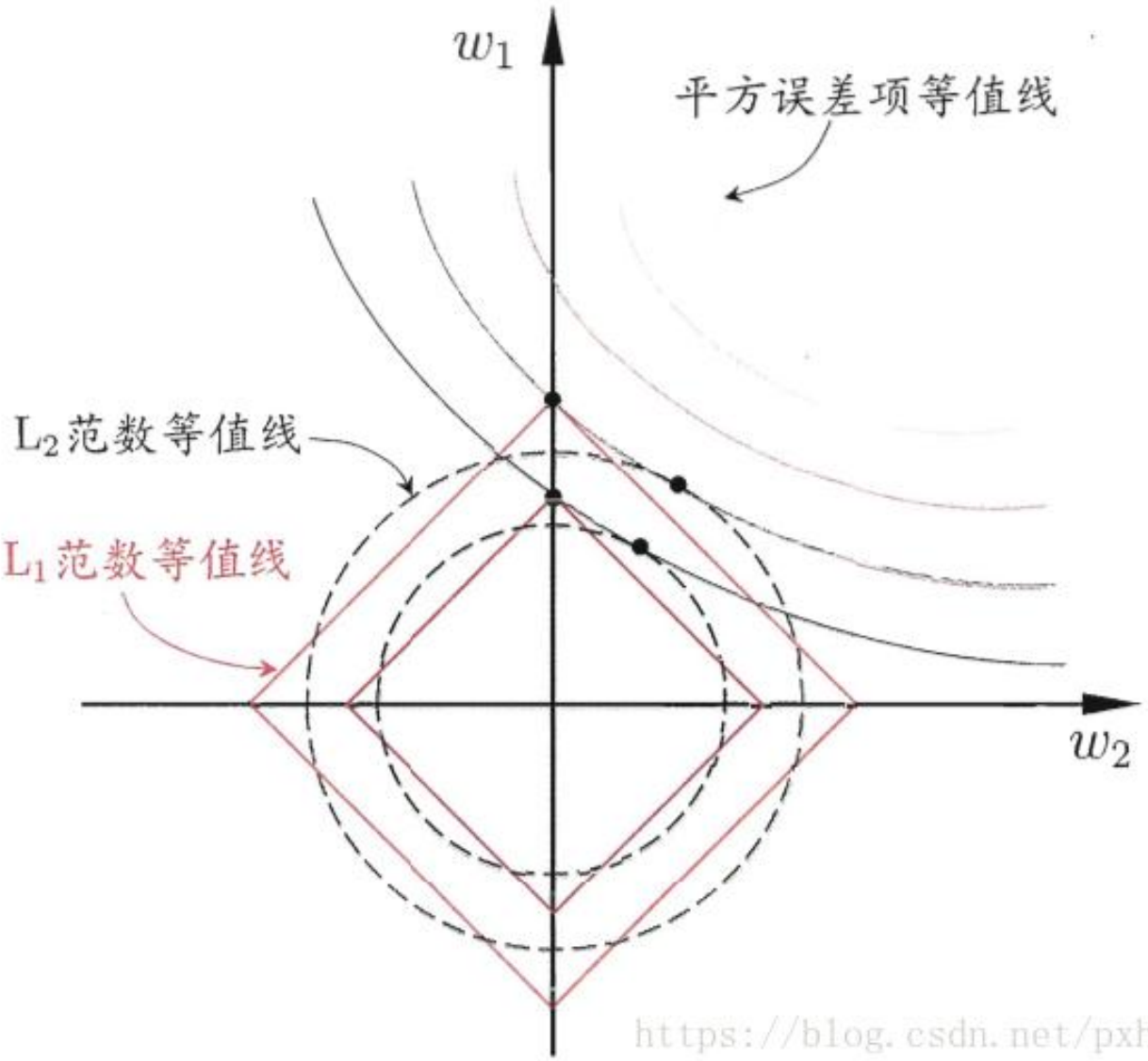
更加灵活的先验：Bridge回归方法

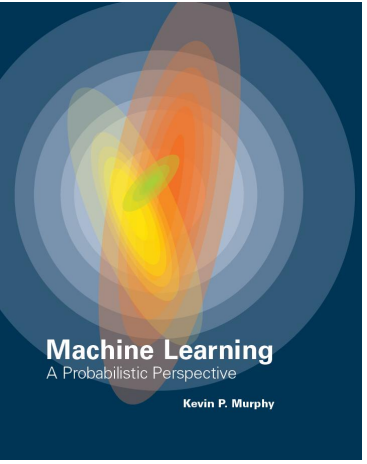


https://blog.csdn.net/marmove
/article/details/85260241

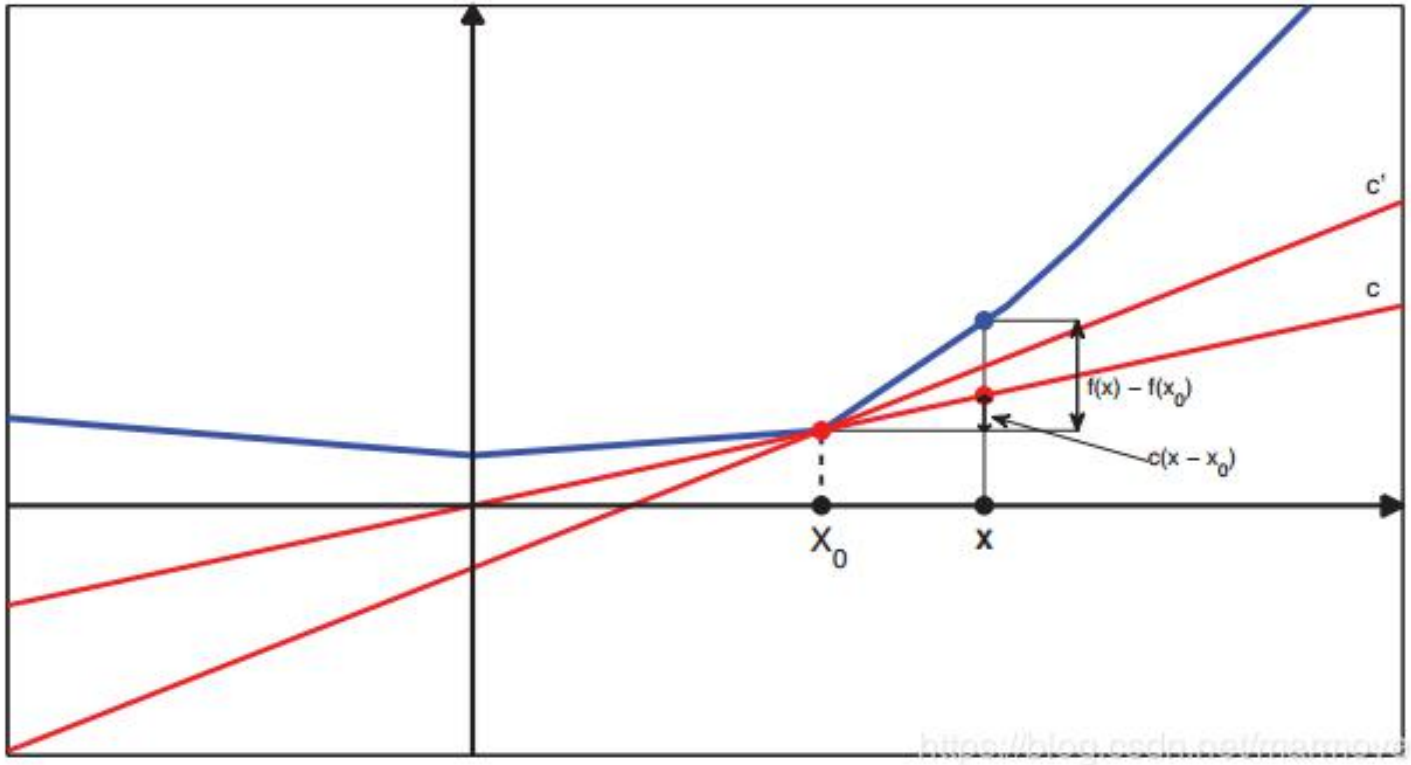


LASSO

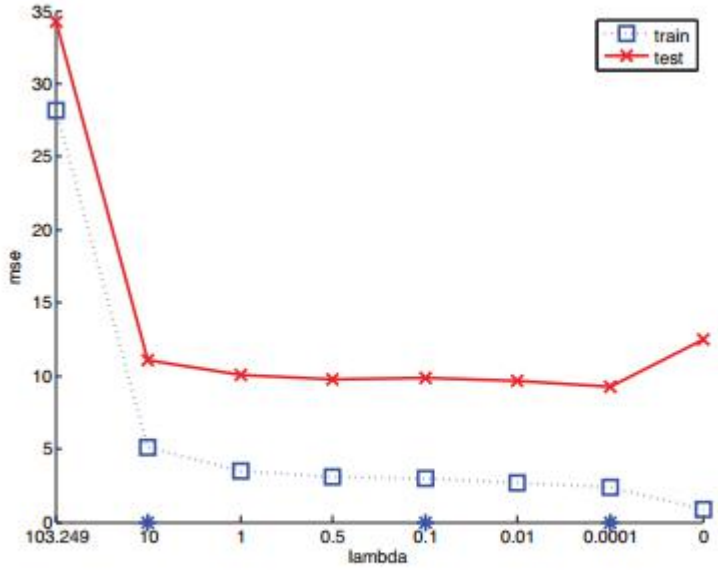



Machine Learning
A Probabilistic Perspective
Kevin P. Murphy

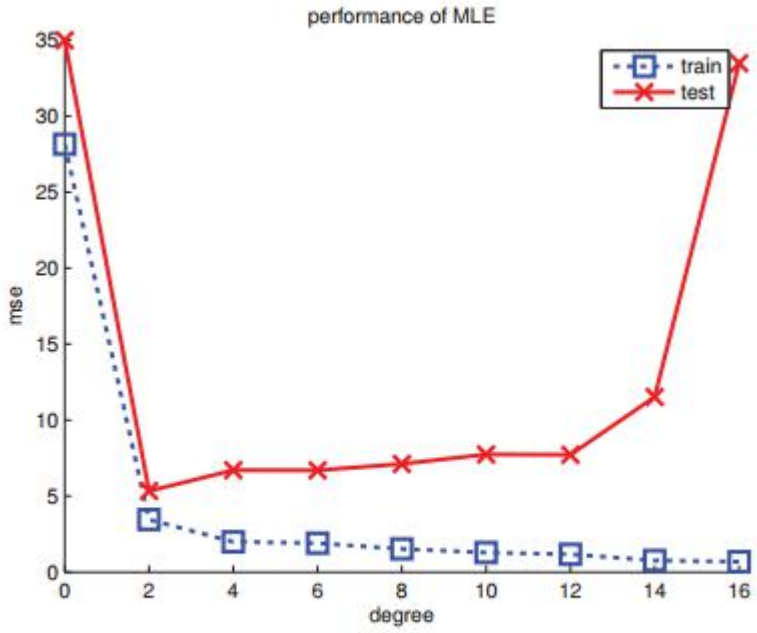
LASSO


c'
c
f(x) – f(x0)
c(x – x0)
X0
X

LASSO



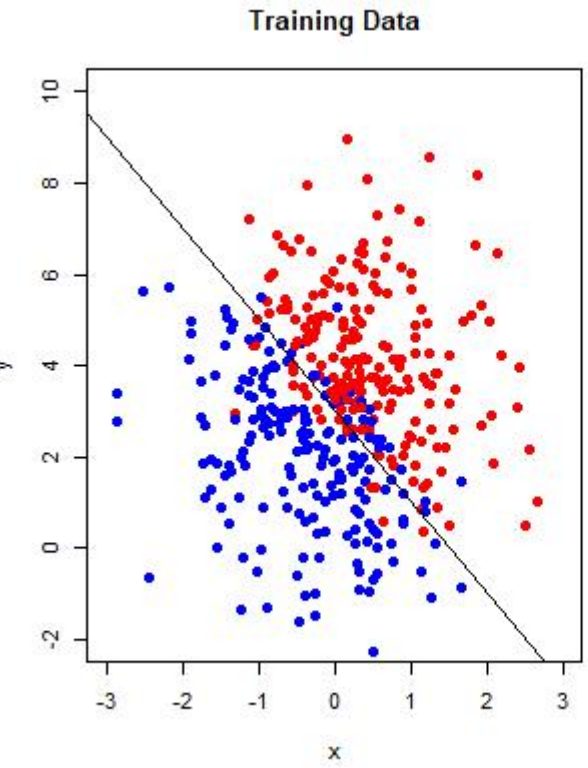
(a)

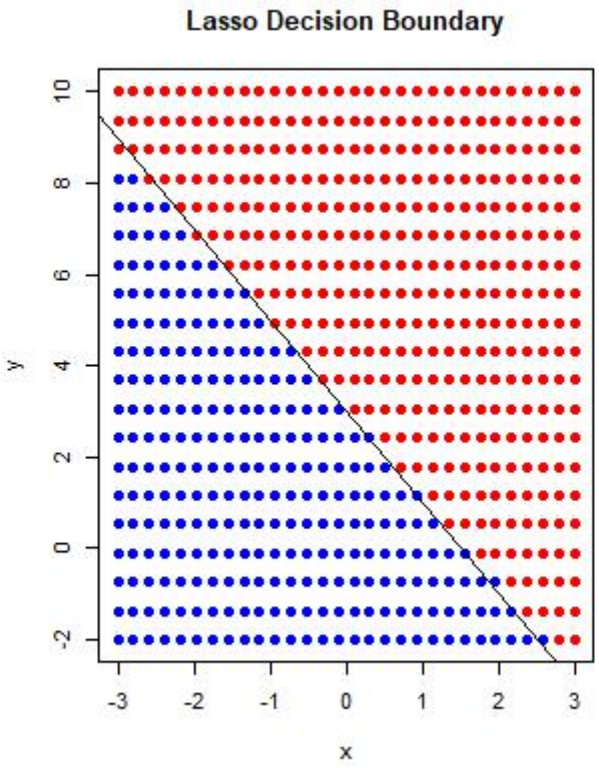


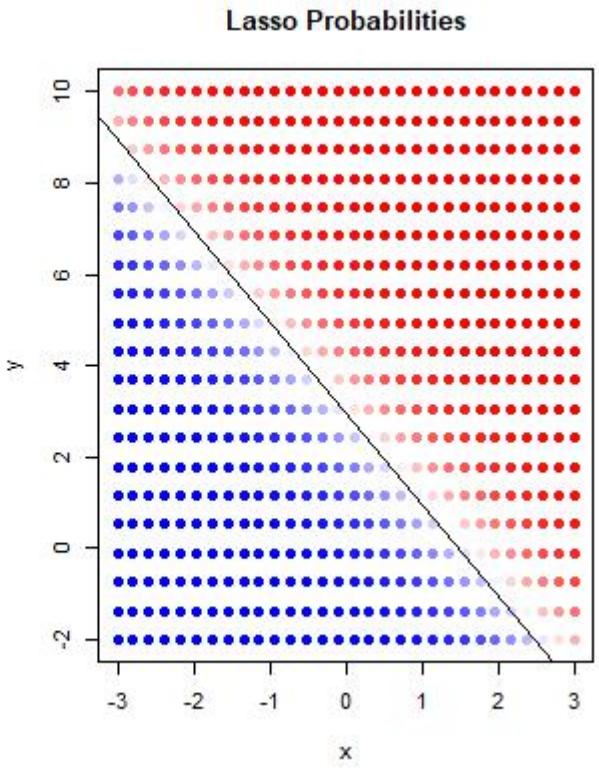
(b)

