第 24 卷 第 1 期
2001 年 1 月

计 算 机 学 报
CHINESE J. COMPUTERS

Vol. 24 No. 1
Jan. 2001

基于支持向量机与无监督聚类相结合的中文网页分类器

李晓黎 刘继敏 史忠植

(中国科学院计算技术研究所 北京 100080)

摘 要 提出了一种将支持向量机与无监督聚类相结合的新分类算法,给出了一种新的网页表示方法并应用于网页分类问题.该算法首先利用无监督聚类分别对训练集中正例和反例聚类,然后挑选一些例子训练 SVM 并获得 SVM 分类器.任何网页可以通过比较其与聚类中心的距离决定采用无监督聚类方法或 SVM 分类器进行分类.该算法充分利用了 SVM 准确率高与无监督聚类速度快的优点.实验表明它不仅具有较高的训练效率,而且有很高的精确度.

关键词 支持向量机,聚类,网页分类

中图法分类号:TP391

A Chinese Web Page Classifier Based on Support Vector Machine and Unsupervised Clustering

LI Xiao-Li LIU Ji-Min SHI Zhong-Zhi

(*Institute of Computing Technology*, *Chinese Academy of Sciences*, *Beijing* 100080)

Abstract This paper presents a new algorithm that combines Support Vector Machine (SVM) and unsupervised clustering. After analyzing the characteristics of web pages, it proposes a new vector representation of web pages and applies it to web page classification. Given a training set, the algorithm clusters positive and negative examples respectively by the unsupervised clustering algorithm (UC), which will produce a number of positive and negative centers. Then, it selects only some of the examples to input to SVM according to ISUC algorithm. At the end, it constructs a classifier through SVM learning. Any text can be classified by comparing the distance of clustering centers or by SVM. If the text nears one cluster center of a category and far away from all the cluster centers of other categories, UC can classify it rightly with high possibility, otherwise SVM is employed to decide the category it belongs. The algorithm utilizes the virtues of SVM and unsupervised clustering. The experiment shows that it not only improves training efficiency, but also has good precision.

Keywords support vector machine, clustering, text classification

1 引 言

Internet 网上海量信息使得网页分类成为一个日益重要的研究领域.传统上,网页分类是由人来完成的.即人在分析了网页的内容后,给它一个比较合适的类别.很明显,这需要大量的人力资源.随着网页信息的快速增长,特别是 Internet 上在线信息的增

收到日期:1999-11-17.本课题得到国家自然科学基金(69803010)、国家"八六三"高技术研究发展计划(863-511-946-010)资助.李晓黎,男,1969 年生,博士研究生,主要研究方向为数据采掘、信息检索.刘继敏,男,1967 年生,博士,主要研究方向为图像信息检索、人工智能.史忠植,男,1941 年生,研究员,博士生导师,主要研究方向为人工智能、知识工程.

加,再靠人工的方式来处理是不切实际的.同时,由于分类可以在较大程度上解决目前网上信息杂乱的现象,并方便用户准确地定位所需的信息和分流信息.因此,网页自动分类已成为一项具有较大实用价值的关键技术,是组织和管理数据的有力手段.

目前关于文本分类的文献较多. Apte 用决策树技术来获取分类器[1];Yang 构造了一种近邻算法进行分类[2];Lewis 采用了一个线性分类器[3];Cohen设计了一种建立在权值更新基础上休眠专家算法[4].关于网页自动分类的文献很少, Lin Shian-Hua 通过采掘词语关联来抽取网上文档的分类知识[5],该方法是一种语义方法.

用以上所提及的一些方法对网页(或文本)分类时,首先将网页表示为向量,然后计算向量之间在向量空间中的距离作为分类依据.如文献[2]先用余弦距离计算训练集中每一向量与待分类向量的距离,然后选取 K 个最近距离进行综合分类;而文献[3]先构成类别向量,然后以向量的内积计算待分类向量与类别向量的距离.按照此类方法,那些处在类与类的交界处的属于不同类的向量很容易产生分类错误.

