一种聚簇消减大规模数据的支持向量分类算法

陈光喜[1] 徐 健[2] 成 彦[1]
(桂林电子科技大学数学与计算科学学院 桂林 541004)[1]
(安徽财经大学统计与应用数学学院 蚌埠 233030)[2]

摘 要 针对支持向量分类机对大规模数据集训练速度慢的瓶颈,提出一种聚簇消减数据集方法。首先建立样本中心距离函数,计算聚簇集的比例半径,然后利用聚簇集镜像扫描样本点确定簇集类,同一类样本特性的聚簇集中只保留代表样本点,建立异类点删除矩阵,通过上述方法消减样本集。证明了这种簇消减算法有较低的时间复杂度,并利用实验说明了保留代表点的有效意义。最后通过随机数据和UCI标准数据库验证了算法在保证分类精度的同时提高了分类速度。

关键词 支持向量机,聚簇集,大规模数据集,训练速度
中图法分类号 TP181 **文献标识码** A

Cluster Method of Support Vector Machine to Solve Large-scale Data Set Classification

CHEN Guang-xi[1] XU Jian[2] CHENG Yan[1]
(School of Mathematics & Computing Science, Guilin University of Electronic Technology, Guilin 541004, China)[1]
(School of Statistics & Applied Mathematics, Anhui University of Finance & Economics, Bengbu 233030, China)[2]

Abstract A cluster Support Vector Machines (C-SVM) method for large-scale data set classification was presented to accelerate speed. Firstly, using function of centre distance calculated radius ratio. Then, data set was scanned by cluster mirror. By remaining representative data for cluster and installing deleted matrix sample set was remarkably reduced. It is proved that the new method has lower time complexity. Experiments with random data and UCI databases verify the efficiency of the C-SVM. Moreover, classification accuracy is gained at adjustment threshold value.

Keywords SVM, Cluster, Large-scale data set, Training speed

1 引言

基于统计学习理论的支持向量机(Support Vector Machine,SVM)是新近研究机器学习、数据挖掘、人工智能领域的一个热点。SVM将求解的问题最终归结为一个线性约束的凸二次规划问题,求出的解是全局最优的唯一解。SVM基于结构风险最小化原则,利用核函数将非线性问题转化为特征空间的线性问题。分类SVM有两个出发点,即最大间隔原则和核技巧。

本文针对SVM对大规模数据集训练速度慢的瓶颈,提出一种聚簇消减数据集方法。通过采用聚簇集镜像扫描样本点,在同一类样本特性的聚簇集中保留代表样本点,并建立异类点删除矩阵,通过上述方法消减样本集。本文证明了这种簇消减算法有较低的时间复杂度,并利用实验说明了保留代表点的有效意义。最后通过随机数据和UCI标准数据库验证了算法在保证分类精度的同时提高了分类速度。

文章第2节概括了支持向量分类机的相关内容;第3节讨论采用聚簇缩减样本集的两种思路,并通过实验改进算法;时间复杂度也将在这部分讨论;随机数据和UCI标准数据库的实验和结论出现在第4部分;最后是改进的方向。

2 线性分类学习机

定义1 训练集 $T=\{(x_1,y_1),\cdots,(x_m,y_m)\}\in(X\times Y)^m$,其中 $x_i\in X=R^n$,$y_i\in Y=\{1,-1\}$,$i=1,\cdots,m$。若存在 $w\in R^n$,$b\in R$ 和正数 ε,使得对所有使 $y_i=1$ 的下标 i,有 $(w\cdot x_i)+b\geqslant\varepsilon$,而对所有使 $y_i=-1$ 的下标 i,有 $(w\cdot x_i)+b\leqslant-\varepsilon$ 则称训练集 T 线性可分。同时也称相应的分类问题是线性可分的[1]。

2.1 处理分类问题的传统方法

一般处理SVM的方法有平分最近点法和最大间隔法。而最大间隔法得到的一个优化问题为QP问题,则更易转化和求解,所以大多数SVM的算法都以最大间隔法为蓝本。

图1表示线性可分的情况,图2表示非线性可分的情况。已知法向量为 w 时分类超平面并不唯一,图中的 l,l_1 和 l_2 就是3个典型的超平面,l_1,l_2 是由 l 平移分别刚好接触到正、负类的点,故选取 w 使 l_1 和 l_2 的间隔达到最大就是要求解的问题。l_1 和 l_2 分别为 $(w\cdot x)+b=1$ 和 $(w\cdot x)+b=-1$,与此对应的分类超平面 l 就为 $(w\cdot x)+b=0$。直接计算可以

