第48卷第4期 中国科学技术大学学报 Vol.48,No.4
2018年4月 JOURNAL OF UNIVERSITY OF SCIENCE AND TECHNOLOGY OF CHINA Apr. 2018

文章编号:0253-2778(2018)04-0331-10

基于数据划分的 k-近邻分类加速算法机理分析

宋云胜[1],王 杰[1],梁吉业[1,2]

(1.山西大学计算机与信息技术学院,山西太原 030006;2.计算智能与中文信息处理教育部重点实验室(山西大学),山西太原 030006)

摘要:k-近邻(kNN)分类算法因具有不对数据分布做任何假设、操作简单且泛化性能较强的特点,在人脸识别、文本分类、情感分析等领域被广泛使用.kNN 分类算法不需要训练过程,其简单存储训练实例并根据测试实例与存储的训练实例进行相似度比较来预测分类.由于 kNN 分类算法需要计算测试实例与所有训练实例之间的相似度,故难以高效地处理大规模数据.为此提出将寻找近邻的过程转化为一个优化问题,并给出了原始优化问题与使用数据划分优化问题的最优解下目标函数差异的估计.通过对此估计的理论分析表明,聚类划分可以有效的减小此差异,进而保证基于聚类的 k-近邻分类(DC-kNN)算法具有较强的泛化性能.在公开数据集的实验结果显示,DC-kNN 分类算法在很大程度上为测试实例提供了与原始 kNN 分类算法相同的 k 个近邻进而获得较高的分类精度.

关键词:k-近邻;数据划分;局部信息;实例子集;聚类

中图分类号:TP391 **文献标识码**:A doi:10.3969/j.issn.0253-2778.2018.04.009

引用格式:宋云胜,王杰,梁吉业. 基于数据划分的 k-近邻分类加速算法机理分析[J]. 中国科学技术大学学报,2018,48(4):331-340.

SONG Yunsheng, WANG Jie, LIANG Jiye. Mechanism analysis of the accelerator for k-nearest neighbor algorithm based on data partition[J]. Journal of University of Science and Technology of China, 2018,48(4):331-340.

Mechanism analysis of the accelerator for k-nearest neighbor algorithm based on data partition

SONG Yunsheng[1], WANG Jie[1], LIANG Jiye[1,2]

(1. *School of Computer & Information Technology, Shanxi University, Taiyuan* 030006;
2. *Key Laboratory of Computation Intelligence & Chinese Information Processing (Shanxi University), Ministry of Education, Taiyuan* 030006)

Abstract: Due to its absence of hypotheses for the underlying distributions of data, simple execution and strong generation ability, k-nearest neighbor classification algorithm (kNN) is widely used in face recognition, text classification, emotional analysis and other fields. kNN does not need the training process, but it only stores the training instances until the unlabeled instance appears, and executes the predicted process. However, kNN needs to compute the similarity between the unlabeled instance and all the training instances, hence it is difficult to deal with large-scale data. To overcome this difficulty, the process of computing the nearest neighbors is converted to a constrained optimization problem, and an estimation is given of difference on the value of the objective function under the optimal solution with and

收稿日期:2017-09-19;**修回日期**:2018-04-11

基金项目:国家自然科学基金重点项目(U1435212,61432011),山西省重点科技攻关项目(MQ2014-09)资助.

作者简介:宋云胜,男,1984年生,博士生,研究方向:机器学习.E-mail:sys_sd@126.com

通讯作者:梁吉业,博士/教授. E-mail:ljy@sxu.edu.cn

without data partition. The theoretical analysis of this estimation indicates that data partition using clustering can reduce this difference, and the k-nearest neighbor algorithm based on clustering can have a strong generation ability. Experiment results on public datasets show that the k-nearest neighbor algorithm based on clustering can largely obtain the same nearest neighbors of raw kNN, thus obtaining higher classification accuracy.

Key words: k-nearest neighbor; data partition; local information; instance subset; clustering

0 引言

kNN分类算法是一种惰性学习法,其不需要事先建立分类模型,只是简单地存储训练实例.在给定测试实例时,kNN分类算法首先计算该实例与训练集中所有实例之间的相似度,然后寻找与其相似度最近的前k个训练实例,最后根据这些实例的类别利用多数表决的方式进行预测[1].kNN分类算法因具有充实的理论基础、较强的泛化性能以及对数据分布不做假设等优点,得到广泛应用[2-5]且被国际数据挖掘会议(ICDM)评选为机器学习十大经典算法之一[6].

随着传感与互联网技术的高速发展,来自于各行各业的数据正以指数量级增长,大数据成为政府、学术界以及工业界的关注焦点,大数据分析与挖掘的研究成果也被广泛应用于物联网、健康医疗、电子商务、金融等各个领域.kNN分类算法在预测分类时,因需要计算测试实例与训练集中所有实例之间的相似度,导致其难以有效地处理大数据[7],因而kNN分类算法在大数据环境下的效率面临着巨大的挑战[7].为了提高kNN分类算法的处理大规模数据的效率,大量的改进型算法被提出[8-11].针对kNN分类的加速算法通常可以分为两类:基于数据划分的kNN分类(DP-kNN)算法和基于实例选择的kNN分类(IS-kNN)算法[12].

DP-kNN算法主要是利用分而治之的思想将训练集划分为若干个子集,在预测时测试实例利用与其最相似子集内的实例进行预测分类[13].具体而言,首先将训练集所在特征空间划分为若干个子区域,然后判定测试实例所在划分后的子区域,最后在此区域对应的实例子集内寻找其k个近邻并进行预测.此类算法主要利用了kNN分类算法的局部学习特性:在预测时测试实例的标签仅与其在训练集T中k个最相似的实例相关,因此在对训练集T进行划分时,应尽量保证每个实例在其划分后的子集内的k个近邻与其在原始训练集T中的一致.现有的大多数数据划分算法并未从理论上考虑实例在原始训练集上的近邻与划分后子集上近邻的一致性程度,故难以保证算法具有较强的泛化性能.

