基于 SVM 的特征加权 KNN 算法*

陈振洲[1,2]，李 磊[1]，姚正安[2]
（1. 中山大学软件研究所，广东 广州 510275；
2. 中山大学数学与计算科学学院，广东 广州 510275）

摘 要：作为一种非参数的分类算法，K-近邻（KNN）算法是非常有效和容易实现的。它已经广泛应用于分类、回归和模式识别等。在应用 KNN 算法解决问题的时候，要注意两个方面的问题——样本权重和特征权重。利用 SVM 来确定特征的权重，提出了基于 SVM 的特征加权算法（FWKNN，feature weighted KNN）。实验表明，在一定的条件下，FWKNN 能够极大地提高分类准确率。

关键词：支持向量机；K-近邻算法；距离加权；特征加权
中图分类号：TP301.6 文献标识码：A 文章编号：0529-6579（2005）01-0017-04

Cover 和 Hart[1] 提出的最近邻法已经成为一种非常有效的非参数分类算法。最近邻法是一种消极（lazy）学习法——学习过程只是简单地存储已知的训练数据；当遇到新的查询样本时，一系列相似的样本被取出，并用来分类新的查询样本。

对于最优的 K 值，KNN 分类器提供了很好的分类性能。可以看出，如果 K 无限大，KNN 分类规则就变成了 Bayes 最优分类规则[2]。对 KNN 算法的一个明显的改进是对 K 个近邻的贡献加权（距离加权 KNN 算法）[3]，将较大的权值赋给较近的近邻——它对训练数据中的噪声有很好的健壮性，而且当给定足够大的训练集合时也非常有效。

应用 KNN 算法的一个实践问题是，样本的距离是根据样本的所有特征（属性）计算的。在这些特征中，有些特征与分类是强相关的；有些特征与分类是弱相关的；还有一些特征（可能是大部分）与分类不相关。这样近邻间的距离会被大量的不相关特征所支配，这钟情况也称为维度灾难（curse of dimensionality），最近邻方法对这个问题特别敏感。

本文利用新的机器学习算法 SVM 来解决 KNN 算法所面临的维度灾难问题。虽然 SVM 的提出主要是解决分类和回归问题，但是 SVM 实际上可以给出数据样本特征的相对权重，这就为我们在 KNN 算法中利用 SVM 解决维度灾难问题提供了一个很好的方法。

本文的组织如下：第 1 节主要介绍 KNN；第 2 节介绍支持向量机（SVM）；第 3 节主要研究基于 SVM 的特征加权 KNN（FWKNN）算法；第 4 节是实验结果和分析；最后是结语。

1 K 近邻（KNN）算法

KNN 算法假定所有的样本对应于 n 维空间 $\mathbf{R}^n$ 中的点，一个样本的最近邻是根据标准的欧氏距离定义的。任意的样本 $\boldsymbol{x}$ 表示为特征向量 $\boldsymbol{x}=(\boldsymbol{x}^1, \boldsymbol{x}^2, \cdots, \boldsymbol{x}^m)$，$\boldsymbol{x}^i$ 表示样本 $\boldsymbol{x}$ 的第 i 个特征的值。那么，2 个样本 x_i 和 x_j 的距离定义为 $d(x_i, x_j)$，其中：

$$d(x_i, x_j)=\sqrt{\sum_{l=1}^{m}(x_i^l-x_j^l)^2}$$

在最近邻学习中，目标函数可以是离散值（分类问题），也可以是连续值（回归问题），本文主要讨论目标函数为离散值（分类问题）的问题——f：$\mathbf{R}^n \to V$，其中 $V=\{v_1, \cdots, v_s\}$。KNN 算法的返回值 $f(x_q)$ 就是对 $f(x_q)$ 的估计，它就是距离 x_q 最近的 k 个训练样本中最普遍的 f 值，其中：

$$f(x_q) \leftarrow \arg\max_{v\in V}\sum_{i=1}^{k}\delta(v, f(x_i)),$$

$$\begin{cases}\delta(a, b)=1, \text{若 } a=b\\ \delta(a, b)=0, \text{否则}\end{cases}$$

对 KNN 算法的一个明显的改进是对 K 个近邻的贡献加权（距离加权 KNN 算法），将较大的权值赋给较近的近邻（样本距离越近，相似度越高，贡献也就越大），一般是根据与每个近邻的距离 x_q 平方的倒数加权这个距离的贡献：

