ISSN 1004-9037,CODEN SCYCE4
Journal of Data Acquisition and Processing Vol. 32,No. 6,Nov. 2017,pp. 1153－1162
DOI:10.16337/j.1004-9037.2017.06.010

http://sjcj.nuaa.edu.cn
E-mail:sjcj@nuaa.edu.cn
Tel/Fax:＋86-025-84892742

基于分层抽样的 k 近邻分类加速算法

宋云胜[1]　梁吉业[1,2]

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

摘　要:k 近邻(k nearest neighbor, kNN)分类作为数据挖掘中最典型的算法之一,以较高的泛化性能以及充足的理论基础被广泛应用。然而 kNN 在测试时需要计算待识别实例与所有训练实例之间的距离,以至于在面对大规模数据时需要大量的时间。为此,提出一种基于分层抽样的 kNN 加速算法(KNN based on stratified sampling,SS-kNN)。首先将训练实例所在的空间划分为若干个实例个数相等的区域,然后从每个区域内抽取实例,最后判定待识别实例落入划分区域中的哪一个,并从此区域以及相邻区域抽取的实例中寻找其 k 个近邻。与原始 kNN 算法以及基于随机抽样的 kNN 算法相比,SS-kNN 算法可以获得与其相近分类精度,但将其运行速度分别提高大约 399 倍和 16 倍。

关键词:分层抽样;数据划分;近邻;分类精度;运行时间

中图分类号:TP181　　**文献标志码**:A

Acceleration Algorithm for k Nearest Neighbor Classification Based on Stratified Sampling

Song Yunsheng[1], Liang Jiye[1,2]

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

Abstract: k nearest neighbor (kNN), which is one of the most typical data mining algorithms, is widely applied in various areas due to its better generation ability and sufficient theory results. The method needs to compute the distances between the test instances and all the training instances during executing prediction. However, it costs substantial time as facing the large-scale data. To solve the problem, we propose an acceleration algorithm for k nearest neighbor classification based on stratified sampling (SS-kNN). In the method, SS-kNN firstly divides the instance space into several subranges with the same number of instances, and then samples instances from each subrange, finally judges which subrange the test instance sit and finds its nearest neighbors from this subrange. Compared with kNN and its variant based on the random sampling, SS-kNN could not only obtain the similar classification accuracy, but also accelerates the running time by an average of 399 and 16 times respectively.

Key words: stratified sampling; data partition; nearest neighbor; classification accuracy; running time

---

基金项目:国家自然科学基金(61432011, U1435212)重点资助项目;山西省煤炭重点科技攻关计划(MQ2014-09)资助项目。
收稿日期:2015-06-05;**修订日期**:2016-06-30

引 言

现代信息化时代,各行各业数据不断的涌现导致数据量呈井喷式增长,其量级不再以 GB 或者 TB 来衡量而是以 PB 或者 ZB 级为单位。大数据的出现开始为人类的生活提供了重要的信息和社会财富。世界零售业巨头沃尔玛利用客户购买商品的信息发现啤酒和尿布放在一起可以大大增加销售量;著名的网络公司 Google 通对网络用户搜索的关键词加以分析,在 H1N1 爆发的几周前成功地预测了冬季流感的传播;Farecast 系统用了将近十万亿的美国国内机票价格记录建立预测模型,其预测准确率高达 75%,平均每张机票可节省 50 美元[1-5]。大数据的出现一方面带来了巨大的便利,另一方面对如何有效地利用大数据提出了更大的挑战。

kNN 分类算法是数据挖掘中一个重要的学习算法,其因操作简单且拥有充足的理论基础被广泛地应用,并被国际数据挖掘大会(IEEE international conference on data mining, ICDM)评选为机器学习中十大经典算法之一[6-8]。与其他的分类学习算法不同,kNN 不需要训练过程且不对数据的分布作任何的假设,而是直接从数据本身出发。在给定待检验的实例后,kNN 根据其与存储的训练实例的相似性对其进行预测分类。

虽然 kNN 分类算法不需要训练过程,但是在进行预测分类时需要计算待识别实例与所有训练时间之间的距离,从而导致预测试时间较长[9-10]。为了在保持分类精度相对不变的前提下大幅度削减预测开销,大量的基于空间几何结构的数据压缩算法被提出,其主要利用训练集 T 中实例之间的距离来选择代表性的实例子集 S。与原始的 kNN 分类算法相比,经过数据压缩之后的 kNN 分类算法只需计算待识别实例与 S 中的实例距离,故其预测时间以及存储空间大大缩减。现有的针对 kNN 分类的数据压缩算法主要是寻找训练集 T 中靠近类边界的实例[11-14],其往往需要计算所有训练实例之间的距离才可以判定哪些实例是靠近分类边界的实例,然而在面对大规模数据时是难以实现的[15-17]。

