基于 SVM 的文本分类器

目录

摘要

文本分类是文本挖掘的一个重要组成部分，它可以提高信息检索的速度和准确率 [1]。SVM（Support Vector Machine，支持向量机）方法基于 SLT（Statistical Learning Theory，统计学习理论），适合大样本集的分类，它将分类和降维结合在一起，所以非常适用于文本分类[2]。

我们的程序实现了一个基于 SVM 的中文文本分类器。它对给定的中文文本进行分词、特征提取等操作，最后把文本用特征向量空间模型表示。这些文本特征向量就可以直接提交给 SVM 进行训练和分类。对于用来训练的文本特征向量，需要预先人工判断其类别，并一起提交给 SVM。训练好的 SVM 就可以用来对实际的文本进行分类。

我们在代码实现的基础之上，还对该文本分类器的执行效率和分类的正确率进行了一系列的测试和分析。根据实验结果，当类别数目为 2，且每个类别特征数目为 2000 时，分类的正确率为 96.99%；当类别数目为 10，且每个类别特征数目为 4000 时，分类的正确率为 93.88%。

关键词：文本分类；分词；特征向量；SVM

1. 概述

1.1. 文本分类

文本分类是在给定分类体系下，根据文本内容自动确定文本类别的过程[2]。人工分类非常费时，效率过低。90 年代以来，众多的统计方法和机器学习方法应用于自动文本分类。

自动文本分类过程如图 1-1 所示。首先对文本进行预处理，将文本用模型表示，进行特征提取；然后构造并训练分类器；最后用分类器对新文本进行分类[2]。

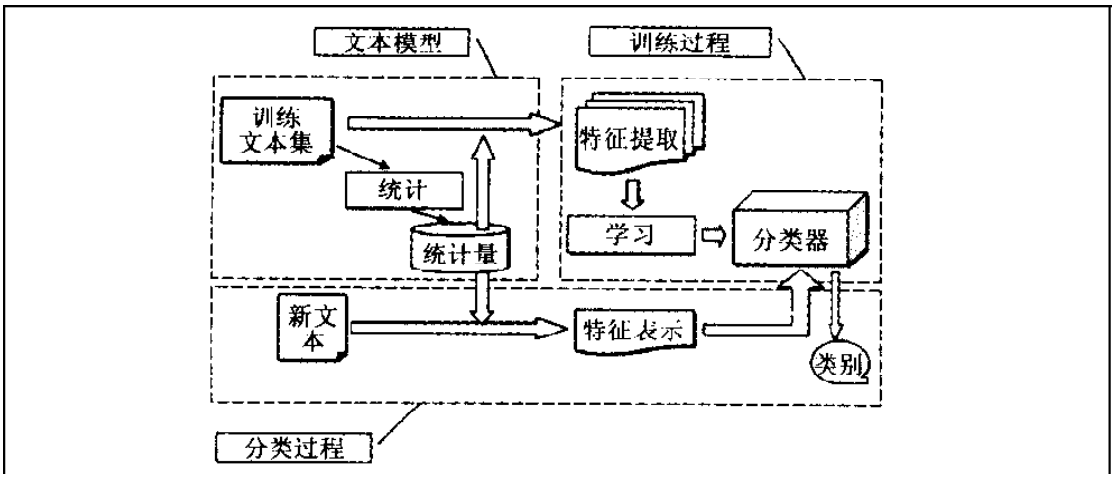

图1-1 自动文本分类过程

更详细的，目前典型的中文文本分类技术过程如下：

1. 准备测试集文本；
2. 测试集文本人工分成若干类别；
3. 利用词典进行文本分词；
4. 去停用词、合并数字人名，进行一定的降维；
5. 采用 TF-IDF 公式计算词条权重；
6. 特征项抽取（利用互信息量、词熵、X2 统计量等方法）；
7. 特征降维处理，方法有主成分分析、潜在语义标引、非负矩阵分解）；
8. 构造分类器（比如 SVM、向量距离分类法、KNN 算法、贝叶斯算法等）；
9. 用特征词汇描述测试文本；
10. 取部分测试文本用来训练分类器，另一部分测试，并根据结果进行一些改进和优化；
11. 利用训练好的分类器对新文本进行分类。

