A new two-layer nearest neighbor selection method for kNN classifier

摘要

kNN 分类器是一种经典的分类算法，在许多领域得到了应用。然而，kNN 分类器的性能受到简单的邻居选择方法的限制，即最近邻规则，最近邻规则只使用查询的一层邻域信息。该文章提出了一种新的基于两层邻域信息的邻域选择方法，称为两层最近邻规则。同时考虑查询的邻域和该邻域内所有选定训练实例的邻域，然后根据查询与上述邻域中所有选定训练实例的距离、分布关系和后向近邻关系确定查询的两层近邻。在论文复现过程中，发现了一种特殊情况，会导致查询节点无法得到两层近邻信息，采用 kNN 算法进行了补齐。

关键词： kNN；分类算法；最近邻规则

1 引言

请在此部分对选题背景，选题依据以及选题意义进行描述。k 近邻（k-nearest neighbor，简称 kNN）是一种经典的分类算法，因其简单、有效、直观等优点，在多个领域中均有应用。kNN 的基本思想是根据查询在训练集中的 k 近邻来确定查询的类别。具体来说，它根据多数表决法，寻找查询在 k 个近邻多对应的类别中出现频率最高的类。由于 kNN 分类器是一种非参数分类方法，因此不需要训练过程。在 k/N→0 的约束下，kNN 分类器可以渐进接近最优贝叶斯分类器达到的分类性能，其中 N 为训练实例总数 [5]。

虽然 kNN 分类器有着许多显著优点，但是它仍然存在一些问题。首先，kNN 分类器对于 k 值非常敏感。kNN 分类器需要使用固定的 k 值去对查询进行分类，然而由于查询在训练集中的空间位置不同，k 值只对一部分的查询是可以进行正确分类，对另一部分可能就很难进行正确分类。因此，kNN 分类器的性能往往取决于 k 值的选择。目前已经提出了许多的自适应方法 [1,3,6]。其次，kNN 分类器使用简单的多数表决法。在分类过程中，多数表决法同样考虑 k 近邻的作用。但是，由于这些 k 近邻与查询之间的相似度不同，因此它们的分类能力实际上是不同的。对此，各种采用基于距离加权投票的方法被应用于 kNN 分类器的性能优化上 [2]。第三，kNN 分类器通常会受到离群值的影响，特别是在小型数据集中，基于局部均值的分类器是最常见的解决办法。

除此之外，还需要考虑最近邻（nearest-neighbor，简称 NN）规则的优化，NN 规则的相似度度量过于简单，仅使用了点对点的距离来度量查询和实例之间的相似性，完全丢弃了实例的分布信息。Sanchez 等人首先提出了最近质心邻居（nearest centroid neighbor，简称 NCN）的概念，并利用这一规则设计出了 k 最近质心邻居（kNCN）分类器 [4]。其次，NN 规则的单

边相似度不够全面，他只考虑了查询的最近邻居，没有从训练实例的角度考虑查询是否也是它的最近邻居。因此，Pan 等人提出了一般近邻（General nearest neighbor，简称 GNN）的概念，要求训练实例是查询的 k 近邻，或者查询是训练实例的 k 近邻，这样的训练实例可以作为查询的一般近邻，并基于此思想开发出来 GNN 分类器。第三，受到《自然人类行为》的启发，推断可以使用 k 近邻的邻域信息来丰富查询的邻域结构，可能有助于提高 kNN 分类器的性能。

为了解决 kNN 分类器在最近邻选择过程中使用的经典 NN 规则中存在的上述问题，本文提出了一种新的邻居选择方法，称为双层最近邻 (two-layer nearest neighbor，简称 TLNN) 规则。与 NN 规则相比，TLNN 规则有三个主要优点:
(1)TLNN 规则使用了两层邻域信息。第一层是查询的邻域，称为第一层邻域，第二层是每个第一层最近邻居的邻域，称为第二层邻域。
(2) TLNN 规则考虑了查询和第二层近邻的分布关系。那些不仅离查询更近，而且分布在查询周围的第二层近邻将与第一层邻域一起构成扩展邻域。
(3) TLNN 规则约束查询与扩展近邻之间的反向近邻关系。它从扩展近邻的角度考虑查询与扩展近邻之间的相似性。对于任何扩展最近邻，如果查询在其邻域中，则将其保留为两层邻域中的两层最近邻，最终用于分类决策。基于所提出的 TLNN 规则，本文提出了 kTLNN 分类器，并根据多数表决法进行分类决策。

