平滑支持向量机聚类研究

李传宗，席 斌，耿 代
（厦门大学 模式识别与智能系统研究所 福建 厦门 361005）

【摘 要】：支持向量聚类(Support Vector Clustering, SVC)的运算有较高的计算复杂性，本文在优化过程中引入惩罚函数，以此作为目标函数的惩罚项，并用一个平滑函数来近似正号函数，并将优化问题的不等式约束消去，得到一个无约束问题。再利用 BFGS- Armijo 算法来求解该无约束问题。理论和仿真结果表明该方法提高优化问题的求解效率。

【关键词】：支持向量机；支持向量机聚类；平滑函数；BFGS- Armijo 算法；MST

核函数是支持向量机的基础，支持向量机 (Support Vector Machine, SVM)被广泛用于解决分类、回归及新知识检测(Novelty Detection)等问题，近来又被用作聚类分析。于文献[3]中，Tax 将支持向量方法用于数据域描述 (即单一分类) 中，提出了基于 Gauss 核的 SVDD(Support Vector Domain Description)算法，Ben-Hur 等将该算法进一步发展为一种新的无监督非参数型的聚类算法---支持向量聚类(Support Vector Clustering, SVC). SVC 算法主要分为两部分：基于支持向量机训练和聚类标识。其中 SVM 训练部分负责新知识模型的训练，包括 Gaussian 核宽度系数的优化、Hilbert 空间最小包络超球体半径的计算、Lagrange 乘子的计算以及有界支持向量(Support Vector, SVs)的选取。聚类标识部分首先生成聚类标识关联矩阵，再通过 DFS(Depth-first Search)算法根据关联矩阵进行聚类分配。

SVC 算法的瓶颈主要集中在两方面：计算 Lagrange 乘子的二次问题和关联矩阵的计算。因此本文提出了平滑支持向量机聚类(Smooth Support Vector Clustering-SSVC)。在训练支持向量点时，通过引入惩罚函数，并在目标函数中对其进行惩罚，将原问题转化为二次无约束优化问题，并引入平滑方法优化该问题。这样就可以采用传统的 Newton 方法进行求解，大大提高了支持向量点的训练效率。而聚类标识运算则采用文献[1]中的方法进行求解。经过大量的数值实验表明：对于线性可分的数据，SSVC 与传统的聚类方法都能取得较好的聚类效果；但对非线性可分的数据，SSVC 不仅降低了聚类的运算时间，而且较传统方法能得到更好的聚类效果。特别对于大数据集，SSVC 的优势更为明显。

1. SVC 边界描述及聚类标识算法

基于文献[3]，Ben-Hur 等提出的支持向量聚类算法如下：设数据空间 $\chi\subseteq R^d$，数据集 $\{x_i\}\subseteq\chi$，包含 n 个数据点，运用非线性变换 Φ 将数据从 χ 映射到高维特征空间，寻找 Hilbert 空间最小包络超球体半径 R，表达为：

$$\|\Phi(x_i)-a\|^2\le R^2+\xi_i \quad (\forall_i,\xi_i\ge 0) \tag{1}$$

式中：a 为超球体球心；ξ_i 为松弛变量。ξ_i 缩小了超球体半径，允许软边界的存在。引入式(1)的 Lagrange 形式：

$$\min_{R,a,\xi}\max_{\beta,\mu} L=R^2-\sum_i(R^2+\xi_i-\|\Phi(x_i)-a\|^2)\bullet\beta_i-\sum\xi_i\mu_i+C\sum\xi_i \tag{2}$$

式中：$\beta_i\ge 0,\mu_i\ge 0$，均为 Lagrange 乘子；C 为常数；$C\sum\xi_i$ 为惩罚项。根据 Karush-Kuhn-Kucker 条件，将式(2)转化为 Wolfe 对偶形式：

$$\begin{cases}\max\limits_{\beta_i} W=\sum\limits_i(\Phi(x_i))^2\beta_i-\sum\limits_{i,j}\beta_i\beta_j\Phi(x_i)\Phi(x_j)\\ s.t.0\le\beta_i\le C,\sum\limits_i\beta_i=1,i=1,2,\ldots,n\end{cases} \tag{3}$$

