无监督学习
Unsupervised Learning

徐 君

提纲

- 背景介绍
- 常用无监督学习算法
  - 聚类算法
  - 矩阵分解/话题模型
  - 文本表达
- 总结

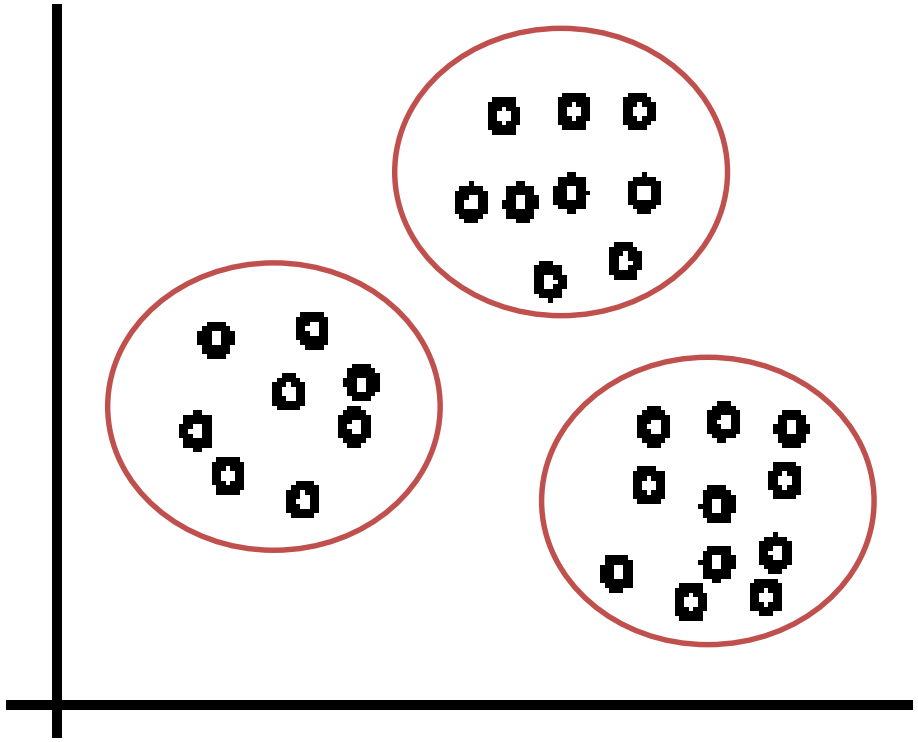
监督学习和无监督学习比较

- 监督学习
  - 建模数据中输入特征与目标类别之间的联系
  - 目标：在未知数据上进行精确预测
- 无监督学习
  - 只有输入特征，没有目标类别
  - 目标：发现数据内在的规律和结构
- 无监督学习的优点
  - 可以利用大量的数据，节省标注成本
  - 有些应用无法事先确定目标类别（如搜索结果聚类）

聚类

- 聚类的目标是发现数据中相似群，成为簇(cluster)
  - 簇内中的数据彼此距离小，簇间数据距离大
- 聚类不需要提供标注每一个簇的样本，因此被成为无监督学习
- 无监督学习往往被认为就是聚类，但本课程还将学习其他无监督学习内容
  - 矩阵分解
  - 话题模型

聚类

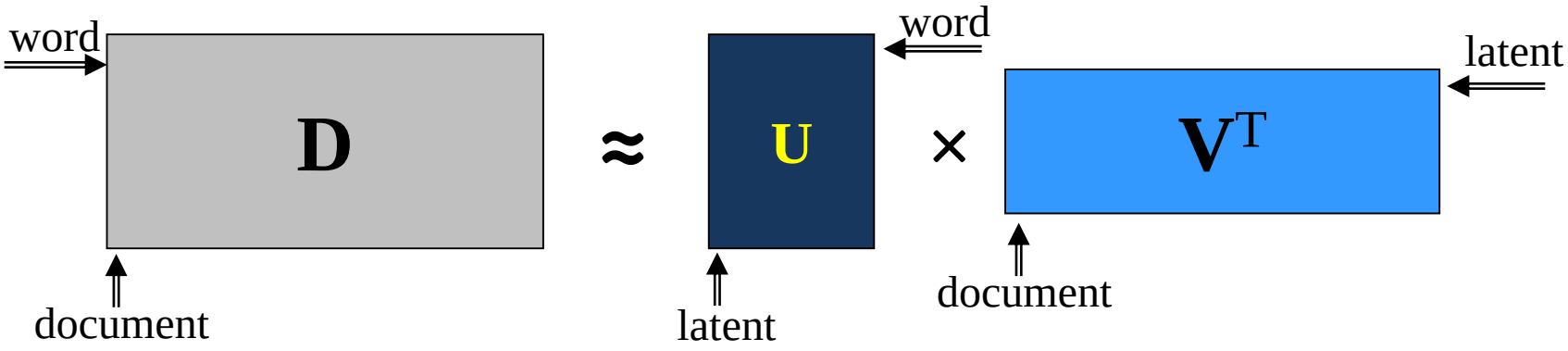


- 数据点自然聚成三个簇

聚类应用举例

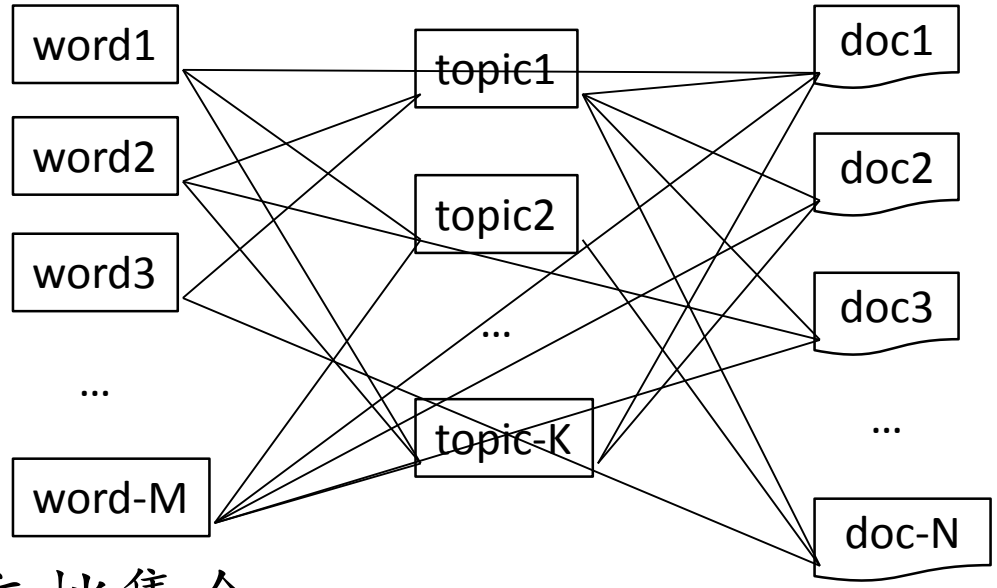
- 网页搜索结果聚类（无法事先确定目标类别）

矩阵分解



- 输入：矩阵D
- 输出：矩阵U和V
- U × V是D的低秩(low rank)近似
  - 秩: 线性无关组的个数
  - $\text{rank}(D) \leq \min(\text{rank}(U), \text{rank}(V))$

矩阵分解应用:主题模型



- 输入：文档集合
- 目标：发现文档中“潜在的”话题
- 输出
  - 潜在话题
  - 用话题表达的文档

解决搜索中查询-文档匹配问题

- 搜索中的查询-文档匹配
  - 查询: “OPEC”， 文档: “oil”

$$s(q,d) = \alpha s_{topic}(q,d) + (1-\alpha)s_{term}(q,d)$$

Topic1	Topic2	Topic3	Topic4	Topic5	Topic6	Topic7	Topic8	Topic9	Topic10
OPEC	Africa	contra	school	Noriega	firefight	plane	Saturday	Iran	senate
oil	South	Sandinista	student	Panama	ACR	crash	coastal	Iranian	Reagan
cent	African	rebel	teacher	Panamanian	forest	flight	estimate	Iraq	billion
barrel	Angola	Nicaragua	education	Delval	park	air	western	hostage	budget
price	apartheid	Nicaraguan	college	canal	blaze	airline	Minsch	Iraqi	Trade

提纲

- 背景介绍
- 常用无监督学习算法
  - 聚类算法
  - 矩阵分解/话题模型
  - 文本表达
- 总结

聚类需要考虑的几个因素

- 聚类算法
  - 基于划分的聚类算法
  - 基于层次的聚类算法
  - ……
- 相似度/距离函数
- 聚类质量评价
  - 最小化类内距(inter-clusters distance)
  - 最大化类间距(intra-clusters distance)

K-Means

- 最著名的聚类算法
- 基于数据划分，一个数据点只能属于一个簇
- 输入：$D = \{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_N\}$

  数据点$\mathbf{x}_i \in R^d$是d维数值型向量
- K-Means算法将数据划分为K个簇
  - K值由用户指定
  - 每一个簇有一个中心，称为centroid

K-Means算法

- 输入：数据$D = \{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_N\}$，簇数目K
  1. 随机选取K个种子数据点(seeds)作为K个簇中心
  2. repeat
  3. foreach $\mathbf{x} \in \boldsymbol{D}$ do
  4. 计算**x**与每一个簇中心的距离
  5. 将**x**指配到距离最近的簇中心
  6. endfor
  7. 用当前的簇内点重新计算K个簇中心位置
  8. until (达到终止条件)

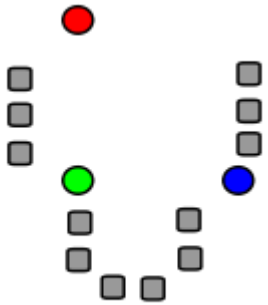
终止条件

- 数据不再（很少）被重新指定到不同的簇
- 中心位置不再发生变化或者变化很小
- SSE(sum of squared error)不再减少

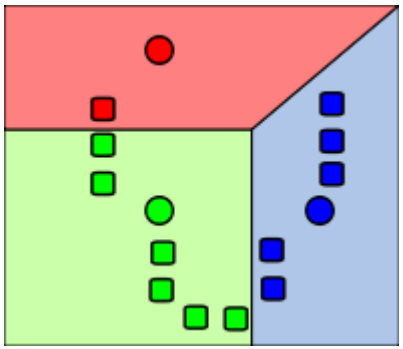
$$\text{SSE} = \sum_{k=1}^{K} \sum_{\mathrm{x}_i \in C_k} dist(\mathbf{x}_i, \mathbf{m}_k)$$

  - C_k:第k个簇
  - $\mathbf{m}_k$： C_k的中心
  - $dist(\mathbf{x}_i, \mathbf{m}_k)$：数据点$\mathbf{x}_i$与簇中心$\mathbf{m}_k$的距离

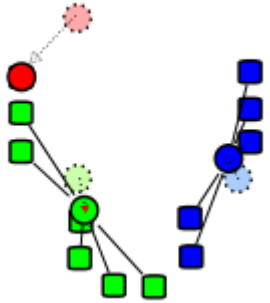
K-Means运行过程



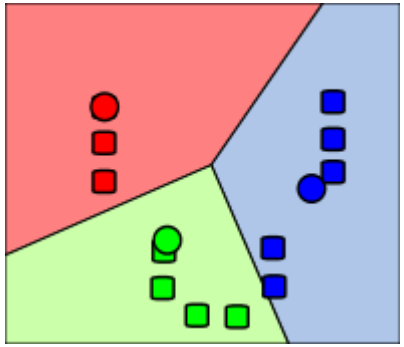
初始中心



数据划分



更新中心



更新数据划分

距离的定义

- 在欧氏空间中，平均点(mean)可以定义为

$$\mathbf{m}_k = \frac{1}{|C_k|}\sum_{\mathrm{x}_i \in C_k} \mathbf{x}_i$$

其中$|C_k|$为第k个簇中含有的数据点数目。

- 数据点$\mathbf{x}_i$与中心$\mathbf{m}_k$的距离定义为

$$dist(\mathbf{x}_i, \mathbf{m}_k) = |\mathbf{x}_i - \mathbf{m}_k|$$
$$= \sqrt{(x_{i1} - m_{k1})^2 + \cdots + (x_{id} - m_{kd})^2}$$

