基于扩展潜在语义结构的文本分类模型

王 明 文

Email: mwwang@jxnu.edu.cn

江西师范大学

论文提出的背景

- 自动文本分类就是在给定的分类体系下，根据文本的内容自动地确定文本关联的类别。
- 当前，已经有很多基于统计和机器学习的文本分类算法，如：回归模型、K近邻、决策树、朴素贝叶斯和支持向量机等。

向量空间模型（VSM）

- 向量空间模型（VSM），是当前最常用的文档表示模型。

$$X=\begin{pmatrix} x_{11} & x_{12} & \cdots & x_{1n} \\ x_{21} & x_{22} & \cdots & x_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ x_{m1} & x_{m2} & \cdots & x_{mn} \end{pmatrix}=(X_{\bullet 1},X_{\bullet 2},\cdots,X_{\bullet n})=\begin{pmatrix} X_{1\bullet} \\ X_{2\bullet} \\ \vdots \\ X_{m\bullet} \end{pmatrix}$$

$\mathbf{X_{.j}}$ 代表一个词；

$\mathbf{X_{i.}}$ 代表一个文档。

- 很多分类算法都是基于从文本中抽取关键词的方法。在这种方法中，假定一个关键词唯一地代表一个概念或语义单元。
- 然而实际的情况是：一个词往往有多个不同的含义，多个不同的词也可以表示同一个语义。这就是所谓的一词多义和多词一义问题。
- 一词多义和多词一义，是所有基于语义的算法必须解决的两个主要问题。

- 潜在语义索引（LSI: Latent Semantic Indexing），是近年来比较有效的算法之一。
- LSI 把原始的向量空间转换成潜在语义空间，文档就在转换后的语义空间上进行表示和比较。

$$\mathbf{X} = \mathbf{U}_r \Sigma_r \mathbf{V}_r^{\mathbf{T}} \approx \mathbf{U}_k \Sigma_k \mathbf{V}_k^{\mathbf{T}}$$

$$\mathbf{X}_{i\bullet} \rightarrow \mathbf{X}_{i\bullet} \mathbf{V}_k$$

- 文档维数的减小，既降低了文本分类和检索的计算复杂度，也去除了文当矩阵中的一部分噪音。
- 实验表明这种方法可以在一定程度上解决一词多义和多词一义问题。

- 然而，LSI在降低维数的同时也会丢失结构信息。LSI基于文档信息来建立语义空间，得到的特征空间会保留原始文档矩阵中最主要的全局信息。
- 但有种情况是：一些对稀有类别分类贡献很大的特征，放在全局下考虑却会变得不重要了。这样的特征在维数约减的过程中，就很容易被滤掉。
- 事实上也是，稀有类中出现的词很可能是文档集中的非常见词，而很有可能被滤掉。而如果这样，特定类别的分类精度就肯定会受影响。

例：

$$\mathbf{X} = \begin{array}{c} \\ d_1 \\ d_2 \\ d_3 \end{array} \begin{array}{c} \begin{array}{ccc} t_1 & t_2 & t_3 \end{array} \\ \begin{pmatrix} 1/\sqrt{2} & 1/\sqrt{2} & 0 \\ 1/\sqrt{2} & 1/\sqrt{2} & 0 \\ 0 & 0 & 1 \end{pmatrix} \end{array} \begin{array}{c} class \\ 1 \\ 1 \\ 2 \end{array}$$

