潜在语义分析 (Latent semantic analysis, LSA)

潜在语义分析：

- 无监督学习方法，一种文本降维方法
- 通过[矩阵分解]发现文本与单词之间基于话题的语义关系
- 于1990年提出，在文本信息检索、推荐系统、图像处理等领域有广泛应用。

1 单词向量空间与话题向量空间

1.1 单词向量空间

单词向量空间模型（word vector space model）：给定一个文本，用一个向量表示文本“语义”。向量每一维对应一个单词，取值为单词出现的频率或权重。向量内积或标准化内积表示“文本相似度”。

单词-文本矩阵：
(1) 文本集合 $D=\{d_1,d_2,\cdots,d_n\}$; 单词集合 $W=\{w_1,w_2,\cdots,w_m\}$.
(2) 单词-文本矩阵可表示为： $X=(x_{ij})\in\mathbb{R}^{m\times n}$ 其中 x_{ij} 表示单词w_i在文本d_j中出现的频数或权重。

权重常用TF-IDF（单词频率-逆文本频率，term frequency-inverse document frequency）表示。

$$\mathrm{TFIDF}_{ij}=\frac{\mathrm{tf}_{ij}}{\mathrm{tf}_{\cdot j}}\log\frac{\mathrm{df}}{\mathrm{df}_i} \tag{1.1}$$

(1) tf_{ij}: 单词w_i在文本d_j中出现的频数； $\mathrm{tf}_{\cdot j}=\sum_i \mathrm{tf}_{ij}$;
(2) df_i: 包含单词w_i的文本数；$\mathrm{df}=n$ 是总文本数。
解读：
(1) 单词频率越高，权重越高;
(2) 单词出现的文本越少，则权重越高（该单词更加能够刻画该文本的特点）.

单词文本矩阵的第 j 列 $x_j = (x_{1j}, x_{2j}, \cdots, x_{mj})^\top$表示$d_j$的信息。 文本 d_i与 d_j 之间的相似度：

$$x_i \cdot x_j, \quad \text{or} \quad \frac{x_i \cdot x_j}{\|x_i\| \|x_j\|} \tag{1.2}$$

即向量内积或者标准化内积（余弦）。

解读：

(1) 模型优点：模型简单，计算效率高;

(2) 模型局限：不一定能准确刻画语义相似度。自然语言存在一词多义与多词一义，例如：

a. 张三 | 爱 | 吃 | 苹果

b. 李四 | 喜欢 | 橘子

1.2 话题向量空间

文本的语义相似度可以用两者“话题”（topic）相似度表示。话题指文本讨论的内容或者主题。

话题向量空间

假设文本共有k个话题，每个话题由定义在单词集合W上的m维向量表示: $t_l = (t_{1l}, t_{2l}, \cdots, t_{ml})^\top$。 其中 t_{il} 是 w_i在话题t_l上的权重。

这k个话题向量$t_1, t_2, \cdots, t_k$ 张成一个话题向量空间。

单词话题矩阵 ： $T = (t_1, t_2, \cdots, t_k) = (t_{il} : 1 \leq i \leq m, 1 \leq l \leq k)$.

文本在话题向量空间中的表示

文本d_j在单词向量空间中的表示为 $x_j \in \mathbb{R}^m$。将x_j投影到话题向量空间T中得到 $y_j = (y_{lj} : 1 \leq l \leq k)^\top \in \mathbb{R}^k$($k$是低维向量)。$y_{lj}$代表文本$d_j$在话题$t_l$上的权重。

话题文本矩阵：$Y = (y_1, y_2, \cdots, y_n) \in \mathbb{R}^{k \times n}$.

可以将单词x_j通过k个话题进行线性近似：

$$x_j \approx y_{1j}t_1 + y_{2j}t_2 + \cdots + y_{kj}t_k.$$

即：可以用单词-话题矩阵T以及话题-文本矩阵Y近似表示单词-文本矩阵X:

$$X \approx TY. \tag{1.3}$$

文本相似度：

(1) 在原始单词向量空间中，文本之间的相似度可以表示为 $x_i \cdot x_j$;

(2) 在话题空间中，文本相似度近似表示为 $y_i \cdot y_j$.

