大数据下的快速 KNN 分类算法*

苏毅娟[1], 邓振云[2†], 程德波[2], 宗 鸣[2]
(1. 广西师范学院 计算机与信息工程学院, 南宁 530023; 2. 广西师范大学 广西多源信息挖掘与安全重点实验室和广西区域多源信息集成与智能处理协同创新中心, 广西 桂林 541004)

摘 要: 针对 K 最近邻算法测试复杂度至少为线性,导致其在大数据样本情况下的效率很低的问题,提出了一种应用于大数据下的快速 KNN 分类算法。该算法创新性地在 K 最近邻算法中引入训练过程,即通过线性复杂度聚类方法对大数据样本进行分块,然后在测试过程中找出与待测样本距离最近的块,并将其作为新的训练样本进行 K 最近邻分类。这样的过程大幅度地减少了 K 最近邻算法的测试开销,使其能在大数据集中得以应用。实验表明,该算法在与经典 KNN 分类准确率保持近似的情况下,分类的速度明显快于经典 KNN 算法。

关键词: K 最近邻; 测试复杂度; 大数据; 分块; 聚类中心
中图分类号: TP301.6 **文献标志码**: A **文章编号**: 1001-3695(2016)04-1003-04
doi:10.3969/j.issn.1001-3695.2016.04.009

Fast KNN classification algorithm under big data

Su Yijuan[1], Deng Zhenyun[2†], Cheng Debo[2], Zong Ming[2]
(1. *College of Computer & Information Technology, Guangxi Teachers Education University, Nanning* 530023, *China*; 2. *Guangxi Key Laboratory of Multi-source Information Mining & Security and Guangxi Collaborative Innovation Center of Multi-Source Information Interation & Intelligent Processing, Guangxi Normal University, Guilin Guangxi* 541004, *China*)

Abstract: Aiming at the problems of the K-nearest neighbor algorithm, testing complex is linear at least, and lead to the accuracy is low when the samples are large. This paper proposed a fast KNN classification algorithm faster than the traditional KNN did. The proposed algorithm innovatively introduced the training process during the KNN method, i.e., the algorithm blocked the big data by linear complexity clustering. Then, the algorithm selected the nearest cluster as new training samples and established a classification model. This process reduced the KNN algorithm testing overhead, which made the proposed algorithm could be applied to big data. Experiments result shows that the accuracy of the proposed KNN classification is similarity than the traditional KNN, but the classification speed has been significantly improved.

Key words: K-nearest neighbor(KNN); testing complex; big data; block; cluster centers

0 引言

随着互联网的迅猛发展,大数据不断地产生,分类作为当前数据挖掘中最实用的技术之一,已得到广泛的应用。目前常用的分类方法有决策树、人工神经网络、SVM、Bayes、KNN 等。KNN 算法因其简单和有效在分类算法中得到了广泛的应用,其基本思想是:在训练样本中找到待测样本的 k 个最近邻样本,然后根据这 k 个最近邻样本的类进行投票,以此来决定测试样本的类别。但是 KNN 在寻找最近邻样本的过程中,需要逐个计算测试样本与每个训练样本的距离(或相似度),当训练样本为大数据时,将会产生很高的计算开销,如果大数据集是保存在内存(或硬盘)中,这种逐个扫描的方式几乎不可行。

目前提出了许多对于 KNN 的改进算法,文献[1,2]提出一种从所有已知样本中选取测试样本的 k 个最近邻,然后建立分类器进行分类的方法,提高了分类的性能。文献[3]提出了一种基于密度的训练样本裁剪方法,使样本分布密度趋于平均,降低了 KNN 的计算量。文献[4]提出了一种基于利用中心文档代替原始样本建立分类模型的方法,减少了 KNN 算法需要进行相似计算的样本数,从而达到提高分类速度的目的。文献[5]提出了减少大量计算的排类算法和归类算法,在不影响原有准确率的情况下,构建了一种基于 KNN 的快速分类算法,提高了分类的速度。这些算法主要是通过快速搜索算法、降维[6~8]或通过一定的优化策略直接减少需要进行相关性计算的样本数,从而提高分类的效率。但当面临大数据和高维样本时,这些分类方法的效率将会大幅度降低[9~11]。

针对 KNN 算法无训练过程的特点[12,13],本文创新性地对其引入一个训练过程,即首先采用线性复杂度聚类方法对大数据进行分块。而在测试过程,对于每一个测试样本,首先找出与待测样本距离最近的聚类中心所在的簇作为新的训练样本集,然后对新训练集建立分类模型进行分类。由于聚类后簇内样本的相似度高,所以该算法能达到既减少计算量,又能保持