距离的定义

- 距离函数需要满足的条件

  假设$\mathbf{x}_i, \mathbf{x}_j$和$\mathbf{x}_k$为数据点，距离函数$dist(\cdot,\cdot)$满足

  - $dist(\mathbf{x}_i, \mathbf{x}_j) \geq 0, dist(\mathbf{x}_i, \mathbf{x}_j) = 0$当且仅当$\mathbf{x}_i = \mathbf{x}_j$
  - $dist(\mathbf{x}_i, \mathbf{x}_j) = dist(\mathbf{x}_j, \mathbf{x}_i)$
  - 三角不等式：$dist(\mathbf{x}_i, \mathbf{x}_j) \leq dist(\mathbf{x}_i, \mathbf{x}_k) + dist(\mathbf{x}_k, \mathbf{x}_j)$

- 有很多不同的距离定义
  - 数值型数据
  - 类别型数据

Minkowski距离

$$dist(\mathbf{x}_i, \mathbf{x}_j) = \left(\left(x_{i1} - x_{j1}\right)^h + \cdots + \left(x_{id} - x_{jd}\right)^h \right)^{\frac{1}{h}}$$

其中h为正整数

- 欧氏距离(Euclidean distance) h=2

$$dist(\mathbf{x}_i, \mathbf{x}_j) = \sqrt{\left(x_{i1} - x_{j1}\right)^2 + \cdots + \left(x_{id} - x_{jd}\right)^2}$$

- 带权欧氏距离

$$dist(\mathbf{x}_i, \mathbf{x}_j) = \sqrt{w_1\left(x_{i1} - x_{j1}\right)^2 + \cdots + w_d\left(x_{id} - x_{jd}\right)^2}$$

- 曼哈顿距离(Manhattan distance) h=1

$$dist(\mathbf{x}_i, \mathbf{x}_j) = \left|x_{i1} - x_{j1}\right| + \cdots + \left|x_{id} - x_{jd}\right|$$

- Chebychev距离 $h \to +\infty$

$$dist(\mathbf{x}_i, \mathbf{x}_j) = \max\left(\left|x_{i1} - x_{j1}\right|, \cdots, \left|x_{id} - x_{jd}\right|\right)$$

二值数据距离定义

- 二值数据（取值为0或者1, binary）
  - 如性别、是否参加工作等
- 将两个向量的匹配情况表示为一个confusion matrix

Data point i \ Data point j	1	0	
1	a	b	$a+b$
0	c	d	$c+d$
	$a+c$	$b+d$	$a+b+c+d$

(10)

a: the number of attributes with the value of 1 for both data points.
b: the number of attributes for which $x_{if} = 1$ and $x_{jf} = 0$, where x_{if} (x_{jf}) is the value of the fth attribute of the data point $\mathbf{x}_i$ ($\mathbf{x}_j$).
c: the number of attributes for which $x_{if} = 0$ and $x_{jf} = 1$.
d: the number of attributes with the value of 0 for both data points.

二值数据距离定义

- 数值1与数值0具有相同的权重（如性别）

$$dist(\mathbf{x}_i, \mathbf{x}_j) = \frac{b + c}{a + b + c + d}$$

$\mathbf{x}_1$	1	1	1	0	1	0	0
$\mathbf{x}_2$	0	1	1	0	0	1	0

$$dist(\mathbf{x}_i, \mathbf{x}_j) = \frac{2+1}{2+2+1+2} = \frac{3}{7} = 0.429$$

二值数据距离定义

- 数值1比数值0更加重要（如文档中出现某个关键词），Jaccard coefficient

$$dist(\mathbf{x}_i, \mathbf{x}_j) = \frac{b + c}{a + b + c}$$

$\mathbf{x}_1$	1	1	1	0	1	0	0
$\mathbf{x}_2$	0	1	1	0	0	1	0

$$dist(\mathbf{x}_i, \mathbf{x}_j) = \frac{2 + 1}{2 + 2 + 1} = 0.6$$

多值数据距离定义

- 有多于两个的取值
  - 颜色
  - 收入水平
- Simple matching method
  - $\mathbf{x}_i$和$\mathbf{x}_j$具有d个特征，其中取值相同的特征有q个，则距离定义为

$$dist(\mathbf{x}_i, \mathbf{x}_j) = \frac{d - q}{d}$$

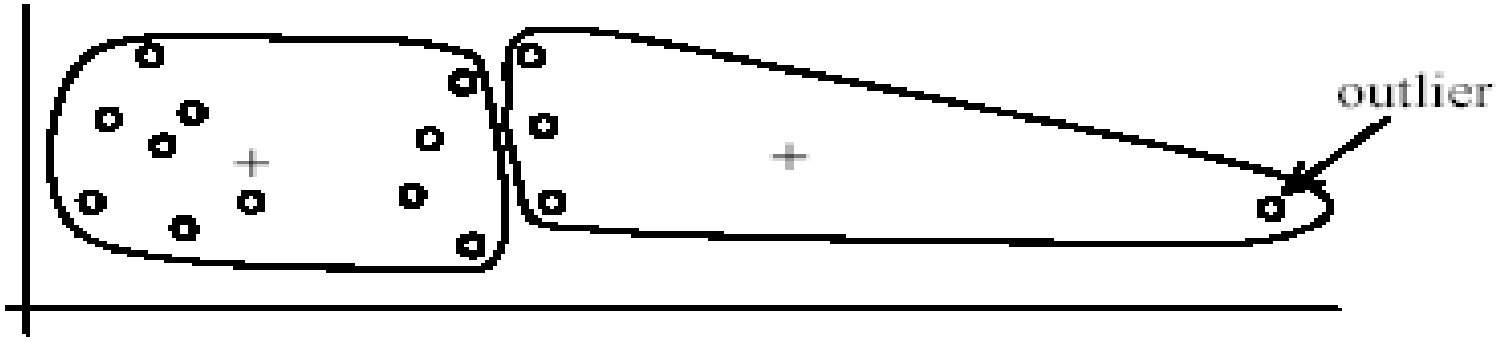
K-Means优点

- 简单：易于理解和实现
- 高效：时间复杂度为$O(TKN)$,其中T为运行轮数，K为簇数目，N为样本数。
- K-Means是最广为人知的聚类算法

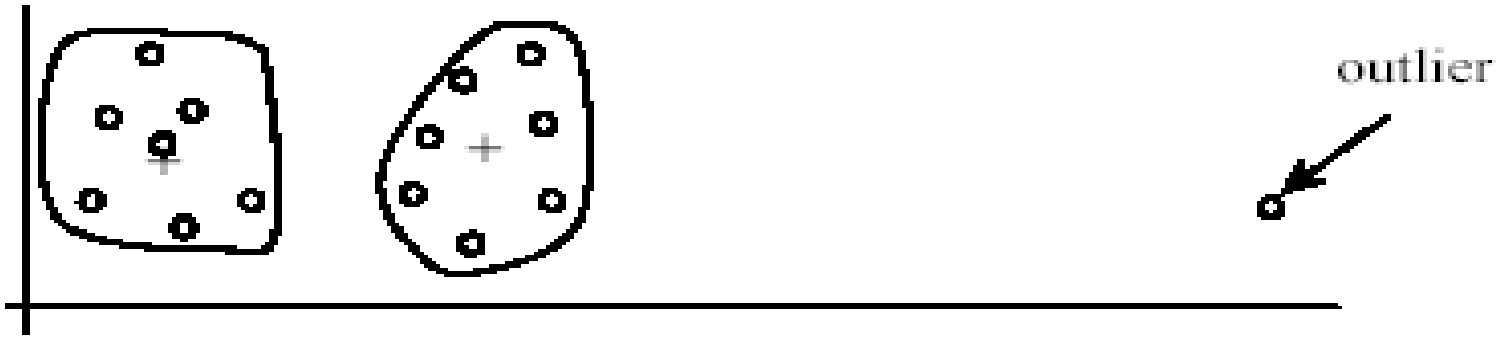
K-Means缺点

- 局部最优解
  - 每一个数据点贪心选择距离最近的中心
  - 不同初始中心可能得到不同的簇
- 平均值点(“mean”)有定义
  - 一般来说要求特征为数值型
  - 对于类别型特征不适应
- 用户指定K值
  - 比较难以提前预知
- 对离群点(outlier)敏感
  - 离群点为距离其他数据点很远的点
  - 可能是在数据准备/预处理过程中出现的错误

离群点导致的问题



(A): Undesirable clusters

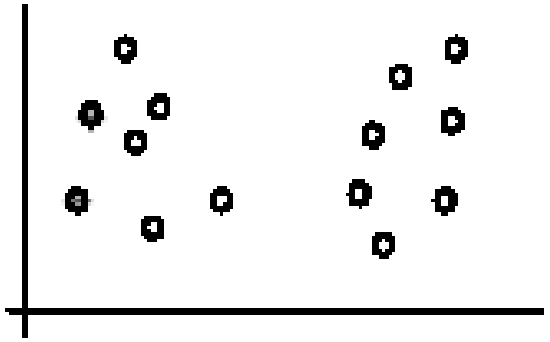


(B): Ideal clusters

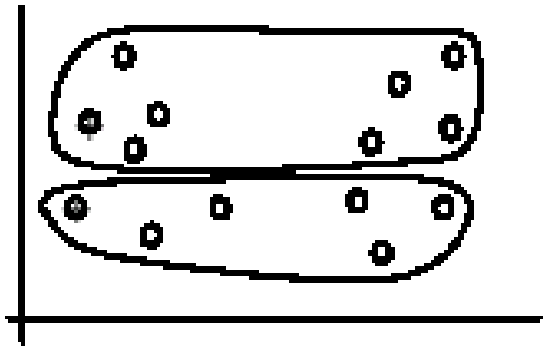
避免离群点的方法

- 删除数据
  - 在聚类过程中移除距离中心点过远的数据（显著大于其他数据与中心点的距离）
- 随机采样
  - 离群点被采样到的概率很小
  - 未被采样到的数据在算法运行结束后直接指定给响应的中心点

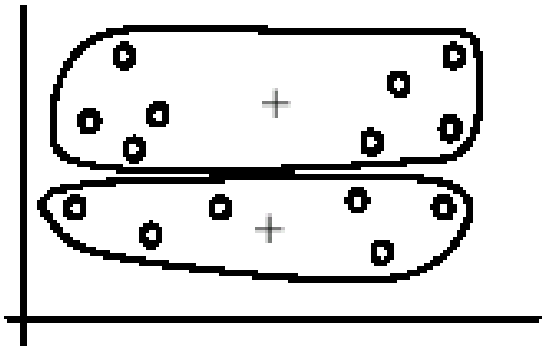
初始中心对聚类结果的影响



(A). Random selection of seeds (centroids)

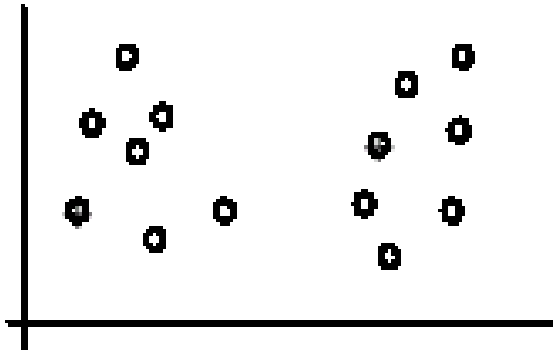


(B). Iteration 1

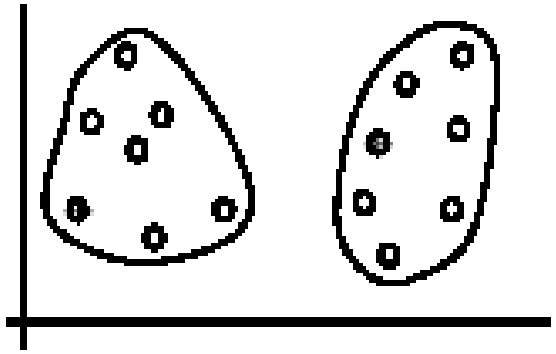


(C). Iteration 2

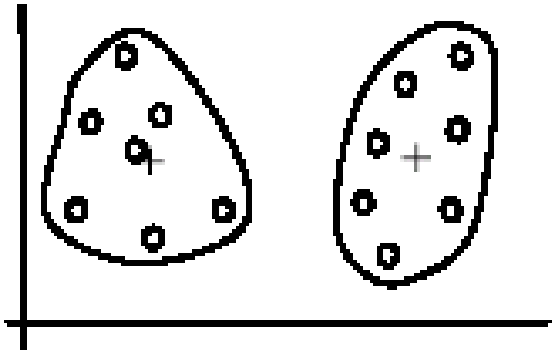
初始中心对聚类结果的影响



(A). Random selection of k seeds (centroids)