概率抽样作为数据预处理的一种重要的方式被广泛地应用,其在训练集 T 中每个实例被赋予一个抽样概率的基础上抽取一定数量的实例[18]。与基于几何结构数据压缩算法不同,概率抽样并未利用训练实例的结构信息,只需要事先定义每个实例被抽中的概率。概率抽样主要有顺序性抽样、简单随机抽样、分层抽样和适应性抽样等方法。虽然简单随机抽样可以简单地执行,但是并不能保证抽取实例在集合 S 中的顺序与其在 T 中的顺序一致。在关联规则学习中,当数据量较大时难以在有效的时间内实现 Aprior 算法。为此,2002 年 Chen 等[19]提出了一种两阶段抽样方法(Two step sampling, TSS)来获取原始数据 T 的子集 S,然后在 S 上运行关联规则算法。TSS 方法首先从 T 中随机地抽取容量相对较大的实例子集 T_1 并估计每个项的支持度,然后从 T_1 中逐个删除那些对频繁项支持度的大小没有影响的实例,并得到最终的实例子集 S。面对大规模数据如何从有限个数的分类模型集合 H 中选择一个最优的模型十分困难。针对此问题,Domingo 等[20]提出了一种适应性抽样(Adaptive sampling, AS)方法来获取接近最优的分类模型 h_0,$U(h_0) \geqslant (1-\varepsilon)U(h_*)$,其中函数 $U(h)$ 为度量模型 h 的性能,h_* 为最优的模型,ε 为取值于(0,1)的实数。AS 不断地从 T 中随机抽取实例直至存在某个模型 $h_0 \in H$ 满足某个终止条件。该方法不仅避免了事先确定抽样样本量,并且从理论上证明了其以 $1-\delta$ 的概率输出接近最优的分类模型 h_0。Byrd 等[21]针对在梯度下降法中每次迭代需要利用全部的训练实例计算梯度的问题,提出了利用简单随机抽样抽取一定数量的训练实例的方式来估计方向梯度的方法,并给了抽取实例规模大小的计算公式。Kleiner 等[22]针对 Boostrap 抽样难以处理大规模数据的问题,利用分而治之的策略提出了一种多重 Boostrap 抽样的方法,首先从原始数据中无放回的抽取样定量的多个样本集,然后在每个样本集上运行 Boostrap 抽样,最后综合所有样本集上的结果。此方法不仅大幅度地缩减了存储空间以及运行时间,而且理论上证明此方法的渐进收敛性。此外还有其他的抽样方法实现数据压缩[23-25]。

现有的概率抽样方法适用于总体只有各个实例的名录并再无其他信息的情况,但在实际中当给定

训练集 T 之后便可以粗略地获取其空间结构信息。由于 kNN 是一个局部学习算法,待识别的实例 x 仅与其距离最近的 k 个实例相关且均在输入空间中的同一区域内。为此,本文提出了一种基于分层抽样的 kNN 分类加速算法,不仅加速了 kNN 分类过程,而且保持了分类精度相对不变。该算法首先利用训练实例的空间几何结构信息将其划分为实例个数相等的域,然后快速地识别待识别实例所在的区域,最后在此区域及相邻区域抽取的实例中寻找近邻。实现表明,与传统的基于简单随机抽样的 kNN 算法(KNN based random sampling, RS-kNN)相比,SS-kNN 算法获得了与其相近的分类精度但运行时间大幅度地缩短。

1 相关概念

令训练集 $T=\{(\boldsymbol{x}_1,\boldsymbol{y}_1),(\boldsymbol{x}_2,\boldsymbol{y}_2),\cdots,(\boldsymbol{x}_N,\boldsymbol{y}_N)\}$,其中每个实例 $\boldsymbol{x}_i$ 由 m 维特征向量 $(\boldsymbol{x}_{i1},\boldsymbol{x}_{i2},\cdots,\boldsymbol{x}_{im})^{\mathrm{T}}$ 表示,$\boldsymbol{x}_{ij}$ 为实例 $\boldsymbol{x}_i$ 在第 j 个属性上的取值,$\boldsymbol{y}_i$ 为实例对应的标签,m,N 分别为属性以及实例的个数。

kNN 分类算法是基于类比的学习,其通过比较给定待识别实例和与它相似的训练实例学习。当给定一个待识别的实例 $\boldsymbol{x}$,kNN 首先计算 $\boldsymbol{x}$ 与训练集 T 中的实例 $\boldsymbol{x}_i$ 的距离(或者相似度) d_i,然后将距离 d_i 从小到大的顺序排序并取前 k 个距离对应的实例作为 $\boldsymbol{x}$ 的 k 个近邻,最后这 k 个近邻中哪一类的实例个数最多就将 $\boldsymbol{x}$ 判定属于那一类。

SS-kNN 算法主要分为 3 个步骤:实例空间的划分、实例的抽取以及待识别实例的预测。本节将从这 3 个步骤讨论 SS-kNN 算法。

1.1 空间的划分

由于 kNN 是一个局部学习算法,其主要利用实例在输入空间中与其距离最近的 k 个实例来学习。因此在对训练实例所在的空间 S 进行划分时,必须注意以下 3 个条件:

(1) 局部性。待识别的实例 $\boldsymbol{x}$ 仅与在训练集 T 中的 k 个近邻有关,而并非所有的实例。故在将 S 划分时,应尽量保证处在 S 中同一区域内的实例仍在划分后的同一区域内。换言之,尽量促使训练集 T 中的实例 $\boldsymbol{x}_i$ 及其 k 个近邻在划分后的同一区域内。

(2) 易操作性。kNN 在预测时,其时间复杂度为 $O(sN)$,其中 s 为待识别实例的个数。故数据划分算法的是时间复杂度不能高于 $O(sN)$,否则与初始相违背。另一方面,判定待识别的实例落入哪个区域时的时间复杂度也不能高于 $O(sN)$。

(3) 均衡性。在给定空间划分后,为了快速地查询到待识别实例 $\boldsymbol{x}$ 所在的区域,需要促使各个处于区域内的实例个数相等。

随机划分是实现数据划分的最直接方式,其将每个实例随机地放入若干个子集中。虽然此方式简单易行且划分后子集的大小相对均衡,但并未考虑到实例在输入空间中的位置关系,以至于不能保证实例以及其近邻在同一个子集内。Voronoi 图是一种特殊的数据划分方式,其将平面划分为若干个邻域且每个邻域内只含有一个实例,所有落入邻域内的实例具有相同的标签。然而 Voronoi 图是针对平面而构建的,其构建过程十分复杂且时间复杂度较高,并且判定待识别实例落入哪个邻域也十分耗时。为了满足以上数据划分的 3 个条件,必须考虑实例在输入空间中的位置信息,并且划分算法不能过于复杂。为了满足以上 3 个数据划分的条件,提出了一种基于平行超平面的数据划分算法。

