基于特征关联性的特征选择算法研究

重庆大学计算机学院(400044)　李　云　叶春晓　李　季　刘嘉敏　吴中福

摘　要：从特征与特征、特征与类的关联性出发，说明了非搜索性特征选择的原理及相关算法。

关键词：非搜索性特征选择　关联性　特征选择算法

从原始特征集中选出最优特征子集是模式识别和机器学习等领域的一个关键问题，同时也是一个棘手问题。现已证明，最优(最小)特征子集选择 OFSS(Optimal Feature Subset Selection)是 NP 难问题。

实际应用中有二种特征选择的问题：一种是从原始特征集中选出固定数目的特征，使得分类器的错误率最小，这是一个无约束的组合优化问题；另一种是对于给定的允许错误率，求维数最小的特征子集，这是一种有约束的最优化问题。另外特征选择可以分为监督特征选择和非监督特征选择。

特征选择算法从模型上一般可以分为过滤器和封装器。前者是将特征选择作为一个预处理过程，独立于学习算法，而后者是将学习算法的结果作为特征子集评价准则的一部分。一般过滤器模型的时间复杂度较低，准确性不高，而封装器模型的时间复杂度较高，准确性也较高。随着数据量和特征维数的不断增长，过滤器模型更具有现实意义。

特征选择算法从原理上可以分为非搜索性和搜索性算法二种。大部分常用算法都是搜索性特征选择算法。它利用一定的评价准则对所得的子集进行评估，选取最佳特征子集，通常都包含着一个搜索过程。常见搜索算法包括传统算法、遗传算法(GA)、模拟退火算法(SN)、Tabu 搜索算法(TS)和神经网络算法(NN)。传统算法实际是搜索有向图，包括顺序前进法(SFS)、顺序后退法(SBS)、顺序浮动前进法(SFFS)、增 n 减 1 法(PTA)、分支界定法(BB)等，它们各有所长，又都存在一些弊端。关于这方面的讨论已有大量的文献可供参考，这里不再赘述。常见搜索算法包括四个部分：

(1)搜索开始点的选择。如选择特征子集初始为空集或是整个特征集。

(2)搜索策略。与搜索开始点相对应有前向选择、反向剔除以及双向选择等。

(3)评价准则。主要包括二类。一类采用评价函数，如熵、类间距离等；另一类是相关学习算法的学习准确率，它主要用在封装器模型的搜索算法中，如 naive bayes 分类器准确率、KNN 分类器准确率等。

(4)终止条件。指算法结束所要满足的要求。

不同算法在上面四个部分采用的策略不同，如遗传算法经常采用的终止条件为：是否达到预定算法的最大代数，是否找到某个较优的染色体，连续几次迭代后得到的解群中最优解是否变化。

在有关特征选择的文献中，主要是关注搜索性算法，对于非搜索性算法的介绍不多。而机器学习中，随着特征维数的不断增长，非搜索性算法越来越受到重视。非搜索的特征选择通过计算特征间及特征与类别的关联性，通过一定策略消除冗余和不相关特征，其关键之处是特征关联性定义及相关算法的设计。对于分类系统，不相关特征和冗余特征对其分类性能有很大影响，如 naive bayes 分类器对冗余特征很敏感，而最近邻分类器受不相关特征的影响比较大。非搜索性特征选择算法属于过滤器特征选择模型。

1　关联性定义

通常，一个特征被定义为好，则表明该特征与类别标记是相关的，且与其他相关特征是不关联的或弱相关的，即不是冗余的。定义特征子集为好的，则表明其中的特征与类是强关联的，而互相是不关联的。

变量关联性的度量方法大体上可分为二类：一类是线性关联，包括：线性关联系数、Pearn 积矩相关、最小衰减误差平方和最大信息压缩等。另一类建立在信息理论上，如熵等。

现假设有二个变量 X 和 Y，则：

(1)Pearson 积矩相关

$$r_{XY}=\frac{\sum_i\left(x_i-\overline{x_i}\right)\left(y_i-\overline{y_i}\right)}{\sqrt{\sum_i\left(x_i-\overline{x_i}\right)^2}\sqrt{\sum_i\left(y_i-\overline{y_i}\right)^2}}$$