支持向量机(SVM)是一种建立在统计学习理论基础上的机器学习方法[6,7].通过学习算法,SVM 可以自动寻找那些对分类有较好区分能力的支持向量,由此构造出的分类器可以最大化类与类的间隔,因而有较好的推广性能和较高的分类准确率.SVM 已被用于孤立的手写体识别[8]、语音识别、人脸识别[9].但是,对网页分类这样的大规模的数据集而言,训练例子往往很多,SVM 需要的训练时间太长,因而不可接受.一些方法使用启发式规则来简化计算,但必须满足某些限制条件,否则,并不能减少计算复杂度.

无监督聚类(UC)是一种较简单的聚类方法.在给定聚类半径后,通过分别对每类网页进行聚类并获得若干聚类中心.之后,我们可以利用中心来分类:即对任意网页,计算其与各类中心的距离;找到最近的中心后,该中心所对应的类就是网页的所属类.该方法的特点是分类速度快但准确率低.

将 SVM 与 UC 方法结合起来,有可能既保证有快的训练速度,又有较高的分类准确率.这正是本文所要探讨的问题.我们的做法是:在训练阶段,用 UC 方法聚类后,对每一个正的聚类中心,根据中心周围的反例极有可能是支持向量的特点,仅选取部分反例交给 SVM 学习.这样便大大加快了 SVM 的训练速度.在识别阶段,分别计算待识别的网页同正例中心与反例中心的最短距离,若距离差较大,就直接用 UC 分类,否则用 SVM 进行分类.

本文的其余部分组织如下:在第 2 节描述了中文网页的表示之后,第 3 节给出了一种 SVM 与无监督聚类(UC)相结合的网页分类算法,第 4 节提供了试验结果,第 5 节得出结论.

2　中文网页的表示方法

我们先来看一下一般文本的表示.

若所有文本的全部特征总数是 n,则构成一个 n 维的向量空间.其中每一个文本被表示为一个 n 维向量$(w_1, w_2, \cdots, w_n)$.向量在每一维上的分量对应该特征在这篇文本中的权值.在 Salton[10] 提出的文本表示方法中,

$$w_i = \frac{tf_i \times \log(N/n_i)}{\sqrt{\sum_j (tf_j \times \log(N/n_j))^2}}.$$

其中,tf_i 表示该特征在给定文本中出现的次数;N 是训练集中所含文本的总数;n_i 是出现该特征的文本数.该公式是经验公式,但实践表明它是特征表示方法中的一个简单、费用较低的工具,其效果和信息增益(IG)、χ^2-test(CHI)相当,优于其它方法,如互信息(MI)与特征增强方法(TS)[11].

在中文网页表示中,我们先用双向最大匹配法进行自动分词,然后利用数据采掘方法获取汉语的词性规则[12],进行词性标注.只有名词和动词等有实际意义的词才作为特征,这大大减少了特征总数.

毫无疑问,与文本数据不同,网页数据是一种半结构化的数据.在网页表示中,对任一特征而言,有两个因素影响特征的权值.一是词在 HTML 文档中出现的词频,另一个是该词在该文档中出现的位置.

仔细分析 HTML 文件的格式,可以发现其中有一些信息是对分类无益的.如段落标记〈P〉、行中断符〈BR〉、文档类型〈! DOCTYPE〉等等.我们真正关心的是如下的标记:

题头〈TITLE〉,标题〈H1〉,〈H2〉,…,〈H6〉,粗体〈B〉,下划线〈U〉,斜体〈I〉,链接〈A HREF =" ……. HTML ">,〈Meta name = " description " content = " …… ">,〈Meta name =" keywords " content =" …… ">以及〈Meta name =" classification " content =" …… ">.

根据这些标记表示的含义可知:在 TITLE 中的内容是最重要的,它概括和总结了整个网页的内容,

因此在分类中起关键作用.其次,在 H1,…,H6 中的内容是网页的基本组成部分,它具体地阐述了网页的基本构成,H1 到 H6 的重要性依次降低.而 B,U,I 三种格式起强调作用,从一定侧面反映了相关内容.URL 中的关键字告诉用户一些相关信息和链接资源,作用相对小一些.最后,Meta 中的数据也提供了一些有用信息.但由于其格式不规范,而且经常不出现,因而只起借鉴作用.