1.2. SLT 和 SVM

SLT 是一种专门研究小样本情况下机器学习规律的理论。该理论针对小样本统计问题建立了一套新的理论体系，在这种体系下的统计推理规则不仅考虑了对渐近性能的要求，而且追求在有限信息的条件下得到最优结果[3]。

SVM 建立在统计学习理论的 VC 维理论和结构风险最小化（SRM）原理基础上，根据有限的样本信息在模型的复杂性和学习能力之间寻求最佳折衷，以期获得最好的推广能力[3]。

SVM 的主要优点在于：

- 专门针对有限样本情况，其目标是得到现有信息下的最优解而不仅仅是样本数趋于无穷大时的最优值；
- 算法最终将转化为一个二次型寻优问题，从理论上说，得到的将是全局最优点，解决了在神经网络方法中无法避免的局部极值问题；
- 算法将实际问题通过非线性变换转换到高维的特征空间，在高维特征空间中构造线性判别函数来实现原空间中的非线性判别函数，从而保证学习机器有较好的推广能力，同时它巧妙地解决了维数问题，其算法复杂度与样本维数无关。

1.3. 文本分类器简介

本文实现了一个文本分类器，它包含典型文本分类技术的各个步骤。该文本分类器采用 SCW 库进行分词，利用信息论中的互信息量抽取特征，把文本表示成特征向量空间模型，利用 TF*IDF 计算文本向量中每个特征的权重，并利用 SVM-Multiclass 作为分类器进行文本分类。