其中：$\overline{x_i}$、$\overline{y_i}$是 X 和 Y 的均值。

(2)线性关联系数

$$\rho_{XY}=\frac{COV(X,Y)}{\sqrt{D(X)D(Y)}}$$

其中：$COV(\ )$，$D(\ )$分别为协方差和方差。

(3)最小衰减误差

$$e_{XY}=D(Y)(1-\rho^2_{XY})$$

(4)最大信息压缩

$$2\lambda_{XY}=\left(D(X)+D(Y)-\sqrt{(D(X)+D(Y))^2-4D(X)D(Y)(1-\rho^2_{XY})}\right)$$

(5)对称不确定性

$$SU_{XY}=2\left[\frac{H(X)-H(X|Y)}{H(X)+H(Y)}\right]$$

其中：$H(X)=-\sum_i P(x_i)\log_2^{P(x_i)}$

$$H(X|Y)=-\sum_i P(y_j)\sum_i P(x_i|y_j)\log_2^{P(x_i|y_j)}$$

$P(x_i)$为 X 所有值的先验概率

$P(x_i|y_j)$为在给定 Y 值下 X 的后验概率

(6)利用 Relief 算法的规则求出特征与类的相关性

Relief 算法从训练集中随机选取 m 个样本实例，通过所选取的样本与属于同类和不同类的二个最近邻样本的差异，求出每个样本的各个特征与类的相关性，然后再求平均值，就得到每个特征与类的相关性。在其扩展算法 ReliefF 中，不是从同类和不同类中各仅选出一个最近邻样本，而是选出 k 个最近邻样本，求平均值，从而得到每个样本实例中各个特征与类的相关性。

2 算法设计

利用特征关联性设计算法的方法，主要分为二类。

2.1 特征关联性作为特征子集的评价准则

通过计算特征子集中的特征关联性来判定特征的优劣。常用的评价函数有如 CFS 算法中利用的 Merit，它综合考虑特征与特征的关联性和特征与类的关联性。还有一些算法将 represent entropy 用来度量特征子集的冗余性，它仅考虑特征间的关联性。由它们构成的算法也属于搜索性特征选择算法。

(1)$Merit_S=\frac{k\overline{r_{cf}}}{\sqrt{k+k(k-1)\overline{rff}}}$（其中：$S$ 为特征子集，k 为特征数，$\overline{r_{cf}}$为所有特征与类的平均关联程度，$\overline{rff}$为特征之间平均关联程度）。它的分子为特征的分类性，而分母为特征子集中特征冗余性。

(2)假设 d 维特征子集所构成的协方差矩阵的特征值为 λ_j，$j=1,\wedge,d$，定义：

$$\lambda'_j=\frac{\lambda_j}{\sum_{j=1}^{d}\lambda_j}$$，则 $0\leqslant\lambda'_j\leqslant1$，$\sum_{j=1}^{d}\lambda'_j=1$

这样 represent entropy 定义为：

$$H_R=-\sum_{j=1}^{d}\lambda'_j\log^{\lambda'_j}$$

实验证明，最后选取的特征子集应该使得 H_R 的取值尽可能地高。

2.2 特征关联性消除冗余特征和不相关特征

(1)利用特征间的关联性(也可以认为是一种相似性)，可以消除冗余特征。消除冗余的算法则包括二个部分：①将原始特征空间进行聚类，通常应该使得特征聚类的冗余性非常高，如参考文献[4]中利用 KNN 聚类；②选择每个聚类中代表性的特征，一般选取距离和最小的特征。它属于非搜索性算法。现将参考文献[4]中给出的具体算法描述如下：

假设特征总数为 D，初始特征集为 $O=\{F_i,i=1,\wedge,D\}$，$S(F_i,F_j)$表示特征 F_i 与 F_j 间的关联性，而 R 为消除冗余的特征集，r^k_i 用来描述 F_i 与其在 R 中的第 k 个最近邻特征之间的关联性。

第一步：初始化，选取 $k\leqslant D-1$，$R\leftarrow o$。

第二步：对每一个特征 $F_i\in R$，计算 r^k_i。

第三步：保留 r^k_i 最小的特征 $F_{i'}$，并抛弃 $F_{i'}$的 k 个最近邻特征，$\varepsilon=r^k_{i'}$。

第四步：IF k>cardinality(R)-1:k=cardinality(R)-1。

第五步：IF k=1：GO TO 第八步。

第六步：WHILE $r^k_i>\varepsilon$DO:

{k=k-1

$r^k_{i'}=\inf\limits_{F_i\in R} r^k_{i'}$

IF k=1：GO TO 第八步}

第七步：GO TO 第二步。

第八步：输出 R。

如果训练集有 n 个样本，则该算法的时间复杂度为 $O(D^2n)$，不同的 k 值可以动态调整特征相关性的门限值，也就是动态调整特征的聚类程度，最终 k 可以控制 R 的大小。

