

文章编号:1001-9081(2009)04-1124-04

一种改进的词序列核算法

徐 峰¹, 罗军勇¹, 温 涛²

(1. 信息工程大学 信息工程学院, 郑州 450002; 2. 61600 部队, 北京 100076)

(Frankday@263.net)

摘要: 在深入研究 Kandol 提出的词序列核(WSK)算法的基础上,提出了一种降低时间复杂度和空间复杂度的文本特征提取算法,并在一个测试集上进行了分类性能测试,结果表明提出的特征提取算法与词序列核算法相比较,在对文本分类性能损失较小的情况下,能够显著地降低特征提取时间、空间复杂度。

关键词: 词序列核; 特征抽取; 核方法

中图分类号: TP391 **文献标志码:**A

Improved word-sequence kernel algorithm

XU Feng¹, LUO Jun-yong¹, WEN Tao²

(1. Institute of Information Engineering, Information Engineering University, Zhengzhou Henan 450002, China;

2. 61600 Group, Beijing 100076, China)

Abstract: Based on the study of the Word-Sequence Kernel (WSK) algorithm put forward by Cancedda, this paper proposed a feature extraction algorithm which could decrease the time and space complexity. A series of classification performance test were carried out, and the experimental results show that, compared with the WSK algorithm, the feature extraction algorithm can reduce the time and space complexity, with less loss of the text classification performance.

Key words: Word-Sequence Kernel (WSK); feature extraction; kernel algorithm

0 引言

数字信息网络日益发展,信息容量急剧增长。面对海量文本信息,人们需要辅助的自动工具来帮助人们更好地发现、过滤和管理这些信息资源。如何将文本信息进行结构化表示是文本处理的关键技术之一。

近年,Cancedda 在字符串核(String Kernel)^[2-3]算法的基础上进一步提出了词序列核(Word-Sequence Kernels, WSK)的文本表示算法^[1],该算法以词(word)为文本表示的最小单元(粒度)。虽然这会极大地增加粒度集合的空间大小,但是对每篇文档而言却可以降低文本表示序列的长度。由于词序列核的计算量仅仅依赖于序列的长度^[1],因此词序列核的文本表示方法能够显著提高计算效率,可以对字符串核不得不采取近似的文档集合进行处理。此外,词序列核的文本表示方法更多地保留了文本中的语义等有用信息。

尽管同字符串核相比,词序列核大大降低了计算量,但是将较长的文章表示为结构化的数据仍需要巨大的计算量。为避免大量的数据运算,本文提出了一种降低时间复杂度和空间复杂度的文本特征提取算法,并就此算法进行了分类性能测试,结果表明本文的特征提取算法与文献[1]中介绍的词序列核算法相比较,在对文本分类性能损失较小的情况下能够显著地降低特征提取时间、空间复杂度。

1 序列核分类算法原理

1.1 核方法及其表示

如图 1 所示,对于分类器我们可以理解为是将一个输入

空间 X 映射到一个类别集合 H 上的函数。但是,对于线性分类器来说,当 X 空间中的样本线性不可分时,线性分类器就无法正确分类。通过升维的办法,在高维空间中使用线性分类器对在 X 空间中线性不可分的样本集进行分类尝试,就有可能解决分类问题。核方法正是使用了这种思想,实现样本空间向高维空间的映射^[2]。核方法具有诸如便于集成先验知识、良好的过拟合控制以及高效的计算等优点,当与支持向量机(Support Vector Machine, SVM)相结合,其优势能够得到充分体现^[4]。

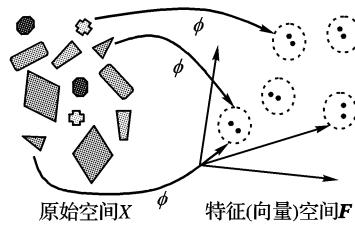


图 1 映射关系

目前文本绝大部分是基于一个标准的 bag of words 由词频(或者附加加权和归一化)来表示的。对字词位置信息的忽略而导致的信息损失远大于在向量空间中使用强有力的方法带来的信息量补偿。尽管一些方法引入了短语、多词单元或者位置联合出现统计的概念,但是其基本的文本表示仍然是基于向量空间描述的^[5-6]。

