一种新的基于距离加权的模板约简 K 近邻算法

杨金福 宋 敏* 李明爱
(北京工业大学电子信息与控制工程学院 北京 100124)

摘 要：作为一种非参数的分类算法，K 近邻(KNN)算法简单有效并且易于实现。但传统的 KNN 算法认为所有的近邻样本贡献相等，这就使得算法容易受到噪声的干扰，同时对于大的数据集，KNN 的计算代价非常大。针对上述问题，该文提出了一种新的基于距离加权的模板约简 K 近邻算法(TWKNN)。利用模板约简技术，将训练集中远离分类边界的样本去掉，同时按照各个近邻与待测样本的距离为 K 个近邻赋予不同的权值，增强了算法的鲁棒性。实验结果表明，该方法可以有效地减少训练样本数目，同时还能保持传统 KNN 的分类精度。

关键词：模式识别；距离加权；模板约简；K 近邻(KNN)
中图分类号：TP181 **文献标识码**： A **文章编号**：1009-5896(2011)10-2378-06
DOI: 10.3724/SP.J.1146.2011.00051

A Novel Template Reduction K-Nearest Neighbor Classification Method Based on Weighted Distance

Yang Jin-fu Song Min Li Ming-ai
(*College of Electronic Information and Control Engineering*, *Beijing University of Technology*, *Beijing* 100124, *China*)

Abstract: As a nonparametric classification algorithm, K-Nearest Neighbor (KNN) is very efficient and can be easily realized. However, the traditional KNN suggests that the contributions of all K nearest neighbors are equal, which makes it easy to be disturbed by noises. Meanwhile, for large data sets, the computational demands for classifying patterns using KNN can be prohibitive. In this paper, a new Template reduction KNN algorithm based on Weighted distance (TWKNN) is proposed. Firstly, the points that are far away from the classification boundary are dropped by the template reduction technique. Then, in the process of classification, the K nearest neighbors' weights of the test sample are set according to the Euclidean distance metric, which can enhance the robustness of the algorithm. Experimental results show that the proposed approach effectively reduces the number of training samples while maintaining the same level of classification accuracy as the traditional KNN.

Key words: Pattern recognition; Weighted distance; Template reduction; K-Nearest Neighbor (KNN)

1 引言

K-近邻(K-Nearest Neighbor, KNN)算法[1]是最近邻算法的一个推广。基于最近邻的思想，取测试样本的K个近邻，使用“投票”原则作为分类决策，即K个近邻中，多数样本的类别就是待测样本的类别。作为一种非参数的分类算法，KNN原理简单、直观、易于实现，被广泛应用于分类、回归等模式识别领域中[2–5]。

然而，KNN算法存在两个需要解决的问题。一是如何进行K值(近邻个数)的选取。根据贝叶斯决策准则，为得到可靠的分类结果，K值应越大越好，但另一方面K个近邻样本又要与测试样本越近越好。因此，实际情况中就需要做出折中。一般方法是先确定一个初始值，然后根据实验结果不断调试，最终达到最优。很多学者对这一问题进行了研究，比较典型的有Gora等人[6]提出了一种自动选取最优K值的K近邻算法。Dudai[7]则提出了一种加权KNN方法，根据近邻样本到测试样本的距离，将较大的权值赋给较近的近邻。这样即使K值再大，对测试样本的类别判定起作用的仍然是与它较近的样本，这种加权的方法使得KNN算法对K值的选取不再敏感，增强了原算法的鲁棒性，因此得到广泛的使用。

KNN中存在的另一个问题是分类过程中，对于每一个待分类样本都要计算它与全体训练样本的距离，即需要把全部的训练样本都放入内存，对于比较大的数据集很容易造成内存不足及运算时间过长的问题。如果在分类之前，先约简掉训练集中的部分样本，同时还能保证不影响最终的分类精度，那么这一问题就得到了解决。基于这一思想，学者们提出了很多降低训练样本数目的方法，大致可以归

2011-01-18 收到，2011-06-07 改回
国家自然科学基金(61075110)和北京市自然科学基金(4082004, 4112011)资助课题
*通信作者：宋敏 songmin@emails.bjut.edu.cn