- 左边是lasso，MSE随着变化的情况
- 右边则是subset selection随着K的变化
- 这两个算法的性能是比较接近的

Lasso Model

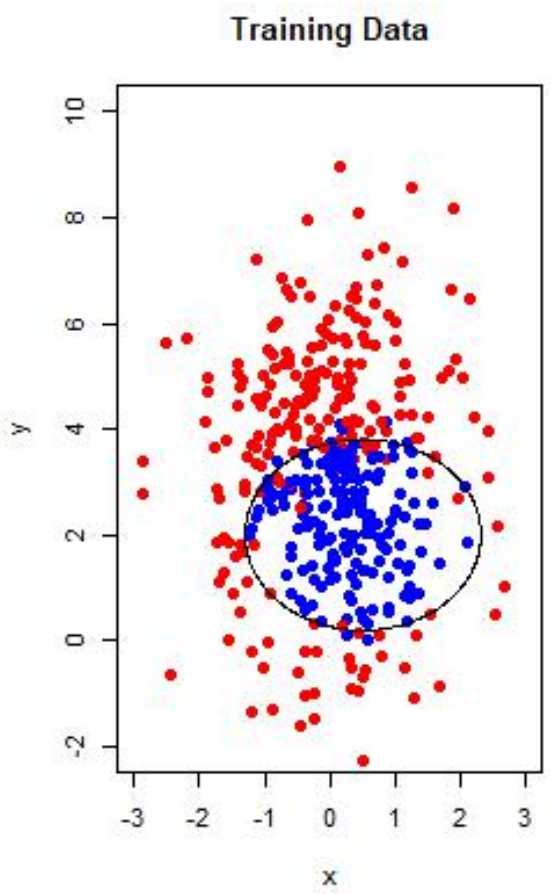


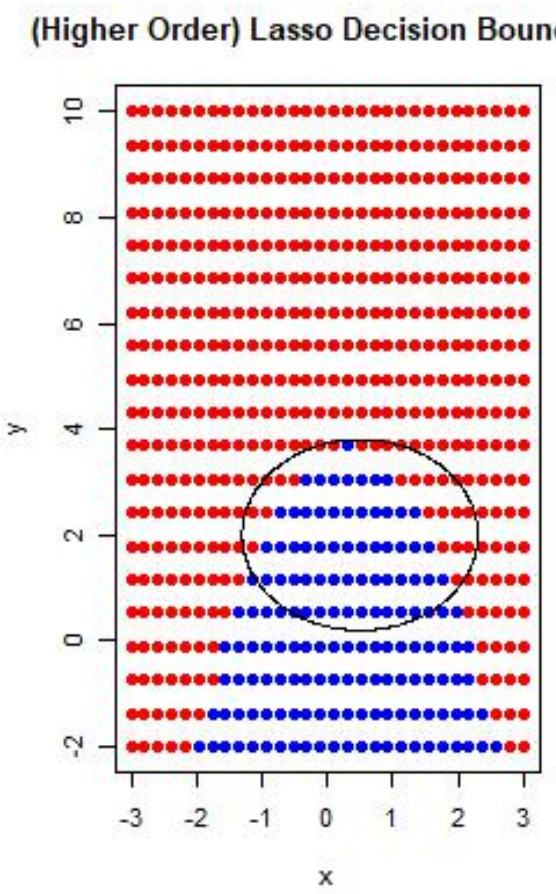


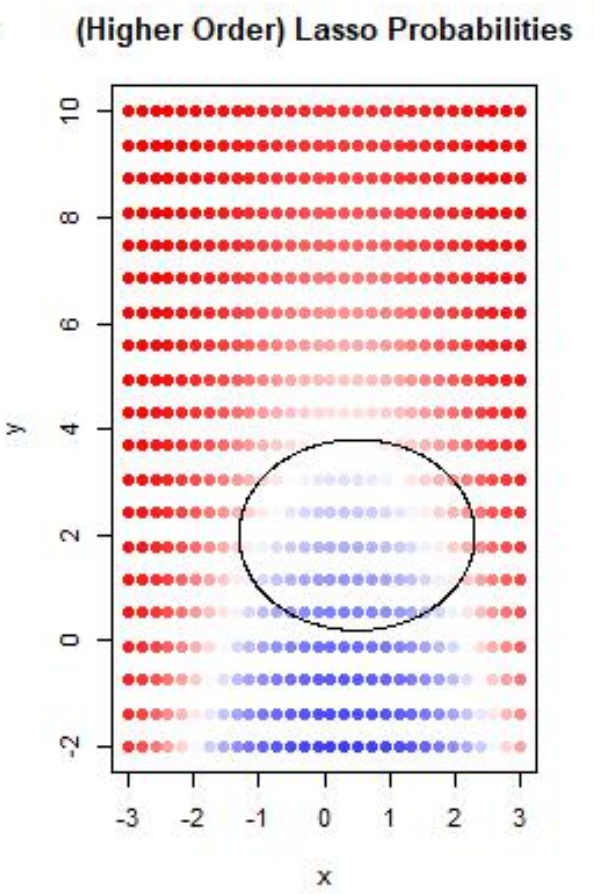


Reference: https://healthcare.ai/visual-tour-lasso-random-forest/

Lasso Model



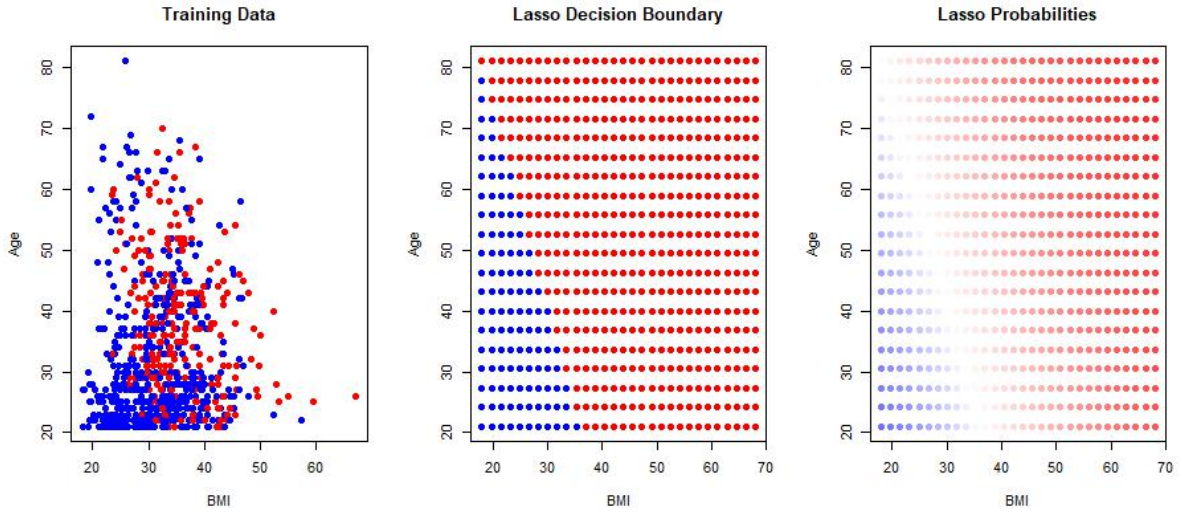




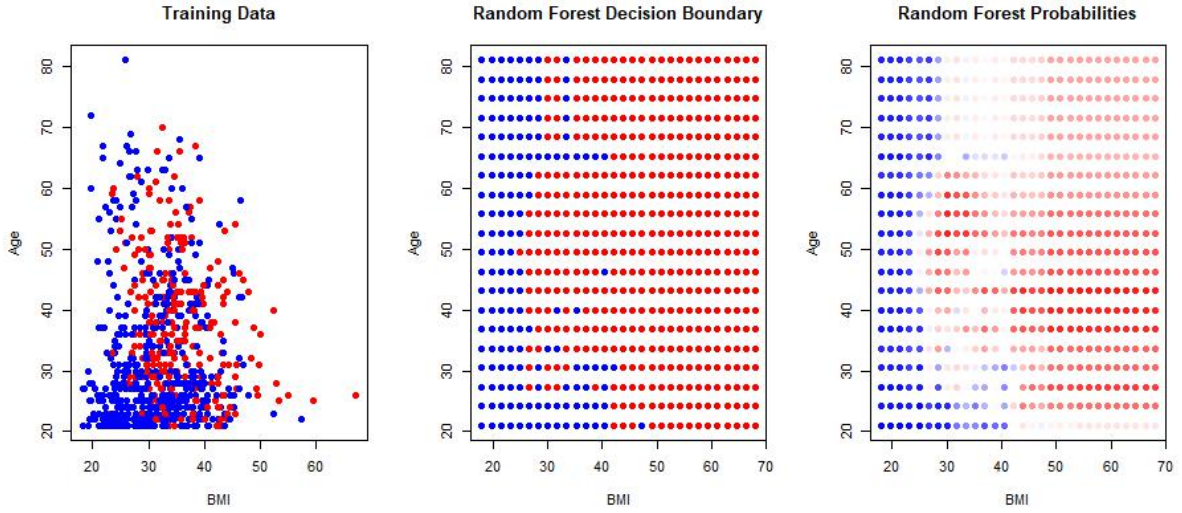
Reference: https://healthcare.ai/visual-tour-lasso-random-forest/

Lasso vs. Random forest

Lasso



Random forest



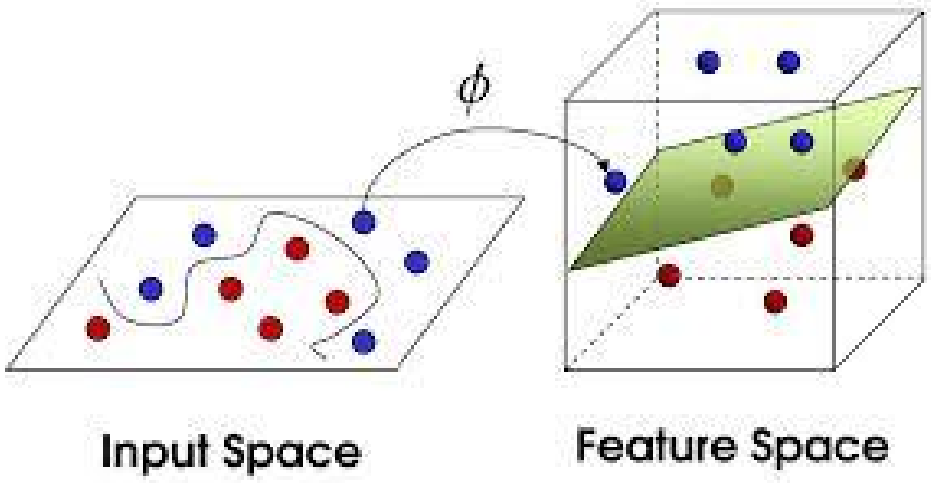
Reference: https://healthcare.ai/visual-tour-lasso-random-forest/

Kernel functions

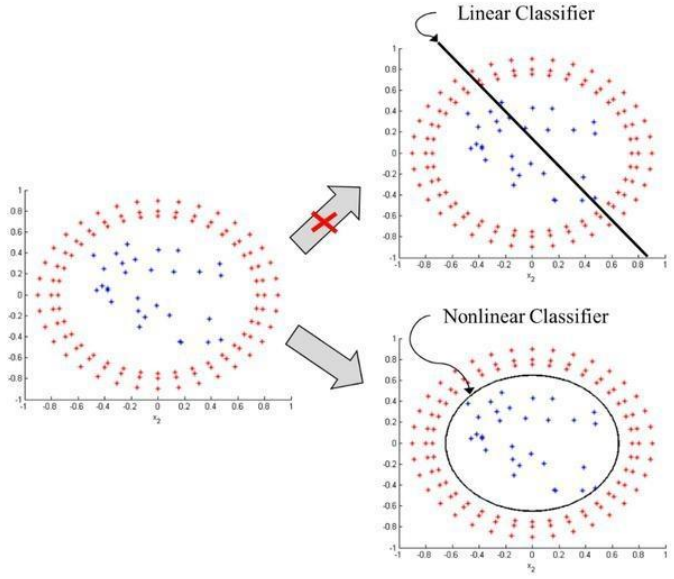
- 一般英文文献对Kernel有两种提法：
  - Kernel Function：核函数
  - Kernel Trick：一种运算技巧而已，不涉及什么高深莫测的东西

Kernel functions

- 通过某非线性变换，将输入空间映射到高维特征空间，本质还是对原有数据增加维度。
- 核函数和映射没有关系。**核函数只是用来计算映射到高维空间之后的内积的一种简便方法。**
- 核函数包括：线性核函数、多项式核函数、高斯核函数等，其中高斯核函数最常用，可以将数据映射到无穷维。