收稿日期: 2014-11-30; **修回日期**: 2015-01-23 **基金项目**: 国家自然科学基金资助项目(61450001,61263035,61573270);国家“863”计划资助项目(2012AA011005);国家“973”计划资助项目(2013CB329404);广西自然科学基金资助项目(2012GXNSFGA060004,2014jjAA70175,2015GXNSFAA139306,2015GXNSFCB13901);广西八桂创新团队、广西百人计划和广西高校科学技术研究重点项目(2013ZD04)

作者简介: 苏毅娟(1976-),女,广西桂林人,副教授,主要研究方向为机器学习和数据挖掘;邓振云(1991-),男(通信作者),江西南昌人,硕士,主要研究方向为机器学习和数据挖掘(597277287@qq.com);程德波(1990-),男,江西丰城人,硕士,主要研究方向为数据挖掘和机器学习;宗鸣(1990-),男,江苏泰州人,硕士,主要研究方向为机器学习和数据挖掘.

较高分类准确率的目的。

1 基于聚类的 KNN 改进算法

本文算法包括两个过程,即训练过程和测试过程。训练过程主要选取适用的聚类算法对大数据样本进行分类;测试过程主要是对测试样本在它最近的簇运行 KNN 算法。

1.1 训练过程

聚类是数据分析中最基本的一种技术,它在数据挖掘、机器学习、模式识别方面都得到广泛应用。所谓聚类就是将数据对象分组成多个类和簇的过程。其中,聚类所生成的簇包含簇内样本相似性高和簇之间差异性大两个性质。

目前的聚类方法有很多,如基于划分的 K-means 算法、基于层次的 BIRCH 算法[14] 和基于密度的 DBSCAN 算法[15] 等。但是面临高维大样本数据时,常见的聚类算法将在时间复杂度方面受到极大挑战。因此,对高维大样本数据进行聚类需要满足一些条件:复杂度低,最好是线性的,如文献[16,17]。为此,本文采用了文献[18]提出的聚类算法,即基于界标的谱聚类(landmark-based spectral clustering,LSC)算法,该算法主要是选取 $p(p<<n)$ 个具有代表性的点作为界标点,并将这 p 个界标点进行线性组合以替代原始数据。常见的谱聚类算法[19] 通常用所有样本表达每个数据,这种方法大大降低了相似矩阵的复杂度,即从二次降到了一次,从而也顺带把特征根求解的复杂度降低到了线性[20]。

LSC 算法通过"压缩"原始数据找出一组基向量去代表每个数据点,即找出 p 个具有代表的界标点。常见的选取界标点的方法有随机选取法和 K-means 聚类。随机选取法即随机选取界标点;而 K-means 聚类算法通过简单重复算法几次(通常低于 10 次),把得到的聚类中心作为界标点。本文重复 K-means 聚类算法 10 次,把得到的聚类中心作为界标点。

把每个界标点当做一个向量组成界标矩阵 $\boldsymbol{U}$,LSC 算法用得到的 p 个界标点表示所有的原始数据点 $\boldsymbol{X}=[x_1,\cdots,x_n]\in \mathrm{R}^{m\times n}$,即找出 $\boldsymbol{X}$ 在界标矩阵上的投影 $\boldsymbol{Z}$。

$$z_{ji}=\frac{K_h(x_i,u_j)}{\sum_{j'\in U_{\langle i\rangle}}K_h(x_i,u_{j'})}\quad j\in U_{\langle i\rangle} \tag{1}$$

其中:u_j 是矩阵 $\boldsymbol{U}$ 的第 j 个列向量,$U_{\langle i\rangle}$ 是由 x_i 的 r 个最近邻界标组成的 $\boldsymbol{U}$ 的子矩阵。易知,矩阵 $\boldsymbol{Z}$ 的运行时间为 $O(pmn)$。接着对其进行谱聚类分析,首先构造图矩阵 $\boldsymbol{W}$。

$$\boldsymbol{W}=\hat{\boldsymbol{Z}}^{\mathrm{T}}\hat{\boldsymbol{Z}} \tag{2}$$

其中:$\hat{\boldsymbol{Z}}=D^{-1/2}\boldsymbol{Z}$,$D$ 是 $\boldsymbol{Z}$ 中行向量的和,则矩阵 $\boldsymbol{W}$ 即为一个单位矩阵 $\boldsymbol{I}$。当得到单位图矩阵 $\boldsymbol{W}$ 后,通过奇异值分解计算 $\hat{\boldsymbol{Z}}$ 的特征向量如下:

$$\hat{\boldsymbol{Z}}=\boldsymbol{A}\boldsymbol{\Sigma}\boldsymbol{B}^{\mathrm{T}} \tag{3}$$