为剪辑法(editing)和压缩法(condensing)两种。对于剪辑法，是对训练样本进行处理以增强其泛化能力。具体做法是去掉那些会造成错误分类的样本或者被其它类别样本包围着的样本，例如文献[8,9]；而压缩法则是基于以下观点：处于决策边界附近的样本对于分类的正确性起着至关重要的作用，而远离决策边界的样本则对分类几乎没有影响。该方法是在不改变决策边界的前提下将远离边界的样本去掉，最终获得一个较小的训练集(即原训练集的一个子集)。对于这种约简算法，比较常用的有Hart[10]在1968年提出的压缩最近邻准则(Condensed Nearest Neighbor rule, CNN)算法，该方法可以有效降低训练集大小，但常常会保留一些远离分类边界的样本。文献[11]中给出了基于Voronoi图的压缩法，由该算法得到的压缩集(约简样本后的训练集)不仅可以将训练样本正确分类，而且根据这个压缩集可以得到与使用所有训练样本一样的分类边界。但由于引入了Voronoi图，所以算法非常复杂。文献[12]提出了递减冗余优化过程法(Decremental Reduction Optimization Procedure 1, DROP1)，并且以此为基础提出了一系列改进算法：DROP2,⋯,DROP5。其后，Wu等人[13]提出了改进的KNN算法(Improved KNN, IKNN)，该算法通过反复迭代将训练集中不能与待测样本匹配的大部分样本都约简掉，尤其适用于样本特征维数很高的情况。另外，文献[14]提出了一种模板约简KNN算法(Template Reduction for KNN, TRKNN)，该算法定义了一个最近邻链，基于这个链可以将训练集分为压缩集(通常由分类边界附近的样本组成，即新的训练集)和被约简集合(通常是内部样本)两个部分。除以上几种算法之外，文献[15,16]给出了基于其它原理的压缩算法。文献[17]提出了一种结合剪辑法和压缩法的混合模型算法。

在TRKNN的基础上，本文提出了一种基于距离加权的模板约简K近邻算法(Template reduction K-Nearest Neighbor algorithm based on Weighted distance, TWKNN)。其基本思想是首先根据TRKNN算法将训练集中远离分类边界的样本约简掉，然后按照测试样本与各个近邻的距离对K个近邻的贡献进行加权。这样不仅可以克服对于大训练集会出现的计算量过大的问题，而且还能克服K值的选取对最终分类结果的影响。既有效降低了训练集的大小，还能满足分类精度的要求。在多个数据集上的实验结果表明，该算法可以有效降低训练集的大小，同时保证分类正确率不受影响。

本文其余部分组织如下：第 2 节介绍模板约简K近邻算法；第 3 节将给出本文所提出的TWKNN的具体算法步骤；第 4 节是实验及结果分析；最后是本文结论。

2　模板约简 K 近邻算法(TRKNN)

首先我们引入最近邻链的概念。为了便于叙述，以两类问题为例进行说明：给定一个已知类别的样本集合(即训练集)，其中的某一样本 x_i (表示为 x_{i0})来自于第 1 类，则可以在属于第 2 类的样本集中找到与 x_i 距离最近的样本并将其标记为 x_{i1}，即 $x_{i1} \equiv NN(x_{i0})$。同理，在类别 1 中又可以找到 x_{i1} 的最近邻 $x_{i2} \equiv NN(x_{i1})$。以此类推，有 $x_{i,j+1} \equiv NN(x_{ij})$，直到两个样本彼此互为最近邻。这些交替来自于两个类别的最近邻序列就称之为样本 x_i 的最近邻链。

图 1 为最近邻链的生成过程示意图[14]。图中的样本属于两个类别(分别用加号和圆点表示)。从类别 1 中的样本 x_1 开始，找到其在类别 2 中的最近邻 x_{11}，它们之间的距离用 d_{10} 表示，然后再从样本 x_{11} 开始在类别 1 中寻找距离它最近的样本并标记为 x_{12}，x_{11} 与 x_{12} 之间的距离为 d_{11}，继续往下寻找类别 2 中距离 x_{12} 最近的样本，可以发现其最近邻为 x_{11}，即样本 x_{11} 与 x_{12} 互为最近邻，于是对应于 x_1 的链 C_1 终止。图中所示的其余 3 条链的形成过程同理。观察这些链可以发现，沿着链的延伸方向，越靠后的样本就越接近分类边界。这一规律将为最终的样本约简提供基础。