与DP-kNN算法不同,IS-kNN算法并没有利用所有的训练实例,而是在训练集的某个代表性子集上寻找测试实例的k个近邻,其中子集的获取是利用利用实例选择算法[14].因其代表性子集的容量比训练集小,故可以大幅度提高测试实例寻找近邻的效率.实例选择是一种重要的数据预处理方法,其根据训练实例在相似度以及标签上的差异,从训练集中移除噪声实例以及远离分类决策面的实例.由于在大多数数据集中远离分类决策面的实例比靠近分类决策面的实例数目多,故实例选择算法可以实现大幅度缩减训练集规模并保持分类精度相对不变的目的;然而现有的大多数实例选择算法的时间复杂度为训练集容量大小的平方,难以有效的处理大规模数据.

针对基于数据划分的kNN分类存在的问题,本文从优化的角度对其性能进行了理论分析.①从理论上分析了数据划分对kNN分类算法性能的影响,并给出了相应数据划分准则;②基于此准则,我们提出利用k-means聚类对数据进行划分,并给出了一种快速寻找近邻的机制;③与现有的两种典型的算法相比,在公开数据集上的实验结果表明,基于聚类的近邻算法在很大程度上可以为测试实例给出与原始kNN分类算法相似的k个近邻,且获得了较高的分类精度.

1 相关工作

DP-kNN算法主要分为3个步骤:首先将当前训练实例所在的特征空间划分为若干各子区域,然后判定测试实例所在区域,最后在落入此区域内的实例子集中寻找近邻序列并预测[14-15].因kNN分类算法是一个局部学习算法,故在对训练集进行数据划分时应保证实例在划分前后的近邻序列一致.现有的大多数针对kNN分类算法的数据划分算法是

基于二叉树结构的，即从原始集合开始，将当前数据递归地划分为容量相近的两个子集直至满足终止条件.Friedman 等[16]首先提出了 KD 树的概念，其利用数据的属性递归的将 k 维特征空间划分为若干子区域，并将落入每个区域内的数据作为一个子集.然而面对高维复杂数据时，会出现某些信息量大的属性在建树的过程中没有使用的现象；针对此问题，Verma 等[17]提出了一种最大化方差的 KD 树(MKD-tree)算法，其选择在当前数据上的属性值方差最大的属性作为节点进行划分.MKD-tree 算法每次对当前数据划分时仅使用某个属性，会造成部分信息损失.为此，基于主成分分析的二叉树(PCA-tree)算法[13]被提出，其根据当前数据在第一主成分上的得分以及对应的中位数值进行划分；此外，还有基于哈希近似的以及基于近邻图结构的数据划分算法[18].然而现有的大多数数据划分算法未从理论上研究数据划分对 kNN 分类算法的影响.

IS-kNN 算法主要是在训练集的某个容量相对较小的代表性子集中寻找测试实例的 k 个近邻并进行预测，其中代表性子集是利用各种实例选择算法获取[19-20].Hart[21]提出了一种基于 1NN 的压缩近邻(CNN)算法，其获得训练集 T 的子集 S 使得集合 $T-S$ 中的实例均被 S 正确地分类，即 $T-S$ 中的实例与其在 S 中的近邻的标签相同.CNN 算法首先从训练集中随机地选择一个实例放入集合 S，然后每次从 $T-S$ 中选取一个实例并判定其与在集合 S 中的近邻的标签是否相同：如果一致则将其放在 S 中，重复上述过程直至集合 $T-S$ 为空集.CNN 算法虽然可以获得容量相对较小的子集 S，但是此算法对数据读取的顺序非常敏感，且时间复杂度为训练集中实例个数的平方.为了克服此困难，Angiulli[11]提出了一种快速的压缩近邻(FCNN)算法.FCNN 算法首先选择最靠近各类中心的实例放入集合 S，然后迭代地从 $T-S$ 中的未被集合 S 正确分类的实例集中选择代表性实例放入集合 S，重复此选择过程直至 S 可以将 $T-S$ 中的所有实例正确地分类.

FCNN 算法不仅与读取数据的顺序无关，而且其时间复杂度为 $O(|T||S|)$，其中 $|T|$ 为集合 T 的大小.为了提升 FCNN 算法处理大规模数据的效率，文献[22]提出了一种基于并行分布式计算的 FCNN 算法.虽然以上算法可以实现大幅度缩减数据规模的目的，但其并未考虑噪声实例的影响.针对此问题，大量的剪辑算法被提出，其主要思想是从集合中除去那些与近邻标签不一致的实例.虽然 CNN 系列算法与剪辑算法均在训练误差保持相对不变的提前下，缩减了训练集的规模.然而这些算法均未考虑训练实例在特征空间的局部稀疏性，故对 kNN 分类算法产生负面影响.为此，Nikolaidis 等[23]提出了一种类边界保留的算法，其首先利用编辑算法除去训练集中的噪声实例，然后利用训练实例在特征空间的潜在分布的几何特性将训练集划分为边界实例子集和非边界实例子集，最后分别从这两个子集中选择代表性实例合并为最终的实例选择子集.此外，还有基于图形的、基于搜索算法等改进型的 kNN 分类算法[24].然而基于实例选择的 kNN 分类算法需要计算所有实例之间的相似度而导致其难以处理大规模数据.

本文针对 kNN 分类算法的特点从优化的角度研究了数据划分对 DP-kNN 算法算法的影响，并给出了相关理论分析.

