一种优化的 k-NN 文本分类算法

闫 鹏[1,2] 郑雪峰[1] 朱建勇[2] 肖赟泓[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 k-NN是经典的文本分类算法之一,在解决概念漂移问题上尤其具有优势,但其运行速度低下的缺点也非常严重,为此它通常借助特征选择降维方法来避免维度灾难、提高运行效率。但特征选择又会引起信息丢失等问题,不利于分类系统整体性能的提高。从文本向量的稀疏性特点出发,对传统的k-NN算法进行了诸多优化。优化算法简化了欧氏距离分类模型,大大降低了系统的运算开销,使运行效率有了质的提高。此外,优化算法还舍弃了特征选择预处理过程,从而可以完全避免因特征选择而引起的诸多不利问题,其分类性能也远远超出了普通k-NN。实验显示,优化算法在性能与效率双方面都有非常优秀的表现,它为传统的k-NN算法注入了新的活力,并可以在解决概念漂移等问题上发挥更大的作用。

关键词 文本分类,特征选择,k-NN分类法,概念漂移
中图法分类号 TP181 **文献标识码** A

Optimized k-NN Text Categorization Approach

YAN Peng[1,2] ZHENG Xue-feng[1] ZHU Jian-yong[2] XIAO Yun-hong[1]
(School of Information Engineering, University of Science and Technology Beijing, Beijing 100083, China)[1]
(The State Information Center, Beijing 100045, China)[2]

Abstract As one of the most classical TC approaches, k-NN is advantaged in tackling concept drift. However, to avoid curse of dimensionality, it has to employ FS(feature selection) method to reduce dimensionality of feature space and improve operation efficiency. But FS process will generally cause information losing and thus has some side-effects on the whole performance of approach. According to sparsity of text vectors, an optimized k-NN approach was presented in paper. This optimized approach greatly simplified euclidean distance model and reduced the operation cost without any information losing. So it can simultaneously achieve much higher both performance and efficiency than general k-NN approach. It then enhanced the advantage of k-NN in managing concept drift.

Keywords Text categorization, Feature selection, k-NN, Concept drift

1 引言

1.1 基于 k-NN 的文本分类系统的一般应用模式

Widmer 等人于 1996 年提出的"概念漂移(Concept Drift)"问题[1],现在成为数据流挖掘领域研究的热点问题之一。其实,这一问题在文本分类研究方面也非常重要,它使得文本分类器或分类系统必须具备良好的持续学习能力,否则其性能就会不断下降以至不可使用。

k-最近邻(k-NN)算法作为消极学习型(lazy Learner)分类法的典型代表,在应对概念漂移问题上,可谓得天独厚[2],因为它可以利用增量式更新的方式,非常方便地持续更新训练集,以不断吸收包含有概念漂移信息的新训练文本,从而使分类系统能够跟得上概念漂移变化的步伐。

k-NN算法在20世纪60年代就已经成为了非常重要的分类方法[3]。它不仅在应对概念漂移问题上优势明显,而且还具有简单直观、容易实现,准确率高等优点。此外,如果训练集足够大,它还能对噪音数据表现出非常好的鲁棒性[4]。

然而,k-NN算法却有一个非常致命的缺点——速度低下,因为它在对每一个查询实例(Query Instance)进行分类时,都需要搜索整个训练集来寻找最近邻,所以它的运算开销巨大,时间代价高昂,这导致了它的运行速度非常低下。

因此,在基于k-NN算法的文本分类系统中,为了提高运行速度,同时也为了避免出现维度灾难(Curse of Dimensionality),在预处理阶段,分类系统通常需要借助于特征选择等方法来进行向量空间模型(VSM, Vector Space Model)的降维处理,以便将特征空间(Feature Space)的维度控制在分类器能够承受的范围之内。

目前,特征选择主要采用评估函数法,常用的评估函数有信息增益(IG, Information Gain)、互信息量(MI, Mutual Information)、卡方统计(CHI),等等,其中以IG和CHI的效果