Kernel functions



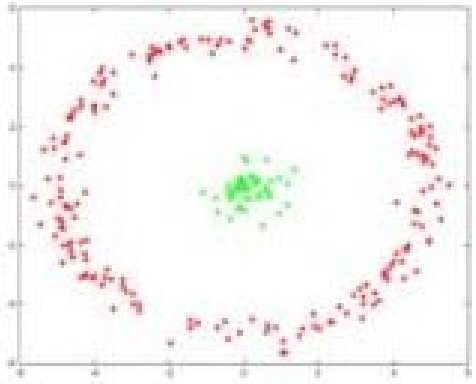


Figure 1: Original Data

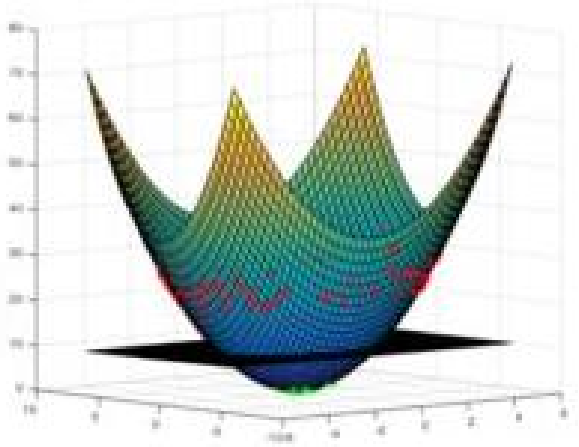


Figure 2: Data on feature space

Kernel functions

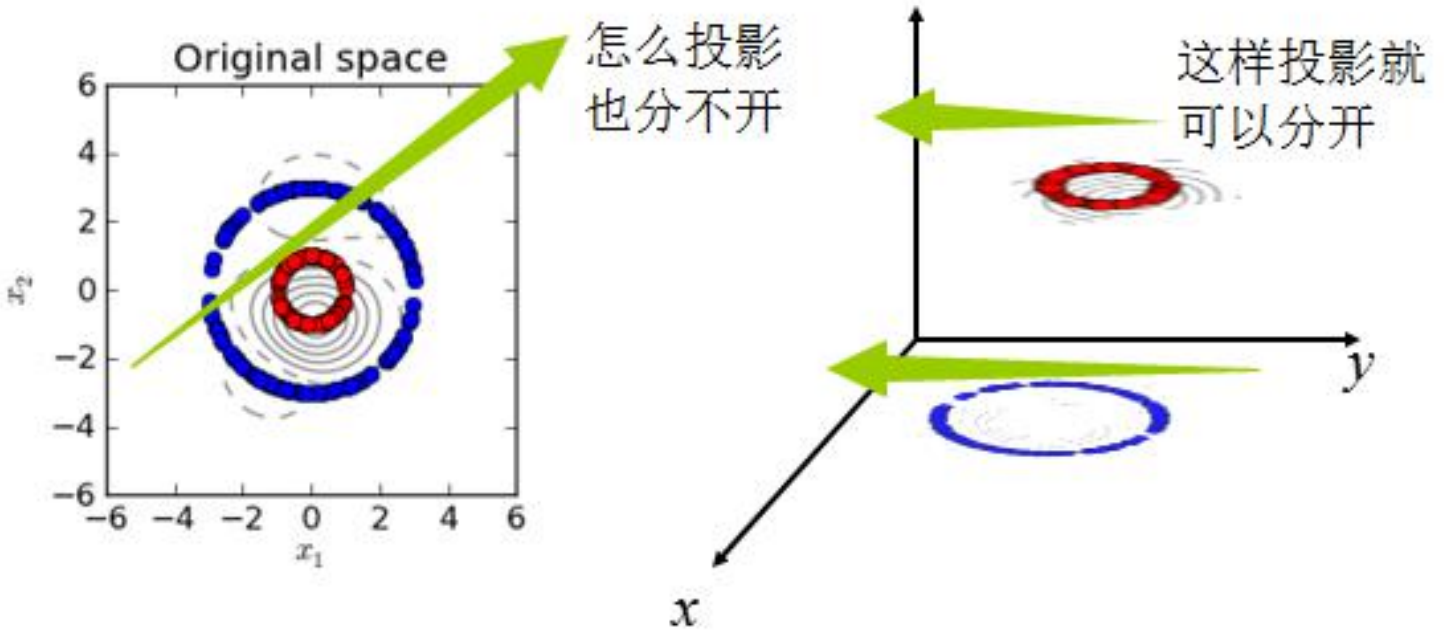
- 我们如果想进行原本就线性不可分的数据集进行分割
  - 选项一：容忍错误分类
  - 选项二：我们可以对Input Space做Feature Expansion，把数据集映射到高维中去，形成了Feature Space。我们几乎可以认为：原本在低维中线性不可分的数据集在足够高的维度中存在线性可分的超平面

- 核技巧(kernel trick)的作用：降低计算的复杂度，甚至把不可能的计算变为可能

Kernel functions

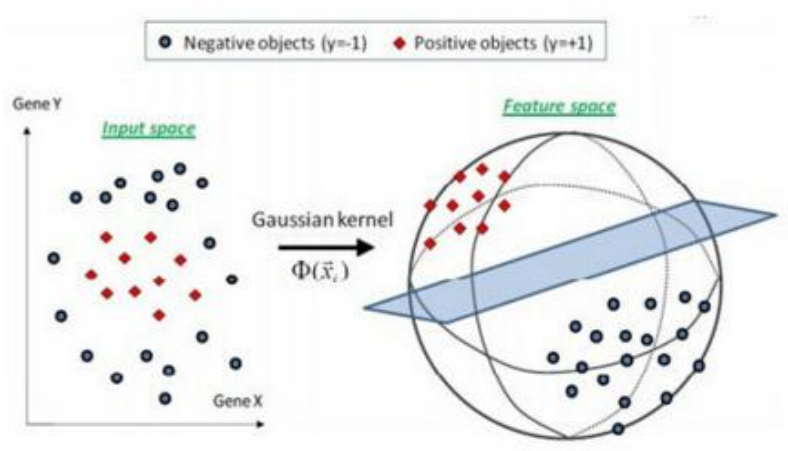
为什么PCA这样一个降维算法也用核函数呢？

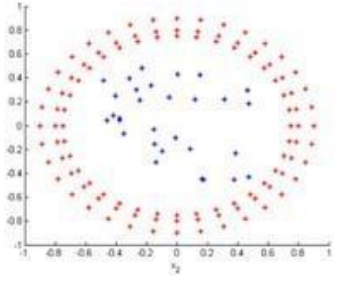
- 同样是降到1维，先通过Kernel映射到三3维，再投影到1维，就容易分离开
- 这就是Kernel在PCA降维中的应用



Kernel functions

- 通过使用非线性映射算法将低维输入空间线性不可分的样本转化为高维特征空间使其线性可分
- 从而使得高维特征空间采用线性算法对样本的非线性特征进行线性分析成为可能



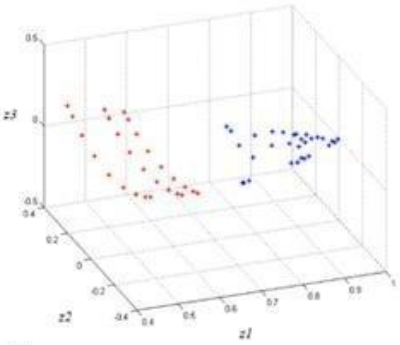


RBF kernel
(σ =1)

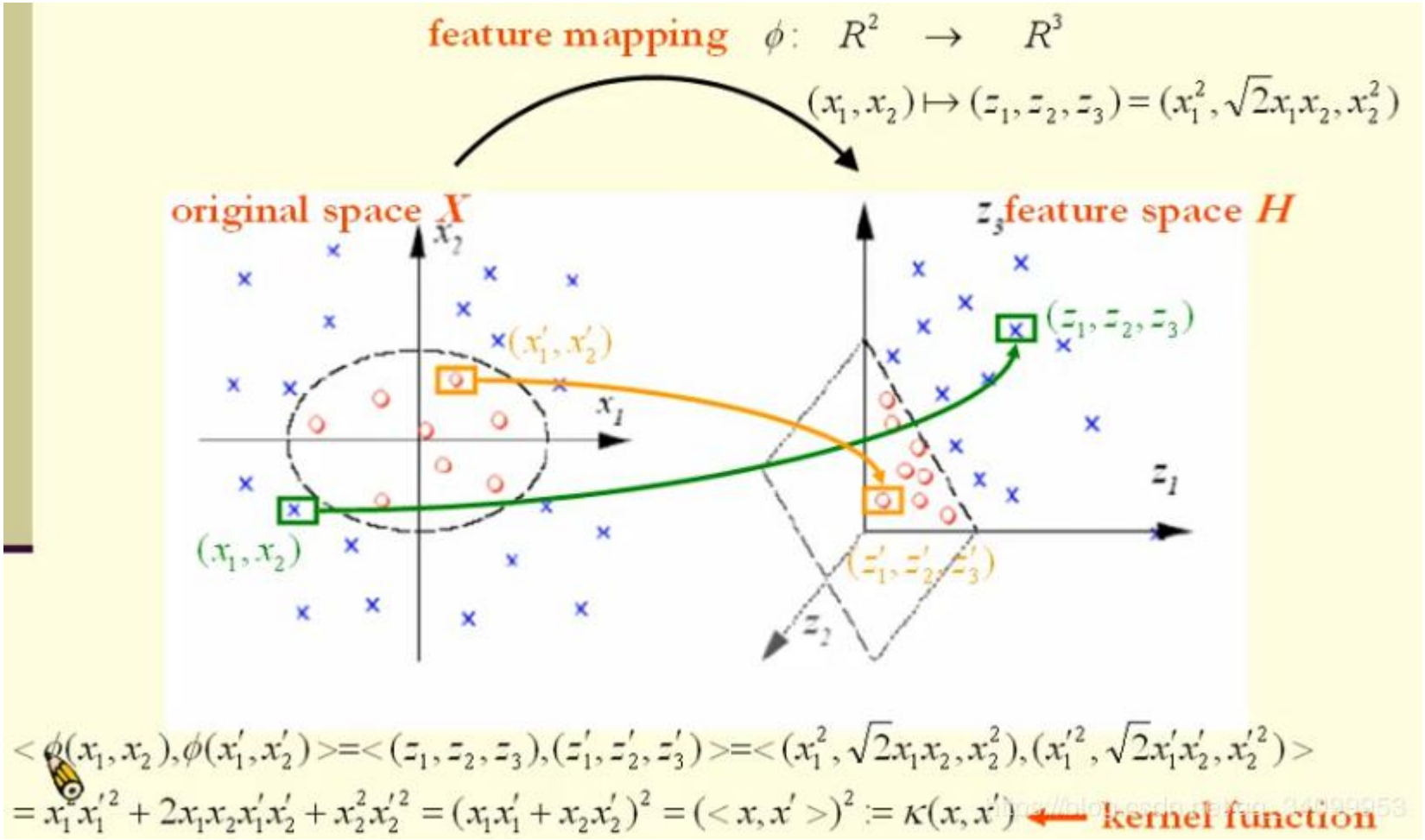
Map function: $\varphi(x)$

$$x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

$$\varphi(x) = \begin{bmatrix} z_1 \\ z_2 \\ z_3 \end{bmatrix} = \begin{bmatrix} e^{-x_1^2} e^{-x_2^2} \\ \frac{(2)^1}{1!} x_1 x_2 e^{-x_1^2} e^{-\sigma x} \\ \frac{(2)^2}{2!} x_1^2 x_2^2 e^{-x_1^2} e^{-x_2^2} \end{bmatrix}$$