2　相关概念

令训练集 $T=\{(x_1,y_1),(x_2,y_2),\cdots,(x_N,y_N)\}$ 是由多类的实例组成，其中每个实例 x_i 由 m 维特征向量 $x_i=(x_{i1},x_{i2},\cdots,x_{im})$ 表示，x_{ij} 为实例 x_i 在第 j 个属性上的取值，y_i 为实例 x_i 对应的标签，m，N 分别为属性以及实例的个数，$i,j=1,2,\cdots,N$.

kNN 分类算法是基于类比的学习，通过比较给定测试实例与它相似的训练实例进行预测.当给定一个测试的实例 x，首先计算 x 与训练集 T 中每个实例 x_i 的相似度 d_i，然后将所有的实例按其对应的相似度 d_i 从大到小的顺序排序，最后取前 k 个实例作为 x 的 k 个近邻，并根据 k 个近邻中哪一类的实例个数最多就将 x 判定属于哪一类的原则进行预测分类.

3　基于数据划分 kNN 分类算法的机理分析

kNN 算法是一个局部学习算法，其中测试的预测标签仅与其在训练集中的 k 个近邻的标签相关，因此在进行数据划分时应尽量保证每个训练实例与其在训练集中的 k 个近邻仍在同一个划分后的子集内.对于给定的训练集 T，kNN 分类算法获取所有训练实例在 T 中 k 个近邻可以转化为求解以下

优化问题：

$$\max_{A\in R^{N\times N}} \operatorname{tr}(AD)$$
$$\text{s.t} \quad \sum_{j=1}^{N} A_{ij}=k, A_{ij}\in\{0,1\}, i,j=1,2,\cdots,N \tag{1}$$

式中，$\boldsymbol{A}=[A_{ij}]_{N\times N}\in R^{N\times N}$ 为布尔矩阵，当 $A_{ij}=1$ 时，实例 x_j 是实例 x_i 的 k 个近邻之一，否则不是，$i,j=1,2,\cdots,N$；$\operatorname{tr}(\boldsymbol{AD})$ 为布尔矩阵 $\boldsymbol{A}$ 与相似度矩阵 $\boldsymbol{D}$ 乘积的迹，矩阵 $\boldsymbol{D}$ 的每个元素 D_{ij} 为实例 x_i 与实例 x_j 之间的相似度，.令 $\boldsymbol{A}^*$ 为优化问题(1)的最优解.

假设训练集 T 划分被为 n 个互不相交子集 T_l，其中 $l=1,2,\cdots,n$.基于数据划分的 k 近邻分类算法在每个划分后的子集 T_l 内寻找所属实例的 T_l 个近邻.对于划分后的每个实例子集 T_l，每个实例 $x\in T_l$ 寻找其在 T_l 中的 k 个近邻可以转化为求解以下最优化问题：

$$\max_{A^l\in R^{nl\times nl}} \operatorname{tr}(\boldsymbol{A}^l\boldsymbol{D}^l)$$
$$\text{s.t} \quad \sum_{j=1}^{nl} A_{ij}^l=k, A_{ij}^l\in\{0,1\}, i,j=1,2,\cdots,n_l \tag{2}$$

式中，$\overline{\boldsymbol{A}}^l\in R^{nl\times nl}$ 为布尔矩阵，当 $\overline{A}_{ij}^l=1$ 时，实例 $x_j\in T_l$ 为实例 $x_i\in T_l$ 的某个 k 近邻之一，否则实例 $x_j\in T_l$ 不是，$i,j=1,2,\cdots,n_l$；矩阵 $\boldsymbol{D}^l\in R^{nl\times nl}$ 为矩阵 $\boldsymbol{D}$ 的行列指数均为 V_l 的 n_l 阶子阵，V_l 为集合 T_l 中所有实例下标的集合，n_l 为集合 T_l 的大小，$l=1,2,\cdots,n$.令 $\overline{\boldsymbol{A}}^l$ 为针对实例子集 T_l 求解优化问题(2)得到最优解，其中 $l=1,2\cdots,n$.

令 $\pi(x_i)\in\{1,2,\cdots,n\}$ 为实例 $x_i\in T$ 所属划分子集的下标.当 $\pi(x_i)=l$，则有 $x_i\in T_l$，其中 $i=1,2,\cdots,N$，$l=1,2,\cdots,n$.在本质上，基于数据划分的近邻算法将优化问题(1)近似地分解为个 n 子优化问题(2)，并分别解决这些子优化问题.将每个子问题的最优解组合为新的矩阵 $\overline{\boldsymbol{A}}=[\overline{A}_{ij}]_{N\times N}\in R^{N\times N}$，其对应的每个元素为 $\overline{A}_{ij}=I_{(\pi(x_i),\pi(x_j))}\overline{A}_{ij}^l$，二值函数 $I_{(a,b)}=1$ 当且仅当 $a=b$，否则 $I_{(a,b)}=0$.显然，矩阵 $\overline{\boldsymbol{A}}$ 为优化问题(1)的最优解矩阵 $\boldsymbol{A}^*$ 的一个近似.为了保证算法的性能，应使得 $f(\boldsymbol{A}^*)$ 与 $f(\overline{\boldsymbol{A}})$ 之间的差异最小.为了度量两者之间的差异，我们引入了以下引理和定理.

引理 3.1 $\overline{\boldsymbol{A}}$ 为以下优化问题的最优解

$$\max_{A\in R^{n\times n}} \operatorname{tr}(\boldsymbol{A}\overline{\boldsymbol{D}})$$
$$\text{s.t}\sum_{j=1}^{N} A_{ij}=k, A_{ij}\in\{0,1\}, i=1,2,\cdots,N \tag{3}$$

式中，矩阵 $\overline{\boldsymbol{D}}=[\overline{D}_{ij}]_{N\times N}\in R^{N\times N}$，$\overline{D}_{ij}=I_{(\pi(x_i),\pi(x_j))}D_{ij}$

证明 由矩阵 $\overline{\boldsymbol{D}}$ 的定义可知：

$$\overline{D}_{ij}=\begin{cases} D_{ij}, \pi(x_i)=\pi(x_j) \\ 0, \pi(x_i)\neq\pi(x_j) \end{cases}.$$