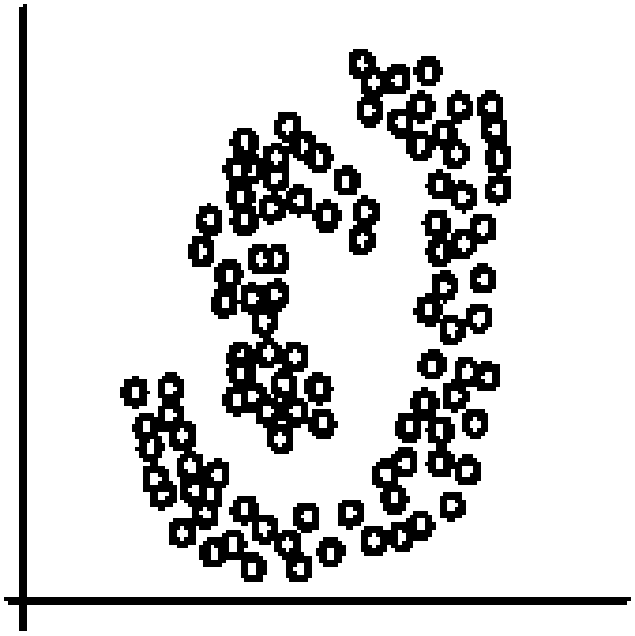
(B). Iteration 1



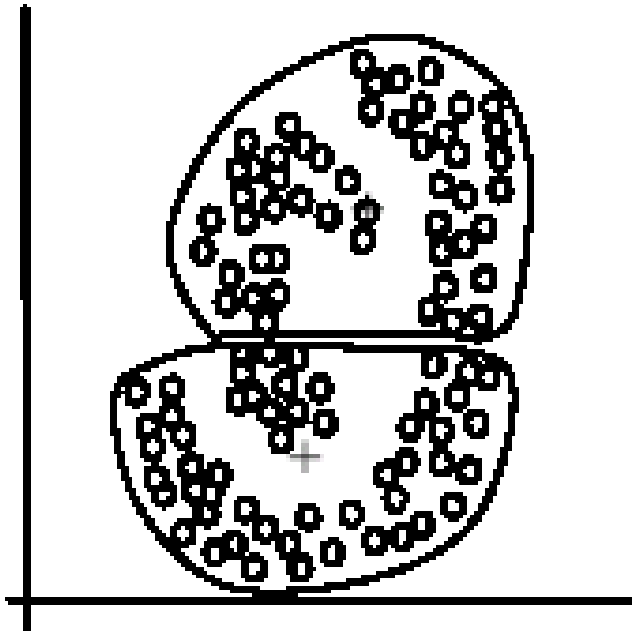
(C). Iteration 2

复杂形状聚类

- K-Means进行数据指定时独立考虑单个数据点与中心的距离，不考虑其它数据点



(A): Two natural clusters



(B): *k*-means clusters

K-Means总结

- 虽然存在诸多缺陷，K-Means任然是最为广泛使用的算法(简单、高效)
  - 其它算法都有优点和缺陷
- 不同的算法基于不同的假设，适合不同的数据和应用任务
  - 没有普遍意义上的“最优”算法
- 聚类评价：没有统一的直接评价标准，难以比较不同的聚类算法优劣
  - 无监督学习普遍存在的问题
  - 间接评价：评价使用聚类结果后对其他应用任务的改进幅度

聚类结果评价

- 对聚类结果评价很困难，原因在于
  - 我们没有正确答案
- 间接评价
  - 将聚类结果用于其他应用（如搜索排序），评价对其他应用效果的提升幅度
- 直接评价

直接评价方法

- 假设每一个簇就是一个类别，则可以利用标注好的数据对聚类算法进行评价
- 根据聚类结果与标注数据构造confusion matrix
  - 假设数据D被标注成k个类别$C = \{c_1, \cdots, c_k\}$，聚类结果将D划分为k个簇$\mathrm{D} = \{D_1, \cdots, D_k\}$

评价方法

- 熵(entropy)
  - 单个簇的熵定义$H(D_i) = \sum_{j=1}^{k} \Pr(c_j) \log \Pr(c_j)$, 其中$\Pr(c_i)$为$D_i$中标注为$c_i$的数据比例
  - 聚类结果的总体熵为

$$H_{total}(D) = \sum_{i=1}^{k} \frac{|D_i|}{|D|} H(D_i)$$

- 簇的纯度(purity)
  - 单个簇的纯度定义$\text{purity}(D_i) = \max_j \Pr(c_j)$
  - 聚类结果的总体纯度

$$\text{purity}_{total}(D) = \sum_{i=1}^{k} \frac{|D_i|}{|D|} \text{purity}(D_i)$$

举例

Example 14: Assume we have a text collection D of 900 documents from three topics (or three classes), Science, Sports, and Politics. Each class has 300 documents. Each document in D is labeled with one of the topics (classes). We use this collection to perform clustering to find three clusters. Note that class/topic labels are not used in clustering. After clustering, we want to measure the effectiveness of the clustering algorithm.

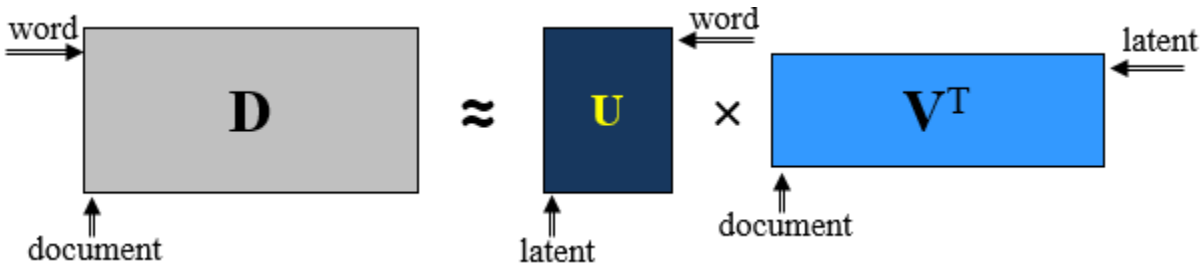
Cluster	Science	Sports	Politics		Entropy	Purity
1	250	20	10		0.589	0.893
2	20	180	80		1.198	0.643
3	30	100	210		1.257	0.617
Total	300	300	300		1.031	0.711

提纲

- 背景介绍
- 常用无监督学习算法
  - 聚类算法
  - 矩阵分解/话题模型
  - 文本表达
- 总结

矩阵分解背景介绍

- 数的分解
  - 6=2*3，12=2*3*3
- 矩阵分解(matrix factorization, matrix decomposition):一个矩阵分解为多个矩阵的乘积



  - 高维的矩阵的低秩近似

矩阵的秩

- 对于一个$M \times N$矩阵A,其秩rank(A)为矩阵A中线性无关的行(或者列)的最大数目
- 矩阵其行向量或者列向量所张成的空间的维度
  - 行秩=列秩
  - rank(A)≤ $\min(M, N)$
  - rank(AB)≤ $\min\big(\text{rank}(A), \text{rank}(B)\big)$

$$\begin{bmatrix} 1 & 2 & 1 \\ -2 & -3 & 1 \\ 3 & 5 & 0 \end{bmatrix}$$

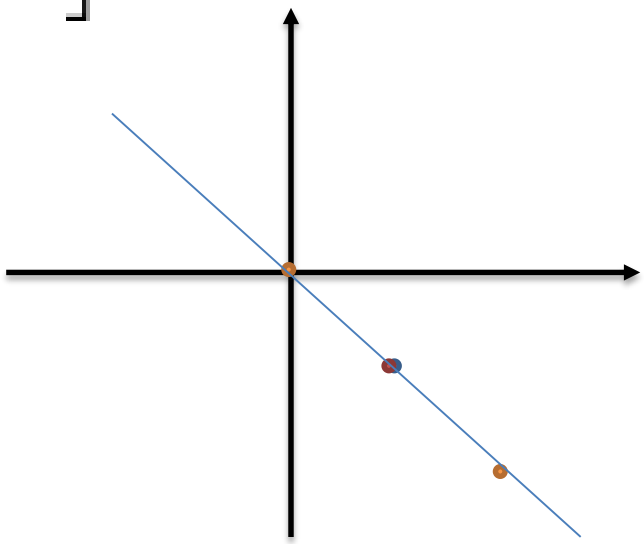
rank=2

$$A = \begin{bmatrix} 1 & 1 & 0 & 2 \\ -1 & -1 & 0 & -2 \end{bmatrix}$$

rank=1

列向量张成的空间

$$A = \begin{bmatrix} 1 & 1 & 0 & 2 \\ -1 & -1 & 0 & -2 \end{bmatrix}$$



按行看，4维空间，2个数据点关于原点中心对称

按列看，2维空间，4个点位于过原点的直线上

矩阵分解的应用

- 推荐系统
  - 用户-物品矩阵，用户对物品的评分
  - 协同过滤：给用户推荐新的物品
- 文本挖掘
  - 文档-词矩阵，单词在文档中出现的次数
  - 话题模型：用话题表达文档
- 社交网络
  - 节点-节点矩阵，表示节点间的链接（边）
  - 链接预测：在图中建立新的链接

常用的矩阵分解算法

- SVD分解(Singular value decomposition)
  - 数学优美
  - 计算复杂
- 非负矩阵分解NMF (nonnegative matrix factorization)
  - 对输入数据和输出矩阵有非负要求
  - 放弃正交限制
  - 容易理解
  - 计算相对简单，能够大规模并行

SVD分解

- 输入：矩阵$D_{M\times N}=(\mathbf{x}_1,\cdots,\mathbf{x}_N),\mathbf{x}_i\in R^M$
- SVD分解

$$D=\sum_{k=1}^{p}\sigma_k\mathbf{u}_k\mathbf{v}_k^T=U\Sigma V^T$$

$\Sigma=\begin{bmatrix}\sigma_1 & 0 & 0\\ 0 & \ddots & 0\\ 0 & 0 & \sigma_p\end{bmatrix}$:特征值(eigenvalues)，表示此维度上的“方差”（能量）

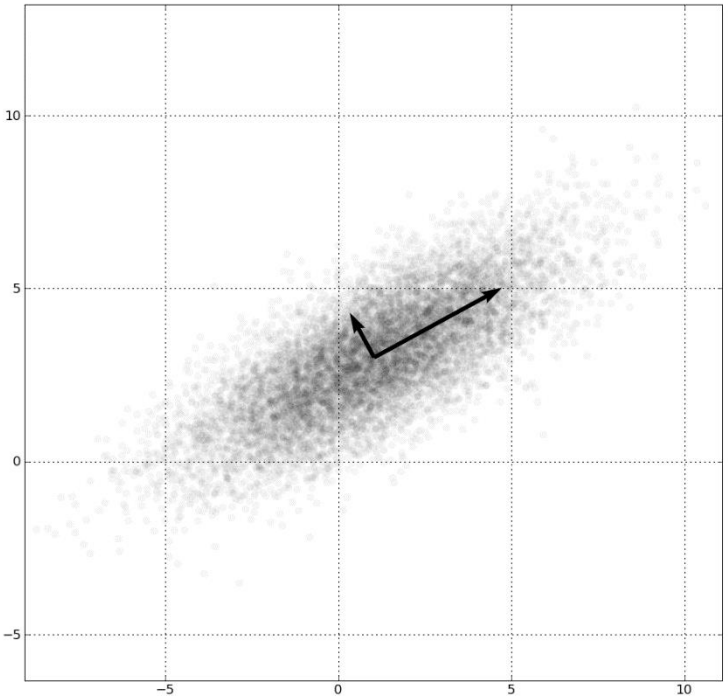
$\mathbf{u}_k$、$\mathbf{v}_k^T$:第σ_k对应的特征向量(eigenvectors)