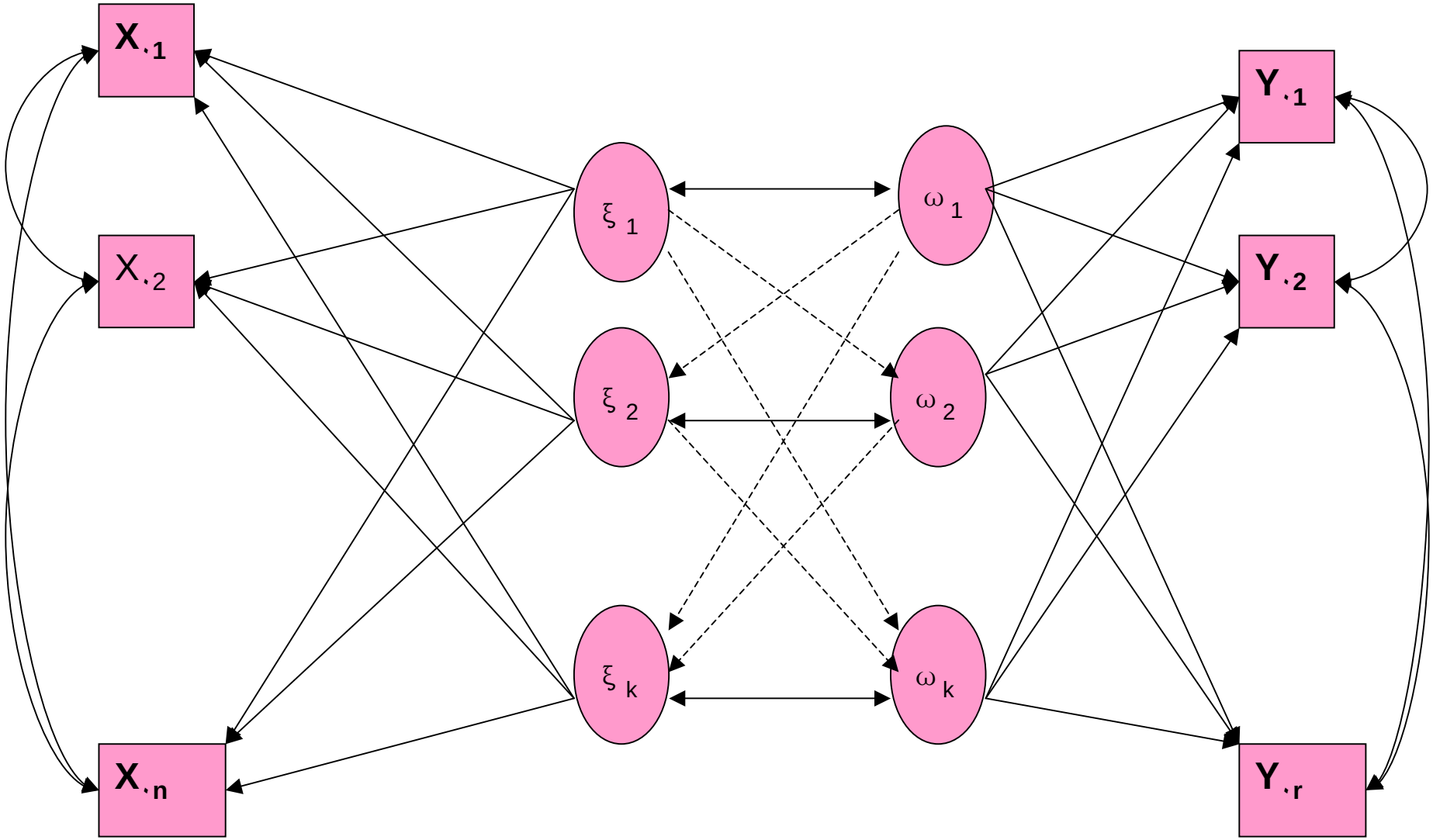
$$\mathbf{X} = \begin{pmatrix} 1/\sqrt{2} & 0 \\ 1/\sqrt{2} & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} \sqrt{2} & \\ & 1 \end{pmatrix} \begin{pmatrix} 1/\sqrt{2} & 1/\sqrt{2} & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

$$\xrightarrow{k=1} \begin{pmatrix} 1/\sqrt{2} \\ 1/\sqrt{2} \\ 0 \end{pmatrix} \begin{pmatrix} \sqrt{2} \end{pmatrix} \begin{pmatrix} 1/\sqrt{2} & 1/\sqrt{2} & 0 \end{pmatrix}$$

