一种有效的 VSKNN 算法

于祥雨 [1,†]

[1] 杭州师范大学理学院, 杭州 *310036*

摘 要: 维规约有属性 (特征) 选择和维变换两个方面, 排除无关属性可以有效提高分类准确率. 本文主要基于属性选择提出一种有效的 VSKNN 算法, 该算法根据给定阈值确定无关属性, 再通过保留特征的数据进行 KNN 算法. 本文通过对 KNN 算法和 VSKNN 算法在 7 个数据集进行实验, 结果表明 VSKNN 算法是有效的.

关键词: KNN; 分类; 维规约

Abstract: Dimensionality conventions have two properties of feature selection and dimension transformation, excluding irrelevant attributes can effectively improve the classification accuracy. This paper mainly proposes an effective VSKNN algorithm based on attribute selection, which determine irrelevant attributes according to a given threshold, and then the KNN algorithm is carried out by preserving the attribute data. Through experiments on KNN algorithm and VSKNN algorithm in seven data sets, the results show that the VSKNN algorithm is effective.

Keywords: KNN; Classification; Victoria Statute

1 引言

随着大数据概念的兴起, 数据挖掘 [4], 机器学习 [7] 和人工智能 [6] 研究领域备受关注. 最近邻规则分类 (K-Nearest Neighbor, KNN) 算法 [1] 作为数据挖掘十大经典算法之一, 在实际问题中具有广泛的应用. 在数据挖掘方面, 虽然大数据集具备更好的挖掘潜力, 但是依然无法保证挖掘效果比小型数据集的效果好. 数据集特征数量并非越多越好, 有的特征对结果的影响很小, 还有可能存在干扰结果的特征以及特征之间存在高度的相关性, 因此合理, 适当, 科学的维规约 [5] 对分类结果有显著的影响.

† 作者简介: 于祥雨 (1989-), 男, 山东郓城人, 硕士研究生, 主要研究方向为大数据分析.

E-mail : xiangyuyu819@yeah.net

目前, 关于维规约的方法有多种. 比如: 通过相关性来剔除不相关或高度相关的特征, 主成分分析法 (Principal Component Analysis, PCA) [2] 以及局部线性嵌入 (Locally Linear Embedding, LLE) [3] 等. 不同维规约方法都有不同的适应情况, 比如: 通过相关性可以降低维数使得回归或分类效果更好, 主成分分析法主要是一种数据预处理方法, 局部线性嵌入是一种非线性降维方法, 对于具有拓扑结构的数据具有较好的效果. 本文基于同类之间数据特征高度相似, 不同类之间数据特征差异较大的思想, 提出一种同类别数据中同维度的数据趋于一致的迭代算法, 进而获得各个类对应的权向量, 将各个类的权向量构造成一个矩阵, 通过统计学原理计算其均值和标准差来确定需要剔除的特征, 对于剔除后数据集进行 KNN 算法, 即维规约 KNN 算法 (Victoria Statute KNN, VSKNN), 实验结果表明本文提出的算法在部分数据集上比传统 KNN 算法具有更好的效果.

2 预备知识

2.1 留出法

留出法 (hold-out) [8] 是将数据集 D 划分为两个互斥的集合, 分别记为训练集 S, 测试集 T, 即: $D = S \bigcup T$, $S \bigcap T = \emptyset$.

留出法在操作过程中, 需要尽量保证划分的训练集和测试集的分布一致, 这样才能避免因划分过程引入额外的偏差进而对结果产生影响.

单次采用留出法得到的估计结果不稳定 [8], 因此需要通过随机抽样的方法进行多次留出法, 以估计结果的平均值作为该算法的准确率. 另外, 训练集和测试集划分比例不同对评估结果有不同的影响. 训练集数量过大, 其预测结果往往不够稳定, 准确. 若训练集数量过少, 训练集 S 与数据集 D 之间会造成很大差别, 会严重降低预测结果的保真性 (Fidelity), 针对这一问题尚未有好的解决方案.