式中，$i=1,2,\cdots,N$，$j=1,2,\cdots,N$.此外，由分块矩阵的计算性质可知，$\operatorname{tr}(\boldsymbol{A}\overline{\boldsymbol{D}})=\sum_{l=1}^{n}\operatorname{tr}(\boldsymbol{A}^l\boldsymbol{D}^l)$.因此，优化问题 $\max_{A\in R^{n\times n}}\operatorname{tr}(\boldsymbol{A}\overline{\boldsymbol{D}})$ 可以完全分解为 n 个子优化问题，即

$$\max_{A\in R^{n\times n}}\operatorname{tr}(\boldsymbol{A}\overline{\boldsymbol{D}})=\sum_{l=1}^{n}\max_{A^l\in R^{nl\times nl}}\operatorname{tr}(\boldsymbol{A}^l\boldsymbol{D}^l)$$

因所有子优化问题之间是独立的，且矩阵 $\overline{\boldsymbol{A}}^l$ 是子优化问题 $\max\operatorname{tr}(\boldsymbol{A}^l\boldsymbol{D}^l)$ 的最优解，故矩阵 $\overline{\boldsymbol{A}}$ 是优化问题 $\max_{A\in R^{n\times n}}\operatorname{tr}(\boldsymbol{A}\overline{\boldsymbol{D}})$ 的最优解.

定理 3.1 对于给定的数据集 T 以及其对应的划分标识集合 $\{\pi(x_1),\pi(x_2),\cdots,\pi(x_N)\}$，则有

$$f(\boldsymbol{A}^*)-f(\overline{\boldsymbol{A}})\leqslant\sum_{i=1}^{N}\sum_{j\notin\wedge^i}D_{ji}$$

式中，$f(A)=\operatorname{tr}(AD)$，$\wedge^i$ 为与实例 x_i 不属于同一个划分后子集中所有实例下标的集合.

证明 令函数 $\overline{f}(\boldsymbol{A})=\operatorname{tr}(\boldsymbol{A}\overline{\boldsymbol{D}})$.由矩阵 $\overline{\boldsymbol{A}}$ 的定义可知：

$$\overline{f}(\overline{\boldsymbol{A}})=\sum_{i=1}^{N}\sum_{j\in\wedge^i}\overline{A}_{ij}D_{ji}=$$
$$\Big(\sum_{i=1}^{N}\sum_{j\in\wedge^i}\overline{A}_{ij}D_{ji}+\sum_{i=1}^{N}\sum_{j\notin\wedge^i}\overline{A}_{ij}D_{ji}\Big)-\sum_{i=1}^{N}\sum_{j\notin\wedge^i}\overline{A}_{ij}D_{ji}=$$
$$\sum_{i=1}^{N}\sum_{j=1}^{N}\overline{A}_{ij}D_{ji}-\sum_{i=1}^{N}\sum_{j\notin\wedge^i}\overline{A}_{ij}D_{ji}=$$
$$f(\overline{\boldsymbol{A}})-\sum_{i=1}^{N}\sum_{j\notin\wedge^i}\overline{A}_{ij}D_{ji}$$

$$\overline{f}(\boldsymbol{A}^*)=\sum_{i=1}^{N}\sum_{j\in\wedge^i}A_{ij}^*D_{ji}=$$
$$\Big(\sum_{i=1}^{N}\sum_{j\in\wedge^i}A_{ij}^*D_{ji}+\sum_{i=1}^{N}\sum_{j\notin\wedge^i}A_{ij}^*D_{ji}\Big)-$$
$$\sum_{i=1}^{N}\sum_{j\notin\wedge^i}A_{ij}^*D_{ji}=$$
$$\sum_{i=1}^{N}\sum_{j=1}^{N}A_{ij}^*D_{ji}-\sum_{i=1}^{N}\sum_{j\notin\wedge^i}A_{ij}^*D_{ji}=$$
$$f(\boldsymbol{A}^*)-\sum_{i=1}^{N}\sum_{j\notin\wedge^i}A_{ij}^*D_{ji}$$

综上并结合引理 1,则有

$$f(\boldsymbol{A}^*)-f(\overline{\boldsymbol{A}})\leqslant\sum_{i=1}^{N}\sum_{j\notin\wedge i}\overline{A}_{ij}D_{ji}-\sum_{i=1}^{N}\sum_{j\notin\wedge i}A_{ij}^{*}D_{ji}=\sum_{i=1}^{N}\sum_{j\notin\wedge i}(\overline{A}_{ij}-A_{ij}^{*})D_{ji}\leqslant\sum_{i=1}^{N}\sum_{j\notin\wedge i}D_{ji}$$

由优化问题(1)的定义可知,其最优解 $\boldsymbol{A}^*$ 是使得目标函数 $f(\boldsymbol{A})$ 最大化时的矩阵. 减小目标函数 $f(\boldsymbol{A}^*)$ 与 $f(\overline{\boldsymbol{A}})$ 之间的差异可以促使近似解 $\overline{\boldsymbol{A}}$ 与最优解 $\boldsymbol{A}^*$ 之间的差异变小,因此我们需要最小化其差异的估计 $\sum_{i=1}^{N}\sum_{j\notin\wedge i}D_{ji}$,即应促使不在同一划分子集内实例间的相似度变小.为实现此目标,我们选择了 k-means 聚类算法对训练集 T 进行划分.k-means 聚类是一种空间划分算法,其将训练实例所在的特征空间划分为若干个区域,并将落入每个区域内的实例作为一个"类"[25-26].一方面,k-means 聚类可以促使"类"间实例的相似度小;另一方面,k-means 聚类因均有均匀效应可以将数据划分为若干个容量近似相等的实例子集,而且其时间复杂度为线性的,可以有效地处理大规模数据[27].

