动态文本分类中概念漂移问题的解决算法研究

闫 鹏[1,2] 郑雪峰[1] 李明祥[1] 曾广平[1]
(北京科技大学信息工程学院 北京 100083)[1] (国家信息中心 北京 100045)[2]

摘 要 以当前的"消极学习型分类法"加"动态更新训练集"的组合模式,不足以解决好动态文本分类中的概念漂移问题。为此,受消极分类法基本思想的启发,并借鉴 k-NN 算法的优点,提出了针对概念漂移问题的"消极特征选择模式"的概念和基于此模式的动态文本分类算法。测试结果表明,新算法很好地解决了当前存在的难点问题,具有高可靠性、高实用性等优点。

关键词 文本分类,概念漂移,消极学习,特征选择

Lazy Feature Selection Approach to Concept Drift in Dynamic Text Classification

YAN Peng[1,2] ZHENG Xue-feng[1] LI Ming-xiang[1] ZENG Guang-ping[1]
(College of Information Engineering, Beijing University Science and Technology, Beijing 100083, China)[1]
(State Information Center, Beijing 100045, China)[2]

Abstract The current dynamic text classification mode of lazy learning algorithms plus updating training set can't deal with concept drift well. So the paper issued a new approach based on lazy feature selection (LFS), a new mode derived from the main idea of lazy learning algorithm. This new approach also adopted many advantages of k-NN, and achieved excellent effect to resolve concept drift in dynamic text classification.

Keywords Text classification, Concept drift, Lazy learning, Feature selection

1 引言

1.1 静态文本分类与动态文本分类

文本分类问题,从总体上可以分为静态的和动态的两大类。在静态的文本分类中,文本的类别特征不会变化。而在动态的文本分类中,文本的类别特征随时间的推移在不断地发生变化,即存在着"概念漂移"现象。

所谓概念漂移(Concept Drift)是指这样一种现象:某些文本的类别特征通常依赖于它内部所包含的一些"隐性内容(hidden context)"。这些隐性内容随着时间的推移在不断发生着微小而隐蔽的变化,事先难以预知,事后也不易觉察,但是,当这些变化积累到一定程度时,却会引起整个目标概念(target concept)发生改变。目前,这种概念漂移现象在垃圾邮件判别问题中普遍存在[1]。

因为存在着概念漂移现象,所以动态文本分类问题比较复杂。分类算法如果不能处理好这个问题的话,其性能就会不断下降,直到无法正常使用。

1.2 积极学习型分类法与消极学习型分类法

在各种基于机器学习的文本分类算法中,可以分为积极学习型(eager learner)和消极学习型(lazy learner)两大类方法[2],以下分别简称为积极分类法和消极分类法。积极分类法包括朴素贝叶斯法(NB)、支持向量机法(SVM)等。这类算法一般包括"由训练数据建立分类模型(归纳步)"和"将模型应用于查询案例(演绎步)"两个步骤。消极分类法则是将对训练数据的建模推迟到需要对查询案例(query case)进行分类时再进行,当且仅当查询案例和某一训练案例匹配时才进行分类,消极分类法以 k-最近邻(k-NN)算法最具代表性。

积极分类法擅长于静态文本分类问题。对于含有"概念漂移"现象的动态文本分类问题,以 k-NN 为代表的消极分类法则受到了青睐,这是由于消极分类法的训练案例集非常易于更新,因此它在处理概念漂移问题时具有很多积极分类法不可比拟的优势[3,4]。

需要强调,上述两大类算法在实际应用时,都必须首先进行特征选择(Feature Selection),以降低向量空间模型(VSM)的特征空间的维度,避免发生"维灾难"。

1.3 问题分析

针对含有概念漂移现象的动态文本分类问题,用消极分类法来应对的基本思想是恰当的,但在具体方法上,目前的研究工作都局限在"以训练案例集的不断更新来应对"的做法上,这样做虽有一定效果,但也存在很大缺陷,因为它忽视了一个更加重要的方面,即 VSM 特征空间的更新问题。

我们知道,通过追踪新案例中出现的一些新的特征词句,就可以发现概念漂移的痕迹。但是,如果没有特征空间的更新,这些新特征词就很难进入到特征空间之中,它们所承载着的概念漂移的重要信息也就不能发挥作用。当然,这样的特征空间也不能反映类别特征的新变化。