Kernel functions



Kernel functions

- 有这么多种类的kernel，你要用什么kernel函数在你的特征上?你挑到kernel了，kernel参数怎么调整？

- 如果你有跑过SVM，你就知道这样会玩多久了。。。

Kernel functions

核函数具有以下性质：

（1）核函数的引入避免了“维数灾难”，大大减小了计算量。而输入空间的维数n对核函数矩阵无影响，因此，核函数方法可以有效处理高维输入

（2）无需知道非线性变换函数Φ的形式和参数

（3）核函数的形式和参数的变化会隐式地改变从输入空间到特征空间的映射，进而对特征空间的性质产生影响，最终改变各种核函数方法的性能

（4）核函数方法可以和不同的算法相结合，形成多种不同的基于核函数技术的方法，且这两部分的设计可以单独进行，并可以为不同的应用选择不同的核函数和算法

Kernel functions

以下是几种常用的核函数表示：

- 线性核（Linear Kernel）

$$k(x,y)=x^{T}y+c$$

- 多项式核（Polynomial Kernel）

$$k(x,y)=\left(ax^{T}y+c\right)^{d}$$

- 径向基核函数（Radial Basis Function）也叫高斯核（Gaussian Kernel）

$$k(x,y)=\exp\left(-\gamma\|x-y\|^{2}\right)$$

SVM

- 支持向量机（Support Vector Machine, SVM）
  - 基本模型是在特征空间上找到最佳的分离超平面使得训练集上正负样本间隔最大
  - SVM是用来解决二分类问题的有监督学习算法，在引入了核方法之后SVM也可以用来解决非线性问题

SVM

线性可分

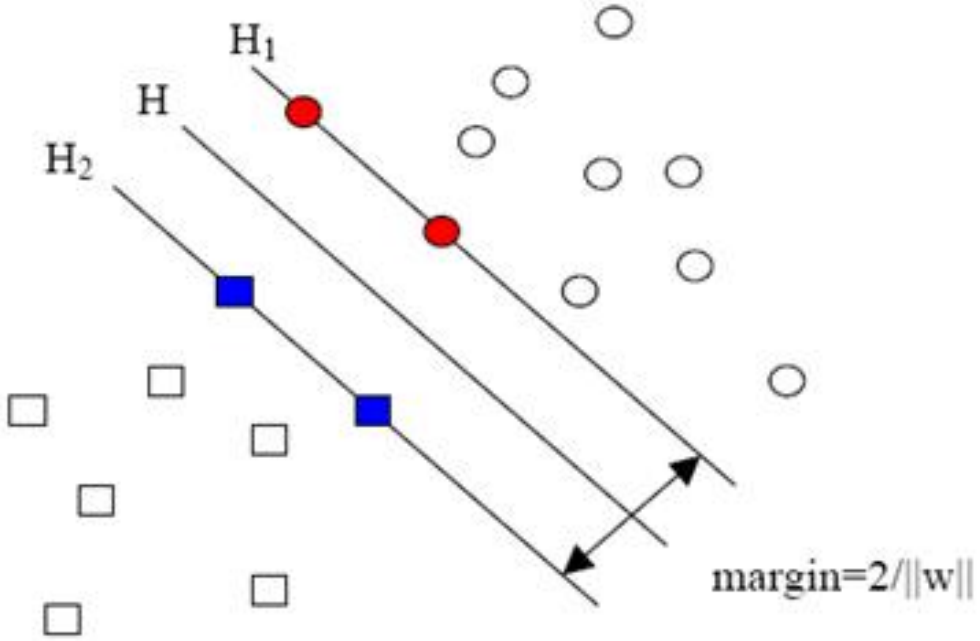
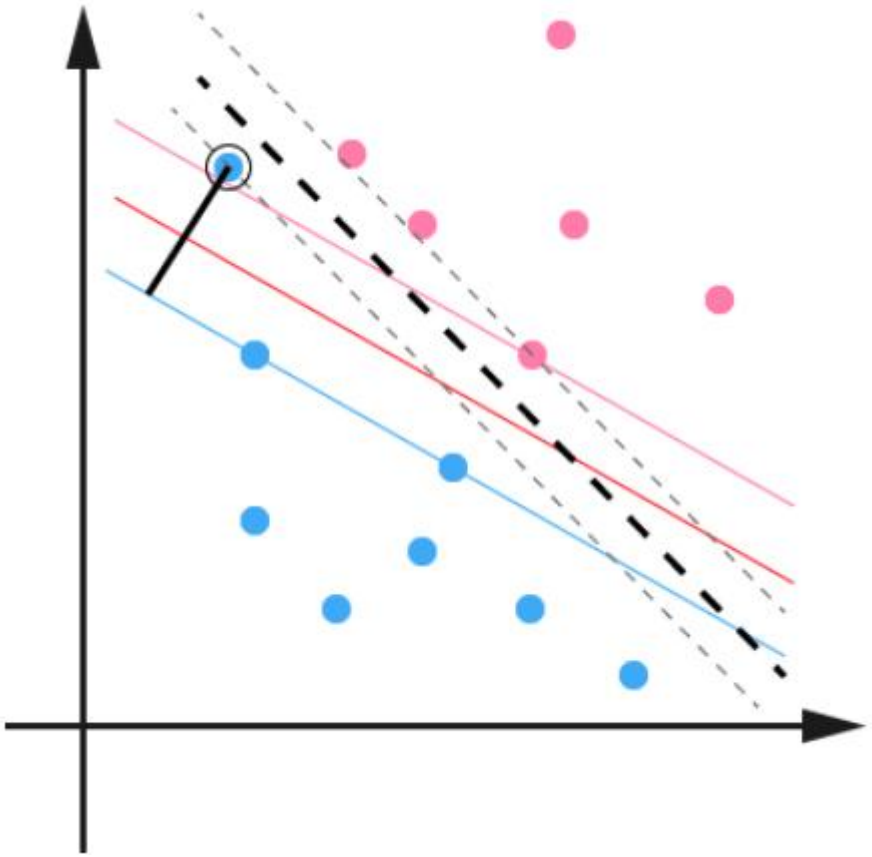
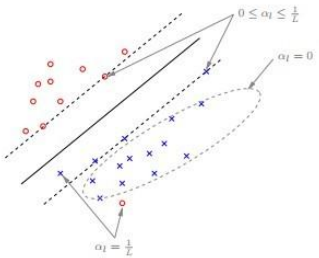


