DPFS:一种基于动态规划的文本特征选择算法

任永功 林 楠
(辽宁师范大学计算机与信息技术学院 大连 116029)

摘 要 在文本特征选择过程中,针对原始特征空间维数过高、计算量过大、并且存在较大不相关性和冗余性,提出了一种基于动态规划思想的文本特征选择算法(DPFS)。首先,结合动态规划思想,基于特征与类别的相关性分析,对原始特征集合进行特征筛选,保留与类别具有强相关性和弱相关性的特征;然后,再次结合动态规划思想,对特征子集做冗余性分析,滤除弱相关且冗余的特征;最后,得到一个近似最优特征子集。实验结果表明,此算法在对数据降维和在降维过程中减少计算量是有效的。

关键词 特征选择,相关性,冗余性,动态规划

DPFS: A Text Feature Selection Algorithm Based on Dynamic Programming

REN Yong-gong LIN Nan
(School of Computer and Information Technology, Liaoning Normal University, Dalian 116029, China)

Abstract In the process of the text feature selection, the space dimension of the original characteristics is too high, excessive calculation, and there is greater irrelevance and redundancy features, a feature selection algorithm based on dynamic programming (DPFS) was proposed. First, selecting the feature from the original feature set based on dynamic programming and the analysis of features' relevance and classes' relevance, reserve features that are the strong correlation and the weak correlation; and then, for the feature subset based on dynamic programming, delete features that are weak correlation and redundant; finally, gaining a similar best features subset. The result of experiment shows that, this method is effective for reduced data-dimension and reduces the amount of calculation in the course of reduce data-dimension.

Keywords Features selection, Correlation, Redundancy, Dynamic programming

1 引言

文本分类是文本挖掘(text mining)中的一项重要技术,其大致可分为3个步骤:文本的向量模型表示[1]、文本特征选择和分类器训练。其中,文本特征选择是文本分类的关键问题之一。特征选择[2]就是从一个原始的特征集合中选择一个最优特征子集的过程。我们认为在特征选择过程中,只有被选中的文本特征向量中包含足够的类别信息才有可能通过分类器实现正确分类,然而特征中是否已包含足够的类别信息是很难确定的,为了更有效地提高识别精度,在对特征进一步进行处理之前,必须去掉两类特征:一类是与分类目标不相关的特征;另一类是与其它特征有弱相关性的冗余特征。最后得到一个保留原有特征集合的全部或大部分分类信息的最优特征子集,使分类效果不变或降低很小。

一般的特征选择算法只注重单一特征的评价[3],只考虑到特征和类别之间的相关性,为了提高识别率和分类效果,总是最大限度地提取特征信息。因此,在进行文本特征选择的过程中,我们遇到的两个问题就是:

第一,特征维数过大,且特征存在较大的不相关性和冗余性的问题。

第二,在做冗余性分析时,用来评估的量度计算量过大的问题。

鉴于以上两个问题,在本文中我们结合动态规划的思想,从特征的不相关性和冗余性的全局考虑,采用对称的不确定量度(RMI)对特征不相关性和冗余性进行判别,提出了一种基于动态规划思想的特征选择算法(DPFS),本算法主要完成的工作如下:

第一,结合动态规划思想,基于特征与类别的相关性分析对原始特征集合进行特征筛选,保留与类别具有较强相关性的特征。

第二,针对经过筛选而精简的特征子集,结合动态规划的思想,对特征子集做冗余性分析,最终得到一个近似最优特征子集。

2 相关工作

特征提取算法一般是构造一个权重函数,对特征集中的每一个特征进行独立评估,这样每个特征都获得一个评估分,然后对所有的特征按照其评估分大小进行排序,选取预定数

到稿日期:2008-11-17 返修日期:2009-01-14 本文受国家自然科学基金项目(60603047),辽宁省科技计划项目(2008216014),辽宁省教育厅高等学校科研基金(2008341),大连市优秀青年科技人才基金(2008J23JH026)资助。