训练集 T 中每个实例 $\boldsymbol{x}_i$ 由 m 维的特征向量所表示,而这些特征向量存在的空间称为特征空间。在此特征空间中每一维对应于一个特征,于是实例 $\boldsymbol{x}_i$ 可以看作是在此空间中的一个点。虽然在有些学习算法中假定输入空间与特征空间不同,在 kNN 分类算法中经常假定特征空间为输入空间。众所周知,空间中的 $n-1$ 个平行的超平面 L_l 可以将输入空间划分为 n 个互不相交的区域 R_j,而且落在每个区域

内的实例在很大程度上能够保证它以及其近邻仍在此区域内。如果将落入每个区域 R_j 内的实例看作一个子集 T_j，则其将训练集 T 划分为 n 个互不相交的子集 T_j，$T=\bigcup_{j=1}^{n} T_j$，且对于任意的 $i\neq j$，有 $T_i \cap T_j=\varnothing$。

在图 1 所示的二维平面中有 3 类实例，每类实例使用不同的颜色表示。4 条平行的直线 L_1，L_2，L_3，L_4 将二维平面划分为 5 个互不相交的区域。如果将位于每个区域内的实例作为一个子集，显然其将这 3 类实例划分为 5 个互不相交的子集。明显地看到除了靠近直线的实例之外，位于每个区域内的大部分实例及其近邻仍在同一个区域。故此方法在很大程度上可以保持实例的局部性。

虽然利用 $n-1$ 个平行的超平面可以实现在保持局部性前提下对训练集 T 进行划分，但仍需要确定每一个训练实例 $\boldsymbol{x}_i$ 落入输入空间中的哪一个区域。令 $\boldsymbol{\omega}=(\omega_1,\omega_2,\cdots,\omega_m)$ 为这个 $n-1$ 个平行超平面的方向向量，则超平面 L_l 可以表示为 $\boldsymbol{\omega x}+b_l=0$，其中 b_l 为截距。若将超平面 L_l 按其截距 b_l 从小到大的顺序排序，则纵向来看，截距最大的超平面在最上侧，而截距最小的在最下侧。记 $L_{(1)},L_{(2)},\cdots,L_{(n-1)}$ 为排序后的超平面，那么划分后的区域依次为超平面 $L_{(1)}$ 以上的区域 R_1，超平面 $L_{(l)}$ 与 $L_{(l+1)}$ 之间的区域 R_{l+1}，以及超平面 $L_{(n-1)}$ 以下的区域 R_n，$l=1,2,\cdots,n-2$。由于划分后的区域是由超平面来界定的，故在判定实例 $\boldsymbol{x}_i$ 在哪一个区域时只需要确定其在超平面 $L_{(l)}$ 的下侧还是上侧。众所周知，若实例 $\boldsymbol{x}_i$ 在超平面 $L_{(l)}$ 的上侧，则 $\boldsymbol{\omega x}_i+b_{(l)}=0$；若实例 $\boldsymbol{x}_i$ 在超平面 $L_{(l)}$ 的上侧，则 $\boldsymbol{\omega x}_i+b_{(l)}>0$；若实例 $\boldsymbol{x}_i$ 在超平面 $L_{(l)}$ 下侧，则 $\boldsymbol{\omega x}_i+b_{(l)}<0$。因此，只需要判定 $\boldsymbol{\omega x}_i$ 与 $-b_{(l)}$ 的大小便可以对 $\boldsymbol{x}_i$ 和 $L_{(l)}$ 的位置关系进行判定。

若令 $b_{(0)}=+\infty$，$b_{(n)}=-\infty$，则划分后的区域表示为 $R_{(j)}=\{\boldsymbol{x}\in \boldsymbol{S}:\boldsymbol{\omega x}_i\in[-b_{j-1},-b_{(j)})\}$，$j=1,2,\cdots,n$。因此，对于训练集中的每个实例 $\boldsymbol{x}_i$ 计算 $v_i=\boldsymbol{\omega x}_i$，然后判定 v_i 落入哪个区间，则 $\boldsymbol{x}_i$ 必定在位于此区间对应的那个区域。为了保证落入每个区域内的实例个数近似相等，取 $b_{(l)}=-p(v_i,l/n)$，其中 $p(v_i,l/n)$ 为数列 $v_1,\cdots,v_N$ 的 l/n 分位数，$l=1,2,\cdots,n-1$。显然，数列 $v_1,\cdots,v_N$ 中落入每个区间 $[p_{j-1},p_j)$ 内实数的个数近似相等。

