基于类别平均距离的加权 KNN 分类算法①

严晓明

(福建师范大学 数学与计算机科学学院, 福州 350117)

摘 要: 本文提出了一种改进的 KNN 分类算法, 利用样本集合中同类别样本点间距离都十分接近的特点辅助 KNN 算法分类. 将待分类样本点的 K 个最近邻样本点分别求出样本点所属类别的类别平均距离和样本点与待分类样本点距离的差值比, 如果大于一个阈值, 就将该样本点从 K 个最近邻的样本点中删除, 再用此差值比对不同类别的样本点个数进行加权后执行多数投票, 来决定待分类样本点所属的类别. 改进后的 KNN 算法提高了分类的精度, 并且时间复杂度与传统 KNN 算法相当.

关键词: 类别平均距离; KNN; 加权算法

Weighted KNN Classification Algorithm Based on Mean Distance of Category

YAN Xiao-Ming

(School of Mathematics and Computer Science, Fujian Normal University, Fuzhou 350108, China)

Abstract: In this paper, an improved KNN classification algorithm is proposed by using characteristics that the points distributed in the same category of sample collection are in close distance as an assistant to classify KNN algorithm. The way to deal with the k-nearest neighboring sample points is calculating the average distance between categories that the sample points belong to and the differences of unspecified sample points respectively. If the data calculated is greater than a certain threshold, delete this sample point from k-nearest neighboring samples, then determine the categories of unspecified sample points through majority voting. The improved KNN algorithm enhances the precision of classification and maintains the same time complexity as the traditional KNN algorithm.

Key words: mean distance of category; KNN; weighted algorithm

KNN[1]算法通过待分类样本与其近邻样本点距离的多数投票结果进行分类, 以其简单, 有效的特点而被广泛地应用于数据挖掘领域. 但是传统的KNN算法是一种懒惰的学习方法, 没有预先建立模型后再进行分类, 其具有以下的缺点: 1)每个样本点都必须存储, 存储量大; 2)每次对新样本点分类时, 都要计算所有的样本点与待分类样本的距离, 计算量大; 3)参数 K 值只能由经验进行设置, 并且对某一数据集分类效果较好的取值方法对于其他数据集并没有很大的参考意义.

为了克服传统KNN算法的缺点, 许多学者从不同角度提出了改进的方法. 由于在对待分类样本点确定类别时, 需要将样本集里的每个样本分别计算与待分类样本点的距离, 因此, 减小样本点总数就能减小计算距离的时间开销, 一些学者从减小样本集里样本点总数的角度来改进KNN算法, 其中很多改进的方法都是对原始样本集以一定的方式裁剪掉某些样本点后再进行 KNN 分类, 如: Hart 的 Condensing 算法[2]、Devijver 的 MultiEdit 算法[3]等, 复旦大学的李荣陆[4]提出基于密度的KNN分类算法, 考虑了不同类别的样本点间密度的差异对分类结果的影响, 并通过对高密度类别的样本点进行裁剪以及在低密度类别的区域填充样本点的方法提高了分类的精度; 也有一些学者提出了通过对样本点进行线性变换以达到提高 KNN 分类算法精度的目的[5,6]; 崔正斌[7]等人提出用群智能方法对样本特征维数进行约简的方法来提高 KNN 算法性能, 在没有改变样本点总数的同时, 由于考虑了样

① 基金项目:福建省教育厅 B 类基金(JB11036)
收稿时间:2013-07-07;收到修改稿时间:2013-08-19

本不同特征的重要性也提高 KNN 算法的精度.

分类是利用数据集中已有标签样本的信息去给出待分类样本点的类别, 若能最大程度地保留已知标签样本的信息参与分类操作, 可以提高分类的性能. 传统的KNN算法和以裁剪样本点方式改进的KNN算法, 没有充分利用到全体已有标签的样本点的信息, 只用到了待分类样本的 K 个最近邻, 而其他带标签的样本也隐藏了对分类十分有用的信息并没有进行利用; 以样本特征约简方式改进的KNN算法, 通过对样本特征的取舍提高了分类精度, 但是样本特征的重要性对不同的待分类样本一般也是不相同的, 若对不同的待分类别样本点都对样本集中所有样本点各个维的重要性进行重新计算, 时间消耗是一个很大的问题. 本文提出了一种将样本集中同类别的样本点之间距离的平均值对KNN分类算法进行加权的方法, 使得分类结果更加可靠, 精度也得到进一步的提高.