字符串核和词序列核有别于向量空间模型的文本表示方法,它通过计算两个序列共享的匹配子序列的个数来得到两个序列的相似性。对于非连续的序列,根据序列内包含的间隔数量对序列核进行惩罚。

收稿日期:2008-10-28;修回日期:2009-01-04。

作者简介: 徐峰(1978-),男,辽宁丹东人,工程师,硕士研究生,主要研究方向:计算机软件与理论、人工智能软件; 罗军勇(1964-),男,江西南昌人,教授,主要研究方向:计算机软件与理论、信息安全; 温涛(1977-),男,陕西西安人,工程师,博士,主要研究方向:模式识别、图像处理、信号分析。

1.2 词序列核

由于词序列表示方法的粒度是词(word),而字符串序列的粒度是字母,这会极大地增加构成词序列的元素的空间 $|\Sigma|$,数量级从几百增加到几万,特征空间的维数也相应增加。

但是对每篇文章而言,一个文本不可能用到所有的词,通常一个文本只包含了少部分词,即文本表示向量的取值非常稀疏,数据集的内在维数远低于样本向量空间的维数,因此应当对其进行维数约简。维数约简的目标是从高维输入数据集中发现内在低维特征集,从空间的角度来说,就是寻找原向量空间的一个低维子空间,并在此子空间中尽可能多地保留有用信息。

目前的维数约简方法大体可分为两类:特征选择(Feature Selection)和特征抽取(Feature Extraction)。

特征选择是根据某种准则从原始特征中选择部分最有类别区分能力的特征。

特征抽取是依据某种原则构造从原始特征空间到低维空间的一个变换,从而将原始特征空间所包含的分类信息转移到新的低维空间中来。

从隐含语义分析的角度考虑,通过矩阵的不同分解和化简来获取将向量语义或统计信息向低维空间压缩的线性映射,如差量(differential) LSI^[7-8]、主成分分析^[9]、线性判别分析^[10]和概念索引^[11]等。从信息熵角度,研究项分布相似性的聚类方法,如基于全局信息(Global Information, GI)的词项聚类等。

常用的特征提取与特征选择算法的效果在不同情况下互有高低或相当。虽然选择方法因为复杂度较低而应用更为广泛,但提取方法得到的特征更接近文本的语义描述,因此有较大的研究价值。

对于一篇文章而言,词序列可以大大降低符号序列的平均长度。而序列核的计算量依赖于符号序列的长度,因此词序列核方法较字符串核方法计算量明显减少,可以显著提高计算效率。词序列核能够在字符串序列核不得不取近似的数据集上进行计算^{[1]1059}。

使用不同的 λ 值是将先验知识纳入序列核中的途径之一^{[1]1066}。以前的序列核,包括字符串核在内仅仅依赖于事件发生的频度或者检索词频,这种方式缺少特征加权,而特征加权对于检索很重要。词序列核方法对标准的序列核方法进行了两个扩展:第一个扩展是给间隙赋予不同的 λ 权值,允许使用现成的加权方法,例如逆文档频率(Inverse Document Frequency, IDF);第二个扩展是对间隙和符号匹配采用不同的权重。对于富含信息的符号,根据是被用于匹配还是间隙中而区别对待,扩展可以结合已有词性信息和IDF综合使用。我们希望对包含高度相关的符号(如:名词)的间隙施以严厉的处罚(赋予小的 λ_c),而对包含高度相关匹配的符号给予奖励(也就是赋予大的 λ_c)。这个要求在以前的核序列处理中无法实现。一种解决方法是,对间隙和匹配分别引入两个单独的权重集合。进行简化,可以根据词性来衰减间隙,而根据IDF信息来加权匹配符号。

在介绍词序列核之前,先定义几个符号: Σ 是一个由词构成的集合;词序列是由 Σ 中的元素构成的序列; Σ^n 表示长度为 n 的词序列;对于词序列 $s, t, |s|$ 表示序列 s 的长度, st 表示将 s 和 t 连接起来,序列 $s[i:j]$ 是 s 中的子串: $s[i:j] = s_i s_{i+1} \cdots s_j$ 。