易知,左奇异向量 $\boldsymbol{A}=[a_1,\cdots,a_k]\in \mathrm{R}^{p\times p}$ 为矩阵 $\hat{\boldsymbol{Z}}\hat{\boldsymbol{Z}}^{\mathrm{T}}$ 前 k 个特征向量,右奇异向量 $\boldsymbol{B}=[b_1,\cdots,b_k]\in \mathrm{R}^{n\times p}$ 为矩阵 $\hat{\boldsymbol{Z}}^{\mathrm{T}}\hat{\boldsymbol{Z}}$ 的特征向量。由于矩阵 $\boldsymbol{W}=\hat{\boldsymbol{Z}}^{\mathrm{T}}\hat{\boldsymbol{Z}}$ 为 $p\times p$ 维,可知向量 $\boldsymbol{A}$ 的计算时间为 $O(p^3)$,而特征向量 $\boldsymbol{B}$ 的计算时间则可通过下式得出:

$$\boldsymbol{B}^{\mathrm{T}}=\boldsymbol{\Sigma}^{-1}\boldsymbol{A}^{\mathrm{T}}\hat{\boldsymbol{Z}} \tag{4}$$

特征向量 $\boldsymbol{B}$ 的总时间为 $O(p^3+p^2n)$[21]。由于 $p<<n$,所以当选取特征向量 $\boldsymbol{B}$ 进行 K-means 聚类时,算法运行的总时间则由 $O(n^3)$ 降为 $O(p^3)$,算法的效率得到明显的提高。这种低复杂度的算法非常适合在大数据方面的应用。最后,给出 LSC 算法的具体步骤,如下所示:

算法 1 LSC 算法

输入:n 个数据点 $x_1,x_2,\cdots,x_n\in \mathrm{R}^m$ 以及聚类个数 k。

输出:k 个子簇。

通过 K-means 选取 p 个界标点;
根据式(1)投影原始数据 $\boldsymbol{X}$ 到界标矩阵,得到原始矩阵的表示 $\boldsymbol{Z}\in \mathrm{R}^{p\times n}$;
根据矩阵 $\boldsymbol{Z}$,计算 $\boldsymbol{Z}\boldsymbol{Z}^{\mathrm{T}}$ 的前 k 个特征向量 $\boldsymbol{A}$;
根据式(4)计算特征向量 $\boldsymbol{B}$;
运用 K-means 对特征向量 $\boldsymbol{B}$ 进行最终的聚类,并输出 k 个子簇。

1.2 测试过程

本文在采用 LSC 算法得到 k 个子簇并求出 k 个聚类中心后,找出距离待测样本最近的聚类中心所在的子簇,将其作为新的训练样本。由于通过 LSC 聚类得到的子簇中的样本都是彼此相似的,所以在选定的子簇中进行 KNN 分类可以充分保证分类的准确率。最后,给出基于 LSC 聚类的 KNN 改进算法的具体步骤,如下所示:

算法 2 LC-KNN 算法

输入:数据集。

输出:待测样本类标签的预测值。

使用 LSC 聚类方法对训练样本进行聚类,得到 m 个聚类中心 $C_1,C_2,C_3,\cdots,C_m$;
计算待测样本 y 与所有聚类中心的距离 $D(y,C_i)$,将与之距离最近的聚类中心所在的簇作为新的训练样本,即 $\mathrm{New}X_i=\min\{D(y,C_i)\}\ i=1,2,\cdots,m$;
在新训练样本 $\mathrm{New}X_i$ 中对待测样本进行 KNN 分类,得到待测样本 y 的 k 个最近邻,并通过投票确定其类标签的预测值 class_i。

通过算法 2 可知,当聚类的个数较多,即 m 较大时,新的训练样本 $\mathrm{New}X_i$ 的样本数将远小于原始样本数,可以很容易达到大幅度减少 KNN 的计算量[22,23]、提高分类速度的目的。但是随着 m 的增大,聚类的开销也会同时增加,且新训练样本 $\mathrm{New}X_i$ 中的样本数将会逐渐减小,这很有可能会导致分类准确率的下降。为了避免这种情况,聚类的个数 m 需要设置在一个比较合理的数值。

极端情况下,假定令 m 为样本个数时,则为 1NN 算法,该算法分类的效率较高,但当训练样本分布比较集中时,则很有可能导致分类准确率降低;而令 $m=1$ 时,则为经典的 KNN 算法。因此,常见 K 最近邻算法是本文算法的一种特例。通过以上分析可知,聚类的簇的个数 m 越大,KNN 分类需要扫描的样本数也就越少,运行速度越快。但是考虑到样本集为大数据,如果 m 较小,簇中样本数依然很大,不能在内存(数据空间)中运行。因此,假设程序运行占用的总内存为 M,计算机系统内存为 M_0,样本集中最小类样本的个数为 n_0,那么 m 的取值范围可表示为 $M/M_0<m<n_0$。