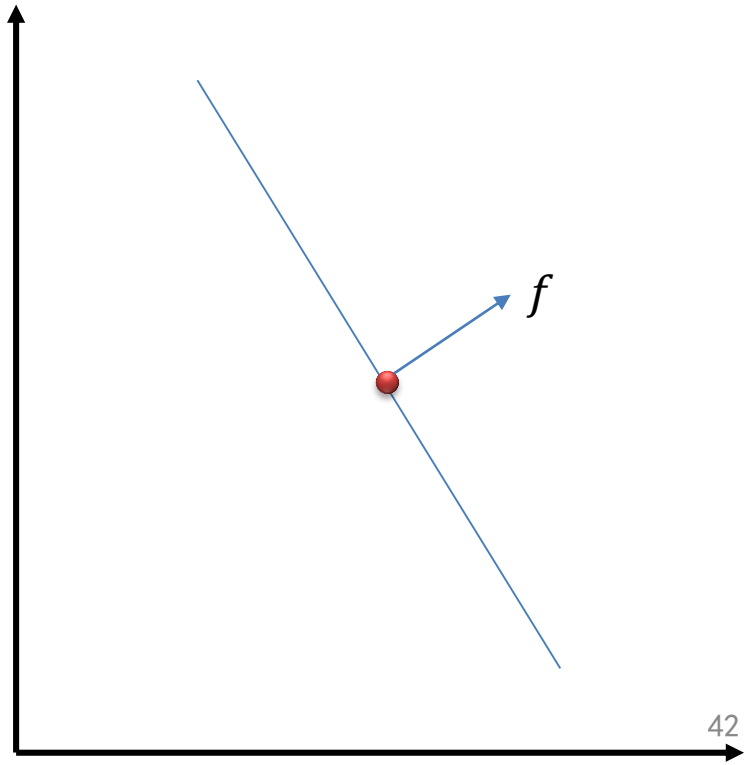
$U^TU=I,V^TV=I$: U和V均为正交阵

MIT《线性代数》公开课视频:
http://www.tudou.com/listplay/f9lC7r_8RhA/XA4jLFYnuHI.html
http://open.163.com/movie/2010/11/1/G/M6V0BQC4M_M6V2B5R1G.html

特征根的(物理)意义

- 统计上：方差
- 物理上：能量

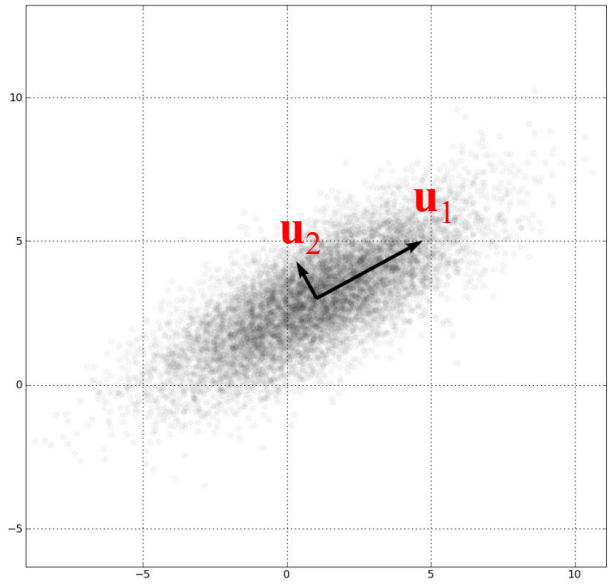




左特征向量U的意义

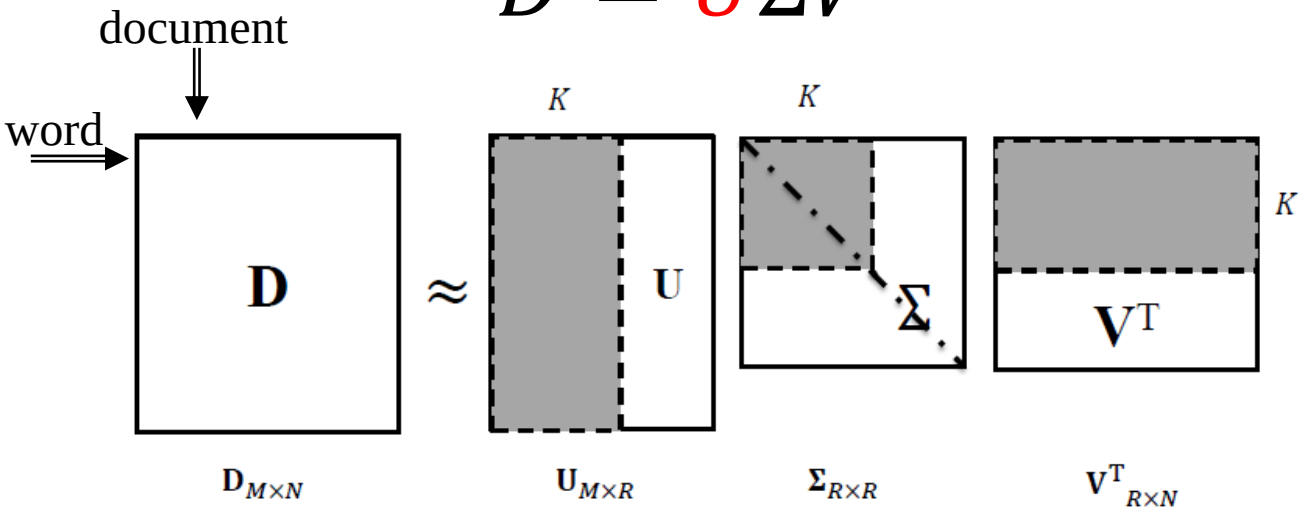
- U的每一列对应一个方向(成分)
- U中的不同列对应的方向互相垂直
- 将Σ中的特征值从大到小排序，U(和V)中的列$\mathbf{u}_i$(和行$\mathbf{v}_i$)也进行相应调整
  - $\mathbf{u}_1$对应方差(能量)最大的方向
  - $\mathbf{u}_2$对应与$\mathbf{u}_1$垂直的方差最大的方向
  - $\mathbf{u}_3$对应与$\mathbf{u}_1$ 和 $\mathbf{u}_2$垂直的方差最大的方向
  - ……

$$D = U\Sigma V^T$$



左特征向量U的意义：举例

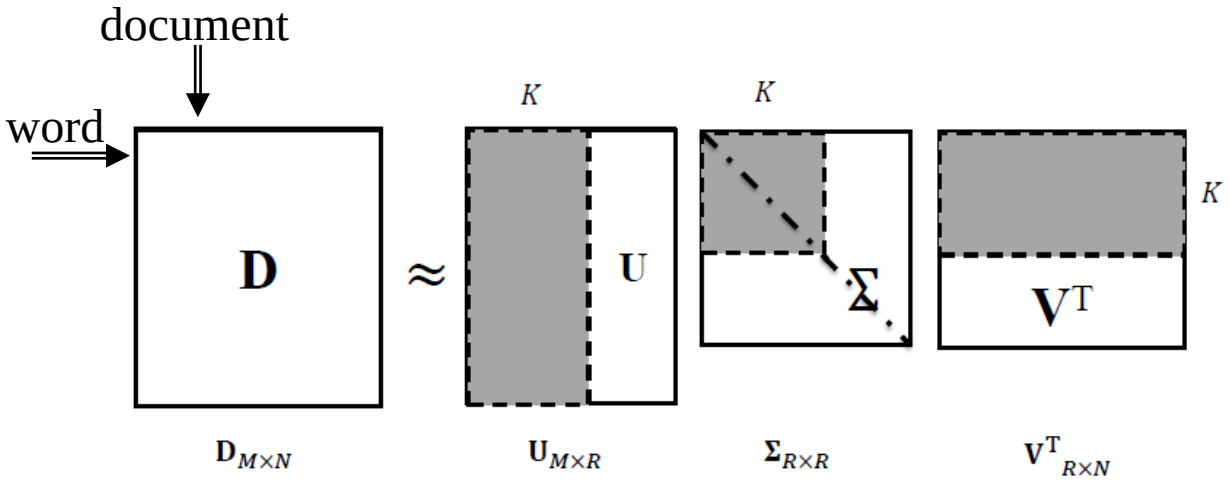
$$D = U\Sigma V^T$$



- D: 每一个文档由所有的M个单词进行表示，每一个维度表示此单词是否在文档中出现
- $\mathbf{u}_i$为M维向量，每一个维度对应一个单词
- $\mathbf{u}_1$对应由所有单词张成的空间中，方差最大的方向
- $\mathbf{u}_2$对应与$\mathbf{u}_1$垂直的所有方向中，方差最大的方向

右特征向量V的意义:举例

- $D = U\Sigma V^T \Rightarrow D^T = V\Sigma U^T$

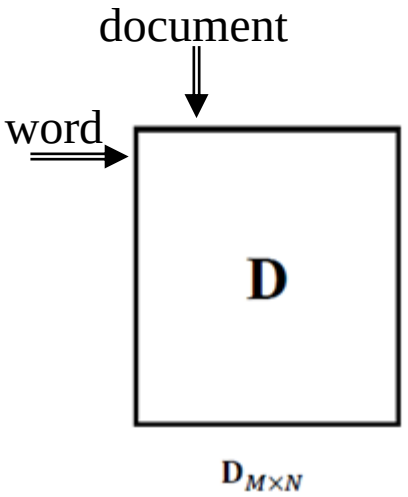


- 从另外一个角度看待被分解矩阵D:每一个单词由所有的N个文档进行表示，每一个维度表示此文档中是否出现该单词
- $\mathbf{v}_i$为N维向量，每一个维度对应一个文档
- $\mathbf{v}_1$对应由所有文档张成的空间中，方差最大的方向
- $\mathbf{v}_2$对应与$\mathbf{v}_1$垂直的所有方向中，方差最大的方向

SVD分解与主成分分析(PCA)

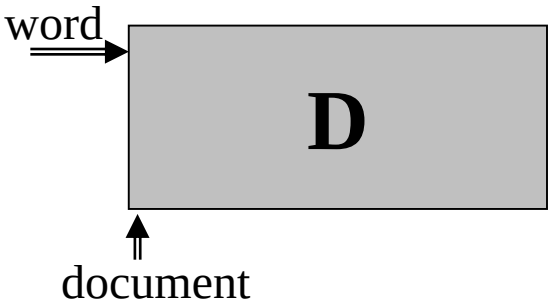
$$D = U\Sigma V^T$$

- PCA为SVD分解的一个应用
  - 只在单词张开的空间中看矩阵D
  - 给定矩阵D，$S_{M\times M} = D \times D^T$
  - $$S_{M\times M} = U\Sigma V^T \times (U\Sigma V^T)^T$$
    $$= U\Sigma V^T \times V\Sigma U^T$$
    $$= U(\Sigma)^2 U^T$$
  - $$(\Sigma)^2 = \begin{pmatrix} \sigma_1^2 & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & \sigma_R^2 \end{pmatrix}$$
  - U中对应σ^2最大的向量为主成分