我们令 u 是 s 中的子串,令 $i = (i_1, i_2, \dots, i_{|u|})$ 是 u 在 s 中的下标序列,即 $u = (s_{i_1}, s_{i_2}, \dots, s_{i_{|u|}})$;再定义 $l(i) = i_{|u|} - i_1 + 1$ 是子序列 u 在 s 中的跨度;对一个序列 s 的特征映射,由定义在 u 坐标系上的 $\varphi_u(s)$ 来表示, $\varphi_u(s) = \sum_{i: u=s[i]} \lambda^{l(i)}, 0 \leq \lambda \leq 1$ 。则由所有的子序列($u \in \Sigma^n$)的特征值的积的和表示序列 s 和 t 的特征向量的内积。

现在定义两个序列 s, t 的经过加权的序列核:

$$K_n(s, t) = \sum_{u \in \Sigma^n} \hat{\varphi}_u(s) \hat{\varphi}_u(t) = \sum_{u \in \Sigma^n} \sum_{i: u=s[i]} \prod_{j: u=t[j]} \lambda_{m, u}^2 \prod_{i_1 < k < i_{|u|}, k \notin i} \lambda_{g, s_k} \prod_{j_1 < l < j_{|u|}, l \notin j} \lambda_{g, t_l} \quad (1)$$

对于 K 的估计,我们定义 K' 和 K'' :

$$K'_i(s, t) = \sum_{u \in \Sigma} \sum_{i: u=s[i]} \sum_{j: u=t[j]} \prod_{k=1}^i \lambda_{m, u}^2 \prod_{i_1 < l \leq i, l \notin i} \lambda_{g, s_l} \prod_{j_1 < r \leq j, r \notin j} \lambda_{g, t_r} \quad (2)$$

$$K''_i(sx, t) = \sum_{j: t_j=x} K'_{i-1}(s, t[1:j-1]) \lambda_{m, x}^2 \prod_{l=j+1}^{|t|} \lambda_{g, t_l}; \quad i = 1, \dots, n-1 \quad (3)$$

使用字符串核中的递推公式^[2],并将式(2)~(3)代入后,则得到下式^[1]:

$$\begin{cases} K_n(s, t) = 0; \min(|s|, |t|) < n \\ K_n(sx, t) = K_n(s, t) + \sum_{j: t_j=x} \lambda_{m, x}^2 K'_{n-1}(s, t[1:j-1]) \end{cases} \quad (4)$$

该公式表明为匹配符号和间隙符号分别考虑权重集合并不影响词序列核计算的复杂度。

1.3 问题

这种基于核方法的文章分类算法的主要缺陷仍然是速度过慢与存储膨胀。在一篇文章的长度为 L ,每次抽取 N 个字的情况下,使用这种方法进行文章特征提取所获得的特征个数为:

$$K = C_L^N \quad (5)$$

随着文章长度 L 和抽取长度 N 增加,特征个数以指数级进行增长,运算时间和存储空间均以近似指数级增长。

以空间消耗为例,当文章长度为100、抽取长度为3时,特征个数 K 为: $K = \frac{100 \times 99 \times 98}{3 \times 2 \times 1} = 16170$ 。

若汉字采用utf16编码,每个汉字占用2B空间,则其消耗的空间 M 为: $M = 16170 \times 3 \times 2 = 0.925 \text{ MB}$ 。

当文章的长度为1000、抽取长度为4时,特征个数 K 为: $K = \frac{1000 \times 999 \times 998 \times 997}{4 \times 3 \times 2 \times 1} = 41417124750$ 。

消耗的空间 M 为: $M = 41417124750 \times 4 \times 2 = 308.6 \text{ GB}$ 。存储空间占用过大,难以应用到实际情况中。

2 对词序列核方法的改进

测试过程中发现,采用词序列核方法对文章分类的运算时间同文章的长度存在着近似指数函数的变化,即随着文章长度的增加,运算时间增长得飞快,当文章长度增加到500字以上的时候,运算时间过长,达到无法忍受的程度。针对这一问题,我们考虑了一种方法对其进行改进。

传统的词序列核将文章视为一个关系/概念字串,字与字、词与词仅有两种关系,即匹配和间隙,没有包含句子这个概念。也就是说只要它们是匹配的,即使分割在两个不同句子之间,同样会被赋予一个较高的权重;如果它们是间隙的,即使它们之间仅仅间隔了一系列修饰从句也会被赋予一个低

