计算机视觉的统计方法与机器学习

第三课：数据聚类2

龚怡宏

西安交通大学

联系方式：ygong@mail.xjtu.edu.cn

本堂课要学习的聚类方法

基于非负矩阵因子分解的聚类算法（Non-negative Matrix Factorization)

- 单线性NMF模型（Single Linear NMF Model)
- 双线性NMF模型（ Bilinear NMF Model)

单线性NMF模型 (Single Linear Model)

- 输入：
  - 需聚类的数据集 $\mathbf{D} = (\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_n)$
  - 群的个数k，$k<<n$
- 假设：
  - 每个数据点$\mathbf{x}_i$都是一个 m 维特征向量：

$$\mathbf{x}_i = \left[x_{1i}, x_{2i}, \cdots, x_{mi}\right]^T$$

  - 每个群的中心点可表示为：

$$\mathbf{u}_k = \left[u_{1k}, u_{2k}, \cdots, u_{mk}\right]^T$$

单线性NMF模型的中心思想

- 每个群的中心 $\mathbf{u}_k$代表一种具体概念，而每一个数据点 $\mathbf{x}_i$ 是这些概念的线性组合，用数学公式表示，就是：

$$\mathbf{x}_i = v_{i1}\mathbf{u}_1 + v_{i2}\mathbf{u}_2 + \cdots + v_{ik}\mathbf{u}_k$$

- 写成矩阵形式，就有：

$$\mathbf{X}=\mathbf{U}\cdot\mathbf{V}^{\mathrm{T}}$$

- 其中$\mathbf{X}$是 $m\times n$ 的**数据集矩阵** $\mathbf{X}=[\mathbf{x}_1,\mathbf{x}_2,\cdots,\mathbf{x}_n]$
- $\mathbf{U}$是 $m\times k$ 的非负**中心点矩阵** $\mathbf{U}=[\mathbf{u}_1,\mathbf{u}_2,\cdots,\mathbf{u}_k]$
- $\mathbf{V}^{\mathrm{T}}$是 $k\times n$ 的非负矩阵 $\mathbf{V}^{\mathrm{T}}=[\mathbf{v}_1,\mathbf{v}_2,\cdots,\mathbf{v}_n]$，

$\mathbf{v}_i=[v_{i1},x_{i2},\cdots,v_{ik}]^T$

单线性NMF模型的损失函数及求解算法

- **损失函数：** $J = \frac{1}{2} \| \mathbf{X} - \mathbf{U}\mathbf{V}^{\mathrm{T}} \|^2$ (3−1)

我们的目标是将 **X** 分解为两个非负矩阵**U**，**V**的乘积，使得上述损失函数最小。

- **求解算法：** 使用标准的非负矩阵分解算法便可求解，在此省略详细推导，最终结果如下：

$$u_{ij} \leftarrow u_{ij} \frac{(\mathbf{X}\mathbf{V})_{ij}}{(\mathbf{U}\mathbf{V}^{\mathrm{T}}\mathbf{V})_{ij}} \qquad (3-2)$$

$$v_{ij} \leftarrow v_{ij} \frac{(\mathbf{X}^{\mathrm{T}}\mathbf{U})_{ij}}{(\mathbf{V}\mathbf{U}^{\mathrm{T}}\mathbf{U})_{ij}} \qquad (3-3)$$

单线性NMF模型解的讨论

- 损失函数 J 的解并不唯一。如果**U**和**V**是 J 的解，则对任意正对角矩阵**D**来说，**UD** 和 $\mathbf{VD}^{-1}$同样是 J 的解。
- 为了使解唯一，我们归一化**U**的列向量：

$$v_{ij} \leftarrow v_{ij} \sqrt{\sum_i u_{ij}^2} \qquad (3-4)$$

$$u_{ij} \leftarrow \frac{u_{ij}}{\sqrt{\sum_i u_{ij}^2}} \qquad (3-5)$$

- 矩阵**V**的元素 v_{ij} 表示数据点 i 对群 j 的归属程度。如果数据点 i 只属于群 j，则 v_{ij} 的值将很大，而 $\mathbf{v}_i = \left[v_{i1}, v_{i2}, \cdots, v_{ik}\right]^T$ 其余元素的值则很小。