在确定新训练样本 $\mathrm{New}X_i$、对测试样本选取 k 个最近邻过程中,k 值是非常重要的参数之一。文献[24]建议 $k=\sqrt{n}\ (n>100)$,n 是测试样本的个数,但这种取法通常不能得到满意的结果,并且该想法也没有理论保证。由于本文提出的基于 LSC 方法得到的子簇 $\mathrm{New}X_i$ 中样本之间都是相关的,考虑到在大数据中需要减少 KNN 的计算量,所以 k 值的选取不宜过大,在保持较高分类准确率的情况下,k 值的选取应尽可能小。通过实验发现,当 $k=1$、2 时,分类的结果较优。

1.3 图例说明

KNN 分类算法虽然简单、有效且准确率高,但是缺点也很明显,其时间复杂度几乎与样本数成正比。这是因为 KNN 算法

是一种懒惰的基于实例的学习方法,每次在寻找待测样本的 k 个最近邻时,都需要计算其与所有训练样本的距离,当训练样本较大时,分类的时间也成正比增加。考虑到目前 KNN 分类都是在数据空间中进行,在样本为大数据情况下要扫描数据集(或者从磁盘/内存中读取)一遍几乎不可能。为了解决这个问题,本文将样本集分成多个较小的块(保证每个块都能够在内存中运行),并将具有代表性的块作为新的训练样本进行 KNN 分类。如图 1 所示,对样本集进行聚类得到三个聚类中心,分别计算测试样本与三个聚类中心的距离。可知测试样本 1 距离簇 1 的聚类中心最近,因此本算法将簇 1 作为样本 1 的新的训练样本;而对于测试样本 2,其最近邻样本中虽然包含簇 2、簇 3 的样本,但是距离最近的聚类中心为簇 3,所以将簇 3 作为新的测试集。考虑到仅将簇 3 作为新的训练样本可能会影响分类准确率,但通过 LSC 算法得到的簇内样本之间具有较高的相似性,簇簇之间样本差异性大,所以将簇 3 作为新的训练样本集进行 KNN 分类时,依旧能够保持较高的分类准确率。

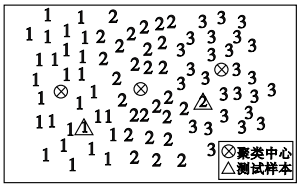

图1　改进算法中测试样本选择训练样本集的过程

2　实验设计

为了验证算法的有效性,本文通过 MATLAB 编程实现本文算法,并在 Win7 系统下的 MATLAB 7.1 软件上进行实验。本文以分类准确率和时间作为评价指标,对随机分块 KNN 算法、基于 LSC 的 KNN 改进算法以及经典 KNN 算法进行对比。实验所用数据来源于 LIBSVM 和 UCI 数据集,数据集的基本信息如表 1 所示。

表 1　数据集基本信息

No	数据集	样本数	属性数	类数	No	数据集	样本数	属性数	类数
1	usps	9 298	256	10	4	letter	20 000	16	26
2	mnist	70 000	780	10	5	pendigits	10 992	16	10
3	gisette	7 000	5 000	2	6	satimage	6 435	36	6

2.1　分簇数 m 的确定

改进算法中 m 是非常重要的参数之一。为了确定参数 m,本文对随机分块 KNN 算法和本文改进算法在六个数据集上分别重复 10 次,实验结果不但报告每次实验的结果而且报告 10 次结果的均值和方差。实验结果如表 2 ~ 7 所示。

表 2　两种算法在 usps 数据集上的对比

分块数 m	评价指标	随机分块 KNN	基于聚类的 KNN
10	correct	0.9027 ±1.6498e -005	0.9355 ±7.1306e -006
	time	3.5589 ±0.0107	3.7605 ±0.0242
15	correct	0.8964 ±5.1803 -005	0.9338 ±4.1625e -006
	time	2.4857 ±0.0032	2.7260 ±0.0077
20	correct	0.8770 ±7.4889e -005	0.9300 ±4.9238e -006
	time	2.3202 ±0.0010	2.5157 ±0.01928
25	correct	0.8793 ±4.9917e -005	0.9284 ±1.0637e -005
	time	1.8586 ±0.0008	1.9971 ±0.0042
30	correct	0.8607 ±4.6629e -005	0.9275 ±1.1596e -005
	time	1.6441 ±0.0002	1.9249 ±0.0023