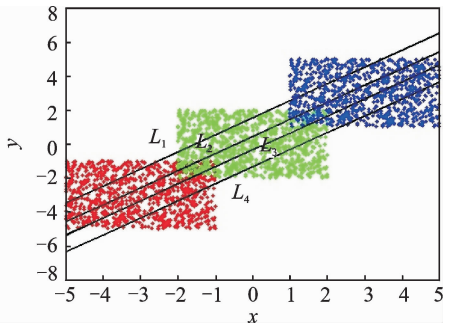

图 1 二维平面中的 3 类实例

Fig. 1 Instances from three classes in two-dimensional space

1.2 抽样样本的获取

在将训练实例所在的空间 $\boldsymbol{S}$ 划分为若干个实例个数相等的区域 $\mathbf{R}_{(j)}$ 后，从每个区域 $\mathbf{R}_{(j)}$ 对应的子集 $T_{(j)}=\{\boldsymbol{x}_i\in T:\boldsymbol{x}_i\in\mathbf{R}_{(j)}\}$ 内随机抽取容量为 $[\alpha\mid T_{(j)}\mid/n]$ 的实例子集 $S_{(j)}$，最后将所有的子集 $S_{(j)}$ 合并为最终的抽样实例子集 $S=\bigcup_{j=1}^{n} S_{(j)}$，其中 α 为抽样比例，$[\alpha\mid T_{(j)}\mid/n]$ 为最接近 $\alpha N/n$ 的整数，$\mid T_{(j)}\mid$ 为集合 $T_{(j)}$ 的大小。

1.3 待识别实例的预测

对于给定待识别的实例 $\boldsymbol{x}$，计算 $v=\boldsymbol{wx}$，然后判定 v 属于哪个区间，则 $\boldsymbol{x}$ 必定在位于此区间对应的那个区域。假定实例 $\boldsymbol{x}$ 落入区域 $R_{(h)}$，虽然可以利用 $S_{(h)}$ 对待识别实例 $\boldsymbol{x}$ 进行预测，但并不能完全保证 $\boldsymbol{x}$ 在 S 中的 k 个近邻仍在 $S_{(h)}$ 中。为了促使待识别实例及其 k 个近邻在同一个区域，将待识别实例 $\boldsymbol{x}$ 所在的区域 $R_{(h)}$ 扩大为它以及与它相邻的两个区域的并集，相应的 $S_{(h)}$ 扩充为 $S(\boldsymbol{x})$，扩充的规则为

$$S(x)=\begin{cases}S_{(1)}\cup S_{(2)}\cup S_{(3)} & x\in\mathbf{R}_{(1)}\\ S_{(j-1)}\cup S_{(j)}\cup S_{(j+1)} & x\in\mathbf{R}_{(j)}\\ S_{(n-2)}\cup S_{(n-1)}\cup S_{(n)} & x\in\mathbf{R}_{(n)}\end{cases}\tag{1}$$

式中,$j=2,\cdots,n-1$。最后实例 $\boldsymbol{x}$ 只需要在从区域 $R(x)$ 中抽取的实例子集中寻找其 k 个近邻。

图 2 描述了二维平面中由红黑两种颜色标示的两类实例,4 条平行的直线 L_1,L_2,L_3,L_4 将平面划分为 5 个区域 R_1,R_2,R_3,R_4,R_5。假设由蓝色标示的待识别的实例 $\boldsymbol{x}$ 落入区域 R_3 中,明显地看到实例 $\boldsymbol{x}$ 在 R_3 中的 1 近邻为 $\boldsymbol{x}_1$,依照 kNN 分类则实例 $\boldsymbol{x}$ 标示为红色。然而 $\boldsymbol{x}$ 在整个平面中的近邻为 $\boldsymbol{x}_5$,应该标示为黑色。显然,实例 $\boldsymbol{x}$ 在区域 R_3 并未正确地找到其真正的 1 近邻。为此,依据以上规则将区域 R_3 扩充为 $R(\boldsymbol{x})=R_2\cup R_3\cup R_4$。明显地看到,实例 $\boldsymbol{x}$ 在 $R(\boldsymbol{x})$ 中的 1 近邻为 $\boldsymbol{x}_5$,这与在整个平面中寻找的 1 近邻一致。此外,实例 $\boldsymbol{x}$ 在区域 $R(\boldsymbol{x})$ 中的 5 个近邻依次为 $\boldsymbol{x}_5,\boldsymbol{x}_1,\boldsymbol{x}_7,\boldsymbol{x}_4,\boldsymbol{x}_2$,这也与在整个平面中寻找的 5 个最近近邻保持一致。综上所述,SS-kNN 算法的基本流程如图 3 所示。

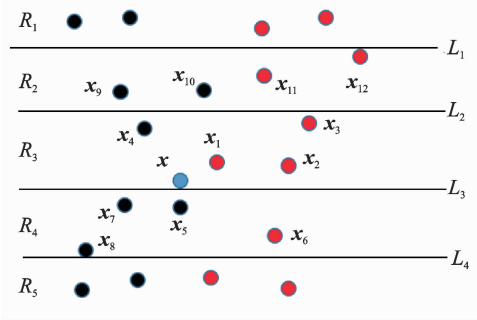

图 2 二维空间的划分

Fig. 2 Partition of two-dimensional space

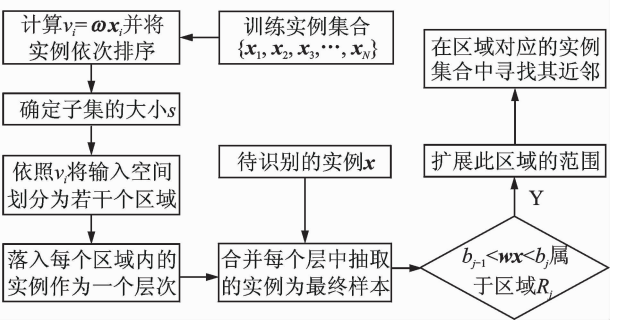

图 3 SS-kNN 算法流程图

Fig. 3 Flow chart of SS-kNN algorithm

2 实验结果与分析

2.1 实验准备

为了比较 SS-kNN 算法与 kNN,RS-kNN 算法在性能上的差异,选择了 LibSVM[26] 以及 UCI[27] 中的 11 个数据集,这些数据集的容量均大于 50 000,其中最大的为 5 000 000。表 1 给出了这些数据集的相关信息。