基于单线性NMF模型的聚类算法

- 为数据集 **D** 构建数据集矩阵**X**，它的第 i 列代表数据点 i 的特征向量 。
- 用非负矩阵分解（NMF）算法将 **X**分解为两个非负矩阵**U**和**V**。
- 使用式3-4和3-5来对矩阵**U**和**V**进行归一化。
- 用**V**来确定每个数据点 i 的群标签。具体来说，找出矩阵 $\mathbf{V}^T$ 的列向量 v_i 的最大元素c，$c = \arg\max_j v_{ij}$ 然后将数据点 i 分配到第c个群中。

双线性NMF模型 (Bilinear NMF Model)

在单线性NMF模型中，对聚类中心非负的要求不仅使适用场合受到约束。因此，我们提出了双线性NMF模型。

双线性NMF模型的中心思想

- 每个群的中心点 $\mathbf{r}_c$代表一种具体概念，而每一个数据点 $\mathbf{x}_i$ 是这些概念的线性组合：

$$\mathbf{x}_i = v_{i1}\mathbf{r}_1 + v_{i2}\mathbf{r}_2 + \cdots + v_{ik}\mathbf{r}_k = \sum_{c=1}^{k} v_{ic}\mathbf{r}_c \qquad (3\text{-}6)$$

- 每个群的中心点 $\mathbf{r}_c$ 是所有数据点 $\mathbf{x}_i$ 的线性组合：

$$\mathbf{r}_c = \omega_{1c}\mathbf{x}_1 + \omega_{2c}\mathbf{x}_2 + \cdots + \omega_{nc}\mathbf{x}_n = \sum_{i=1}^{n} \omega_{ic}\mathbf{x}_i \qquad (3\text{-}7)$$

- 将(3-7)式带入(3-6)式，我们得到：

$$\mathbf{x}_i = \sum_{c=1}^{k} v_{ic}\mathbf{r}_c = \sum_{c=1}^{k} v_{ic} \sum_{j=1}^{n} \mathbf{x}_j \omega_{jc} \qquad (3-9)$$

- 写成矩阵形式，我们得到：

$$\mathbf{X} \approx \mathbf{XWV}^T \qquad (3-10)$$

$\mathbf{W} = [\omega_{ic}]$是$n \times k$的关联矩阵。

双线性NMF模型的损失函数及求解算法

- **损失函数：** $J = \frac{1}{2} \| \mathbf{X} - \mathbf{X}\mathbf{W}\mathbf{V}^T \|^2$ (3−11)

我们的目标是将 $\mathbf{X}$ 分解为两个非负矩阵$\mathbf{W}$和$\mathbf{V}$，使得上述损失函数最小。

- **求解算法：**在此省略详细推导，最终结果如下：

$$\omega_{ij} \leftarrow \omega_{ij} \frac{(\mathbf{KV})_{ij} + \sqrt{(\mathbf{KV})_{ij}^2 + 4\mathbf{P}_{ij}^+ \mathbf{P}_{ij}^-}}{2\mathbf{P}_{ij}^+} \quad (3-12)$$

其中 $\mathbf{K} = \mathbf{X}^T\mathbf{X}$；$\mathbf{K}^+$和 $\mathbf{K}^-$ 是每个元素都为正数的对称矩阵，且满足 $\mathbf{K} = \mathbf{K}^+ - \mathbf{K}^-$。

另外，$\mathbf{P}^+ = \mathbf{K}^+\mathbf{W}\mathbf{V}^T\mathbf{V}$，$\mathbf{P}^- = \mathbf{K}^-\mathbf{W}\mathbf{V}^T\mathbf{V}$。

双线性NMF模型的求解算法

由于矩阵**K**的全部元素均为正，也就是说，$\mathbf{K}^{-}=\mathbf{0}$，我们对(3-12)进行简化，得：

$$\omega_{ij} \leftarrow \omega_{ij} \frac{(\mathbf{KV})_{ij}}{\mathbf{P}_{ij}} \qquad (3-13)$$