4　DC-kNN 分类算法

在给定测试实例 x 以及特征空间划分的条件下,首先计算实例 x 与所有"类"中心的相似度,然后根据相似度大小判定实例 x 属于哪一"类",最后利用此"类"内的是实例以及 kNN 分类算法对 x 进行预测.当测试实例 x 处在所属"类"的边缘时,难以保证其寻找的 k 个近邻与原始的 kNN 分类算法相同.为此,我们将实例 x 处在所属"类"扩展为 λ 个"类"的并集,而这些"类"的类中心为实例 x 在 n 个类中心中的前 λ 个最相似的. λ 的取值越大,预测精度越高但预测时间越长.大量的实验表明,当 $\lambda=2$ 时在很大程度上可以获取与 kNN 分类算法相同的 k 个近邻且预测时间小幅度增加.

综上所述 DC-kNN 分类算法的流程如下所示.

算法 4.1　DC-kNN 分类算法

输入:训练集 $T=\{(x_1,y_1),(x_2,y_2),\cdots\cdots,(x_N,y_N)\}$,子集个数 n,测试实例 x,近邻数 k.

输出:测试实例的预测标签 $\hat{y}$.

Step1　利用 k-means 聚类算法将集合 T 划分为 n 个互不相交的子集 T_l,其中 l=1,2,…,n;

Step2　寻找测试实例 x 与 n 个子集中最相似的 3 个子集 T_x^1,T_x^2,T_x^3;

Step3　测试实例 x 在集合 $T_x=T_x^1\cup T_x^2\cup T_x^3$ 中寻找其 k 个近邻并获取预测标签 $\hat{y}$.

DC-kNN 分类算法分为数据划分阶段与预测阶段.在数据划分阶段,k-means 聚类算法的时间复杂度为 $O(N)$;在预测阶段,DC-kNN 算法计算测试实例与 s 个划分后的"类"中心以及集合 T_x 中所有实例的相似度,此计算过程的时间复杂度为 $O(|T_x|)$,因此,DC-kNN 的时间复杂度为 $O(N)$,可以有效地处理大规模数据.

5　实验分析

5.1　实验设定

为了验证 DC-kNN 分类算法的有效性,我们选择了规模较大的 10 个公开数据集[28-29],并与基于 KD-tree 的 kNN (MKD-kNN) 算法以及基于 PCA-tree 的 kNN(PCA-kNN) 算法作了比较.选择的 10 个数据集的信息如表 1 所示,其中每个数据集的规模均大于 90 000.

表 1　数据集描述

Tab.1　The summary of datasets

datasets	# objects	# attributes	# classes
MiniBooNE	94 158	50	2
Seismic	98 528	50	3
Aloi	108 000	128	1 000
Mnist	350 000	95	2
Webspam	350 000	254	2
Skin-noskin	245 057	3	2
Cifa	60 000	3072	2
Poker	1 025 010	10	10
Acoustic	98 528	50	3
Combined	98 528	100	3

DC-kNN,MKD-kNN 以及 PCA-kNN 分类算法均是 kNN 分类算法的近似,为了评价这些算法对于所有训练实例在数据划分前后近邻序列保持一致的程度,我们计算训练匹配比例 $R_{\mathrm{tr}}=N_{\mathrm{tr}}/N_1$,其中 N_{tr},N_1 分别为数据划分前后的 k 个近邻完全相同的训练实例的个数以及所有训练实例的个数. R_{tr} 取值越大,则算法保持数据的局部性越强;否则,算法保持数据的局部性越弱.测试精度是评价分类器性能的重要指标,其主要表征测试实例的标签是否与预测标签一致,并不能体现近似近邻算法得到的测试实例近邻序列与原始算法得到的是否一致.为此我们还计算了测试匹配比例 $R_{ts}=N_{ts}/N_2$,其中 N_{ts},N_2 分别为在数据划分前后的 k 个近邻与划分

前完全相同的测试实例的个数以及所有测试实例的个数.为了估计算法的性能指标在数据集上的取值,我们采用十折交叉验证的方法.首先将当前数据集随机地划分为10个互不相交且大小近似相等且的子集,然后将其中一个子集作为测试集,剩余的作为训练集,最后将这10次结果的平均作为最终结果.此外,我们采用符号秩和检验[27,30]从整体上评价DC-kNN分类算法与其他算法在性能上是否存在显著性差异,其中原假设为存在显著性差异而备择假设为不存在显着性差异.

以下实验采用欧氏距离来度量实例间的相似度.为了避免不同属性间因量纲带来的影响,所有的属性取值均被归一化至[0,1]区间内.DC-kNN分类算法与待比较算法的性能均受划分后子集大小的影响,故需要在不同划分数目下对其进行比较.根据文献[31]的建议,我们选择 $s=500,1000,2000,5000$ 这4个取值作为划分后的子集大小的限制阈值,进而根据其确定划分后子集的个数.近邻数 k 取值的大小影响数据划分的kNN分类算法的性能:近邻数 k 取值越大,其获取与原始kNN分类算法相同的 k 个近邻的能力越弱.根据文献[32]的建议,我们选择了 $k=7$,显著性水平 $\alpha=0.05$.

5.2 实验结果

我们将从测试精度、训练匹配比以及测试配比这3个指标出发,对3种算法在数据上的性能进行分析与比较.

5.2.1 测试精度

测试精度是评价分类器分类性能的重要指标.为此,在参数 s 的4个不同取值下对这3种算法的测试精度进行比较,并将其实验结果以及相应的统计量罗列在表2~5中.

