

## 七、贝叶斯分类器

主讲教师：周志华

# 贝叶斯决策论 (Bayesian decision theory)

概率框架下实施决策的基本理论

给定  $N$  个类别，令  $\lambda_{ij}$  代表将第  $j$  类样本误分类为第  $i$  类所产生的损失，则基于后验概率将样本  $x$  分到第  $i$  类的条件风险为：

$$R(c_i | x) = \sum_{j=1}^N \lambda_{ij} P(c_j | x)$$

贝叶斯判定准则 (Bayes decision rule) :

$$h^*(x) = \arg \min_{c \in \mathcal{Y}} R(c | x)$$

- $h^*$  称为贝叶斯最优分类器 (Bayes optimal classifier)，其总体风险称为贝叶斯风险 (Bayes risk)
- 反映了学习性能的理论上限

# 判别式 vs. 生成式

$P(c | x)$  在现实中通常难以直接获得

从这个角度来看，机器学习所要实现的是基于有限的训练样本  
尽可能准确地估计出后验概率

两种基本策略：

判别式 (discriminative) 模型

思路：直接对  $P(c | x)$  建模

代表：

- 决策树
- BP 神经网络
- SVM

生成式 (generative) 模型

思路：先对联合概率分布  $P(x, c)$  建模，再由此获得  $P(c | x)$

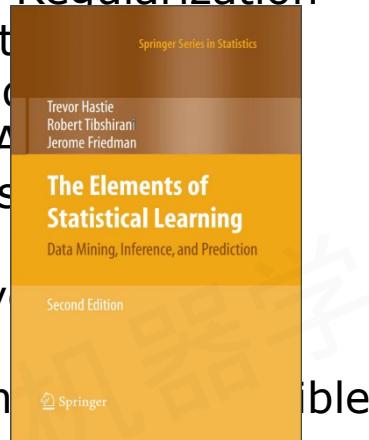
$$P(c | x) = \frac{P(x, c)}{P(x)}$$

代表：贝叶斯分类器

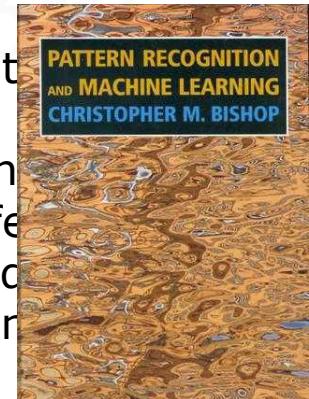
注意：贝叶斯分类器  $\neq$  贝叶斯学习  
(Bayesian learning)

# 机器学习中不同学派的视野差别极大，以两本名著的目录为例

1. Introduction
2. Overview of Supervised Learning
3. Linear Methods for Regression
4. Linear Methods for Classification
5. Basic Expansions and Regularization
6. Kernel Smoothing Methods
7. Model Assessment and Selection
8. Model Inference and Averaging
9. Additive Models, Trees, and Other Methods
10. Boosting and Additive Methods
11. Neural Networks
12. Support Vector Machines and Kernel Discriminants
13. Prototype Methods and Nearest-Neighbors
14. Unsupervised Learning
15. Random Forests
16. Ensemble Learning
17. Undirected Graphical Models
18. High-dimensional problems



1. Introduction
2. Probability Distributions
3. Linear Models for Regression
4. Linear Models for Classification
5. Neural Networks
6. Kernel Methods
7. Sparse Kernel Methods
8. Graphical Models
9. Mixture Models and Latent Variable Models
10. Approximate Inference
11. Sampling Methods
12. Continuous Latent Variable Models
13. Sequential Data Models
14. Combining Models



# 贝叶斯定理



$$P(c | x) = \frac{P(x, c)}{P(x)}$$

根据贝叶斯定理，有

Thomas Bayes  
(1701?-1761)

$$P(c | x) = \frac{P(c) P(x | c)}{P(x)}$$

先验概率 (prior)

样本空间中各类样本所占的比例，可通过各类样本出现的频率估计（大数定律）

样本相对于类标记的类条件概率 (class-conditional probability)，亦称 似然 (likelihood)

证据 (evidence)  
因子，与类别无关

主要困难在于估计似然

$$P(x | c)$$

# 极大似然估计

先假设某种概率分布形式，再基于训练样例对参数进行估计

假定  $P(\mathbf{x} | c)$  具有确定的概率分布形式，且被参数  $\theta_c$  唯一确定，则任务就是利用训练集  $D$  来估计参数  $\theta_c$

$\theta_c$  对于训练集  $D$  中第  $c$  类样本组成的集合  $D_c$  的似然(likelihood)为

$$P(D_c | \theta_c) = \prod_{\mathbf{x} \in D_c} P(\mathbf{x} | \theta_c)$$

连乘易造成下溢，因此通常使用对数似然 (log-likelihood)

$$LL(\theta_c) = \log P(D_c | \theta_c) = \sum_{\mathbf{x} \in D_c} \log P(\mathbf{x} | \theta_c)$$

于是， $\theta_c$  的极大似然估计为  $\hat{\theta}_c = \arg \max_{\theta_c} LL(\theta_c)$

估计结果的准确性严重依赖于所假设的概率分布形式是否符合潜在的真实分布

# 朴素贝叶斯分类器 (naïve Bayes classifier)

$$P(c \mid \mathbf{x}) = \frac{P(c) P(\mathbf{x} \mid c)}{P(\mathbf{x})}$$

主要障碍：所有属性上的联合概率  
难以从有限训练样本估计获得

组合爆炸；样本稀疏

基本思路：假定属性相互独立？

$$P(c \mid \mathbf{x}) = \frac{P(c) P(\mathbf{x} \mid c)}{P(\mathbf{x})} = \frac{P(c)}{P(\mathbf{x})} \prod_{i=1}^d P(x_i \mid c)$$

$d$  为属性数， $x_i$  为  $\mathbf{x}$  在第  $i$  个属性上的取值

$P(\mathbf{x})$  对所有类别相同，于是

$$h_{nb}(\mathbf{x}) = \arg \max_{c \in \mathcal{Y}} P(c) \prod_{i=1}^d P(x_i \mid c)$$

# 朴素贝叶斯分类器

□ 估计  $P(c)$ :  $P(c) = \frac{|D_c|}{|D|}$

□ 估计  $P(x|c)$ :

- 对离散属性, 令  $D_{c,x_i}$  表示  $D_c$  中在第  $i$  个属性上取值为  $x_i$  的样本组成的集合, 则

$$P(x_i | c) = \frac{|D_{c,x_i}|}{|D_c|}$$

- 对连续属性, 考虑概率密度函数, 假定  $p(x_i | c) \sim \mathcal{N}(\mu_{c,i}, \sigma_{c,i}^2)$

$$p(x_i | c) = \frac{1}{\sqrt{2\pi}\sigma_{c,i}} \exp\left(-\frac{(x_i - \mu_{c,i})^2}{2\sigma_{c,i}^2}\right)$$

# 一个例子

(青绿; 稍蜷; 浊响; 清晰)

- 好瓜 or 坏瓜 ?