2.2 KNN 算法

已知类别数据集 (训练集)$T_{train} = \{(x_i, y_i)\}_{i=1}^{n}$, 对未知类别数据集 T_{test} 进行分类. 下面给出 KNN 算法 [1] 的伪代码:

算法 1 KNN 算法

输入:

训练集 T_{train} 和 k 值,

未知类别数据集 T_{test} .

输出:

未知类别数据集 T_{test} 对应的分类标签

1: **for** i **do**

2: 计算未知类别样本 $t_{test}^i \in T_{test}$ 与 T_{train} 的欧式距离 (或其他距离);

3: 将距离以升序的方式进行排序;

4: 选取前 k 个已知类别数据集的数据;

5: 统计前 k 个数据所属不同类别出现的频数;

6: 频数最大的类别的标签作为未知类别样本 t_{test}^i 的标签.

7: **end for**

8: **return**

T_{test} 数据集对应的类别标签

2.3 迭代式

定义 2.1. 给定一初始权重 $W_0 = (w_1, w_2, ..., w_m)$, $w_i > 0$, $\sum_{i=1}^m w_i = 1$, 对于第 j 类训练集子集 T^j

$$\begin{cases} T_{n+1}^j = T_n^j W_n \\ W_n = \frac{\alpha \cdot arg\ max(W_{n-1})}{||\alpha \cdot arg\ max(W_{n-1})||} \end{cases} \tag{2.1}$$

其中, $\alpha \in (0,1)$ 是一个常数, 也可以是一个关于迭代步数的递减函数, α 的取值大小影响收敛速度和效果;$||\cdot||$ 是二范数, 即每次迭代均要将 W 进行归一化.

针对以上迭代式, 需要给出合适的收敛条件, 其收敛条件

$$||max(std(T_{n+1}^j)) - min(std(T_{n+1}^j))|| < \epsilon \tag{2.2}$$

其中, $std(T_{n+1}^j)$ 为第 j 类训练集迭代 $n+1$ 次后矩阵同维度数据的标准差的 $1 \times m$ 向量, $max(std(T_{n+1}^j))$ 和 $min(std(T_{n+1}^j))$ 分别是向量中的最大值和最小值; ϵ 为允许误差, 本文取值为 1×10^{-4}.

3 VSKNN 算法

3.1 算法原理

根据定义 2.1 计算训练集中各个类对应的最佳权重, 然后根据最佳权重矩阵, 计算各个特征对应的算术平均数, 标准差. 本文通过均值累计阈值预先筛选出剔除的特征, 再结合各个维

度的标准差, 综合筛选出需要剔除的特征. 本文以 Iris 数据集为例 ($\epsilon = 1 \times 10^{-4}$, $\alpha = 0.02$, $max\_iter = 10000$), 其结果如表所示.

表 1: Iris 数据集不同类对应的权重

Label	SepalLength	SepalWidth	PetalLength	PetalWidth
setosa	0.331126	0.00662252	0.331126	0.331126
versicolor	0.00980392	0.490196	0.00980392	0.490196
virginica	0.0188679	0.0188679	0.0188679	0.943396

通过表 1 显示: 对于类别 setosa 而言, SepalLength, PetalLength 和 PetalWidth 贡献值最大, 均为 0.331126, SepalWidth 贡献值最小; 对于类别 versicolor 而言, SepalWidth 和 PetalWidth 贡献值最大, 其次是 SepalLength 和 PetalLength 一致; 对于类别 virginica 而言, PetalWidth 贡献值最大, 其次是 SepalLength, SepalWidth 和 PetalLength.