LSI: SVD分解在文本数据上的应用

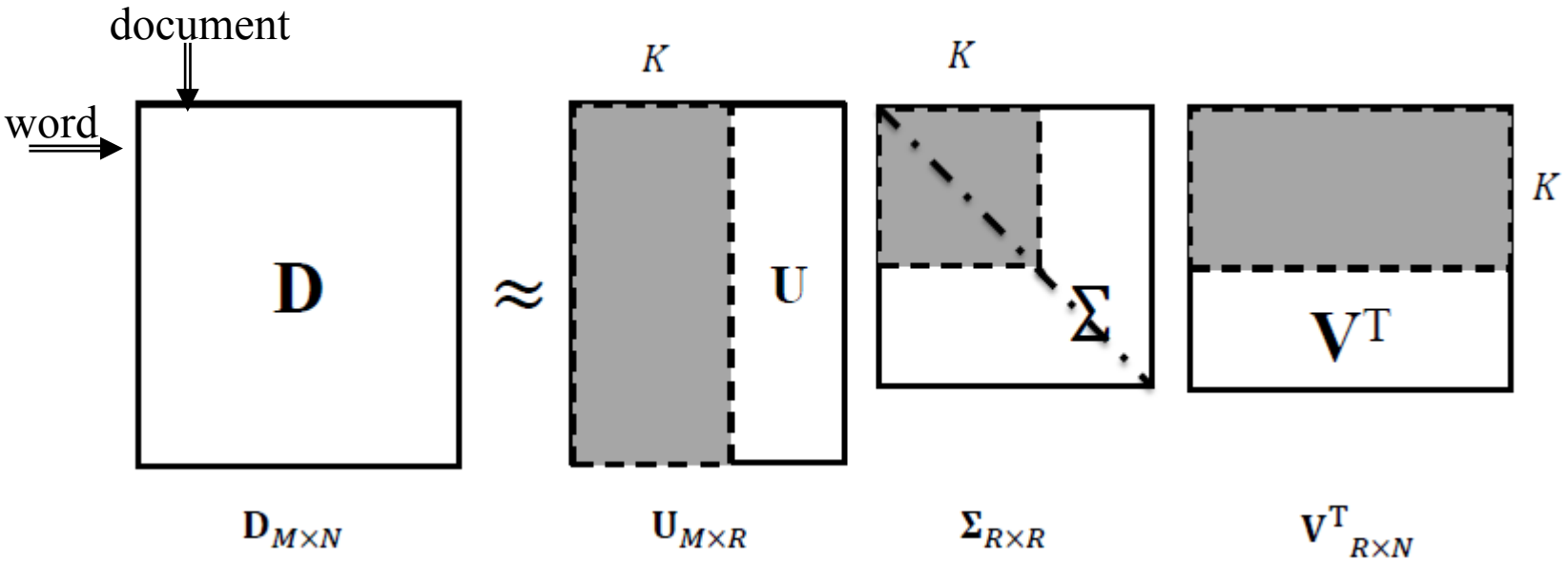
- LSI(latent semantic indexing)是常用的文本分析话题模型
- 在词袋(Bag of words)假设下，文档集合表示为文档-词的矩阵
  - d_{ij}:第i个词在第j个文档中是否出现(0-1)/出现次数(tf)/tf-idf
  - 忽略了词出现的位置和相对顺序



LSI

Deerwester et al., Indexing by Latent Semantic Analysis. 1990.

- 对D进行SVD分解，将特征值矩阵Σ按照从大到小排序，保留K个最大的特征值，其余置0,得到Σ_K
- $U\Sigma_K V^T$为D的一个低秩近似
  - $\text{rank}(U\Sigma_K V^T) = K \leq \text{rank}(D)$
  - K为隐空间维度(话题数目）



LSI

- 矩阵$U_{\mathrm{M}\times K}=[\mathbf{u}_1,\mathbf{u}_2,\cdots,\mathbf{u}_K]$(由于对应的特征根$\sigma_i$为0，其余$N-K$个特征向量对计算不造成影响)
  - $\mathbf{u}_k$:第k个话题，由带权的词向量组成

Topic1	Topic2	Topic3	Topic4	Topic5	Topic6	Topic7	Topic8	Topic9	Topic10
OPEC	Africa	contra	school	Noriega	firefight	plane	Saturday	Iran	senate
oil	South	Sandinista	student	Panama	ACR	crash	coastal	Iranian	Reagan
cent	African	rebel	teacher	Panamanian	forest	flight	estimate	Iraq	billion
barrel	Angola	Nicaragua	education	Delval	park	air	western	hostage	budget
price	apartheid	Nicaraguan	college	canal	blaze	airline	Minsch	Iraqi	Trade

- 矩阵$V=[\mathbf{v}_1^T,\cdots,\mathbf{v}_N^T]$
  - $\mathbf{v}_n^T$:由话题进行表达的第n个文档
- LSI的原理：发现词-文档矩阵中的低秩结构
  - 语义相关的词倾向共现在同一个文档中

LSI执行步骤

- 构建词-文档矩阵(通常为稀疏矩阵，一个文档中出现的词数有限)
- 执行Rank-reduced SVD
  - 前K个之后的特征值被置为零
  - 有快速算法，从大到小依次生成K个特征值和特征向量
- 输出特征向量，形成话题和用话题表示的文档

举例

Technical Memo Titles

c1: *Human* machine *interface* for ABC *computer* applications

c2: A *survey* of *user* opinion of *computer system response time*

c3: The *EPS user interface* management *system*

c4: *System* and *human system* engineering testing of *EPS*

c5: Relation of *user* perceived *response time* to error measurement

m1: The generation of random, binary, ordered *trees*

m2: The intersection *graph* of paths in *trees*

m3: *Graph minors* IV: Widths of *trees* and well-quasi-ordering

m4: *Graph minors*: A *survey*

举例

document ⇓

D

word ⟹

	c1	c2	c3	c4	c5	m1	m2	m3	m4
human	1	0	0	1	0	0	0	0	0
interface	1	0	1	0	0	0	0	0	0
computer	1	1	0	0	0	0	0	0	0
user	0	1	1	0	1	0	0	0	0
system	0	1	1	2	0	0	0	0	0
response	0	1	0	0	1	0	0	0	0
time	0	1	0	0	1	0	0	0	0
EPS	0	0	1	1	0	0	0	0	0
survey	0	1	0	0	0	0	0	0	1
trees	0	0	0	0	0	1	1	1	0
graph	0	0	0	0	0	0	1	1	1
minors	0	0	0	0	0	0	0	1	1

举例

$$D = U\Sigma V^T$$

$U =$ (latent →, word ⟹)

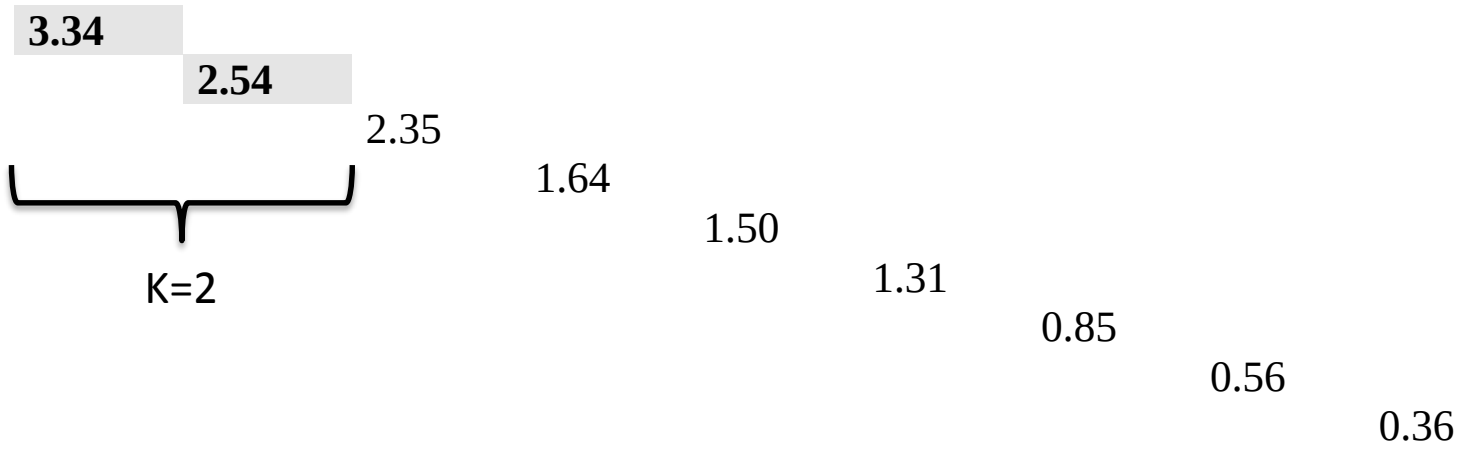
0.22	**-0.11**	0.29	-0.41	-0.11	-0.34	0.52	-0.06	-0.41
0.20	**-0.07**	0.14	-0.55	0.28	0.50	-0.07	-0.01	-0.11
0.24	**0.04**	-0.16	-0.59	-0.11	-0.25	-0.30	0.06	0.49
0.40	**0.06**	-0.34	0.10	0.33	0.38	0.00	0.00	0.01
0.64	**-0.17**	0.36	0.33	-0.16	-0.21	-0.17	0.03	0.27
0.27	**0.11**	-0.43	0.07	0.08	-0.17	0.28	-0.02	-0.05
0.27	**0.11**	-0.43	0.07	0.08	-0.17	0.28	-0.02	-0.05
0.30	**-0.14**	0.33	0.19	0.11	0.27	0.03	-0.02	-0.17
0.21	**0.27**	-0.18	-0.03	-0.54	0.08	-0.47	-0.04	-0.58
0.01	**0.49**	0.23	0.03	0.59	-0.39	-0.29	0.25	-0.23
0.04	**0.62**	0.22	0.00	-0.07	0.11	0.16	-0.68	0.23
0.03	**0.45**	0.14	-0.01	-0.30	0.28	0.34	0.68	0.18

K=2

举例

$$D = U\Sigma V^T$$

Σ=



举例

$$D = U\Sigma V^T$$

latent

$V=$ ↓

document →

0.20	**0.61**	0.46	0.54	0.28	0.00	0.01	0.02	0.08
-0.06	**0.17**	-0.13	-0.23	0.11	0.19	0.44	0.62	0.53
0.11	**-0.50**	0.21	0.57	-0.51	0.10	0.19	0.25	0.08
-0.95	**-0.03**	0.04	0.27	0.15	0.02	0.02	0.01	-0.03
0.05	**-0.21**	0.38	-0.21	0.33	0.39	0.35	0.15	-0.60
-0.08	**-0.26**	0.72	-0.37	0.03	-0.30	-0.21	0.00	0.36
0.18	**-0.43**	-0.24	0.26	0.67	-0.34	-0.15	0.25	0.04
-0.01	**0.05**	0.01	-0.02	-0.06	0.45	-0.76	0.45	-0.07
-0.06	**0.24**	0.02	-0.08	-0.26	-0.62	0.02	0.52	-0.45

K=2

举例

$$U\Sigma_{\mathrm{K}}V^{T}$$

	c1	c2	c3	c4	c5	m1	m2	m3	m4
human	**0.16**	**0.40**	**0.38**	**0.47**	**0.18**	**-0.05**	**-0.12**	**-0.16**	**-0.09**
interface	0.14	0.37	0.33	0.40	0.16	-0.03	-0.07	-0.10	-0.04
computer	0.15	0.51	0.36	0.41	0.24	0.02	0.06	0.09	0.12
user	**0.26**	**0.84**	**0.61**	**0.70**	**0.39**	**0.03**	**0.08**	**0.12**	**0.19**
system	0.45	1.23	1.05	1.27	0.56	-0.07	-0.15	-0.21	-0.05
response	0.16	0.58	0.38	0.42	0.28	0.06	0.13	0.19	0.22
time	0.16	0.58	0.38	0.42	0.28	0.06	0.13	0.19	0.22
EPS	0.22	0.55	0.51	0.63	0.24	-0.07	-0.14	-0.20	-0.11
survey	0.10	0.53	0.23	0.21	0.27	0.14	0.31	0.44	**0.42**
trees	-0.06	0.23	-0.14	-0.27	0.14	0.24	0.55	0.77	**0.66**
graph	-0.06	0.34	-0.15	-0.30	0.20	0.31	0.69	0.98	0.85
minors	**-0.04**	**0.25**	**-0.10**	**-0.21**	**0.15**	**0.22**	**0.50**	**0.71**	**0.62**