根据 SVM 理论，用 Mercer 核来表示输入空间在特征空间的点积形式，即

$$K(x_i,x_j)=\Phi(x_i)\bullet\Phi(x_j) \tag{4}$$

因此式(3)可表示为

$$\begin{cases}\max\limits_{\beta_i} W=\sum\limits_i K(x_i,x_i)\beta_i-\sum\limits_{i,j}\beta_i\beta_j K(x_i,x_j)\\ s.t.0\le\beta_i\le C,\sum\limits_i\beta_i=1,i=1,2,\ldots,n\end{cases} \tag{5}$$

求解该二次规划得到 $\{\beta_i\}$。根据文献 [3] 知道 β_i=C 的点位 BSVs，$0<\beta_i<C$ 的点属于 SVs。则任一点在特征空间中的像到球心的距离可表示为

$$R^2(x)=K(x,x)-2\sum_i\beta_iK(x,x_i)+\sum_{i,j}\beta_i\beta_jK(x_i,x_j) \tag{6}$$

球体半径 R 为任意支持向量到球心的距离：$R=R(x_i)$，x_i 为支持向量。因此，聚类边界可由数据空间中满足 $\{x|R(x)=R\}$ 条件点的包络线集来构建。其中 SVs 在边界上，BSVs 在边界外，其它点在聚类范围内。

对于聚类标识的运算采用文献[1]的方法。设数据空间 $\chi\subseteq R^d$，对于数据集 $\{x_i\}\subseteq\chi$，映射到 Hilbert 空间 Φ 的 Euclidean 距离可表示为：

$$d_H(x_i,x_j)=[K(x_i,x_i)-2K(x_i,x_j)+K(x_j,x_j)]^2 \tag{7}$$

应用 Gauss 核简化式(7)得到 Hilbert 空间中边的权重表达式为：

$$w(x_i,x_j)=\sqrt{1-K(x_i,x_j)} \tag{8}$$

根据式(8)生成输入空间完全图各边的权重(不计 BSVs)，采用 Kruskal 算法，生成最小生成树(Minimum Spanning Tree, MST)及边集合 E；实验中发现，MST 中聚类的分割往往发生在树的主干部分。因此经过运算得到 MST 的主干树 。定义 MST 中所有与任意顶点 相连各边平均权重为：

$$\overline{w_v}=\frac{1}{d(v)}\sum_{ue V}w(u,v) \tag{9}$$

式中：V 为 MST 中与 v 相连的顶点的集合；$d(v)$为顶点 v 的度；u 为 *MST* 中所有与 v 相关联的顶点；$w(u,v)$为 v 和 u 的连接权重。则定义边 $e(u,v)$，定义其不相容性度量

$$\phi(u,v)=\max(w(u,v)-\overline{w_v},w(u,v)-\overline{w_u}) \tag{10}$$

则 MST 主干边集合进行聚类标识运算步骤如下：

1)计算 E^* 中各边的 ϕ 函数值

2)搜索具有最大 ϕ 值得边 $e_{max}(u,v)$

3)对 $e_{max}(u,v)$进行聚类标识运算，如果其间存在样点 x，有 R(x)≤R，则结束运算并返回，否则执行 4)

4)由 E^* 中去除 $e_{max}(u,v)$，同时标识关联矩阵 B：$B_{uv}=B_{vu}=0$；获得与 $e_{max}(u,v)$端点 u 相连通的边集合 E_u^* 及端点 v 相连通的边的集合 $E_v^*=E^*-E_u^*$

5)转到(2)，重复对 E_u^*,E_v^* 的运算

6)对 B 进行 DFS 运算，得到所有数据点的最终聚类

2. 改进的优化算法

原 SVC 的优化问题为如下：

$$\begin{cases}\max\limits_{\beta_i} W=1-\sum\limits_{i,j}\beta_i\beta_jK(x_i,x_j)\\ s.t.0\le\beta_i\le C,i=1,2,\ldots,n\end{cases}$$