类似的，我们也可以求得矩阵**V**的迭代式以及简化式：

$$v_{ij} \leftarrow v_{ij} \frac{(\mathbf{KW})_{ij}+\sqrt{(\mathbf{KW})_{ij}^{2}+4\mathbf{Q}_{ij}^{+}\mathbf{Q}_{ij}^{-}}}{2\mathbf{Q}_{ij}^{+}} \qquad (3-14)$$

$$v_{ij} \leftarrow v_{ij} \frac{(\mathbf{KW})_{ij}}{\mathbf{Q}_{ij}^{+}} \qquad (3-15)$$

其中 $\mathbf{Q}^{+}=\mathbf{V}\mathbf{W}^{T}\mathbf{K}^{+}\mathbf{W}$, $\mathbf{Q}^{-}=\mathbf{V}\mathbf{W}^{T}\mathbf{K}^{-}\mathbf{W}$

双线性NMF模型的求解算法

与单线性模型类似，由于最小化损失函数 J 的解并不唯一，我们需要对$\mathbf{W}$和$\mathbf{V}$进行归一化，加入约束 $\|\mathbf{r}_c\|=1$之后，得标准化公式：

$$\mathbf{V} \leftarrow \mathbf{V}\left[\operatorname{diag}(\mathbf{W}^T\mathbf{K}\mathbf{W})\right]^{1/2} \tag{3-16}$$

$$\mathbf{W} \leftarrow \mathbf{W}\left[\operatorname{diag}(\mathbf{W}^T\mathbf{K}\mathbf{W})\right]^{-1/2} \tag{3-17}$$

核函数的导入

- 上述算法中，$\mathbf{K} = \mathbf{X}^T\mathbf{X}$，$k_{ij} = \mathbf{x}_i \cdot \mathbf{x}_j$
- 除了这种标准的核函数之外，我们还可定义任意的核函数：

$$k_{ij} = \kappa(\mathbf{x}_i, \mathbf{x}_j)$$

- 可以使用任意的核函数这一性质可以显著提高双线性模型的聚类精度。

基于双线性NMF模型的聚类算法

1. 为数据集 **D** 构建数据集矩阵 **X**，其第 i 列代表数据点 i 的特征向量。
2. 使用适当的核函数 $\kappa(\mathbf{x}_i, \mathbf{x}_j)$构造核矩阵**K**。
3. 通过式3-12与3-14交替迭代，求得损失函数 J 的最优解。
4. 利用式3-16和3-17对**W**和**V**进行归一化。
5. 用 $\mathbf{V}^{\mathrm{T}} = [\mathbf{v}_1, \mathbf{v}_2, \cdots, \mathbf{v}_n]$ 来确定每个数据点的群标签。具体来说，找出矩阵每个列向量 $\mathbf{v}_i$ 的最大要素，$x = \arg\max_c v_{ic}$，将数据点 i 分配至第 x 个群中。