图2　线性可分情况下的最优分类线

SVM

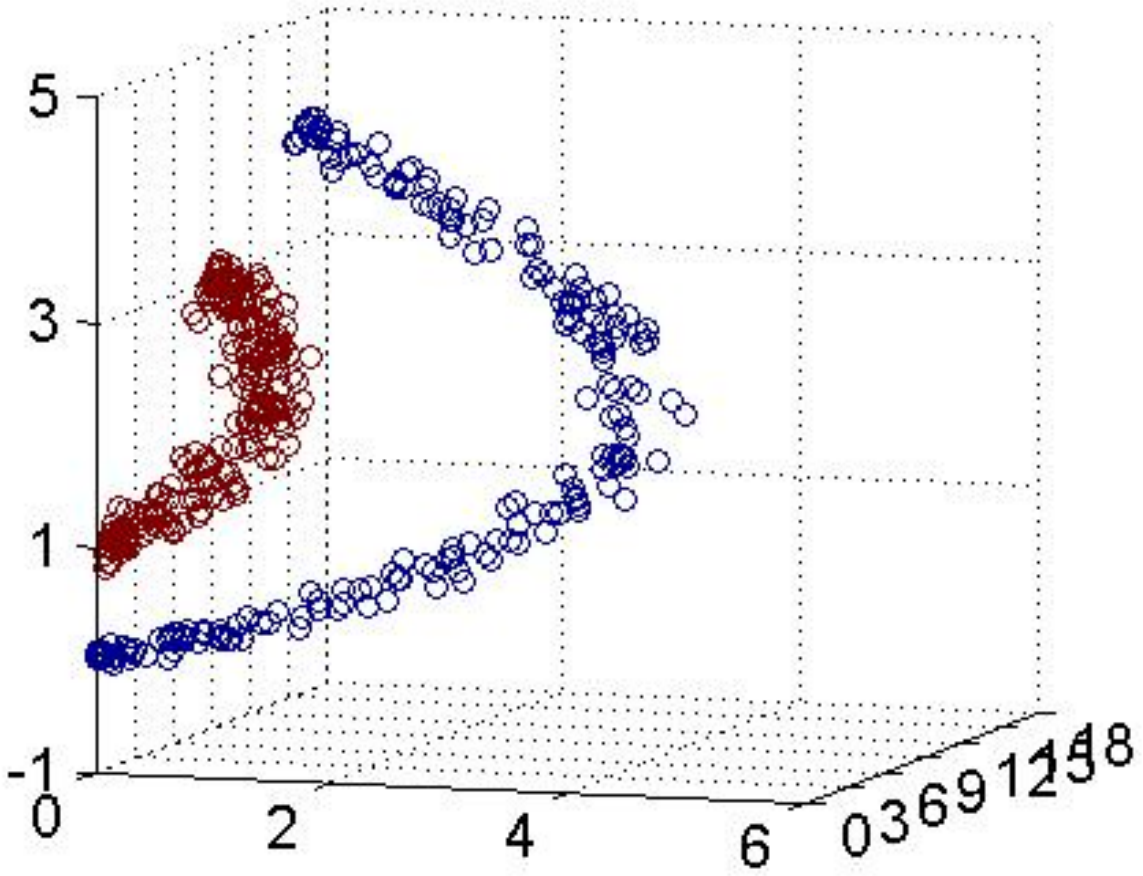
线性可分





SVM

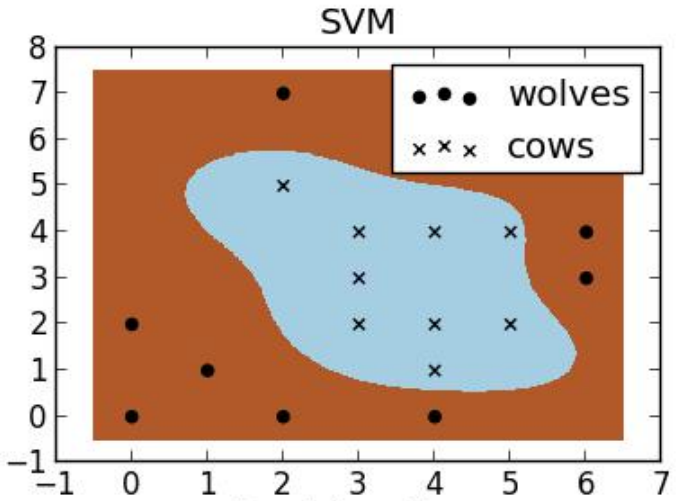
线性不可分（非线性数据，核函数）

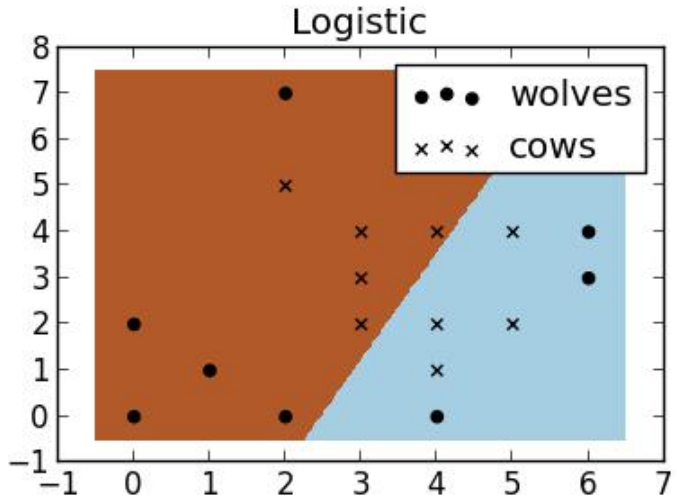


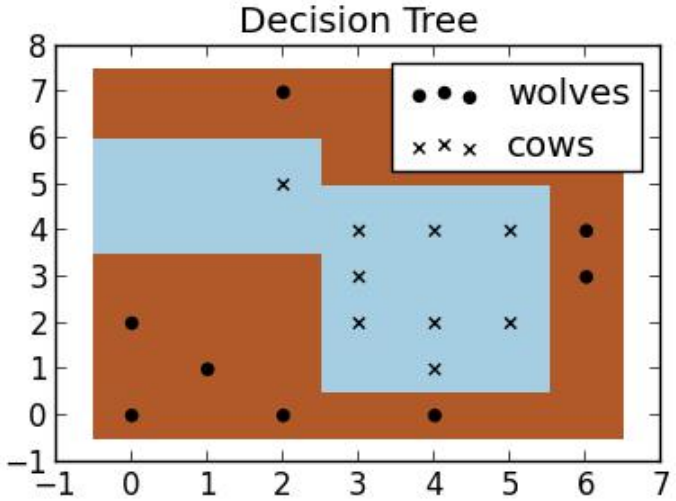
SVM + Kernel function

- 实际中，我们会经常遇到线性不可分的情况，此时，我们的常用做法是把样例特征映射到高维空间中去
- 但进一步，如果凡是遇到线性不可分的样例，一律映射到高维空间，那么这个维度大小是会高到可怕的。那咋办呢？
- 此时，用核函数！
  - 核函数的价值在于：它虽然也是将特征进行从低维到高维的转换，但核函数事先在低维上进行计算，而将实质上的分类效果表现在了高维上，也就如上文所说的避免了直接在高维空间中的复杂计算