特征空间是各种分类算法运行的基础平台,如果它与类别特征的变化相脱节,分类算法对概念漂移的敏感度就会不断下降。此时,即使通过训练案例集的动态更新,可以对解决概念漂移问题有所帮助,但所起的作用也必然十分有限,因其

闫 鹏 博士研究生,高级工程师,CCF 会员,主要研究领域为计算机应用、网络安全;郑雪峰 博导,教授,主要研究领域为网络与信息安全;李明祥 博士研究生,主要研究领域为网络安全;曾广平 博导,教授,主要研究领域为 Linux 操作系统内核重构及应用、智能网络与智能通信等。

有“舍本逐末”之弊病。

所以，特征空间的动态更新是追踪和解决好概念漂移问题的前提和基础，其重要性要远大于训练案例集的更新。文献[1]可能也觉察到了这一点，所以在结论部分提出了“每天对训练案例集进行一次更新”的同时，要“定期重新进行特征选择”(即更新特征空间)的做法。

但是，以这种“定期重新进行特征选择”的方法来解决特征空间的动态更新问题，未免过于简单，原因在于：

1)以现有的特征选择模式，所有的特征词都要经过信息增益(IG)或互信息量(MI)等评估函数的统一筛选之后，分值高者才有可能被选中。但是，在开始阶段，含有新特征词的文本案例自然是极少数，相应地，这些新特征词的 IG 或 MI 分值必然很低，它们很难通得过评估函数的筛选而被及时选中，所以，特征空间的更新必然滞后。

2)特征选择对分类系统而言，开销相当大，频繁地重做特征选择，系统难以容忍。

3)周期性地重做特征选择是一个解决办法，但是，周期的长短不易把握：周期过短，会出现前面两点的问题，周期过长，更会使特征空间的更新明显滞后。

因此，以目前的处理模式，并不能解决好特征空间的更新问题，进而使得概念漂移问题的解决也受到了极大影响。

针对上述情况，我们受消极分类法基本思想的启发，另辟蹊径，提出了解决动态文本分类中概念漂移问题的“消极特征选择模式”和基于此模式的分类算法。本文第 2 节对 k-NN 算法的欧氏距离分类函数进行了剖析，提出了简单而高效的基于“向量相似度”的新分类函数；在第 3 节中，以新分类函数为依据，提出了“消极特征选择模式”的基本思想，详细描述了基于此模式的分类算法的主要步骤。最后，通过仿真实验，对新算法的性能和效率进行了测试。

2 基于向量相似度的分类函数

本节中，首先确定适宜的文本向量属性值的表示形式，然后对 k-NN 算法的“欧氏距离”分类函数进行分析，最后提出简单高效的新分类函数。

2.1 文本向量属性值的表示方法

文本向量属性值的表示方法，概括起来有两种：一是布尔(Boolean)表示法，即“1”表示某属性(特征词)在此文本中已出现，“0”表示未出现；另一种是数值(Numeric)表示法，通常用特征词在文本中出现的频率(Term Frequency，TF)来表示。

文献[5,6]对这两种表示方法的实际效果进行了分析和对比实验，结果表明，数值表示法虽然增加了权重信息和复杂度，但实际效果却并没有明显提高。相比之下，布尔法更加简单可行，而且，文献[6]认为布尔法利于提高案例的检索速度，所以，本文采用布尔法表示文本向量的属性值。

2.2 欧氏距离分类函数的不足

在 k-最近邻(k-NN)算法中，一般以欧氏距离表示向量间的相似度。欧氏距离的定义如下：

定义 1 设在 n 维空间中两个点 $X=(x_1,x_2,\cdots,x_n)$ 和 $Y=(y_1,y_2,\cdots,y_n)$，它们之间的欧氏距离定义为：

$$d(X,Y)=\sqrt{\sum_{k=1}^{n}(x_k-y_k)^2} \tag{1}$$

其中，n 是维数，x_k 和 y_k 分别是 X 和 Y 的第 k 个属性的值。

k-NN 算法根据公式在训练案例集中检索出与查询案例最为接近的 k 个训练案例，然后在这 k 个训练案例中，再进一步找出包含训练案例数最多的类别，并以此类别来决定查询案例的类别。

显然，k-NN 算法用向量间的相异度(距离)来表示向量之间的相似性，相异度越小，两个向量就越相似。这种分类算法比较准确，但检索速度很慢，它虽然不需要建立学习模型，但在进行分类时，查询案例要按照公式，与训练案例集中的训练案例一一匹配，计算量之大是显而易见的，所以速度很慢也并不奇怪。