表 2: Iris 基本统计量

特征	均值	累计值 (均值)	标准差
PetalWidth	0.588239	0.588239	0.317692
SepalWidth	0.171896	0.760135	0.275724
PetalLength	0.119933	0.880067	0.182955
SepalLength	0.119933	1.000000	0.182955

通过表 2 显示: 特征 PetalWidth 的值最大, 高达 0.588239, 其标准差为 0.317692, 也是所有特征中最大的; 特征 SepalWidth 的值次之, 均值为 0.171896, 标准差为 0.275724; 特征 PetalLength 和 SepalLength 一致, 均值都为 0.119933, 标准差均为 0.182955.

根据给定的累计值阈值 $\sigma = 0.98$, (a) 由于 PetalLength 的累计值为 0.880067, SepalLength 的累计值为 1.000000, 因此初步剔除 1 个特征 SepalLength; (b) 根据初步剔除特征个数 $n = 1$, 筛选出前 n 个最小的标准差, 即为 SepalLength. 通过方法 (a) 和 (b) 求交集, 作为需要剔除的特征, 然后再对剔除特征的数据集进行 KNN 算法.

3.2 伪代码

本小节给出本文算法的伪代码, 如下所示.

算法 2 VSKNN 算法

输入:

训练集 T_{train}, 参数 α , 最大迭代步数 $iter_{max}$

误差 ϵ, 未知类别数据集 T_{test}, 初始权重 W_0, 累计阈值 σ.

输出:

未知类别数据集 T_{test} 对应的分类标签

for $j \in C$ **do**
  if $||max(std(T^j)) - min(std(T^j))|| < \epsilon$ **then**
    ϵ 值过大, 结束遍历, 调整参数 ϵ, 重新计算.
  else
    while $iter < iter_{max}$ and $||max(std(T^j_{n+1})) - min(std(T^j_{n+1}))|| > \epsilon$ **do**
      通过定义 (2.1) 求解第 j 类数据的迭代最佳随机向量 w_j 以及 b_j.
    end while
  end if
end for
存储迭代后不同类别的 $new\_W = (b, W) = \{(b_j, w_j)\}_{j=1}^{c}$, 如表 1 格式.
计算 new_W 的算术平均数, 累计值和标准差, 并以累计值排序 (升序).
根据阈值 σ, 方法 (a) 和 (b) 确定删除特征.
对训练集 T_{train} 和未知类别数据集 T_{test} 删除确定的特征, 获得新的训练集 T'_{train} 和新未知类别数据集 T'_{test}.
通过算法 1(KNN 算法) 对 T'_{test} 在训练集 T'_{train} 上进行分类.
return

T_{test} 数据集对应的类别标签

4 实验结果与分析

4.1 实验数据集

本文从 UCI 机器学习知识库 (http://archive.ics.uci.edu/ml/index.php) 获取 7 个数据集, 分别为: Iris, Balance, Wine , Glass, Segmentation, Winequality-white 以及 Winequality-red. 关于数据集的整体信息描述详见表 3. 为确保实验结果的合理性与科学性, 本文实验选择的训练集和测试集均为随机抽选, 将每个数据集样本量的 80% 作为训练集, 20% 作为测试集.

本文实验通过 Python3 软件编程实现, 运行系统:OS X EI Capitan, 处理器: 1.6GHz intel Core i5, 内存: 8GB 1600MHz DDR3.

表 3: UCI 机器学习知识库数据集和相关说明

数据集	样本数量	特征	类别数	训练集样本量	测试集样本量
Wine	178	13	3	142	36
Iris	150	4	3	120	30
Glass	214	9	6	150	64
Balance	625	4	3	500	125
Segmentation	210	19	7	168	42
Winequality-white	4898	11	7	3918	980
Winequality-red	1599	11	6	1279	320

4.2 实验结果

本小节通过留出法获取训练集和测试集, 再通过传统 KNN 算法和 VSKNN 算法进行对比. K 值取值范围为 1-20, 实验给出分类准确率最高的结果及对应的 K 值. 各个数据集的实验结果如下表所示.