编号	色泽	根蒂	敲声	纹理	好瓜
1	青绿	蜷缩	浊响	清晰	是
2	乌黑	蜷缩	沉闷	清晰	是
3	乌黑	蜷缩	浊响	清晰	是
4	青绿	蜷缩	沉闷	清晰	是
5	浅白	蜷缩	浊响	清晰	是
6	青绿	稍蜷	浊响	清晰	是
7	乌黑	稍蜷	浊响	稍糊	是
8	乌黑	稍蜷	浊响	清晰	是
9	乌黑	稍蜷	沉闷	稍糊	否
10	青绿	硬挺	清脆	清晰	否
11	浅白	硬挺	清脆	模糊	否
12	浅白	蜷缩	浊响	模糊	否
13	青绿	稍蜷	浊响	稍糊	否
14	浅白	稍蜷	沉闷	稍糊	否
15	乌黑	稍蜷	浊响	清晰	否
16	浅白	蜷缩	浊响	模糊	否
17	青绿	蜷缩	沉闷	稍糊	否

$$P(\text{青绿}|\text{好瓜}) = 3/8 \quad P(\text{青绿}|\text{坏瓜}) = 3/9$$

$$P(\text{稍蜷}|\text{好瓜}) = 3/8 \quad P(\text{稍蜷}|\text{坏瓜}) = 4/9$$

$$P(\text{浊响}|\text{好瓜}) = 6/8 \quad P(\text{浊响}|\text{坏瓜}) = 4/9$$

$$P(\text{清晰}|\text{好瓜}) = 7/8 \quad P(\text{清晰}|\text{坏瓜}) = 2/9$$

$$P(\text{好瓜}=\text{yes}) = 8/17 \quad P(\text{好瓜}=\text{no}) = 9/17$$

$$P(\text{青绿}|\text{好瓜}) \quad P(\text{稍蜷}|\text{好瓜}) \quad P(\text{浊响}|\text{好瓜})$$

$$P(\text{清晰}|\text{好瓜}) \quad P(\text{好瓜}=\text{yes}) = 3/8 \times 3/8 \\ \times 6/8 \times 7/8 \times 8/17$$

$$P(\text{青绿}|\text{坏瓜}) \quad P(\text{稍蜷}|\text{坏瓜}) \quad P(\text{浊响}|\text{坏瓜})$$

$$P(\text{清晰}|\text{坏瓜}) \quad P(\text{好瓜}=\text{no}) = 3/9 \times 4/9 \times \\ 4/9 \times 2/9 \times 9/17$$

好瓜 !

# 拉普拉斯修正 (Laplacian correction)

若某个属性值在训练集中没有与某个类同时出现过，则直接计算会出现问题，因为概率连乘将“抹去”其他属性提供的信息

例如，若训练集中未出现“敲声=清脆”的好瓜，  
则模型在遇到“敲声=清脆”的测试样本时 .....

令  $N$  表示训练集  $D$  中可能的类别数， $N_i$  表示第  $i$  个属性可能的取值数

$$\hat{P}(c) = \frac{|D_c| + 1}{|D| + N}, \quad \hat{P}(x_i | c) = \frac{|D_{c,x_i}| + 1}{|D_c| + N_i}$$

假设了属性值与类别的均匀分布，这是额外引入的 **bias**

# 朴素贝叶斯分类器的使用

---

- 若对预测速度要求高
  - 预计算所有概率估值，使用时“查表”
- 若数据更替频繁
  - 不进行任何训练，收到预测请求时再估值  
(懒惰学习, lazy learning)
- 若数据不断增加
  - 基于现有估值，对新样本涉及的概率估值进行修正  
(增量学习, incremental learning)

# 半朴素贝叶斯分类器

朴素贝叶斯分类器的“属性独立性假设”在现实中往往难以成立

## 半朴素贝叶斯分类器 (semi-naïve Bayes classifier)

基本思路：适当考虑一部分属性间的相互依赖信息

最常用策略：**独依赖估计** (One-Dependent Estimator, ODE)

假设每个属性在类别之外最多仅依赖一个其他属性

$$P(c \mid \mathbf{x}) \propto P(c) \prod_{i=1}^d P(x_i \mid c, pa_i)$$

*x<sub>i</sub>* 的“父属性”