到稿日期:2008-04-29 本文受国家自然科学基金(编号:10501009和10661005),桂电软环境项目和安徽财经大学青年基金资助。
陈光喜(1971-),博士,副教授,研究生导师,研究方向为智能算法、数字水印与信息隐藏等;徐 健(1982-),男,讲师,研究方向为数据挖掘、知识发现等。

求得间距为$\frac{2}{\|w\|}$，得到如下规划：

$$\min_{w,b}\frac{1}{2}\|w\|^2 \tag{1}$$

$$s.t.\ y_i((w\cdot x_i)+b)\geqslant 1, i=1,\cdots,m \tag{2}$$

得最优解 w^* 和 b^*，分化超平面$(w^*\cdot x)+b^*=0$，得决策函数 $f(x)=sgn((w^*\cdot x)+b^*)$。

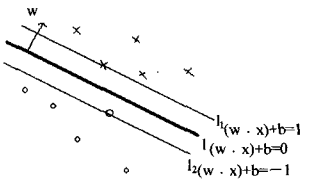

图 1　线性可分的情况

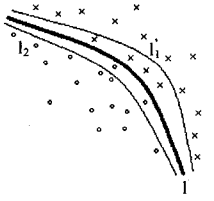

图 2　非线性可分的情况

2.2　最大间隔法的求解与改进

式(1)、式(2)的求解可以转换为求解对偶问题中 lagrange 乘子，由如下的定理来说明。

定理 1　若(w^*,b^*)为原始最优化问题式(1)、式(2)的解，其对偶问题有解 $\alpha^*=(\alpha_1^*,\alpha_2^*,\cdots,\alpha_m^*)$，使得 $w^*=\sum_{i=1}^{m}\alpha_i^* y_i x_i$，$b^*=y_j-\sum_{i=1}^{m}y_i\alpha_i^*(x_i\cdot x_j)$，其中下标 $j\in\{j|\alpha_j^*>0\}$。或者 $w^*=\sum_{i=1}^{m}\alpha_i^* y_i x_i$，$b^*=-(w^*\cdot\sum_{i=1}^{m}\alpha_i^* x_i)/(2\sum_{y_i=1}\alpha_i^*)$，反之也成立[7]。

定理 1 得到了线性可分算法，但对于大规模数据分类来说，一般不容易做到线性可分。

(1)引入松弛变量 $\xi=(\xi_1\cdots\xi_m)^T$，$\sum_{i=1}^{m}\xi_i$ 用来描述训练集错分程度，同时引入惩罚参数 C 平衡$\sum_{i=1}^{m}\xi_i$ 和$\frac{1}{2}\|w\|^2$。式(1)和式(2)变为

$$\min_{w,b,\xi}\frac{1}{2}\|w\|^2+C\sum_{i=1}^{m}\xi_i \tag{3}$$

$$s.t.\ y_i((w\cdot x_i)+b)+\xi_i\geqslant 1,\xi_i>0,i=1,\cdots,m \tag{4}$$

同样可以采用定理 1 的方法得到对偶问题：

$$\min_{\alpha}\frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{m}y_i y_j\alpha_i\alpha_j K(x_i\cdot x_j)-\sum_{j=1}^{m}\alpha_j \tag{5}$$

$$s.t.\ \sum_{i=1}^{m}y_i\alpha_i=0, C\geqslant\alpha_i\geqslant 0, i=1,\cdots,m \tag{6}$$

