ISSN 1673-9418 CODEN JKYTA8
Journal of Frontiers of Computer Science and Technology
1673-9418/2011/05(11)-1037-11
DOI: 10.3778/j.issn.1673-9418.2011.11.009

E-mail: fcst@vip.163.com
http://www.ceaj.org
Tel: +86-10-51616056

多代表点的子空间分类算法*

张健飞, 陈黎飞+, 郭躬德, 李 南
福建师范大学 数学与计算机科学学院, 福州 350007

Multi-Representatives-Based Algorithm for Subspace Classification*

ZHANG Jianfei, CHEN Lifei+, GUO Gongde, LI Nan

School of Mathematics and Computer Science, Fujian Normal University, Fuzhou 350007, China
+ Corresponding author: E-mail: clfei@fjnu.edu.cn

ZHANG Jianfei, CHEN Lifei, GUO Gongde, et al. Multi-representatives-based algorithm for subspace classification. Journal of Frontiers of Computer Science and Technology, 2011, 5(11): 1037-1047.

Abstract: The multi-representatives nearest neighbor classifier, which builds classification model using model clusters centered with representatives, and determines the number of nearest neighbors automatically, has been proposed to overcome the shortcomings of traditional nearest neighbor algorithms. However, it would increase the number of model clusters when the samples in different categories are overlapped, and subsequently the prediction accuracy is affected. This paper proposes a multi-representatives-based algorithm for subspace classification, where the training samples are projected onto some different subspaces in order to construct the classification model consisting of model clusters in individual subspaces. This method makes the overlapped samples belonging to different classes in the entire space easily separable, so that the classification performances can be improved. In comparison with other methods such as traditional *k*NN (*k* nearest neighbor), *k*NNModel, SVM (support vector machine), etc., the experimental results show that the proposed method significantly improves the accuracy of the classification on datasets with complex category structures.

Key words: projection; subspace; representative; classification model

---

*The National Natural Science Foundation of China under Grant No. 61070062 (国家自然科学基金); the Natural Science Foundation of Fujian Province of China under Grant No. 2009J01273 (福建省自然科学基金); the Key Scientific Research Project of the Higher Education Institutions of Fujian Province of China under Grant No. JK2009006 (福建省省属高校科研专项重点项目).
Received 2011-03, Accepted 2011-05.

摘 要：多代表点近邻分类克服了传统近邻分类算法的缺点，使用以代表点为中心的模型簇构造分类模型并自动确定近邻数目。此类算法在不同类别的样本存在大量重叠时将导致模型簇数量增大，造成预测精度下降。提出了一种多代表点的子空间分类算法，将不同类别的训练样本投影到多个不同的子空间，使用子空间模型簇构造分类模型，有效分隔了不同类别样本在全空间中重叠的区域，以提高分类性能。与传统的*k*NN(*k* nearest neighbor)、*k*NNModel、SVM(support vector machine)等分类算法的实验对比结果表明，新方法可以对复杂类别结构数据进行有效分类，且较好地提高了分类精度。

关键词：投影；子空间；多代表点；分类模型

文献标识码：A 中图分类号：TP311

1 引言

分类(classification)技术在数据挖掘的许多领域有着重要的应用，例如影像处理、模式识别、医学诊断和信贷检查等[1]。分类的目的是从训练样本集中学习得出分类模型，用来预测新样本的类别。目前已提出了多种类型的分类算法，如决策树、支撑向量机[2](support vector machine, SVM)等，其中，由 Cover 和 Hart 提出的 *k*-最近邻分类算法(*k* nearest neighbor, *k*NN)[3]是一种已被广泛研究的非参数分类算法，具有简单、易实现、应用范围广的优点。但是，*k*NN[3]是基于实例的分类方法，由于没有显式地构造分类模型，在对新样本进行分类时，需计算新样本与每个训练样本间的相似性，分类速度较慢；另外，*k*NN[3]还存在算法参数 *k*(近邻数目)难以确定的缺点。

基于以上问题，已提出了多种改进的 *k*NN 算法，其中由 Guo 等人提出的 *k*NNModel 算法[4]是一种基于代表点(representative)思想的改进算法。*k*NNModel 为各类样本构造称为模型簇[4]的分类模型，每个模型簇以一个代表点及其所覆盖的特定空间区域的统计信息来表示，因而可以有效地约简数据；代表点是位于覆盖区域中心的一个训练样本，根据覆盖区域内样本属于同一个类别的限制条件，*k*NNModel 算法[4]使用一种贪婪型的搜索算法，根据样本的分布情况自动地确定构造模型簇所需的近邻数目。*k*NNModel 算法[4]的缺点在于没有对代表点集合进行优化，当训练数据中存在复杂的类别结构，比如类别间相互重叠时，将出现模型簇数目剧增或部分样本未被覆盖的情况，降低了分类精度[5]。针对这些缺点，陈黎飞等人提出了多代表点学习算法(multi-representatives for efficient classification, MEC)，该算法以结构化风险最小理论(structural risk minimization)[7]为基础，使用无监督的聚类方法为每个类别构造一个优化的多代表点集合，并自动地确定代表点的数目，进一步提高了基于代表点的分类器的分类性能。

然而，在许多应用领域，数据中的类别存在更为复杂的重叠现象，严重制约了上述方法的有效性。例如，在文本分类上，不同的文档类别经常出现共享某个相似主题的现象，在向量空间模型(vector space model, VSM)中表现为类别之间的重叠[8]。另一方面，文档类别可以用其主题相关的关键字集合来刻画，也就是说，不同的文档类别通常是与不同的特征子空间相关联的，这意味着文档类别间的重叠部分可以跨越不同的子空间。包括 *k*NN[3]、*k*NNModel[4]和 MEC[6]在内的近邻分类算法都是在同一个特征空间(实际上是全空间)为文档类别选择其模型簇的代表点。显然这些方法无法有效区分不同文档类别的类边界，尤其是在类别重叠的区域。