如果从对角阵中截去第二个特征值，那么我们得到的特征空间将只能反映第一类别的信息，而反映第二个类别的信息被完全滤去了。

扩展LSI模型

- 针对上述问题，我们提出了一种对LSI扩展的分类模型：扩展潜在语义结构文本分类模型。
- 与LSI模型类似，我们也希望从原始空间中得到一个潜在语义空间；然而不同的是，我们要在尽量保留文档信息的同时，通过对文档信息矩阵X 和文档类别信息矩阵Y 建模，把文档和类别之间的关联也考虑进来。
- 我们希望扩展后的分类模型能够表现出比LSI模型更好的分类性能。


X·1
X·2
X·n
ξ 1
ξ 2
ξ k
ω 1
ω 2
ω k
Y·1
Y·2
Y·r

- 我们希望通过建立若干对潜在语义变量来表示上图的交叉信息，
  就如：$(\xi_1, \omega_1), (\xi_2, \omega_2), \ldots, (\xi_k, \omega_k)$
- 其中，ξ 代表矩阵中的潜在语义信息，ω 代表矩阵中的潜在信息。
- (ξ_i, ω_i)按他们代表信息的重要程度降序排列，(ξ_1, ω_1)也就是代表最重要的信息，(ξ_2, ω_2)代表次重要的信息，依次类推。

- 变量对(ξ_i, ω_i)的确定条件：
- （a）变量ξ_i要尽可能好的表示矩阵X的信息 => $Var(\xi_i) \rightarrow max$；
- （b）变量ω_i要尽可能好的表示矩阵Y的信息 => $Var(\omega_i) \rightarrow max$；
- （c）变量对(ξ_i, ω_i)要尽可能好的表示矩阵X和Y之间的联系 => $r(\xi_i, \omega_i) \rightarrow max$；其中$r(\bullet, \bullet)$代表相关系数。

- 为了解决三个极值问题，我们在统计方法上把其整合成一个极值问题：

- Cov(ξ_i, ω_i) -> max

- 其中协方差

$$Cov(\xi_i, \omega_i) = \sqrt{Var(\xi_i) Var(\omega_i)} \times r(\xi_i, \omega_i)$$

由潜在语义对(ξ_i, ω_i)表示的信息可以如下方式进行计算：

$$\hat{\mathbf{X}}_i = \xi_i (\xi_i^{\mathrm{T}} \xi_i)^{-1} \xi_i^{\mathrm{T}} \mathbf{X}_{i-1}$$

$$\hat{\mathbf{Y}}_i = \omega_i (\omega_i^{\mathrm{T}} \omega_i)^{-1} \omega_i^{\mathrm{T}} \mathbf{Y}_{i-1}$$

把已经表示的信息从原始矩阵中去除后，就得到了剩余的矩阵信息：

$$\mathbf{X}_i = \mathbf{X}_{i-1} - \hat{\mathbf{X}}_i \qquad \mathbf{Y}_i = \mathbf{Y}_{i-1} - \hat{\mathbf{Y}}_i$$

算法 (LSR/PLS1):

$\mathbf{E}_0 = \mathbf{X}$; $\mathbf{f}_0 = \mathbf{y}$;

FOR k=1,…,s DO

$\xi_k = \mathbf{E}_{k-1}^{\mathbf{T}} \mathbf{f}_{k-1}$;

$\mathbf{t}_k = \mathbf{E}_{k-1} \xi_k$;

$\mathbf{p}_k = \mathbf{E}_{k-1}^{\mathbf{T}} \mathbf{t}_k / \mathbf{t}_k^{\mathbf{T}} \mathbf{t}_k$;

$\omega_k = \mathbf{f}_{k-1}^{\mathbf{T}} \mathbf{t}_k / \mathbf{t}_k^{\mathbf{T}} \mathbf{t}_k$

$\mathbf{E}_k = \mathbf{E}_{k-1} - \mathbf{t}_k \mathbf{p}_k^{\mathbf{T}}$;

$\mathbf{f}_k = \mathbf{f}_{k-1} - \mathbf{t}_k \omega_k^{\mathbf{T}}$;

ENDFOR

算法复杂度：O(mns)