谱聚类 vs. NMF聚类

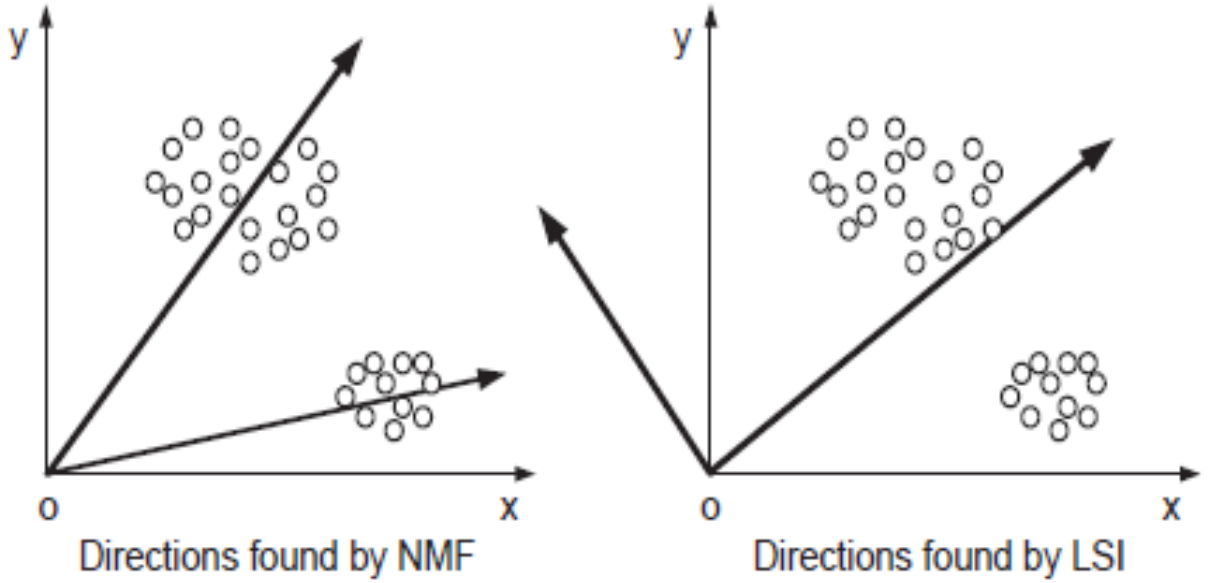
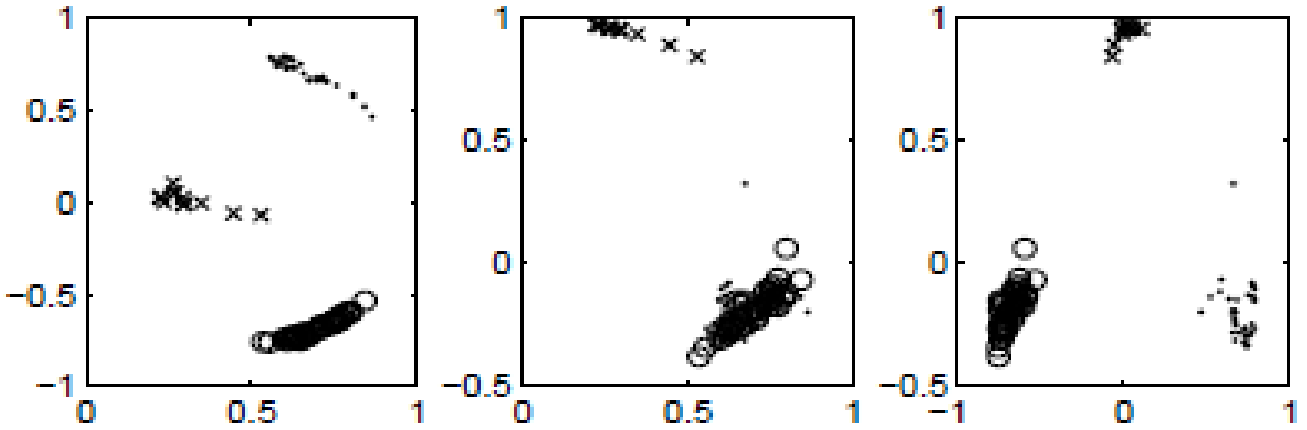
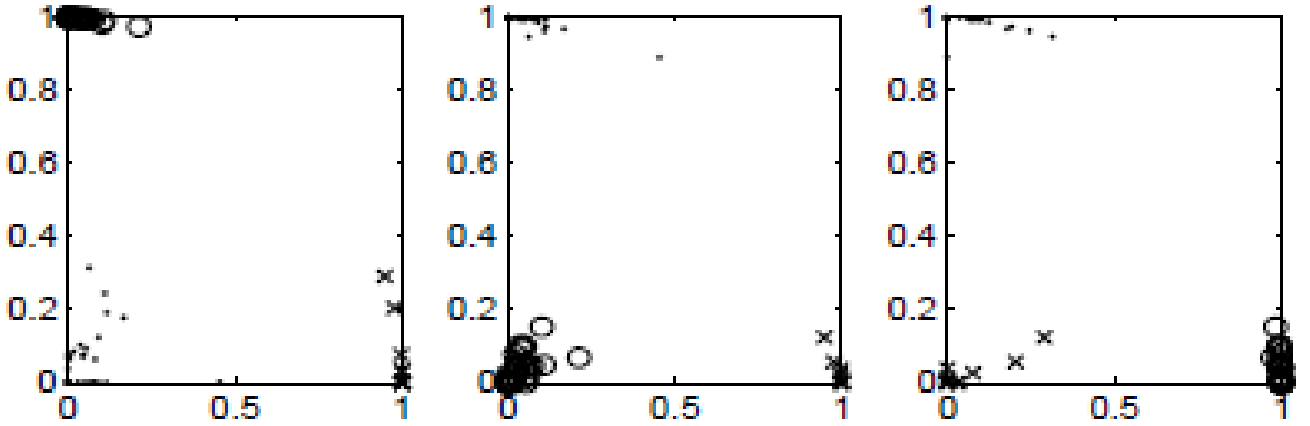