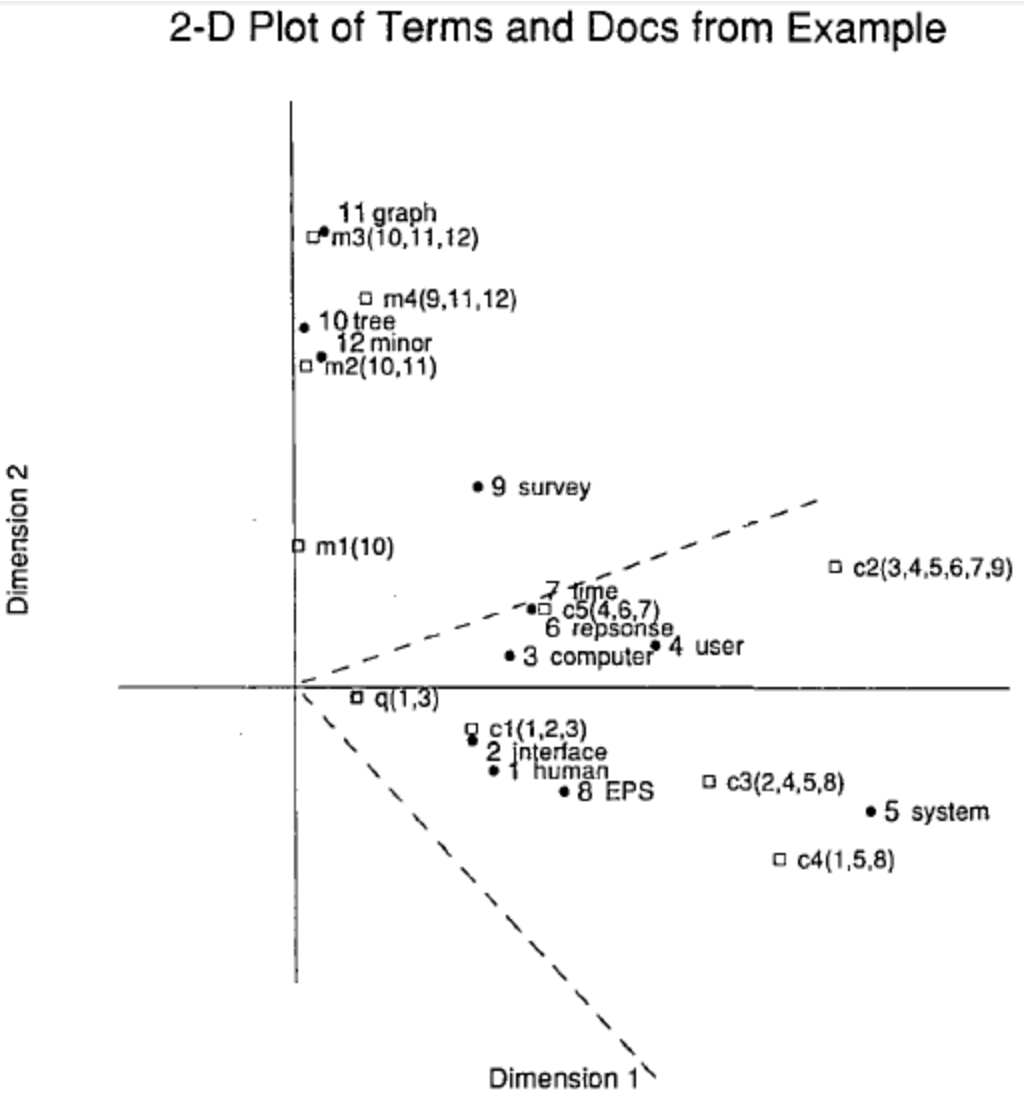


FIG. 1. A two-dimensional plot of 12 Terms and 9 Documents from the sampe TM set. Terms are represented by filled circles. Documents are shown as open squares, and component terms are indicated parenthetically. The query ("human computer interaction") is represented as a pseudo-document at point q. Axes are scaled for Document-Document or Term-Term comparisons. The dotted cone represents the region whose points are within a cosine of .9 from the query q. All documents about human-computer (c1–c5) are "near" the query (i.e., within this cone), but none of the graph theory documents (m1–m4) are nearby. In this reduced space, even documents c3 and c5 which share no terms with the query are near it.

LSI的总结

- 对单词-文档矩阵进行SVD分解，保留最大的K个特征值，其余置0
- 是对原始输入矩阵的低秩近似
- 优点
  - 数学优美
  - 有效
- 缺点
  - 低秩矩阵和U、V中有负值，不好理解
  - 特征向量两两正交，计算难度大

非负矩阵分解

- 非负矩阵分解(nonnegative matrix factorization, NMF)

$$D \approx UV^T$$

  - 输入矩阵D所有的数值非负
  - 输出矩阵U和V的值也非负
  - 不再要求U（和V）为正交矩阵

D. D. Lee and H. S. Seung. Learning the parts of objects by non-negative matrix factorization. Nature, 401(6755):788-791, October 1999

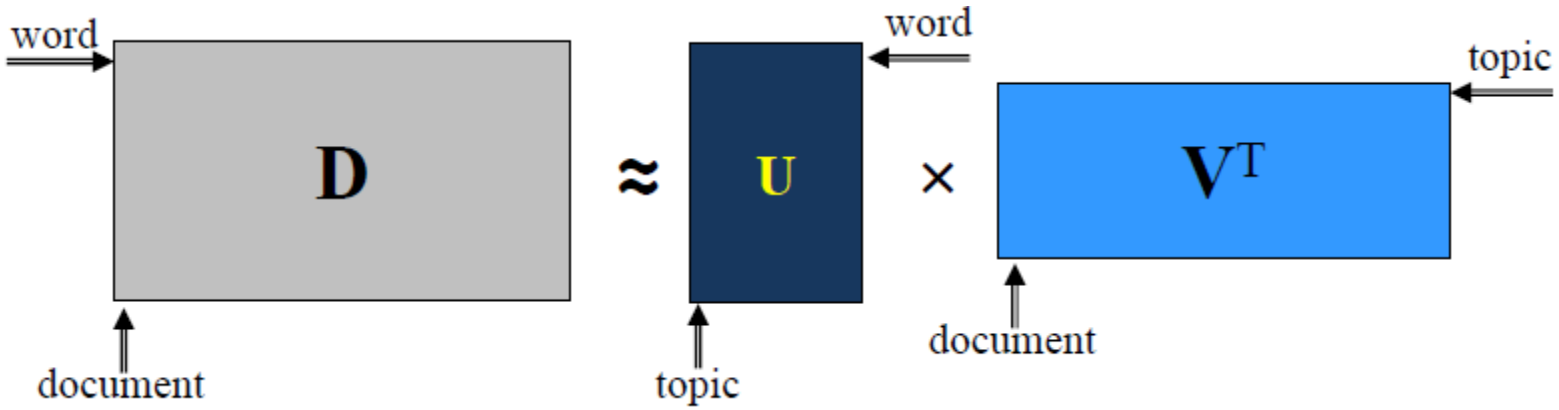
非负矩阵分解算法

- 优化目标

$$\min_{U,V} |D - U \times V^T|^2$$

$$\text{s.t. } u_{ij} \geq 0; v_{ij} \geq 0$$

- $|D - U \times V^T|^2 = \sum_{i=1}^{M} \sum_{j=1}^{N} \left(d_{ij} - \mathbf{u}_i \mathbf{v}_j^T\right)^2$



优化算法

- 目标函数非凸，无全局最优解
- 在固定U（或者V）后，目标函数对V（或者U）是凸函数
- 交替优化
  - 1. 随机对U赋值
  - 2. 固定U，最优化V
  - 3. 固定V，最优化U
  - 4. 重复2、3，直至收敛

NMF算法

- 输入：$D_{M\times N}$, 隐空间维度K
- 输出： U,V
- 1. U ← random nonnegative values
- 2. **repeat**
- 3. **foreach** $v_{ij} \in V$

Theorem 1 *The Euclidean distance* $||V - WH||$ *is nonincreasing under the update rules*

$$H_{a\mu} \leftarrow H_{a\mu}\frac{(W^TV)_{a\mu}}{(W^TWH)_{a\mu}} \qquad W_{ia} \leftarrow W_{ia}\frac{(VH^T)_{ia}}{(WHH^T)_{ia}} \tag{4}$$

The Euclidean distance is invariant under these updates if and only if W *and* H *are at a stationary point of the distance.*

$(DV^TV)_{ij}$

- 8. **end for**
- 9. **until** converge
- 10. **return** U, V

NMF应用(图像)

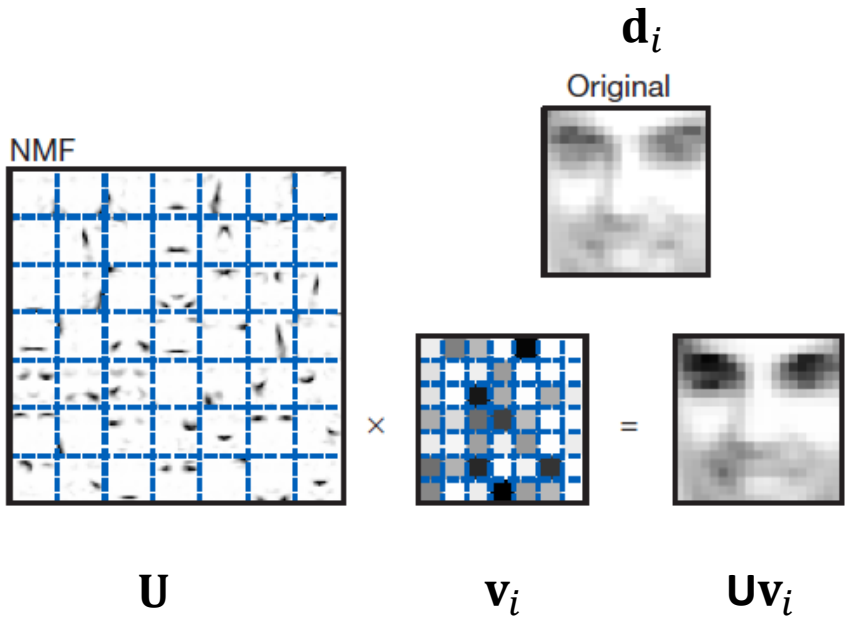


Figure 1 Non-negative matrix factorization (NMF) learns a parts-based representation of faces, whereas vector quantization (VQ) and principal components analysis (PCA) learn holistic representations. The three learning methods were applied to a database of $m = 2{,}429$ facial images, each consisting of $n = 19 \times 19$ pixels, and constituting an $n \times m$ matrix V. All three find approximate factorizations of the form $V \approx WH$, but with three different types of constraints on W and H, as described more fully in the main text and methods. As shown in the 7×7 montages, each method has learned a set of $r = 49$ basis images. Positive values are illustrated with black pixels and negative values with red pixels. A particular instance of a face, shown at top right, is approximately represented by a linear superposition of basis images. The coefficients of the linear superposition are shown next to each montage, in a 7×7 grid, and the resulting superpositions are shown on the other side of the equality sign. Unlike VQ and PCA, NMF learns to represent faces with a set of basis images resembling parts of faces.