下面给出最近邻链的定义[14]。样本 x_i (类别为 ω_i)的最近邻链 C_i 定义为

$$\left.\begin{array}{l}\text{样本序列:}\, x_{i0}, x_{i1}, x_{i2}, \cdots, x_{ik} \\ \text{距离序列:}\, d_{i0}, d_{i1}, d_{i2}, \cdots, d_{ik}\end{array}\right\} \tag{1}$$

(1)对于样本序列有：

(a)起始样本为 $x_{i0} = x_i$。

(b)当 $x_{i,k+1} = x_{i,k-1}$ 时，链终止于 $x_{i,k}$。

(c) x_{ij} 为 $x_{i,j-1}$ 的最近邻。当 j 为偶数时，x_{ij} 的类别为 ω_i；当 j 为奇数时，x_{ij} 则属于与 ω_i 不同的类别。

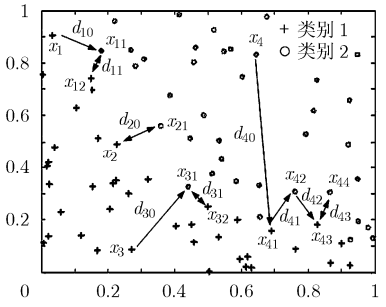

图 1 最近邻链的生成过程示意图

(2)对于距离序列有：

(a)距离 $d_{ij}=\left\|x_{i,j+1}-x_{ij}\right\|^2$ 为样本 $x_{i,j+1}$ 与 x_{ij} 之间的欧式距离。

(b)该序列为非增序列，即 $d_{ij}\geq d_{i,j+1}$。

基于上述最近邻链，可以完成样本的约简，具体步骤为(同样以两类问题为例进行说明)：

(1)求出训练集中每一个样本 x_i 所对应的最近邻链 C_i；

(2)依次对所有的最近邻链中的样本进行约简判定：如果有 $d_{ij}>\alpha\cdot d_{i,j+1}$($\alpha$ 为比例阈值：$\alpha>1$)则删除该链中的样本 x_{ij}。需要注意的是，在每一个链 C_i 中都只考虑排在第偶数位的样本，即 $j=0,2,4,\cdots$ 直到链终止，这样可以保证每个链中被约简掉的样本都来自于同一个类别。

至此，所有满足约简条件的样本都被删除掉，剩下的样本则组成新的训练集并最终用于 KNN 分类。另外，以上算法步骤中，每一步都考虑了所有的训练集样本，因此该算法不受样本出现顺序的影响。

对于最近邻链，若从某个内部点(远离边界的点)起始，那么链中该样本与其近邻之间的距离就会很大，沿着链延伸的方向，会发现这些链越来越逼近分类边界。与此同时，链中相邻样本间的距离也会不断减小并且最终呈现近似平稳的状态。这个规律为样本的约简提供了基本的依据：最近邻链中所记录的两个相邻的距离值如果出现明显的降低，则相应的样本就被认为是一个内部样本且可以删除；反之，如果比较两个距离值没有明显的下降，那么相应的样本很可能处于分类边界附近，则保留该样本。

以上算法的描述是以两类问题为例进行说明的。对于多类的情况，则需要对任意两类的训练样本都进行本算法运算，最终完成每一类样本的约简。假设训练集中的样本属于 3 个类别 $\omega_1,\omega_2,\omega_3$，则类别 ω_1 的样本要分别与类别 ω_2,ω_3 的样本进行两次约简，其余两类样本同理。需要注意的是，分别完成约简后，很可能出现对于同一个类别，两次约简后剩余的样本不完全相同的情况。假设有属于类别 ω_1 的样本 x_i，处在类别 ω_1,ω_2 边界附近，却远离 ω_1 与 ω_3 的边界，则会出现第 1 次约简中被保留，而第 2 次约简中却被删除的情况。对于这样的样本最终都保留在训练集中，即取最终约简后每一类剩余样本的并集，以最大限度地保证分类正确率不受影响。

3 基于距离加权的模板约简 K 近邻算法 (TWKNN)