Fig. 3.5. Illustration of the differences between Spectral and NMF Clustering techniques

谱聚类 vs. NMF聚类



(a) Data distributions in the subspaces $\mathbf{u}_1$–$\mathbf{u}_2$, $\mathbf{u}_1$–$\mathbf{u}_3$, and $\mathbf{u}_2$–$\mathbf{u}_3$ derived by the Normalized Cut



(b) Data distributions in the subspaces $\mathbf{v}_1$–$\mathbf{v}_2$, $\mathbf{v}_1$–$\mathbf{v}_3$, and $\mathbf{v}_2$–$\mathbf{v}_3$ derived by the bilinear NMF model

谱聚类及NMF聚类实例学习

一、文档分析基础

二、文档集

三、评价指标

四、性能评估和比较

一、文档分析基础

文档的词频向量 (Term-frequency vector)

- 文档 i 一般用词频向量 $\mathbf{x}_i = [x_{1i}, x_{2i}, \cdots, x_{ni}]$ 表示，n 代表文档全集所包含关键词的总数，x_{ji} 代表第 j 个关键词
- 文档的关键词，是 ming这两项预处理后留下
  - stop-words
  - stemming

stop-words是指使句子保持语法正确性的词语，比

stemming主要是去除词语间的细微差异，比如take和takes，different和difference，只保留词的原型

一、文档分析基础

加权词频向量 (Weighted Term-frequency vector)

- 加权的词频向量 $\mathbf{a}_i = [a_{1i}, a_{2i}, \cdots, a_{ni}]$ 的定义是：

$$a_{ji} = L(x_{ji})G(x_{ji})$$

- $L(x_{ji})$ 表示文件 i 中第 j 个关键词的局部权重 (local weighting)。
- $G(x_{ji})$ 表示第 j 个关键词的全局权重 (global weighting)。

局部权重的种类

- No weight：$L(i) = \mathrm{tf}(i)$，$\mathrm{tf}(i)$表示关键词 i 在文档中出现的次数。
- Binary weight：若关键词 i 在文档中出现次数至少为1，则$L(i) = 1$；若没出现过，则$L(i) = 0$。
- Augmented weight：$L(i) = 0.5 + 0.5 \cdot (\mathrm{tf}(i) / \mathrm{tf}(max))$，其中$\mathrm{tf}(max)$指文档中最频繁关键词出现的次数。
- Logarithm weight：$L(i) = \log(1 + \mathrm{tf}(i))$

全局权重的种类

- No weight：对任意关键词 i，均有$G(i) = 1$。
- Inverse document frequency (idf)：$G(i)=\log(n/n(i))$，其中 n 表示文档集中文档的总数，而 $n(i)$指包含关键词 i 的文档个数。

归一化

- 归一化

  对于加权词频向量 $\mathbf{a}_i$，我们可以选择用其长度 $|\mathbf{a}_i|$ 对其归一化，也可直接使用。

- 方案

  从上述内容可以看出，我们有4×2×2=16种选择方案，在这里，我们选择最常用的 tf-idf 方案，并将特征向量 $\mathbf{a}_i$ 归一化到标准长度。

无局部权重：$L(i) = \mathrm{tf}(i)$

idf全局权重：$G(i)=\log(n/n(i))$

相似度度量 (similarity measures)

谱聚类中，图关联矩阵**A**的元素 a_{ij} 表示文档 i 与 j 之间的相似度。常用的相似度度量包括：

- Cosine similarity

$$a_{ij} = \cos(\mathbf{x}_i, \mathbf{x}_j) = \frac{\mathbf{x}_i^T \cdot \mathbf{x}_j}{\| \mathbf{x}_i \| \| \mathbf{x}_j \|}$$