本文在 MEC 算法[6]的基础上，提出了一种基于投影的多代表点学习算法(subspace classification based on multi-representatives, SCM)。与 MEC[6]不同，SCM 在为每个类别构造优化的多代表点集合的同时，使用局部特征加权技术[9]学习得到不同代表点所代表的覆盖区域内同类样本不同的最优投影子

空间; 在对待测样本分类时, 分别将样本投影到不同的子空间, 再根据多代表点模型簇的性质进行分类。直观上, 新方法通过区分类别重叠区域样本关联的不同特征空间来识别类别的边界, 从而提高了多代表点分类的精度。

本文组织结构如下：第 2 章介绍背景知识与相关工作; 第 3 章对新算法进行描述; 第 4 章给出实验环境和实验结果分析; 第 5 章总结全文, 并给出未来的研究方向。

2 背景知识与相关工作

给定 N 个样本组成的训练数据集 $X=\{\boldsymbol{x}_1,\boldsymbol{x}_2,\ldots,\boldsymbol{x}_N\}$, 其中每个样本 $\boldsymbol{x}_i(i=1,2,\ldots,N)$是一个 D 维欧氏空间的向量, 记为 $\boldsymbol{x}_i=<x_{i1},x_{i2},\ldots,x_{iD}>$。$y_i$ 表示 $\boldsymbol{x}_i$ 的类别标号, $y_i=1,2,\ldots,K$, 其中 $K(K>1)$表示训练数据集中包含的类别数目。分类任务的目的是建立任意样本 $\boldsymbol{x}_i$ 与其类别标号 y_i 之间的映射关系, 即分类模型; 再利用分类模型预测新样本 $\boldsymbol{x}_t$ 的类别标号 y_t。为叙述方便, 以下部分用 $X_l(l=1,2,\ldots)$表示 X 的样本子集, $X_1\cup X_2\cup\ldots=X$; 假设数据样本间的相异度用欧几里得距离[10]函数 $dist(\cdot)$来衡量。kNNModel 算法[4]建立的分类模型是如下用三元组表示的模型簇的集合:

$$P_l=(Cls(l),Sim(l),\boldsymbol{v}_l),\quad l=1,2,\ldots,\alpha$$

其中, α 为模型簇的数目; $\boldsymbol{v}_l$ 表示第 l 个模型簇的中心点; $Sim(l)$为该模型簇的覆盖半径; $Cls(l)$为簇的类别标号。直观上, 模型簇 P_l 描述了一个以 $\boldsymbol{v}_l$ 为中心以 $Sim(l)$为半径的空间区域, 在这个区域内所有训练样本具有相同的类别标号 $Cls(l)$。在分类阶段, 首先检查空间区域能够覆盖新样本的模型簇, 再根据这些模型簇的类别标号来预测样本的类别。模型簇的中心和覆盖半径分别用式(1)和式(2)计算：

$$\boldsymbol{v}_l=\frac{1}{Num(l)}\sum_{\boldsymbol{x}_i\in X_l}\boldsymbol{x}_i \tag{1}$$

$$Sim(l)=\begin{cases} dist(\boldsymbol{v}_l,NM_l), \text{ if } dist(\boldsymbol{v}_l,FH_l)>dist(\boldsymbol{v}_l,NM_l) \\ \dfrac{dist(\boldsymbol{v}_l,NM_l)+dist(\boldsymbol{v}_l,FH_l)}{2}, \text{otherwise} \end{cases} \tag{2}$$

这里, $Num(l)$表示模型簇覆盖范围内的(即 P_l 所代表的)样本数目; FH_l 为簇内距中心最远的同类点; NM_l 为距中心最近的异类点。

在 MEC 算法[6]中, 训练样本中的每个类别可以对应于多个模型簇(一个模型簇由一个中心点代表, 因此称为多代表点的分类), 使用 k-means(k 均值)算法[11]对每个类别的训练样本进行部分聚类(partial clustering), 以求取一组优化的中心点集合。k-means 算法[11]的参数 k, 也就是代表点的数目, 根据分类模型的经验风险来确定。MEC 指出[6], 对于一个给定的训练样本集, 代表点数目越多, 意味着分类模型的预测风险越大。依据结构化风险最小理论[7], MEC 选择对应经验风险极小值的模型簇为优化目标。

在许多实际应用数据中, 不同样本类别之间存在复杂的重叠现象, 尤其在簇边界区域。如图 1 所示, 由 X、Y、Z 构成的三维特征空间中, 两类训练样本集(分别用浅灰色、黑色表示)相互重叠, 在全空间中为这个区域的训练样本构造模型簇, 必然导致模型簇数目的大量增加, 因而也增大了分类模型的预测风险。此时, 若将样本分别投影到两个不同的二维子空间$\{X,Y\}$和$\{Y,Z\}$上, 则可以较为容易地为这些重叠的样本构造不同的投影子空间上的模型簇。图 1 中标出的两个大空心圆对应浅灰色样本的子空间模型簇覆盖范围, 这样浅灰色样本就可以用这两个不同子空间$\{X,Y\}$和$\{Y,Z\}$上的两个代表点(深灰色点)及其覆盖半径来表示。本文将在 MEC 算法[6]的基础上, 通过局部加权技术