事实上，特征空间包含的属性(特征词)虽然成千上万，但对于每一个具体的文本向量而言，绝大多数只可能包含特征空间的一小部分属性，即只有一小部分属性的值为非“0”，其余绝大部分属性的值都为“0”。因此，文本向量通常是非常稀疏的，如果能通过适当的压缩方法，有效地剔除掉这些占绝大多数的“‘0’值属性”，必然会大大提高分类算法的效率。

但是，欧氏距离的计算法则是两个向量的相同属性之间“先相减、再平方”，最后再计算总和的平方根，因此要去除这些“0”值属性比较困难。

所以，在本文中，我们借鉴 k-NN 算法的主要思想，但放弃了以欧氏距离为分类函数的做法，转而寻求“以相似度来表示向量间的相似性”的分类函数。我们希望新的分类函数能够明显提高分类法的效率，并对解决概念漂移问题有所帮助。

2.3 基于向量相似度的新分类函数

关于文本向量间的相似性，除了相异度之外，还可以用相似度来表示，自然，相似度值越大，二者越相似。通常使用余弦相似度(cosine similarity)来度量文本向量的相似性，它的定义如下[7]：

定义 2 如果 X 和 Y 是两个文本向量，$X=(x_1,x_2,\cdots,x_n)$，$Y=(y_1,y_2,\cdots,y_n)$，则二者的相似度为：

$$\mathrm{sim}(X,Y)=\cos(X,Y)=\frac{X\cdot Y}{\|X\|\ \|Y\|}=\frac{\sum_{k=1}^{n}x_k y_k}{\sqrt{\sum_{k=1}^{n}x_k^2}\sqrt{\sum_{k=1}^{n}y_k^2}} \tag{2}$$

其中，$\|X\|$ 和 $\|Y\|$ 分别表示文本向量 X 和 Y 的模，$X\cdot Y$ 表示二者的内积。

从式(2)可知，两个文本向量的相似度，与二者自身的模以及它们的内积有关。而内积的计算法则是两个向量相同属性的属性值两两相乘再求总和。我们知道，只要一方的某一属性值为“0”，那么这个属性的乘积必然为“0”，它对内积的大小就没有任何贡献，所以内积的计算，只需考虑两个向量中同时为非“0”的那些属性即可。

又因为在 2.1 节中已经说明，本文用布尔法(1/0)表示文本向量的属性值，因此，对式(2)可做如下转换：

$\|X\|$：在数值上相当于“X”中所含的非“0”属性个数的平方根，用 $\sqrt{f_x}$ 表示。

$\|Y\|$：在数值上相当于“Y”中所含的非“0”属性个数的平方根，用 $\sqrt{f_y}$ 表示。

$X\cdot Y$：在数值上相当于“X”，“Y”中相同属性的属性值同时为非“0”的属性个数，用 f_{xy} 表示。

所以，式(2)可以转换为：

$$\mathrm{sim}(X,Y)=\frac{f_{xy}}{\sqrt{f_x f_y}}\propto\frac{f_{xy}^2}{f_x f_y} \tag{3}$$

与式(1)相比,式(3)的计算量大大降低。不仅如此,式(3)还为下面提出的"消极特征选择模式",提供了理论上的根据。

3 消极特征选择模式与基于此模式的分类算法描述

在引言中,我们谈到,目前的特征选择模式使 VSM 特征空间的更新明显滞后,影响了概念漂移问题的解决,亟需改进。本节中,为了解决这一问题,我们受消极分类法思想的启发,并依据式(3),提出了"消极特征选择模式"和基于此模式的动态文本分类的新算法。

为便于表述,本文用 Lazy Feature Selection (LFS)来表示"消极特征选择模式"。LFS 基本原理的介绍及推导过程如下:

设在 n 维特征空间中,$Y=(y_1,y_2,\cdots,y_n)$表示查询案例;$X_i=(x_{i1},x_{i2},\cdots,x_{in})$表示训练案例集中的第 i 个案例,从公式可知:X_i与 Y 的相似度与 f_x,f_y,f_{xy}三个因素密切相关。因为 f_x,f_y 在进行原始文本抽词时即已确定,所以只能对 f_{xy}做些文章。

f_{xy}表示 X_i与 Y 中属性值同时为非"0"的属性个数。同时,f_{xy}也可以看成是"在以 Y 中的非'0'属性组成的特征子空间中,X_i 所含的非'0'属性的个数"。或者说,如果我们以 Y 中所含的非"0"属性组成一个过滤窗口,那么,f_{xy}就相当于在这个窗口中,X_i 所含的非"0"属性的个数。