(2)利用特征间的关联性，消除不相关特征，如 Relief 及其扩展算法(Relief，ReliefF，RReliefF)。利用前面描述的方法求出各个特征与类的相关性，并进行排序，相关性大于某个门限值的特征就构成最后的特征子集，这样就消除了不相关的特征，它属于搜索性算法。如果采用 n 个样本，每个样本用 D 个特征来描述，则 Relief 的时间复杂度为 $O(nD)$。

(3)同时消除不相关特征和冗余特征。如 FCBF 算法，采用 SU 来计算特征关联性，它是找出那些与类的关联性大于某一门限值、且大于与其他特征的关联性的特征，并称该特征为支配性(predominant)特征，因此特征选择就转化为鉴别支配性特征。它也是一种非搜索性算法，现将其具体描述如下。

假设特征总数为 D，初始特征集为 $S=\{F_i, i=1, \wedge, D\}$，$SU_{i,j}$ 表示特征 F_i 与 F_j 间的关联性，C 为类别，$SU_{i,c}$ 表示特征 F_i 与 C 间的关联性，δ 为选定的门限值。该算法的伪代码如下：

```
BEGIN
  FOR i=1 to D DO
      计算 Fi 的 SUi,c;
      IF (SUi,c≥δ)
      将 Fi 放入 S′;
   END
  将 S′中特征按 SUi,c 的降序排序;
  Fp=getfirstelement(S′);
DO BEGIN
      Fq=getnextelement(S′,Fp);
      IF (Fq≠NULL)
         DO BEGIN
         F′q=Fq;
         IF (SUp,q≥SUq,c)
           删除 Fq;
           Fq=getnextelement(S′,F′q);
           ELSE
              Fq=getnextelement(S′,Fq);
         END UNTIL (Fq==NULL)
      Fp=getnextelement(S′,Fp);
      END UNTIL (Fp==NULL)
      输出 S′;
END
```

如果训练集有 n 个实例，则该算法的时间复杂度为 $O(nD\log^D)$。

3 应用领域

对于高维特征选择系统，可以充分利用特征关联性进行特征选择或者预处理。采用的模型为：先利用特征关联性剔除不相关特征，再消除冗余特征，最后利用组合特征选择算法进行选择(如 SFFS、SFBS 等)，如图 1 所示。为满足不同的要求，可以选取虚框中一个或者某二个模块实现特征选择。这样就可以构成多种特征选择算法，既包括前面总结的几类，也包括它们与一些搜索性算法的结合，如过滤不相关特征+SFFS、消除冗余特征+SFFS 等。它们之间的顺序是很重要的。通常，遵循该结构的算法，其时间复杂度最小。当然，总的时间复杂度还与每个部分所采用的算法有关。

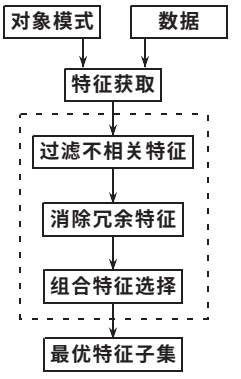

图 1　高维特征选择系统结构

4 结束语

非搜索性特征选择技术比较适合于高维特征选择，时间复杂度相对较低。相反，搜索特征选择的时间复杂度相当高。在当今数据量和特征维数以指数增长的情况下，考虑用非搜索性特征选择技术，利用特征关联性进行特征选择是很有现实意义的，更是值得去研究的一个方向。

参考文献

1　陈彬，洪家荣，王亚东.最优特征子集选择问题.计算机学报，1997；2(20)

2　张鸿宾，孙广煜.TABU 搜索在特征选择中的应用.自动化学报，1999；7(4)

3　Jain A，Zongsker D.Feature selection：evaluation application and small sample performance.IEEE Trans Pattern Recognition and Machine Intelligence.1997；2(19)

4　Mitra P，Murthy C A，Pal S K.Unsupervised feature selection using feature similarity.IEEE Trans Pattern Recognition and Machine Intelligence.2002；3(24)

5　Bins J，Draper B A.Feature selection from huge feature sets.In：Proceedings of the 8th IEEE international conference on computer vision，2001；(2)

6　Koller D，Sahami M.Towards optimal feature selection.In：Proceedings of 13th international conference on machine learning，1996

7　Hall M.Correlation-based feature selection for discrete and numeric class machine learning.In：Proceedings of the 17th international conference on machine learning，2000

(收稿日期：2003-12-20)