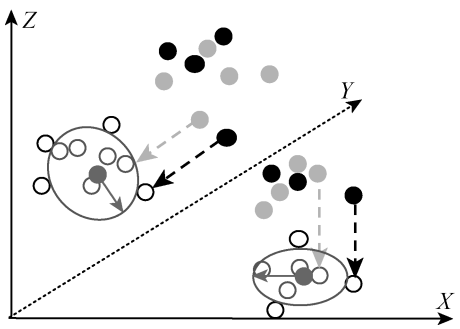


Fig.1 Example of casting subspace model cluster

图 **1** 投影子空间模型簇

学习得到不同类别样本对应的最优投影子空间，进而将样本分别投影到这些子空间上，学习样本的多个代表点的集合，构造出存在于子空间上的分类模型。

3 SCM 分类

首先给出子空间模型簇的形式定义，并描述分类过程；接着通过介绍多代表点和特征权重的学习过程，给出子空间模型簇和分类模型的训练算法，并分析算法的复杂度。

3.1 分类模型

SCM 的分类模型 $M=\{SP_l|l=1,2,\ldots,\alpha\}$，其中，每个 SP_l 称为子空间模型簇。

定义 **1**(子空间模型簇) 子空间模型簇 SP_l 是一个四元组 $SP_l=(Cls(l),SRadius(l),\boldsymbol{v}_l,\boldsymbol{W}_l)$。这里，

$\boldsymbol{v}_l=<v_{l1},v_{l2},\ldots,v_{lD}>$为 SP_l 的中心点(代表点)，由式(1)定义；

$\boldsymbol{W}_l$ 是一个对角矩阵，表示 SP_l 的投影子空间：

$$\boldsymbol{W}_l=\begin{pmatrix} w_{l1} & & & \\ & w_{l2} & & \\ & & \ddots & \\ & & & w_{lD} \end{pmatrix} \tag{3}$$

其中，元素 w_{ld} 为赋予第 $d(d=1,2,\ldots,D)$ 个特征的权重值，满足

$$\forall d=1,2,...,D: w_{ld}\geqslant 0; \sum_{d=1}^{D} w_{ld}=1$$

$SRadius(l)$ 表示 SP_l 在由 W_l 定义的投影子空间中的覆盖半径；

$Cls(l)$ 为子空间中以 $\boldsymbol{v}_l$ 为中心、$SRadius(l)$ 为半径的覆盖范围内，所有样本的类别标号。

数值上，w_{ld} 反映了第 d 个特征相对于类别 $Cls(l)$ 的关联程度，关联程度越高，其数值就越大。在高维数据的投影聚类中[12]，$\boldsymbol{W}_l$ 与一类模糊投影子空间相对应。为检验样本 $\boldsymbol{x}_i$ 和 $\boldsymbol{x}_j$ 在子空间中的相似度，需要首先将样本投影到子空间，使用加权欧氏距离函数[10]来衡量：

$$dist_{\boldsymbol{W}_l}(\boldsymbol{x}_i,\boldsymbol{x}_j)=\sqrt{\sum_{d=1}^{D} w_{ld}(x_{id}-x_{jd})^2} \tag{4}$$

相应地，$SRadius(l)$ 是模型簇 SP_l 在投影子空间 $\boldsymbol{W}_l$ 上的覆盖区域半径，用下式计算：

$$SRadius(l)=\begin{cases} dist_{\boldsymbol{W}_l}(\boldsymbol{v}_l,NM_l), \text{if } dist_{\boldsymbol{W}_l}(\boldsymbol{v}_l,FH_l)>dist_{\boldsymbol{W}_l}(\boldsymbol{v}_l,NM_l) \\ \dfrac{dist_{\boldsymbol{W}_l}(\boldsymbol{v}_l,NM_l)+dist_{\boldsymbol{W}_l}(\boldsymbol{v}_l,FH_l)}{2}, \text{otherwise} \end{cases} \tag{5}$$

这里，类似于 MEC 算法[6]，FH_l 表示距中心 $\boldsymbol{v}_l$ 最远(以式(4)为距离度量函数)的同类点；NM_l 为距中心最近的异类点。

SCM 的分类阶段算法 SCMprediction 的目的是使用给定的分类模型 M 预测待测样本 $\boldsymbol{x}_t$ 的类别 y_t。算法过程如下：

算法 **1** SCMprediction

输入：分类模型 $M=\{SP_l|l=1,2,\ldots,\alpha\}$，类别标号集合 S。

输出：$\boldsymbol{x}_t$ 的类别 y_t。

步骤 **1** 设 $S=\varnothing$。

步骤 **2** 对 $l=1,2,\ldots,\alpha$，用式(4)计算 $\boldsymbol{v}_l$ 与 $\boldsymbol{x}_t$ 的相似度，若 $dist_{\boldsymbol{W}_l}(\boldsymbol{x}_t,\boldsymbol{v}_l)\leqslant SRadius(l)$，则 $S=S\cup\{Cls(l)\}$。

步骤 **3** 若 S 中元素个数 $|S|=1$，输出该类别标号，算法终止；否则，对 $l=1,2,\ldots,\alpha$，找到使得 $dist_{\boldsymbol{W}_l}(\boldsymbol{x}_t,\boldsymbol{v}_l)$ 最小的 $\boldsymbol{v}_l$，输出该 $\boldsymbol{v}_l$ 类别标号。

SCMprediction 首先将 $\boldsymbol{x}_t$ 投影到各个模型簇对应的子空间上，以搜索所有覆盖 $\boldsymbol{x}_t$ 的子空间模型簇。如果这些模型簇的类别标号相同，则将这些模型簇共同的类别标号赋于 $\boldsymbol{x}_t$；否则，根据最近邻原则，把与 $\boldsymbol{x}_t$ 最相似的代表点对应的类别标号赋于 $\boldsymbol{x}_t$。SCMprediction 的时间复杂度为 $O(\alpha)$。