因此,根据式(3),我们可以用查询案例 Y 中所含的全部非"0"属性组成一个"特征选择的过滤窗口",并以此窗口来过滤 X_i 中的属性。

经过这样的特征选择,可以首先得到:在新的特征空间$(y_1,y_2,\cdots,y_n)$中,训练案例 X_i 包含的属性子集;进而,可以容易地再得到:在 X_i 的这个属性子集中,X_i 自身所含的非"0"属性的个数,即"f_{xy}"。

因为这种特征选择模式,不是发生在建立 VSM 之初,而是发生在需要对具体的查询案例"Y"进行分类时才进行,而且以 Y 中所包含的非"0"属性为特征选择的过滤窗口,这恰与消极学习(Lazy Learner)的思想和方法非常相似,所以我们称之为"消极特征选择模式(LFS)"。下面将基于 LFS 模式的分类算法详细描述如下:

Step 1 在各原始训练文本中,抽取特征词,不必进行特征选择,全部输入训练案例集。训练案例集向量采用压缩形式表示,即:只列出对应的原始文本中实际出现的特征词。所以,第 i 个训练案例 X_i 可以表示为:

$$X_i=(\langle x_{i1},w\rangle\langle x_{i2},w\rangle\cdots\langle x_{imi},w\rangle\langle M_i,m_i\rangle) \tag{4}$$

在压缩模型中,m_i 为 X_i 对应的原始文本中实际包含的特征词(即属性)的个数;x_{ij}表示其中的第 j 个属性,$j\in\{1\cdots m_i\}$;w 为属性值,因为压缩模型中只包含真正出现的属性,且采用布尔法表示属性值,所以有"$w\equiv1$";M_i 为 X_i 的模的平方,根据模的定义,有"$M_i=m_i\cdot w^2=m_i$"(因为 $w\equiv1$)。

Step 2 当有查询案例 Y 到来时,参照 Step 2 的方法处理,将 Y 表示为:

$$Y=(\langle y_1,w\rangle,\langle y_2,w\rangle\cdots\langle y_n,w\rangle,\langle M_y,n\rangle) \tag{5}$$

其中,n 为 Y 中实际包含的特征词(即属性)的个数;y_j 表示第 j 个属性;w 为属性值,且有"$w\equiv1$";M_y 为 Y 的模的平方,有"$M_y=n\cdot w^2=n$"。

Step 3 以 Y 中所含的全部非"0"属性$(y_1,y_2,\cdots,y_n)$构成一个新的特征空间。

Step 4 在新特征空间$(y_1,y_2,\cdots,y_n)$中,容易得到训练案例 X_i 所含的非"0"属性个数 n_i,进而再按下式计算 X_i 与 Y 的相似度 $\mathrm{sim}(X_i,Y)$。$\mathrm{sim}(X_i,Y)$值越大,二者越相似:

$$\mathrm{sim}(X_i,Y)=\frac{n_i}{\sqrt{nm_i}}\propto\frac{n_i^2}{nm_i} \tag{6}$$

Step 5 借鉴 k-NN 算法的思想,选取 k 个与查询案例 Y 最为相似的训练案例,并在这 k 个训练案例中,根据所含案例数最多的类别来决定 Y 的类别。

Step 6 参照有关训练案例集的更新与维护算法(例如文献[1]中的 CBE 算法或文献[8]中的 BBNR+CRR 算法等),选择适宜的新案例,更新训练案例集,并去除冗余。

需要说明,在上述的新算法中,Step 1—Step 5 主要用于动态文本的分类步骤,而要使概念漂移问题得到圆满解决,还必须充分利用 Step 6 中的有关训练案例集更新与维护的算法,将携带有概念漂移重要信息的新案例源源不断地加入到训练案例集中才行。

由于很多文献已针对训练案例集的更新与维护问题做了大量工作,提出了很好的算法,因此我们在新算法的 Step 6 中直接使用这些成果,目前没有再多做有关的工作。

总之,由于基于 LFS 模式,新算法在对具体的查询案例分类之前不进行任何特征选择处理,因此,通过训练案例集的更新,新增的训练案例中所包含的新特征词将被全部保留,自然,这些新特征词所携带的有关概念漂移的重要信息也会得以完整保留,不必担心因特征选择而被淘汰。