到稿日期:2008-11-06 返修日期:2009-07-24

闫 鹏(1970—),男,博士研究生,高级工程师,CCF学生会员,主要研究领域为计算机应用、网络安全,E-mail:yanpeng@mx.cei.gov.cn;郑雪峰(1951—),教授,博士生导师,主要研究领域为网络与信息安全;朱建勇(1972—),男,博士,主要研究领域为自动问答技术、电子政务、电子商务、信息安全等;肖赟泓(1977—),男,博士研究生,主要研究领域为计算机安全与网络安全。

为最好[5]。

归纳起来，当前基于 k-NN 算法的文本分类系统，一般都遵循如图 1 所示的应用模式。

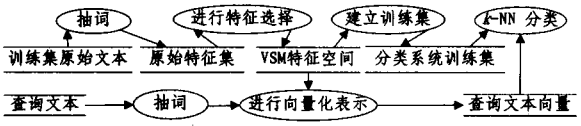

图 1　基于 k-NN 算法的文本分类系统的一般应用模式

1.2　特征选择存在的问题分析

特征选择降维方法是一把双刃剑，它可以降低 k-NN 分类器的运算开销、提高其运行效率，但同时它也会引起很多负面问题：

1)不利于及时识别和处理概念漂移问题。

因为现有的特征选择基本上采用评估函数法，即所有的特征词都要经过评估函数的统一筛选，分值高者才有可能被选入特征空间。但是，在开始时，含有概念漂移重要信息的新特征词，只可能存在于少量的文本之中，相应的，它们的评估函数分值必然很低，所以这些新特征词很难通得过评估函数的筛选而被及时选中。事实上，只有当它们的评估函数值增长到一定程度时，才有可能进入到特征空间中。而此时分类器对概念漂移问题的识别和判断早已滞后了。

2)信息丢失问题

特征选择是在原始特征全集中，通过一定的方法挑选出一个特征子集(或属性子集)，然后以子集来代全集用于文本分类，所以它不可避免地存在着信息丢失的问题。信息丢失显然会降低分类器的分类性能。

3)特征空间的维度确定问题

特征空间的维度确定问题非常重要，它不仅直接关系到算法的效率，而且还与信息丢失的程度相关联，进而会影响分类器的分类性能。但是，目前对此问题尚无理想的方法，基本上是根据实验效果而定[6,7]。由于数据以及具体分类对象的差异性，这种通过实验而定的做法，在实际应用时还显得比较随意，难以掌握。

4)k-NN 算法的运行效率问题

即使经过特征选择降维处理，特征空间的维度通常仍在上千维左右，例如在著名的文献[5]中，它的维度即在 2000～16000 之间，在这样高的维度下运行，k-NN 分类器仍旧存在着速度慢、效率低的弱点。

基于上述问题，本文提出了一种既基于 k-NN 算法，但又不必进行特征选择处理的优化算法，该算法不仅大大降低了分类器的运算开销，同时还避免了因特征选择而引起的信息丢失等诸多问题，因而较图 1 所示的普通 k-NN 分类系统，该算法在性能与效率两个方面，都有了非常显著的提高。

在下文中，首先提出了文本向量的压缩表示模型，然后以此为基础，再结合欧氏距离公式，提出了优化算法的分类模型表达式及算法的详细描述，最后，通过对比实验，验证了优化算法的实际效果。

2　基于 k-NN 的优化算法描述

2.1　普通的 k-NN 算法简介

k-NN 算法通常以“欧氏距离(Euclidean Distance)”为其分类模型，欧氏距离公式的定义如下[8]：

定义 1　设在 n 维空间中有两个点 $X=(x_1,x_2,\cdots,x_n)$ 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 个属性值。

k-NN 算法的基本思想是：当有待分类的查询文本到来时，根据欧氏距离公式在训练集中找到与之最为接近的 k 个训练文本，然后再通过“多数表决”的方法，以这 k 个训练文本中，占多数的文本类别来决定查询文本的类别。

2.2　文本向量的稀疏性特点及其压缩表示模型

目前主要采用向量空间模型(VSM)来表示文本向量，在通常的 VSM 表示方法中，需要首先确定特征空间的维度。这个维度值，一般在进行特征选择时确定。

前文谈到，本文的优化算法无需进行特征选择处理，那么，自然是以原始特征全集所含属性的个数来作为 VSM 特征空间的维度值，即以原始特征全集构成 VSM 的特征空间。这在理论上说是非常简单的，但在具体应用时却会遇到一些难以解决的实际问题：