任永功(1972—),男,教授,博士,研究方向为数据挖掘技术等,E-mail: renyg@dl.cn;林 楠(1984—),女,硕士研究生,研究方向为文本挖掘。

目的特征作为结果的特征子集。主要的方法有：文本频数(Document Frequency，DF)、信息增益(Information Gain，IC)、互信息(Mutual Information，MI)、开方校验(Chi-Square)、期望交叉熵(Expected Cross Entropy)、优势率(Odds Ratio)、文本权证(The Weight of Evidence for Text)等。虽然采用了以上这些特征选择算法后可以大幅度提高分类器的性能，但是它们只考虑了特征与类别之间的相关性，而忽略了特征之间的不相关性，因此很容易出现以下问题：在有些情况下，某些特征之间的相关性很大，即它们之间很相似，但它们与类别的相关性也很大，于是这些相似的特征子集都被作为候选特征选入了最优特征子集，这就导致了特征子集中存在着大量的冗余特征[4-6]，从而影响了分类器的性能。而这种情况在某些类别的训练集数目不足够多的情况下将会更加糟糕，因为在稀疏类别中的特征比那些在主要类别中特征的评估值要低，传统的特征选择算法往往会倾向于那些主要类别中的特征，而且有些特征选择算法的计算代价是很大的。

3 基本定义

John，Kohavi 和 Pfleger 将特征分成 3 个互不相交的类别，即强相关、弱相关和不相关的特征。设 F 是所有特征的集合，F_i 是其中一个特征，$S_i=F-\{F_i\}$，C 是给定的类别，则相关性的 3 种类别在下面给出。

定义 1(强相关性) 如果特征 F_i 满足 $P(C|F_i,S_i)\neq P(C|S_i)$，则称特征 F_i 是与类别 C 强相关的。一个特征的强相关性表明该特征对一个最优的特征子集总是必需的，在不影响最初的类别分布的情况下该特征不能被去除。

定义 2(弱相关性) 如果特征 F_i 满足 $P(C|F_i,S_i)\neq P(C|S_i)$，且存在 $S_i'\subset S_i$，使得 $P(C|F_i,S_i')\neq P(C|S_i')$，则称特征 F_i 是与类别 C 弱相关的。一个特征的弱相关性表明该特征对一个最优的特征子集并不总是必需的，但是在某种条件下可能加入到一个最优的特征子集中去。

定义 3(不相关性) 如果特征 F_i 满足 $\forall S_i'\subseteq S_i$，$P(C|F_i,S_i')=P(C|S_i')$，则称特征 F_i 是与类别 C 不相关的。不相关性表明该特征在最优特征子集中总是不必要的。所以，一个最优特征子集应该是由强相关性特征和部分弱相关性特征组成的。

特征冗余一般是以特征关联来确定，普遍认为如果两个特征的数值完全相互关联，那么它们彼此是冗余的，实际上，当一个特征与一组特征部分地相互关联的时候，不可能直接决定该特征是冗余的。

定义 4(冗余特征) 设 G 是当前特征集，如果一个特征是弱相关的，并且存在 G 中的一个特征子集 M_i 形成关于该特征的 Markov Blanket，则该特征是冗余的。

定义 5(Markov Blanket) 给定一个特征 F_i，对于 $M_i\subset F(F\notin M_i)$，如果 M_i 满足公式 $P(F-M_i-F_i,C\mid F_i,M_i)=P(F-M_i-F_i,C\mid M_i)$，则称 M_i 是关于 F_i 的一条 Markov Blanket[7]。在 Markov Blanket 的定义中，其条件指出 M_i 不只包含特征 F_i 与类别 C 之间相关的信息，同时也需要包含特征 F_i 与所有其他特征之间的相关信息。强相关特征不存在 Markov Blanket。