表2 当 $s=500$ 时3种算法的测试精度

Tab.2 The test accuracy of three algorithms as $s=500$

datasets	DC-kNN	MKD-kNN	PCA-kNN
MiniBooNE	0.880	0.875	0.871
Seismic	0.718	0.720	0.716
Aloi	0.913	0.842	0.838
Mnist	0.979	0.969	0.971
Webspam	0.980	0.976	0.975
Skin-noskin	1.000	0.996	0.998
Cifa	0.720	0.709	0.718
Poker	0.538	0.545	0.535
Acoustic	0.752	0.739	0.740
Combined	0.827	0.815	0.817
Average	0.831	0.819	0.818
Median	0.854	0.829	0.828
P-value		0.025	0.002

续表2

表3 当 $s=1000$ 时3种算法的测试精度

Tab.3 The test accuracy of three algorithms as $s=1000$

datasets	DC-kNN	MKD-kNN	PCA-kNN
MiniBooNE	0.828	0.818	0.811
Seismic	0.717	0.714	0.715
Aloi	0.958	0.852	0.854
Mnist	0.980	0.970	0.974
Webspam	0.980	0.975	0.972
Skin-noskin	0.998	0.995	0.996
Cifa	0.720	0.709	0.718
Poker	0.545	0.545	0.534
Acoustic	0.755	0.742	0.744
Combined	0.828	0.818	0.821
Average	0.831	0.824	0.824
Median	0.828	0.818	0.816
P-value		0.002	0.002

表4 当 $s=2000$ 时3种算法的测试精度

Tab.4 The test accuracy of three algorithms as $s=2000$

datasets	DC-kNN	MKD-kNN	PCA-kNN
MiniBooNE	0.880	0.877	0.874
Seismic	0.715	0.718	0.715
Aloi	0.921	0.865	0.872
Mnist	0.980	0.972	0.975
Webspam	0.980	0.975	0.972
Skin-noskin	1.000	0.996	1.000
Cifa	0.719	0.715	0.714
Poker	0.538	0.542	0.533
Acoustic	0.755	0.742	0.747
Combined	0.828	0.821	0.828
Average	0.832	0.822	0.823
Median	0.854	0.843	0.850
P-value		0.025	0.016

表 5　当 $s=5000$ 时 3 种算法的测试精度

Tab.5　The test accuracy of three algorithms as $s=1000$

datasets	DC-kNN	MKD-kNN	PCA-kNN
MiniBooNE	0.880	0.878	0.876
Seismic	0.717	0.716	0.714
Aloi	0.921	0.876	0.887
Mnist	0.981	0.981	0.976
Webspam	0.980	0.975	0.972
Skin-noskin	1.000	1.000	1.000
Cifa	0.719	0.715	0.714
Poker	0.538	0.539	0.536
Acoustic	0.755	0.745	0.749
Combined	0.828	0.824	0.825
Average	0.832	0.825	0.825
Median	0.854	0.850	0.851
P-value		0.023	0.004

由表 2～5 的统计分析结果可知，检验 P 值均小于给定的显著性水平 0.05，且 DC-kNN 算法的测试精度在所有数据集上的均值与中位数均大于 MKD-kNN 与 PCA-kNN 算法，故 DC-kNN 算法的泛化性能明显的均优于 MKD-kNN 与 PCA-kNN. 特别地，对多分类数据集 Aloi（1000 类），DC-kNN 算法的测试精度明显的高于另外两种算法，因此在参数 s 的不同取值下，DC-kNN 算法的测试精度均获得了最高的预测精度.

5.2.2　近邻匹配比例

基于数据划分的 kNN 算法的分类性能与其获得与原始 kNN 算法相同的近邻的匹配程度呈正相关.算法的近邻匹配程度度愈高，则其预测精度愈高.由上节实验结果可知，DC-kNN 算法获得了最高的分类精度，尤其是对于多分类问题，故 DC-kNN 算法在很大程度上可以获得与原始 kNN 算法相同的近邻.为了验证此结论，我们计算了训练匹配比以及测试匹配比，并将其在参数 s 不同取值下的结果分别罗列在表 6～9 以及表 10～13 中.

表 6　当 $s=500$ 时 3 种算法的训练匹配比

Tab.6　The matching rate about training process of three algorithms as $s=500$

datasets	DC-kNN	MKD-kNN	PCA-kNN
MiniBooNE	0.360	0.126	0.123
Seismic	0.309	0.170	0.200
Aloi	0.650	0.386	0.396
Mnist	0.685	0.321	0.350
Webspam	0.748	0.327	0.354
Skin-noskin	0.946	0.907	0.915
Cifa	0.077	0.005	0.022
Poker	0.128	0.081	0.076
Acoustic	0.362	0.117	0.205
Combined	0.291	0.079	0.135
Average	0.456	0.252	0.278
Median	0.361	0.148	0.203
P-value		0.002	0.002

续表 6

表 7　当 $s=1000$ 时 3 种算法的训练匹配比

Tab.7 The matching rate about training process of three algorithms as $s=1000$

datasets	DC-kNN	MKD-kNN	PCA-kNN
MiniBooNE	0.460	0.240	0.173
Seismic	0.412	0.227	0.136
Aloi	0.616	0.378	0.412
Mnist	0.728	0.335	0.404
Webspam	0.748	0.617	0.444
Skin-noskin	0.985	0.907	0.923
Cifa	0.087	0.005	0.042
Poker	0.130	0.111	0.076
Acoustic	0.454	0.164	0.293
Combined	0.385	0.119	0.206
Average	0.501	0.310	0.311
Median	0.457	0.234	0.250
P-value		0.002	0.002

表 8　当 $s=2000$ 时 3 种算法的训练匹配比

Tab.8　The matching rate about training process of three algorithms as $s=2000$

datasets	DC-kNN	MKD-kNN	PCA-kNN
MiniBooNE	0.577	0.279	0.236
Seismic	0.496	0.300	0.357
Aloi	0.718	0.456	0.543
Mnist	0.810	0.120	0.508
Webspam	0.808	0.657	0.484
Skin-noskin	0.961	0.944	0.942
Cifa	0.244	0.013	0.085
Poker	0.215	0.180	0.118
Acoustic	0.548	0.230	0.395
Combined	0.510	0.167	0.306
Average	0.589	0.335	0.397
Median	0.563	0.255	0.376
P-value		0.002	0.002

表 9 当 $s=5000$ 时 3 种算法的训练匹配比

Tab.9 The matching rate about training process of three algorithms as $s=5000$

datasets	DC-kNN	MKD-kNN	PCA-kNN
MiniBooNE	0.678	0.384	0.303
Seismic	0.616	0.385	0.452
Aloi	0.775	0.519	0.640
Mnist	0.847	0.517	0.614
Webspam	0.888	0.717	0.674
Skin-noskin	0.975	0.961	0.958
Cifa	0.304	0.033	0.105
Poker	0.355	0.421	0.305
Acoustic	0.650	0.298	0.506
Combined	0.624	0.228	0.408
Average	0.671	0.446	0.497
Median	0.664	0.403	0.479
P-value		0.006	0.002

由表 6～9 中的统计结果可知，DC-kNN 算法的训练匹配比在每个数据集上明显的均高于 MKD-kNN 与 PCA-kNN 算法.特别地，当参数 s 取值为 500，1 000 以及 2 000 时，DC-kNN 算法的训练匹配比是另外两种算法的近两倍.此外，秩和检验 P 值均小于给定的显著性水平 0.05，故在训练匹配比上 DC-kNN 算法与另外两种算法存在显著性差异.