综上所述,为了精确表示网页的内容,定义标记集

$$S = \{\text{TITLE}, \text{H1}, \text{H2}, \text{H3}, \text{H4}, \text{H5}, \text{H6}, \text{B}, \text{U}, \text{I}, \text{URL}, \text{Meta}\}.$$

权值集 $W = \{W_\Lambda \mid \Lambda \in S\}$,则

$$w_i = \frac{\sum_{\Lambda \in S} (W_\Lambda \times tf_i^\Lambda) \times \log(N/n_i)}{\sqrt{\sum_j (\sum_\Lambda (W_\Lambda \times tf_j^\Lambda) \times \log(N/n_j))^2}},$$

这里 W_Λ 标记 Λ 对应的权值,并且 $W_{\text{TITLE}} > W_{\text{H1}} > W_{\text{H2}} > W_{\text{H3}} > \cdots > W_{\text{Meta}}$.而 tf_i^Λ 为特征 i 在标记 Λ 中出现的频率.

3 SVM 与无监督聚类(UC)相结合的网页分类

在 WEB 网页分类中,为了提高分类精度,每一类的识别被视为一个独立的两类分类问题.假设所有网页为 k 类,记为 $L = \{\alpha_1, \alpha_2, \cdots, \alpha_k\}$.设属于类 α_i 的网页个数为 N_i,我们可以将 k 类的分类问题转化为两类分类问题:对任何一类 α_i 而言,训练正例是该类所包含的全部网页;而反例是在训练集中不属于该类的所有其它类的网页.即 α_i 类的正例总数为 N_i;反例总数为 $\sum_{j=1, j\neq i}^{k} N_j$.由此可见,任何一类中反例数远远大于正例数.若分类的总数 k 与每类所含的元素个数 N_i 较大,则两类分类问题的训练集中反例的比例是很大的.

对给定一个类 $\alpha \in L$,其两类分类问题的训练集 $E = \{(x_1, y_1), (x_2, y_2), \cdots, (x_l, y_l)\}$,其中,$x_i \in \boldsymbol{R}^n$ 为一个网页向量,$y_i \in \{+1, -1\}$.若 $y_i = +1$ 表示 $x_i \in \alpha$,同理 $y_i = -1$ 表示 $x_i \notin \alpha$.

对于任意的测试页 x,问题是如何决定 $x \in \alpha$ 或 $x \notin \alpha$,所做出的决策应有最小的错误概率.显然,要确定 x 究竟属于哪一类,在类别分布等概率的前提下,要进行 $\frac{k+1}{2}$ 次两类分类器的比较.因而,识别效率较低.

SVM 建立在计算学习理论的结构风险最小化原则之上.其主要思想是针对两类分类问题,在高维空间中寻找一个超平面作为两类的分割,以保证最小的分类错误率.

用 SVM 实现分类,首先要从原始空间中抽取特征,将原始空间中的样本映射为高维特征空间中的一个向量,以解决原始空间中线性不可分的问题.

令训练集 $E = \{(z_i, y_i) \mid i = 1, 2, \cdots, l\}$,其中 $z_i \in R^N$,$y_i \in \{-1, +1\}$,我们求 (w, b) 使得 $R(w, b) = \int \frac{1}{2} |f_{w,b}(z) - y| \mathrm{d}\rho(x, y)$ 达到最小.其中 $\rho(x, y)$ 表示特征向量 $\boldsymbol{x}$ 与所属类别的联合分布密度,$f_{w,b}(z) = \operatorname{sgn}[w \cdot z + b]$.为了求出 (w, b),或者说,求出分类器 $f_{w,b}(z)$,导致求解如下的二次优化问题:$\max W(\alpha) = \sum_{i=1}^{l} \alpha_i - \frac{1}{2} \sum \alpha_i \alpha_j y_i y_j (z_i \cdot z_j)$,同时满足 $0 \leq \alpha_i \leq \gamma$ $(i = 1, 2, \cdots, l)$ 与 $\sum_{i=1}^{l} \alpha_i y_i = 0$.