图 1 表明一个整个的特征集可以完全地划分为 4 个基本部分：不相关的特征、冗余的特征(一部分弱相关特征)、弱相关但非冗余的特征和强相关的特征。一个最优的特征子集本质上包含所有在弱相关且非冗余的特征和强相关的特征。值得指出的是：尽管弱相关且冗余的特征和弱相关且非冗余的特征是不相交的，但对它们的不同的分割是由 Markov Blanket 过滤法产生的。

图 1 相关特征和冗余特征示意图

定义 6(对称的不确定量度 RMI)

$$RMI(X,Y)=2\frac{I(X;Y)}{H(X)+H(Y)}$$

其中，$I(X;Y)$是指特征 X 与特征 Y 之间的互信息，$H(X)$和 $H(Y)$分别是随机特征 X 和 Y 的信息熵。这里要给出关于互信息和信息熵的定义。

定义 7(互信息) 两个随机变量 X,Y，它们的密度分布函数分别为 $p(x)$，$p(y)$，联合概率分布为 $p(x,y)$，则 X,Y 的互信息为：

$$I(X,Y)=\iint p(x,y)\log_2\frac{p(x,y)}{p(x)p(y)}\mathrm{d}x\mathrm{d}y$$

当 X,Y 为离散变量时，则

$$I(X,Y)=\sum_i\sum_j p(x,y)\log_2\frac{p(x,y)}{p(x)p(y)}\mathrm{d}x\mathrm{d}y$$

互信息越大，说明两个变量的相关性越强。

定义 8(信息熵) 设 X 是一个离散随机变量，它可能的取值为 x 的概率为 $p(x)$，那么定义 $H(X)=\sum_{x\in X}p(x)\log_2\frac{1}{p(x)}$。它可以作为数据集合的不纯度或者不规则程度的量度。所谓的不规则程度指的是集合中数据元素之间的依赖关系的强弱。

为了表达方便，区别特征之间的两种类型的相关性，我们给出了以下两个定义。

定义 9(C-相关) 任何的特征 F_i 和类别 C 之间的相关度叫做 C-相关，用 $RMI(i,C)$表示。

定义 10(F-相关) 特征 F_i 和 $F_j(i\neq j)$之间的相关度叫做 F-相关，用 $RMI(i,j)$表示。

为了提高算法的计算效率，我们预先给定阈值 p，再对每个特征计算 C-相关，如果某个特征 F_i 的 C-相关大于预先给定的阈值 p，即 $RMI(i,C)>p$，则该特征与类别 C 的相关度较大，并由此确定特征 F_i 为类别相关的特征。被选择出来的相关特征将在随后进行下一步冗余性分析。

4 动态规划思想及基于动态规划思想的特征选择算法(DPFS)

4.1 动态规划思想

用动态规划法求最优解的问题与我们要获得最优特征子集的问题相类似。动态规划法求最优解的问题，经分解得到的子问题往往不是互相独立，一般动态规划法分为 4 个步骤：(1)找出最优解的性质，并刻画其结构特征；(2)递归地为最优解定义；(3)以自底向上的方式计算出最优解；(4)根据计算最优解过程中得到的信息，构造最优解(有需要时再进行)。

动态规划的基本思想[8]是能够保存已解决的子问题的答案，而在需要时再找出已求得的答案，这样就可以避免大量重复计算，从而减少算法的执行时间。为了达到此目的，可以用

一个表来记录所有已解决的子问题的答案。不管该子问题以后是否被用到，只要它被计算过，就将其结果填入表中。我们将动态规划的这一基本思想结合到对特征相关性和冗余性筛选的过程中，可以大大地减少 C-相关和 F-相关的计算量，从而提高获取最优特征子集的效率。

4.2 基于动态规划思想的特征选择算法(DPFS)