一旦这些新词在新的查询案例中也出现时,它们就可以成功匹配,自然而然地发挥作用。所以,新算法不仅使得特征空间的更新问题迎刃而解,同时还很巧妙地解决了概念漂移问题。

4 算法验证

4.1 实验说明

关于基于 LFS 模式的分类算法(以下简称 LFS 法,并主要指第 3 节中的 Step 1—Step 5 步骤)的性能和效率测试,因条件所限,我们设计包括以下两个步骤:

第一,首先验证新算法对于静态文本分类问题的效率和性能,这是它能处理好概念漂移问题的前提和基础;第二,适当更新训练案例集,测试算法此时的性能指标有何变化,以此表明新算法对更新训练案例集的敏感性。敏感性越高,处理概念漂移问题时的实际效果越好。

因为垃圾邮件判别是一个非常典型的含有概念漂移现象的动态文本分类问题,所以本文以垃圾邮件判别问题为实验对象,进行算法验证。为了便于说明问题,我们选取当前垃圾邮件判别中常见的"根据互信息量(MI)进行特征选择 + 以 k-NN 算法进行分类"的组合模式(以下用"MI/k-NN"表示)作为参照对象。

根据习惯,在本节中,改称查询案例为测试案例,查询案例集,亦然。

4.2 测试环境

硬件配置:dell PowerEdge 2600 PC 服务器(标准配置)

操作系统及应用软件:RHAS4,Oracle9i,perl 5.8,DBD-Oracle-1.18

4.3 测试数据(语料集)

下载 lingspam 语料库(http://www.iit.demokritos.gr/skel/i-config/downloads/lingspam_public.tar.gz),然后对压缩文件解包,选取 lemm_stop 目录下若干文件夹中的邮件样本进行实验。

4.4 测试方法及结果

4.4.1 测试 LFS 法对于垃圾邮件判别问题的效率和性能

方法如下:

Step 1　用 perl 程序对 part1，part2 两个文件夹下的全部邮件样本进行抽词处理，然后输入到 oracle 数据库中，供测试使用。

Step 2　分别抽取这两个文件夹下各占 1/2 的正常邮件和垃圾邮件，共计 290 封，组成训练案例集；再将其余邮件，共计 288 封，组成测试案例集。

Step 3　以此训练集和测试集，分别测试 LFS 法[1)] 和 MI/k-NN 法[2)] 的效率（以“用时”表示）和性能（以“正确率”和“错误率”表示）。

需要说明，两种分类方法都与参数“k”有关，此外，MI/k-NN 法还与特征选择的维度“n”有关，所以，在实验中，我们参照文献[1,8-10]，选取较常用的参数值，分别进行了测试。

具体测试结果见表 1。

表 1　LFS 法与 MI/k-NN 法的分类效果比较

LFS				MI/k-NN								
				维度 n=1000				k=3				
k	用时[1]（分）	正确率（%）	错误率（%）	特征选择用时（分）[2]	k	分类用时（分）	正确率（%）	错误率（%）	n	分类用时（分）	正确率（%）	错误率（%）
3	4.96	95.83	4.17	4.91	3	21.86	84.03	15.97	500	15.07	83.33	16.67
5	5.04	96.88	3.13	4.91	5	22.32	83.33	16.67	700	17.88	83.33	16.67
7	5.11	96.18	3.82	4.91	7	22.27	83.33	16.67	1000	21.86	84.03	15.97
9	5.20	95.14	4.86	4.91	9	22.91	83.33	16.67	1200	25.21	84.03	15.97

表 1 显示，$k=5$ 时，LFS 法的相对效果最佳。而对于 MI/k-NN 法，$k=3$，$n=1000$ 时的相对效果最好。为使对比的效果更加明显，根据表 1，我们整理出表 2 和表 3，如下：

表 2　LFS 法与 MI/k-NN 法在效率和性能上的对比（k 取不同数值）

k	特征选择与分类的总用时（分）			错误率（%）		
	LFS	MI/k-NN（n=1000）	LFS 占 MI/k-NN 的比例（%）	LFS	MI/k-NN（n=1000）	LFS 占 MI/k-NN 的比例
3	4.96	26.78	18.52	4.17	15.97	26.09
5	5.04	27.23	18.49	3.13	16.67	18.75
7	5.11	27.18	18.81	3.82	16.67	22.92
9	5.20	27.82	18.69	4.86	16.67	29.17