其中所选取的正分量 α_j^* 要求 $C>\alpha_j^*>0$。

(2)当问题是非线性可分时，通过使用核函数 $K(x_i,x_j)$ 将线性不可分问题变为线性可分问题，将原训练集中的点映射到更高维的 Hilbert 空间中：$\Phi(x):x\to X$，令核函数 $K(x_i,x_j)=(\Phi(x_i)\cdot\Phi(x_j))$，即将对偶问题(5)、(6)中内积$(x_i\cdot x_j)$用核函数代替 $K(x_i,x_j)$，则可以得到支持向量分类机。常用核函数有 Gauss 径向基核 $K(x_i,x_j)=\exp(-\|x_i-x_j\|^2/\sigma^2)$等。

3　聚簇集消减算法

3.1　两种聚簇消减思路与改进

由于大规模数据集的采样一般会出现大量重复信息，训练和实时数据处理中这种重复所产生的时间耗费是非常可惜的，故消减数据集的方法就十分必要。

由于多分类问题通常是分为多个二分类问题来解决，所以本方法先对二分类问题进行讨论。设 $T=\{(x_1,y_1),\cdots,(x_m,y_m)\}\in(X\times Y)^m$ 是可分数据集，$X1$ 为正类点样本集，$X2$ 为负类点样本集，那么在 $X1$ 和 $X2$ 中会出现同一类样本点的大量聚集以及一些极少数的误点，即一个异类点出现在大量的同类点中。而支持向量 SVn 是来构成分类超平面的主要因素，消减数据集的思路就是去除大量的重复样本点和一些明显的误点，保留 SVn 样本点。容易产生以下两种思路：

(1)以聚簇集为单位扫描样本点，如图 3 所示，l 为分类超平面，其中 $A1$ 类聚簇集内聚合了同一类样本点，此时由于这类簇集中不含有 SVn，故删去此类簇集；$A2$ 类聚簇集内可以看到含有一个异类点，这个点极有可能是误点，则把此点记录到异类点删除矩阵；$A3$ 类聚簇集内含有一定数量的两类样本点，那么此类簇集中的点极有可能是 SVn，是决定分类超平面的主要因素，因而保留。通过多次迭代循环修正确定，能有效体现分类特性的消减样本点集。

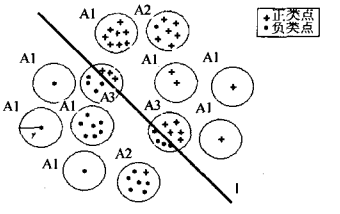

图 3　聚簇集消减数据集示意图

(2)对 $A1$ 类聚簇集的处理方式不同于一般的消减数据集方法全部删除聚簇集，而是只保留一个代表点 Px，即簇集的中心坐标点。

因为在大样本空间中，一个 $A1$ 类聚簇集内同类点很多，故保留一个代表点 Px 并不影响消减数量，而且 Px 可以有效填充分类空间，保证分类空间的正确性。图 4(a)是原始数据集的分类效果图；(b)是按照思路(2)保留了代表点的分类效果图；(c)是按照思路(1)不保留代表点，只对分类面周围的 SVn 点做分类得到的效果图。可以发现，由于(c)图中没有代表点填充分类空间，使得分类空间发生扭曲，得到不正确的分类超平面。可见，仅仅保留而不有效填充分类空间，将会造成分类错误，故而思路(2)是思路(1)的一种很好的改进。

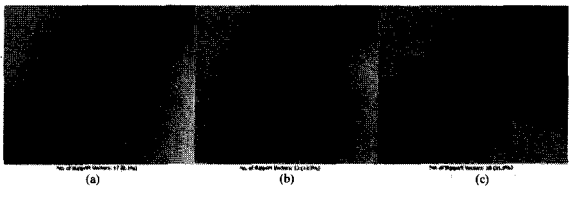

图 4　两种聚类消减算法效果比较图

3.2　聚簇集类与聚簇集半径

由上面的叙述我们采用思路(2)作为聚簇集消减数据集的方法，下面讨论如何确定聚簇集的类别来消减样本集。

定义 2　当输入的样本点集为(x_i,y_i)，$i=1,\cdots,l$，采用欧氏距离 $D(x_i,x_j)=\|x_i-x_j\|_2$ 作为向量之间的距离，定义样本点 x_i 为中心、r 为聚簇半径的聚簇集为 $M(x_i)=\{x_j|\|x_i-x_j\|_2\leqslant r\}$。