即为：

$$\min_{\beta_i} = \sum_{i,j} \beta_i \beta_j K(x_i, x_j)$$
$$s.t \quad \beta_i \le C \quad 且 \quad \beta_i \ge 0 \tag{11}$$

那么根据最优化原理，引入惩罚函数，得到：

$$P(\beta) = \begin{cases} f(\beta), c(\beta) \le 0 \\ f(\beta) + \frac{v}{2}\|c(\beta)\|_2^2, c(\beta) > 0 \end{cases} \tag{12}$$

又由于 $c(\beta)>0$，因此式(12)可写成下面的式子：

$$\min \quad W' = \sum_{i,j} \beta_i \beta_j K(x_i, x_j) + \frac{v}{2} \sum_{i \in I} \|(c_i(\beta))\|_2^2 \tag{13}$$

$$\min \quad W' = \frac{1}{2}\sum_{i,j} \beta_i \beta_j K(x_i, x_j) + \frac{v}{2}(\sum_j \|(C-\beta_j)_+\|_2^2 + \sum_j \|(-\beta)_+\|^2) \tag{14}$$

其中 $(x)_+$ 为正号函数，其表达式为：$(x)_+ = \begin{cases} x, x \ge 0 \\ 0, x < 0 \end{cases}$

由此得到一个无约束的凸二次最优化问题，用迭代算可以高效的求解出该问题的唯一最优解。但是目标函数(15)中惩罚项部分不可导，为使用 Newton 方法，我们使用平滑技术，用一个平滑函数 $p(x,a)$ 来近似正号函数。且平滑函数 $p(x,a)$ n 阶可导。该平滑函数如下表示：

$$p(x,a) = x + \frac{1}{\alpha}\log(1+e^{-\alpha x}), \alpha > 0 \tag{15}$$

那么式(15)可写成：

$$\min \quad W' = \sum_{i,j} \beta_i \beta_j K(x_i, x_j) + \frac{v}{2}\left(\sum_j \|p((C-\beta_j), \alpha)\|_2^2 + \sum_j \|p((-\beta_j), \alpha)\|_2^2\right) \tag{16}$$

其中 a 是平滑参数，v 为常数。

BFGS(Brogden-Fletcher-Goldfarb-Shanno)算法，是解无约束最优化问题最有效的算法之一，它不需要求二阶 Hesse 矩阵，而只利用一阶导数来构造二阶信息的近似矩阵，从而该算法有较好的收敛性质。

3.改进算法的聚类效果仿真

为与传统支持向量机聚类算法进行比较，本文采用 UCI 数据集的 houseware,wine 和 waveform 数据集，用于算法运算时间的实验。还采用 Iris 数据集和圆圈数据集进行聚类效果的对比实验。

实验首先对比了 SMO 算法与平滑支持向量机求解聚类支持向量点所花费的时间，然后又对比了各种聚类标识算法所花费的时间。下面表 1 表示的是平滑支持向量机与 SMO 的求解所花时间对比；表 2 表示的是对比各种聚类算法的时间花费对比。

数据集		支持向量点个数	SMO 求解时间 (s)	SSVC 求解时间(s)
维度	大小			
3	300	57	33.84	21.7
4	300	63	30.19	18.87
6	300	61	35.56	19.23
10	219	42	25.37	12.78
13	178	35	24.53	10.3

表 1

聚类标识时间			错误率（%）		
原始算法(s)	邻接图	最小生成树算法(s)	原始算法(s)	邻接图	最小生成树算法(s)
9095	267	241	28.7	27	15.52
--	412	311	--	18.2	12.4
--	561	521	--	23.17	16.92
6006	1002	827	29.1	15.2	18.1
1582	812	615	28.7	24.7	20.24