- Cosine square similarity

$$a_{ij} = \cos^2(\mathbf{x}_i, \mathbf{x}_j)$$

- Radius-based similarity

$$a_{ij} = \exp\left(-\frac{\| \mathbf{x}_i - \mathbf{x}_j \|^2}{\sigma_{\mathbf{x}}} \right)$$

二、文档集

- 文档处理界两个常用的测试集：TDT2和Reuters-21578
- 上述测试集的每个文档均被手工标定过，每个文档有一个或多个类标签。
- TDT2测试集
  - 1998年里媒体报道的100类文档。
  - 摘自六大媒体机构ABS, CNN, VOA, NYT, PRI, APW。
  - 共计64527个文档，其中7803个文档有唯一标签；不同群包含的文档从1到1485不等。
  - 在实验中，我们剔除了少于5个文档的群，仅保留56个群。群的尺寸很不平衡，最大群的尺寸是最小群的一百多倍。

二、文档集

- Reuters-21578测试集
  - 包括21578个文档，共计135类。
  - 与TDT2相比，聚类更为困难，因为大多数文件有多个群标签，涵盖内容广泛。
  - 在实验中，我们剔除拥有多个标签的文档，以及少于5个文档的群；最终的测试集包括51类，共计9494个文档。

	TDT2	Reuters
No. documents	64527	21578
No. docs. used	7803	9494
No. clusters	100	135
No. clusters used	56	51
Max. cluster size	1485	3945
Min. cluster size	5	5
Med. cluster size	48	30
Avg. cluster size	137	186

三、评价指标

交互信息（mutual information）：给定两组文档聚类方式 C_1 和 C_2，他们的mutual information $\mathrm{MI}(C_1, C_2)$

$$\mathrm{MI}(C_1, C_2) = \sum_{c_{1i} \in C_1, c_{2j} \in C_2} p(c_{1i}, c_{2j}) \cdot \log_2 \frac{p(c_{1i}, c_{2j})}{p(c_{1i}) \cdot p(c_{2j})}$$

其中 $p(c_{1i})$ 和 $p(c_{2j})$ 分别表示从数据集中随机抽取一个文档，它属于群 c_{1i} 和 c_{2j} 的概率，而 $p(c_{1i}, c_{2j})$ 代表了联合概率，也就是文档同时属于群 c_{1i} 和 c_{2j} 的概率。

三、评价指标

$$\mathrm{MI}(C_1, C_2) = \sum_{c_{1i} \in C_1, c_{2j} \in C_2} p(c_{1i}, c_{2j}) \cdot \log_2 \frac{p(c_{1i}, c_{2j})}{p(c_{1i}) \cdot p(c_{2j})}$$

$\mathrm{MI}(C_1, C_2)$ 的特性：

- 当两个聚类完全独立时，$\mathrm{MI}(C_1, C_2)$ 取得最小值0。
- 当两个聚类方式完全一样时，MI取得最大值 $\max(\mathrm{H}(C_1), \mathrm{H}(C_2))$，其中 $\mathrm{H}(C)$ 代表聚类方式 C 下的熵 (entropy)。
- 不需要寻找聚类 C_1 和 C_2 间的对应关系。
- 归一化的MI取值在0到1之间

$$\mathrm{MI}(C_1, C_2) = \frac{\mathrm{MI}(C_1, C_2)}{\max(\mathrm{H}(C_1), \mathrm{H}(C_2))}$$

四、性能评估和比较

- 待评估算法包括以下六种：
  - Traditional K-means (KM)
  - Spectral clustering: Average Weight criterion (AW)
  - Spectral clustering: Ratio Cut criterion (RC)
  - Spectral clustering: Normalized Cut criterion (NC)
  - Single linear NMF model
  - Binear NMF model (BiNMF)
- 上述算法的变种
  - 原始算法+核函数
  - 原始算法+NCW

Normalized Cut Weighting (NCW)