其一，所谓原始特征全集，是对分类系统的训练集而言的，但此刻查询文本如何尚属未知，所以这一“全集”并不能保证将会包含未知的查询文本的全部特征词。或者说，它很可能会漏掉在未知的查询文本中出现的某些新特征词，所以全集并不能保证真正完全。

其二，为了使分类器能够识别和判断概念漂移现象，需要将新的属性或特征词源源不断地追加到特征空间中，这就要求分类系统必须不断地更新其特征空间。然而，在通常的 VSM 表示方法中，特征空间的更新，要求分类系统必须重建原始特征全集，并更新全部训练文本的向量表示形式，这是一项非常耗时的工作。如果频繁地更新特征空间，将会给系统增加巨大的负担，使其难以承受。

因此，特征空间一旦建立，必须保持一段时间的相对稳定，其间，即使查询文本中出现了新的特征词，它也只能被忽略掉。可见，通常的 VSM 表示方法不便于特征空间的动态更新，不利于分类系统及时识别和应对概念漂移问题。

其三，由于文本向量的稀疏性特点，通常的 VSM 表示形式将会浪费大量的存储资源，也不利于提高分类器的检索速度和运行效率。

所谓文本向量的稀疏性，是指尽管特征空间包含有成千上万的属性(即特征词)，但一般情况下，对于每一个具体的文本向量而言，绝大多数只可能包含其中的一小部分属性，即它只有一小部分属性的值为非“0”，其余绝大多数属性的值都为“0”，因此具体的文本向量通常是非常稀疏的[8]。

正是基于文本向量的稀疏性特点，本文提出了采用“压缩模型”的形式表示文本向量的思想，即“只列出文本向量中实际包含的特征词属性及其属性值”。

具体而言，设 X 表示某文本向量，它对应的原始文本中实际包含有 $x_1,x_2,\cdots,x_n$ 等 n 个特征词，则 X 可以用压缩模型表示为：

$$X=(\langle x_1,w_1\rangle,\langle x_2,w_2\rangle\cdots\langle x_n,w_n\rangle) \tag{2}$$

其中，x_i 表示 X 的第 i 个属性，$i\in\{1,\cdots,n\}$；w_i 为 x_i 的属性值。

不难看出，采用这种“压缩模型”的表示方法，上述的 3 个

难点问题将会迎刃而解。

此外，关于如何表示文本向量的属性值问题，目前有两种主要方法[9,10]：一是布尔(Boolean)表示法，即“1”表示某属性(特征词)在此文本中曾出现，“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(TF，Term Frequency)来表示。

布尔法简单实用，在文本分类方面已得到了广泛应用。相比之下，数值法虽然增加了权重等信息，但实际效果却并不十分明显，甚至有时还不如布尔法[9,10]。因此本文采用布尔法(0-1)表示属性值，并在此基础上推导 k-NN 优化算法的分类模型。

2.3 k-NN 优化算法的分类模型

设 X,Y 分别表示两个文本向量，不妨设它们分别包含 m 和 n 个非“0”属性，由于本文采用布尔法表示向量的属性值，故其压缩模型可以分别表示为：

$$X=(\langle x_1,1\rangle,\langle x_2,1\rangle\cdots\langle x_m,1\rangle) \tag{3}$$

$$Y=(\langle y_1,1\rangle,\langle y_2,1\rangle\cdots\langle y_n,1\rangle) \tag{4}$$