表示文档信息最多的特征

$$\mathbf{X} = \begin{array}{c} \begin{matrix} x_1 & x_2 & x_3 \end{matrix} \\ \begin{pmatrix} 1 & 3 & 10 \\ 1 & 2 & 4 \\ 1 & 1 & 6 \\ 1 & 0 & 8 \\ 1 & 0 & 2 \end{pmatrix} \end{array}$$

$$Var(x_1) = 0$$

$$Var(x_2) = \frac{1}{5}\sum (x_{i2} - \bar{x}_2)^2 = 1.36$$

$$Var(x_3) = \frac{1}{5}\sum (x_{i3} - \bar{x}_3)^2 = 8$$

$$Var(x_3) > Var(x_2) > Var(x_1)$$

特征变量 x_3 表示矩阵 X 的信息最多。

新模型的优势：对区别两个不同类别贡献大的重要特征能够被提取出来。

$$\mathbf{X}=\begin{array}{c}\begin{matrix}x_1 & x_2 & x_3\end{matrix} \text{ 类别}\\ \begin{pmatrix}1 & 3 & 10\\ 1 & 2 & 4\\ 1 & 1 & 6\\ 1 & 0 & 8\\ 1 & 0 & 2\end{pmatrix}\begin{matrix}1\\1\\1\\2\\2\end{matrix}\end{array}$$