表 3　两种算法在 mnist 数据集上的对比

分块数 m	评价指标	随机分块 KNN	基于聚类的 KNN
10	correct	0.7221 ±4.8878e -005	0.8389 ±3.1656e -005
	time	2.9369 ±0.508	3.5504 ±0.0927
15	correct	0.6840 ±2.3333e -004	0.8364 ±2.3136e -005
	time	2.8905 ±0.0456	3.1222 ±0.01397
20	correct	0.6657 ±2.4739e -004	0.8353 ±3.3233e -005
	time	2.0564 ±0.0011	2.1490 ±0.0065
25	correct	0.6478 ±2.2689e -004	0.8338 ±8.7844e -005
	time	1.8240 ±0.0020	2.1148 ±0.0094
30	correct	0.6396 ±6.9156e -005	0.8313 ±3.8678e -005
	time	1.5457 ±0.0002	1.7274 ±0.0011

表 4　两种算法在 gisette 数据集上的对比

分块数 m	评价指标	随机分块 KNN	基于聚类的 KNN
10	correct	0.9311 ±5.0989e -005	0.9526 ±1.4711e -005
	time	23.3933 ±0.9677	28.5940 ±3.2405
15	correct	0.9252 ±1.0573e -004	0.9494 ±1.3378e -005
	time	18.0106 ±0.2434	23.1904 ±1.0894
20	correct	0.9166 ±2.8267e -005	0.9411 ±5.4699e -004
	time	12.7685 ±0.0966	16.2759 ±0.8880
25	correct	0.9150 ±7.0000e -005	0.9321 ±6.4810e -004
	time	9.9201 ±0.3696	13.8645 ±1.5093
30	correct	0.9079 ±1.0366e -004	0.9192 ±5.3796e -004
	time	8.4064 ±0.0784	11.3922 ±0.0658

表 5　两种算法在 letter 数据集上的对比

分块数 m	评价指标	随机分块 KNN	基于聚类的 KNN
10	correct	0.7892 ±3.8822e -005	0.9495 ±1.0760e -006
	time	3.2391 ±0.0015	3.2994 ±0.0010
15	correct	0.7932 ±3.7106e -005	0.9469 ±5.5751e -006
	time	3.3808 ±0.0435	3.4334 ±0.0585
20	correct	0.6815 ±1.3812e -004	0.9451 ±1.9756e -006
	time	3.0938 ±5.8392e -004	3.1285 ±3.1243e -004
25	correct	0.7279 ±5.6480e -005	0.9423 ±5.2818e -006
	time	3.3950 ±0.0018	3.4813 ±0.0054
30	correct	0.6214 ±9.8480e -005	0.9403 ±3.9204e -006
	time	3.0889 ±0.0013	3.1168 ±3.8514e -004

表 6　两种算法在 pendigits 数据集上的对比

分块数 m	评价指标	随机分块 KNN	基于聚类的 KNN
10	correct	0.9452 ±3.5382e -005	0.9721 ±4.7991e -006
	time	2.3380 ±0.0041	2.4056 ±0.0101
15	correct	0.9316 ±1.0341e -004	0.9711 ±6.0196e -006
	time	2.5451 ±0.0011	2.5709 ±0.0089
20	correct	0.9163 ±1.5515e -004	0.9700 ±2.5390e -006
	time	2.2233 ±6.4795e -005	2.2554 ±2.1569e -004
25	correct	0.9216 ±1.5677e -004	0.9687 ±3.5642e -006
	time	2.5270 ±0.0056	2.5468 ±0.0083
30	correct	0.9088 ±1.8409e -004	0.9683 ±1.5809e -006
	time	2.1805 ±7.4785e -005	2.2022 ±8.9611e -005

通过表 2 ~ 7 可知,随着分块数 m 的增加,两种算法的分类速度逐渐变快,但相应的分类准确率却逐渐下降。考虑到两种算法都属于近似算法,如果分类的准确率明显低于经典 KNN 算法,那么在大数据分类中将会出现很大的误分率,这显

然是不可取的。此外,从表 2 ~ 7 可知,在分类时间相近的情况下,基于 LSC 的 KNN 算法的分类准确率明显高于随机分块的 KNN 算法,且从表 8 中可以发现,其分类准确率更加接近经典 KNN 算法。

表 7 两种算法在 satimage 数据集上的对比

分块数 m	评价指标	随机分块 KNN	基于聚类的 KNN
10	correct	0.8603 ±8.9122e -005	0.8883 ±8.1139e -006
	time	1.2868 ±6.5495e -005	1.3027 ±1.1429e -004
15	correct	0.7917 ±3.1680e -005	0.9468 ±3.7244e -006
	time	3.8583 ±0.0332	3.9337 ±0.0152
20	correct	0.8418 ±8.8847e -005	0.8884 ±6.8028e -006
	time	1.2292 ±3.2277e -005	1.2463 ±4.3126e -005
25	correct	0.7283 ±1.1039e -004	0.9421 ±8.8449e -006
	time	3.5062 ±0.0061	3.6287 ±0.0052
30	correct	0.8312 ±3.5146e -004	0.8878 ±4.9556e -006
	time	1.2225 ±1.8176e -005	1.2396 ±1.7711e -005