关键是如何确定父属性

# 两种常见方法

## □ SPODE (Super-Parent ODE) :

假设所有属性都依赖于同一属性，称为“超父”(Super-Parent)，然后通过交叉验证等模型选择方法来确定超父属性

## □ TAN (Tree Augmented naïve Bayes) :

以属性间的条件“互信息”(mutual information)为边的权重，构建完全图，再利用最大带权生成树算法，仅保留强相关属性间的依赖性

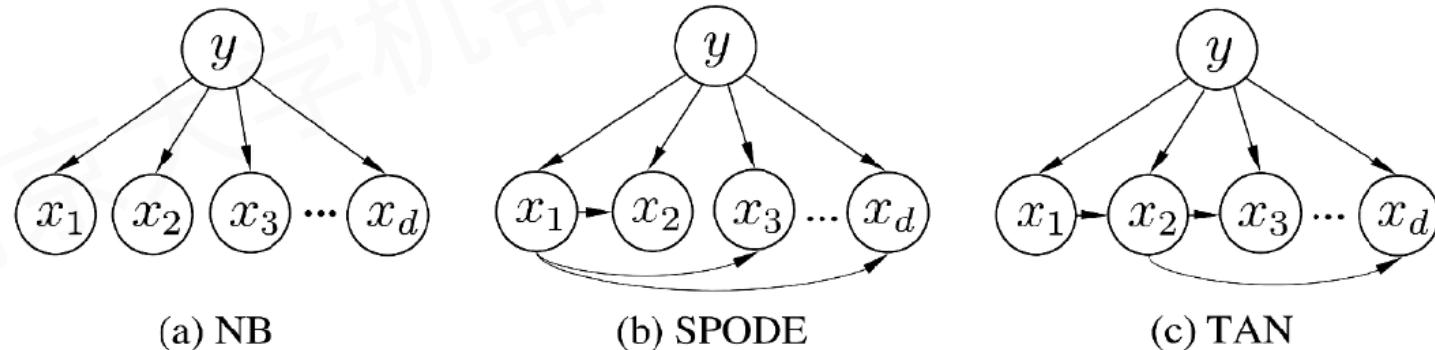


图 7.1 朴素贝叶斯与两种半朴素贝叶斯分类器所考虑的属性依赖关系

# AODE (Averaged One-Dependent Estimator)



- 尝试将每个属性作为超父构建 SPODE
- 将拥有足够训练数据支撑的 SPODE 集成起来作为最终结果

$$P(c \mid \mathbf{x}) \propto \sum_{\substack{i=1 \\ |D_{x_i}| \geq m'}}^d P(c, x_i) \prod_{j=1}^d P(x_j \mid c, x_i)$$

其中  $D_{x_i}$  是在第  $i$  个属性上取值为  $x_i$  的样本的集合， $m'$  为阈值常数

$$\hat{P}(c, x_i) = \frac{|D_{c,x_i}| + 1}{|D| + N \times N_i}, \quad \hat{P}(x_j \mid c, x_i) = \frac{|D_{c,x_i,x_j}| + 1}{|D_{c,x_i}| + N_j}$$

$N$  为  $D$  中可能的类别数， $N_i$  为第  $i$  个属性上可能的取值数

$D_{c,x_i,x_j}$  表示类别为  $c$  且在第  $i$  和第  $j$  个属性上取值分别为  $x_i$  和  $x_j$  的样本集合

Geoff Webb  
澳大利亚  
Monash大学

# 高阶依赖

---

能否通过考虑属性间的高阶依赖来进一步提升泛化性能？

例如最简单的做法：ODE → kDE

将父属性  $pa_i$  替换为包含  $k$  个属性的集合  $\mathbf{pa}_i$

明显障碍：随着  $k$  的增加，估计  $P(x_i | y, \mathbf{pa}_i)$  所需的样本数  
将以指数级增加

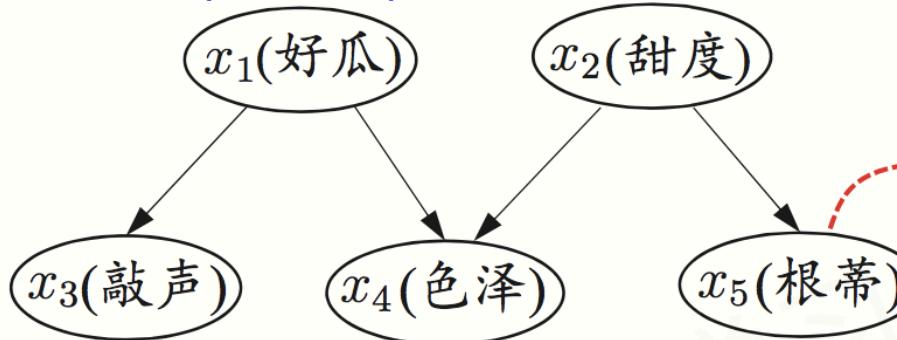
- 训练样本非常充分 → 性能可能提升
- 有限训练样本 → 高阶联合概率估计困难

考虑属性间的高阶依赖，需要其他办法

# 贝叶斯网 (Bayesian network; Bayes network)

亦称“信念网”(brief network)

有向无环图(DAG,  
Directed Acyclic Graph)



条件概率表(CPT,  
Conditional Probability Table)

		根蒂	
		硬挺	蜷缩
甜度	高	0.1	0.9
	低	0.7	0.3

贝叶斯网  $B = \langle G, \Theta \rangle$

结构      参数

1985年 J. Pearl 命名为贝叶斯网，  
为了强调：

- 输入信息的主观本质
- 对贝叶斯条件的依赖性
- 因果与证据推理的区别

概率图模型(Probabilistic graphical model) → 第14章

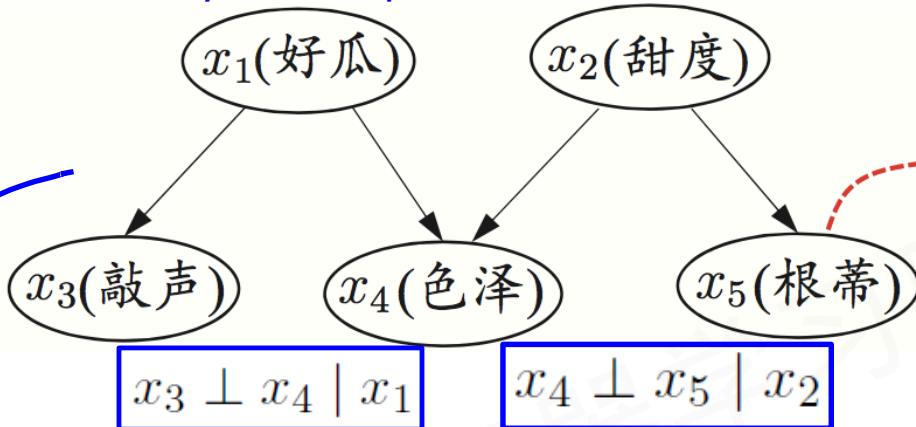
- 有向图模型 → 贝叶斯网
- 无向图模型 → 马尔可夫网



Judea Pearl  
(1936 - )  
2011 图灵奖

# 贝叶斯网 (Bayesian network)

有向无环图 (DAG,  
Directed Acyclic Graph)



条件概率表 (CPT,  
Conditional Probability Table)