设 $L=\{(\omega_s,x_i),i=1,2,\cdots,n;s=1,2,\cdots,c\}$ 是一个由 n 个样本组成的训练集，这些样本属于 c 个类别且每个样本的类别标签 ω_s 均已知。x_t 为待测样本，其类别 ω_t 待测。TWKNN 算法的具体步骤为：

(1)给定比例阈值 α ($\alpha>1$)： 阈值 α 越大，被约简掉的样本数就越少。对于有限的样本集，存在临界值 M，使得当 $\alpha\geq M$ 时，样本集中没有满足约简条件的样本，即约简后的训练集仍等于原始训练集。

给定不同的阈值 α 可以得到不同大小的新训练集。但需要注意，训练样本过少时，势必影响最终的分类正确率。α 的选取原则为在保证分类精度满足要求的前提下，尽可能多地约简训练集中的样本。具体方法是先确定一个初始值(通常选为 1.6 附近的值)，然后根据约简结果不断调试，最终达到最优。

(2)对训练集 L 进行约简：

(a)当 $c=2$ 时，即两类的分类问题。

(i)依次求出训练集 L 中的每一个样本 x_i 所对应的最近邻链 $C_i=\{x_{i0},x_{i1},\cdots,x_{il}\}$，其中 $x_{i0}=x_i,x_{ij}\in L,j=0,1,\cdots,l,l\leq n-1$。$C_i$ 中相邻样本 x_{ij} 与 $x_{i(j+1)}$ 之间的距离用 d_{ij} 表示。对于类别 ω_1 中的样本 x_1，在类别 ω_2 中找到 x_1 的最近邻样本 x_{11}，它们之间的距离为 d_{10}。继续进行，则可以得到在类别 ω_1 中 x_{11} 的最近邻样本 x_{12}，那么有 x_{11} 与 x_{12} 间的距离为 d_{11}，其余同理。

(ii)对于每一个最近邻链 C_i，观察其中相邻样本间的距离，如果有

$$d_{ij}>\alpha d_{i,j+1},\ j=\begin{cases}0,2,4,6,8,\cdots,l-3, & l\text{为奇数}\\ 0,2,4,6,8,\cdots,l-2, & l\text{为偶数}\end{cases} \tag{2}$$

则对该链中的样本 x_{ij}，进行标记。

(iii)删除原始训练集中所有被标记的样本 x_{ij}，得到新的训练集 $\tilde{L}=\{(\omega_s,x_i)\}$，其中 $i=1,2,\cdots,m;s=1,2$，显然 $m\leq n$(当 α 大于临界值 M 时，$m=n$)。

(b)当 $c>2$ 时，即多类的分类问题。 对训练集 L 中的任意两类样本，按照两类问题的算法步骤进行约简，即对于每一个类别的样本，共需约简 $c-1$ 次，得到 $c-1$ 个约简结果。然后将属于同一类别的所有约简结果取并集作为该类样本的最终结果。依次对 c 个类别的样本进行相同处理可以得到最终新的训练集 $\tilde{L}=\{(\omega_s,x_i),i=1,2,\cdots,m;\ s=1,2,\cdots,c\}$，同样 $m\leq n$。

(3)对于待测样本 x_t，在新的训练集 $\tilde{L}$ 中选出与之最近的 k 个样本： 用 $x_{t(1)},x_{t(2)},\cdots,x_{t(k)}$ 表示这 k 个近邻，相应的类别标签为 $\omega_{t(1)},\omega_{t(2)},\cdots,\omega_{t(k)}$。距离度量选用欧式距离，则待测样本 x_t 与这 k 个近邻的距离依次为 $d_{t(1)},d_{t(2)},\cdots,d_{t(k)}$。

(4)为 k 个近邻样本赋权值： 使用距离平方的倒数加权近邻样本的贡献。则有第 j 个近邻的权值为

$$w_{t(j)}=\frac{1}{(d_{t(j)})^2},\quad j=1,2,\cdots,k \tag{3}$$

(5)依据上述权值定义判别函数 G 为

$$G(\omega_s,x_t)=\sum_{j=1}^{k}w_{t(j)}I(\omega_s=\omega_{t(j)}),\ s=1,2,\cdots,c \tag{4}$$

其中$I_{t(j)}=\begin{cases}1, & \text{当}w_{t(j)}=w_s\\ 0, & \text{当}w_{t(j)}\neq w_s\end{cases}$