再设：

f_{11} 为 X,Y 中都包含的属性个数，相应的，X,Y 中对应属性的值都为“1”；

f_{00} 为 X,Y 中都不包含的属性个数，相应的，X,Y 中对应属性的值都为“0”；

f_{10} 为 X 中包含但 Y 中不包含的属性个数，相应的，X 中对应属性的值都为“1”，Y 中对应属性的值都为“0”；

f_{01} 为 X 中不包含但 Y 中包含的属性个数，相应的，X 中对应属性的值都为“0”，Y 中对应属性的值都为“1”；

根据式(3)、式(4)，有：

$$f_{10}=m-f_{11} \tag{5}$$

$$f_{01}=n-f_{11} \tag{6}$$

根据式(1)，有：

$$d(X,Y)=\sqrt{f_{00}\cdot(0-0)^2+f_{10}\cdot(1-0)^2+f_{01}\cdot(0-1)^2+f_{11}\cdot(1-1)^2}=\sqrt{f_{10}+f_{01}} \tag{7}$$

将式(5)、式(6)代入式(7)，则有：

$$d(X,Y)=\sqrt{m+n-2f_{11}}\propto m+n-2f_{11} \tag{8}$$

根据式(8)可知，在布尔表示法下，两个文本向量的欧氏距离，只与它们各自所含的非“0”属性的个数(即 m,n)以及二者共同包含的非“0”属性的个数(即 f_{11})有关。事实上，在进行文本分类时，得到这 3 个数值是非常容易的。所以，相比式(1)，式(8)变得非常简单、易行，分类器的运算开销也将会随之大大降低。

不仅如此，k-NN 分类器是将同一个查询文本“Y”与训练集中不同的训练文本“X”分别计算欧氏距离并排序，那么，式(8)中表示 Y 所含非“0”属性个数的参数“n”，实际上可以被看成是一个常量，它的大小将不会影响最终的排序结果，所以，式(8)还可以再进一步简化为：

$$d(X,Y)\propto m-2f_{11} \tag{9}$$

3 k-NN 优化算法的详细描述

下面将基于 k-NN 的文本分类优化算法详细描述如下：

Step 1　对各训练文本抽取特征词后，全部输入训练集。设训练集中第 i 个训练文本 X_i 包含 m_i 个特征词，即非“0”属性。

Step 2　当有查询文本 Y 到来时，对其进行抽词后，得到它所包含的全部非“0”属性名单。

Step 3　在 Y 所包含的全部非“0”属性的名单之中进一步筛选，容易得到 X_i 与 Y 共同包含的非“0”属性的个数“n_i”。根据式(9)，X_i 与 Y 的欧氏距离 $d(X_i,Y)$ 表示为：

$$d(X_i,Y)\propto m_i-2n_i \tag{10}$$

Step 4　以式(10)为分类模型，逐一计算各 $d(X_i,Y)$ 值。

Step 5　利用“多数表决”方法，选取 k 个与 Y 最为接近的训练文本向量，并在这 k 个训练文本向量中，以所含向量数最多的类别来决定 Y 的类别。

4 算法测试

4.1 实验说明

垃圾邮件判别是一个典型的文本分类问题，所以，本文以垃圾邮件判别问题为实验对象，验证 k-NN 优化算法的实际效果。

实验中选取如图 1 所示的普通 k-NN 分类系统为参照。因为“信息增益(IG①)”在多种评估函数中效果突出[5]、使用广泛，所以本实验以 IG 为 k-NN 普通分类系统的特征选择评估函数。

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，在 lemm_stop 目录下的“part1－part5”文件夹中，各分别随机选取 20 封正常邮件(ham)和 20 封垃圾邮件(spam)，共计 200 封，组成训练集；再各分别随机选取 20 封正常邮件和 20 封垃圾邮件，也共计 200 封，组成测试集。用 perl 程序对这些邮件样本进行抽词，然后输入到 oracle 数据库中，供实验使用。

以此训练集和测试集，分别测试两种分类器的运行效率(以“用时”表示)和性能(以“正确率或错误率”表示)。

4.4 测试结果

为了能充分体现两种算法的实际效果，在实验中将有关参数的取值范围都做了适当扩大，例如，它们都与参数“k”有关，文献[11]建议将 k 值取为 3，现则将 k 的取值范围适当扩大为 3,5,7,9。