在求出 α 之后,利用关系式 $w = \sum_{i=1}^{l} y_i \alpha_i z_i$,便可以求出 w.同时,必然存在 z_i 使得 $|f_{w,b}(z_i)| = 1$,利用这一等式便可以求出 b.最后,为了判断某个样本 x 是否属于类 α,首先计算 $z = \Phi(x)$,再计算如下决策函数:

$$f(z) = \operatorname{sgn}\Big[\sum_{i=1}^{l} y_i \alpha_i (z \cdot z_i) + b\Big].$$

若 $f(z) = 1$,则 x 就属于类 α,否则 x 就不属于该类.

$z = \Phi(x)$ 可有三种形式,在试验中我们取高斯核函数 $\exp\left(-\frac{\| x - x_i \|^2}{c}\right)$,主要由于它具有明显的统计意义.我们可以看到:用 SVM 求解问题,在训练过程中要解一个二次优化问题,因而时间复杂度较大.

3.1 UC 算法及其分类

我们设计了下述的 UC 算法,其特点是聚类速度较快.在给定聚类半径 r 后,输入训练集合 $Z = \{x_1, x_2, \cdots, x_m\}$,$x_i \in \boldsymbol{R}^n$.下述的 UC 算法可以自动实现对集合 Z 的无监督聚类(该方法本身不管 x_i 是正例还是反例),其描述如下:

Step1. $C_1 \leftarrow \{x_1\}$,$O_1 \leftarrow x_1$,$numCluster \leftarrow 1$,$Z \leftarrow \{x_2, x_3, \cdots, x_m\}$.

Step2. 若 $Z = \varnothing$,则 stop.

Step3. 选择 $x_i \in Z$,从已有中心中寻找与 x_i 最接近的中心 O_j,即

$$O_j \leftarrow \arg \min_{k=1}^{numCluster} d(x_i, O_k).$$

Step4. 若 $d(x_i, O_j) < r$,则将 x_i 加入类 C_j,即 $C_j \leftarrow C_j \cup \{x_i\}$,

调整 C_j 类的中心 $O_j \leftarrow \frac{n_j \times O_j + x_i}{n_j + 1}$，$n_j \leftarrow n_j + 1$，Go to Step6.

Step5. 增加一个新类. $numCluster \leftarrow numCluster + 1$，$C_{numCluster} \leftarrow \{x_i\}$，$O_{numCluster} \leftarrow x_i$.

Step6. $Z \leftarrow Z - \{x_i\}$，go to Step2.

在 UC 算法中，$numCluster$ 表示到目前为止所形成的类数；m 为参加聚类的元素总数；$C_1, C_2, \cdots, C_{numCluster}$ 是结果类；O_j 是类 C_j 的中心；n_j 为 C_j 中的元素个数.

算法从 Step2 到 Step6 为一个循环过程. 对于每个元素 $x_i (i = 2, \cdots, m)$，先寻找离它最近的中心及其它们之间的距离，然后根据该距离的大小把它归入已有类或另建一个新类. 算法在循环中的主要时间耗费在寻找每个元素的最近中心上，即要把每个中心遍历一遍. 所以整个算法所花费的时间应该小于 $numCluster \times m$（在算法结束时，$numCluster$ 即为最终所聚成的全部类数）. 因此该算法具有较高的效率.