2 相关工作

考虑一个训练集 $T = \{y_i | y_i \in R^D\}_{i=1}^N$, 其中包含着 N 个训练实例，每个训练实例为有 D 个维度，$C = \{c_i | c_i \in \{w_1, w_2, ..., w_M\}\}_{i=1}^N$ 是每一个训练实例的类别标签，包含着 M 种类别。对于给定的查询 x,kNN 分类器首先计算查询 x 和所有实例之间的欧式距离，对于查询 x 和实例 y_i 之间的距离，由以下公式得到：

$$d(x, y_i) = \sqrt{(x - y_i)^T (x - y_i)}, \quad 1 \le i \le N \tag{1}$$

然后，根据欧式距离选出距离最小的 k 个实例，即为查询 x 的 k 近邻，表示为 $NN_k(x) = \{nn_x^i | nn_x^i \in T\}_{i=1}^k$，其中 $k \le N$，然后查询 x 的 k 最近邻标签可从 C 中得到，表示为 $C_k(x) = \{c_x^i | c_x^i \in \{w_1, w_2, ..., w_M\}\}_{i=1}^k$。最后，kNN 分类器根据多数表决法，即：

$$c_x = arg \max_{w_j} \sum_{nn_x^i \in NN_k(x)} \delta(w_j = c_x^i), \quad 1 \le i \le k, \quad 1 \le j \le M \tag{2}$$

其中 $\delta(w_j = c_x^i) = \begin{cases} 1, & w_j = c_x^i \\ 0, & w_j \neq c_x^i \end{cases}$，由此对查询 x 的 k 最近邻进行投票，出现频率最高的类别即为查询 x 的类别。从 KNN 的原理中可以发现，NN 规则基于距离越小相似度越强的原则选择了最近的 k 个近邻。因此，在传统的 kNN 分类器中，查询 x 只使用了距离它最近的 k 个实例的信息。

考虑一个例子，如图 1所示。其中，x 是来自于类别 w1 的查询，其他点为训练实例。将查询 x 的 3 最近邻点标记为 $y_1, y_2\ y_3$，即图中黑圈内的三个点。根据多数投票原则，kNN 分类器将会使查询 x 错误得分类为类别 w_2。图中的绿色圈为 $y_1, y_2\ y_3$ 的 3 最近邻域，如果将

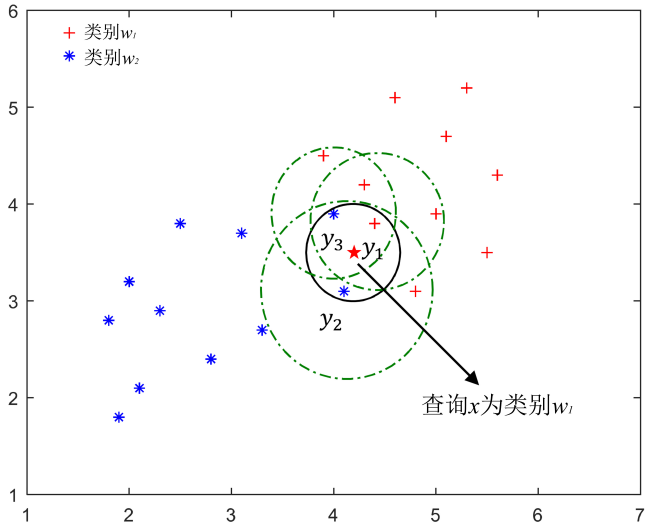

图 1. 两类别分类问题的一个例子（$k = 3$）

y_1, y_2 y_3 的 3 最近邻纳入到考虑范围内，x 的两层邻域中将会有 7 个点，其中 w_1 类有 5 个，此时根据多数表决法，x 将会被正确得分类为 w_1 类。因此，本文提出了一种新的邻居选择方法，称为两层最近邻，并提出了 kTLNN 分类器，综合考虑传统的 k 近邻和他们各自的 k 近邻（称为第二层最近邻）。同时，考虑了它们与查询之间的距离、分布关系、后向近邻关等。