3.2 模型训练

SCM 的模型训练阶段通过局部聚类方法获得每个模型簇的中心(代表点)及对应的最优投影子空间。首先根据不同类别标号将训练样本 X 分为 K 个子集 $X_1,X_2,\ldots,X_K$，分别包含 $N_1,N_2,\ldots,N_K$ 个训练样本，$N_1+N_2+\ldots+N_K=N$；接着，采用无监督的聚类方法从每个子集 $X_k(k=1,2,\ldots,K)$ 中学习得到 α_k 个子空间模型簇。训练过程如图 2 所示。

在软子空间聚类领域，已提出了多种将给定样本集划分成子空间簇的算法[13]，加权 k-means 算法

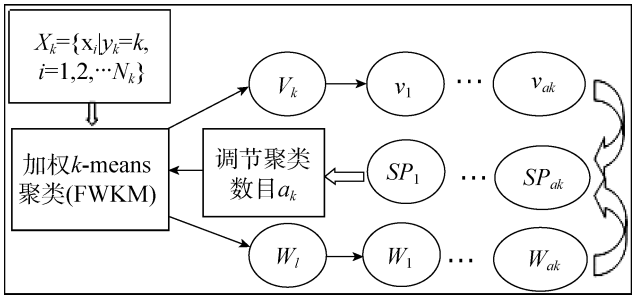


Fig.2 Training of model cluster SP_k

图 2 模型簇 $\boldsymbol{SP_k}$ 训练过程示意图

(feature weighting k-means, FWkM) [14]是其中一种具有代表性的算法, 已广泛应用于文本子空间聚类等领域。由文献[14]给出的 FWkM 是一种加权的 k-means 算法, 算法输入是待划分的样本集及给定的簇数目(对应于图 2 的 α_k), 输出是 α_k 个簇的中心(图 2 用 $\boldsymbol{v}_1,\boldsymbol{v}_2,\ldots,\boldsymbol{v}_{\alpha_k}$ 表示)和其所在的投影子空间(用 $W_1,W_2,\ldots,W_{\alpha_k}$ 表示)。FWkM[14]使用下式计算每个特征的权重：

$$w_{ld}=\left(\sum_{t=1}^{D}\left(\frac{\sum_{i=1}^{n}((\boldsymbol{x}_{id},\boldsymbol{v}_{ld})^2+\delta)}{\sum_{i=1}^{n}((\boldsymbol{x}_{it},\boldsymbol{v}_{lt})^2+\delta)}\right)^{\frac{1}{\beta-1}}\right)^{-1} \tag{6}$$

其中, δ 是为避免分母为 0 而引入的一个很小的数值; β 是用户定义的加权参数。本文根据文献[6]中设定参数的建议, 设定 $\delta=10^{-4}$ 和 β=1.5。

对于一类训练样本 X_k, 使用 FWkM 算法[14]前设定簇数目 α_k, 则学习到的子空间模型簇为 SP_l(l=1,2,…,α_k)。命名算法 2 为 SP-Learning, 根据文献[15]设定控制算法终止的参数为 $\varepsilon=10^{-6}$。

算法 **2** SP-Learning

输入：类别标号为 k(k=1,2,…,K)的训练样本子集 X_k, 聚类后簇的个数 α_k。

输出：子空间模型簇集{SP_l|l=1,2,…,α_k}。

步骤 **1** 随机抽取 α_k 个初始中心点生成 $\boldsymbol{V}_k^{(0)}$($\boldsymbol{V}_k$ 为中心点矩阵), 令所有特征权重为 1/D, 迭代次数 h=0。

步骤 **2** 用式(4)计算各样本与 α_k 个初始中心点的相似度, 并将其划到最相似的中心点, 生成 α_k 个模型簇。

步骤 **3** 用式(1)重新计算各模型簇的中心点 $\boldsymbol{v}_l$ (l=1,2,…,α_k), 更新中心点矩阵为 $\boldsymbol{V}_k^{(h+1)}$。若中心点没有再发生变化(可认为小于某一阈值, 即$\|\boldsymbol{V}_k^{(h+1)}-\boldsymbol{V}_k^{(h)}\|<\varepsilon$), 则算法终止; 否则, h=h+1。

步骤**4** 用式(3)、(6)更新特征权重矩阵[6] $\boldsymbol{W}_l^{(h+1)}$。若$\|\boldsymbol{W}_l^{(h+1)}-\boldsymbol{W}_l^{(h)}\|<\varepsilon$, 则转步骤5; 否则, h=h+1, 转步骤 2。

步骤 **5** 存储 X_k 的多代表点集合 $X_{k\alpha_k}$={$\boldsymbol{v}_l$|l=1,2,…,α_k and $Cls(l)=k$}和特征权重矩阵 $\boldsymbol{W}_l$。

步骤**6** 用式(5)搜索各代表点覆盖区域, 通过定义 1 构造子空间模型簇。

为类似 FWkM[14]这样基于划分的聚类算法估计给定数据集的簇数 α_k 是一个难题[16]。MEC[6]在分析多代表点分类器结构风险的基础上指出, 一个很大的 α_k 取值将导致分类模型预测风险的增加, 而模型的经验风险随 α_k 的增大呈下降趋势。实际上, 分析 3.1 节的分类算法 SCMprediction 可知, 若 α_k=1 则 α=K, 此时 SCMprediction 退化为最近子空间分类[17], 具有最大的经验风险; 另一个极端情况是 α_k=N_k, 此时 α=N, SCMprediction 可以与加权 kNN 分类算法[18]相对应, 经验风险降至最低。因此, 根据结构化风险最小理论[7], 为降低目标分类模型的结构风险, α_k 的取值应在二者之间取得一种平衡。这里采用 MEC[6]提出的策略：选择对应经验风险第一个极小值的 α_k。在 SCM 中, 模型簇集 M_k={SP_1, SP_2,…,SP_{α_k}}的经验风险定义如下：

$$R_{\mathrm{emp}}(M_k)=\frac{1}{N_k}\left(\sum_{\boldsymbol{x}_i\in X_k,y_i=k}I(k\neq SCMprediction(\boldsymbol{x}_i))+\sum_{\boldsymbol{x}_i\notin X_k,y_i\neq k}I(k=SCMprediction(\boldsymbol{x}_i))\right) \tag{7}$$