则最大的$G(\omega_s,x_t)$所对应的类别ω_s判定为待测样本x_t的类别，即

$$\omega_t=\arg\max_{\omega_s}G(\omega_s,x_t) \tag{5}$$

4　实验

4.1 实验数据以及环境

实验在 10 个真实的数据集上进行。数据集均来源于 UCI 数据库，其中数据集 1 来源于 Proben1 数据库中的 cancer1.dt 文件，它建立在 UCI 数据库中的“Breast Cancer Wisconsin”数据集上[18]。数据集 3 为 UCI 数据库中的“Connectionist Bench (Sonar, Mines vs. Rocks)”数据集，为了便于表述，用 Sonar 表示。数据集 5 为 UCI 数据库中的“MONK's Problems”数据集，在实验中用 Monk 表示。第 9 个数据集来源于 UCI 数据库中的“Statlog (Vehicle Silhouettes)”数据集，原数据集包含 Saab、Opel、bus 以及 van 4 个类别，本文实验中的数据集 9(即 Vehicle_1)则包含了属于 Opel、bus 和 van 这 3 个类别的数据。第 10 个数据集来源于 UCI 数据库中的“Statlog (Landsat satellite)”数据集，原数据集包含 red soil, cotton crop, grey soil, damp grey soil, soil with vegetation stubble 以及 very damp grey soil 共 6 个类别，实验中我们选取了“sat.txt” 文件中属于 red soil, grey soil, damp grey soil 这 3 个类别的数据，用 Landsat_1 代表该组数据。上述数据集的相关信息如表 1 所示。

表 1 实验数据集

数据集	样本数(个)	特征数(个)	类别数(个)
Breast Cancer	699	9	2
Pima Indians	768	8	2
Sonar	208	60	2
Statlog(Heart)	270	13	2
Monk	432	7	2
Wine	178	13	3
Balance scale	625	4	3
Iris	150	5	3
Vehicle_1	629	18	3
Landsat_1	1069	36	3

本文的实验是在 CPU 为 AMD Athlon(tm) Dual Core Processor 2.69 GHz，内存为 2 GB 的计算机上进行的，算法采用 MATLAB 7.0.4 软件编程实现。

4.2 实验方法

为了验证本文算法的有效性，在每一个数据集上我们都分别使用传统的 KNN，加权 KNN (用 WKNN 表示)，TRKNN 以及本文所提出的 TWKNN 算法进行分类。近邻评判均选用欧氏距离。

实验采用交叉验证法进行。随机抽取数据集中每一类样本的 4/5 组成训练集，其余的 1/5 作为测试集。实验共进行 20 次，取这 20 次实验结果的平均值作为该数据集的分类结果。

另外，本文对样本数据进行归一化处理采用最小—最大规范化，方法如下：对于每一个属性，找出它在数据集所有样本中对应该属性的最小值与最大值。于是，定义如下：

$$x_j^{'}=\frac{x_j-\min(x_j)}{\max(x_j)-\min(x_j)} \tag{6}$$

其中 $\max(x_j)$ 和 $\min(x_j)$ 分别是数据集中所有样本的第 j 个属性的最大值和最小值。经过归一化后，n 个数据样本的每个属性的值都被映射到[0, 1]区间。

4.3 实验结果

表 2 为使用 TWKNN 算法进行约简后各个数据集的剩余样本数。表 3 分别为使用传统的 KNN、WKNN、TRKNN 以及本文所提出的 TWKNN 算法进行分类的正确率，实验中 K 值分别取 1, 3, 5, 7, 9 五个值，括号内为对应于每种分类算法的训练样本个数。表 4 为表 3 中 4 种算法取得相应分类结果的分类时间(因为 KNN 直接计算测试样本与训练样本之间的距离，不需要进行训练，因此在实验中记录各个方法的分类时间，即测试时间), 表中时间单位均为秒。

表 2 TWKNN 算法的约简结果

数据集	原样本个数	约简后样本个数		
		比例阈值为 1.4	比例阈值为 1.6	比例阈值为 1.8
Breast Cancer	699	126	234	375
Pima Indians	768	476	584	651
Sonar	208	159	182	194
Statlog(Heart)	270	123	150	190
Monk	432	348	396	431
Wine	178	148	172	176
Balance scale	625	260	378	484
Iris	150	128	147	150
Vehicle_1	629	511	579	605
Landsat_1	1069	904	1037	1067