表 3　LFS 法与 MI/k-NN 法在效率和性能上的对比（n 取不同数值）

k=3	LFS	MI/k-NN			
		n=500	n=700	n=1000	n=1200
特征选择与分类的总用时（分）	4.96	19.98	22.79	26.77	30.12
错误率（%）	4.17	16.67	16.67	15.97	15.97

表 2、表 3 中的数据表明，在参数 k，n 多次变化的情况下，LFS 法的用时始终保持在 MI/k-NN 法的 1/5 左右，错误率保持在后者的 1/4 左右，可见，LFS 法在性能和效率两个方面，都明显占优。

4.4.2　测试 LFS 法对训练案例集更新的敏感性

本次试验的目的是比较两种方法对训练案例集更新的敏感性。敏感性越高，越有利于处理好概念漂移问题。由于 $k=3$、$n=1000$ 时，对于 MI/k-NN 法的效果最好，为了突出对比效果，我们让 LFS 迁就一下 MI/k-NN，令它的 k 值也取为 3。具体步骤如下：

Step 1　抽取 part3 文件夹下的正常邮件、垃圾邮件各 1/2，组成测试案例集 A，将其余的邮件组成测试案例集 B。用 perl 程序对案例集 A，B 中的邮件样本进行抽词处理后输入到 oracle 数据库中。

Step 2　应用 4.4.1 节的训练案例集，分别用 LFS 法和 MI/k-NN 法对测试案例集 A 进行垃圾邮件判别，记下结果。

Step 3　再次应用 4.4.1 节的训练案例集，分别用 LFS 法和 MI/k-NN 法对测试案例集 B 进行判别；然后用这两种方法都错判的测试邮件，更新训练案例集。最后，应用新的训练案例集，用这两种方法再次对测试案例集 A 进行判别，记下结果。

Step 4　根据前两步的结果，对比分析 LFS 法和 MI/k-NN 法对训练案例集更新的敏感性，见表 4。

表 4　LFS 法与 MI/k-NN 法对训练案例集更新的敏感性比较

Step	LFS(k=3，单位：%)			MI/k-NN（k=3，n=1000，单位：%）		
	错误率	错误率变化值[1]	错误率变化率[2]	错误率	错误率变化值	错误率变化率
2	4.17			16.67		
3	1.38	−2.79	−66.90	15.17	−1.50	−8.97

1）错误率化值＞step3 的错误率－step2 的错误率

2）错误率变化率＝step3 的错误率/step2 的错误率＊100%

表 4 中的数据显示，在训练集更新后，两种方法的错误率都有所好转，MI/k-NN 法的错误率下降了 1.5 个百分点，占原错误率的 8.97%，而 LFS 法的错误率下降了 2.79 个百分点，占原错误率的 66.90%，可见 LFS 法对于训练集的变化更加敏感。

4.5　结果分析

测试表明，LFS 法在性能和效率两个方面都绝对占优，且表现出了很强的稳定性和健壮性。其主要原因在于：

1）普通的特征选择模式中，无论采用何种具体方法，都会有不同程度的信息丢失。而在 LFS 模式中，多余属性是被自然约减掉的，因而极大程度地减少了因特征选择而带来的信息丢失，使得分类算法的性能指标明显提高。同时也使得新算法对训练案例集的更新更加敏感，这对处理好概念漂移问题非常有利。

2）在基于 LFS 的分类算法中，首先去除了普通的“特征选择”环节，使分类系统省掉了计算并排序成千上万个特征词的 IG 或 MI 值的开销。其次，普通模式中，经过特征选择之后，特征空间的维度仍然在上千维左右，而在 LFS 方法中，分类函数只在查询案例所含非“0”属性的维度范围内进行计算，计算量大大降低。另外，和欧氏距离分类函数（公式）相比，新分类函数（公式或）非常简单。这三点原因，使得新算法的效率明显提高。

结束语　本文提出的“消极特征选择模式”和基于此模式的分类算法，很巧妙地解决动态文本分类中的概念漂移问题，高效可靠、简单易行，具有很强的实际意义和推广价值。下一步的工作，我们准备在训练案例集的更新与降低冗余度等方面，再进行一些优化研究，争取使得基于 LFS 模式的分类算法更加完整与完善。