根据特征相关性的 3 个定义，可知要获得一个最优特征子集[7]，可以通过先去除不相关特征，然后再去除弱相关冗余特征得到，所以结合了动态规划思想的基于特征不相关性和冗余性分析的特征选择方法(DPFS)的算法过程如下：

步骤 1　读取数据集并给定一个阈值，通过该阈值对各个数据集中的所有特征的 C-相关值进行筛选。在训练中，我们发现随着阈值 p 的增大，算法的运行时间会相应减少，可以通过增大阈值来提高算法的运行速度。但是在实验过程中必须选择适当的阈值，在提高运行效率的同时，避免去除过多的特征。下面我们使用数据集 Multi-features 进行举例说明，如表 1 所列，分别取阈值 $p=0$ 和 $p=0.1$，随着阈值的增大，运行时间有大幅度的提高，而且两者具有相似的正确率。

表 1　不同阈值的比较

	提取特征个数	运行时(s)	分类正确率
p=0	130	30.2s	95.62%
p=0.1	27	2.3s	95.89%

步骤 2　根据定义 6 分别计算每个特征 F_i 的 C-相关，即 $RMI(i,C)$。结合动态规划的思想，依次将计算的 C-相关进行存储，然后将预先给定的阈值和计算出的 $RMI(i,C)$ 进行比较，如果某个特征 F_i 的 C-相关大于预先给定的阈值 p，即 $RMI(i,C)>p$，则该特征与类别 C 的关联度较大，并由此可以确定特征 F_i 为与类别相关的特征，将已选择的相关特征加入到特征子集 N 中。

滤除不相关特征的算法描述：

```
输入:原始特征集合 S(F1,F2,…,Fn,C)和阈值 p
输出:特征子集 N(不含有不相关特征)
Begin
1. calculate RMI( i,C) for Fi;
  /*对集合 S 中的每一个特征计算其 C-相关*/
2.   store RMI(i,C);
    /*分别对 RMI(i,C)进行存储*/
3.       if(RMI(i,C)>p);
         /*调用存储的 C-相关与阈值 p 进行比较*/
4.       append Fi to N;
         /*将特征 Fi 存放到新集合 N 中*/
end;
```

步骤 3　对整个原始数据集进行了步骤 2 的处理之后，将特征子集 N 中的特征按照 C-相关的大小按降序排列。随后做冗余性分析。

步骤 4　将排在第一个的特征直接添加到最优特征子集中，因为它的 C-相关最大，可以将其看成是强相关特征。

步骤 5　对步骤 3 处理后剩下的相关特征做冗余性分析。通过估计特征之间的关联性来实现冗余分析。在定义 5 中，如果只考虑特征集 $F=\{F_i,F_j\}$，设 $M_i=F_j$，则 $P(F-M_i-F_i,C|F_i,M_i)=P(C|F_i,F_j)$，$P(F-M_i-F_i,C|M_i)=P(C|F_j)$，即在该特征集中，若 $P(C|F_i,F_j)=P(C|F_j)$，则 F_j 是关于 F_i 的一条 Markov Blanket。对于特征集中的特征 F_j 和特征 F_i，当 $RMI(j,C)>RMI(i,C)$ 时，为了要维持特征与类别之间更多的信息，我们将 $RMI(i,C)$ 作为一个阈值来确定 F-相关 $RMI(i,j)$ 是否是可行的，即特征 F_j 与特征 F_i 之间的互信息是否比特征 F_i 与类别 C 之间的互信息更多。分别计算每个特征之间的 F-相关即 $RMI(i,j)$。在这一步中，引入动态规划的思想，将计算出的 F-相关，存储到一个结构中，在进行比较时，直接进行调取，而不用反复计算浪费时间，以便提高获取最优特征子集的效率。