这里, I(true)=1 和 I(false)=0。直观上, 式(7)通过判断样本 $\boldsymbol{x}_i$ 的真实类别标号 y_i, 并且与使用 SCMprediction 算法得到的类别标号相比较, 统计出错误预测的样本占总样本的比例。

对于一类训练样本 X_k, 用以下 SCMTraining 算法描述构造分类模型 M_k 的过程：

算法 **3** SCMTraining

输入：类别标号为 k 的训练样本子集 X_k。

输出：X_k 的分类模型 $M_k=\{SP_1, SP_2,\ldots,SP_{\alpha_k}\}$。

步骤 **1** 设聚类后簇的个数 $\alpha_k=1$。

步骤 **2** 调用 *SP*-Learning 算法为 X_k 构造模型 M_k, 用式(7)计算 M_k 的经验风险。

步骤 **3** $\alpha_k=\alpha_k+1$。

步骤 **4** 重复步骤 2, 生成模型 M_k', 计算 M_k'的经验风险, 若 $R_{\text{emp}}(M_k')\leqslant R_{\text{emp}}(M_k)$, 则 $M_k=M_k'$, 转步骤 3; 否则算法终止, 输出 M_k。

这里, 对 K 类样本构造分类模型, SCMTraining 需调用 K 次 *SP*-Learning, 算法每次从模型簇数目 1 开始, 通过增加模型簇数目来降低经验风险, 直到经验风险不再下降为止。因此为每类样本建立的分类模型, 是经过训练样本检验的经验风险最小的模型。

SCMTraining 算法构造一个初始模型簇($\alpha_k=1$)的时间复杂度为 $O(N_k)$。设算法终止时模型簇数目为 α_k, 进行了 α_k-1 次聚类, 一次 FWkM 聚类的时间复杂度[14]为 $O(N_k\alpha_k)$, 总的时间复杂度为 $O(N_k\alpha_k^2)$; 算法进行了 α_k-1 次模型簇构造, 时间复杂度为 $O(N_k\alpha_k)$。所以, SCMTraining 的时间复杂度为 $O(N_k\alpha_k^2)$。若给定训练集为 K 个类, 算法将执行 K 次, 则训练过程总的时间复杂度为 $O(KN_k\alpha_k^2)$。

4 实验及结果分析

首先说明数据集及预处理工作; 接着给出实验过程和性能评价标准; 最后实验结果部分对算法有效性和效率进行验证。

4.1 实验环境及实验数据

在配置为 Intel 3.06 GHz CPU、2 GB 内存、120 GB 硬盘, 及操作系统为 Microsoft Windows XP 的计算机上进行实验, 并使用 Java 语言编写的程序实现算法。分别选用 UCI machine learning repository(http://www.ics.uci.edu/~mlearn/databases)数据集和 20-Newsgroups(http://mlg.ucd.ie/files/dataset)文本数据集对算法的性能进行测试。

UCI：“Heart-statlog”是一个涉及实际心脏疾病检测问题的数据集, 该数据集中每个样本包含年龄、性别、血压等 14 个特征属性, 用来预测有无心脏疾病; “SpamBase”是机器学习、模式识别领域中过滤垃圾邮件问题的常用数据集, 该数据集从邮局和个人的垃圾邮件中收集出现频率较高的关键字或特征词汇, 例如像连续大写字母的平均长度和最大长度等词汇特征, 用于判断是否为垃圾邮件。

20-Newsgroups：一个常用的文本数据集。它是由 Ken Lang 收集的来自 20 个不同新闻组的文档。出于效率考虑, 本文只选取一个新闻组(comp.os.ms-windows.misc)的部分样本, 通过特征抽取合成 4 组数据集, 其中每组数据集都包含一个不平衡类, 即其中有一类样本数占总样本数的 10%, 其他各类样本数相等。因此为了更好地对比, 按如下方式给数据集命名：

OS-类别数-不平衡类别数-属性数目

例如 OS-8-1-1000 样本集收集的 4 000 个样本包含的 8 个类分别为 autos、baseball、graphics、mac、med、motor、politics 和 space。其中 autos 为不平衡类, 包含 400 个样本, 其他 7 个类各包含 514 个样本; 数据集中记录了每个样本的 1 000 个特征属性值。更重要的是, 在某个主题下, 这 8 个类中的文档相互间可以共享。例如, 某篇文档以“奥巴马观看棒球比赛”为主题, 那该文档就处于 politics 和 baseball 两个类的重叠区域, 难以确定其类别。因此本文采用 20-Newsgroups 文本数据集, 也能检验各算法在复杂数据集上的分类性能。

数据集的相关信息如表 1 所示。其中, NS 表示数据集中样本个数; ND 表示属性数目; NC 表示类别个数; NCN 表示各类别数据分布。

4.2 预处理

在预处理过程中, 为了减少属性值范围差异对相似度度量的影响, 将数值型数据集的属性值都经过标准化处理变换到[0,1]区间。文本数据集中所有文档用向量空间模型来表示, 并事先将数据变换为单位向量长度。

Table 1 Information about the experimental datasets
表 1 实验数据集的相关信息

DB	NS	ND	NC	NCN
Heart-statlog	270	14	2	45:225
SpamBase	4 601	54	2	1 813:2 788
OS-4-1-500	2 000	500	4	200:600:600:600
OS-5-1-500	2 500	500	5	250:562:562:562:562
OS-7-1-1000	3 500	1000	7	350:525:525:525:525:525:525
OS-8-1-1000	4 000	1000	8	400:514:514:514:514:514:514:514