NMF应用(文本)

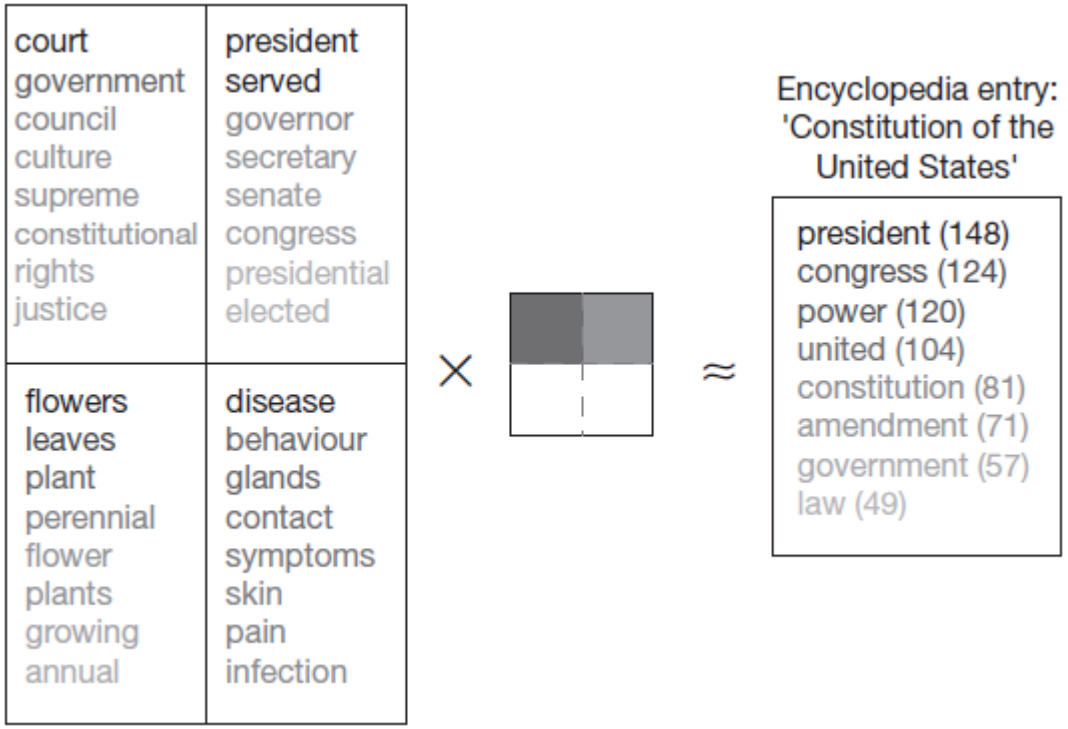


Figure 4 Non-negative matrix factorization (NMF) discovers semantic features of $m = 30{,}991$ articles from the Grolier encyclopedia. For each word in a vocabulary of size $n = 15{,}276$, the number of occurrences was counted in each article and used to form the $15{,}276 \times 30{,}991$ matrix V. Each column of V contained the word counts for a particular article, whereas each row of V contained the counts of a particular word in different articles. The matrix was approximately factorized into the form WH using the algorithm described in Fig. 2. Upper left, four of the $r = 200$ semantic features (columns of W). As they are very high-dimensional vectors, each semantic feature is represented by a list of the eight words with highest frequency in that feature. The darkness of the text indicates the relative frequency of each word within a feature. Right, the eight most frequent words and their counts in the encyclopedia entry on the 'Constitution of the United States'. This word count vector was approximated by a superposition that gave high weight to the upper two semantic features, and none to the lower two, as shown by the four shaded squares in the middle indicating the activities of H. The bottom of the figure exhibits the two semantic features containing 'lead' with high frequencies. Judging from the other words in the features, two different meanings of 'lead' are differentiated by NMF.

$\mathbf{U}$ $\mathbf{v}_i$ $\mathbf{U}\mathbf{v}_i$

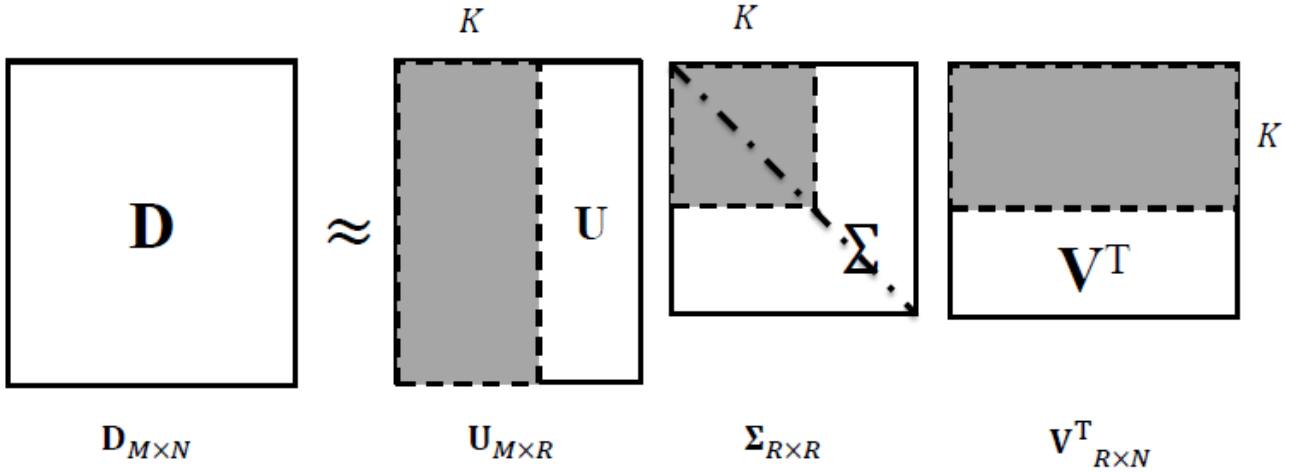
又称为
dictionary

NMF小结

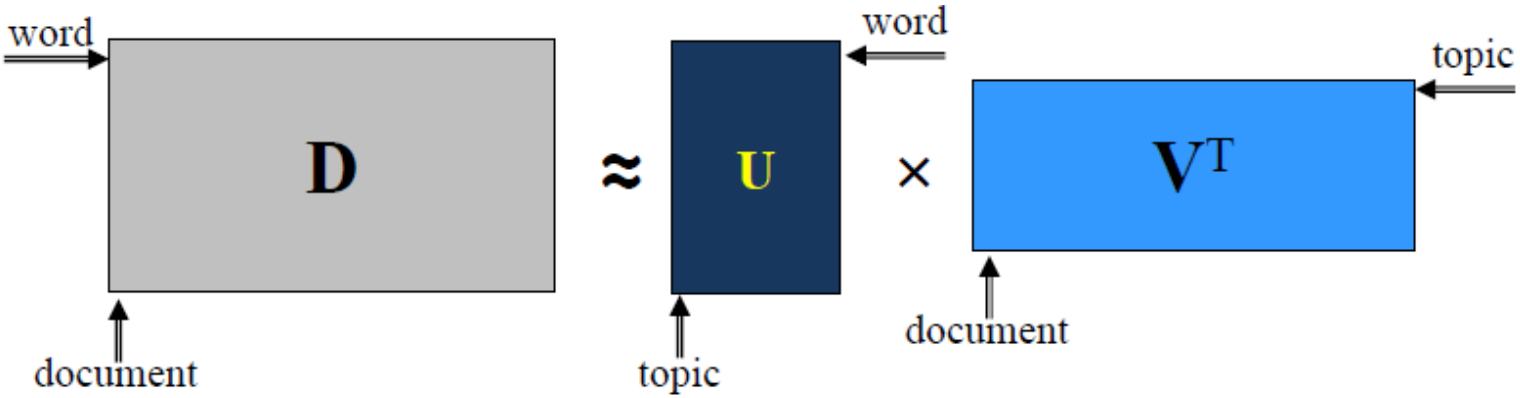
- 要求输入/输出矩阵所有的元素都是非负
  - 很大一部分文本分析任务满足
- 优化二次损失函数
  - 无全局最优解
- U和V矩阵可解释
  - U：学习到的字典(dictionary)
  - V：在学习到的字典意义下，原始数据的表达

矩阵分解中的低秩近似

- LSI



- NMF



低秩近似的直观理解

- 低秩:一些行/列可以表达为其它行/列的线性组合
  - Rank=0:所有的元素为0
- 低秩意味着局部平滑(local smoothness)
  - 在文本中，平滑指文档比较相似
    - 如果一个词在一些文档中出现过，它也会在其他文档中出现（将稀疏的词-文档矩阵变成稠密矩阵）
    - 原始矩阵中：d1中出现w1、w2、w3；d2中只出现了w1、w2。低秩近似倾向认为w3也应该在d2中出现。
  - 在推荐系统中，平滑指用户购买兴趣的相似
    - 原始矩阵中：u1购买过牛奶、面包、火腿肠；u2购买过牛奶、面包。低秩近似倾向认为u2也会买火腿肠。

1	**0**	**3**
2	0	6
3	0	9

$rank = 1$

提纲

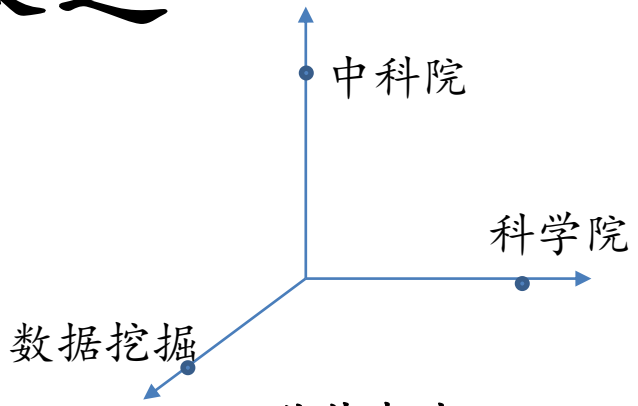
- 背景介绍
- 常用无监督学习算法
  - 聚类算法
  - 矩阵分解/话题模型
  - 文本表达
- 总结

单词的分布式表达

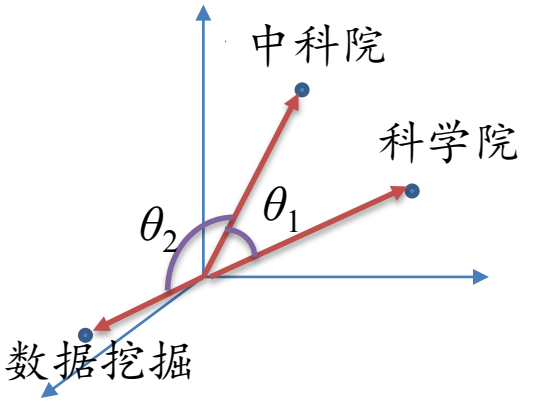
- 问题定义：给定文档集合，将文档中的单词依据其**语义**表达为一个向量
- 单词表达方式
  - 独热表达：每一个词用不同的ID进行表示，可表示为其中一个维度为1，其它全0的向量

    科学院:[1，0，0]，中科院:[0，1，0]，数据挖掘:[0，0，1]
  - 分布式表达:每一个词用一般的向量表示

    科学院:[1.0，0.5，0]，中科院:[0.5，1.0，0]，数据挖掘[0，0.1，1.0]
  - 分布式表达优势：建模单词间语义关系（相似度）



独热表达



分布式表达

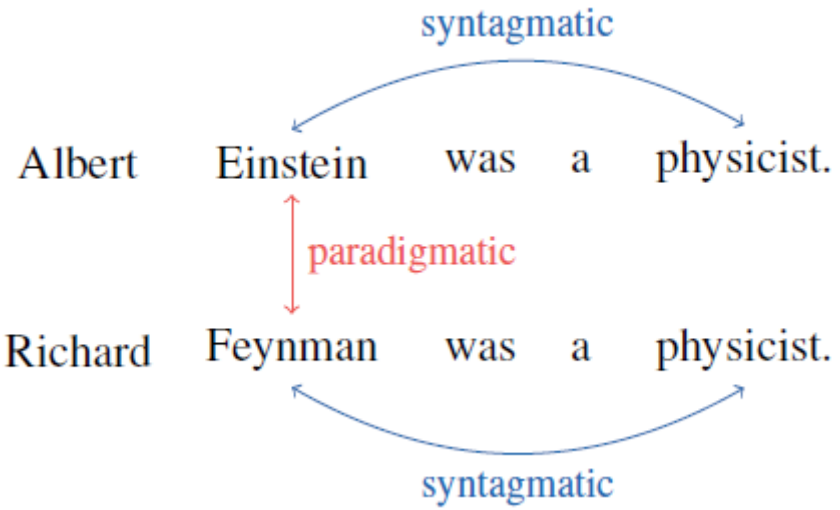
分布式假设
(The distributional hypothesis)

"You shall know a word by the company it keeps."