依据上文分析可知,聚类分簇的个数 m 的大小同样应该是在保证所有子簇都能在内存中运行的情况下,尽可能小。这样既减小了聚类的时间,又提高了分类的速度,并且能保证具有较高的分类准确率。根据表 2 ~ 7 的实验结果显示,在 m = 10 的情况下,分类的结果更优。

2.2 k 值的确定

在算法 2 确定新的训练样本 NewX_i 后,需利用 KNN 算法对测试样本在 NewX_i 中建立分类模型,其中对于 k 值的选取是非常重要的步骤之一。考虑到经 LSC 聚类后簇内的样本已具有很高的相似性,所以 k 值的选择无须太大。为了确定 k 的取值,分别对表 1 中数据集进行了如下实验,结果如图 2 所示。为了保证实验的准确性,实验结果中每个点都是 10 次结果的均值。

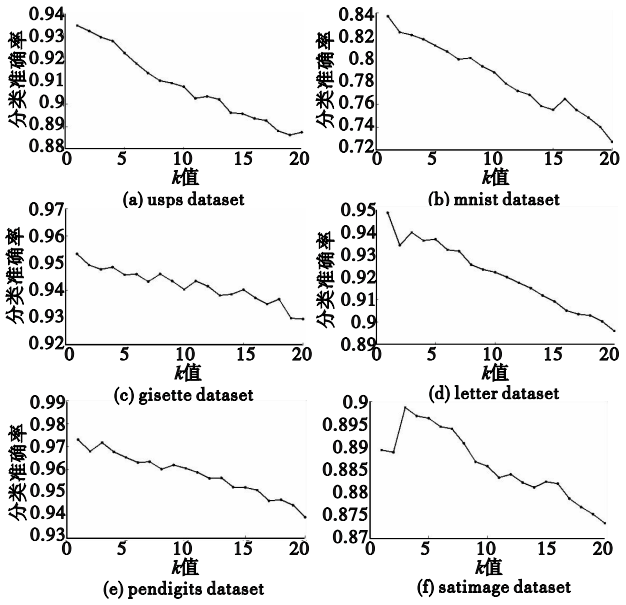

图2 六个数据集分别在不同k值时的分类准确率

从图 2 中曲线可知,随着 k 值增加,分类准确率总体呈下降趋势,这是因为簇内样本已经具有了很高的相似性,如果 k 值选取得过大,且待分类样本又属于训练集中包含数较少的类,那么在选择 k 个最近邻时,实际上并不相似的数据就会被包含进来,从而造成噪声导致分类效果降低,同时也增加了 KNN 算法的计算量。因此,在分类准确率保持较高的情况下,k 值的选取应尽可能小。根据图 2 所示,本文算法在 satimage 数据集上 k = 3 时可以取到 0.8986,高于 k = 1 时的准确率,但是在 k = 1 时,本文算法的实验结果已经优于 KNN。因此,本文统一选取 k = 1。

2.3 三种算法的性能比较

本次实验在分块(簇)数 m = 10、k = 1 的情况下,以准确率和时间为评价指标,对三种算法进行对比实验,实验结果如表 8 所示。

表 8 三种算法的性能比较

No.	随机分块 KNN		基于聚类的 KNN		经典 KNN	
	correct	time	correct	time	correct	time
1	0.902 7	3.558 9	0.935 5	3.760 5	0.948 2	32.876 4
2	0.722 1	2.936 9	0.838 9	3.550 4	0.863 5	24.157 5
3	0.931 1	23.393 3	0.952 6	28.594	0.966 0	217.332 7
4	0.789 2	3.239 1	0.949 5	3.299 4	0.951 8	19.824 6
5	0.945 2	2.338 0	0.972 1	2.405 6	0.978 0	7.298 2
6	0.860 3	1.286 8	0.888 3	1.302 7	0.906 5	3.552 5

由于改进算法为近似算法,通过表 8 可以看出,该算法分类准确率低于经典 KNN 算法 1% ~ 2.4%,但是样本数据足够大时,分类的速度接近于经典 KNN 算法的 7 ~ 9 倍(接近分块数)。而随机分块算法虽然分类速度略高于本文算法,但分类准确率较低。通过上述实验可知,使用基于 LSC 的 KNN 改进算法,能在保持分类准确率较高的情况下,大幅度提高分类的速度,使其能够在大数据中得到应用。