* 收稿日期：2004－03－05
基金项目：广东省工业攻关计划项目（2004B10101004）
作者简介：陈振洲（1974 年生），男，博士生；E-mail：chenzhenzhou2324@sina.com

$$\hat{f}(x_q) \leftarrow \arg\max_{v \in V} \sum_{i=1}^{k} w_i \delta(v, f(x_i)),$$

$$w_i = \frac{1}{d(x_q, x_i)^2},$$

$$\begin{cases} \delta(a, b) = 1, \text{若 } a = b \\ \delta(a, b) = 0, \text{否则} \end{cases}$$

可以看出，近邻之间的距离是由样本的所有特征按相同的度量确定的，这就很可能造成维度灾难的问题（近邻间的距离会被大量的不相关特征所支配）。解决的办法是对与分类强相关的特征赋较高的权重；与分类弱相关的特征赋较低的权重；与分类不相关的特征赋 0 权重。剩下的问题就是如何定量给出特征与分类的相关度。

2 支持向量机

支持向量机（SVM）方法是建立在统计学习理论[4]的 VC 维理论和结构风险最小（SRM）原理基础上的，根据有限的样本信息在模型的复杂性和学习能力（即无错误地识别任意样本的能力）之间寻求最佳折衷，以期获得最好的推广能力。

SVM 是从线性可分情况下的最优分类面发展而来的，基本思想可用图 1 的两维情况说明。图中，实心点和空心点代表 2 类样本，$\boldsymbol{H}$ 为分类面，$\boldsymbol{H}_1$、$\boldsymbol{H}_2$ 之间的距离叫做分类间隔（margin）。所谓最优分类线就是要求分类面不但能将 2 类正确分开（训练错误率为 0），而且使分类间隔最大。分类线方程为 $w \cdot x - b = 0$，可对它进行归一化，使得对线性可分的样本集 (x_i, y_i)，$i = 1, \cdots, n$，$\boldsymbol{x} \in \mathbf{R}^d$，$y \in \{+1, -1\}$ 满足：

$$y_i[(x_i \cdot w_i) - b] - 1 \geqslant 0, i = 1, \cdots, n \quad (1)$$

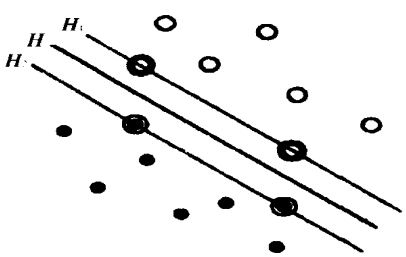

图 1 最优分类面

Fig 1 Optimal separating hyperplane

此时分类间隔等于 $2/\|w\|$，使间隔最大等价于使 $\|w\|^2$ 最小。满足条件(1)且使$\frac{1}{2}\|w\|^2$最小的分类面就叫做最优分类面，$\boldsymbol{H}_1$、$\boldsymbol{H}_2$ 上的训练样本点就称作支持向量。

对于数据为线性不可分的情况，需要引入非负的松弛因子 ξ_i 来允许错分样本的存在，这时约束（1）变成：

$$y_i[(w \cdot x_i) - b] - 1 + \xi_i \geqslant 0, i = 1, \cdots, n \quad (2)$$

而在目标则由 $\frac{1}{2}\|w\|^2$ 加入惩罚项 $C\sum_{i=1}^{n}\xi_i$ 变成：

$$\Phi(w, \xi) = \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{n}\xi_i$$

利用 Lagrange 优化方法可以把上述最优分类问题转化为其对偶问题[5]。目前对 SVM 的研究比较多，主要集中在几个方面：缩短训练时间[6,7]——主要研究如何提高 SVM 的学习效率；参数选择[8]——主要研究中一些自由参数的选择问题；应用研究[9]——主要研究 SVM 在一些具体领域的应用。本文主要研究如何利用 SVM 定量给出样本特征与分类之间的相关度，从而提出一种新的基于 SVM 的特征加权 KNN 算法。