定义 3　设 x_i 的聚簇集为 $C(x_i,r)$，则定义其特征 $T(C(x_i,r))$为：

$$T(C(x_i,r))=\begin{cases}1, & count(A_i^+)>count(A_i^-)\\ -1, & count(A_i^+)\leqslant count(A_i^-)\end{cases} \tag{7}$$

其中 $A_i^+=\{y_j \mid x_j\in M(x_i), y_j=1\}$，$A_i^-=\{y_j \mid x_j\in M(x_i), y_j=-1\}$，$count(\cdot)$表示数量，称 x_i 的聚簇集特征 $T(x_i)$为这个聚簇集的类。

聚簇集消减算法的具体思路如下：$\forall(x_i,y_i)\in T$，得到聚簇集 $C(x_i,r)$，并判断其聚簇集类 $T(C(x_i,r))$。如果 $y_i\cdot T(C(x_i,r))=1$，则样本点(x_i,y_i)的类与其簇集类相同；反之若该值为-1，则样本点(x_i,y_i)的类与其簇集类相异。当 $P(x_i)=\left|\dfrac{count(A_i^+)-count(A_i^-)}{count(A_i^+)+count(A_i^-)}\right|\to 1$ 时，说明此时聚簇集内正负类点数量差别很大。如 $y_i\cdot T(C(x_i,r))\cdot P(x_i)=1$，则说明聚簇集内只包含一类样本点，则保留代表点$(x_i,y_i)$；如果定义一个很小的数 $\varepsilon>0$，有 $|y_i\cdot T(C(x_i,r))\cdot P(x_i)-(-1)|<\varepsilon$，则说明聚簇内点$(x_i,y_i)$可能为 $C(x_i,r)$的误点，计入测试矩阵 A。对消减后的数据集进行训练，使用测试矩阵 A 进行测试。如果可分，则消减矩阵 A，这样就可以消减大量的重复反映分类特性的样本点，而保留可以代表分类特性的样本点，从而在保证了精确度的同时提高了训练速度。

下面的问题是确定聚簇集大小。图 4 中 r 为聚簇集的半径，且在同一次数据集镜像扫描中 r 大小不变。如果 r 越小，则含有的样本点比较少，删除的样本点减少，迭代次数增多，极限情况是一个聚簇集中只有一个样本点，故而将删除所有样本点，显然不合适。如果 r 越大，聚簇集中样本点越多，分为两种情况：(1)簇集在同一类别中，那么一次删除的同类样本点增多；(2)簇集落在分类面周围，则其中的点分属于不同类别，故要保留所有样本点，达不到分类效果。极限情况是聚簇集含有所有样本点，那么算法会认为全体样本集都是 SVn，保留所有点，达不到消减效果。所以如何确定一个合适的 r，就显得十分重要。

我们采用找定长距离的比例因子的方法确定 r。设 $T=\{(x_1,y_1),\cdots,(x_m,y_m)\}\in(X\times Y)^m$ 是可分数据集，$X1$ 为正类点样本集，$X2$ 为负类点样本集，C_X1 和 C_X2 分别为正负类样本集的中心点，则按如下式子计算中心点：

$$C\_X1=\frac{\sum_{i=1}^{count(X1)}X1_i}{count(X1)},\ C\_X2=\frac{\sum_{i=1}^{count(X2)}X2_i}{count(X2)} \tag{8}$$

那么中心点之间的距离 $D(C\_X1,C\_X2)=\|C\_X1-C\_X2\|_2$。

设置比例因子 s，令 $r=s\cdot D(C\_X1,C\_X2)$，故得到聚簇集半径 r 的一种计算方法，故而对于不同的比例因子 s 可以采用网格搜索和遗传算法等方法确定 s 的大小。

3.3 算法步骤和时间复杂度分析

3.3.1 算法步骤

对于以上算法可以采用下面的算法步骤来实现。

Step1 分类数据集 $T=\{(x_1,y_1),\cdots,(x_m,y_m)\}\in(X\times Y)^m$ 为正类集 $X1$ 和负类集 $X2$，计算两类集的中心坐标：C_X1，C_X2 和中心点之间的距离 $D(C\_X1,C\_X2)$。