为了用 UC 算法解决两类分类问题，首先将类 α 的正例集 Ω^+ 和反例集 Ω^- 分别作为 UC 算法的输入，寻找它们各自的中心. 其中，

$$\Omega^+ = \{x_i \mid (x_i, y_i) \in E,\ y_i = 1\},$$

$$\Omega^- = \{x_i \mid (x_i, y_i) \in E,\ y_i = -1\}.$$

假设 Ω^+ 的中心为 $O_1^+, O_2^+, \cdots, O_u^+$，$\Omega^-$ 的中心为 $O_1^-, O_2^-, \cdots, O_v^-$. 接着计算网页 x 到所有正例中心、反例中心的距离. 并令

$$d_x^+ = \min_{i=1}^{u} d(x, O_i^+), \quad d_x^- = \min_{i=1}^{v} d(x, O_i^-).$$

这里，$d(x, y)$ 是网页 x 与 y 的距离. 最终，我们用如下规则决策：

若 $d_x^+ < d_x^-$，则 $x \in \alpha$，否则 $x \notin \alpha$.

很明显，若聚类半径 r 越大，则聚类总数就越少. 这将导致用 UC 算法分类时在训练阶段和识别阶段的高效率. 但就准确率而言，实验证明 UC 方法要明显低于 SVM.

在图 1 中，假设待识别的向量为 x，距离 x 最近的正例反例中心分别为 O^+, O^-，一个给定的决策阈值为 $\varepsilon (\varepsilon > 0)$，则当 x 在区域 $|d_x^+ - d_x^-| < \varepsilon$ 中时（点划线区域，图中的曲线是双曲线），由于该区域为正反例混杂度较高的区域，因而用 UC 方法对 x 进行分类出错的概率较高，这也是 UC 同 SVM 相比有较低正确率的原因. 相反，若 x 在区域 $|d_x^+ - d_x^-| > \varepsilon$ 中，则进行进一步判断. 若 $d_x^+ < d_x^-$，则 x 离某正例中心 O^+ 较近而离所有的反例中心均较远，所以 $x \in \alpha$，否则 $x \notin \alpha$. 此时，由于 x 距正反例的距离差较大，所以用 UC 进行分类基本可保证正确性.

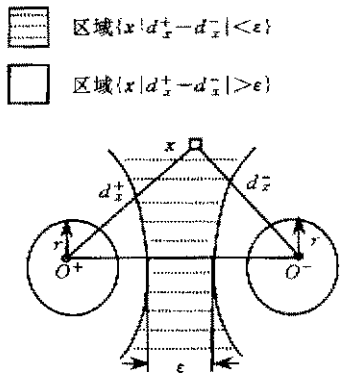

图 1　UC 低准确率的原因

3.2　SVM 与 UC 结合算法（ISUC 算法）

从以上讨论可见，单独使用 SVM 或 UC 并不能以低时间耗费获取高准确率. 而 ISUC 算法则将二者结合起来却有可能达到低的训练代价和高的分类准确性.

在训练阶段，首先给定聚类半径 r 后，用 UC 发现正例集和反例集的中心.

接着挑选部分训练例子交给 SVM 学习. 其原则是 训练集仅挑选全部正例和与正例中心接近的部分反例. 选择这部分反例是由于它们有更高的可能性被选为支持向量，如图 2 所示.

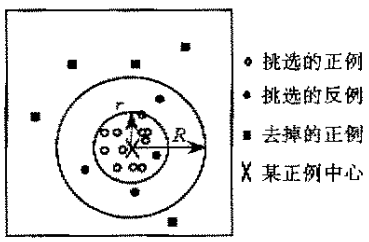

图 2　选择部分训练例子参加 SVM 的学习

严格地说，对于一个给定的切割半径 $R (R > r)$，SVM 的训练集可缩小为

$$\Omega^+ \cup \{x \mid x \in \Omega^- \wedge x \in \bigcup_{i=1}^{k} B_R(O_i^+)\}.$$

其中，$B_R(O_i^+)$ 是以 O_i^+ 为圆心，R 为半径的球.

图 2 仅以一个正例中心为例说明. 设某正例中心为×，则以切割半径 R 为半径的圆中的反例可分为两部分. 一部分是在内圆中（半径为 r），这部分反例尽管很少（由于与正例中心较近），但由于它们与正例混杂，所以极有可能成为支持向量；另一部分在圆环中，这部分的反例相对较多，它们与正例接近程度高. 也有可能成为支持向量. 而在以 R 为半径的圆外，由于它们距正例中心较远，成为支持向量的可能性很小，因此，没有必要将它们交给 SVM 去训练.