1）LFS 法的“用时”，在意义上与“MI/k-NN”法的特征选择与分类两个步骤的总用时相当。

2）“MI/k-NN”法的“特征选择用时”指系统计算各特征词的 MI 值并排序的用时。

参 考 文 献

[1] Delanya S J, Cunninghamb P, Tsymbalb A, et al. A case-based technique for tracking concept drift in spam filtering (J), Knowledge-Based Systems, 2005, 18(4/5): 187-195

[2] Mitchell T M. 机器学习(M). 曾华军, 张银奎, 等译. 北京: 机械工业出版社, 2003, 1: 165-178

[3] Cunningham P, Nowlan N, Delany S J. et al. A Case-Based Approach to Spam Filtering that Can Track Concept Drift (C)// Proceedings of the ICCBR'03 workshop on long-lived CBR systems. Trondheim, Norway, 2003: 115-123

[4] Fdez-Riverola F, Iglesias E L, Díaz F, et al. Applying lazy learning algorithms to tackle concept drift in spam filtering(J). Expert Systems with Applications, 2007, 33: 36-48

[5] Zorkadis V, Karras D A, Panayotou M. Efficient information theoretic strategies for classifier combination, feature extraction and performance evaluation in improving false positives and false negatives for spam e-mail filtering (J). Neural Networks, 2005, 18: 799-807

[6] Delany S J, Cunningham P, Coyle L. An Assessment of Case-Based Reasoning for Spam Filtering (J). Artificial Intelligence Review, 2005, 24(3/4): 359-378

[7] Tan P N, Stenbach M, Kumar V. 数据挖掘导论(M). 范明, 范宏建, 等译. 北京: 人民邮电出版社, 2006(5): 13, 50, 137-138

[8] Delany S J, Cunningham P. An Analysis of Case-Base Editing in a Spam Filtering System [J]. Computer Science, Springer Berlin / Heidelberg, 2004, 3155: 128-141

[9] Androutsopoulos, Koutsias J, Chandrinos K V, et al. An Evaluation of Naive Bayesian Anti-Spam Filtering// Proceedings of the workshop on Machine Learning in the New Information Age (C). 11th European Conference on Machine Learning. Barcelona, Spain: 9-17

[10] Stone T. Parameterization of Naive Bayes for Spam Filtering. Masters comprehensive exam, University of Colorado at Boulder, 2003. http://trevorstone.org/school/spamfiltering.pdf

(上接第 103 页)

这些变迁对之间同步距离的求解。

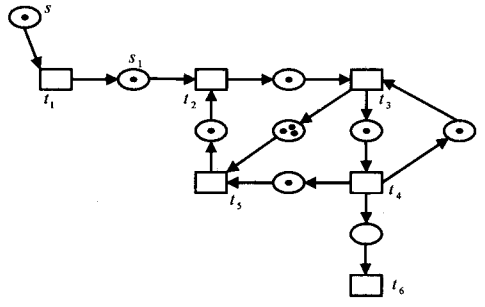

图 3 一个标识 T-网 Σ_2

易知, $T(s^*)=\{t_1,t_2,t_3,t_4,t_5,t_6\}$, 应用定理 3 中 3)我们可以分为以下二种情况来求解这些变迁之间的同步距离:

1)其中 t_2,t_3,t_4,t_5 它们两两之间均存在有向路, 它们所在有向回路的控制库所是 s_1, 所以我们采用 3)中 a)方法来求解它们之间的同步距离。

$$M_{\max}(s_1)=M_0(s)+\delta(t_1,t_2)=2$$

$$M_\delta(\Sigma_2)=\begin{array}{c} \quad t_2\ \ t_3\ \ t_4\ \ t_5 \\ \begin{bmatrix}0&0&1&1\\2&0&1&1\\2&1&0&0\\2&2&3&0\end{bmatrix}\begin{array}{l}t_2\\t_3\\t_4\\t_5\end{array}\end{array}$$

$$sd(t_2,t_3)=\max\{\delta(t_2,t_3),\min\{M_{\max}(s_1),\delta(t_2,t_3)+\delta(t_3,t_2)\}\}=2$$

$$sd(t_2,t_4)=\max\{\delta(t_2,t_4),\min\{M_{\max}(s_1),\delta(t_2,t_4)+\delta(t_4,t_2)\}\}=2$$