步骤 6　如果 $RMI(i,j)>RMI(j,C)$，那么将特征 F_i 从 N 中移出。

步骤 7　反复进行步骤 7，最终获得一个近似最优的特征子集。

滤除冗余特征并得到近似最优特征集合的算法描述：

```
输入:特征子集 N(不含有不相关特征)
输出:近似最优特征子集 Sbest
Begin
1. order N in descending RMI(i,C) value;
  /*对集合 N 中的每个特征按其 C-相关的大小进行降序排列*/
2. Fj=getFirstElement(N);
    /*将排列在首位的特征当作强相关特征,作为最优特征子集的
      第一个特征*/
      do begin
3. Fi=getNextElement(N,Fj);/*取下一个特征*/
4.      if (Fi! =NULL)/*直到取空为止*/
5.      calculate RMI(i,j);
        /*对集合 N 中的每一个特征计算其 F-相关*/
6.      store RMI(i,j);
        /*分别对 RMI(i,j)进行存储*/
7.      if (RMI(i,j)> RMI(i,C));
         /*调用存储的 C-相关和存储的 F-相关进行比较*/
8.       remove Fi from N;
         /*将特征 Fi 从 N 中移出*/
         Fi=getNext Element(N,Fj)
         end until (Fi= =NULL);
         Fj= getNextElement(N,Fj);
         end until(Fj==NULL);
9. Sbest=N;/*得到近似最优特征子集*/
end;
```

5　实验测试与结果分析

本文提出的特征选择算法主要是针对高维数据的特征选择。在实验中，我们使用了文本数据集 20 newsgroup 中的 sports 数据集作为实验的数据集，为了减少文章中的噪声，首先对所有文档进行预处理，使用 stemming 的方法[9]，完成名词复数的去除、动词时态的转换、动词第三人称转换等工作。另外还要去除一些在文本中有出现频率很高、对文本没有区分作用、并且干扰关键词所起的作用的 stopwords。数据集的原始特征数为 24084，分为 5 类。通过 DF 过滤后，特征数为 11167。利用本文提出的算法(DPFS)对数据集进行特征提取，最终得到了 2487 个特征。在实验过程中，我们把 75%的数据集作为训练集，把 25%的数据集作为测试集。

在文本分类领域中，常用的评价指标是准确率(precision)和召回率(recall)

$$\text{准确率}=\frac{\text{正确分类的数量}}{\text{测试数据的总数}}$$

$$召回率=\frac{正确分类的数量}{预先标记的数据比率}$$

综合两个指标，可以得到一个新的综合评价指标，即 $MicroF1=\frac{2(precision*recall)}{precision+recall}$。我们采用 AMB[10] 和 CHI 两种特征选择算法和本文中给出的算法进行比较、分析。

如图 2 和图 3 所示，当取 10 个特征子集数据进行分类时，DPFS 算法的 *MicroF*1 值小于 CHI 方法的 *MicroF*1 值，但是随着特征的增加，DPFS 结果子集表现越好，当特征数为 1000 时，3 种方法的 *MicroF*1 值基本相同，但是用 DPFS 算法的结果子集进行分类时，会取得很好的结果。在实验中，我们发现当 CHI 方法在特征数达到 10000 时，分类效果出现了降低，这说明了特征为数不断加大，促使大量无关特征和冗余特征的产生，导致分类性能的下降。

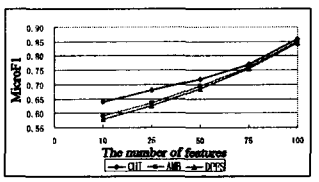

图 2　对低维数据分类的 MicroF1 值比较

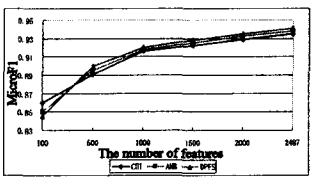

图 3　对高维数据分类的 MicroF1 值比较

在对分类效果做过测试之后，我们又对 3 种算法的运行时间作了比较测试，如图 4 所示，子集特征较少的情况下 CHI 所需要的时间明显要少于其他两种算法，但随着子集特征的维数的增加，本文提出的 DPFS 算法体现出了其优越性，因为本文提出的 DPFS 算法结合了动态规划的思想，即保存已解决的子问题的答案，而在需要时再找出已求得的答案，这样就可以避免大量重复计算，提高运行时间。