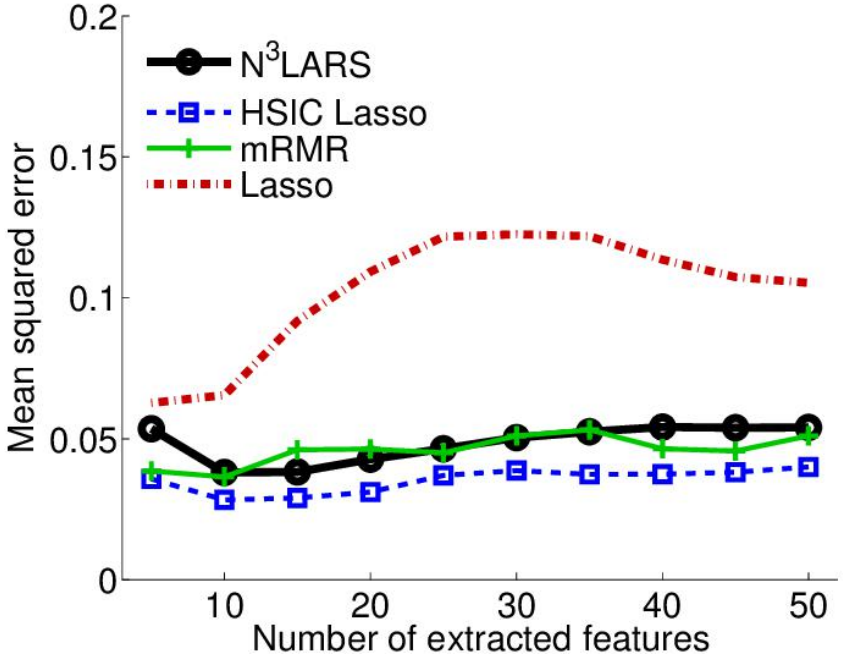
SVM

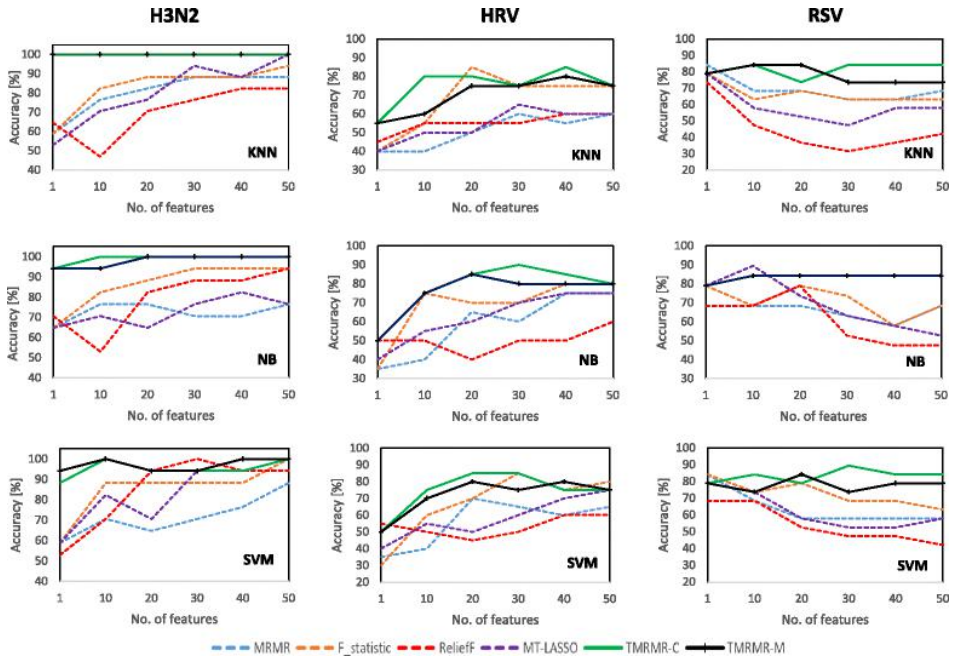

SVM
wolves
cows


Logistic
wolves
cows

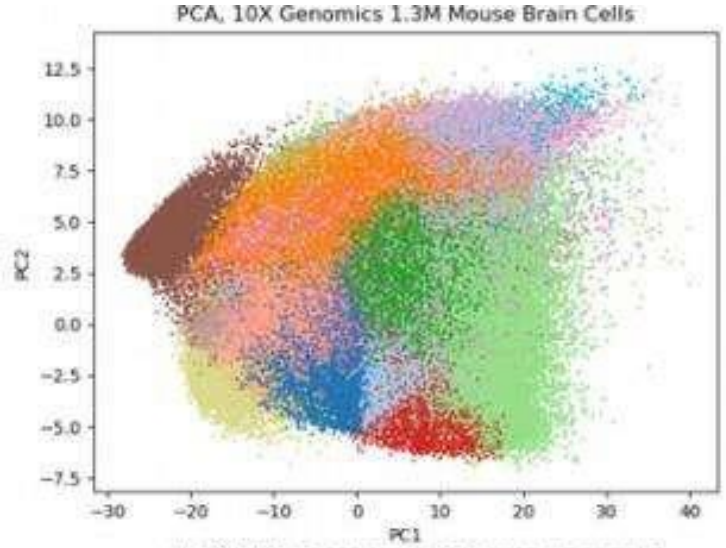

Decision Tree
wolves
cows

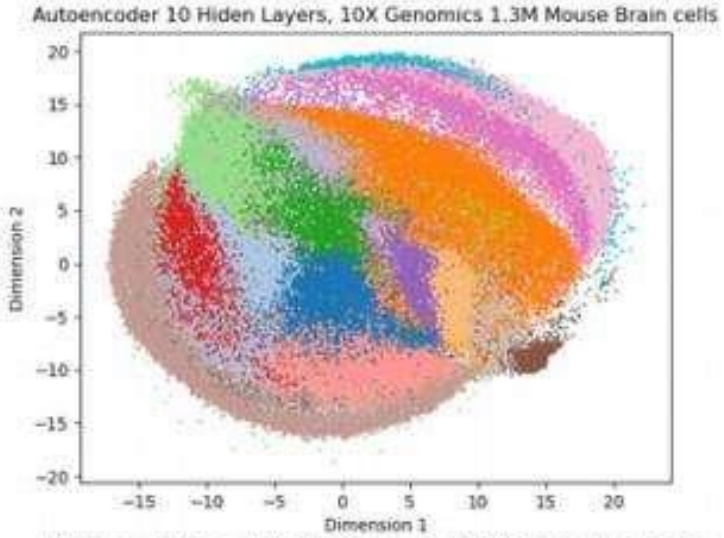
Other variable selection methods

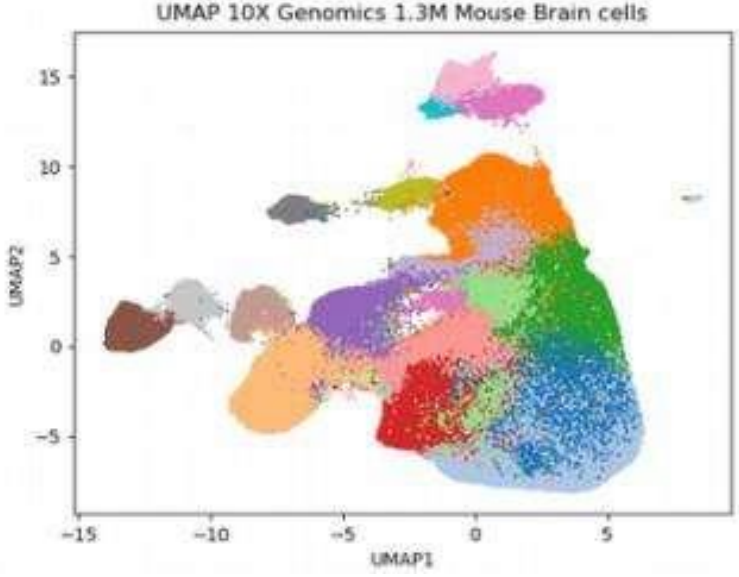


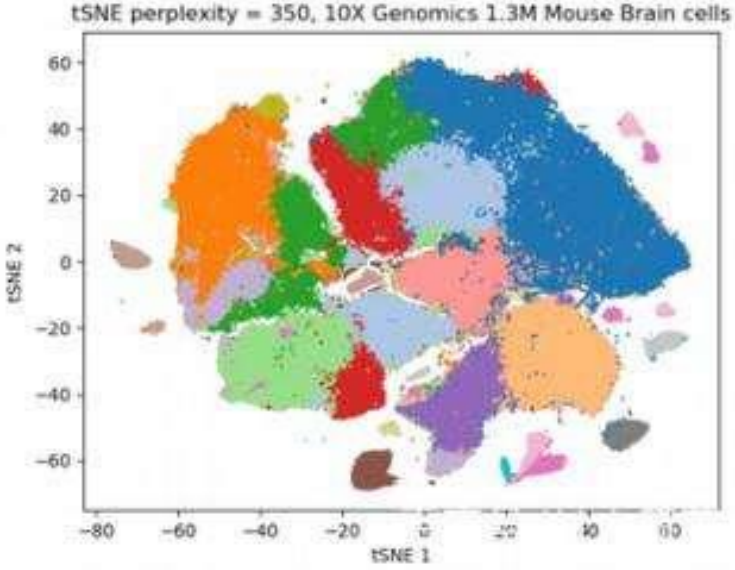