3 结束语

本文针对经典 KNN 算法难以在大数据样本中应用的问题,提出了一种基于 LSC 的 KNN 改进算法。该算法针对 KNN 没有训练过程的特点,创新地引入了一个训练过程,即通过聚类技术找出了远小于原始样本集的新的训练样本集,极大地减少了 KNN 算法的计算量,使其能够在大数据空间(内存)中运行,并且该算法能够保持较高的分类准确率。但是本文算法仍有一些地方需要改进,如新的训练集用于 KNN 分类时,其分类准确率的高低依赖于聚类的效果,如果聚类的效果较好,那么改进算法的分类准确率将完全有可能超过经典算法。

参考文献:

[1] Zhang Shichao. KNN-CF approach: incorporating certainty factor to KNN classification[J]. IEEE Intelligent Informatics Bulletin, 2010, 11(1): 24-33.

[2] Zhang Shichao, Zhang Chengqi, Yan Xiaowei. Post-mining: maintenance of association rules by weighting[J]. Information Systems, 2003, 28(7): 691-707.

[3] 李荣陆,胡运发. 基于密度的 KNN 文本分类器训练样本裁剪方法[J]. 计算机研究与发展, 2004, 41(4): 539-545.

[4] 张孝飞,黄河燕. 一种采用聚类技术改进的 KNN 文本分类方法[J]. 模式识别与人工智能, 2009, 22(6): 936-940.

[5] 李杨,曾海泉,刘庆华,等. 基于 KNN 的快速 Web 文档分类[J]. 小型微型计算机系统, 2004, 25(4): 725-728.

[6] Zhu Xiaofeng, Huang Zi, Yang Yang, *et al*. Self-taught dimensionality reduction on the high-dimensional small-sized data[J]. Pattern Recognition, 2013, 46(1): 215-229.

[7] Zhu Xiaofeng, Huang Zi, Cui Jiangtao, *et al*. Video-to-shot tag propagation by graph sparse group Lasso[J]. IEEE Trans on Multimedia, 2013, 15(3): 633-646.

(下转第 1023 页)

验证了该方法的有效性和可行性。本文的社会网络信任关系预测方法对推荐及可视化等具有一定的应用价值,未来将进一步挖掘用户的潜在属性特征从而丰富该模型。

参考文献:

[1] Liu Guanfeng, Yan Wang, Orgun M A. Social context-aware trust network discovery in complex contextual social networks[C]//Proc of National Conference on Artificial Intelligence. 2012:101-107.

[2] Kamvar S D, Schlosser M T, Garcia-Molina H. The eigentrust algorithm for reputation management in P2P networks[C]//Proc of the 12th International Conference on World Wide Web. New York: ACM Press,2003:640-651.

[3] Ugur K, Golbeck J. Sunny: a new algorithm for trust inference in social networks using probabilistic confidence models[C]//Proc of National Conference on Artificial Intelligence. 2007:1377-1382.

[4] Nikolay K, Thomo A. Trust prediction from user-item ratings [J]. Social Network Analysis and Mining,2013,3(3):749-759.

[5] Wanita S, Nepal S, Paris C. A survey of trust in social networks[J]. ACM Computing Surveys,2013,45(4):Article 47.

[6] Oh H K, Kim J W, Kim S W. A probability-based trust prediction model using trust-message passing[C]//Proc of the 22nd International Conference on World Wide Web Companion. 2013:161-162.

[7] Liu Haifeng, Lim E P, Lauw H W, *et al*. Predicting trusts among users of online communities: an epinions case study[C]//Proc of the 9th ACM Conference on Electronic Commerce. New York: ACM Press,2008:310-319.

[8] Kiyana Z, Aghaie A. Mining trust and distrust relationships in social Web applications[C]//Proc of IEEE International Conference on Intelligent Computer Communication and Processing. [S. l.]: IEEE Press,2010:73-78.

[9] Wang Dashun, Pedreschi D, Song Chaoming, *et al*. Human mobility, social ties, and link prediction[C]//Proc of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. New York: ACM Press,2011:1100-1108.

[10] Guha R V, Kumar R, Raghavan P, *et al*. Propagation of trust and distrust[C]//Proc of the 13th International Conference on World Wide Web. New York: ACM Press,2004:403-412.

[11] Borzymek P, Sydow M, Wierzbicki A. Enriching trust prediction model in social network with user rating similarity[C]//Proc of IEEE International Conference on Computational Aspects of Social Networks. 2009: 40-47.

[12] Xiang Rongjing, Neville J, Rogati M. Modeling relationship strength in online social networks[C]//Proc of the 19th International Conference on World Wide Web. New York: ACM Press,2010:981-990.