表 4: 各个数据集计算结果

数据集	KNN(K)	VSKNN(K)	del_num	σ
Iris	100.00(5)	98.11(9)	1	0.98000
Wine	84.13(1)	92.06(1)	1	0.99500
Glass	66.67(1)	66.67(1)	0	0.98000
Balance	90.87(18)	79.91(12)	1	0.95000
Segmentation	79.73(1)	89.19(12)	2	0.99999
Winequality-white	55.51(1)	57.38(1)	1	0.99990
Winequality-red	53.57(1)	59.64(14)	3	0.99950
均值	75.76	77.57	–	–

通过表 4 可以看出, Iris 数据集中, 当 K = 5 时, KNN 算法的分类准确率为 100%, 高于 VSKNN 算法的 98.11%, VSKNN 算法剔除了 1 个特征, 其 3 个特征的 KNN 算法没有原始结果好; Wine 数据集中, 当 K = 1 时, KNN 算法的分类准确率为 84.13%, 低于 VSKNN 算法的 92.06%, VSKNN 算法剔除 1 个特征, 但是分类准确率有明显的提高; Glass 数据集中, KNN 算法在 K = 1 时分类准确率最高, VSKNN 算法在参数 $\sigma = 0.98000$ 下判断没有可以剔除的特征; Balance 数据集中, KNN 算法在 K = 18 时分类准确率最高, 为 90.87%, 远高于 VSKNN 算法在 K = 12 时的 79.91%; Segmentation 数据集中, KNN 算法在 K = 1 时分类准确率最高, 为 79.73%, 远低于 VSKNN 算法 (K=12) 的 89.12%, 并且剔除 2 个特征; Winequality-white

和 Winequality-red 两个数据中, VSKNN 算法略好于传统 KNN 算法, Winequality-red 数据集剔除了 3 个特征, Winequality-white 剔除了 1 个特征.

4.3 实验分析

本文通过对 7 个数据集进行实验, KNN 算法的平均分类准确率为 75.76%, VSKNN 算法的平均分类准确率为 77.57%, 整体而言, VSKNN 算法的效果并没有显著提高, 但是对 Wine 和 Segmentation 数据集有显著的提高.

本文算法相比于 KNN 算法在计算量上较大, 主要体现在无关特征的筛选上. 另外, 迭代算法的效率受参数 α 和误差 ϵ 的影响, 若 α 设定过小, 会导致收敛速度较低, 若 α 设定过大, 可能在给定的误差范围内无法收敛, 因此需要根据实际情况进行调整. 再者, 累计阈值 σ 的大小对 VSKNN 算法分类准确率有直接的影响, σ 设定较小会致使无关特征数量较多, 使得保留特征数量较小或少于分类类别数量, 进而致使分类效果较少, σ 设定较大导致无关特征数量较少, 或无剔除特征, 最终导致算法无效.

参考文献

[1] Weinberger K Q, Saul L K. Distance Metric Learning for Large Margin Nearest Neighbor Classification[J]. Journal of Machine Learning Research, 2009: 207-244.

[2] Wold S, Esbensen K, Geladi P. Principal component analysis[J]. Chemometrics and intelligent laboratory systems, 1987, 2(1-3): 37-52.

[3] Roweis S T, Saul L K. Nonlinear dimensionality reduction by locally linear embedding[J]. science, 2000, 290(5500): 2323-2326.

[4] 毛国君. 数据挖掘原理与算法 [M]. 北京: 清华大学出版社, 2005.

[5] 许明旺, 施润身. 维规约技术综述 [J]. 计算机应用, 2006, 26(10): 2401-2404.

[6] 孙伟平. 关于人工智能的价值反思 [J]. 哲学研究, 2017 (10): 120-126.

[7] 陈康, 向勇, 喻超. 大数据时代机器学习的新趋势 [J]. 电信科学, 2017, 28(12): 77-85.

[8] 周志华. 机器学习 [M]. 北京: 清华大学出版社, 2016.