Other dimension reduction methods





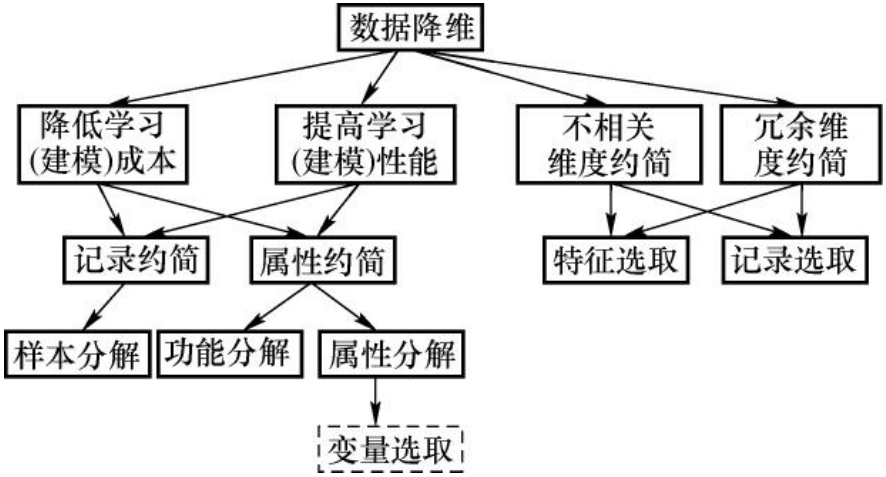




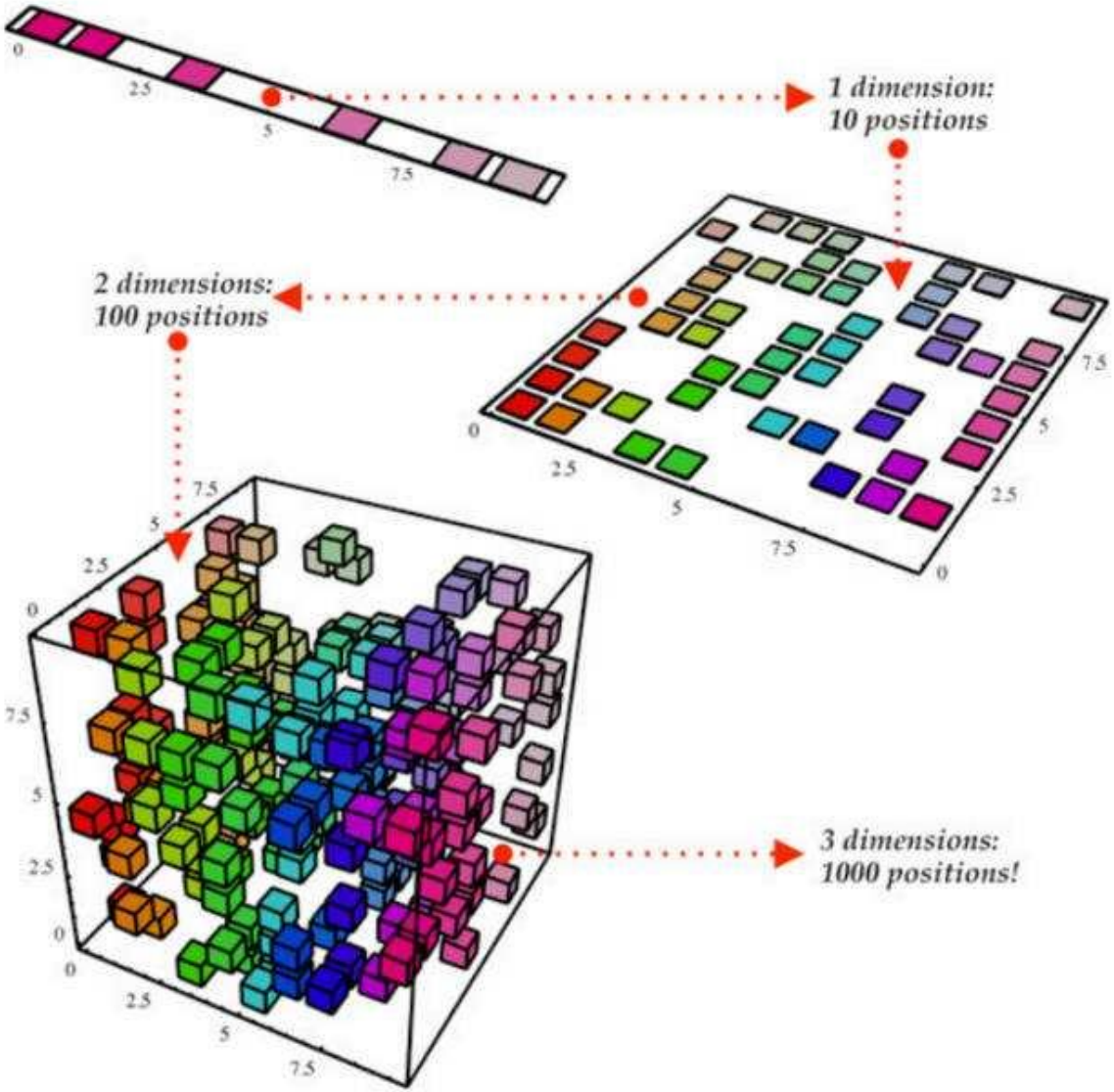
Recap (知识点总结)

- 降维的必要性：
  - 维度爆炸和分类可视化
  - 特征的冗余

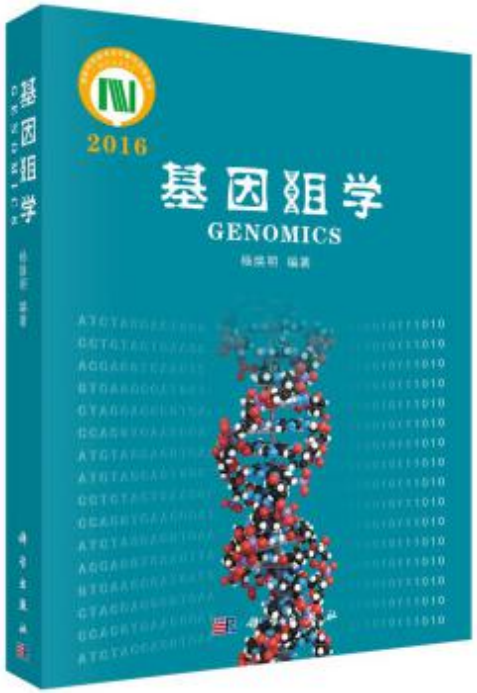
- 降维的方法：
  - 特征选取：mRMR
  - 线性降维：PCA，LDA，CCA
  - 非线性降维：MDS，tSNE
  - 其它特征选取和降维方法：LASSO
  - 核函数的原理和应用