- 除了原始形式之外，我们还对上述算法使用了**NCW**，以处理**数据集不均衡聚类**的情况。
- 不均衡数据集中，属于大群的数据点 i，相似度 a_{ij} (j =1,2,…)普遍偏大，而小群中数据点的 a_{ij} 普遍偏小，导致亲和矩阵(affinity matrix)受大聚类主导作用较大。
- 理想的情况是，每一个群的数据点权重之和都是相近的。
- NCW能够使亲和矩阵接近理想情况。

Normalized Cut Weighting (NCW)

- 在NC谱聚类中，如果将数据点i, j间的相似度定义为$a_{ij} = (\mathbf{x}_i \cdot \mathbf{x}_j)$，则数据集的亲和矩阵是$\mathbf{A} = \mathbf{X}^T\mathbf{X}$，其中 $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_n]$。
- NC谱聚类的损失函数包含 $\mathbf{D}^{-\frac{1}{2}}\mathbf{A}\mathbf{D}^{-\frac{1}{2}} = \mathbf{D}^{-\frac{1}{2}}\mathbf{X}^{\mathrm{T}}\mathbf{X}\mathbf{D}^{-\frac{1}{2}}$，这相当于用 $1/d_{ii}$对特征向量 $\mathbf{x}_i$ 进行加权。

$$F_{NC} = \sum_{c=1}^{k} \frac{\mathbf{W}(\mathbf{S}_c, \bar{\mathbf{S}}_c)}{\mathbf{W}(\mathbf{S}_c, \mathbf{V})}$$

$$= k - \sum_{c=1}^{k} \mathbf{y}_c^T \mathbf{D}^{-\frac{1}{2}} \mathbf{A} \mathbf{D}^{-\frac{1}{2}} \mathbf{y}_c$$

$$d_{ii} = \sum_{j=1}^{n} a_{ij}$$

NCW的本质

- 如果数据点 i 属于一个大群，则它与很多特征向量 $\mathbf{x}_j$ 相似度都很高，权重 $1/d_{ii}$ 就会比较小，数据点 i 的重要度就被大大削减了。
- 如果数据点 i 属于一个小群，则它只与很少一部分特征向量 $\mathbf{x}_j$ 相似，权重 $1/d_{ii}$ 就会非常大，数据点 i 的重要度就被增强了。
- 因此，NCW加权可被用来在**不知道聚类结果**的情况下消除数据集的不均衡。

其他处理

- 对于可以使用核函数的算法（i.e. KM,NC AW, RC, BiNMF），我们使用cosine square similarity作为核函数。
- 聚类数目 k 由2到10不等，对于每个聚类数目k，均从全集中随机选择不同的群进行50组实验，最终的结果由这50组结果求平均得到。
- 对于一些对初始化较为敏感的算法（i.e. KM, NMF, BiNMF），每一组实验均包含10个子实验，并取最好一次作为结果。

四、性能评估和比较

Table 3.3. Performance comparisons using TDT2 corpus

K	2	3	4	5	6	7	8	9	10	avg.
KM	0.866	0.804	0.810	0.760	0.743	0.731	0.696	0.717	0.711	0.760
KM-NCW	**0.943**	**0.909**	0.893	0.893	0.824	0.834	0.785	0.798	0.805	0.854
AW	0.834	0.754	0.743	0.696	0.663	0.679	0.624	0.662	0.656	0.701
RC	0.817	0.713	0.692	0.679	0.628	0.592	0.594	0.626	0.618	0.662
NC	0.954	0.890	0.846	0.802	0.761	0.756	0.695	0.732	0.736	0.797
NMF	0.855	0.778	0.784	0.733	0.702	0.706	0.653	0.681	0.680	0.730
NMF-NCW	**0.973**	**0.931**	0.907	0.866	0.826	0.810	0.771	0.802	0.805	0.855
BiNMF	0.859	0.773	0.789	0.745	0.702	0.703	0.652	0.691	0.682	0.733
BiNMF-NCW	**0.973**	**0.934**	0.912	0.890	0.830	0.824	0.778	0.799	0.812	0.861
following are kernel version of each algorithm with cosine square kernel function										
KM	0.500	0.544	0.594	0.581	0.592	0.622	0.587	0.619	0.616	0.584
KM-NCW	0.364	0.362	0.429	0.454	0.498	0.540	0.534	0.578	0.603	0.485
AW	0.815	0.811	0.771	0.739	0.713	0.716	0.677	0.680	0.712	0.737
RC	0.779	0.768	0.782	0.755	0.748	0.723	0.739	0.793	0.770	0.762
NC	**0.934**	**0.912**	**0.963**	**0.931**	**0.915**	**0.904**	0.876	0.876	0.902	0.912
BiNMF	0.836	0.836	0.809	0.775	0.763	0.747	0.714	0.727	0.744	0.772
BiNMF-NCW	**0.953**	**0.917**	**0.952**	**0.930**	**0.935**	**0.922**	**0.913**	**0.919**	**0.927**	**0.930**