文献[3]中提供了大量的评价算法性能指标,其中分类精度以及运行时间为最常用的两个指标。前者反映了算法的分类能力,而后者是对算法运行速度的评估。为了估计分类精度和运行时间,采用十折交叉验证的方式。在此方法中,当前的数据被随机地分为大小近似相等的 10 个子集,并随机将其中一个子集作为测试集而剩余的作为训练集,然后将此过程重复 10 次,最后将这十次结果的平均作为最终的结果。

为了对这两个算法作一个公平的对比,均使用 Matlab 编写程序且在同一台电脑上运行。在所有的实验中,方向向量 $\boldsymbol{\omega}$ 取定为均值向量,抽样比例为 $\alpha=0.3$,$k=1,3,5,7$,$n=20$。

表 1 实验中使用的 11 个数据集

Tab. 1 11 used datasets in experiment

数据集	简称	特征	大小	类别
Acoustic	ACO	50	98 528	3
Cod-nad	COD	8	59 535	2
Combine	COM	100	98 528	3
Covtype	COV	54	581 012	7
Ijcnn1	IJC	22	141 691	2
Kddcup99	KDD	41	494 020	23
MiniBooNE_PID	MIN	50	130 065	2
Poker	POK	10	1 025 010	10
Shkin-skion	SKI	3	245 057	2
Shuttle	SHU	9	58 000	7
Susy	SUS	18	5 000 000	2

2.2 实验分析

2.2.1 分类精度

表 2,3 罗列了 3 个算法在 $k=1,3,5,7$ 以及不同的数据集上的分类精度。为了直观地比较两种算法在每个数据集上分类精度的差异,绘制了图 4~7。在每

表 2 3 种算法在 $k=1,3$ 时的分类精度

Tab. 2 Accuracy of three algorithms as $k=1,3$

数据集	$k=1$			$k=3$		
	SS-kNN	k-NN	RS-kNN	SS-kNN	k-NN	RS-kNN
ACO	0.665	0.673	0.670	0.705	0.729	0.710
COD	0.668	0.668	0.674	0.664	0.668	0.675
COM	0.757	0.758	0.760	0.799	0.808	0.800
COV	0.917	0.945	0.918	0.908	0.944	0.909
IJC	0.975	0.985	0.975	0.975	0.984	0.975
KDD	0.998	0.999	0.998	0.998	0.998	0.998
MIN	0.841	0.847	0.847	0.862	0.873	0.868
POK	0.476	0.503	0.482	0.504	0.517	0.513
SKI	0.998	1.000	0.999	0.995	1.000	0.996
SHU	0.997	0.998	0.998	0.997	0.998	0.998
SUS	0.663	0.68	0.665	0.696	0.710	0.698
平均	0.814	0.823	0.817	0.828	0.839	0.831
差异		0.009	0.003		0.011	0.003

表 3 3 种算法在 $k=5,7$ 时的分类精度

Tab. 3 Accuracy of three algorithms as $k=5,7$

数据集	$k=5$			$k=7$		
	SS-kNN	k-NN	RS-kNN	SS-kNN	k-NN	RS-kNN
ACO	0.705	0.729	0.710	0.723	0.743	0.730
COD	0.664	0.668	0.675	0.668	0.668	0.676
COM	0.799	0.808	0.800	0.814	0.822	0.816
COV	0.908	0.944	0.909	0.901	0.940	0.903
IJC	0.975	0.984	0.975	0.969	0.983	0.969
KDD	0.998	0.998	0.998	0.998	0.997	0.998
MIN	0.862	0.873	0.868	0.869	0.879	0.869
POK	0.504	0.517	0.513	0.510	0.532	0.523
SKI	0.995	1.000	0.996	0.996	1.000	0.997
SHU	0.997	0.998	0.998	0.996	0.998	0.997
SUS	0.696	0.71	0.698	0.711	0.726	0.713
平均	0.828	0.839	0.831	0.832	0.844	0.835
差异		0.011	0.003		0.012	0.003

幅图中横坐标为数据集，纵坐标为对应数据集上两种算法的分类精度。

由于 SS-kNN 算法是基于随机抽样而设计的近邻算法，且相对于原始数据而言，抽样样本所含的信息有所损失，故其分类精度在一定程度上略低于 kNN 算法。从图 4～7 可以明显地看到 SS-kNN 算法在 11 个数据集上的分类精度均不高于 kNN。具体而言，当 $k=1$ 时，SS-kNN 在数据集 COV，IJC，POK 以及 SUS 上的分类精度略低于 kNN 与 RS-kNN，除此之外均没有明显的差异；当 $k=3$ 时，除了在数据集 ACO，COV，MIN，IJC 以及 SUS 之外，三者没有明显的差异；当 $k=5,7$ 时，SS-kNN 算法在数据集 ACO，COV，POK 以及 SUS 上的分类精度均高于 kNN 与 RS-kNN，其余均相差无几。就平均状态而

言,从表 2～5 可以明显地看到在 k 的不同取值下,与 kNN 相比,差异最大为 0.012,最小为 0.009;与 SS-kNN 相比的差异为 0.003。因此,在整体状况以及 k 的不同取值下,SS-kNN 算法与 kNN,RS-kNN 在分类精度上差异相对较小。