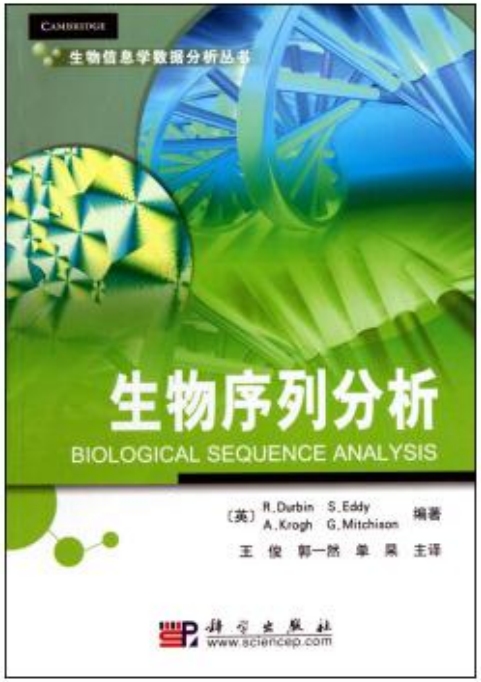


Recap (知识点总结)

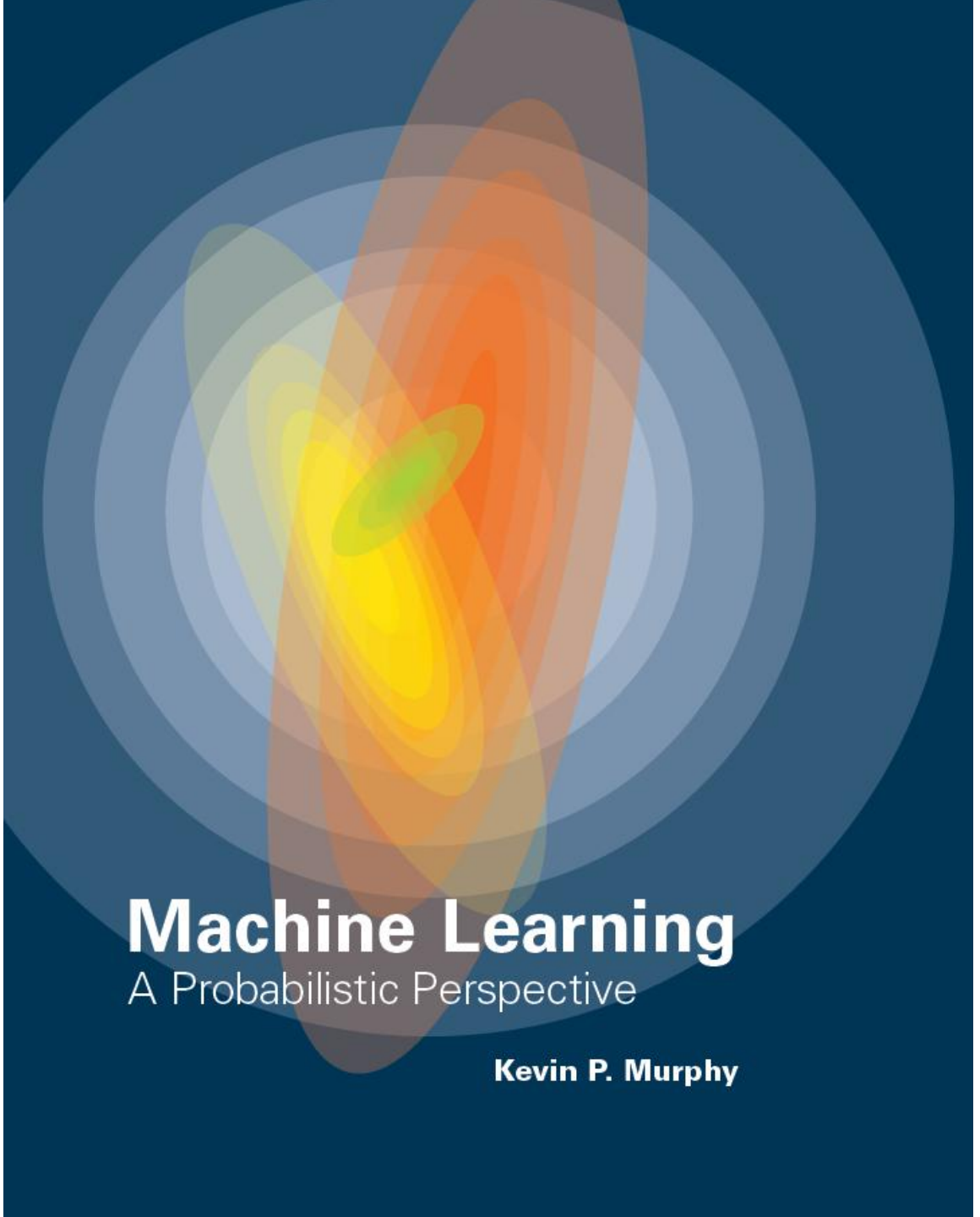

1 dimension:
10 positions
2 dimensions:
100 positions
3 dimensions:
1000 positions!

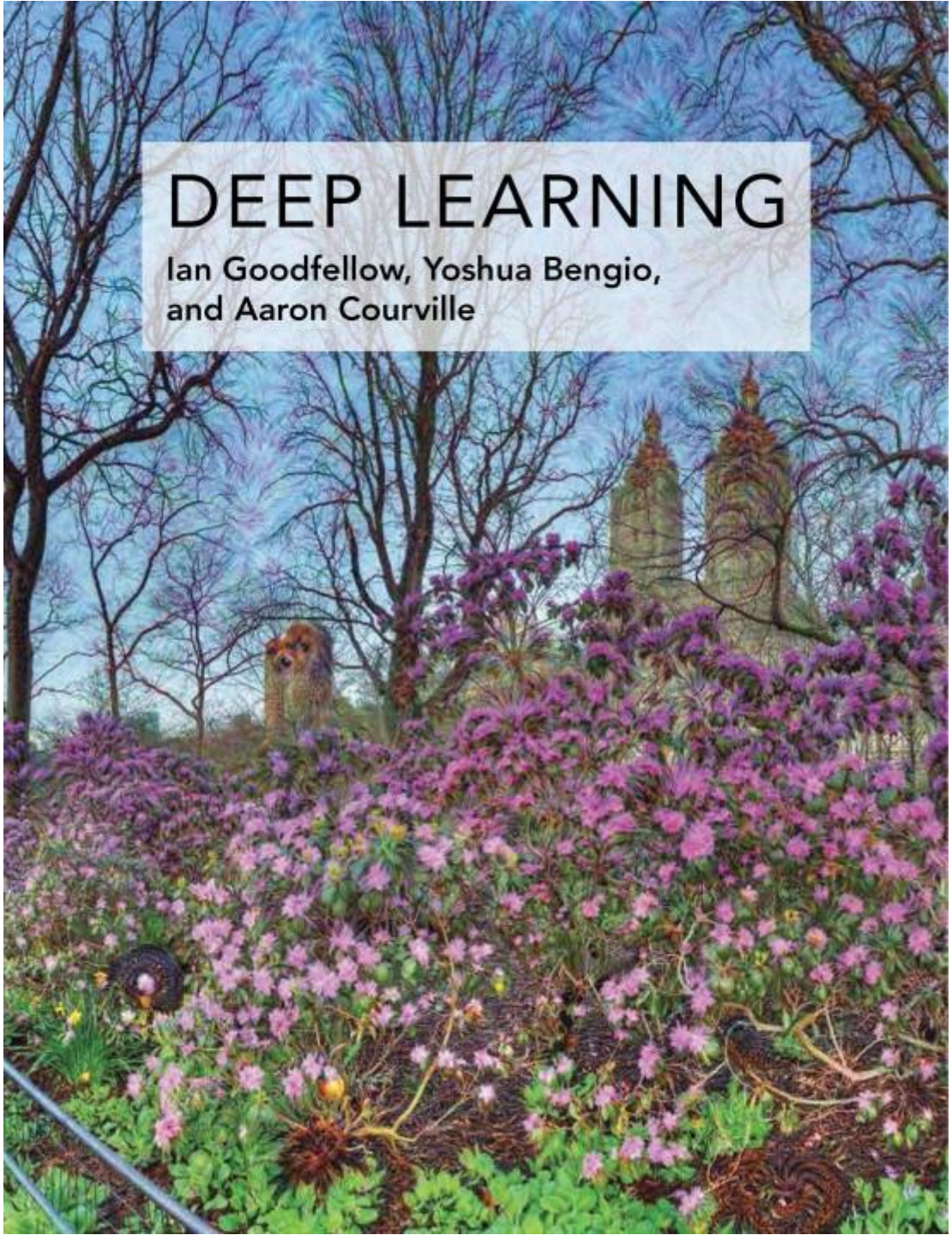
References


2016
基因组学
GENOMICS


CAMBRIDGE
生物信息学数据分析丛书
生物序列分析
BIOLOGICAL SEQUENCE ANALYSIS
R.Durbin S.Eddy
A.Krogh G.Mitchison
编著
王俊 郭一然 单昊 主译
www.sciencep.com


生物统计学
(第五版)


Machine Learning
A Probabilistic Perspective
Kevin P. Murphy


DEEP LEARNING
Ian Goodfellow, Yoshua Bengio,
and Aaron Courville

Slides credits

——生物信息学研究方法概述：北京大学生物信息中心

——生物统计学：卜东波@中国科学院计算技术研究所，邓明华@北京大学

——神经网络与深度学习: 邱锡鹏@复旦大学

——Introduction to Computational Biology and Bioinformatics: Xiaole Shirley Liu Lab@Havard University

——Combinatorial Methods in Computation Biology: Ken Sung Lab@NUS

——Deep Learning in the Life Sciences: MIT