的权重。在语言上很重要的“句子”信息就在这种方法中丢失了。

在汉语中,当两个字存在于不同句子中的时候,其相关性很低,包含的语义信息很少,不足以代表该文章的特性。例如下文:“随着互联网的普及,搜索引擎已经成为人们快速查找信息和资源的重要手段。但目前的搜索引擎主要采用基于关键字的查询。”抽取出的“普查”是一个词,但是它们之间的相关性基本没有,包含的语义信息近似于零,这样的词完全不能代表该文章的特征,也就没有必要将其放入核中进行计算。因此可以考虑通过标点对取词进行截断,在保证信息丢失很少的基础上,减少运算时间和存储空间。即抽取过程中以句子结束符号(包括“。?!等,表示一个句子结束的标点)作为标准,将每一句话作为一个抽取的基本单位进行抽取。

经过改进,词序列核方法的性能有了较大提高。假设一篇文章的长度为 L ,其中句子有 S 个,则每个句子的平均长度为 L/S 。再假设,特征抽取长度为 N ,对长度为 L/S 的文字进行抽取需要的时间为 T ,消耗的存储空间为 M 。这样,对这篇文章进行抽取时,其时间复杂度的最优情况是文章中的所有文字平均分布在每个句子中,也就是每个句子的长度均相同,且为 L/S ;最差情况是此文章中的所有文字集中在一个句子中,该句子长度为 L ,而其他的句子长度均为 0。

在最优情况下,抽取的特征个数为 $S \times C_{L/S}^N$,运算时间为 ST ,消耗的存储空间为 SM ,抽取花费的时间与消耗的空间均与文章长度成正比;最差情况下,其消耗的时间和空间均与原始的词序列核算法相同。

以存储消耗为例,最优情况下,当文章长度为 100,句子数目为 10,抽取长度为 3 的时候,其产生的特征个数 K 为: $K = 10 \times \frac{10 \times 9 \times 8}{3 \times 2 \times 1} = 4800$ 。

若汉字采用 utf16 方式编码,每个汉字占用 2 B,消耗的存储空间 M 为: $M = 4800 \times 3 \times 2 = 28.125$ kB。

当文章的长度为 1000,句子数目为 10,抽取长度为 4 时,产生的特征个数 K 则为: $K = 10 \times \frac{100 \times 99 \times 98 \times 97}{4 \times 3 \times 2 \times 1} = 39212250$ 。

消耗的存储空间 M 为: $M = 39212250 \times 4 \times 2 = 299.16$ MB。

可以看到,其存储消耗量远远小于文献[1]提出的词序列核算法。

3 词序列核的计算与性能对比

3.1 改进目标

这篇文章,我们想回答以下几个问题:

- 1) 对一篇文章进行依赖于标点进行截断的抽取的性能是否会比全文抽取有较大的性能提高?
- 2) 依赖标点进行截断的抽取获得的核是否比全文抽取获得的核在分类准确率上有明显的降低?
- 3) 依赖标点进行截断的抽取获得的核能否大幅降低对于机器性能的要求使得该算法具有更高的实用价值?

3.2 实验条件

实验使用计算包,进行一定改动以适应序列核的计算应用。所有类别的训练集都或多或少的存在负例的不均衡性,这容易导致过学习。提供了一个参数去给训练集中的正例和负例样本的相对重要性进行加权。作为一种启发性的规则,所有的实验都使用此参数集合运行,此参数集合为最接近于

训练集中的负例和正例的比值的整数。

实验主机配置为奔腾双核 1.6 GHz CPU,1 GB 内存,Windows XP Professional 2002。

3.3 抽取性能比较

对于单篇文章,抽取长度为 3 的时候,依赖于符号截断的抽取和全文抽取性能比较,如表 1 所示,其中。文章长度为中文字符个数,抽取时间单位为 ms。

表 1 抽取性能对比

文章长度	依赖符号截断的抽取		全文抽取	
	抽取时间/ms	生成词条数	抽取时间/ms	生成词条数
166	6656	22821	166125	166276
258	17953	43032	334922	345621
315	60328	95235	1276265	855119
653	94563	156470	4213265	1906637