表 3 不同算法在各个数据集上的平均分类正确率(%)

数据集	KNN	WKNN	TRKNN	**TWKNN**
Breast Cancer	95.50(559)	95.70(559)	95.55(286)	**95.75(286)**
Pima Indians	69.88(614)	69.92(614)	69.83(516)	**69.88(516)**
Sonar	77.15(167)	81.37(167)	76.02(145)	**80.05(145)**
Statlog(Heart)	79.07(216)	80.07(216)	78.33(154)	**79.16(154)**
Monk	78.85(346)	82.97(346)	78.99(303)	**84.47(303)**
Wine	91.03(142)	92.17(142)	89.53(119)	**91.53(119)**
Balance scale	82.54(499)	82.68(499)	82.53(384)	**82.68(384)**
Iris	96.30(120)	96.33(120)	96.57(102)	**96.87(102)**
Vehicle_1	77.63(503)	80.26(503)	77.40(466)	**80.28(466)**
Landsat_1	96.85(856)	97.15(856)	96.78(721)	**97.08(721)**

表 4 不同算法在各个数据集上的平均分类时间(s)

数据集	KNN	WKNN	TRKNN	TWKNN
Breast Cancer	0.3236	0.9378	0.1347	0.3513
Pima Indians	0.3973	0.6105	0.2772	0.4257
Sonar	0.0214	0.0232	0.0191	0.0209
Statlog(Heart)	0.0332	0.0366	0.0236	0.0259
Monk	0.0994	0.1304	0.0860	0.1125
Wine	0.0163	0.0175	0.0132	0.0144
Balance scale	0.2470	0.3493	0.1746	0.2486
Iris	0.011	0.0135	0.0102	0.0109
Vehicle_1	0.3276	0.4575	0.2909	0.4181
Landsat_1	1.2322	1.9378	1.0673	1.6520

4.4 实验结果分析

由表 2 可以看到，TWKNN 算法可以有效地降低训练样本集合的大小。选用不同的比例阈值，可以得到不同的约简效果。比例阈值越小，约简掉的样本数就越多。以 Breast Cancer 数据为例，当比例阈值为 1.8 时，有大约 50%的样本被约简掉，而当比例阈值选为 1.4 时，有 80%的样本被约简。实际应用中，如果训练样本过少，分类正确率可能要受到影响，因此要根据数据的特征和实际需要选择合适的比例阈值。

由表 3 可以看到：(1)与传统的 KNN 相比，本文所提出的 TWKNN 算法使用较小的训练集就可以得到更高的分类正确率；(2)与 WKNN 相比，可以发现 TWKNN 与 WKNN 效果基本相当但训练样本数要小，而且在 Breast Cancer 等 4 个数据集上 TWKNN 的平均正确率甚至高于 WKNN，这主要是因为对于这些数据集，TWKNN 约简掉了对分类作用不大甚至起到干扰作用的样本；(3)与 TRKNN 相比，使用相同的训练样本数，TWKNN 的分类正确率明显高于 TRKNN。综合所有比较结果，本文所提出的 TWKNN 算法不仅可以有效的将训练集进行压缩，同时还能使分类正确率满足要求。

表 4 给出了不同算法的分类时间。(1)就所有数据集而言，WKNN 的分类时间整体要大于传统的 KNN；类似地，TWKNN 的分类时间也要大于不加权的 TRKNN 的时间，从算法原理就可以对此作出解释，加权算法需要为每一个近邻赋权值，而不加权的算法则直接进行分类。(2)对比 TRKNN 与 KNN、TWKNN 与 WKNN，可以看出，经过样本约简后的测试时间要远小于没有进行样本约简的算法，达到了通过样本约简来降低计算复杂度的目的。如效果最明显的 Breast Cancer 数据，经过约简后的 TRKNN, TWKNN 的测试时间分别为 KNN, WKNN 测试时间的 1/3，其它数据集上的效果没有这么显著，但均有减少。

5 结论

本文提出了一种基于距离加权的模板约简 K 近邻分类器。利用模板约简技术将训练集中远离分类边界的样本删除，即约简掉对正确分类贡献不大的样本，这样就克服了传统 K 近邻算法中需要将所有的训练样本都放入内存，容易造成内存不足和计算