		根蒂	
		硬挺	蜷缩
甜度	高	0.1	0.9
	低	0.7	0.3

给定父结点集，贝叶斯网假设每个属性与其非后裔属性独立

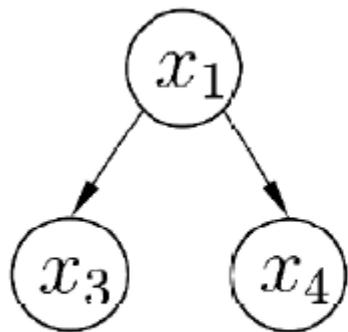
$$P_B(x_1, x_2, \dots, x_d) = \prod_{i=1}^d P_B(x_i \mid \pi_i)$$

父结点集

$$= \prod_{i=1}^d \theta_{x_i \mid \pi_i}$$

$$P(x_1, x_2, x_3, x_4, x_5) = P(x_1)P(x_2)P(x_3 \mid x_1)P(x_4 \mid x_1, x_2)P(x_5 \mid x_2)$$

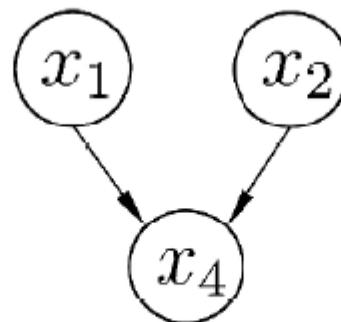
# 三变量间的典型依赖关系



同父结构

条件独立性

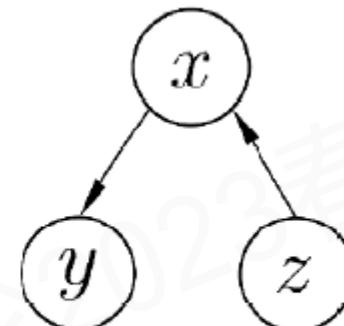
$$x_3 \perp\!\!\! \perp x_4 \mid x_1$$



V型结构

边际独立性

$$x_1 \perp\!\!\! \perp x_2$$



顺序结构

条件独立性

$$y \perp\!\!\! \perp z \mid x$$

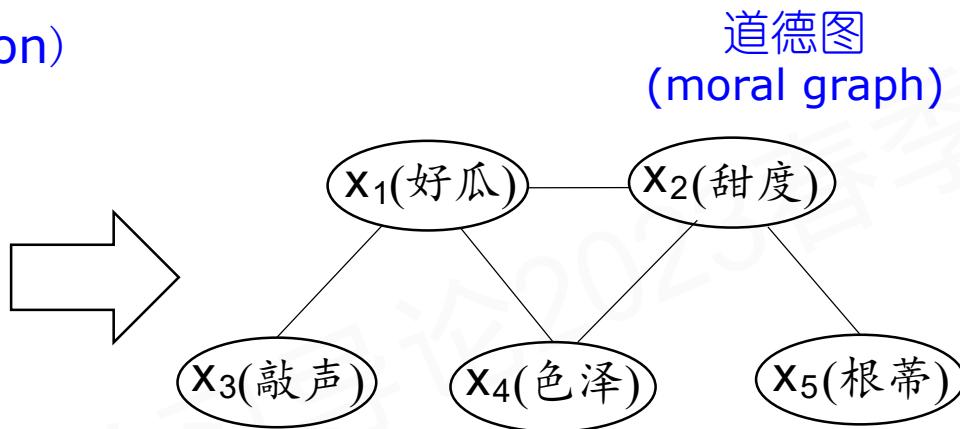
- 若  $x_4$  已知，则  $x_1$  与  $x_2$  不独立
- 若  $x_4$  未知，则  $x_1$  与  $x_2$  独立

# 分析条件独立性

## “有向分离” (D-separation)

先将有向图转变为无向图

- V型结构父结点相连
- 有向边变成无向边



先剪枝，仅保留有向图中  $x, y, z$  及其祖先结点

若  $x$  和  $y$  能在图上被  $z$  分入两个连通分支，则有

$$x \perp y \mid z.$$

由图可得：  $x_3 \perp x_4 \mid x_1$

$$x_3 \perp x_2 \mid x_1$$

$$x_3 \perp x_5 \mid x_1$$

$$x_4 \perp x_5 \mid x_2$$

$$x_3 \perp x_5 \mid x_2$$

.....

得到条件独立性关系之后，估计出条件概率表，就得到了最终网络

# 结构学习

评分函数(score function)评估贝叶斯网与训练数据的契合程度

常用评分函数通常基于信息论准则

回忆“模型选择”

例如 最小描述长度 (MDL, Minimal Description Length)

给定数据集  $D$ , 贝叶斯网  $B = \langle G, \Theta \rangle$  在  $D$  上的评分函数:

$$s(B | D) = f(\theta)|B| - LL(B | D) \quad \text{越小越好}$$

- AIC:  $f(\theta) = 1$
- BIC:  $f(\theta) = \frac{1}{2} \log m$
- ... ...

$|B|$  是贝叶斯网的参数个数

$f(\theta)$  表示描述每个参数  $\theta$  所需的比特数

$$LL(B | D) = \sum_{i=1}^m \log P_B(x_i)$$

搜索最优贝叶斯网络结构是NP难问题

# 推断

推断(inference)：基于已知属性变量的观测值，  
推测其他属性变量的取值

已知属性变量的观测值称为“证据”(evidence)

- 精确推断：直接根据贝叶斯网定义的联合概率分布来精确计算后验概率
- 近似推断：降低精度要求，在有限时间内求得近似解



常见做法：

- 吉布斯采样 (Gibbs sampling)
- 变分推断 (variational inference)

# 吉布斯采样

- 随机产生一个与证据  $\mathbf{E} = \mathbf{e}$  一致的样本  $\mathbf{q}^0$  作为初始点

例如 证据  $\mathbf{E} = \mathbf{e}$ : (色泽; 敲声; 根蒂) = (青绿; 浊响; 蠕缩)

查询目标  $\mathbf{Q} = \mathbf{q}$ : (好瓜; 甜度)= (是;高)

随机产生  $\mathbf{q}^0$ : (否; 高)