$$sd(t_2,t_5)=\max\{\delta(t_2,t_5),\min\{M_{\max}(s_1),\delta(t_2,t_5)+\delta(t_5,t_2)\}\}=2$$

$$sd(t_3,t_4)=\max\{\delta(t_2,t_3),\delta(t_2,t_4),\min\{M_{\max}(s_1),\delta(t_3,t_4)+\delta(t_4,t_3)\}\}=2$$

$$sd(t_3,t_5)=\max\{\delta(t_2,t_3),\delta(t_2,t_5),\min\{M_{\max}(s_1),\delta(t_3,t_5)+\delta(t_5,t_3)\}\}=2$$

$$sd(t_4,t_5)=\max\{\delta(t_2,t_4),\delta(t_2,t_5),\min\{M_{\max}(s_1),\delta(t_4,t_5)+\delta(t_5,t_4)\}\}=2$$

2) 由于变迁 t_1,t_6 与 t_2,t_3,t_4,t_5 之间只存在单向有向路, 而且 t_1 与 t_6 之间也只存在单向有向路, 所以它们之间的同步距离为:

$$t_1: sd(t_1,t_2)=M_0(s)+\delta(t_1,t_2)=2$$
$$sd(t_1,t_3)=M_0(s)+\delta(t_1,t_3)=2$$
$$sd(t_1,t_4)=M_0(s)+\delta(t_1,t_4)=3$$
$$sd(t_1,t_5)=M_0(s)+\delta(t_1,t_5)=3$$
$$sd(t_1,t_6)=M_0(s)+\delta(t_1,t_6)=3$$

$$t_6: sd(t_2,t_6)=\max\{M_0(s)+\delta(t_1,t_2),M_0(s)+\delta(t_1,t_6)\}=3$$
$$sd(t_3,t_6)=\max\{M_0(s)+\delta(t_1,t_3),M_0(s)+\delta(t_1,t_6)\}=3$$
$$sd(t_4,t_6)=\max\{M_0(s)+\delta(t_1,t_4),M_0(s)+\delta(t_1,t_6)\}=3$$
$$sd(t_5,t_6)=\max\{M_0(s)+\delta(t_1,t_5),M_0(s)+\delta(t_1,t_6)\}=3$$

结束语 本文找出了另一个 Petri 网子类——T-网同步距离的求解也是简单易行的, 指出了标识 T-网也同标识 S-图和标识 T-图一样可以直接通过网的结构和初始标识分布情况来得到变迁之间的同步距离, 并给出了相应的求解定理。由于在一个含有源库所的标识 T-网中源库所、相交回路集的控制库所以及控制库所接入变迁之间的关系非常的复杂, 而且每种情况下的同步距离的求解方法都稍有不同, 这就无法以一种统一的公式给出求解方法, 所以这里只给出了最基本的一种情况下的变迁之间的同步距离的求解, 至于其它的情况均可以在此基础上加以少许修改就可得到 (如文中就列出了其中另外两种情况下同步距离的求解方法)。

参 考 文 献

[1] Ezpeleta J, Colorn J M, Martinez J. A Petri net based deadlock prevention policy for Flexible Manufacturing Systems[J]. IEEE Transactions on Robotics and Automation, 1995, 11(2)

[2] 杜玉越, 蒋昌俊. 基于工作流网的实时协同系统模拟技术[J]. 计算机学报, 2004, 27(4)

[3] Shan Z G, In C, En F Y, et al. Modeling and Performance Analysis of a Multiserver Multiqueue System on the Grid[J]// Proc. of the The Ninth IEEE Workshop on Future Trends of Distributed Computing Systems(FTDCS'03). 2003

[4] 顾冠群, 姜爱泉, 罗军舟. 基于 Petri 网的协议并行化处理模型的描述和验证[J]. 计算机学报, 1996, 19, 11

[5] Petri C A. Interpretations of Net Theory[J]. ISF-Report75-07. GMD, St. Augustin, F. R. G., 1975

[6] 袁崇义. Petri 网原理[M]. 北京: 电子工业出版社, 1998

[7] 吴哲辉. Petri 网导论[M]. 北京: 机械工业出版社, 2006

[8] Murata T. Petri nets: Properities, Analysis and Applications[J]// Proc. of the IEEE. 1989, 77(4)

[9] 张军明, 吴哲辉. 标识 S-图中同步距离的计算[J]. 东南大学学报, 1995(5)

[10] 袁崇义. 出现网的同步距离[J]. 应用数学学报, 1984(10)