本文测试数据来源于中文自然语言处理开放平台提供的演示系统文本分类系统（KNN 和 SVM）上的测试文档[4]。

2. 程序实现

我们的程序主要有三个模块组成，他们分别是：倒排文件模块，特征向量模块，SVM 分类模块。整个文本分类的过程主要有训练和分类两个过程组成，整个流程如下图所示：

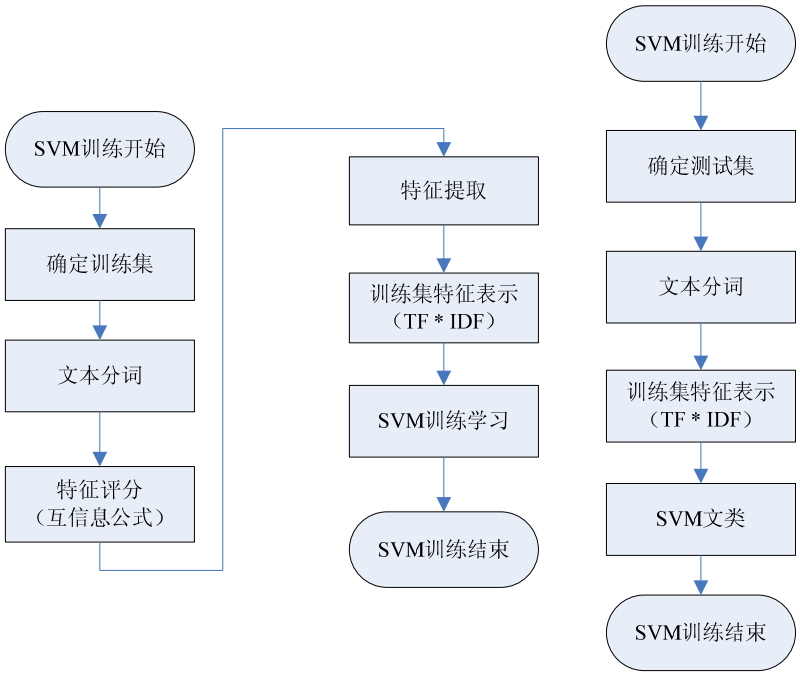

图 2-1 文本分类器流程图

2.1. 倒排文件模块

倒排文件模块主要是建立倒排索引文件，为特征向量模块提供必要的数据结构。现在有两个功能组成，他们分别是：（1）中文文本分词（2）倒排文件创建。

2.1.1. 中文文本分词

所谓的中文分词是一种将连续的汉语文本序列按一定规则拆分为具有独立语义的词组的过程。中文分词是当前分词技术中的一种，分词技术从语言文本结构上来讲大致有两类：一类以英文为代表的西方语言文本，其文本中的词组以空格作为自然间隔，从语义准确性及技术复杂度来讲都比较简单。另一类是以汉语为代表的东亚语言文本，由于文本是由连续文字组成，缺乏有效的间隔，虽有句、段分隔，但在进行机器语言学习、文本语义理解分析过程中都需以词组为最小单位。因此东亚文本语言实现分词技术相对西方文本语言来讲， 更加的复杂和困难。

在我们的系统中主要用到了 CSW 中文分词组件（http://www.vgoogle.net/Product_CSW.asp）来对文本进行分词，得到一个中文词语的列表。在得到分词结果之后，要对分词的结果列表进行处理，主要是去掉文本中对分类无用的词语，如叹词，助词等，去掉停用词（stop word）和标点符号，合并数字人名等（在附注 A 中我

们列出了我们文本分类器中保留的分词词性），同时获得保留下来的词语的词频。经过这些处理之后的词语列表，就为创建倒排文件做好了准备，下面进入倒排文件的创建过程。

2.1.2. 倒排文件创建

在文本分类中创建倒排文件可以提高训练过程中提取特征向量，计算特征向量权重的效率，因为在倒排文件中可以在 O（1）时间复杂度下提供对特征向量选择，特征权重计算所需要的重要变量值，如每一个关键词的总词频和文件词频，出现某个特征向量的文件的个数等。

现在我们分类程序倒排文件的格式主要有两张表格组成，他们的格式分别如下：

Term	ClassifiedType	Frequency

图 2-2 主倒排文件格式

其中 Term 表示倒排文件中的词语，ClassifiedType 表示倒排文件的这个 term 词语属于的类别，即需要 svm 分类的类别，Frequency 表示 term 词语在 ClassifiedType 类别的文件中出现的频率。

Term	FileName	ClassifiedType	Frequency

图 2-3 副倒排文件格式

其中 Term 表示倒排文件中的词语，FileName 表示这个 Term 所出现的文件名，ClassifiedType 表示出现 Term 的文件所属于的文件的类别，即需要 svm 分类的类别，Frequency 表示 Term 词语在文件 FileName 中出现的频率。

2.2. 特征向量模块

特征向量模块主要是把中文文本转化成用特征向量权重表示的文件，便于 svm 对文本进行分类。这里所说的特征向量其实就是在文本中出现，可以标识文本所属类别的词语。从这些需求出发，就涉及到这个模块的两大功能：（1）抽取特征（2）转化文本为特征表示。

2.2.1. 特征抽取

特征的抽取一般是通过构造一个特征评分函数，把测量空间的数据（分词词语）投影到特征空间，得到在特征空间中的值，然后根据这个值对每个特征进行评估，特征选择就成了选择值最高的若干个特征。在我们的程序中选择词条和类别的互信息作为我们的特征评分函数，公式表示如下：

$$M1(T,Ci) = \log\left(\frac{P(T|Ci)}{P(T)}\right)$$

其中 T 表示词条， C_i 表示类别。其中 $P(C_i \mid T)$ 是 T 在 C_i 中出现的概率，P(T)是 T 在整个训练集中出现的概率。用互信息的方法，在某个类别 C_i 中的出现概率高，而在其它类别中的出现概率低的词条 T，将获得较高的词条和类别互信息，也就可能被选取为类别 C_i 的特征。词条和类别的互信息体现了词条和类别的相关程度，互信息越大，词条和类别的相关程度也越大。最后在每一个类中平均抽取 $MI(T, C_i)$ 的值最大的前 K 个词语，作为这次文本分类的特征向量，即如果要分类 N 类，则特征向量的维度为 N * K.