Step2 设置半径比例参数 s，计算聚簇集半径 $r=s\cdot D(C\_X1,C\_X2)$。并使用此半径 r 镜像扫描数据集 T，得到以样本点 x_i 为中心的聚簇集 $C(x_i,r)$和聚簇集特征 $T(C(x_i,r))$。

Step3 判断同类点簇集和异类点簇集，设置一个很小的判断参数 $\varepsilon>0$ 建立和记录矩阵 A 用来记录疑似误点。

1)同类点簇集：如果 $y_i\cdot T(C(x_i,r))\cdot P(x_i)=1$，说明此聚簇集中所有的点都为同一类样本点，那么保留样本点 x_i 为代表点，消减其余的样本点。

2)异类点簇集：如果 $|y_i\cdot T(C(x_i,r))\cdot P(x_i)-(-1)|<\varepsilon$，说明此聚簇集中的样本点 x_i 可能为误点，那么记录检测样本点到记录矩阵 A。

3)其余情况：说明聚簇集 $C(x_i,r)$中两类样本点数量相差不大，故而可能为 SVn，那么把聚簇集中所有的点保留在样本集中。

Step4 设置精确度 P，如果消减后的样本点集 new_T，训练样本集的精确度大于 P，则采用 new_T 作为最终训练样本集，反之转 Step5。

Step5 使用矩阵 A 修正样本点集得到新样本集 new_T，将原始样本集并转 Step4。

3.3.2 时间复杂度

这种簇集消减算法每次只对一个聚集进行两种运算，即判断同类点聚集还是异类点簇集。设样本集大小为 n，由于对同类簇集中所有的点只保留代表点，那么最终扫描的样本点的个数为 m 个，且 m 可视为 n 和聚簇半径 r 的函数，而半径 r 与参数 s 有关，则 $m=f(n,s)$，且 $m<<n$。扫描的计算复杂度为 $O(m)$。其次，由于邻域的大小将决定样本数的多少，假设邻域内包含 t 个样本，则需要进行两种计算，其复杂度为 $2*O(t)$，而 t 的大小与邻域半径即阈值 s 有关，那么每一个邻域内的计算复杂度可以看成是阈值 s 的函数 $O(f(s))$。

综上所述，得到

定理 设样本集的大小为 n，则预处理算法的计算复杂度为 $O(f(n,s))$，其中 s 为所设定的半径比例因子。

这种算法相对于文献[7]消减算法的不同在于降低了计算的时间复杂度。文献[7]的预处理算法在处理样本集时需要对样本集中每个样本点进行邻域特征、正类点比例、负类点比例和孤立点判别 4 种计算操作，并且需要扫描整个样本集大小为 n；而聚簇消减算法扫描的样本集为 m 个，即一边消减一边扫描，最终 $m<<n$。而且对每个样本只进行两类计算，所以可以有效地降低时间复杂度。

4 相关数据实验

为测试聚簇集消减算法，我们采用两种数据实验：随机的数据和 UCI 标准数据库的实验数据。

4.1 使用随机数据实验

图 4 给出了一个对 210 个屏幕随机点取的二维样本点进行的数据实验，选取聚簇集半径比例因子 $s=1/8$，保留所有可能的分类数据点，即不讨论误点情况，只删减同类聚集点，得到图 4(a)为未消减数据集的分类情况；(b)对消减后的数据集的分类情况。实验说明减少了样本集并降低了训练时间。从图上可以发现，分类超平面并没有发生特别显著的变化，精确度的降低可以控制在允许范围之内，分类空间也由代表点进行了有效填充。而且没有减少支持向量，从而也就不会使训练精度降低。可以看出，在提取后训练样本集从 210 个减少到 84 个，减少了 60%，但保留下来的数据点依然很好地体现了分类的特性。