抽取性能比较如图 2 所示。

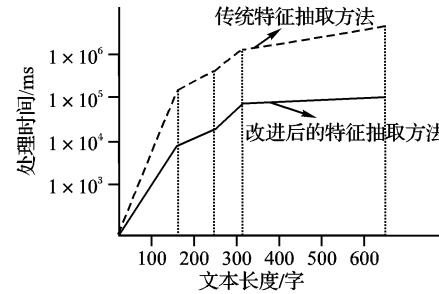


图 2 抽取性能比较

同时,由于计算过程的空间复杂度同生成的词条数目成线性关系,因此,可以通过观察词条数目直观的看到二者对存储空间消耗的差别。

3.4 分类能力比较

对下面的实验结果,我们采用标准的信息检索性能度量方法。从测试分类结果中,我们计算正确的正例 TP (对于正例的测试样本,模型鉴别的结果也是正例的文章的数目),错判的负例 FP (对于负例的测试样本,模型鉴别的结果却是正例的文章的数目,漏警),错判的正例 FN (对于正例的测试样本,模型鉴别的结果却是负例的文章的数目,虚警)。这样有下述的性能度量指标:

精度 正确的正例除以判出的正例:

$$p = TP / (TP + FP) \quad (6)$$

召回率 正确的正例除以应有的正例:

$$r = TP / (TP + FN) \quad (7)$$

F 评价 精度和召回率的调和平均:

$$F_1 = 2pr / (p + r) \quad (8)$$

针对新浪网上的文章,我们抽取 2000 篇文章作为训练集,另取 1000 篇文章作为测试集,我们只考虑一种分类的话,如表 2 所示。

表 2 待测目标

分类	训练集	负例/正例	测试集
财经	2000	22.81	1000

对文章进行预处理,去除数字、英文以及其他无用符号(无用符号包括汉语中一些无特定意义的字符,如“的”、“了”等),仅保留汉字和句子结束符号。经过预处理后的文章平均长度为 192。

理论上, λ_m 和 λ_g 均依赖不同的词性以及位置信息取不

同的值具有最好的效果^[1],但在中文中,完全采用变 λ_m 难以实现,因此我们对抽取的词进行了简单判定:若为一个真实存在的词,依据词性不同(对于具有多重词性的词则优先取高权值)给予不同的权重;对于本不存在于汉语中的词则给予统一的 $\lambda_m = 0.6$,这样得到的实验结果如表3所示。

表3 抽取效果对比

实验参数	依赖符号截断的抽取			全文抽取		
	p	r	F_1	p	r	F_1
变 λ_m , $\lambda_k = 1 - \lambda_m, \mu = 0.5$	89.26	72.3	80.78	90.63	71.20	80.92

3.5 实验结果分析

通过上面的实验,我们对两种抽取方式进行了性能和效果上的测试比较,通过测试比较发现,基于标点符号的词的抽取在时间复杂度上比全文直接抽取有数量级上的降低,在空间复杂度上也具有极大的降低;在效果上,其精度比全文抽取有了一定地降低,而召回率则有一定地提高,总体上讲,两者的效果差别不大。因此可以采用这种方法大大地降低词序列核方法进行文章分类对于计算时间、空间的占用量,使得这种方法具有更大的实用性。

4 结语

由于文献[1]算法其时间复杂度和空间复杂度均会随着文章长度的增加快速增长,因此对于较长文章的处理就十分困难。本文在深入研究文献[1]的词序列核算法的基础上提出了一种改进方法,该方法利用标点符号对文章进行分段特征提取。此算法的时空复杂度与文章长度的变化在最优情况下为一种线形关系,而在最差情况下同原始的词序列核算法相同。最后,我们利用此改进的词序列核特征提取算法结合SVM算法进行了文本分类性能测试,结果表明,在分类性能损失有限的情况下,显著提高了分类的效率。

基于核函数的文本特征提取算法相对于向量空间模型等其他方法而言速度是一个关键问题,如何实现快速有效的计算是当前亟待解决的问题,此问题可以通过一些预处理技术(例如:好的中文分词方法),进一步降低其时间和空间复杂

(上接第1119页)

2)并行爬虫系统页面更新的峰值通常要超过午夜12点后才会回落。根据数据观察可知,这是由于18:00后的时间段访问站点的人数增加,并且校园网网络拥挤度增大,限制了爬虫更新的速度,因此延长了更新的时间段。