3 本文方法

3.1 本文方法概述

本文提出的 TLNN 规则主要思想如下：

(1) 找到 x 的传统 k 近邻构成 x 的第一层邻域;

(2) 求出每个第一层近邻的 k 个近邻，去掉离 x 较远的 k 个近邻，其余构成 x 的第二层邻域;

(3) 去掉不分布在 x 周围的第二层近邻，剩下的第二层近邻与第一层近邻一起构成 x 的扩展邻域;

(4) 去掉那些不包含 x 的扩展近邻，剩下的扩展近邻就是构成 x 两层邻域的两层近邻。

3.2 具体规则

首先给出了查询 x 的第一层邻域定义。给定查询 x，则训练集 T 中 x 的第一层邻域由 $NN_{1st}(x)$ 给出：

$$NN_{1st}(x) = \{nn_{1st,x} | nn_{1st,x} \in NN_k(x)\} \tag{3}$$

其中，$NN_k(x)$ 是查询 x 的 k 最近邻，$nn_{1st,x}$ 是 x 的第一层最近邻。

然后，考虑 x 的每一个第一层近邻的 k 近邻来确定 x 的第二层邻域。那些在第一层近邻的 k 近邻中并且离 x 更近的训练实例构成这个第一层近邻的有效邻域，所有第一层近邻的有效邻域共同构成 x 的第二层邻域，可由下式给出：

$$NN_{2nd}(x) = \{nn_{2nd,x} | nn_{2nd,x} \in NN_{eff}(nn_{1st,x}), nn_{1st,x} \in NN_{1st,x}\} \tag{4}$$

其中 $NN_{2nd}(x)$ 为查询 x 在训练集 T 上的第二层领域，$NN_{eff}(nn_{1st,x})$ 是训练集 T 上第一层最近邻 $nn_{1st,x}$ 的有效邻域，由下式给出：

$$NN_{eff}(nn_{1st,x}) = \{nn_{eff,nn_{1st,x}} | nn_{eff,nn_{1st,x}} \in NN_k(nn_{1st,x}) \wedge d(x, nn_{eff,nn_{1st,x}}) \leq 2R_{NN_{1st}(x)}\} \quad (5)$$

$NN_k(nn_{1st,x})$ 是 $nn_{1st,x}$ 在训练集 T 上的 k 最近邻，$R_{NN_{1st}(x)}$ 是 $NN_{1st}(x)$ 的半径，这个距离来自于 x 到第 k 个第一层最近邻的距离。$nn_{2nd,x}$ 是 x 的第二层最近邻，而 $nn_{eff,nn_{1st,x}}$ 是 $nn_{1st,x}$ 的有效最近邻。

考虑 x 的第二层邻域的存在，即每个第一层最近邻的有效邻域分布，来确定 x 的扩展邻域。那些分布接近 x 的有效邻域将与第一层邻域一起构成 x 的扩展邻域。给定查询 x，用 $NN_{ext}(x)$ 表示训练集 T 中 x 的扩展邻域，由下式给出：

$$NN_{\text{ext}}(x) = \left\{ nn_{ext,x} \;\middle|\; \begin{array}{c} nn_{ext,x} \in NN_{1st}(x) \\ \vee nn_{ext,x} \in \left\{ NN_{eff}(nn_{1st,x}) \,\middle|\, d(x, cent_{NN_{eff}(nn_{1st,x})}) < d(x, nn_{1st,x}), nn_{1st,x} \in NN_{1st}(x) \right\} \end{array} \right\} \quad (6)$$