对随机产生的从 10 个到 500 个样本点的数据集进行实

验，选取聚簇集半径比例因子 $s=1/8$。其中分类正确率都是对同一个原始样本集进行测试的，如图 5 所示。

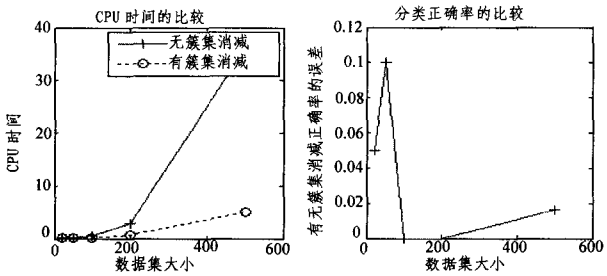

图 5　有无簇集消减方法结果比较图

可以明显看到，随着训练数据集的增大，样本的训练时间差别逐渐变大。图 5 中，在样本集为 500 时，没有簇集消减的训练时间超过了 30s，而采用了簇集消减的时间仅是其 1/6。而且在预测方面，可以看到不论数据集的变化如何，采用和不采用簇集消减算法的训练精度都相差 0.1 之内。并随着数据集增大，训练精度基本不发生大的变化，所以这种聚集消减数据集的方法是可以在保证训练精度的同时减少训练时间的。

4.2　使用 UCI 标准数据库实验

使用预处理方法对 UCI 标准数据库中的 WBCD（美国威斯康辛医学院的乳腺癌数据库）进行处理进行对比实验。数据集一共是 683 个数据样本，其中正类样本（良性肿瘤）444 个（65%），负类样本（恶性肿瘤）239 个（35%）。实验设计将前 383 个样本作为训练数据样本（其中正类数据 214 个（55.9%），负类样本 169 个（44.1%）），用来得到分类超平面。用剩余的 300 个数据作为测试数据样本（其中正类样本 230 个（76.7%），负类样本 70 个（23.3%）），用来检验分类正确率。

采用 CPU 为 AMD Turion (tm) 64 2.01GHz，512M 内存，Microsoft Windows XP(SP2) 系统，Matlab 7.0 进行编程设计。分别对设置聚簇半径比例因子 $s=1/4,1/8,1/16$ 进行比较实验，结果如表 1 至表 4 所列。

1）SVM 训练时采用 Gauss RBF 核（因子为 5）；惩罚因子 $C=300$。

表 1　普通算法得到的结果

样本点(n)	nsv	Nsv%	分类率(%)	CPU 训练时间(s)
383	72	18.8%	96.67%	17.7344

表 2　采用聚集消减得到的结果

	样本点	nsv	Nsv	分类率	CPU 时间(s)	
	(n)		(%)	(%)	消减	训练
s=1/4	146	64	43.8%	96.33%	0.09375	1.5313
s=1/8	190	51	26.8%	97.33%	0.15625	2.9688
s=1/16	191	51	26.7%	97.00%	0.125	2.875

2）SVM 训练时采用 Gauss RBF 核（因子为 1000）；惩罚因子 $C=3000$。

表 3　普通算法得到的结果

样本点(n)	nsv	Nsv%	分类率(%)	CPU 训练时间(s)
383	65	17.0%	99.00%	16.5156

表 4　采用聚集消减得到的结果

样本点	样本点(n)	nsv	Nsv	分类率	CPU 时间(s)	
(n)			(%)	(%)	消减	训练
s=1/4	146	47	32.2%	99.00%	0.10938	1.3906
s=1/8	190	39	20.5%	99.00%	0.14063	2.6406
s=1/16	191	42	22.0%	99.00%	0.14063	2.8125

4.3　结论

从以上两组实验的 4 个表可以发现：

1）从算法的正确率上看，采用聚簇消减算法的分类正确率并没有发生降低，反而在第一组实验中 $s=1/8,1/16$ 时分类正确率到达了 97%，高于普通算法 96.67%。而在第二组实验中调整了 SVM 训练的参数，得到训练精度达到了 99.00%，而且在采用了聚簇集消减算法后训练精度保持在了 99.00%，没有发生变化 。说明保留下来的样本点可以很好地体现分类特性，而且通过改变比例数值可以更好地体现出比原始算法更为准确的分类率。

2）从训练时间来看，两组数据实验中消减数据集的时间仅为 0.1s 左右，而消减完后数据集减少为原训练数据的 1/2 左右，训练时间由 17.7344s，16.5156s 减少为 1～2s 左右。可见，由于数据集消减的有效消减，使得训练时间大幅度降低，仅为原来的 1/16 左右。而且随着数据集的进一步增大，原始算法的训练时间增加将是惊人的，但是采用了我们的聚集消减算法后，可以有效地抑制训练时间的增加，从而解决了 SVM 算法的训练速度的瓶颈。