① IG 又称为“平均互信息量”，其计算公式如下：假设有特征词 w 和类 c，那么，w 和 c 的信息增益(或平均互信息量)为：$IG(w;c)=\sum_{i=1}^{n}p(w,c_i)\log_2\frac{p(w,c_i)}{p(w)p(c_i)}+\sum_{i=1}^{n}p(\overline{w},c_i)\log_2\frac{p(\overline{w},c_i)}{p(\overline{w})p(c_i)}$，详见“孙丽华，谢仲华，陈荣伶. 信息论与纠错编码[M]. 北京：电子工业出版社，2005：14-36”

此外，k-NN 普通算法还与特征空间的维度值“n”密切相关，文献[6]建议“n”在 600～1000 左右比较合适，文献[7]也认为“n”在 1000 维左右效果较好，为增强代表性，将这个范围也稍作扩大，分别定为 500，700，1000，1200 四个维度。

测试结果如表 1，表 2 所列。

表 1　k-NN 文本分类优化算法的测试结果

k	3	5	7	9
用时(秒)	134.1	137.5	145.8	149.3
正确率(%)	77.00	77.50	78.00	85.00
错误率(%)	23.00	22.50	22.00	15.00

表 2　k-NN 文本分类普通算法的测试效果

维度 n	k=3			k=5			k=7			k=9		
	用时(秒)	正确率(%)	错误率(%)	用时(秒)	正确率(%)	错误率(%)	用时(秒)	正确率(%)	错误率(%)	用时(秒)	正确率(%)	错误率(%)
500	659.6	50.00	50.00	661.8	50.00	50.00	669.8	45.00	55.00	676.1	45.00	55.00
700	738.0	50.00	50.00	738.5	50.00	50.00	744.8	43.50	56.50	755.1	54.00	46.00
1000	855.0	50.50	49.50	859.0	53.50	46.50	866.6	55.50	44.50	870.2	50.50	49.50
1200	936.7	50.50	49.50	937.9	54.00	46.00	944.1	53.50	46.50	964.5	53.50	46.50

4.5　结果分析

1)两种算法的运行效率对比及分析：

道理上，k-NN 普通算法的运算开销主要取决于特征空间的维度“n”，相比之下，它受参数“k”的影响很轻微。表 2 中的“用时”随“n”、“k”相变化的情况也完全体现了这一点。因此，为简化对比分析，暂且忽略同一维度下不同“k”值对算法效率(即用时)的影响，计算出表 2 中 k-NN 普通算法在不同的维度值“n”下的平均用时情况，并以它们来代表其运行效率。

同理，也可以计算出表 1 中 k-NN 优化算法的平均用时，也以它来表示其运行效率。对比这两种算法的平均用时，可以反映出二者在效率上的差别情况，如表 3 所列和图 2 所示。

表 3　k-NN 优化算法与普通算法的运行效率比较(单位：秒)

优化算法	普通算法 (n=500)	普通算法 (n=700)	普通算法 (n=1000)	普通算法 (n=1200)
141.7	666.8	744.1	862.7	945.8

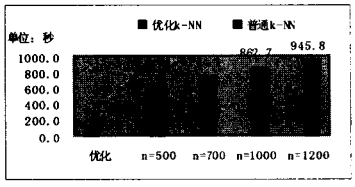

图 2　k-NN 优化算法与普通算法的运行效率比较

如表 3 和图 2 所示，k-NN 普通算法的用时约是优化算法用时的 5～7 倍，可见后者在效率上的优势是非常明显的，这主要是由于二者的运算开销相差悬殊所致。而二者的运算开销之所以相差悬殊，主要原因在于：

其一，优化算法以压缩形式表示文本向量，其运算主要局限在文本向量所含的非“0”属性的范围之内，由于文本向量的稀疏性，这使得它的运算开销较普通算法大为降低。

以本次实验为例，据统计，尽管邮件样本的原始特征全集共含有 14230 个特征词，但实验中的全部 400 个邮件向量，最短的只含有 5 个特征词，最长的也不过有 918 个特征词，平均每个邮件向量只含有 154.72 个特征词。也就是说，实验中，k-NN 优化算法是在 154.72 维的平均维度下运行，而普通算法则是在 500～1200 维的维度下运行，二者的运行维度相差巨大，相应的，它们的运算开销也必然相差悬殊。