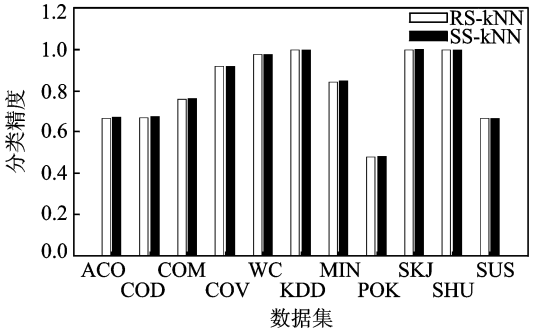

图 4 $k=1$ 时 3 种算法的分类精度的对比

Fig. 4 Comparison of accuracy among three algorithms as $k=1$

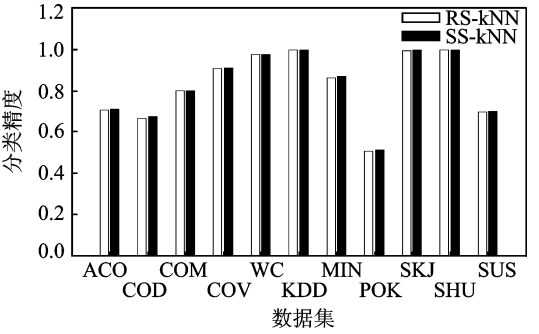

图 5 $k=3$ 时 3 种算法的分类精度的对比

Fig. 5 Comparison of accuracy among three algorithms as $k=3$

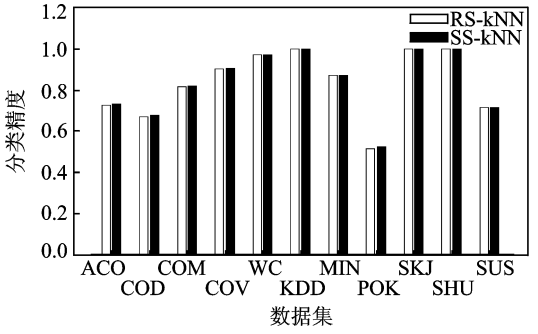

图 6 $k=5$ 时 3 种算法的分类精度的对比

Fig. 6 Comparison of accuracy among three algorithms as $k=5$

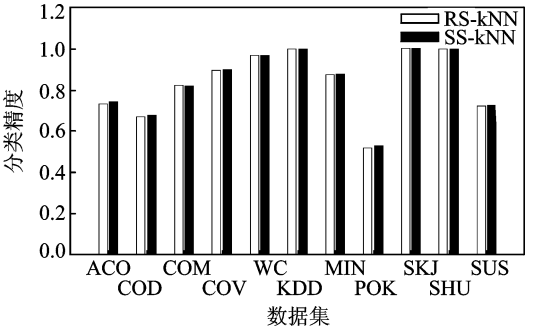

图 7 $k=7$ 时 3 种算法的分类精度的对比

Fig. 7 Comparison of accuracy among three algorithms as $k=7$

2.2.2 运行时间

在现实生活中,不仅要求学习算法的分类性能要高,而且需要其效率相对较高。学习算法的运行时间是算法效率的体现,运行时间越短,则算法的效率越高。在给定待识别实例的近邻序列时,寻找其 k 个近邻只需要取其近邻序列中的前 k 个元素即可。因此,在 k 的不同取值下,kNN 算法的运行时间是没有显著性差异的。为了比较 SS-kNN 算法与 RS-kNN 在运行时间上的差异,选择 $k=7$。 表 4 罗列了 3 种算法在不同数据集上的运行时间,具体如图 8 所示。

从图 8 明显地看到这 3 种算法在每个数据集上的运行时间差异较大。SS-kNN 算法的运行时间明显地低于 kNN,RS-kNN 算法,且运行时间的差异与数据集的大小正相关。在容量相对较大的数据集 COD,COV,KDD,SKI 以及 SUS 上,SS-kNN 算法与 kNN,RS-kNN 算法在运行时间上的差距明显地比其他的要大。就平均而言,3 种算法在不同数据集上的平均运行时间分别为 167.843,66 917.395,2 465.046 s,kNN,RS-kNN 算法的运行时间约是 SS-kNN 算法的 399 倍和 16 倍。

表 4 3 种算法在不同数据集上的运行时间

Tab. 4 Running time of three kinds of algorithms on different datasets

数据集	SS-kNN	kNN	RS-kNN
ACO	19.450	1 688.175	62.525
COD	448.090	388 390.464	13 871.088
COM	32.278	5 411.458	186.602
COV	386.739	89 092.089	3 299.707
IJC	19.525	2 144.408	76.586
KDD	228.234	49 900.023	1 848.149
MIN	18.139	2 537.248	90.616
POK	112.458	30 286.295	1 044.355
SKI	301.201	75 828.432	3 159.518
SHU	28.882	5 983.208	213.686
SUS	251.271	84 829.550	3 262.675
平均	167.843	66 917.395	2 465.046

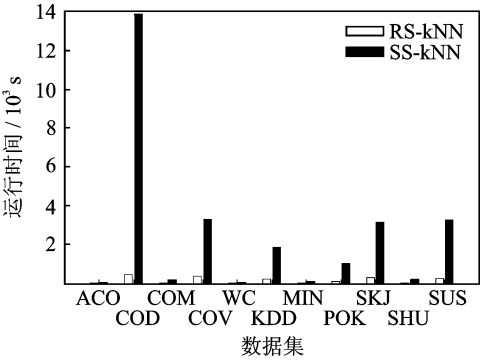

图 8 3 种算法运行时间的比较

Fig. 8 Comparison of running time among three algorithms

2.2.3 参数 n 的影响

参数 n 控制划分子集的个数并直接影响着 SS-kNN 算法的运行时间。n 取值越大，划分后每个实例子集的容量越小，而待识别实例寻找其近邻的空间越小，则 SS-kNN 算法的运行时间越短。为了研究参数 n 对 SS-kNN 算法的影响。选择 $n=10,20,30,40$，并讨论其对分类精度以及运行时间的影响。