$cent_{NN_{eff}(nn_{1st,x})}$ 是第一层最近邻 $nn_{1st,x}$ 及其在中的所有有效近邻的质心。第一层邻域和 x 之间的关系很清楚，它们之间的相似性很强，所以它被保留为扩展邻域的一部分。然而，每个第一层最近邻的有效邻域与 x 之间的关系并不清楚，因此我们根据它们的分布信息度量它们之间的相似性，以确定是否将第一层最近邻的有效邻域保留为扩展邻域的一部分。用两个距离来度量第一层最近邻的有效邻域与 x 的分布关系。一个是第一层最近邻 $nn_{1st,x}$ 与 x 之间的距离 $d(x, nn_{1st,x})$，另一个是 $nn_{1st,x}$ 的有效邻域 (包括 $nn_{1st,x}$) 的质心与 x 之间的距离 $d(x, cent_{NN_{eff}(nn_{1st,x})})$。如果 $d(x, cent_{NN_{eff}(nn_{1st,x})}) < d(x, nn_{1st,x})$，即与第一层最近邻 $nn_{1st,x}$ 和 x 之间的点到点距离相比，考虑其有效邻域后，第一层最近邻 $nn_{1st,x}$ 和 x 的局部质心之间的距离变小。这说明该第一层最近邻 $nn_{1st,x}$ 与 x 的有效邻域的分布关系比自身与 x 的分布关系更接近。因此，可以合理地认为该第一层最近邻 $nn_{1st,x}$ 的有效邻域与 x 有很强的相似性，应该被保留为扩展邻域的一部分。反之，如果 $d(x, cent_{NN_{eff}(nn_{1st,x})}) \geq d(x, nn_{1st,x})$，则表示该第一层最近邻 $nn_{1st,x}$ 与 x 的有效邻域的分布关系比自身与 x 的分布关系更疏离，此时该第一层最近邻 $nn_{1st,x}$ 的有效邻域与 x 的相似度较弱，可以删除。

最后，考虑 x 的扩展邻域与 x 之间的后向近邻关系，确定最终用于分类决策的 x 的双层邻域。从每个扩展近邻的角度来看，保持 x 在其 k_b 近邻内的扩展近邻构成 x 的二层邻域。需要注意的是，这里使用的 k_b 是为了从 x 的角度将其与前向近邻关系中使用的 k 区分开来，公式如下：

$$NN_{two}(x) = \{nn_{two,x} | nn_{two,x} \in NN_{ext}(x) \wedge NN_{kb}(nn_{two,x})\} \quad (7)$$

其中，$NN_{k_b}(nn_{two,x})$ 是 $nn_{two,x}$ 在集合 $T^* = T \cup \{x\}$ 上的 k_b 最近邻，$nn_{two,x}$ 是 x 的两层最近邻。至此，我们得到查询 x 的两层近邻，这些近邻一部分来自第一层邻域，一部分来自第二层邻域，经过以上计算得到的两层最近邻与查询 x 之间具有非常强的相似性。

3.3 kTLNN 分类器

基于 TLNN 规则，提出了 k 两层最近邻（kTLNN）分类器。在 kTLNN 分类器中，使用 TLNN 规则确定查询 x 的 k 个两层最近邻，然后根据多数表决法将查询 x 分配到这 k 个两层近邻中出现频率最高的类别中。kTLNN 分类器的伪代码如算法 1 所示。

算法 1 kTLNN 分类器

输入:

x：查询

T：训练集

C：类别标签集

k：x 的第一层近邻大小和近邻的第一层最近邻大小

k_b：扩展最近邻域的邻域大小

输出:

c_x：查询 x 的分类结果

分类步骤:

步骤 1: 根据公式 (3) 获得查询 x 的 k 个第一层最近邻 $NN_{1st}(x)$;

步骤 2: 根据公式 (4-5) 获得查询 x 的第二层最近邻 $NN_{2nd}(x)$;

步骤 3: 根据公式 (6) 获得查询 x 的扩展最近邻 $NN_{ext}(x)$;

步骤 4: 根据公式 (7) 获得查询 x 的 k 个两层最近邻 $NN_{two}(x)$;

步骤 5: 使用 x 的 k 个两层最近邻 $NN_{two}(x)$ 根据如下的多数表决法确定查询 x 的类别标签 c_x:

$$c_x = arg\max_{w_j} \sum_{nn_x^i \in NN_k(x)} \delta(w_j = c_x^i), \quad 1 \le i \le k, \quad 1 \le j \le M$$

其中 $|NN_{two}(x)|$ 是集合 $NN_{two}(x)$ 中元素的数量。

4 复现细节

4.1 与已有开源代码对比

该文章没有给出源代码，代码由本人根据文章内容及 TLNN 的核心思想复现出。编程环境使用了 python 3.9 环境，代码中不涉及图像的显示，只使用了 python 自带的 random 和 math，为了展示处理进度，使用了 tqdm 模组，未使用其他模组。