1 样本点类别平均距离对分类结果的影响

在样本集合中, 某一类别的所有样本点, 它们的分布都呈现出了一定的规律, 即相邻样本点之间的距离一般是接近的, 这样的信息隐藏在该类别的所有样本点中, 在传统 KNN 分类时, 没有得到有效的利用. 传统的KNN算法进行分类时, 只利用到了待分类样本最近邻的 K 个带标签样本的信息. 而这些带标签的样本信息是十分宝贵的, 应用到的信息越多, 对分类结果的正面影响就越强. 前面介绍的KNN改进算法都对已知签标的样本进行了处理, 在一定程度上削弱了这些样本所包含的信息量.

以下用类别样本点平均距离表示某一类别样本点与其最近邻的同类别样本点之间距离的平均值. 如图 1 所示, 带边框的样本点的类别为B类, 与其最近邻的 5 个同类别 B 类样本点(以下划线标注)的距离平均值即为单个样本点与其同类最近邻样本点的平均距离; 把 B 类所有样本点的与其同类别的最近邻样本点的平均距离进行求平均值的结果, 即为类 B 的类别平均距离. 同一类别只有一个类别平均距离; 类别平均距离反映了同一类别相邻样本点之间距离大致接近的特点. 在图 1 中, 类 A 的样本点与其相邻的 A 类样本点的距离是接近的, 而类 B 的样本点与其相邻的 B 类样本点的距离也是接近的; 同时不同类的类别平均距离一般来说有着明显的区别, 如图 1 中, A 类、B 类的类别平均距离是有明显差别的.

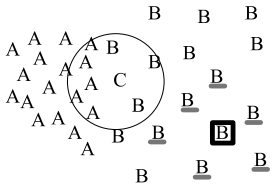

图 1 样本点类别平均距离示意图

(1) 情况 1: 当不同类别的样本分布在不同区域, 而待分类样本在这些类别区域的边缘时, 由于KNN算法取其 K 个最近邻, 这些最近邻样本点的多数可能并非待分类样本的类别, 因此在进行多数投票时, 容易出现错分. 如图 1 所示, 待分类样本 C 属于类别 B, 而此时不论 K 值取值多少, 样本点 C 的近邻中, 类别 A 的样本点总数都是占多数, 传统的KNN分类算法就会将样本点 C 的类别判断为 A. 在传统 KNN 分类时, 如果以 K 近邻样本点所属类别的类别平均距离为辅助, 就可以避免出现错分的情况. 从图 1 中可以看出, B 类的类别平均距离均值要大于类 A, 由于待分类样本点 C 与其 K 个最近邻样本中属于 B 类的 3 个样本点间的平均距离和 B 类的类别平均距离接近, 而样本点 C 与属于 A 类的 5 个样本点平均距离和 A 类的类别平均距离相差较大, 因此, 把待分类样本点 C 的类别判断为类别 B.

(2) 情况 2: 当不同类别样本的分布区域有交叉, 而待分类样本在交叉区域内时, 传统KNN算法所取的 K 个最近邻样本中, 密度大的类取出的样本点占多数, 则进行多数投票时, 也容易产生错分的情况. 如图 2 所示, 待分类样本点 C 的近邻样本点中(图中圆形所示), 类 B 的样本点有 6 个, 类 A 的样本点有 4 个, B 类的类别平均距离较小, 传统的KNN算法将待分类样本点 C 判断为 B 类. 如果以这些近邻样本点所属类别的类别平均距离为辅助, 则把待分类样本点 C 分到 A 类, 这是因为样本点 C 与其近邻的 4 个 A 类样本点的平均距离和 A 类的类别平均距离接近, 而样本点 C 与其近邻的 6 个 B 类样本点的平均距离和 B 类的类别平均距离相差较大.

(3) 情况 3: 当一个类别的样本点分布在另一个类别内, 而待分类样本点 C 在这两个类别中时, 如图 3 所示. 若按传统的KNN算法, 会将待分类样本点C判断为 A 类, 而用类别平均距离进行辅助判断, 则把待

分类样本点 C 判断为 B 类, 这是由于待分类样本点 C 的 K 个近邻中, 与类 B 样本点的平均距离和 B 类的类别平均距离更接近.

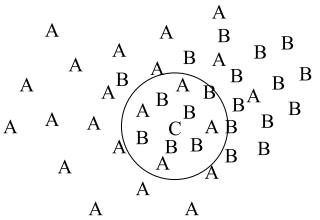

图 2　不同类别样本点分布区域交叉的情况

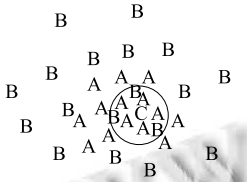

图 3　一个类别的样本点分布在另一个类中间的情况

从上面的分析可以看出, 待分类样本点的 K 个最近邻样本点所属类别的类别平均距离相差越大, 则越好判断出待分类样本点的类别.