量过大的问题，使得 KNN 算法不再受训练样本集大小的制约。同时根据各近邻样本与测试样本之间距离的大小，对 K 个近邻样本赋予不同的权值，使得分类算法对 K 值的选取不再敏感，提高了 K 近邻算法的鲁棒性。实验结果表明，与传统的 KNN 算法、加权 KNN 以及 TRKNN 算法相比，该算法不仅可以有效地减少训练集的样本个数，而且可以保证分类效果不受影响。

参 考 文 献

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

[2] Nasibov E and Kandemir-Cavas C. Efficiency analysis of KNN and minimum distance-based classifiers in enzyme family prediction [J]. *Computational Biology and Chemistry*, 2009, 33(6): 461–464.

[3] Zhang Rui, Jagadish H V, Dai Bing Tian, *et al*.. Optimized algorithms for predictive range and KNN queries on moving objects [J]. *Information Systems*, 2010, 35(8): 911–932.

[4] Yao Bin, Li Fei Fei, and Kumar P. K nearest neighbor queries and kNN-joins in large relational databases (almost) for free [C]. IEEE 26th International Conference on Data Engineering (ICDE), Long Beach, CA, Mar. 1–6, 2010: 4–15.

[5] Toyama J, Kudo M, and Imai H. Probably correct k-nearest neighbor search in high dimensions [J]. *Pattern Recognition*, 2010, 43(4): 1361–1372.

[6] Gora G and Wojna A. A classifier combining rule induction and k-NN method with automated selection of optimal neighborhood [C]. Proceedings of the Thirteenth European Conference on Machine Learning, Heidelberg, Springer Berlin, 2002, 2430: 111–123.

[7] Dudai S A. The distance-weighted k-nearest neighbor rule [J]. *IEEE Transactions on Systems, Man and Cybernetics*, 1976, 6(4): 325–327.

[8] Ferri F and Vidal E. Colour image segmentation and labeling through multiedit-condensing [J]. *Pattern Recognition Letters*, 1992, 13(8): 561–568.

[9] Segata N, Blanzieri E, Delany S J, *et al*.. Noise reduction for instance-based learning with a local maximal margin approach [J]. *Journal of Intelligent Information Systems*, 2010, 35(2): 301–331.

[10] Hart P E. The condensed nearest neighbor rule [J]. *IEEE Transactions on Information Theory*, 1968, IT-14(3): 515–516.

[11] Bhattacharya B K, Poulsen R S, and Toussaint G T. Application of proximity graphs to editing nearest neighbor decision rules [C]. In Proc. 16th Symp. Interface Between Comput. Sci. Statist., 1992: 97–108.

[12] Wilson D R and Martinez T R. Reduction techniques for instance-based learning algorithms [J]. *Machine Learning*, 2000, 38(3): 257–286.

[13] Wu Ying Quan, Ianakiev K, and Govindaraju V. Improved k-nearest neighbor classi cation [J]. *Pattern Recognition*, 2002, 35(10): 2311–2318.

[14] Fayed H A and Atiya A F. A novel template reduction approach for the k-nearest neighbor method [J]. *IEEE Transactions on Neural Networks*, 2009, 20(5): 890–896.

[15] Huang D and Chow T W S. Enhancing density-based data reduction using entropy [J]. *Neural Computation*, 2006, 18(2): 470–495.

[16] Paredes R and Vidal E. Learning prototypes and distances: a prototype reduction technique based on nearest neighbor error minimization [J]. *Pattern Recognition*, 2006, 39(2): 171–179.

[17] Brighton H and Mellish C. Advances in instance selection for instance-based learning algorithms [J]. *Data Mining and Knowledge Discovery*, 2002, 6(2): 153–172.

[18] Prechelt L. Proben1, a set of neural-network benchmark problems. University of Karlsruhe, Germany, 1994. http://page.mi.fu-berlin.de/prechelt/Biblio/1994-21.pdf

杨金福：　男，1977 年生，博士，副教授，研究领域为模式识别与人工智能.

宋　敏：　女，1985 年生，硕士生，研究方向为模式识别.

李明爱：　女，1966 年生，博士，副教授，研究领域为模式识别与脑机接口.