—J.R. Firth

- 一个单词的语义可以通过与其共现的单词得知
  - 不考虑单词的构词法，将单词看成无语义的ID（独热表达）
  - 通过一个大规模文本文档集合(单词共现在文档中)可以挖掘各个单词间的语义关系，对单词进行重新表达（分布式表达）

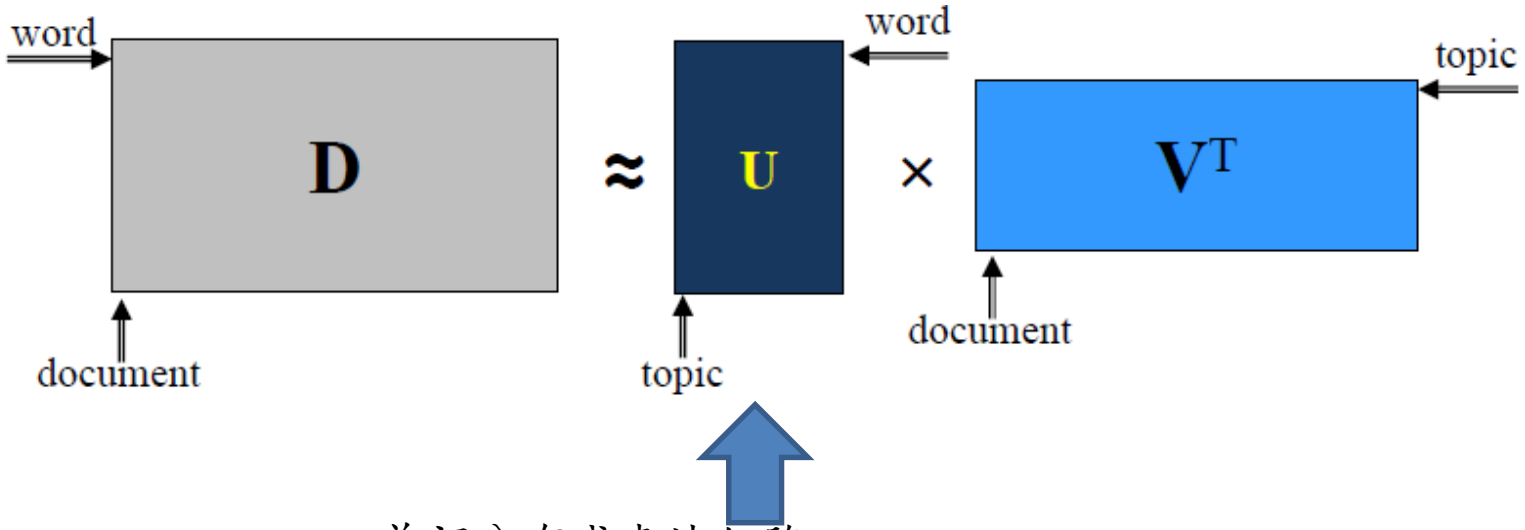
分布式假设的两种解释



- Syntagmatic: 两个单词频繁在文档中共现，则它们具有一定的语义相似度(Einstein ~ physicist)
- Paradigmatic:两个单词频繁在相似的上下文中共现，则它们具有一定的语义相似度 (Einstein ~ Feynman)

利用Syntagmatic假设学习单词的分布式表达

- 分解文档-单词矩阵，如LSI、NMF等



思考：业界普遍认为话题模型（如：LSI、NMF等）应用到短文本集合上（如：微博数据）效果不好，为什么？

利用Paradigmatic假设学习单词的分布式表达(Word2Vec)

Distributed Representations of Words and Phrases and their Compositionality

Tomas Mikolov
Google Inc.
Mountain View
mikolov@google.com

Ilya Sutskever
Google Inc.
Mountain View
ilyasu@google.com

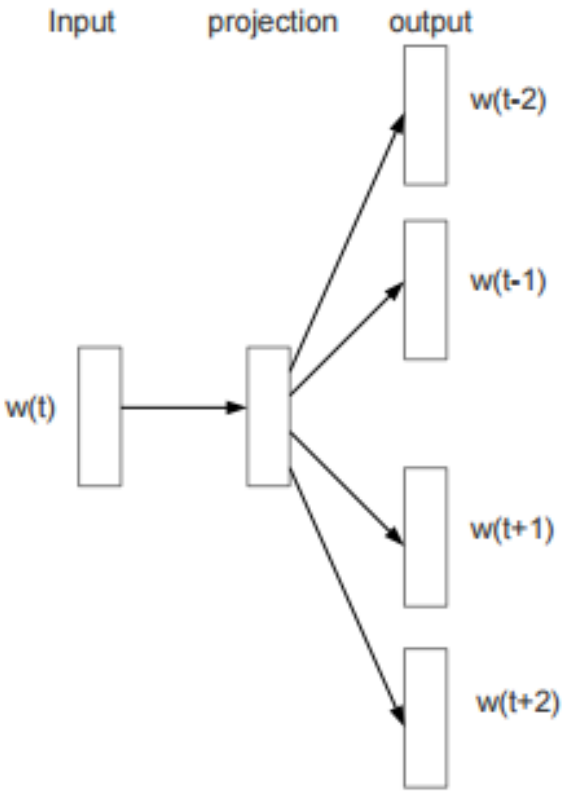
Kai Chen
Google Inc.
Mountain View
kai@google.com

Greg Corrado
Google Inc.
Mountain View
gcorrado@google.com

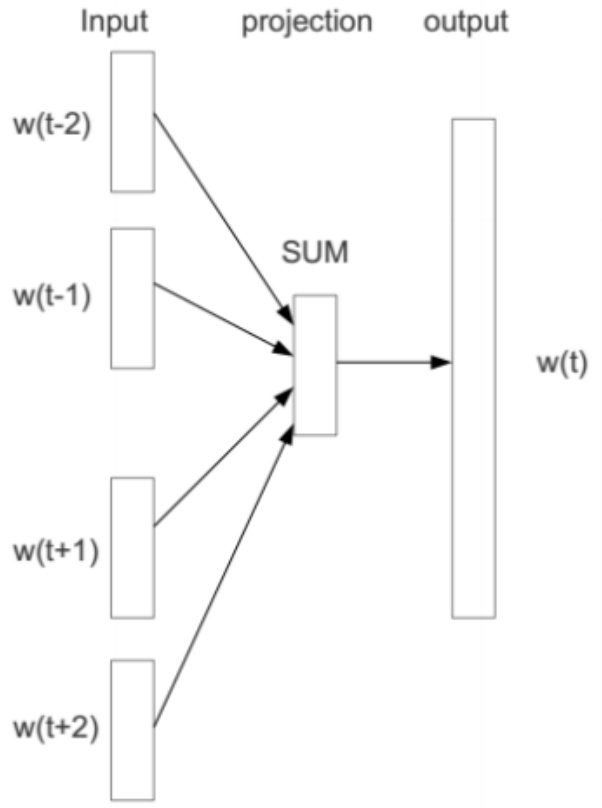
Jeffrey Dean
Google Inc.
Mountain View
jeff@google.com

- 核心思想：一个单词的表达可以通过在它周围出现的单词计算出来（分布式假设）
- 实现方式
  - 1. 初始化：为每一个单词随机生成(固定维度的)表达
  - 2. 更新：扫描文档集合，为每一个单词重新计算其表达
  - 3. 重复2，直至收敛

两种模型架构



Skip-Gram



Continuous bag-of-words (CBOW)

以Skip-Gram为例

优化目标构建：扫描文本数据集合，对遇到的每一个单词，通过当前单词预测从-c到c的窗口中的单词的概率

$$\frac{1}{T}\sum_{t=1}^{T}\sum_{-c\le j\le c, j\neq 0}\log p(w_{t+j}|w_t)$$

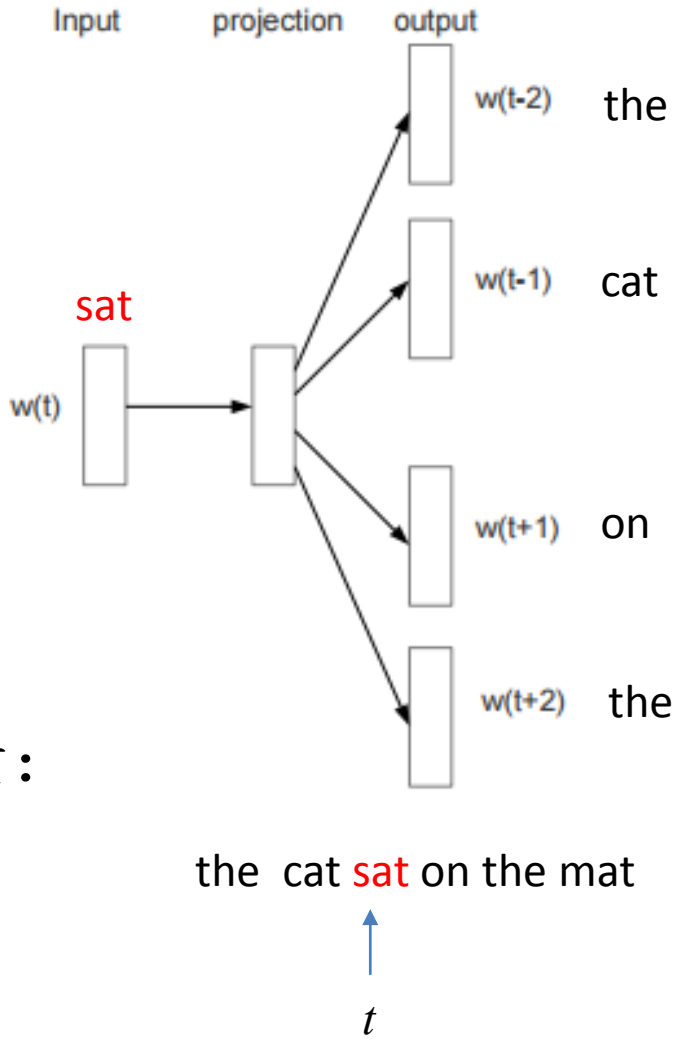
$$p(w_O|w_I)=\frac{\exp\left({v'_{w_O}}^{\top}v_{w_I}\right)}{\sum_{w=1}^{W}\exp\left({v'_w}^{\top}v_{w_I}\right)}$$

其中每一个单词对应两个K维表达向量：

v_{wI}为K维“输入”向量

v'_{wo}和v'_w为K维“输出”向量

注意：与NMF或者LSI相比，没有利用文档边界。



Word2Vec与NMF/LSI的关系

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. GloVe: Global Vectors for Word Representation. EMNLP 2014.

Albert Einstein was a physicist.

Richard Feynman was a physicist.

	Einstein	Feynman	physicist
Einstein	0	0	1
Feynman	0	0	1
physicist	1	1	0

	d_1	d_2
Einstein	1	0
Feynman	0	1
physicist	1	1

Word2vec分解Word-Word矩阵，建模单词间的上下文共现

LSI/NMF分解Word-doc矩阵，建模单词在文档中的共现

提纲

- 背景介绍
- 常用无监督学习算法
  - 聚类算法
  - 矩阵分解/话题模型
  - 文本表达
- 总结

本次课程小结

- 非监督学习
  - 聚类算法：将数据按照其相似度/距离划分为不同的簇
  - 矩阵分解/话题模型：将输入矩阵分解为多个矩阵的乘积，得到其“低秩”近似
  - 文本表达：利用单词的共现信息构建单词语义表达
- 矩阵分解/话题模型也可以看出“软”聚类
  - 每一个话题/隐维度看成一个簇

其他重要的非监督学习模型

- 稀疏编码(sparse coding)和字典学习(dictionary learning)
- 概率话题模型
  - Probabilistic Latent Semantic Indexing (PLSI)
  - Latent Dirchilet Allocation (LDA)
- 深度学习中的非监督学习模型
  - Restricted Boltzmann Machine (RBM)

作业

- 精读论文

  Daniel D. Lee and H. Sebastian Seung: Algorithms for Non-negative Matrix Factorization. NIPS 2000. (NIPS Classic Paper Award)，自己动手推导multiplicative update rules

或者

J H Friedman. Greedy Function Approximation: A Gradient Boosting Machine. 1999.

或者

Ioannis Tsochantaridis et al., Large Margin Methods for Structured and Interdependent Output Variables. JMLR 2005.

- 假设你需要在实验室进行汇报，写20页左右的幻灯片汇报论文
  - 背景、动因、相关工作、主要思想、实验验证、结论
  - 网上对这两篇论文的介绍已经有很多，可以参考别人的写法，请尽量用自己的语言总结
  - 尽量少堆砌公式和大段的话语，多用图、动画、举例
  - 鼓励使用LaTeX

谢谢！