2 潜在语义分析算法

2.1 矩阵奇异值分解算法

潜在语义分析根据话题个数k对单词-文本矩阵X进行截断的奇异值分解：

$$X \approx U_k \Sigma_k V_k^\top$$

其中 $U_k \in \mathbb{R}^{m \times k}$是前k个左特征向量矩阵, $\Sigma_k \in \mathbb{R}^{k \times k}$ 是对角矩阵，V_k是前 k个右特征向量。

可以得到解 $T = U_k$ 以及 $Y = \Sigma_k V_k^\top$.

2.2 非负矩阵分解算法

给定一个非负矩阵 $X \geq 0$ (所有元素非负)，找到两个非负矩阵 $W \in \mathbb{R}^{m \times k} \geq 0$以及$H \in \mathbb{R}^{k \times n} \geq 0$,使得

$$X \approx WH \tag{2.1}$$

由于$k < \min\{m, n\}$, 非负矩阵分解是对原数据的压缩。

其中，$T = W$为话题向量空间；$Y = H$ 为文本在话题向量中间的表示。

非负矩阵分解的求解

1. 损失函数:

(1) 平方损失：

$$\|A - B\|_F^2 = \sum_{i,j}(a_{ij} - b_{ij})^2 \tag{2.2}$$

(2) 散度（divergence）。散度损失函数的定义为:

$$D(A\|B) = \sum_{i,j}\left(a_{ij}\log\frac{a_{ij}}{b_{ij}} - a_{ij} + b_{ij}\right) \tag{2.3}$$

其中，$0\log 0 = 0$。 散度函数在$A = B$时取到下界0。 A 和 B不对称，当 $\sum_{i,j} a_{ij} = \sum_{i,j} b_{ij} = 1$时，即Kullback-Leiber散度（或相对熵）。

注： KL散度用于度量两个概率分布的差异。其定义如下：

$$D_{KL}(P\|Q) = E_{x\sim P}\left[\log\frac{P(x)}{Q(x)}\right]. \tag{2.4}$$

因此非负矩阵分解可以表示为带约束的优化问题：

$$\begin{aligned}&\min_{W,H}\|X - WH\|_F^2 \quad \text{or} \quad \min_{W,H} D(X\|WH)\\ &\text{s.t.} \;\; W, H \geq 0\end{aligned}$$

2. 算法:

Lee and Seung (2001)给出了以上非负矩阵分解的优化算法，算法交替的对W和H进行更新。

Theorem 1. 平方损失 $\|X - WH\|_F^2$ 对下列乘法更新规则：

$$H_{lj} \leftarrow H_{lj} \frac{(W^\top X)_{lj}}{(W^\top WH)_{lj}}$$
$$W_{il} \leftarrow W_{il} \frac{(XH^\top)_{il}}{(WHH^\top)_{il}}$$

是非增的。当且仅当 W 和 H是平方损失的稳定点时函数更新不变。

以上更新规则基于梯度下降算法得到，具体证明见 Lee and Seung (2001)。以平方损失为例，考虑

$$J(W,H) = \frac{1}{2}\|X - WH\|_F^2 = \frac{1}{2}\sum_{i,j}\Big\{X_{ij} - (WH)_{ij}\Big\}^2$$

则

$$\begin{aligned}\frac{\partial J(W,H)}{\partial W_{il}} &= -\sum_j \Big\{X_{ij} - (WH)_{ij}\Big\} H_{lj} \\ &= -\Big\{(XH^\top)_{il} - (WHH^\top)_{il}\Big\}\end{aligned}$$

同样可得

$$\frac{\partial J(W,H)}{\partial H_{lj}} = -\Big\{(W^\top X)_{lj} - (W^\top WH)_{lj}\Big\}$$

则使用梯度下降算法可得：

$$W_{il} \leftarrow W_{il} + \lambda_{il}\Big\{(XH^\top)_{il} - (WHH^\top)_{il}\Big\} \tag{2.5}$$
$$H_{lj} \leftarrow H_{lj} + \mu_{lj}\Big\{(W^\top X)_{lj} - (W^\top WH)_{lj}\Big\} \tag{2.6}$$

选取

$$\lambda_{il} = \frac{W_{il}}{(WHH^\top)_{il}}, \quad \lambda_{lj} = \frac{H_{lj}}{(W^\top WH)_{lj}} \tag{2.7}$$

即可得到更新迭代规则。

注：

(1) 以上算法在更新时，选取初始矩阵非负，则迭代过程中可保证W、H都非负；

(2) 算法对 W和H交替更新，每次迭代需对 W的列向量进行归一化为单位向量。