其二，k-NN 优化算法无需特征选择处理，而普通算法还需要额外计算、排序这 14230 个特征词的 IG 值，这又是一项不菲的运算开销。

其三，优化算法的分类模型(式(10))远比普通算法的分类模型(式(11))简单易行。

2)两种算法的分类性能对比及分析：

表 1、表 2 中的数据显示，两种算法在实验中的性能指标都略有些偏低，这应该与训练集欠优化有关。不过，由于两种算法都使用了相同的实验数据，所以这并不会影响二者性能差异情况的真实反映。

为了简单、直观地反映两种算法在性能上的差异情况，以两表中的正确率指标为分析对象，并暂且忽略因参数的不同而引起的细微差异，然后将两表中的各正确率数值，按其所属的算法进行了分类、排序、去重后，得到表 4 与图 3。

表 4　k-NN 优化算法与普通算法的正确率比较(单位：%)

优化 k-NN	77.00	77.50	78.00	85.00			
普通 k-NN	43.50	45.00	50.00	50.50	53.50	54.00	55.50

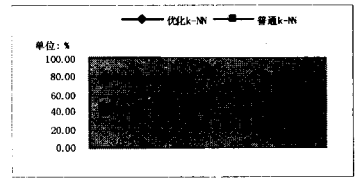

图 3　k-NN 优化算法与普通算法的正确率比较

表 4 和图 3 显示：k-NN 普通算法的正确率围绕在 50%上下波动，而优化算法的正确率却围绕在 80%上下波动，二者之间相差约 30 个百分点，可以说，优化算法将普通算法的正确率提高了约 60%，性能提高的幅度是非常之大的。

这两种算法所用的分类模型，形式虽然各异，但实质上却是等价的。因此，它们之所以在分类性能上出现了如此之大的差异性，关键原因在于普通算法采用了特征选择降维手段。

或者说，是特征选择所固有的信息丢失等诸多负面影响，损害了 k-NN 普通算法的部分性能。而优化算法则可以直接使用最完整且最精简的信息来进行分类，所以它不存在这个问题。因此可以说，是特征选择造成了两种算法性能上的巨大差异。

结束语　普通 k-NN 分类系统借助于特征选择降维手段来提高效率、避免维度灾难的做法，实质上是一种“以性能换效率”的无奈之举。

本文另辟蹊径，提出了 k-NN 文本分类的优化算法。优化算法充分利用文本向量的稀疏性特点，找到了一条既可以大大降低系统运算开销、提高运行效率，同时又可以不必进行特征选择的新道路。该算法巧妙地避免了因特征选择而引起

的诸多负面问题，从而可以将最完整且最精简的信息直接提供给分类模型，实现了性能与效率的双赢。

此外，优化算法采用压缩模型的形式表示文本向量，这也为它频繁、快捷地更新训练集创造了极为有利的条件，从而可以使它在解决概念漂移等问题上发挥更大的优势与作用。

参 考 文 献

[1] Widmer G K M. Learning in the presence of concept drift and hidden contexts[J]. Machine Learning, 1996, 23(1): 69-101

[2] Fdez-riverola F, Iglesias E L, Me′ndez F D R, et al. Applying lazy learning algorithms to tackle concept drift in spam filtering [J]. Expert Systems with Applications, 2007, 33: 36-48

[3] Witten I H, Frank E. Data Mining: Practical Machine Learning Tools and Techniques(2 ed)[M]. Beijing: China Machine Press, 2006

[4] Mitchell T M. Machine learning[M]. Beijing: China Machine Press, 2003: 165-178

[5] Yang Y, Pedersen J. A comparative study on feature selection in text categorization[M]. San Francisco: Morgan Kaufmann Publishers, 1997

[6] Delanya S J, Cunninghamb P. An analysis of case-base editing in a spam filtering system[J]. Computer Science, 2004, 3155: 128-141

[7] Stone T. Parameterization of naive bayes for spam filtering[R]. Masters comprehensive exam. University of Colorado at Boulder, 2003

[8] Tan P, Stenbach M, Kumar V. Introduction to data mining[M]. Beijing: People Posts & Telecom Press, 2006: 13-50

[9] 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

[10] 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

[11] 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

(上接第 212 页)