表 10 当 $s=500$ 时 3 种算法的测试匹配比

Tab.10 The matching rate about test process of three algorithms as $s=500$

datasets	DC-kNN	MKD-kNN	PCA-kNN
MiniBooNE	0.852	0.130	0.124
Seismic	0.721	0.164	0.199
Aloi	0.914	0.389	0.401
Mnist	0.887	0.321	0.346
Webspam	0.937	0.331	0.350
Skin-noskin	0.998	0.907	0.913
Cifa	0.368	0.006	0.023
Poker	0.536	0.088	0.076
Acoustic	0.771	0.113	0.200
Combined	0.725	0.076	0.135
Average	0.771	0.252	0.277
Median	0.812	0.147	0.200
P-value		0.002	0.002

表 11 当 $s=1000$ 时 3 种算法的测试匹配比

Tab.11 The matching rate about test process of three algorithms as $s=1000$

datasets	DC-kNN	MKD-kNN	PCA-kNN
MiniBooNE	0.918	0.240	0.172
Seismic	0.818	0.227	0.136
Aloi	0.947	0.378	0.412
Mnist	0.917	0.335	0.401
Webspam	0.937	0.617	0.440
Skin-noskin	0.995	0.907	0.924
Cifa	0.398	0.005	0.043
Poker	0.538	0.111	0.075
Acoustic	0.857	0.164	0.290
Combined	0.835	0.119	0.206
Average	0.816	0.310	0.310
Median	0.887	0.234	0.248
P-value		0.002	0.002

表 12 当 $s=2000$ 时 3 种算法的测试匹配比

Tab.12 The matching rate about test process of three algorithms as $s=2000$

datasets	DC-kNN	MKD-kNN	PCA-kNN
MiniBooNE	0.969	0.231	0.280
Seismic	0.897	0.354	0.294
Aloi	0.971	0.547	0.456
Mnist	0.953	0.505	0.452
Webspam	0.957	0.480	0.651
Skin-noskin	0.999	0.941	0.944
Cifa	0.781	0.088	0.013
Poker	0.600	0.116	0.180
Acoustic	0.914	0.388	0.227
Combined	0.913	0.302	0.161
Average	0.895	0.395	0.366
Median	0.934	0.371	0.287
P-value		0.002	0.002

表 13　当 $s=5000$ 时 3 种算法的测试匹配比

Tab.13　The matching rate about test process of three algorithms as $s=5000$

datasets	DC-kNN	MKD-kNN	PCA-kNN
MiniBooNE	0.991	0.385	0.300
Seismic	0.981	0.379	0.450
Aloi	0.987	0.517	0.647
Mnist	0.962	0.512	0.614
Webspam	0.997	0.711	0.670
Skin-noskin	0.999	0.960	0.958
Cifa	0.851	0.033	0.108
Poker	0.745	0.418	0.305
Acoustic	0.966	0.297	0.498
Combined	0.961	0.223	0.398
Average	0.944	0.444	0.495
Median	0.973	0.401	0.474
P-value		0.002	0.002

由表 10-13 可知，在参数 s 的不同取下，DC-kNN 分类算法的测试匹配比在所有数据集上的均值以及中位数均超过了 0.77，而其他的两种算法的测试匹配比的中位数分别为 0.446，0.497.具体而言，当参数 s 取定 500，1 000 或者 2 000 时，DC-kNN 算法的测试匹配比大约是另外两种算法的两倍；当 $s=5000$ 时，DC-kNN 算法的测试匹配比大约是另外两种算法的一倍多.上述实验结果表明，DC-kNN 可以在很大程度上保证测试实例在划分后的子集内的 k 个近邻与原始的 kNN 分类一致，故其测试精度高于另外两种算法.

综上所述，本文将寻找近邻的过程转化为一个有约束的优化问题，并给出了原始优化问题与使用数据划分优化问题的最优解下目标函数差异的估计.聚类算法可以大幅度的缩减小此差异的估计，换言之，利用聚类划分之后得到的近邻在很大程度上与原始 kNN 算法保持一致.结合 5.2.1 与 5.2.2 的实验结果可知，与 MKD-tree、PCA-tree 算法相比，DC-kNN 算法在很大程度上获取了与原始 kNN 分类算法相同的 k 个近邻，而且其获得了较高的分类精度，故实验结论验证了局部数据集 k 近邻分类结果与原始 k-近邻分类结果误差尽可能小（匹配度更高）的算法，在很大程度得出更好的精度.

6　结论

本文从优化的角度研究了数据划分对 kNN 分类加速算法的影响，从理论上证明了数据划分时使得不在同一划分子集内的实例相似度较小是保证 kNN 加速分类算法泛化性能的关键.基于该理论，我们利用 k-means 聚类算法来实现划分进而鲁棒地实施 kNN 加速分类算法.在多个真实数据集上的实验结果表明，与其他基于数据划分的算法近邻算法相比，基于 k-means 聚类的 kNN 分类算法在很大程度上可以获得与原始 kNN 分类算法相同的 k 个近邻进而得到较高的分类精度；同时该算法的时间复杂度为线性的，所以可以有效地处理大规模数据.