- 进行  $T$  次采样，每次采样中逐个考察每个非证据变量：假定所有其他属性取当前值，推断出采样概率，然后根据该概率采样

例如：先假定 {色泽=青绿; 敲声=浊响; 根蒂=蠕缩; 甜度=高}，推断出“好瓜”的采样概率，然后采样；假设采样结果为“好瓜=是”；

然后根据 {色泽=青绿; 敲声=浊响; 根蒂=蠕缩; 好瓜=是}，推断出“甜度”的采样概率，然后采样；假设采样结果为“甜度=高”；……

- 假定经过  $T$  次采样的得到与“查询目标”  $\mathbf{q}$  一致的样本共有  $n_q$  个，则可近似估算出后验概率

$$P(\mathbf{Q} = \mathbf{q} \mid \mathbf{E} = \mathbf{e}) \simeq \frac{n_q}{T}$$

# EM算法

如何处理“未观测到的”变量？

例如，西瓜已经脱落的根蒂，无法看出是“蜷缩”还是“坚挺”，  
则训练样本的“根蒂”属性变量值未知

未观测变量 → 隐变量 (latent variable)

EM(Expectation-Maximization) 算法是估计隐变量的利器

令  $\mathbf{X}$  表示已观测变量集， $\mathbf{Z}$  表示隐变量集，欲对模型参数  $\Theta$  做极大似然估计，则应最大化对数似然函数

$$LL(\Theta \mid \mathbf{X}, \mathbf{Z}) = \ln P(\mathbf{X}, \mathbf{Z} \mid \Theta)$$

$\mathbf{Z}$  是隐变量，无法直接求解。怎么办？

# EM算法(续)

对隐变量  $Z$  计算期望，根据训练数据最大化对数“边际似然”  
(marginal likelihood)

$$LL(\Theta | \mathbf{X}) = \ln P(\mathbf{X} | \Theta) = \ln \sum_{\mathbf{Z}} P(\mathbf{X}, \mathbf{Z} | \Theta)$$

以初始值  $\Theta^0$  为起点，迭代执行以下步骤直至收敛：

- 基于  $\Theta^t$  推断隐变量  $Z$  的期望，记为  $\mathbf{Z}^t$
- 基于已观测变量  $\mathbf{X}$  和  $\mathbf{Z}^t$  对参数  $\Theta$  做极大似然估计，记为  $\Theta^{t+1}$

**E步**: 当  $\Theta$  已知  $\rightarrow$  根据训练数据推断隐变量  $Z$

**M步**: 当  $Z$  已知  $\rightarrow$  对  $\Theta$  做极大似然估计

一般形式：**E-M** 两个步骤交替计算，直至收敛：

- **E步 - 计算期望**: 利用当前估计的参数值计算对数似然的期望值；
- **M步 - 最大化**: 寻找能使**E步**产生的似然期望最大化的参数值；

# EM算法的应用 (GMM模型回顾)

采用概率模型来表达聚类原型

$n$  维样本空间中的随机向量  $\mathbf{x}$  若服从高斯分布, 则其概率密度函数为

$$p(\mathbf{x}) = \frac{1}{(2\pi)^{\frac{n}{2}} |\Sigma|^{\frac{1}{2}}} e^{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^T \Sigma^{-1} (\mathbf{x}-\boldsymbol{\mu})}$$

假设样本由下面这个高斯混合分布生成:

生成式模型

$$p_{\mathcal{M}}(\mathbf{x}) = \sum_{i=1}^k \alpha_i \cdot p(\mathbf{x} \mid \boldsymbol{\mu}_i, \Sigma_i)$$

样本  $\mathbf{x}_j$  由第  $i$  个高斯混合成分生成的后验概率为:

$$p_{\mathcal{M}}(z_j = i \mid \mathbf{x}_j) = \frac{P(z_j = i) \cdot p_{\mathcal{M}}(\mathbf{x}_j \mid z_j = i)}{p_{\mathcal{M}}(\mathbf{x}_j)} = \frac{\alpha_i \cdot p(\mathbf{x}_j \mid \boldsymbol{\mu}_i, \Sigma_i)}{\sum_{l=1}^k \alpha_l \cdot p(\mathbf{x}_j \mid \boldsymbol{\mu}_l, \Sigma_l)}$$

简记为  $\gamma_{ji}$  ( $i = 1, 2, \dots, k$ )

# EM算法的应用 (GMM模型回顾)

参数估计可采用极大似然法，考虑最大化对数似然

$$LL(D) = \ln \left( \prod_{j=1}^m p_{\mathcal{M}}(\mathbf{x}_j) \right) = \sum_{j=1}^m \ln \left( \sum_{i=1}^k \alpha_i \cdot p(\mathbf{x}_j \mid \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i) \right)$$

## EM 算法：

- (E步) 根据当前参数计算每个样本属于每个高斯成分的后验概率  $\gamma_{ji}$
- (M步) 更新模型参数 $\{(\alpha_i, \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i) \mid 1 \leq i \leq k\}$

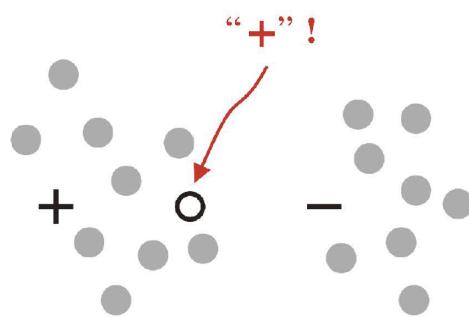
输入: 样本集  $D = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_m\}$ ;  
高斯混合成分个数  $k$ .