5 结语

目前,针对搜索引擎的并行爬虫系统正在受到越来越多的关注。本文描述了一个增量更新的并行Web爬虫系统。通过将夹角余弦向量法与增量更新思想相结合,使得增量更新的并行Web爬虫系统的抓取效率得以提高,同时通过对网页更新度的预估,减少了爬虫系统更新页面的工作量。

参考文献:

- [1] KIM S J, LEE S H. An empirical study on the change of Web pages [C]// Proceedings of the 7th Asia-Pacific Web Conference on Web Technologies Research and Development: APWeb 2005, LNCS 3399. Heidelberg: Springer-Verlag, 2005: 632–642.
- [2] 北大网络实验室. Web InfoMall[EB/OL]. [2008-08-11]. <http://www.infomall.cn/>.
- [3] CHO J, GARCIA-MOLINA H. Parallel crawlers[C]// Proceedings

度。这些是在后续的研究中的主要内容。

参考文献:

- [1] CANCEDDA N, GAUSSIER E, GOUTTE C, et al. Word-sequence kernels[J]. Journal of Machine Learning Research, 2003, 3: 1059–1082.
- [2] LODHI H, SAUNDERS C, SHAWE-TAYLOR J, et al. Text classification using string kernels[J]. Journal of Machine Learning Research, 2002, 2: 419–444.
- [3] SUN A X, LIM E P, NG W K. Performance measurement framework for hierarchical text classification[J]. Journal of the American Society for Information Science and Technology, 2003, 54(11): 1014–1028.
- [4] 陆振波, 章新华, 康春玉. 基于支持向量机的水中目标识别[J]. 信息与控制, 2003, 32(z1): 739–742.
- [5] PEREZ-CARBALLO J, STRZALKOWSKI T. Natural language information retrieval: Progress report[J]. Information Processing and Management, 2000, 36(1): 155–178.
- [6] WONG S K M, ZIARKO W, WONG P C N. Generalized vector space model in information retrieval[C]// Proceedings of the ACM SIGIR Conference on Research and Development in Information Retrieval: SIGIR'85. New York: ACM Press, 1985: 18–25.
- [7] CHEN L, TOKUDA N, NAGAI A. A new differential LSI space-based probabilistic document classifier[J]. Information Processing Letters, 2003, 88(5): 203–212.
- [8] KIM H, HOWLAND P, PARK H. Dimension reduction in text classification with support vector machines[J]. Journal of Machine Learning Research, 2005, 6(1): 37–53.
- [9] JOLLIFFE I T. Principal Component Analysis[M]. New York: Springer-Verlag, 1986.
- [10] MARTINEZ A M, KAK A C. PCA versus LDA[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2001, 23(2): 228–233.
- [11] KARYPIS G, HAN E H. Fast supervised dimensionality reduction algorithm with applications to document categorization and retrieval [C]// Proceedings of the 9th ACM International Conference on Information and Knowledge Management: CIKM'00. New York: ACM Press, 2000: 12–19.

of the 11th International Conference on World Wide Web: WWW 2002. New York: ACM Press, 2002: 124–135.

- [4] 孟涛, 王继民, 同宏飞. 网页变化与增量搜集技术[J]. 软件学报, 2006, 17(5): 1051–1067.
- [5] 沈文勤, 李庆超, 邵志清. 搜索引擎的渐增式爬行和备份式更新模式[J]. 华东理工大学学报, 2004, 30(3): 284–287.
- [6] 程菲, 汪建海, 罗健. 增量更新Crawler进行Web收集方法研究[J]. 计算机工程与科学, 2006, 28(12): 28–30.
- [7] CHO J, GARCIA-MOLINA H. The evolution of the Web and implications for an incremental crawler[C]// Proceedings of the 26th International Conference on Very Large Databases. San Francisco: Morgan Kaufmann Publishers, 2000: 200–209.
- [8] FETTERLY D, MANASSE M, NAJORK M, et al. A large-scale study of the evolution of Web pages[C]// Proceedings of the 12th International Conference on World Wide Web. New York: ACM Press, 2003: 669–678.
- [9] SALTON G, BUCKLEY C. Term-weighting approaches in automatic retrieval[J]. Information Processing and Management, 1998, 24(5): 513–523.