3 基于 SVM 的特征加权 KNN 算法

按距离加权的 KNN 算法（FWKNN）是一种非常有效的归纳推理方法，它对训练数据中的噪声有很好的健壮性，而且当给定足够大的训练集合时也非常有效。但是，样本间的距离是根据样本的欧氏空间的所有坐标轴（样本的所有特征）按相同度量计算的。这里举例说明这种策略的问题：假设每个样本有 30 个特征描述，但在这些特征中只有 2 个特征与分类强相关；有 3 个特征与分类弱相关；其它 25 个特征与分类无关。在这种情况下，2 个强相关特征的值和 3 个弱相关特征的值一致的样本可能在这个 30 维样本空间中相距很远，这就造成依赖这 30 个特征的相似性度量的 KNN 算法会有很大的误差。这种由于存在许多不相关特征所导致的难题，有时也称为维度灾难，最近邻方法对这个问题特别敏感。

解决上述问题的一个有效方法是对样本的特征进行权重分析：与分类强相关的特征赋予较高的权重；与分类弱相关的特征赋予较低的权重；与分类不相关的特征赋予 0 权重。剩下的问题就是如何定量给出特征与分类的相关度。

3.1 SVM 与特征权重

首先考虑线性 SVM 的最优分类函数一般形式：

$$f(x) = \operatorname{sgn}\{w \cdot x - b\} =$$

$$\operatorname{sgn}\{\sum_{i=1}^{n} \alpha_i y_i (x_i \cdot x) - b\}$$

相应的权重向量

$$w = \sum_{i=1}^{n} y_i \alpha_i x_i, \alpha_i \geqslant 0$$

设样本集 X 有 n 个样本，任意样本 $\boldsymbol{x}_i \in \boldsymbol{X}$ 是一个 m 维行向量 $\boldsymbol{x}_i = (x_i^1, x_i^2, \cdots, x_i^m)$，即 $\boldsymbol{X}$ 是一个矩阵且 $\boldsymbol{X} \in \mathbf{R}^{m\times n}$；$\alpha_i$ 为每个样本对应的 Lagrange 乘子；求得的权值 $\boldsymbol{w}$ 为一 m 维行向量 $\boldsymbol{w} = (w^1, w^2 \cdots, w^m)$，则：

定理　若存在一个 $w^p = 0, 1 \leqslant p \leqslant m$，则训练样本集 $\boldsymbol{X}$ 的第 p 个特征与分类无关。

证明　根据上面的最优分类函数 $f(\boldsymbol{x}) = \mathrm{sgn}\{(\boldsymbol{w} \cdot \boldsymbol{x}) - b\}$，对任意样本 $\boldsymbol{x}_i$ 有：

$$f(\boldsymbol{x}_i) = \mathrm{sgn}\{(\boldsymbol{w} \cdot \boldsymbol{x}_i) - b\} =$$
$$\mathrm{sgn}\{(w^1 x_i^1 + \cdots + w^m x_i^m) - b\} =$$
$$\mathrm{sgn}\{(w^1 x_i^1 + \cdots + w^{p-1} x_i^{p-1} +$$
$$w^{p+1} x_i^{p+1} q + \cdots + w^m x_i^m) - b)\}$$

也就是说训练样本集 $\boldsymbol{X}$ 的第 p 个特征存在与否都不会影响分类函数的输出，即训练样本集 $\boldsymbol{X}$ 的第 p 个特征与分类无关。

推论　可以通过 $|w^i|(1 \leqslant i \leqslant m)$ 的值来定量给出训练样本集 $\boldsymbol{X}$ 的第 i 个特征与分类的相关性：由定理可以知道，$|w^i|$ 的值越小，训练样本集相应的第 i 个特征越不重要。

3.2　基于 SVM 的特征加权 KNN（FWKNN）算法

通过上面的分析知道，利用 SVM 可以定量确定样本的每个特征与分类的相关度——由分类函数的权重向量给出：

$$\boldsymbol{w} = (w^1, w^2, \cdots, w^m) = \sum_{i=1}^{n} y_i \alpha_i x_i,\ \alpha_i \geqslant 0$$