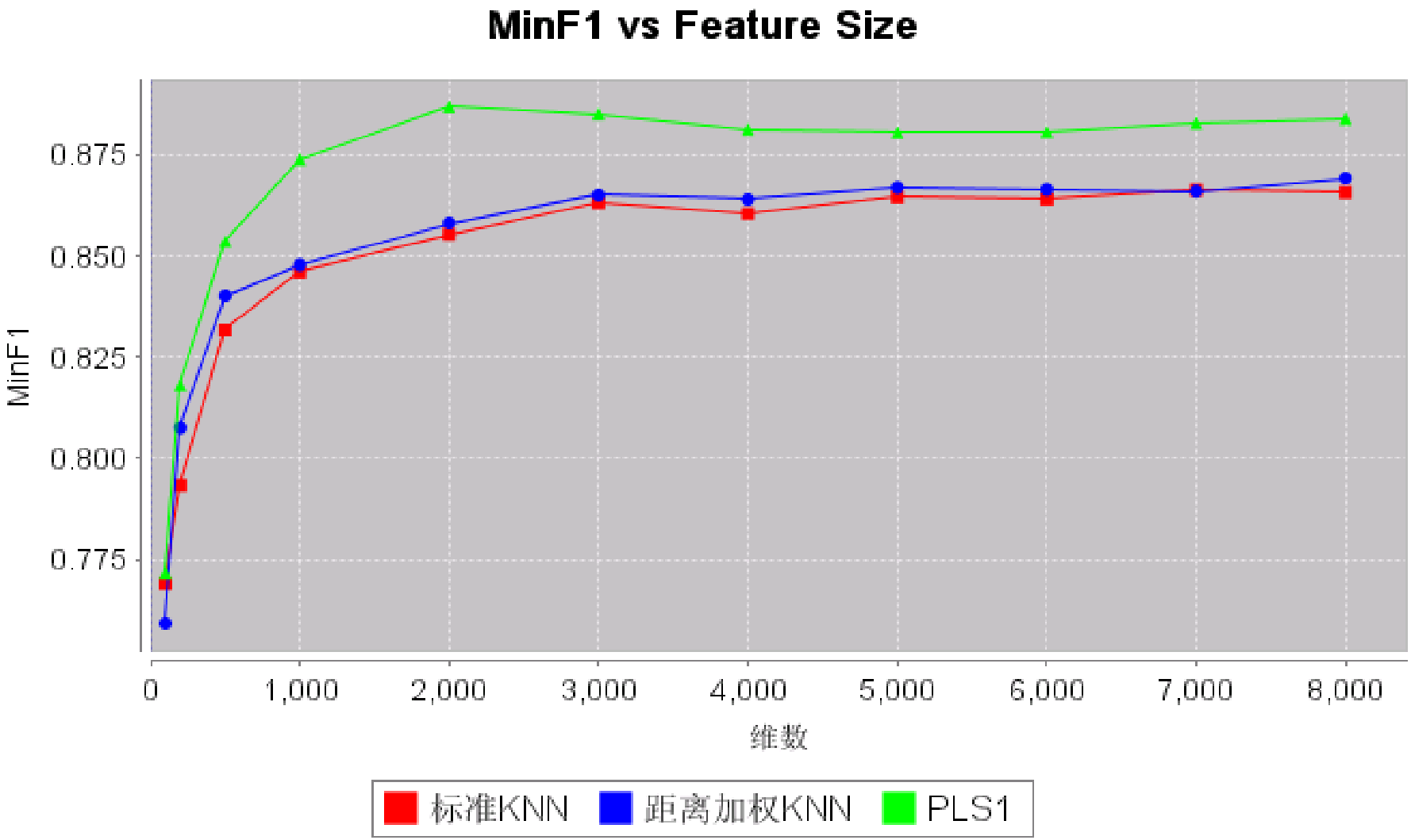
$$r(x_1, class)=0$$

$$r(x_2, class)=\frac{1}{5}\left(\frac{-2.4}{\sqrt{1.36\times 0.24}}\right)=-0.84$$

$$r(x_3, class)=\frac{1}{5}\left(\frac{-2}{\sqrt{8\times 0.24}}\right)=-0.29$$

X_2 是对分类最重要（贡献最大）的特征。

实验结果



如未特别说明，实验用语料库均为：“复旦大学中文文本分类语料库”