4.2 创新点

如图 2所示，是我在代码复现过程中遇到的问题。当 $k = 1, k_b = 1$ 时，查询 x 的 1 最近邻为 y_1，y_1 的 1 最近邻为 y_2,y_2 到查询 x 的距离小于两倍的 $d(x, y_1)$，虽然 y_2 由于 $d(x, y_1) < d(x, cent_{y_1,y_2})$ 而被剔除，但是根据公式 (7)，由于 y_1 的 1 最近邻为 y_2，导致查询 x 的两层最近邻集合 $NN_{two}(x)$ 为空，无法判断查询 x 的类别。根据复现代码在较小的数据集上的运行情况来看，可能会因为 $NN_{two}(x) = \varnothing$ 导致代码无法继续进行而报错。因此，在代码复现中，考虑了两种方法：方法一是当 $NN_{two}(x) = \varnothing$ 时将查询 x 的类别标签置空，方法二是当 $NN_{two}(x)$ 为空集时，使用查询 x 的 k_b 个最近邻作为结果。发现当将查询 x 置空时，尤其当 k 和 k_b 较小时，错误率将异常得高，而使用查询 x 的 k_b 最近邻时，错误率接近文中的结果。在论文中，没有表明当查询 x 的 k 个两层最近邻不存在，或者说 $NN_{two}(x) = \varnothing$ 时的处理方法。但是根据文中的计算结果来看，作者进行了一定的处理，因此，我们在代码复现中，补齐了作者的工作。

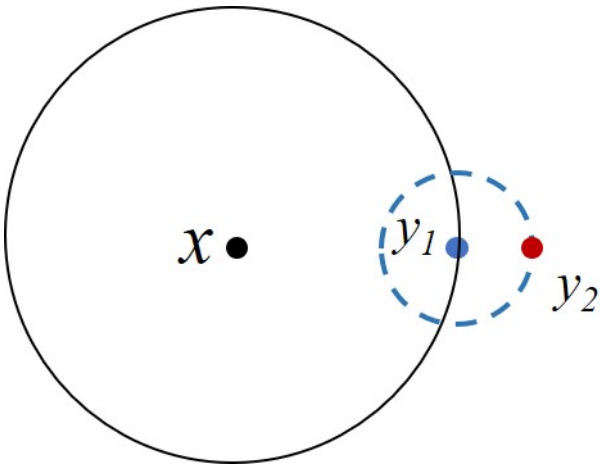

图 2. 特殊情况

4.3 实验设置

本文在复现时使用了 Dermatology、Glass 和 Ionosphere 三个数据集，三个数据集的信息如表 1所示。

表 1. 数据集的具体信息

数据集	实例数	维度	类别数
Dermatology	358	34	6
Glass	214	9	6
Ionosphere	351	34	2

在本文的所有实验中，对每个数据集使用十倍交叉验证方法来验证分类算法。也就是说，将数据集分成 10 个子集，每次进行 10 次交叉验证时，一个子集作为测试集，其余 9 个子集作为训练集，每次进行完十次交叉验证后每个实例将会获得一个标签。方法一将进行五次重复实验，方法二将进行十次重复实验，实验最终结果为多次重复实验的最佳值。

在参数设置方面，设定 $k = 1, 2, ..., 20$，$k_b = rate * k$，其中 $rate \in \{1.0, 1.2, 1.4, 1.6, 1.8, 2.0\}$，共 120 种 (k, k_b) 参数对。

5 实验结果分析

首先研究了三个数据集在两种方法下的结果表现，其结果如图 3- 5所示。从图中可以看到，当 kTLNN 分类器采用方法 2 时，错误率一直保持在较低情况下。在相同的 $rate$ 值下，当 k 较小时，方法 1 的错误率较高，尤其是当 $k = 1$ 时。当 k 值逐渐增大时，方法 1 的错误率逐渐接近方法 2 的错误率，这表明在 k 值较大的情况下，即使是实例数较小的数据集，kTLNN 算法也可以找到两层最近邻。实验结果表明，当 k 值偏小时，查询 x 存在仅通过 TLNN 规则无法找到两层最近邻的情况。