本文本分类器抽取特征主要分为两步：

1. 计算每个词条在各个类别中的互信息量；
2. 依次抽取每个类别中互信息量最大并且不在特征集合中的词，加入到特征集合中。这样迭代抽取，直到抽取的特征个数等于用户配置的最大特征个数或者抽取了所有特征为止。此时每个类别中抽取的特征个数基本相等。

2.2.2. 文本转化

这个功能就是把中文文本转化成可以被 SVM 分类器识别的特征权重表示。程序的过程主要如下：首先将要分类的文本进行分词，然后判断每一个词语是否属于特征向量，如果这个词语属于特征向量的一个维度，就计算的它的 TF*IDF 的值作为它的特征向量权重。整个过程如下图:

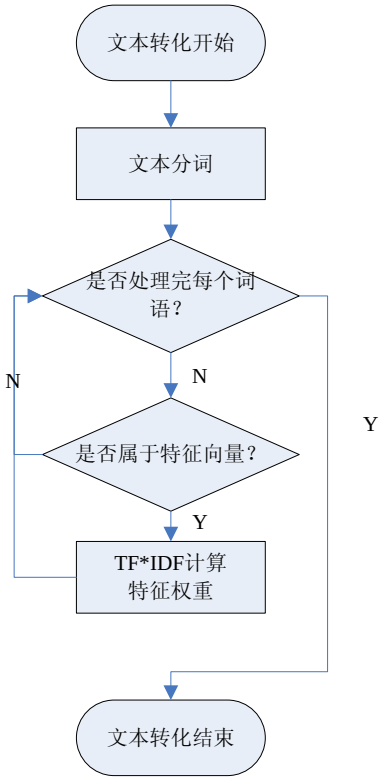

图 2-4 文本转化过程

其中 TF*IDF 的计算主要由下面公式来计算：

$$W_{ik} = TF_{ik} * \log(N / n_k)$$

其中 Tk 是在文档 Di 中的词语（特征向量），TF_{ik} 是词语 Tk 在文档 Di 出现的频率，N 是所有文件的个数，nk 是文档中出现词语 Tk 的文档个数。选择 TF*IDF 作为特征向量权重的计算公式，我们的想法是提高某些可以区别类别但是出现频率又比较小的词语的权重。

2.3. SVM 模块

我们的程序采用开源工具 SVM-Multiclass[5]作为文本分类器，并修改了其中的部分代码，

以适应文本分类输出结果评价的需要。输出结果除了分类信息外，主要包括运行时间、总体分类准确率和各类别分类的准确率。

SVM-Multiclass 是多类别 SVM 的一个实现，它以带权重的向量数据作为输入，可以把数据分成可配置数目的类别。它包括训练和分类两部分。这些正好能满足本文的文本分类器的需求。

3. 实验与总结

3.1. 实验和分析

本文主要完成两系列实验，针对于文本分类器的不同平均特征数目和不同类别，测试其性能和分类正确率。

本文测试数据来源于中文自然语言处理开放平台提供的演示系统文本分类系统（KNN 和 SVM）上的测试文档[4]。

表 3-1 为文本分类器不同平均特征数目测试结果，总共有 10 个类别，SVM 采用线性核函数。从结果看，SVM 学习（Learn）CPU 时间随平均特征数目的增加而显著增加，而分类（Classify）时间基本无多大变化，并且很短。分类正确率随平均特征数目的增加而增加。并且，在 500~1000 之间增加最快，此后慢慢变缓。当每个类别特征数目为 4000 时，分类的正确率为 93.88%，此时各类别的详细结果如表 3-2。从中看出，有些类别分类的准确率已经达到 100%。