2　基于类别平均距离的加权KNN分类器

KNN 算法没有事先建立和存储分类模型, 只在分类时才去计算待分类样本点与样本数据集中所有样本的距离. 在样本数据集中某一类别的样本点没有发生变化时, 该类别的类别平均距离保持不变, 为了避免在每次分类操作时都要去计算每一个已知类别的类别平均距离, 可设置数组对该数据进行保存. 因此在算法中共要设置两个数组, 一个保存类别名称, 另一个保存该类的类别平均距离, 数组的长度为样本集的总类别数. 当待分类样本点的类别标签确定后, 要更新这个相应数组中的类别平均距离.

计算样本集中各类别平均距离的步骤:

输入: 一个样本集共有 m 个类别, 类别标签分别为 X_i, 每个类别分别有 n_i 个样本, 其中 $1 \le i \le m$, 样本集共有 n 个样本, 每个样本的维数共有 k 个, 以上标的形式来标记 n^j, 其中 $1 \le j \le k$;

输出: 保存类别标签的数组 $array[1..m]$ 和保存类别平均距离的数组 $array\_d[1..m]$.

① 将 $array[1..m]$ 置空; $array\_d[1..m]$ 置空;

② 步骤中还要用到两个临时数组: 数组 $temp[1..n]$, 用来保存计算具体类别中每个样本点的近邻样本点, 每次计算后就置空; 另一个数组 $temp\_d[1..n]$, 用来保存 $temp[1..n]$ 数组中的样本与其最近邻样本点的平均距离. 由于各类的样本总数未知, 因此数组的大小设置为样本集里的样本点总数 n. 将两个数组分别置空, 并且另外设置一个计数器 q, 清 0.

③ 取样本集中第 1 个类别的第一个样本点 n_1, 将该样本点每一维的特征 n_1^j 都与该类别中其它所有的样本点第 j 维的值进行比较, 找出在第 j 维特征上与样本 n_1 的第 j 维特征差值最小的样本 n_{11}^j, 其中 $1 \le j \le k$, 如果这个样本不在数组 $temp[1..n]$ 中, 就将其保存到 $temp[1..n]$ 中.

④ 对数组 $temp[1..n]$ 中的每个样本, 分别求与第 1 个类别的第一个样本点 n_1 的距离, 并将差别最大的值去掉, 再求距离均值, 将其存入数组 $temp\_d[1..n]$ 中, 并将计数器 q 加 1.

⑤ 取出样本集中第 1 个类别的下一个样本, 重复步骤 3 到步骤 4, 一直到该类别的所有样本点都取完为止. 并计算 $temp\_d[1..n]$ 数组前 q 个元素的均值, 保存到 $array\_d[1..m]$ 的第一个元素的位置, 同时将类别名称 X_1 存储到数组元素 $array[1]$ 中.

⑥ 对样本集里其他类别, 重复步骤 2 到 5, 直到样本集中所有类别的类别平均距离都求出为止.

在求某个具体类别 X 中的一个样本点 n 与其周围最近邻的同类样本点的平均距离时, 一般的做法是求出样本点 n 和类别 X 中其它的样本点的距离, 再找出最小的若干个求均值, 但是这样做法的缺点是无法确定要找出多少个与样本点 n 近邻的类别 X 中的样本点. 因此, 这里采用了一个方法, 基于和样本点 n 最接近的同类样本点, 必定有若干维的特征与样本点 n 的差值最小的考虑, 步骤 3 和步骤 4 先找到与样本点 n 的每维特征的差值最小的样本点构成集合, 再求集合中的样本点和样本点 n 距离的平均值. 当新加入一个已知类别标签的样本点后, 把该类别的所有样本点执行一次计算类别平均均距离的操作.

在得到了每个类别的平均距离后, 对待分类样本进行 KNN 分类的步骤如下:

输入: 一个 m 个类别的样本集、待分类样本点、类别名数组 $array[1..m]$ 和类别平均距离数组 $array\_d[1..m]$.

输出: 待分类样本类别标签.

① 找到与待分类样本欧氏距离最小的 K 个样本点.