这将大大减少训练集的规模和训练时间.

训练阶段的 ISUC 算法可描述为

Step1. $i \leftarrow 1, S_T \leftarrow \varnothing$.

Step2. 若 $i > u$, 则 go to Step6(u 为正例集的聚类中心个数).

Step3. 对于中心 O_i^+,寻找所有满足 $d(x_j, O_i^+) < R$(其中 $j = 1, \cdots, k$)的向量 $x_1, x_2, \cdots, x_k \in \Omega^-$.

Step4. $S_T \leftarrow S_T \cup \{x_1, x_2, \cdots, x_k\}$.

Step5. $i \leftarrow i + 1$, go to Step2.

Step6. 令 $S_T \leftarrow S_T \cup \Omega^+$,将 S_T 交给 SVM 进行训练,最终获得 SVM 分类器.

在测试阶段(或者说识别阶段),对于任意给定的向量 x 与决策阈值 ε,首先计算 d_x^+ 与 d_x^-,然后判断是否 $|d_x^+ - d_x^-| > \varepsilon$(见图 1).若是,则表明 x 离最接近正反例中心的距离差较大,这时我们直接用 UC 方法对 x 分类.具体分两种情况,一种是向量 x 接近某个正例中心而远离所有的反例中心(即满足 $d_x^+ < \min_{i=1}^{v} d(x, O_i^-)$),它表明 $x \in \alpha$);另一种是向量 x 接近懦个反例中心而远离所有的正例中心($x \notin \alpha$).在这两种情况下,用 UC 方法对 x 分类具有相当大的把握.同时由于向量 x 仅与一些聚类中心求距离,所以分类效率较高.否则,由于分类界限模糊,用 UC 方法分类难于抉择.此时我们调用 SVM 作决策,充分利用其在两类边界处具有高区分能力的特点,从而也可获得高的准确率.由以上分析可见,决策阈值 ε 是决定在 ISUC 算法中采用哪种方法进行分类的关键.

假设测试集为 T,则测试阶段的 ISUC 算法可描述如下:

Step1. 若 $T = \varnothing$,则算法结束,否则取 $x \in T$.

Step2. 计算 $d_x^+ \leftarrow \min_{i=1}^{u} d(x, O_i^+)$, $d_x^- \leftarrow \min_{i=1}^{v} d(x, O_i^-)$.

Step3. 若 $|d_x^+ - d_x^-| > \varepsilon$,则 go to Step7.

Step4. 调用SVM分类器 $f(x) = \mathrm{sgn}[\sum_{i=1}^{l} y_i \alpha_i (x \cdot x_i) + b]$ 进行分类.

Step5. 若 $f(x) = 1$,则 $x \in \alpha$, 否则 $x \notin \alpha$.

Step6. go to Step8.

Step7. 若 $d_x^+ < d_x^-$,则 $x \in \alpha$,否则 $x \notin \alpha$.

Step8. $T \leftarrow T - \{x\}$, go to Step1.

4 试验结果

下载了 13548 个中文网页后,我们通过人工方式将其分为 13 类.即政治、军事公安、商业经济、法律条例、农田水利、体育、医疗卫生、工业、科技教育、旅游交通、文化生活、宗教种族、天文地理.仍然有 283 篇文档无法按此分类体系归类.如网页‘蛇岛蝮蛇增多’找不到对应类,而网页‘部队驻地办事处不得办公司’、‘法国组成世界杯赛医疗队’、‘中国科学家揭开乙肝病毒在人肝中持续存在之谜’则可分到两类甚至三类中.

将剩余的 13265 篇文档分成两个集合.其一是训练集,它包含了 9000 篇文档,另一个是测试集,包含 4265 篇文档.然后将各个网页表示成向量.

本试验检查了三种不同的分类技术的性能.表 1 示出了 UC 与 SVM 各自的分类性能.这里我们只给出了体育、政治、经济三种有代表性的领域.原因是:体育类是人工分类中最好分的类,有较少的分类歧义;而政治类包括了较多的子领域,如外交、时事、党政及突发性事件,有时难以和经济、工业、科技教育、农业等其它领域相区别.如国家领导人接见经济、企业等领域的代表团.经济类的情况与政治类相似.