表 3-1 文本分类器不同平均特征数目测试（10 个类别，线性核函数）

序号	特征数目/类别	CPU-Seconds without IO (Learn/Classify)	测试文本数	正确文本数	正确率
1	500	3.86/0.01	932	574	61.59%
2	1000	9.28/0.05	932	705	75.65%
3	1500	19.92/0.02	932	755	81.01%
4	2000	34.95/0.00	932	803	86.16%
5	3000	166.02/0.03	932	852	91.42%
6	4000	194.70/0.06	932	875	93.88%

表 3-2 各类别详细结果表（对应表 3-1 每个类别特征数目为 4000 时）

类别	总文档数	分类正确数	分类错误数	其它类别文档错归到该类的数目	正确率
1．环境	67	60	7	0	90%
2．计算机	66	66	0	0	100%
3．交通	71	65	6	1	92%
4．教育	73	66	7	2	90%
5．经济	108	100	8	11	93%
6．军事	83	73	10	9	88%
7．体育	149	149	0	8	100%
8．医药	68	64	4	2	94%
9．艺术	82	76	6	1	93%
10．政治	165	156	9	23	95%

表 3.3 为文本分类器不同类别数目的测试结果，每个类别特征数目为 2000，SVM 也采用线性核函数。从表中数据来看，SVM 学习 CPU 时间随类别数目的增加而增加，而分类时

间基本无多大变化。分类正确率总体来看基本随类别数目的增加而增加，在 2 类时最高。4 类与 5 类之间降低较多，5 类之后基本变化不大。

表 3-3 文本分类器不同类别数测试（每个类别特征数目为 2000）

序号	类别数目	CPU-Seconds without IO (Learn/Classify)	测试文本数	正确文本数	正确率
1	2	0.81/0.00	133	129	96.99%
2	3	4.78/0.00	204	191	93.63%
3	4	7.70/0.00	277	258	93.14%
4	5	5.39/0.00	617	530	86.23%
5	7	17.19/0.03	385	332	85.90%
6	10	34.95/0.00	932	803	86.16%

3.2. 总结

本文实现了一个基于 SVM 的中文文本分类器。从分类结果来看，性能和分类正确率都还不错。

该文本分类器仍有诸多改进之处，如下：

- 目前所选择的分词库并不是很稳定，分出的结果也有提高的可能；
- SVM 核函数为线性核函数，采用其他核函数或许有更高的性能；
- 分词后降维处理主要根据词性和 Stop Words，加入一些现成的算法去掉一些无用的词是可能的。并且从获得的特征来看，的确有一些无用的词语存在；
- 目前采用 SVM-Multiclass，有可能的话，或许可以针对文本分类的特点，自己实现。

附注 A：汉语词性对照表[北大标准/中科院标准]中用于本程序的词性

词性编码	词性名称	注 解
a	形容词	取英语形容词 adjective 的第 1 个字母 •
an	名形词	具有名词功能的形容词。形容词代码 a 和名词代码 n 并在一起
b	区别词	取汉字“别”的声母
f	方位词	取汉字“方”
h	前接成分	取英语 head 的第 1 个字母
i	成语	取英语成语 idiom 的第 1 个字母
j	简称略语	取汉字“简”的声母
k	后接成分	
l	习用语	习用语尚未成为成语，有点“临时性”，取“临”的声母
Ng	名语素	名词性语素。名词代码为 n，语素代码 g 前面置以 N
n	名词	取英语名词 noun 的第 1 个字母
nr	人名	名词代码 n 和“人(ren)”的声母并在一起
ns	地名	名词代码 n 和处所词代码 s 并在一起
nt	机构团体	“团”的声母为 t，名词代码 n 和 t 并在一起
nz	其他专名	“专”的声母的第 1 个字母为 z，名词代码 n 和 z 并在一起
r	代词	取英语代词 pronoun 的第 2 个字母，因 p 已用于介词
s	处所词	取英语 space 的第 1 个字母
v	动词	取英语动词 verb 的第一个字母
vg	动语素	动词性语素。动词代码为 v。在语素的代码 g 前面置以 V
vd	副动词	直接作状语的动词。动词和副词的代码并在一起
vn	名动词	指具有名词功能的动词。动词和名词的代码并在一起
z	状态词	取汉字“状”的声母的前一个字母
un	未知词	不可识别词及用户自定义词组。取英文 Unkonwn 首两个字母。(非北大标准，CSW 分词中定义)