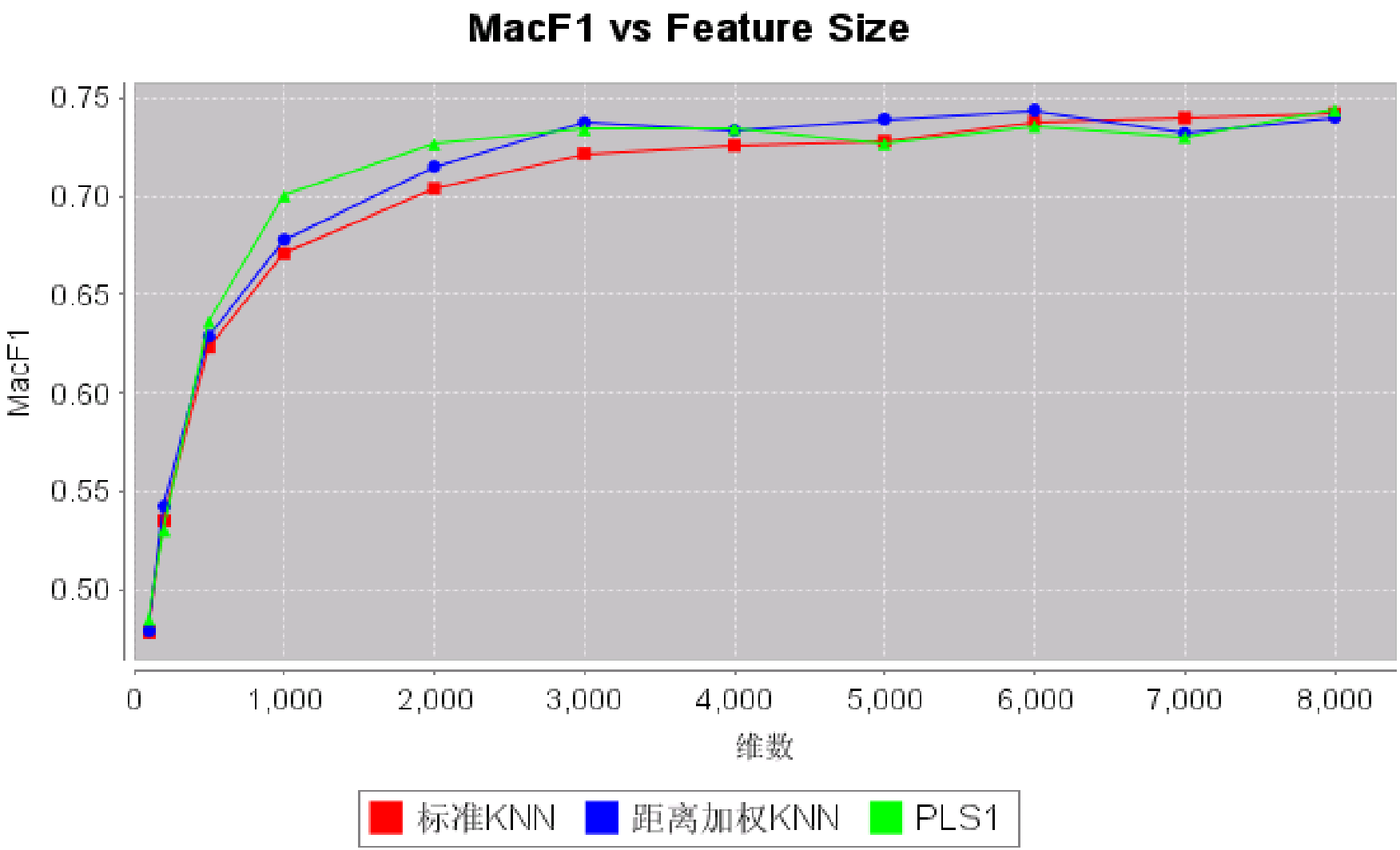

MacF1 vs Feature Size
0.75
0.70
0.65
0.60
0.55
0.50
MacF1
0
1,000
2,000
3,000
4,000
5,000
6,000
7,000
8,000
维数
标准KNN
距离加权KNN
PLS1

表1 特征维数变化下的微平均F1和宏平均F1

特征维数	100	200	500	1,000	2,000	3,000	4,000	5,000	6,000	7,000
微平均F1	0.772	0.818	0.854	0.874	**0.887**	0.885	0.882	0.881	0.881	0.883
宏平均F1	0.485	0.530	0.636	0.700	0.731	0.734	0.734	0.727	**0.736**	0.730

表2：2000维特征下各个类别的准确率、召回率和F1值

类别	训练文档正例	测试文档正例	准确率	召回率	F1
Economy	1369	1127	0.919	0.898	0.908
Sports	1204	980	0.968	0.922	0.945
Computer	1019	591	0.972	0.944	**0.958**
Politics	1010	989	0.917	0.918	0.918
Agriculture	847	635	0.903	0.943	0.923
Environment	805	371	0.953	0.930	0.941
Art	510	286	0.734	0.839	0.783
Space	506	248	0.941	0.907	0.924
History	466	468	0.757	0.778	0.767
Military	74	75	0.559	0.507	0.531
Education	58	58	0.623	0.569	0.595
Transport	57	58	0.796	0.672	0.729
Law	51	52	0.926	0.481	0.633
Medical	51	52	0.735	0.481	0.581
Philosophy	40	33	0.520	0.394	0.448
Mine	33	29	0.720	0.621	0.667
Literature	33	32	0.800	0.125	**0.216**
Energy	30	31	0.905	0.613	0.731
Electronics	26	26	0.750	0.577	0.652
Communication	25	22	0.773	0.773	0.773

表3：PLS1算法在特征维数变化下的微平均F1和宏平均F1

实验语料库: **Reuter 21578**

特征维数		100	200	500	1,000	2,000	3,000	4,000	5,000	6,000	7,000	8,000
Top 10	$Micro_{avg}$ F1	0.896	0.922	0.926	**0.929**	**0.929**	0.928	0.921	0.919	0.921	0.922	0.916
	$Macro_{avg}$ F1	0.836	0.875	0.881	0.877	0.880	**0.881**	0.874	0.863	0.868	0.870	0.867
Other 80	$Micro_{avg}$ F1	0.581	0.664	0.712	0.722	**0.723**	0.717	0.713	0.702	0.689	0.691	0.677
	$Macro_{avg}$ F1	0.354	0.419	0.543	0.567	0.568	**0.570**	0.555	0.553	0.533	0.552	0.539
All 90	$Micro_{avg}$ F1	0.815	0.857	0.873	0.878	**0.879**	0.876	0.870	0.865	0.863	0.866	0.855
	$Macro_{avg}$ F1	0.408	0.470	0.581	0.601	0.603	**0.604**	0.591	0.587	0.570	0.587	0.576

表4：Reuter 21578 下各常见分类模型 F1值比较

部分实验结果摘自：Zhang, T. and Oles, F. J. (2001) Text categorization based on regularized linear classification methods, *Information Retrieval*, volume 4, pp. 5-31.