在表 1 中,无论采用 SVM 或者 UC 方法,体育类的正确率均高于其余两类.这说明在向量空间中该类向量与其它类较远,界限较清晰.同时,对三类领域而言 SVM 的准确率均要高于 UC 的分类准确率.这表明 SVM 在处理接近的不同类向量时确有其较精确的区分能力.

表 1 三种领域的 SVM 与 UC 算法的性能比较

方法	体育			政治			经济		
	准确度(%)	正中心数	反中心数	准确度(%)	正中心数	反中心数	准确度(%)	正中心数	反中心数
SVM	98.60			90.97			90.91		
UC($r = 0.3$)	90.97	327	3239	89.32	561	3009	89.48	495	3071
UC($r = 0.7$)	96.23	268	2696	89.53	491	2531	89.53	424	2596
UC($r = 1.0$)	96.83	194	1817	89.48	314	1760	88.68	306	1766
UC($r = 1.2$)	96.51	105	706	87.67	79	483	83.95	88	495
UC($r = 1.4$)	92.82	1	1	74.23	1	1	67.59	1	1

注:对任意两个已正规化的向量 X, Y, $|X - Y|^2 = |X|^2 + |Y|^2 - 2|X||Y| = 2 - 2|X||Y| \leq 2$,故 $r \leq |X - Y| \leq \sqrt{2}$.

对 UC 的聚类半径 r 而言,其大小对识别正确率的影响不是太大,仅当 $r = 1.4$ 时 UC 的正确率有较大下降.正反例的聚类中心数随 r 增加而有较大减少.在我们的分类器中取 $r = 1.2$.此时,正反例中心数分别为 105 及 706 个,可保证有较高的识别效率.

表 2 给出了 ISUC 算法识别时的详细情况.这里针对参数 r, R 不同的取值比较了该算法的性能.我们取决策阈值 $\varepsilon = 0.3$,该值决定了对一个向量究竟应采用 UC 还是 SVM 进行分类.

表 2　ISUC 算法的性能

类别	r	R	# $\text{Nexample}_{\text{cut}}$	# Call_{SVM}	# SV	$\text{Precision}_{\text{SVM}}$(%)	ISUCprecision(%)
体育	0.3	0.5	7820	1553	112	10.03	74.90
	0.3	1.0	6008	1553	678	83.25	95.12
	0.3	1.3	2351	1553	937	97.1	98.81
	1.2	1.25	4450	2561	543	96.72	99.19
	1.2	1.3	2132	2561	651	97.54	98.57
	1.2	1.4	1023	2561	709	97.66	98.62
政治	0.3	0.5	8251	2195	120	21.10	70.09
	0.3	1.0	5422	2195	835	68.22	88.20
	0.3	1.3	2198	2195	1323	83.97	91.86
	1.2	1.25	5321	3094	1425	83.29	91.25
	1.2	1.3	3800	3094	1672	87.10	91.42
	1.2	1.4	1923	3094	1714	87.56	91.42
经济	0.3	0.5	7693	2093	342	26.11	70.51
	0.3	1.0	5017	2093	625	59.68	80.55
	0.3	1.3	1296	2093	1389	82.78	91.80
	1.2	1.25	5982	3290	1322	87.78	91.27
	1.2	1.3	4336	3290	1481	87.97	91.34
	1.2	1.4	1295	3290	1481	87.97	91.34

表 2 说明:对某领域的分类问题,在 r 相同时,调用 SVM 的总数(# Call_{SVM})不变.其原因是聚类中心是由 r 决定的.保持 r 不变,随 R 的增加,被删除的反例数($\text{Nexample}_{\text{cut}}$)减少,从而导致了调用 SVM 的准确率($\text{Precision}_{\text{SVM}}$) 与 ISUC 总准确率(ISUCprecision)均有提高.这主要是由于 SVM 有了更大的训练集,产生了更多的支持向量(# SV).同时它花费了较长的训练时间.