过程:

```
1: 初始化高斯混合分布的模型参数  $\{(\alpha_i, \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i) \mid 1 \leq i \leq k\}$ 
2: repeat
3:   for  $j = 1, 2, \dots, m$  do
4:     根据式(9.30)计算  $\mathbf{x}_j$  由各混合成分生成的后验概率, 即
        $\gamma_{ji} = p_{\mathcal{M}}(z_j = i \mid \mathbf{x}_j) (1 \leq i \leq k)$ 
5:   end for
6:   for  $i = 1, 2, \dots, k$  do
7:     计算新均值向量:  $\boldsymbol{\mu}'_i = \frac{\sum_{j=1}^m \gamma_{ji} \mathbf{x}_j}{\sum_{j=1}^m \gamma_{ji}}$ ;
8:     计算新协方差矩阵:  $\boldsymbol{\Sigma}'_i = \frac{\sum_{j=1}^m \gamma_{ji} (\mathbf{x}_j - \boldsymbol{\mu}'_i)(\mathbf{x}_j - \boldsymbol{\mu}'_i)^T}{\sum_{j=1}^m \gamma_{ji}}$ ;
9:     计算新混合系数:  $\alpha'_i = \frac{\sum_{j=1}^m \gamma_{ji}}{m}$ ;
10:    end for
11:    将模型参数  $\{(\alpha_i, \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i) \mid 1 \leq i \leq k\}$  更新为  $\{(\alpha'_i, \boldsymbol{\mu}'_i, \boldsymbol{\Sigma}'_i) \mid 1 \leq i \leq k\}$ 
12: until 满足停止条件
13:  $C_i = \emptyset (1 \leq i \leq k)$ 
14: for  $j = 1, 2, \dots, m$  do
15:   根据式(9.31)确定  $\mathbf{x}_j$  的簇标记  $\lambda_j$ ;
16:   将  $\mathbf{x}_j$  划入相应的簇:  $C_{\lambda_j} = C_{\lambda_j} \cup \{\mathbf{x}_j\}$ 
17: end for
```

输出: 簇划分  $\mathcal{C} = \{C_1, C_2, \dots, C_k\}$

# EM算法的应用（半监督学习）



存在大量无标记数据  
考虑最大化对数似然

标记数据  $D_l$ ：可知对应的高斯分布

无标记数据  $D_u$ ：使用和聚类一致的思路

$$\begin{aligned} LL(D_l \cup D_u) = & \sum_{(x_j, y_j) \in D_l} \ln \left( \sum_{i=1}^N \alpha_i \cdot p(x_j | \mu_i, \Sigma_i) \cdot p(y_j | \Theta = i, x_j) \right) \\ & + \sum_{x_j \in D_u} \ln \left( \sum_{i=1}^N \alpha_i \cdot p(x_j | \mu_i, \Sigma_i) \right) \end{aligned}$$

半监督学习 → 13章

## EM 算法：

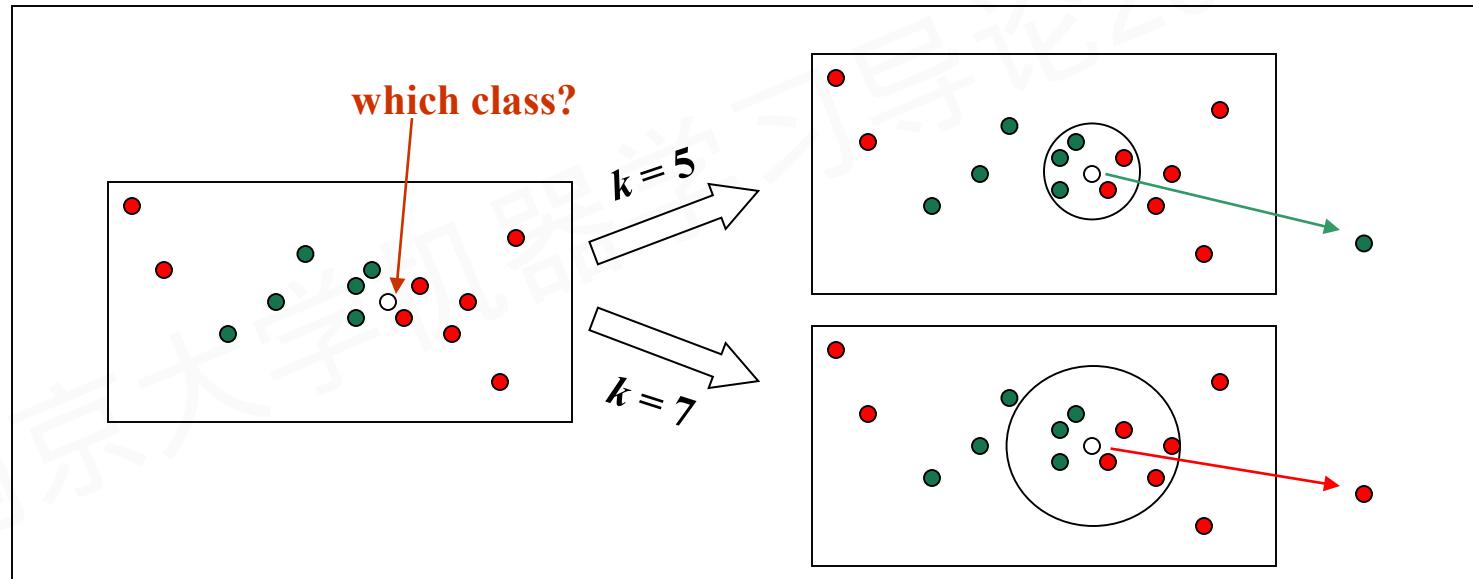
- (E步) 根据当前参数计算无标记样本属于每个高斯成分的后验概率  $\gamma_{ji}$
- (M步) 更新模型参数  $\{(\alpha_i, \mu_i, \Sigma_i) \mid 1 \leq i \leq k\}$

# $k$ 近邻分类

$k$  近邻 ( $k$ -Nearest Neighbor, kNN)

懒惰学习 (lazy learning) 的代表

基本思路：  
近朱者赤，近墨者黑  
(投票法；平均法)



关键：  $k$  值选取； 距离计算

# 最近邻学习器和贝叶斯最优分类器

给定测试样本  $\mathbf{x}$  , 若其最近邻样本为  $\mathbf{z}$  , 则最近邻分类器出错的概率就是  $\mathbf{x}$  和  $\mathbf{z}$  类别标记不同的概率,

$$\begin{aligned} P(\text{err}) &= 1 - \sum_{c \in \mathcal{Y}} P(c \mid \mathbf{x}) P(c \mid \mathbf{z}) \\ &\simeq 1 - \sum_{c \in \mathcal{Y}} P^2(c \mid \mathbf{x}) \\ &\leq 1 - P^2(c^* \mid \mathbf{x}) \\ &= (1 + P(c^* \mid \mathbf{x})) (1 - P(c^* \mid \mathbf{x})) \\ &\leq 2 \times (1 - P(c^* \mid \mathbf{x})). \end{aligned}$$

最近邻分离器的泛化错误率  
不会超过贝叶斯最优分类器  
错误率的两倍!