参考文献（References）

[1] COVER T, HART P. Nearest neighbor pattern classification[J]. IEEE Transactions on Information Theory, 2002, 13(1): 21-27.

[2] XU B H, FU Y W, JIANG Y G, et al. Heterogeneous knowledge transfer in video emotion recognition, attribution and summarization[J]. IEEE Transactions on Affective Computing, 2018, 9(2): 255-270.

[3] 周志华. 机器学习[M]. 北京：清华大学出版社，2016.

[4] 张敏灵. 一种新型多标记懒惰学习算法[J]. 计算机研究与发展，2012，49(11)：2271-2282.

[5] FRIEDMAN J, HASTIE T, TIBSHIRANI R. The Elements of Statistical Learning [M]. Berlin: Springer, 2001.

[6] WU X, KUMAR V, QUINLAN J R, et al. Top 10 algorithms in data mining [J]. Knowledge and Information Systems, 2008, 14(1): 1-37.

[7] KONONENKO I, KUKAR M. Machine Learning and Data Mining: Introduction to Principles and Algorithms [M]. Chichester: Harwood Publishing Limited, 2007.

[8] LI Y, MAGUIRE L. Selecting critical patterns based on local geometrical and statistical information [J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2011, 33(6): 1189-1201.

[9] MUJA M, LOWE D G. Scalable nearest neighbor algorithms for high dimensional data [J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2014, 36(11): 2227-2240.

[10] MARCHIORI E. Class conditional nearest neighbor for large margin instance selection[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2010,

32(2): 364-370.

[11] ANGIULLI F. Fast nearest neighbor condensation for large data sets classification[J]. IEEE Transactions on Knowledge and Data Engineering, 2007, 19(11): 1450-1464.

[12] LIU T, MOORE A W, YANG K, et al. An investigation of practical approximate nearest neighbor algorithms [C]// Proc of Conference on Neural Information Processing Systems. Vancouver, Canada: MIT Press, 2005: 825-832.

[13] MCFEE B, LANCKRIET G R G. Large-scale music similarity search with spatial trees[C]// Proceedings of the 12th International Society for Music Information Retrieval Conference. Florida: ISMIR Press, 2014: 566-574.

[14] GARCIA S, DERRAC J, CANO J, et al. Prototype selection for nearest neighbor classification: Taxonomy and empirical study[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2012, 34(3): 417-435.

[15] HSIEH C J, SI S, DHILLON I S. A divide-and-conquer solver for kernel support vector machines [C]// Proceedings of the 27th International Conference on Machine Learning. Haifa: IMLS Press, 2014: 566-574.

[16] FRIEDMAN J H, BENTLEY J L, FINKEL R A. An algorithm for finding best matches in logarithmic expected time[J]. ACM Transactions on Mathematical Software, 1977, 3(3): 209-226.

[17] VERMA N, KPOTUFE S, DASGUPTA S. Which spatial partition trees are adaptive to intrinsic dimension? [C]// Proceedings of the 25h Conference on Uncertainty in Artificial Intelligence. Montreal, Canada: AUAI Press, 2009: 565-574.

[18] SLANEY M, CASEY M. Locality-sensitive hashing for finding nearest neighbors [J]. IEEE Signal Processing Magazine, 2008, 25(2): 128-131.

[19] OLVERA-LÓPEZ J A, CARRASCO-OCHOA J A, MARTÍNEZ-TRINIDAD J F, et al. A review of instance selection methods[J]. Artificial Intelligence Review, 2010, 34(2): 133-143.

[20] BRIGHTON H, MELLISH C. Advances in instance selection for instance-based learning algorithms[J]. Data Mining and Knowledge Discovery, 2002, 6(2): 153-172.

[21] HART P. The condensed nearest neighbor rule[J]. IEEE Transactions on Information Theory, 1968, 14(3): 515-516.

[22] ANGIULLI F, FOLINO G. Distributed nearest neighbor-based condensation of very large data sets[J]. IEEE Transactions on Knowledge and Data Engineering, 2007, 19(12): 1593-1606.

[23] NIKOLAIDIS K, GOULERMAS J Y, WU Q H. A class boundary preserving algorithm for data condensation[J]. Pattern Recognition, 2011, 44(3): 704-715.

[24] ZHANG H, SUN G. Optimal reference subset selection for nearest neighbor classification by tabu search [J]. Pattern Recognition, 2002, 35(7): 1481-1490.

[25] 王熙照, 王亚东, 湛燕, 等. 学习特征权值对 K-均值聚类算法的优化[J]. 计算机研究与发展, 2003, 40(6): 869-873.

[26] 杨润玲, 高新波. 基于加权模糊 c 均值聚类的快速图像自动分割算法[J]. 中国图象图形学报, 2007, 12(12): 2105-2112.

[27] DEMŠAR J. Statistical comparisons of classifiers over multiple data sets[J]. Journal of Machine Learning Research, 2006, 7(1): 1-30.

[28] CHANG C C, LIN C J. LIBSVM: A library for support vector machines[J]. ACM Transactions on Intelligent Systems and Technology, 2011, 2(3): No.27.

[29] BLAKE C L, MERZ C J. UCI Repository of machine learning databases[EB/OL]. [2017-06-15] http://www.ics.uci.edu/~mlearn/MLRepository.Html, 1998.

[30] WILCOXON F. Individual comparisons by ranking methods[J]. Biometrics Bulletin, 1945, 1(6): 80-83.

[31] GARCÍA-OSORIO C, HARO-GARCÍA A, GARCÍA-PEDRAJAS N. Democratic instance selection: A linear complexity instance selection algorithm based on classifier ensemble concepts[J]. Artificial Intelligence, 2010, 174(5): 410-441.

[32] KORDOS M, BLACHNIK M, STRZEMPA D. Do we need whatever more than k-NN? [C]// Proceedings of the 10th International Conference on Artificial Intelligence and Soft Computing. Zakopane, Poland: Springer, 2010: 414-421.