当 $r = 1.2$ 且 $R \in [1.25, 1.4]$ 时,随着 R 的增加,支持向量的个数 # SV 的增加速度减缓,甚至不变.同时 ISUCprecision 变化很小,但超过了单独使用 SVM 时的准确率.对此现象的解释是:SVM 有可能将极少量低维空间可分的向量经映射到高维空间后,变成不可分的向量.

可以注意到 $r = 1.2$ 且 $R = 1.25$ 时, ISUC 的聚类数相对较小同时又删除了较多的反例(平均约 1/3).这将加快 UC 的识别效率和 SVM 的训练速度.更为重要的是其分类正确率很高.

其它领域的实验结果也支持以上的结论.

5　结　论

将 SVM 与 UC 方法结合起来是一种有效的分类方法.它通过减少部分反例,降低了 SVM 的运行时间复杂度,从而部分解决了 SVM 在训练中高耗费的问题.而这一点是将 SVM 用于实践的关键所在.此外,当待分类向量在初始向量空间易于分类时,它应用了 UC 方法在识别过程中的高效性与准确性,否则,它利用 SVM 的较好的区分性能获得高的分类准确性.因此,该方法充分利用了两种方法的优点,既获得了高的训练速度,又加快了识别速度同时保证了较高的识别准确率. 该方法在中文网页分类中获得了较为成功的应用. 作者进一步的工作,一是在网页表示中考虑词的语义信息,即特征间的关系;二是如何在 ISUC 算法中自适应地选择参数,如聚类半径 r、切割半径 R 及决策阈值 ε 的选择.

参　考　文　献

1　Apte C, Damerau F, Weiss S. Automated learning of decision

rules for text categorization. ACM Transactions on Information System, 1994, 12(3): 233 – 251

2 Yang Y. Expert network: Effective and efficient learning from human decisions in text categorization and retrieval. In: Proc Seventeenth International ACM SIGIR Conference on Research and Development in Information Retrieval, Dublin, 1994. 13 – 22

3 Lewis D D, Schapore R E, Callan J P, Papka R. Training algorithms for linear text classifiers. In: Proc Nineteenth International ACM SIGIR Conference on Research and Development in Information Retrieval, Zurich, 1996. 298 – 306

4 Cohen W W, Singer Y. Context-sensitive learning methods for text categorization. In: Proc Nineteenth International ACM SIGIR Conference on Research and Development in Information Retrieval, Zurich, 1996. 307 – 315

5 Lin Shian-Hua. Extracting classification knowledge of internet documents with mining term associations: A sementic approach. In: Proc International ACM SIGIR Conference on Research and Development in Information Retrieval, Melbourne, 1998. 241 – 249

6 Vapnik V. The Nature of Statistical Learning Theory. New York: Springer-Verlag, 1995

7 Vapnik V. Estimation of Dependences Based on Empirical Data. New York: Springer-Verlag, 1982

8 Bernhard Scholkopf, Sung Kah-Kay *et al*. Comparing support vector machines with gaussian kernels to radical basis function classifiers. IEEE Transactions on Signal Processing, 1997, 45(11): 2758 – 2765

9 Edgar Osuna, Robert Freund, Federico Girosi. Training support vector machines: An application to face detection. In: Proc IEEE Conference on Computer Vision and Pattern Recognition, Puerto, 1997. 130 – 136

10 Salton. Introduction to Modern Information Retrieval. New York: McGraw-Hill Book Company, 1983

11 Yang Yi-Ming, Jan O Pederson. A comparative study on feature selection in text categorization. In: Proc 14th International Conference on Machine Learning, Nashville, 1997. 412 – 420

12 Li Xiao-Li, Shi Zhong-Zhi. A data mining method applying to acquire part of speech rules in Chinese text. Computer Research and Development, Accepted(in Chinese)

(李晓黎,史忠植. 用数据采掘方法获取汉语词性规则. 计算机研究与发展,已录用)