附注 B：用户手册

在使用本文本分类器需把系统时间改为 2007 年或之前，因为本系统使用的 CSW 库免费版截止在 2007 年或之前。

本文本分类器包括 3 个可执行文件：

- **SplitSystem.exe**

对原始文本数据进行分词、特征提取，及把文本以特征向量表示。

命令形式为：

SplitSystem.exe 1|2

参数为 1 时，根据配置文件提取特征，保存在文件中，并生成文本特征向量，同样保存在文件中。

参数为 2 时，根据参数为 1 时生成的特征文件，生成文本特征向量，保存在文件中。

2 比 1 少了提取特征的过程，实际上 1 生成的文本向量用于训练，而 2 是把要分类的文本生成文本向量，进行分类。

SplitSystem.exe 会生成很多结果文件，文件名和路径都可以在配置文件中进行配置（sample\properties.dat）。以程序包中提供的 Demo 为例（生成的文件都在 sample\result\中），参数为 1 时生成特征文件 feature_*_*.dat，倒排索引文件 invertedAuxFile.txt 和 invertedMainFile.txt，文本向量 testName_*_*.dat，文本向量对应文件 trainName_*_*.dat。参数 2 是和 1 相比，只是不会生成特征文件，实际它在生成文本向量时，就是读取参数 1 生成的特征文件。

- **svm_multiclass_learn.exe**

对多类别 SVM 进行训练。这部分基本就是 SVM-Multiclass 官方网站[5]提供的功能，所以具体参数可参考它。

- **svm_multiclass_classify.exe**

进行文本分类。这部分和上一部分类似，不过在生成结果文件中添加了一些统计信息，包括 CPU 运行时间、总的分类正确错误个数及准备率、每一类分类正确错误个数及准确率。这些添加在分类结果文件的末尾，并打印在屏幕上。

配置文件 properties.dat 包含的属性如表 B-1：

表 B-1 文本分类器不同类别数测试（每个类别特征数目为 2000）

属性	说明	例子
featureFileName	特征文件名前缀	..\sample\result\feature
splitComponent	分词库所在目录	..\SplitSystem\component\
invertedMainFile	倒排文件主索引	..\sample\result\invertedMainFile.txt
invertedAuxFile	倒排文件部分索引	..\sample\result\invertedAuxFile.txt
textVectorFile	文本向量文件名前缀	..\sample\result\trainVector
textNameFile	文本向量对照文件名	..\sample\result\trainName
featureMaxNum	所有分类特征总个数最大值	10000
classNum	类别个数	10
0*	各个类别所在的目录，以 01 开始	..\sample\sample\train\计算机\

另处，在目录 bat 下已经提供了一些脚本，包括：

feature.bat：提取特征，同时生成文本向量；

vector.bat：生成文本向量；

learn.bat：训练 SVM-Multiclass；
classify.bat：使用 SVM-Multiclass 进行分类。

参考文献

[1] 秦进，陈笑蓉，汪维家，陆汝占．文本分类中的特征抽取．计算机应用, 23卷2期. 2003.2
[2] 高洁，吉根林．文本分类技术研究．计算机应用研究. 2004
[3] Burges, C.J.C. A tutorial on support vector machines for pattern recognition. Data Mining and Knowledge Discovery, 1998, 2(2): 1-47.
[4] 文本分类系统介绍页面：http://www.nlp.org.cn/docs/doclist.php?cat_id=16&type=12，下载地址：http://www.nlp.org.cn/docs/docredirect.php?doc_id=1023
[5] http://www.cs.cornell.edu/People/tj/svm%5Flight/svm_multiclass.html. 2007.7.1