系，来探讨 BAM 的稳定性对神经网络的研究也将是一个有意义的课题。

参 考 文 献

[1] Wille R. Restructuring lattice theory: an approach based on hierarchies of concepts[M] // Rival I, ed. Ordered Sets. Dordrecht: Reidel, 1982: 445-470

[2] Ganter B, Wille R. Formal Concept Analysis : Mathematical Foundations[M]. Berlin: Springer-Verlag, 1999

[3] Yao Y Y. Concept lattices in rough set theory[C] // Proceedings of 23rd International Meeting of the North American Fuzzy Information Processing Society. 2004

[4] Qu K S, Zhai Y H. Generating complete set of implications for formal contexts, Knowl. BasedSyst. 2008, doi: 10. 1016/ j. knosys. 2008. 03. 001

[5] Qu K S, Liang J Y, Wang J H, et al. The algebraic properties of concept lattice[J]. Journal of Systems Science and Information, Research Information Ltd UK, 2004, 2(2): 271-277

[6] 曲开社，翟岩慧. 偏序集、包含度与形式概念分析[J]. 计算机学报，2006, 29(2)

[7] Belohlavek R. Fuzzy logical bidirectional associative memory [J]. Information Sciences, 2000, 128: 91-103

[8] 曲开社，翟岩慧，梁吉业，等. 形式概念分析对粗糙集理论的表示及扩展[J]. 软件学报，2007, 18(9): 2174-2182

[9] Zupa B, Bohance M. Learning by discovering concept hierarchies. Artificial Intelligence, 1999, 109: 211-242

[10] Tonella. Using a concept lattice of decomposition slices for program understanding and impact analysis[J]. IEEE Transactions on Software Engineering, 2003, 29(6): 495-509

[11] Dekel U. Revealing Java class structure with concept lattices [D]. Technion -Israel Institute of Technology, 2003

[12] Arevalo G, Mens T. Analyzing object - oriented application frameworks using concept analysis[J]. LNCS, 2002, 2426: 53-63

[13] Dekel U, Gil Y. Revealing Class Structure with Concept Lattices [C] // Proc. 10th Working Conference on Reverse Engineering. 2003

[14] Valtchev P, Missaoui R, Godin R, et al. Generating Frequent Itemsets Incrementally: Two Novel Approaches Based on Galois Lattice Theory[J]. J. Expt. Theor. Artif. Intell, 2002, 14: 115-142

[15] 谢志鹏，刘宗田. 概念格与关联规则发现[J]. 计算机研究与发展，2000, 37(12): 1415-1421

[16] 梁吉业，王俊红. 基于概念格的规则产生集挖掘算法[J]. 计算机研究与发展，2004, 41(8): 1339-1344

[17] Qu Kai-she, Zhai Yan-hui, Liang Ji-ye, et al. Study of decision implications based on formal concept analysis[J]. International Journal of General Systems, 2007, 36(2): 147-156

[18] Kosko B. Adaptive Bidirectional Associative Memories[J]. Applied Optics, 1987, 26(23): 4947-4860

[19] Kosko B. Constructing an associative memory[J]. Byte, 1987, 12 (10): 137-144

[20] Kosko B. Bidirectional Associative Memory[J]. IEEE Transactions on Systems, Man and Cybernetics, 1988, 18(1): 49-60

[21] 谢志鹏，刘宗田. 概念格的快速渐进式构造算法[J]. 计算机学报，2002, 25(5): 490-495

[22] 翟岩慧，曲开社，曹桃云. 基于矩阵秩的概念格生成算法[J]. 电脑开发与应用，2006, 19(5): 11-12

[23] Kuznetsov S, Obiedkov S. Comparing Performance of Algorithms for Generating Concept Lattices[J]. J. Experimental and Theoretical Artificial Intelligence, 2002, 14: 189-216

[24] Stumme G, Taouil R, Bastide Y, et al. Computing Iceberg Concept Lattices with TITANIC[J]. Journal on Knowledge and Data Engineering, 2002

[25] 于海斌，薛劲松，王浩波，等. 双向联想记忆神经网络的一种编码策略[J]. 电子学报，1997, 25(5): 6-10