4.3 实验过程

本文选择 *k*NN[3]、SVM[2]、*k*NNModel[4]以及基于中心点的分类算法[19]作为比较对象。SVM 算法用新西兰 Waikato 大学开发的著名数据挖掘软件 WEKA(Waikato environment for knowledge analysis)中的序贯最小优化(sequential minimal optimization, SMO)来实现；*k*NNModel 算法设置容忍度参数为 0[4]，即构造的模型簇中不包含任何异类点。实验时对每个数据集采用 5-折交叉验证(5-fold cross validation)，即将数据集随机划分为 5 个互不相交的子集，轮流将其中一个子集作为测试集，其他子集为训练集。每个算法都在这 5 对训练集和测试集上运行一遍，并取 5 份结果的算术平均作为最终的结果。

由于 *k*NN 算法的 *k*(近邻数目)难以确定，本实验测试 k=1 和 k=3 两种情况，分别记为 1-NN 和 3-NN。

4.4 性能评估标准

实验采用 *Micro-F*1(微平均)和 *Macro-F*1(宏平均)指标[20]评估分类器的分类精度。假设训练集的样本类别个数为 K，对第 k(k=1,2,…,K)类样本而言，t_k 为被正确分到第 k 类的样本数目，m_k 为实际被分到第 k 类的样本数目，n_k 为第 k 类样本真实的数目，则第 k 类样本分类的 $F1_k$ 值[21]为：

$$F1_k=\frac{2\times recall_k\times precision_k}{recall_k+precision_k}$$

其中，$recall_k$ 和 $precision_k$ 表示第 k 类样本分类的召回率和准确率[22]，计算如下：

$$recall_k=\frac{t_k}{m_k}\qquad precision_k=\frac{t_k}{n_k}$$

*Micro-F*1 先计算所有类别中正确分类和错误分类的样本总数，再求准确率、召回率和 *F*1 值；*Macro-F*1 先计算每个类别的准确率、召回率和 *F*1 值，然后取算术平均值。计算公式如下：

$$Micro\text{-}F1=\frac{2\times\sum_{k=1}^{K}recall_k\times\sum_{k=1}^{K}precision_k}{\sum_{k=1}^{K}recall_k+\sum_{k=1}^{K}precision_k} \tag{8}$$

$$Macro\text{-}F1=\frac{1}{K}\sum_{k=1}^{K}F1_k \tag{9}$$

4.5 实验结果分析

如表 2 和表 3 所示，在 Heart-statlog 数据集上 SCM 算法达到了与对比算法相当的分类精度。在 54 维的 SpamBase 数据集上的分类精度达到了 93%以上，高于 *k*NN 和 SVM 算法，只比基于中心点的分类算法稍逊。以上两组结果也验证了 SCM 算法的有效性。因为 *k*NNModel 在构造模型簇时，采用物理点作为模型簇中心，所以在有类别重叠的 OS 数据集上，*k*NNModel 的模型簇的构造严重受到样本分布的影响。同样 1-NN 和 3-NN 算法是基于近邻点的投票原则来确定其类别，当不同类别样本重叠分布在一起时，用近邻点的类别来投票决定未知样本类别就不准确，其分类精度也如表中所示，出现明显的下降趋势。SVM、基于中心点的算法和 SCM 算法，在 UCI 数据集上的分类精度基本持平。但是随着 OS 数据集的维数从 500 维升高到 1 000 维，重叠类别增加，SCM 的分类精度的优势逐渐明显，特别是在两组 1 000 维数据集上，SCM 算法比 SVM

Table 2 Comparison of the classification accuracies of different algorithms (*Micro-F*1)

表 **2** 不同算法在分类精度上的比较(***Micro-F*1**)

Classifier	UCI 数据集		OS 数据集			
	Heart-statlog	SpamBase	4-1-500	5-1-500	7-1-1000	8-1-1000
SCM	0.872 5	0.932 5	0.935 2	0.929 3	0.885 7	0.879 9
1-NN	0.882 1	0.904 2	0.857 0	0.862 3	0.794 6	0.757 0
3-NN	0.864 2	0.892 2	0.761 5	0.753 3	0.682 5	0.648 1
SVM	0.893 5	0.912 5	0.904 5	0.911 5	0.859 5	0.847 4
*k*NNModel	0.841 3	0.860 5	0.815 5	0.842 9	0.736 4	0.722 3
基于中心点	0.887 3	0.935 3	0.929 5	0.884 1	0.856 2	0.831 9

Table 3 Comparison of the classification accuracies of different algorithms (*Macro-F*1)

表 **3** 不同算法在分类精度上的比较(***Macro-F*1**)

Classifier	UCI 数据集		OS 数据集			
	Heart-statlog	SpamBase	4-1-500	5-1-500	7-1-1000	8-1-1000
SCM	0.866 7	0.931 8	0.934 2	0.928 0	0.881 8	0.873 7
1-NN	0.876 0	0.898 6	0.854 0	0.859 1	0.780 0	0.743 5
3-NN	0.851 6	0.888 9	0.739 3	0.748 9	0.675 2	0.638 3
SVM	0.889 5	0.909 6	0.903 8	0.909 8	0.857 1	0.846 2
*k*NNModel	0.848 1	0.872 5	0.827 5	0.845 0	0.740 6	0.745 6
基于中心点	0.885 2	0.934 0	0.927 4	0.880 0	0.855 2	0.821 4

和基于中心点的算法平均高出 3%以上。可见本文提出的 SCM 算法在稳定性和分类精度上较理想。分析其原因是，在不同子空间上构造的分类模型，能够充分代表原始样本，把高维数据投影到子空间上，维度得到降低，同时类别重叠的样本也可以在不同的子空间上较好地分开。