② 对 K 个近邻中的每个样本点 N_i, 执行以下操作: 若 N_i 与待分类样本点的距离 d_i 和 N_i 的类别平均距离 $array\_d[i]$ 的差值之比 $(d_i - array\_d[i])/array\_d[i]$ 的绝对值大于一个阈值, 则把 N_i 从 K 个最近邻中删除. 再转步骤 4.

③ 若步骤 2 中的情况不成立, 即 $\left|(d_i - array\_d[i])/array\_d[i]\right|$ 的值小于设定的阈值, 则设置 $\alpha_i = 1 + \beta * \left|(d_i - array\_d[i])/array\_d[i]\right|$, 其中 i 为类别; β 是一个整数值, 可以根据实际情况设置一个大于 1 的值, 默认设定为 5; α_i 是一个大于 1 的值, 用于对待分类样本点的 K 个近邻进行多数投票时, 对类别 i 的样本点个数进行加权.

④ 对待分类样本按筛选过后的最近邻样本点进行加权多数投票, 确定类别.

K 最近邻分类算法的时间复杂度为 $O(n^2)$, 本文算法在 KNN 算法前增加了计算每个样点点类别平均距离的步骤, 时间复杂度也为 $O(n^2)$, 因此, 算法的时间复杂度仍然为 $O(n^2)$; 本文算法所开辟的空间和样本点相关, 而不随着样本的维数增加而增加, 因此空间复杂度为 $O(n)$.

3 实验与结果分析

为了验证算法的有效性, 用 matlab2011B 下的 KNN 分类算法及本文的改进算法(以下用 KNN_Improved 算法表示), 在 UCI 公共数据集上取出 4 个数据集进行了实验; 实验环境的计算机配置: CPU 为 Core i3, 内存 1G, 操作系统为 Windows XP. 实验中的数据均在该实验环境中取得.

从 UCI 中取出的数据集分别为: Iris、Liver、Page-blocks、Glass. 其中数据集 Iris 的属性总数为 4, 有 3 个类别, 各类别的样本数为 50:50:50; 数据集 Liver 的属性总数为 6, 有 2 个类别, 各类别的样本数 145:200; 数据集 Page-blocks, 有 5 个类别, 每个类别的样本数为 4913:329:28:88:115; 数据集 Glass 的属性总数为 9 个, 共有 7 个类别, 每个类别的样本数为 70:17:76:0:13:9:29, 其中有一个类别标签为 vehicle windows, 样本数为 0, 由于没有该类别的样本, 在实验中将该类别删除, 即 Glass 数据集的类别数设为 6.

用 KNN 算法以及 KNN_Improved 算法对上述的四个数据集进行实验对比,取参数 K 值分别为 3, 5, 7; 用折数为 5 折和 10 折进行交叉验证, 由于随机地从数据集中不分类别地取出一定折数的样本, 可能会导致某些原本样本总数就很少的类别中的样本在训练集中数量过少而影响分类结果, 因此在交叉验证时, 对某一数据集中的样本按不同类别分别取出一定数量的样本后再合并, 以下实验中的交叉验证都是采用这样的方式生成测试集和训练集; 采用 F1-Measure 作为评价分类算法的指标, 即在公式 $F_\alpha = ((\alpha^2+1)*p*r)/(\alpha^2*p+r)$ 中取 $\alpha = 1$, 即认为正确率 p 和召回率 r 是一样重要的.

当数据集采用 5 折交叉验证时, KNN 算法和 KNN_Improved 算法在不同的参数 K 下各执行了 30 次后平均的 F1 值如表 1 所示:

表 1 KNN_Improved 与传统 KNN 在 5 折交叉验证时的 F1-Measure 对比

数据集	K=3		K=5		K=7	
	KNN	KNN_ Improved	KNN	KNN_ Improved	KNN	KNN_ Improved
Iris	0.933	0.926	0.940	0.933	0.933	0.933
Liver	0.630	0.630	0.641	0.641	0.682	0.667
Page-blocks	0.811	0.827	0.836	0.845	0.854	0.866
Glass	0.702	0.718	0.718	0.736	0.725	0.736

当采用 5 折交叉验证时, 此时样本集被分成 5 份, 每次取 1 份作测试集, 4 份作训练集, 而每类别中都是随机取出样本点的, 此时各类别中当作训练集的样本点分布并不一定能反映出该类别样本点的分布情况. 从表 1 中可以看出, KNN_Improved 算法的 F1 值在 5 折交叉验证时基本和 KNN 算法接近, 而在样本集中不同类别样本点数量相差较大的 Page-blocks 数据集和样本点中某个类别样本点总数较少的 Glass 数据集上, KNN_Improved 算法的性能优于传统的 KNN 算法; 可以看出 KNN_Improved 算法在类别总数较多, 且各类别的样本点总数不均衡或者某些类别样本点数量很少的情况下, 可以取得比相对于传统的 KNN 算法更好的分类效果.