但是在真实的应用中， 我们是否能够准确的找到  $k$  近邻呢？ → 第10章  
使用何种距离进行相似性判断？

# 距离度量的种类

## □ 闵可夫斯基距离

$$\text{dist}_{\text{mk}}(x_i, x_j) = \left( \sum_{u=1}^n |x_{iu} - x_{ju}|^p \right)^{\frac{1}{p}}$$

## □ 欧式距离 (Euclidean Distance)

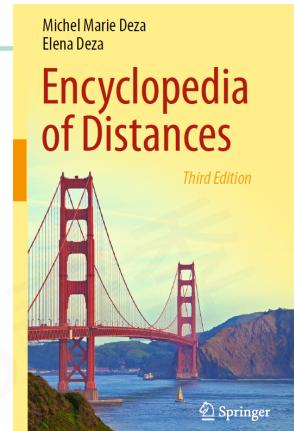
$$\text{dist}_{\text{ed}}(x_i, x_j) = \|x_i - x_j\|_2 = \sqrt{\sum_{u=1}^n |x_{iu} - x_{ju}|^2}$$

## □ 曼哈顿距离 (Manhattan distance)

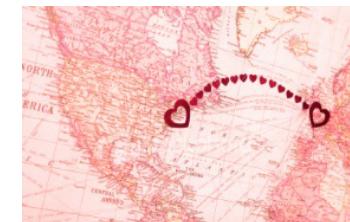
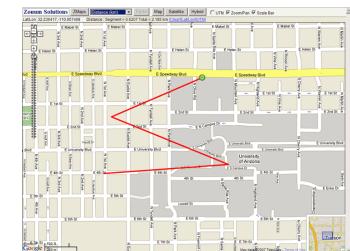
$$\text{dist}_{\text{man}}(x_i, x_j) = \|x_i - x_j\|_1 = \sum_{u=1}^n |x_{iu} - x_{ju}|$$

## □ Geodesic Distance

人工定义距离 → 从数据中学习合适的距离



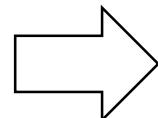
包含各种类型  
距离的大百科



# 距离度量学习 (distance metric learning)

降维的主要目的是希望找到一个“合适的”低维空间

每个空间对应了在样本属性上定义的一个距离度量



能否直接“学出”合适距离？

首先，要有可以通过学习来“参数化”的距离度量形式

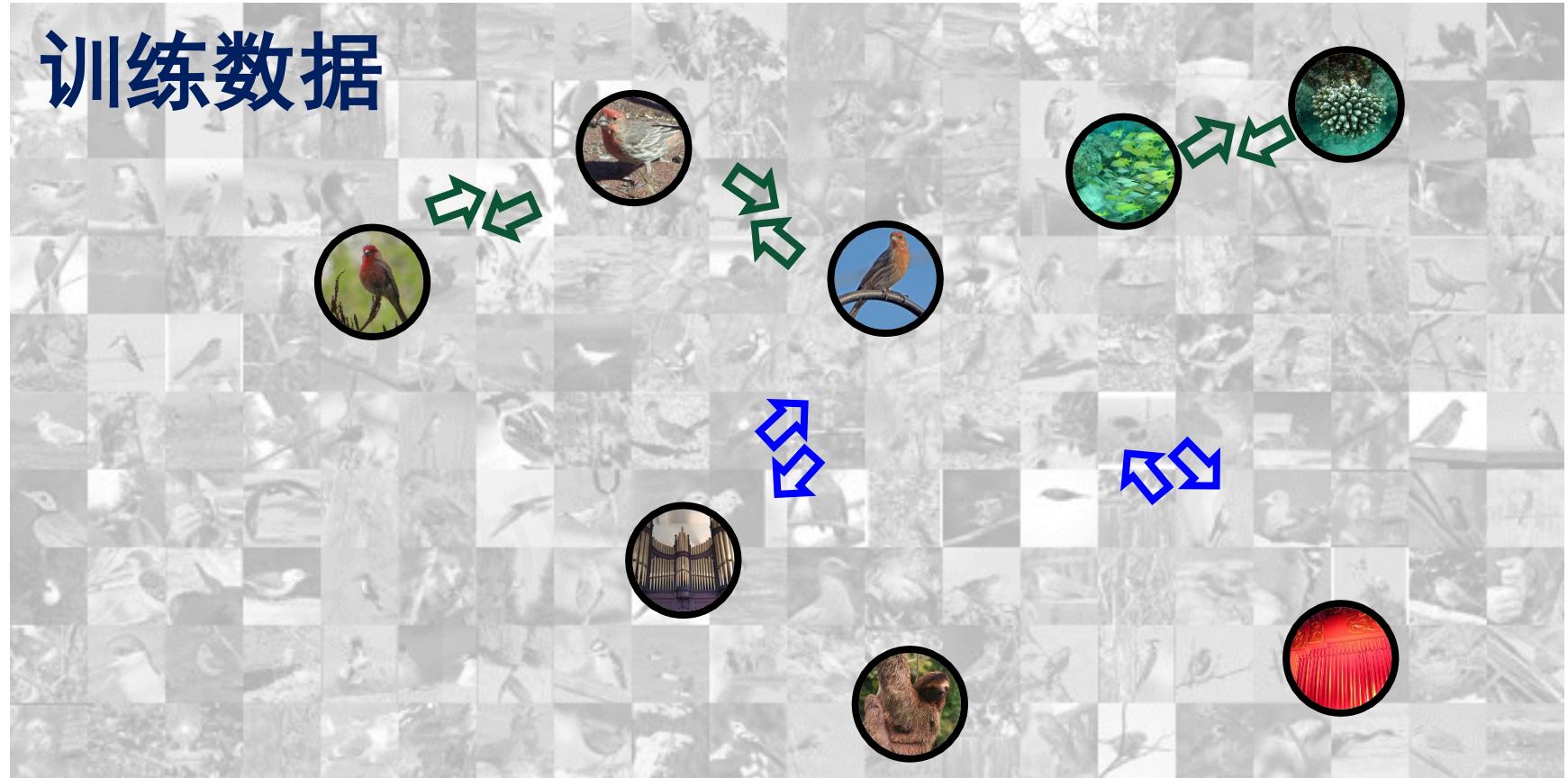
马氏距离 (Mahalanobis distance) 是一个很好的选择：

$$\text{dist}_{\text{mah}}^2(\mathbf{x}_i, \mathbf{x}_j) = (\mathbf{x}_i - \mathbf{x}_j)^T \mathbf{M} (\mathbf{x}_i - \mathbf{x}_j) = \|\mathbf{x}_i - \mathbf{x}_j\|_{\mathbf{M}}^2$$