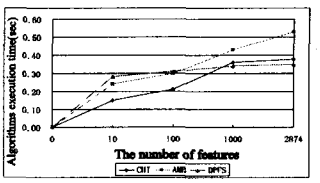

图 4　算法执行时间的比较

综合上述两个实验结果，我们可以看出，传统的 CHI 特征选择算法在选择特征较小时，能够选择出一部分较好的特征，但是不能选择出近似最优的特征子集。AMB 算法虽然能够在这个集合中选择出近似最优的特征子集，但计算量较大。因此可以看出，文本提出的 DPFS 算法不仅能选择出近似最优特征，还通过减少计算量来提高算法的运行时间。

结束语　本文提出的一种基于动态规划思想的特征选择算法，与传统的特征选择算法不同，它能更为全局地考虑，去掉冗余的特征，得到一个近似最优的特征子集。由于其加入了动态规划的思想，在算法的运行时间上也有了一定程度的提高，从实验来看，本文提出的算法与其他算法相比较，有一定的优势。对于将来的工作，我们可能将对本算法的相关值的存储结构进行优化以及将其应用在 Web 挖掘中做一些工作。

参 考 文 献

[1] Greengrass E. Information retrieval: A surrvey[R]. DOD. Maryland. 2000

[2] Yang Yiming. Pedersen J O. A Comparative Study on Feature Selection in Text Categorization [C] // Anon. Proceedings of 14th International Conference on Machine Learning(ICML-97). Nashville: TN, 1997: 412-420

[3] He Jing-song, Shi Ze-sheng. Method of feature selection using signal analysis[J]. Journal of University of science and Technology of China, 2001

[4] Yu L, Liu H. Efficient Feature Selection via Analysis of Relevance and Redundancy[J]. Journal of Machine Learning Research, 2004, 10: 1205-1224

[5] Qu G Z, Hariri S, Yousif M. A New Dependency and Correlation Analysis for Feature[J]. IEEE Trans. on Knowledge and Data Engineering, 2005, 17: 1199-1207

[6] Yu L, LIU H. Feature selection for high-dimensional data: a fast correlation-based filter solution[A] // Proceedings of the Twentieth International Conference on Machine Learning[C]. Washington, 2003: 856-863

[7] Yan J, Liu N, Zhang B, et al. OCFS: Optimal orthogonal centroid feature selection for text categorization[C] // Proceedings of the ACMSIG on Information Retrieval. Salvador, Brazil, 2005: 122-129

[8] 王晓东，计算机算法设计与分析(第三版)[M]. 北京：电子工业出版社，2007

[9] Porter. An Algorithm for Suffix Stripping Program[J]. 1980, 14(3): 130-137

[10] Cui Zi-feng, Xu Bao-wen, Zhang Wei-feng, et al. An Approximate Markov Blanket Feature Selection Algorithm[J]. 2007, 30(12): 2074-2081

(上接第 146 页)

[13] Foukarakis I E, Kostaridis A I, Biniaris C G, et al. Webmages: An agent platform based on web services[J]. Computer Communications, 2007, 30(3): 538-545

[14] Therani M, Uttamsingh N. A declarative approach to composing web services in dynamic environments[J]. Decision Support Systems, 2006, 41(2): 325-357

[15] Zakaria M. On coordinating personalized composite web services [J]. Information and Software Technology, 2006, 48(7): 540-548

[16] Hwang San-Yih, Lim Ee-Peng, Lee Chien-Hsiang, et al. On Composing a Reliable Composite Web Service: A Study of Dynamic Web Service Selection[C] // IEEE International Conference on Web Services. 2007: 184-191