四、性能评估和比较

Table 3.4. Performance comparisons using Reuters-215768 corpus

K	2	3	4	5	6	7	8	9	10	avg.
KM	0.404	0.402	0.461	0.525	0.561	0.548	0.583	0.597	0.618	0.522
KM-NC	0.438	0.462	0.525	0.554	0.592	0.577	0.594	0.607	0.618	0.552
AW	**0.443**	0.415	0.488	0.531	0.571	0.542	0.587	0.594	0.611	0.531
RC	0.417	0.381	0.505	0.460	0.485	0.456	0.548	0.484	0.495	0.470
NC	**0.484**	0.461	0.555	0.592	0.617	0.594	0.640	0.634	0.643	0.580
NMF	**0.480**	0.426	0.498	0.559	0.591	0.552	0.603	0.601	0.623	0.548
NMF-NCW	**0.494**	**0.500**	**0.586**	**0.615**	0.637	**0.613**	**0.654**	**0.659**	0.658	0.602
BiNMF	**0.480**	0.429	0.503	0.563	0.592	0.556	0.613	0.609	0.629	0.553
BiNMF-NCW	**0.496**	**0.505**	**0.595**	**0.616**	**0.644**	**0.615**	**0.660**	**0.660**	**0.665**	**0.606**
following are kernel version of each algorithm with cosine square kernel function										
KM	0.249	0.205	0.243	0.272	0.302	0.312	0.325	0.330	0.356	0.288
KM-NCW	0.258	0.147	0.198	0.209	0.232	0.228	0.253	0.254	0.270	0.228
AW	0.326	0.342	0.384	0.433	0.498	0.482	0.524	0.559	0.559	0.456
RC	**0.467**	0.453	**0.554**	0.517	0.518	0.516	0.577	0.574	0.573	0.528
NC	**0.485**	**0.478**	**0.587**	0.586	0.597	0.582	0.633	0.633	0.642	0.580
BiNMF	0.325	0.332	0.369	0.441	0.484	0.469	0.502	0.542	0.548	0.446
BiNMF-NCW	**0.454**	**0.472**	**0.582**	0.569	0.579	**0.602**	0.612	0.627	0.628	0.569

性能评估总结

- 无论对哪个文档集来说，RC的性能都是最差的。
- 使用核函数时，传统的K-Means是最差的。
- 无论是否使用核函数，BiNMF-NCW通常都是最好。
- 在TDT2文档集上，除了加核K-means算法，大多数算法使用 NCW加权后都能提高聚类精度(NC vs. AW, KM-NCW vs. KM, BiNMF-NCW vs. BiNMF)。
- 对TDT2而言，使用cosing square核函数可以使得AW, RC, NC, BiNMF, BiNMF-NCW算法提高至少5%的聚类性能，然而对Reuters来说却恰恰相反。

性能评估总结

- K-means算法加核后性能反而变坏。
- 我们推测这可能是由于局部最优导致的。
- 为了验证这一推测，我们将正确的聚类结果设为初始点，这时，聚类精度得到了显著提高。
- 然而，尽管BiNMF也存在局部最优的问题，但它的性能却非常好。

性能评估总结

- TDT2文档集比Reuters简单。对这样的简单数据集，多数算法都能取得一个好的结果。
- 算法间的差异在Reuters这样的复杂数据集上体现的尤为明显。
- 与传统K-Means相比，NC和NMF显示出了明显的优势。
- 使用不同的核函可以带来明显不同的结果：高维 (high order) 核函数对由相对紧凑的数据集较为高效，而低维 (low order) 核函数则适用于比较散乱的数据集。

End