其中  $\mathbf{M}$  是一个半正定对称矩阵，亦称“度量矩阵”

距离度量学习就是要对  $\mathbf{M}$  进行学习

# 距离度量学习 (distance metric learning)



核心思想：拉进相似样本之间的距离，推远不相似样本之间的距离

# 距离度量学习 (distance metric learning)

对  $\mathbf{M}$  进行学习的目标是什么？

- 某种分类器的性能

例如，若以近邻分类器的性能为目标，则得到 NCA

- 领域知识

例如，若已知“必连”(must-link) 约束集合  $\mathcal{M}$  与“勿连”(cannot-link) 约束集合  $\mathcal{C}$ ，则可通过求解这个凸优化问题得到  $\mathbf{M}$ ：

$$\min_{\mathbf{M}} \sum_{(\mathbf{x}_i, \mathbf{x}_j) \in \mathcal{M}} \|\mathbf{x}_i - \mathbf{x}_j\|_{\mathbf{M}}^2$$

$$\text{s.t. } \sum_{(\mathbf{x}_i, \mathbf{x}_k) \in \mathcal{C}} \|\mathbf{x}_i - \mathbf{x}_k\|_{\mathbf{M}}^2 \geq 1 ,$$

$$\mathbf{M} \succeq 0 ,$$

# 如何进行相似性的指导

使用二元组、三元组构成弱监督信息

更相似 ?



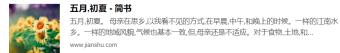
更不相似 ?



相似性关系能够被广泛获取



图片来源

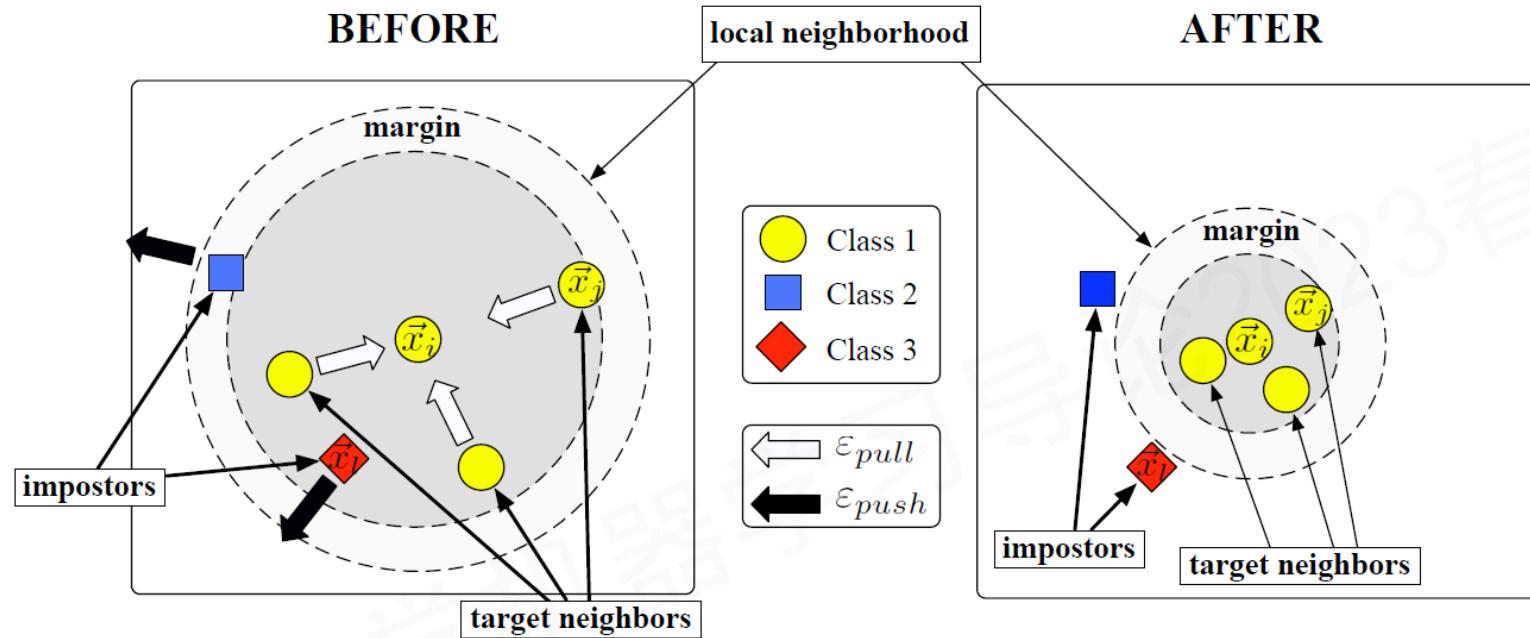


识图系统中  
用户的点击



社交网络中用  
户的好友关系

# 距离度量学习 – LMNN: Large Margin Nearest Neighbors



不相似样本远离

$$\varepsilon_{push}(\mathbf{L}) = \sum_{i,j \sim i} \sum_l (1 - y_{il}) [1 + \|\mathbf{L}(\vec{x}_i - \vec{x}_j)\|^2 - \|\mathbf{L}(\vec{x}_i - \vec{x}_l)\|^2]_+$$

相似样本接近

$$\varepsilon_{pull}(\mathbf{L}) = \sum_{j \sim i} \|\mathbf{L}(\vec{x}_i - \vec{x}_j)\|^2.$$

$$\varepsilon(\mathbf{L}) = (1 - \mu) \varepsilon_{pull}(\mathbf{L}) + \mu \varepsilon_{push}(\mathbf{L}).$$

前往下一站.....