从图 9-12 可以明显地看出，在 k 的取值确定的条件下，针对 n 的不同取值，SS-kNN 算法的分类精

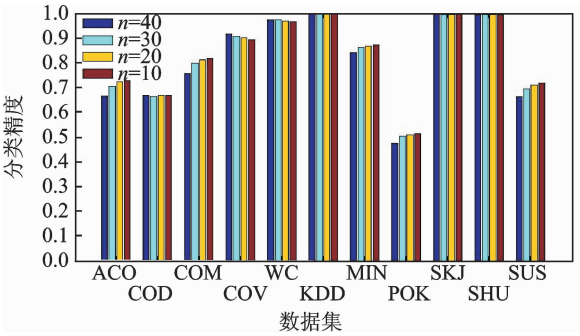

图 9 SS-1NN 算法在 n 的不同取值下的分类精度

Fig. 9 Accuracy of SS-1NN with different n

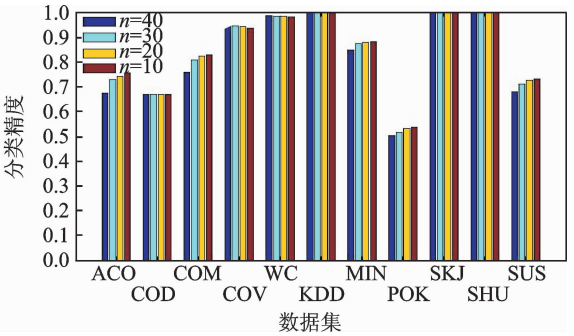

图 10 SS-3NN 算法在 n 的不同取值下的分类精度

Fig. 10 Accuracy of SS-3NN with different n

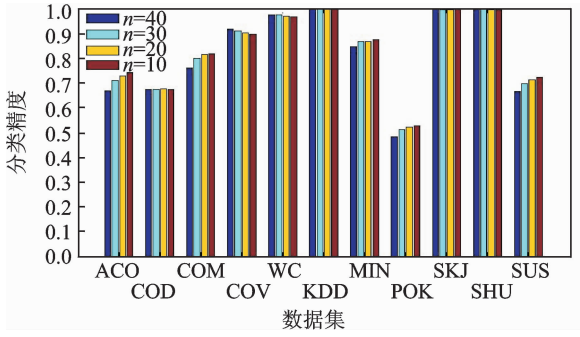

图 11 SS-5NN 算法在 n 的不同取值下的分类精度

Fig. 11 Accuracy of SS-5NN with different n

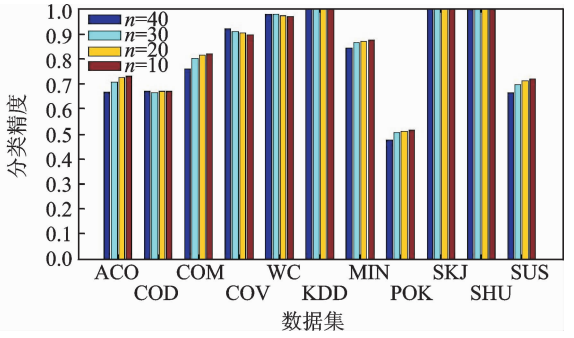

图 12 SS-7NN 算法在 n 的不同取值下的分类精度

Fig. 12 Accuracy of SS-7NN with different n

度并没有显著性变化。换言之,当 n 取值于区间[10,40]时,SS-kNN 算法的分类性能并不受 t 值变化的影响。而在图 13 中,随着 n 值的不断增大,SS-kNN 算法中搜索待检验实例的近邻的实例子集容量不断变小,进而算法的运行时间不断减少,而这验证了之前的推断。

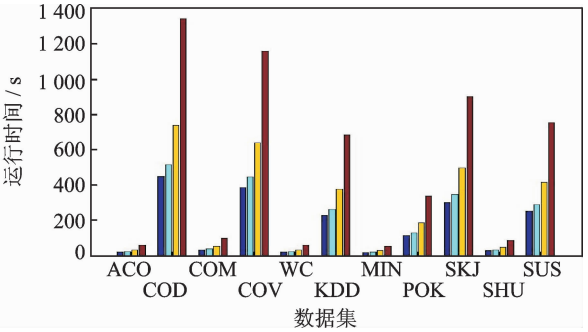

图 13 SS-kNN 算法在 n 的不同取值下的运行时间

Fig. 13 Running time of SS-kNN with different n

3 结束语

为了解决传统 kNN 算法面对大规模数据预测时间较长的缺陷,本文利用分而治之的思想提出了一种基于分层抽样的 kNN 分类加速算法。首先利用多个平行的超平面将训练实例 T 所在的空间划分为若干个子区域并将落入每个子区域内的实例作为一个子集 T_i,然后从 T_i 中抽取实例子集 S_i,最后将子集 S_i 合并作为最终的抽样实例子集 $S=\bigcup S_i$。对于给定的待识别实例 $\boldsymbol{x}$,首先判定其处在哪一个区域,从此区域以及相邻区域抽取的实例中去寻找 $\boldsymbol{x}$ 的 k 个近邻并提供预测标签。实验表明,在不同的数据集以及 k 的不同取值下,SS-kNN 算法均获得了与 RS-kNN 算法相近的分类精度,但其运行时间大大缩短。在平均状态下 SS-kNN 算法可以将 kNN,RS-kNN 算法的运行速度分别提高约 399 倍和 16 倍。此外,本文还对 SS-kNN 算法中涉及的参数 n 的影响做了研究,实验表明参数 n 的大小没有对 SS-kNN 算法的分类精度产生明显的影响,但其运行时间随着 t 的增大而降低。本文建议 n 的取值在区间[10,40]内。

随机抽样方法有助于在保持数据价值总量相对不变的前提下降低数据规模,是一种重要的数据预处理方法,被广泛地应用到人工智能领域。面对汹涌而来的大数据,如何针对大数据价值密度相对较低的特点设计合适的抽样方法是一个巨大的挑战。此外,样本量的大小是抽样方法的关键,需要发展一个合适的算法来确定样本量。

参考文献:

[1] 维克托.迈尔-舍恩伯格.大数据时代[M].杭州:浙江人民出版社,2012:2-43.
Mayer-Schonberger V. The age of big data[M]. Hangzhou: Zhejiang People's Publishing House,2012:2-43.