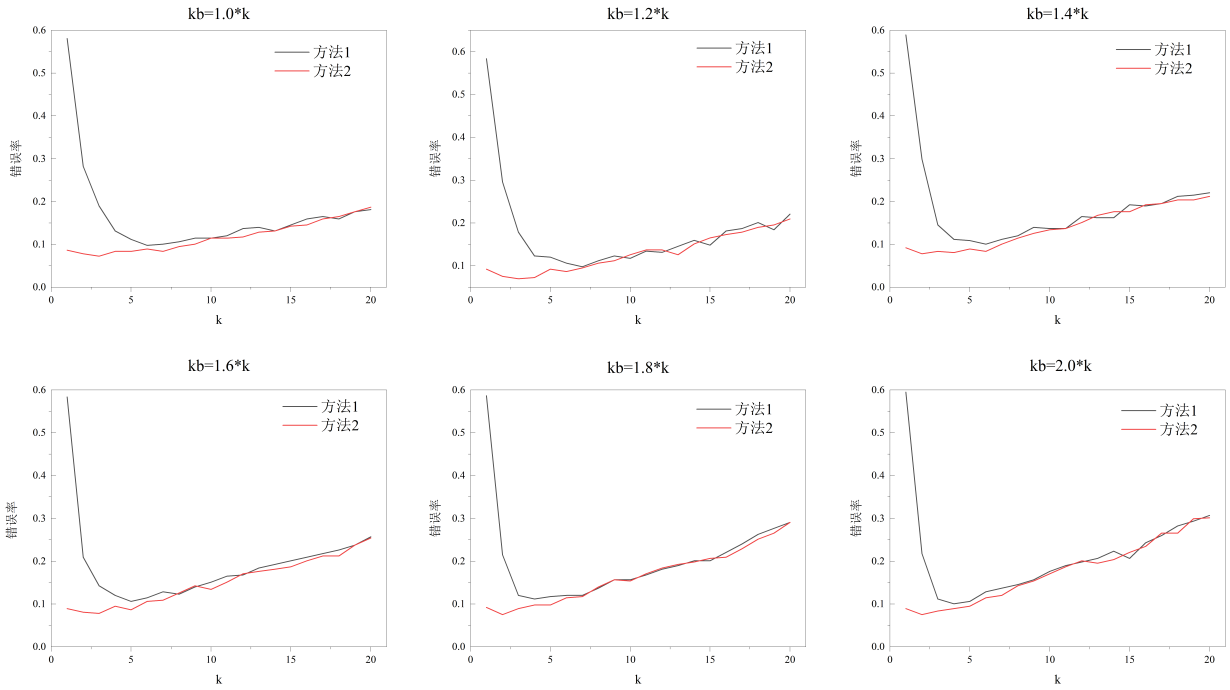

图 3. Dermatology 数据集在不同 (k, k_b) 参数对时两种方法下的错误率

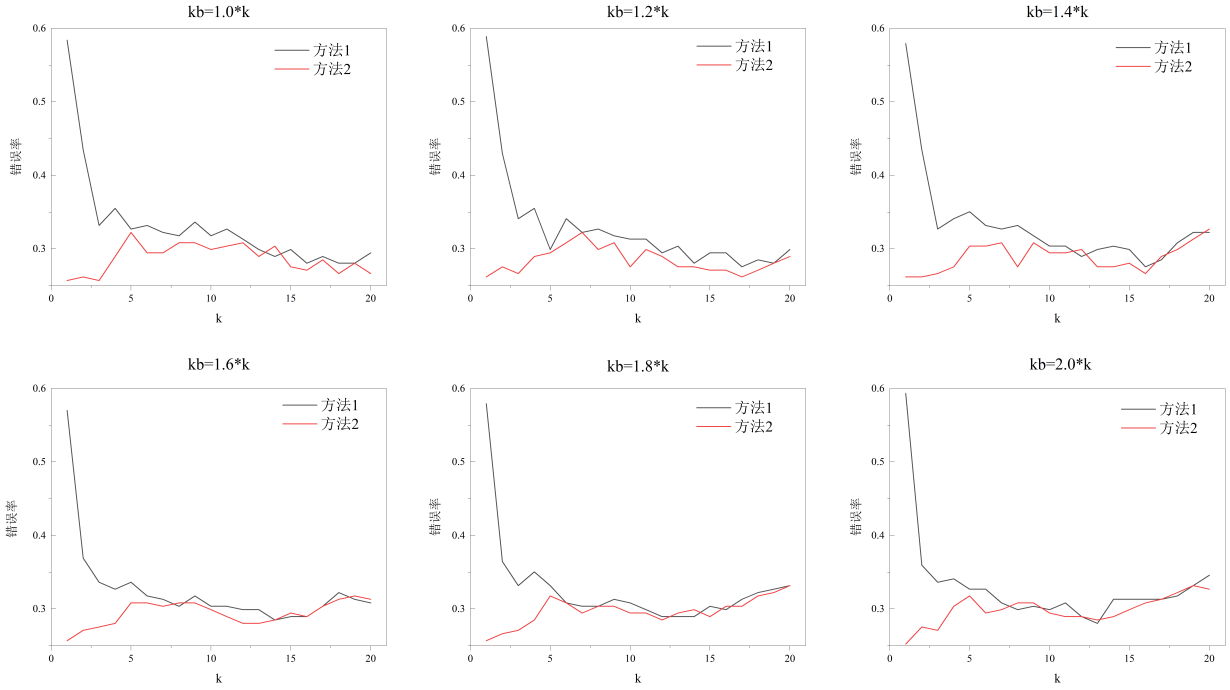