类别 \ 模型	Naive Bayes	Linear regression	Mod Least Squares	Logistic regression	SVM	Mod SVM	PLS1 (2000)
earn	0.966	0.971	**0.984**	**0.984**	0.981	0.981	0.978
acq	0.917	0.932	0.954	0.952	0.953	0.945	**0.956**
money-fx	0.700	0.732	0.760	0.752	0.744	0.745	**0.809**
grain	0.766	**0.916**	0.903	0.884	0.896	0.906	0.909
crude	0.841	0.862	0.849	0.859	0.848	0.840	**0.884**
trade	0.523	0.708	0.763	0.729	0.734	0.748	**0.816**
interest	0.682	0.752	0.757	0.781	0.759	0.747	**0.792**
wheat	0.581	**0.896**	0.885	0.882	0.889	**0.896**	0.853
ship	0.764	0.807	0.836	0.819	0.824	0.838	**0.884**
corn	0.524	0.893	0.881	0.887	0.862	0.867	**0.920**
$Micro_{avg}F1$ (All 90)	0.770	0.860	0.872	0.864	0.865	0.865	**0.879**

下一步的工作

- 找到一种自动确定潜在语义对的数量***s*** 的方法；
- 对多类分类算法模型进行检测和评价。

基于模糊-粗糙集的文本分类方法

背景

- 在文本自动分类问题中*k*近邻分类方法是一种简洁、直观的方法。
- 然而传统*k*近邻分类方法对待分类文本*X*的*k*个最近邻同等看待，即不考虑X与其近邻之间的距离，但在实际应用中，这种距离是不可忽略的，并且*k*值不易确定。

- 利用Fuzzy-rough集理论改进传统的*k*-NN用于文本分类。
- Fuzzy-rough 集理论能处理在多类分类问题中由于类的重叠引起训练样本的模糊不确定性，以及属性不足引起类边界的粗糙不确定性。

模糊－粗糙集

- 模糊－粗糙集是对粗糙集理论和模糊集理论的推广。当等价类的元素所属的类别不明确时，可将等价关系表示成模糊关系的形式 F = $\{F_1,F_2,\ldots F_H\}$，F_j，$j\in\{1,2,..H\}$ 。
- 给定X上的一个模糊划分θ，利用上近似$\overline{\theta}$和下近似$\underline{\theta}$的形式表达任一模糊集合F，称$\overline{\theta}(F)$和$\underline{\theta}(F)$为模糊－粗糙集 。

- 定义模糊上下近似如下[3]：

$$\mu_{\bar{\theta}}(F_i) = \sup_{x\in\theta} \mu_{F_1}(x) \times \mu_F(x) \qquad \forall x$$

$$\mu_{\underline{\theta}}(F_i) = \inf_{x\in\theta} \mu_{F_1}(x) \times \mu_{\mathrm{F}}(x) \qquad \forall x$$

上式表示了模糊事件F的可能性和必然性程度。

基于Fuzzy-Rough的文本分类方法

- 邻域空间和*k*的确定

传统的*k*-NN方法中*k*的值要通过训练得到，然而在处理多类分类问题中不同的类的最优的*k*值是不同的; 并且由于数据的分布不同导致这些邻近点之间的距离有很大的差异；为了灵活的处理这个问题，可使用待分类文本与训练集文本的距离构造邻域空间W，根据这个基于距离的尺度找出邻近点。

不确定性的出现和处理

- 假设X＝{x_1,x_2,x_3,…x_n }为已知类别的训练集文本，C={C_1,C_2,…,C_c}为已知的文本所属类别。
- 令x^S为待分类文本，　＝ {x_{s1},x_{s2},…,x_{sk}}是x^S的邻域空间中的k个训练样本构成的集合，由于类之间的重叠，这k个训练样本并不一定属于同一个类，而可能是属于多个类，，这导致邻近文本类别出现模糊不确定性；我们采用如下方法来处理这类的模糊性：