故而我们的聚簇消减算法可以很好地保持数据集的分类特性，从而保证了分类精度，所以消减的方式是可以信赖的。

结束语　本文主要对 SVM 在训练时间上进行改进。通过对传统 SVM 的论述，并采用对原有的一些预处理方法在处理数据时速度慢的情况，提出一种基于聚簇集消减数据的方法。本方法保证了消减后的数据集的分类特性不变，从而得到分类精度不变或提高。主要达到了提高训练速度的目的，这在 SVM 的应用中将有很大的实际意义。如果适当调整聚簇集方法中的簇集半径比例因子，可以达到既减少训练时间又提高精度的双赢目的，所以在阈值的选取上还可以继续研究。本方法中是采用实验搜索比例因子的办法，还可以应用网格法以及遗传算法来对相关因子进行搜索，得到最佳比例因子，这些方法都是值得继续研究的方向。我们发现，这种聚簇集消减的算法是针对数据集的消减的，然而消减完的数据集可以很好地保持原有的分类特性，所以不光可以应用在 SVM 训练中，还可以应用到其他的方法中，具有很好的推广性。

参 考 文 献

[1]　邓乃扬，田英杰. 数据挖掘中的新方法——支持向量机[M]. 科学出版社，2004

[3]　王晓丹，王积勤. 支持向量机训练和实现算法综述[J]. 计算机工程与应用，2004，13：75-78

[4]　李红莲. 一种改进的支持向量机 NN-SVM[J]. 计算机学报，2003，26(8)：1015-1020

[5]　白亮，老松杨. 支持向量机训练算法比较研究[J]. 计算机工程与应用，2005，17：79-81

[6]　胡懋智. 各种不同类型的 SVM 及其性能比较分析[J]. 计算机工程与应用，2005，12：27-40

[7]　徐健，陈光喜. 处理大规模数据分类的支持向量预处理方法[J]. 计算机应用，2007，27：257-260

[8]　Zheng Chun-Hong，Jiao Li-Cheng. Fuzzy Pre-extracting Method

For Support Vector Machine[A]// Proceedings of the First International Conference on Machine Learning and Cybernetics. Beijing, November 2002: 4-5

[9] Mangasarian O L, Musicant D R. Successive overrelaxation for support vector machines[J]. IEEE Tangasarian on Neural Networks, 1999, 10: 1032-1037

[10] Vapnik V N. Statistical Learning Theory[M]. New York: Wiley, 1998

(上接第 128 页)

证明该关联方法确实通过 SWRL 规则关联了另一简单对等本体；2) 分析器根据 SWRL 规则进行相关的查询操作，并记录了整个查询过程中访问到的本体，其中 count 是查询深度，即查询某个类的实例时最多可关联的其它类的个数。本程序设定 count 为 2，此值可根据具体情况进行修改。

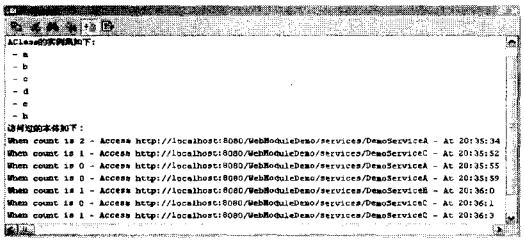

图 1 程序运行结果图

3.2 结果分析与讨论

3.2.1 SWRL 规则扩展的有效性

在本实验中，分析器对形如$\neg CClass(?\ x)$的规则的语义能正确解释，把查询返回的实例$\{\{c,d,e,f\}\}$作为待拒绝的结果集。这使扩展后的 SWRL 规则基本上能支持类间的逻辑与、逻辑或和逻辑非操作，符合更多具体应用的要求。用于测试的本体应用程序最终得到的类 *AClass* 的实例集是$\{a,b,c,d,e,h\}$，与推论出的最终被返回的"待返回的结果集"$\{a,b,c,d,e,h\}$相同。表明基于 SWRL 规则的本体关联方法能够实现简单对等本体之间类的映射。