[13] Tang Jiliang, Gao Huiji, Hu Xia, *et al*. Exploiting homophily effect for trust prediction[C]//Proc of the 6th ACM International Conference on Web Search and Data Mining. New York: ACM Press,2013: 53-62.

[14] Zhang Yu, Tong Yu. Mining trust relationships from online social networks[J]. Journal of Computer Science and Technology,2012, 27(3):492-505.

[15] Cai Guoyong, Lyu Rui, Tang Jiliang, *et al*. Temporal dynamics in social trust prediction[J]. Wuhan University Journal of Nature Science,2014,19(5):369-378.

[16] Xin Pan, Deng Guishi, Liu Jianguo. Weighted bipartite network and personalized recommendation[J]. Physics Procedia,2010,3(5): 1867-1876.

[17] Liu Jianguo, Zhou Tao, Che Hong'an, *et al*. Effects of high-order correlations on personalized recommendations for bipartite networks [J]. Physica A: Statistical Mechanics and its Applications, 2010,389(4):881-886.

[18] Zhu Yuxiao, Lyu Linyuan. Evaluation metrics for recommender systems [J]. Journal of University of Electronic Science and Technology of China,2012,41(2):163-175.

(上接第 1006 页)

[8] Zhu Xiaofeng, Huang Zi, Cheng Hong, *et al*. Sparse hashing for fast multimedia search[J]. ACM Trans on Information Systems,2013, 31(2):9.

[9] Zhu Xiaofeng, Huang Zi, Shen Hengtao, *et al*. Dimensionality reduction by mixed kernel canonical correlation analysis[J]. Pattern Recognition, 2012, 45(8): 3003-3016.

[10] Zhu Xiaofeng, Zhang Shichao, Jin Zhi, *et al*. Missing value estimation for mixed-attribute data sets[J]. IEEE Trans on Knowledge Data Engineering,2011,23(1):110-121.

[11] Zhao Yanchang, Zhang Shichao. Generalized dimension-reduction framework for recent-biased time series analysis[J]. IEEE Trans on Knowledge and Data Engineering,2006,18(2):231-244.

[12] Qin Yongsong, Zhang Shichao, Zhu Xiaofeng, *et al*. Semi-parametric optimization for missing data imputation[J]. Applied Intelligence, 2007,27(1):79-88.

[13] Wu Xindong, Zhang Shicao. Synthesizing high-frequency rules from different data sources[J]. IEEE Trans on Knowledge and Date Engineering,2003,15(2):353-367.

[14] Zhang Tian, Ramakrishnan R, Livny M. BIRCH: an efficient data clustering method for very large databases[J]. ACM SIGMOD Record,1999,25(2):103-114.

[15] Ester M, Kriegel H P, Sander J, *et al*. A density-based algorithm for discovering clusters in large spatial databases with noise[C]//Proc of the 2nd International Conference on Knowladge Discovery and Data Mining. 1996:274-287.

[16] Ng A Y, Jordan M I, Weiss Y. On spectral clustering: analysis and an algorithm[M]//Advance in Neural Information Processing Systems. Cambridge: MIT Press,2002.

[17] Filippone M, Camastra F, Masulli F, *et al*. A survey of kernel and spectral methods for clustering[J]. Pattern Recognition,2007,41 (1):176-190.

[18] Chen Xinlei, Cai Deng. Large scale spectral clustering with landmark-based representation[C]//Proc of the 25th AAAI Conference on Artificial Intelligence. [S. l.]: AAAI,2011:313-318.

[19] Zhu Xiaofeng, Zhang Lei, Huang Zi. A sparse embedding and least variance encoding approach to hashing[J]. IEEE Trans on Image Processing,2014,23(9):3737-3750.

[20] Zhu Xiaofeng, Suk Heung-Il, Shen Dinggang. A novel matrix-similarity based loss function for joint regression and classification in AD diagnosis[J]. NeuroImage, 2014,100(10):91-105.

[21] Liu Wei, He Junfeng, Chang Shihfu. Large graph construction for scalable semi-supervised learning[C]//Proc of the 27th International Conference on Machine Learning. 2010.

[22] Wu Xindong, Zhang Chengqi, Zhang Shichao. Efficient mining of both positive and negative association rules[J]. ACM Trans on Information Systems,2004,22(3):381-405.

[23] Zhang Shichao, Qin Zhenxing, Ling C X, *et al*. Missing is useful: missing values in cost-sensitive decision trees[J]. IEEE Trans on Knowledge and Data Engineering,2005,17(12):1689-1693.

[24] Lall U, Sharma A. A nearest neighbor bootstrap for resampling hydrologic time series[J]. Water Resource Research,1996,32(3): 679-693.