- 设训练文本x_{si}的已知类别为C_q，则x_{si}属于类C_j的隶属度定义如下：

$$\mu_{c_j}(x_{si}) = \begin{cases} 0.51 + 0.49\, n_j / K & if \;\; j = q \\ 0.49\, n_j / K & if \;\; j \neq q \end{cases}$$

- n_j表示在K个邻近点中属于类c_j的个数，K为邻近点数。

- 对于测试文本x^S，若已知x^S属于类c_q,但邻域空间中的k个点并不都来自类c_q，只根据邻近点并不能完全确定x^S的类别，这样导致待测试样本类别的粗糙不确定性，也就是按当前文档集已有的属性（即关键词）并不能将不同类完全划分，类间的边界是粗糙的；这时x^s与邻域空间中的文本的关系称为模糊相似；
- 记为 $\tilde{\mu}_{x^s}(y)$ $y \in W$ 。

- 模糊相似值的计算可以有不同的方法，文中采用的方法如下：

$$\tilde{\mu}_{x^s}(y) = (1 + \alpha \| x - y \|^{2/(q-1)})^{-1} \qquad y \in W;$$

- ||x-y||：x与y间的距离，这里采用欧式距离；
- q：训练参数，调节距离对函数值的影响；
- α：调节参数，用于控制函数值的变换范围，可以有多种取法，可以固定或根据距离来定。

- 对于待分类文本和邻域空间之间出现的这两类不确定性，我们使用模糊粗糙隶属函数(记为 $\tau_{C_j}(x^s)$)来综合两者的影响。

$$\tau_{C_j}(x^s) = \frac{1}{|W|}\sum_{y\in W}\tilde{\mu}_{x^s}(y)\mu_{C_j}(y)$$

- 最后待分类文本的类归属由模糊－粗糙隶属函数值来确定，哪个类的函数值最大，则判断它属于该类。即：如果

$$\tau_{C_q}(x^s) = \max(\tau_{C_j}(x^s)), \text{j=1,2,…,c}$$

则$x^s \in C_q$。

分类算法

- 计算待分类文本与训练集各文本间的距离；
- 排序后，根据距离构建邻域空间，同时确定适宜该样本的k值；
- 对邻域空间中的k个训练文本，计算待分类文本关于各个类的模糊—粗糙隶属函数的值；
- 根据函数值的大小，判断待分类文本的类标签。

实验结果

方法 / 类别	传统 k 近邻法		模糊 k 近邻法		证据理论 k 近邻法		模糊一粗糙集法	
	召回率	精度	召回率	精度	召回率	精度	召回率	精度
earn	0.968	0.855	0.972	0.857	0.933	0.836	0.971	0.864
acq	0.812	0.931	0.804	0.935	0.745	0.87	0.813	0.93
Money -supply	0.821	0.575	0.857	0.6	0.964	0.844	0.857	0.788
coffee	0.955	0.913	1.0	0.957	1.0	0.846	0.985	0.913
ship	0.667	0.828	0.667	0.828	0.583	0.808	0.667	0.89
sugar	0.96	0.923	0.96	0.89	0.88	0.917	0.96	0.931
trade	0.893	0.77	0.91	0.773	0.893	0.817	0.893	0.831
crude	0.744	0.978	0.752	0.978	0.802	0.858	0.76	0.958
Money -fx	0.787	0.77	0.787	0.778	0.708	0.733	0.842	0.815
interest	0.42	0.92	0.494	0.93	0.605	0.766	0.52	0.857
宏平均	0.803	0.846	0.82	0.852	0.811	0.83	0.827	0.877

结束语

- 从实验结果可以看出，由于Fuzzy-rough 集方法同时考虑了邻近点的模糊性和粗糙性，一定程度上提高了分类的精度和召回率。由于没有充分考虑文档数较多的类对文档数较少的类的干扰，导致这些小类分类的质量较低，在未来的工作中将考虑这部分因素的影响。目前该实验使用的是英文数据集，下一步实验中将使用中文数据集，并尝试使用其他的特征表示方法和维数约简方法，还将探讨更好的确定邻域空间的方法。

谢谢大家！