其中 α_i 为每个样本对应的 Lagrange 乘子。

特征权重（特征与分类的相关度）确定之后，就可以修改样本之间的距离函数以便更好地反映实际问题。任意的样本 $\boldsymbol{x}$ 表示为特征向量 $\boldsymbol{x} = (x^1, x^2, \cdots, x^m)$，$x^j$ 表示样本 $\boldsymbol{x}$ 的第 j 个特征的值，w^j 表示样本的第 i 个特征与分类的相关度。那么，2 个样本 x_i 和 x_j 的特征加权距离定义为

$$d_w(x_i, x_j) = \sqrt{\sum_{l=1}^{m} (w^l)^2 (x_i^l - x_j^l)^2}$$

w^l 为 SVM 分类函数的权重分量；

最后，对于测试样本 x_q，特征加权 KNN 算法的返回值为：

$$f(x_q) \leftarrow \arg\max_{v \in V} \sum_{i=1}^{k} c_i\, \delta(v, f(x_i)),$$
$$c_i = \frac{1}{d_w(x_q, x_i)^2}, \begin{cases} \delta(a, b) = 1, \text{若 } a = b \\ \delta(a, b) = 0, \text{否则} \end{cases}$$

在特别情况下，如果选择 k = 1，那么 1*NN* 算法就将 $f(x_i)$ 赋给 $f(x_q)$，其中 x_i 是最靠近 x_q 的训练样本。

下面给出算法的基本流程：

FWKNN（TrainData，TestSample x_q）

1：Set k；　//设置要考虑的近邻数目；

2：Decide Lagrange mulipliers α_is by training SVM with TrainData；　//训练 SVM；

3：Compute weight vector $\boldsymbol{w} = (w^1, w^2, \cdots, w^m) = \sum_{i}^{n} \alpha_i y_i x_i$；　//计算特征的权重向量；

4：Find out the k nearest neighbors of TestSample x_q by $d_w(x_i, x_j) = \sqrt{\sum_{l=1}^{m} (w^l)^2 (x_i^l - x_j^l)^2}$；　//找出最靠近要分类样本 x_q 的 k 个样本 $x_1, x_2, \cdots, x_k$；

5：Return $f(x_q) \leftarrow \arg\max_{v \in V} \sum_{i=1}^{k} c_i\, \delta(v, f(x_i))$，

$$c_i = \frac{1}{d_w(x_q, x_i)^2}, \begin{cases} \delta(a, b) = 1, \text{若 } a = b \\ \delta(a, b) = 0, \text{否则。} \end{cases}$$

4　实验结果与分析

4.1　算法与数据集

在实验中，本文主要比较算法 KNN 与算法 FWKNN 的分类性能，并且这 2 个算法都是距离加权的。利用 VC++ 6.0 实现算法，运行的平台为 256M 内存、celeron 2.0G 以及 Win2000 Server。以下是本文实验中的数据集（为了使数据特征之间具有可比性，已对数据集进行了标准化处理）：

数据集 A：该问题包含 3 个与分类线性相关的属性，7 个与分类无关的属性。训练集有 200 个样本，测试集有 1000 个样本。

数据集 B：该问题包含 3 个与分类非线性相关的属性，7 个与分类无关的属性。训练集有 200 个样本，测试集有 1000 个样本。

数据集 C：该问题包含 10 个与分类非线性相关的属性，90 个多余的属性（所谓多余属性是指该属性可以由其它的属性线性表示）。训练集有 1000 个样本，测试集有 2000 个样本。

数据集 D：UCI 上的“Breast Cancer”数据库，这是一个实际的乳癌检测问题。该问题包含 30 个特征（根据细胞核的图像提取），用来区分乳癌是良性还是恶性的。训练集有 306 个样本，测试集有 263 个样本。

数据集 E：UCI 上的“Multiple Features”数据库，这是一个手写数字识别问题，其中每个数字的数字化图像由 649 个特征表示。这里要解决的任务是将“0”和其它数字区分开来。训练集有 1 000 个样本，测试集有 1 000 个样本。

4.2 结果与分析

表 1 给出了 KNN 算法和 FWKNN 算法（对于不同的最近邻数目 k）在不同的测试集上的性能比较。可以看出，在大部分情况下 FWKNN 算法比 KNN 算法表现优异——数据集 A、C、D、E 上的测试结果足以说明问题。我们同时也看到，在数据集 B 上 FWKNN 算法表现出的性能比 KNN 算法差。也就是说，对于不同分布的数据集，FWKNN 算法表现出不同的性能——对于特征与分类线性相关的数据集（A），FWKNN 算法表现出非常好的性能；对于特征与分类非线性相关或多余特征比较多的数据集（B，C），FWKNN 算法的效果不是很好。FWKNN 算法可以很好地处理线性相关特征和无关特征，却不能很好地处理非线性相关特征和多余特征，这是我们以后准备研究的问题。