当数据集采用 10 折交叉验证时, 其他条件同上, KNN 算法和 KNN_Improved 算法的平均 F1 值如表 2 所示.

表 2 KNN_Improved 与传统 KNN 在 10 折交叉验证时的 F1-Measure 对比

数据集	K=3		K=5		K=7	
	KNN	KNN_ Improved	KNN	KNN_ Improved	KNN	KNN_ Improved
Iris	0.937	0.937	0.937	0.937	0.940	0.940
Liver	0.654	0.643	0.663	0.663	0.682	0.682
Page-blocks	0.829	0.863	0.832	0.874	0.854	0.874
Glass	0.702	0.702	0.725	0.755	0.725	0.755

从表 2 中可以看出, 当采用 10 折交叉验证时, 此时每类别取出作为训练集的已知标签的样本数较多, KNN_Improved 算法可以达到比 KNN 算法更好的分类效果. 对于 Glass、Page-blocks 数据集这样的情况, KNN_Improved 算法的分类效果比对数据集 Iris、Liver 分类时提高得要明显.

4 结语

本文针对传统 KNN 算法的不足, 从已有标签的同类别样本的特点出发, 提出了一种结合 K 近邻样本点的类别平均距离对分类进行加权多数投票的方法. 在对 UCI 数据集的实验中, KNN_Improved 算法都与传统 KNN 算法的 F1 值相当或略优, 特别在 K 近邻样本中各类别的样本点总数不均衡或者某些类别样本点数量很少时, KNN_Improved 算法的优势更加明显.

参考文献

1 Cover T, Hart P. Nearest neighbor pattern classification. IEEE Trans. on Information Theory, 1967, 13: 21−27.

2 Hart P. The condensed nearest neighbor rule. IEEE Trans. on Information Theory, 1968, 14(3): 515−516.

3 Devijver P, Kittler J. Pattern Recognition: A Statistical Approach. Englewood Cliffs: PrenticeHall, 1982.

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

5 Goldberger J, Roweis S, Hinton G, Salakhutdinov R. Neighborhood components analysis. Proc. of the Advances in Neural Information Processing Systems. Vancouver. Canada, MIT Press. 2004. 512−520.

6 Torresani L, Lee K. Large margin component analysis. Proc. of the Advances in Neural Information Processing Systems. Vancouver. Canada, MIT Press. 2007. 1385−1392.

7 崔正斌,汤光明.基于遗传算法和 KNN 的软件度量属性选择研究.计算机工程与应用,2010,46(30):57−60.

(上接第 141 页)

结合的虹膜识别算法.小型微型计算机系统,2010,1846−1849.

9 Gao Y, Zheng B, Chen G., Li Q, Chen C, Chen G.. Efficient mutual nearest neighbor query processing for moving object trajectories. Information Sciences, 2010, 180: 2170−2195.

10 Liu B, Pan J, McKay RI. Entropy-based metrics in swarm clustering. International Journal of Intelligent Systems, 2009,(24): 989−1011.

11 Nie Z, Kambhampati S. A Frequency-based Approach for Mining Coverage Statistics in Data Integration. http://www.public.asu.edu/~zaiqingn/freqbased.pdf.

12 单世民,王新艳,张宪超.高维分类属性的子空间聚类算法.小型微型计算机系统,2009,30(10):2016−2021.

13 毕志升,王甲海,印鉴.基于差分演化算法的软子空间聚类.计算机学报,2012,35(10):2116−2128.

14 Tatu A, Zhang LS, Bertini E, Schreck T, Keim D, Bremm S, von Landesberger T. ClustNails: Visual analysis of subspace clusters. Tsinghua Science and Technology, 2012, 17(4): 419−428.

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

16 杨明,王飞.一种基于局部随机子空间的分类集成算法.模式识别与人工智能,2012,25(4):595−603.

17 张健飞,陈黎飞,郭躬德,李南.多代表点子空间分类算法.计算机科学与探索,2011,(11):1037−1048.

18 李南,郭躬德.基于子空间集成的概念漂移数据流分类算法.计算机系统应用,2011,20(12):241−248.

19 UCI Repository of Machine Learning Databases. http://repository.seasr.org/Datasets/UCI/arff/. 2012-12-12.