表 2

为验证本文的聚类效果，对比了传统聚类方法：K 均值聚类、自组织映射 SOM。对于不同的两种数据分别进行实验，得到如下的实验结果。

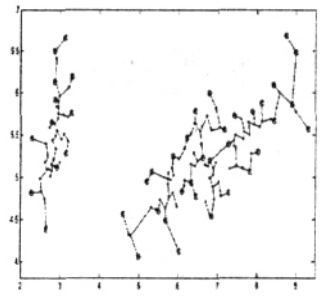

图 1 SSVC Iris 的聚类结果

	K-means	SOM	MST-SSVC
K=2	33.33%	66.67%	
K=3	4%	6%	4%(q=5,r=1)
K=4	40%	18.66%	
K=5	83.33%	38%	

表 3 Iris 数据聚类错分率比较

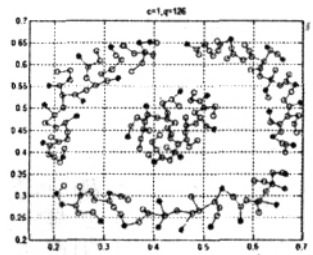

图 2 SSVC 圆圈数据聚类结果

	K-means	SOM	MST-SSVC
K=2	89%	88.5%	
K=3	64.5%	59.7%	
K=4	27%	37%	5.3%(q=122,r=1)
K=5	87%	88%	

表 4 圆圈数据聚类错分率的比较

通过以上的数值实验表明：

1)对比以上支持向量机的算法，SSVC 可以在精度基本不变的情况下，改善 SVC 聚类算法的时间性能。

2)传统算法对于初始值较为敏感，而 SSVC 没有初值选择这个问题，因而新算法具有更好的健壮性。

3)从实验效果来看，算法对非线性，非高斯分布的数据比传统算法有着更好的聚类效果。

4. 结论

对于 SVC 支持向量的训练，本文借鉴文献[2]的方法，通过引入惩罚函数，并在目标函数中对其进行二次范数的惩罚。结合正号函数的特性及该优化问题的特性，用一正号函数来代替原惩罚项，并消去了约束。最终将一个约束二次规划问题转化一个无约束的二次规划问题。由于正号函数不可导，因此本文中用一个平滑函数近似该正号函数，并采用 BFGS_Armijo 算法求解二次规划问题。这样大大提高了支持向量的训练效率。对于大样本集的情况，本文的算法较其他常用的算法如 SMO, SOR 等有较大的时间复杂性优势。对于聚类标识的运算，本文采用了文献[1]中的方法，进一步的提高了聚类效率和质量。

参考文献：

1.吕常魁、姜澄宇、王宁生 一种支持向量聚类的快速算法 [J] 华南理工大学学报 2005; 33(1)

2.Yuh- Jye Lee,Wen- Feng Hsieh,Chien- Ming Huang. A Smooth Support Vector Machine for Classification[J]. Cinputational Optimization and Applications, 2001, 22(1) : 5~21

3.Asa Ben- Hur David Horn Hava T.Siegelmann. Support Vector Clustering[J]. Journal of Machine Learning Research, 2001,2 :125- 137

4.Fredric M.Ham Ivica Kostanic Principles of Neurocomputing for Science & Engineering[M]. 北京: 机械工业出版社, 2003

5.薛毅 最优化原理与方法[M]. 北京: 北京工业大学出版社, 2001

6.Sing- Tze Bow. Pattern Recognition lgorithm[M]. New York: Electrical engineering and electronics, 1984

7.J.Platt. Fast training of support vector machines using sequential minimal optimization[M]. InB.Schlkopf,C.J.Burges, and A.J.Smola, editors AdvancesinKernel Methods- Support Vector Learning. MITPress, 1999

8.R.Ng, J.Han. Efficient and effective clustering method for spatial data mining[J]. Proc.1994 Int.Conf.Veery Large Data Bases(VLDB'94), 144- 155

9.S.Guha, R.Rastogi, K.Shim.CURE. an efficient clustering algorithm for large database[J]. Information Systems. 2001,26(1)8

10.Francesco Camastra. A Novel Kernel Method for Clustering[J]. IEEE Transactions on Knowledge And Data Engineering. Vol.27,No.5,MAY 2005