表 1 算法在测试集上的性能比较（测试准确率）

Tab. 1 Performance on test sets (accuracy) %

算法	KNN		
	$k=1$	$k=5$	$k=10$
A	82.40	87.40	91.10
B	87.70	92.50	92.50
C	59.40	61.95	62.75
D	93.16	95.82	96.58
E	99.70	99.80	99.80
算法	FWKNN		
	$k=1$	$k=5$	$k=10$
A	96.20	96.90	97.00
B	85.70	88.10	89.30
C	59.65	62.55	62.75
D	96.96	98.10	98.48
E	100	100	100

5 结 语

本文首先讨论 KNN 的基本思想及其改进算法（距离加权），然后分析了目前 KNN 算法存在的问题（curse of dimensionality），最后提出了基于线性 SVM 的特征加权 KNN 算法 FWKNN。实验结果表明，在大部分数据集（尤其是在特征与分类线性相关的数据集）上 FWKNN 算法表现出良好的效果，但是在一些存在非线性相关和多余特征的数据集上 FWKNN 算法的效果并不突出。

进一步的工作是：对于存在非线性相关的数据集，利用核函数将数据映射到特征空间，在特征空间上执行 KNN 算法；对于存在多余特征的数据集，利用特征选择的方法对特征集进行压缩，再进行 KNN 求解。

参考文献：

[1] COVER T M, HART P E. Nearest neighbor pattern classification [J]. In Trans IEEE Inform Theory, 1967, IT-13: 21-27.

[2] CHO T H, CONNERS R W, ARAMAN P A. A comparison of rule-based, K-nearest neighbor, and neural net classifiers for automation[C]. Proceedings, Developing and Managing Expert System Programs, 1991, 202-209.

[3] DUDANI S A. The distance-weighted k-nearest-neighbor rule [J]. IEEE Trans Syst Man Cyber, 1976, 6: 325-327.

[4] VAPNIK V N. The nature of statistical learning theory[M]. New York: Springer-Verlag, 1995. 张学工, 译. 统计学习理论的本质[M]. 北京: 清华大学出版社, 1999.

[5] BURGES J C. A tutorial on support vector machines for pattern recognition [M]. Bell Laboratories, Lucent Technologies, Boston, 1997.

[6] KEERTHI S S, SHEVADE S K, BHATTACHARYYA C, et al. Improvements to Platt's SMO algorithm for SVM classifier design[J]. Neural Computation, 2001, 13(3): 637-649.

[7] LIN C J. A formal analysis of stopping criteria of decomposition methods for support vector machines[J]. IEEE Transaction on Neural Networks 2002, 13(5): 1045-1052.

[8] LEE J H, LIN C J. Automatic model selection for support vector machines[EB/OL]. Available from http://www.csie.ntu.edu.tw/~cjlin/papers.html, 2000.

[9] 田盛丰, 黄厚宽. 基于支持向量机的数据库学习算法[J]. 计算机研究与发展, 2000, 37(1): 18-22.

Feature-Weighted K-Nearest Neighbor Algorithm with SVM

CHEN Zhen-zhou[1,2], *LI Lei*[1], *YAO Zheng-an*[2]

(1. Institute of Software, Sun Yat-sen University, Guangzhou 510275, China;
2. School of Mathematics and Computational Sciences, Sun Yat-sen University, Guangzhou 510275, China)

Abstract: As a nonparametric approach, K-Nearest Neighbor (K-NN) algorithm is very efficient and easily to be realized. The k-NN algorithm has been successfully used in classification, regression and pattern recognition. Two aspects, the weight of samples and the weight of features, must be paid attention to when applying K-NN to solve problems. SVM (support vector machine) to quantify the weight of features was used and FWKNN (feature weighted K-nearest neighbor algorithm with SVM) was proposed. Experiments on artificial and natural datasets FWKNN was proposed showed that, in most cases, FWKNN improve the accuracy of classification.

Key words: SVM; K-nearest neighbor; distance weighted; feature weighted