3.2.2 本体自动动态更新

实验中返回的 *AClass* 的实例集与初始给出的 *DemoA* 中 *AClass* 的实例集并不相同，这是因为分析器根据用户编写的 SWRL 规则关联了其它的简单对等本体，从关联的本体中进行类 *AClass* 的实例查询，从而获得并返回更完整的类 *AClass* 的实例和信息。

测试用例中的规则分析器实现了对 SWRL 规则的语义分析，是一个基于一阶谓词逻辑的 SWRL 规则的推理机。在测试用例的实现过程中，不同本体之间是一种对等的关系，每个本体只存放了高一层大本体的部分知识，本体的构建者只需要维护和处理自己的本体以及不同本体间映射关系的构建。

3.2.3 基于关联的重用机制

如果本体的使用者想重用已有本体中构建的概念(类)，或者从已有本体中查找出与自己的本体相关的概念(类)知识，那么我们不需要使用相关度计算来实现本体之间概念(类)的映射。从上述测试用例看出，可以通过 SWRL 规则去关联不同的本体中的类，实现异构本体间类的通信，从而从不同的角度去解决本体中类的重用问题，而且这种方法并不针对某一领域本体，具有灵活性和通用性。

结束语 本文提出的"基于 SWRL 规则的本体关联方法"，对以下几方面问题进行了初步研究：通过 SWRL 规则实现了多个本体之间在类层面的对应关系，由此实现了在"简单对等(peer-to-peer)本体"设想中简单本体之间的关联；理论上，根据本体的一般定义，SWRL 规则支持的一般逻辑谓词的语义可以被解释为有名的类或者是匿名的类[7]，类是本体中的重要组成部分，开发一个本体的基本过程是围绕着类来进行的[12]。本文即是利用 SWRL 规则，从集合论的角度分析了 SWRL 规则的语义，对其语义基于查询结果进行了扩展，并设计了"SWRL 规则分析器"实现之。在现阶段，本文提出的"基于 SWRL 规则的本体关联方法"实现的只是本体之间在类上的对应关系。在后续的研究中，可以通过增加相应的把 SWRL 规则支持的一般逻辑谓词的语义解释为类的功能，使得本文提出的"SWRL 规则分析器"能够处理更加一般的由 SWRL 规则描述的本体之间的关联。

参 考 文 献

[1] Rousset M - C. Small Can Be Beautiful in the Semantic Web. 2004. http://www.springerlink.com/(qqu30yz2tpautoadmmfoax45)/app/home/contribution.asp?referrer=parent&backto=issue,2,57

[2] 郑丽萍. 本体映射的研究. 硕士学位论文. 山东科技大学, 2005

[3] Office, D. s. I. E. The DAML Rules. 2004. http://www.daml.org/swrl

[4] W 3 C. OWL Web Ontology Language Semantics and Abstract Syntax. 2004. http://www.w3.org/TR/2004/REC-owl-semantics-20040210

[5] Boley H B G, Tabet S. Rule Markup Tutorial. 2005. http://www.ruleml.org/papers/tutorial-ruleml-20050513.html

[6] Smith M K, Deborah C W, McGuinness L. OWL Web Ontology Language Guide. 2004. http://www.w3.org/TR/owl-guide

[7] Horrocks I, Harold Boley P F P-S, Tabet S, et al. SWRL: A Semantic Web Rule Language Combining OWL and RuleML. 2004. http://www.w3.org/Submission/2004/SUBM-SWRL-20040521

[8] 左孝凌，李为监，刘永才. 离散数学. 第一版. 上海：上海科学技术文献出版社，1982

[9] Ian Horrocks P F P-S, Bechhofer S, Tsarkov D. OWL rules: A proposal and prototype implementation. Web Semantics, 2005: 23-40

[10] Knublauch H, et al. The Protege OWL Plugin: An Open Development Environment for Semantic Web Applications. 2004

[11] Horridge M, et al. A Practical Guide To Building OWL Ontologies Using The Protege-OWL Plugin and CO-ODE Tools Edition 1.0. 2004. http://protege.stanford.edu/doc/users.html

[12] Vet P E v d a N J I M. Bottom-Up Construction of Ontologies. 2002. http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=706054