通过图 3 和图 4，可以直观地比较算法的分类效率。其中纵轴表示 lb *T*, *T* 在图 3 和图 4 中分别代表各算法在训练和测试阶段的执行时间(单位 ms)。因为 *k*NN 是一种懒分类器[3]，没有训练过程，测试阶段需在所有样本中搜索最近邻，所以其分类阶段耗时最高(本文取 1-NN 和 3-NN 算法的平均效率)。*k*NNModel 不仅训练效率低，测试时间开销也随着样本结构复杂程度的增加而急剧上升。SCM 算法因为在训练阶段为样本找到最佳的投影子空间来构造分类模型，寻找多代表点，训练效率较基于中心点的分类算法低，但与 SVM 算法不相上下。因为对未知样本进行分类时，需要在学习到的所有投影子空间上判断其类别，影响了分类效率，所以 SCM 算法的测试效率比 SVM 算法稍差。

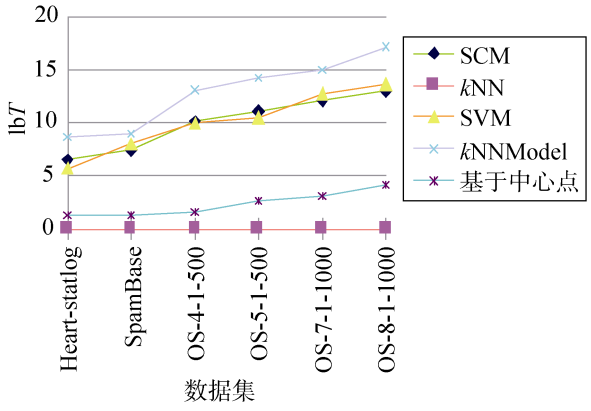


Fig.3 Comparison of training efficiency

图 **3** 训练效率对比

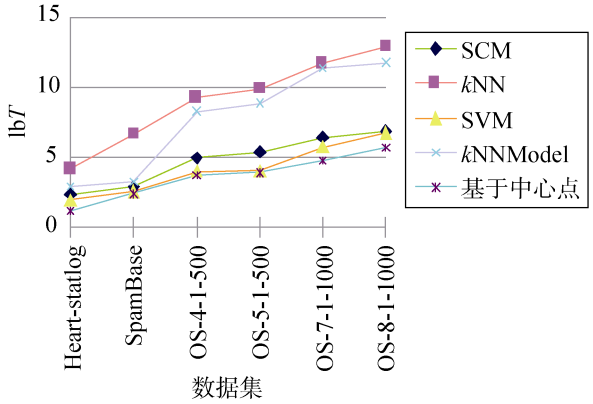


Fig.4 Comparison of testing efficiency

图 4 测试效率对比

5 结论和进一步研究方向

本文针对近邻分类算法的代表点学习问题，提出了 SCM 算法。新算法通过基于局部特征加权的聚类算法，在不同的特征子空间上构造分类模型，优化多代表点集合。与传统的 *k*NN、SVM、*k*NNModel 和基于中心点的分类算法进行实验对比，结果显示新算法能够对类别结构复杂的高维数据进行有效分类。下一步的工作包括探究控制模型簇数目的因素和噪声对模型簇中心点选择的影响等，以进一步提高算法的性能。

References:

[1] Liu Wenjun, Gu Yundong, Li Hongxing. The comprehensive weighed classification algorithm based on rough sets[J]. Fuzzy Systems and Mathematics, 2007, 21(1): 128–136.

[2] Leopold E, Kindermann J. Text categorization with support vector machines: how to represent texts in input space[J]. Machine Learning, 2002, 46(1/3): 423–444.

[3] Cover T M, Hart P E. Nearest neighbor pattern classification[J]. IEEE Transactions on Information Theory, 1967, 13(1): 21–27.

[4] Guo Gongde, Wang Hui, Bell D, et al. *K*NN model-based approach in classification[C]//LNCS 2888: Proceedings of the 2nd International Conference on Ontologies, Database and Application of Semantics (ODBASE '03), Catania, Italy, November 3-7, 2003. Berlin, Germany: Springer-Verlag, 2003: 986–996.

[5] Xin Yi, Guo Gongde, Chen Lifei, et al. Output code algorithm for hierarchical error correcting based on *K*NN-Model[J]. Journal of Computer Applications, 2009, 29(11): 3051–3055.

[6] Chen Lifei, Guo Gongde. A multi-representatives learning algorithm for nearest neighbor classification[J/OL]. Pattern Recognition and Artificial Intelligence. [2010-11]. http://218.241.156.197:81/Jwebprai/CN/article/showNewArticle.do.

[7] Burges C J C. A turotial on support vector machines for pattern recognition[J]. Data Mining and Knowledge Discovery, 1998, 2(2): 121–167.

[8] Bu Fanjun, Qian Xuezhong. *K*-nearest neighbor text categorization algorithm based on vector projection[J]. Computer Engineering and Design, 2009, 30(21): 4939–4941.

[9] Gao Ying, Liu Dayou, Xu Yi. Framework of feature weighted clustering algorithm[J]. Computer Science, 2008, 35(10): 152–154.

[10] Hand D, Mannila H, Smyth P. Principles of data mining[M]. Zhang Yinkui, Liao Li, Song Jun, et al. Beijing: China Machine Press, 2003.

[11] Han J, Kamber M. Data mining: concepts and techniques[M]. Fan Ming, Meng Xiaofeng. 2nd ed. Beijing: China Machine Press, 2007.

[12] Du Yi, Lu Detang, Huang Feng, et al. A random projected clustering algorithm facing high-dimensional categorical data[J]. Mini-Micro Systems, 2006, 27(9): 1605–1607.