[2] 吉根林,赵斌. 时空轨迹大数据模式挖掘研究进展[J]. 数据采集与处理,2015,30(1):47-58.
Ji Genlin,Zhao Bin. Research progress in pattern mining for big spatio-temporal trajectories[J]. Journal of Data Acquisition and Processing,2015,30(1):47-58.

[3] 王元卓,靳小龙,程学旗. 网络大数据:现状与展望[J]. 计算机学报,2013,36(6):1125-1138.
Wang Yuanzuo, Jin Xiaolong, Cheng Xueqi. Network big data: Present and future[J]. Chinese Journal of Computers, 2013, 36(6): 1125-1138.

[4] 张引,陈敏,廖小飞. 大数据应用的现状与展望[J]. 计算机研究与发展,2013,50(11):216-233.
Zhang Yin, Chen Min, Liao Xiaofei. Big data applications: A survey[J]. Journal of Computer Research and Development, 2013, 50(11): 216-233.

[5] 金澈清,钱卫宁,周敏奇,等. 数据管理系统评测基准:从传统数据库到新兴大数据[J]. 计算机学报,2015,38(1):18-34.
Jin Cheqing, Qian Weining, Zhou Minqi, et al. Benchmarking data management systems: From traditional database to emergent big data[J]. Chinese Journal of Computers, 2015, 38(1): 18-34.

[6] Han J,Kamber M. Data mining: Concepts and techniques [M]. Burlington: Morgan Kaufmann,2011: 1-25.

[7] 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.

[8] Cover T M,Hart P E. Nearest neighbor pattern classification [J]. IEEE Transaction on Information Theory,1976,13(1): 21-27.

[9] Kononenko I,Kukar M. Machine learning and data mining:Introduction to principles and algorithms [M]. Chichester: Hor-

wood Publishing, 2007: 2-24.

[10] Garcia S, Derrac J, Cano J R, 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.

[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] 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.

[13] 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.

[14] Czarnowski I. Cluster-based instance selection for machine classification [J]. Knowledge and Information Systems, 2012, 30(1): 113-133.

[15] Biçici E, Yuret D. Optimizing instance selection for statistical machine translation with feature decay algorithms [J]. IEEE/ACM Transactions on Audio, Speech, and Language Processing, 2015, 23(2): 339-350.

[16] de Haro-García A, García-Pedrajas N. A divide-and-conquer recursive approach for scaling up instance selection algorithms [J]. Data Mining and Knowledge Discovery, 2009, 18(3): 392-418.

[17] García-Osorio C, de 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.

[18] 金勇进, 杜子芳, 蒋妍. 抽样技术[M]. 北京: 中国人民大学出版社, 2015: 1-22.
Jin Yongjin, Du Zifang, Jiang Yan. Sampling technique[M]. Beijing: Chinese People's Publishing House, 2015: 1-22.

[19] Chen B, Haas P, Scheuermann P. A new two-phase sampling based algorithm for discovering association rules[C]//Proceedings of the eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. Edmonton: ACM, 2002: 462-468.

[20] Domingo C, Gavaldà R, Watanabe O. Adaptive sampling methods for scaling up knowledge discovery algorithms [J]. Data Mining and Knowledge Discovery, 2002, 6(2): 131-152.

[21] Byrd R H, Chin G M, Nocedal J, et al. Sample size selection in optimization methods for machine learning [J]. Mathematical Programming, 2012, 134(1): 127-155.

[22] Kleiner A, Talwalkar A, Sarkar P, et al. A scalable bootstrap for massive data [J]. Journal of the Royal Statistical Society: Series B (Statistical Methodology), 2014, 76(4): 795-816.

[23] Jayaram N, Baker J W. Efficient sampling and data reduction techniques for probabilistic seismic lifeline risk assessment [J]. Earthquake Engineering & Structural Dynamics, 2010, 39(10): 1109-1131.

[24] Wang L, Leckie C, Kotagiri R, et al. Approximate pairwise clustering for large data sets via sampling plus extension [J]. Pattern Recognition, 2011, 44(2): 222-235.

[25] Liang J, Wang F, Dang C, et al. An efficient rough feature selection algorithm with a multi-granulation view [J]. International Journal of Approximate Reasoning, 2012, 53(6): 912-926.

[26] Chang C C, Lin C J. LIBSVM: A library for support vector machines [J]. ACM Transactions on Intelligent Systems and Technology (TIST), 2011, 2(3): 27.

[27] Blake C, Keogh E, Merz C J. UCI repository of machine learning databases [EB/OL]. http://www.ics.uci.edu/~mlearn/MLRepository.html, 2015-12-12.

作者简介:

宋云胜(1984-),男,博士研究生,研究方向:统计机器学习,E-mail:sys_sd@126.com。

梁吉业(1962-),男,教授,博士生导师,研究方向:数据挖掘、机器学习和粒计算等。