[13] Shan Shimin, Yan Yan, Zhang Xianchao. Subspace clustering algorithm based on *k* most similar clustering[J]. Computer Engineering, 2009, 35(14): 4–6.

[14] Jing L, Ng M K, Xu J. An entropy weighting *k*-means algorithm for subspace clustering of high-dimensional sparse data[J]. IEEE Transactions on Knowledge and Data Engineering, 2007, 19(8): 1026–1041.

[15] Chen Lifei, Guo Gongde, Jiang Qingshan. An adaptive algorithm for soft subspace clustering[J]. Journal of Software, 2010, 21(10): 2513–2523.

[16] Liu Yongguo, Zhang Wei, Chen Kefei, et al. An estima-

tion algorithm for number of clusters based on tabu search[J]. Computer Science, 2005, 32(1): 168–171.

[17] Lin Qing, Wang Min. Non-leap subspace clustering on categorical data[J]. Computer Engineering and Design, 2009, 30(6): 1449–1451.

[18] Chen Zhenzhou, Li Lei, Yao Zhengan. Feature-weighted *k*-nearest neighbor algorithm with SVM[J]. Acta Scientiarum Naturalium Universitatis Sunyatseni, 2005, 44(1): 17–20.

[19] Chen Lifei, Ye Yanfang, Jiang Qingshan. A new centroid-based classifier for text categorization[C]//Proceedings of the IEEE 22nd International Conference on Advanced Information Networking and Applications-Workshop (AINA '08), Ginowan, Japan, March 25-28, 2008. Washington, DC, USA: IEEE Computer Society, 2008: 1217–1222.

[20] Jiang Bin, Li Xiang, Wang Hongqiang, et al. Methods of pattern classification[J]. Systems Engineering and Electronics, 2007, 29(1): 99–102.

[21] Sebastiani F. Machine learning in automated text categorization[J]. ACM Computing Surveys, 2002, 34(1): 1–47.

[22] Yang Yiming, Pedersen J O. A comparative study on feature selection in text categorization[C]//Proceedings of the 14th International Conference on Machine Learning (ICML '97). San Francisco: Morgan Kaufmann, 1997: 412–420.

附中文参考文献：

[1] 刘文军, 谷云东, 李洪兴. 基于加权综合的分类算法[J]. 模糊系统与数学, 2007, 21(1): 128–136.

[5] 辛轶, 郭躬德, 陈黎飞, 等. 基于 KNN 模型的层次纠错输出编码算法[J]. 计算机应用, 2009, 29(11): 3051–3055.

[6] 陈黎飞, 郭躬德. 最近邻分类的多代表点学习算法[J/OL]. 模式识别与人工智能. [2010-11]. http://218.241.156.197:81/Jwebprai/CN/article/showNewArticle.do.

[8] 卜凡军, 钱学忠. 基于向量投影的 KNN 文本分类算法[J]. 计算机工程与设计, 2009, 30(21): 4939–4941.

[9] 高滢, 刘大有, 徐益. 一种特征加权的聚类算法框架[J]. 计算机科学, 2008, 35(10): 152–154.

[10] Hand D, Mannila H, Smyth P. 数据挖掘原理[M]. 张银奎, 廖丽, 宋俊, 等译. 北京: 机械工业出版社, 2003.

[11] Han J, Kamber M.数据挖掘:概念与技术[M]. 范明, 孟小峰, 译. 2 版. 北京: 机械工业出版社, 2007.

[12] 杜奕, 卢德唐, 黄丰, 等.一种面向高维符合数据的随机投影聚类算法[J]. 小型微型计算机系统, 2006, 27(9): 1605–1607.

[13] 单世民, 闫妍, 张宪超. 基于 *k* 最相似聚类的子空间聚类算法[J]. 计算机工程, 2009, 35(14): 4–6.

[15] 陈黎飞, 郭躬德, 姜青山. 自适应的软子空间聚类算法[J]. 软件学报, 2010, 21(10): 2513–2523.

[16] 刘勇国, 张伟, 陈克非, 等. 基于禁忌搜索的聚类簇数目估算算法[J]. 计算机科学, 2005, 32(1): 168–171.

[17] 林庆, 王敏. 无重叠子空间分类聚类算法[J]. 计算机工程与设计, 2009, 30(6): 1449–1451.

[18] 陈振洲, 李磊, 姚正安. 基于 SVM 的特征加权 *K*NN 算法[J]. 中山大学学报: 自然科学版, 2005, 44(1): 17–20.

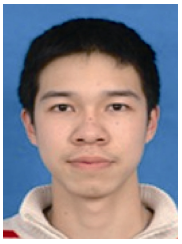
[20] 姜斌, 黎湘, 王宏强, 等. 模式分类方法研究[J]. 系统工程与电子技术, 2007, 29(1): 99–102.

ZHANG Jianfei was born in 1988. He is a master candidate at Fujian Normal University. His research interest is data mining.

张健飞(1988—), 男, 福建师范大学硕士研究生, 主要研究领域为数据挖掘。

CHEN Lifei was born in 1972. He is a doctor and master supervisor at Fujian Normal University. His research interests include data mining and pattern recognition.
陈黎飞(1972—)，男，博士，福建师范大学硕士生导师，主要研究领域为数据挖掘，模式识别。

GUO Gongde was born in 1965. He is a professor and doctoral supervisor at Fujian Normal University. His research interests include artificial intelligence and data mining.
郭躬德(1965—)，男，福建师范大学教授、博士生导师，主要研究领域为人工智能，数据挖掘。



LI Nan was born in 1987. He is a master candidate at Fujian Normal University. His research interest is data mining.
李南(1987—)，男，福建师范大学硕士研究生，主要研究领域为数据挖掘。