图 4. Glass 数据集在不同 (k, k_b) 参数对时两种方法下的错误率

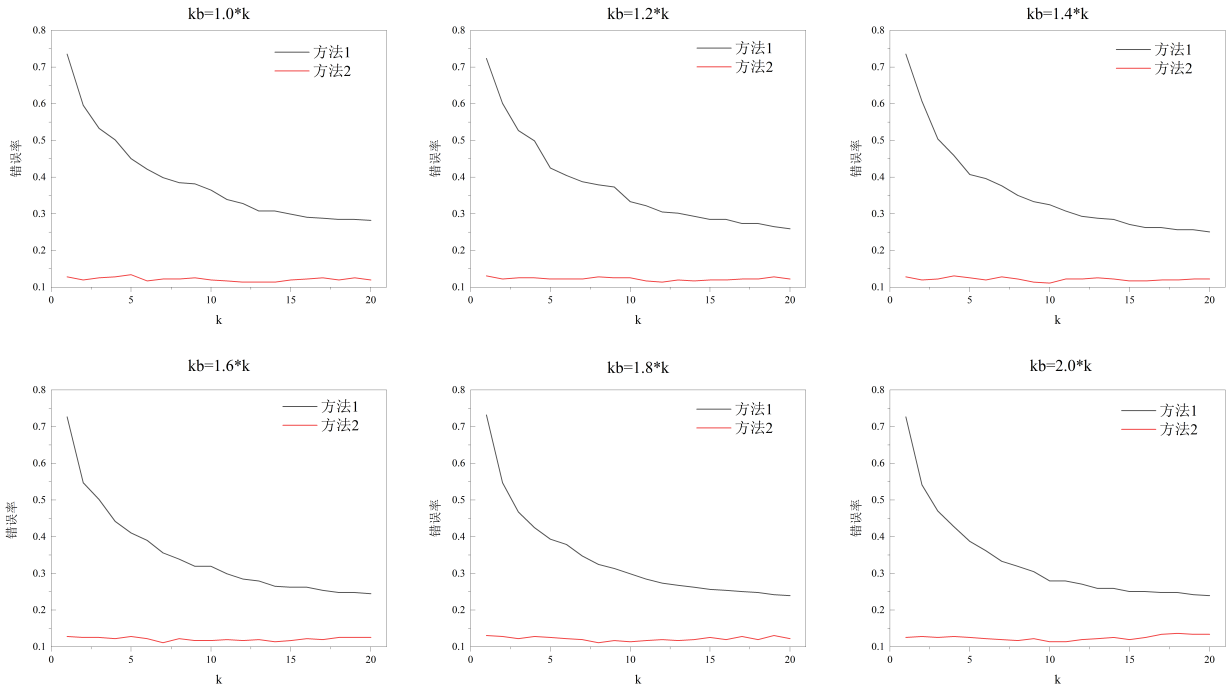

图 5. Ionosphere 数据集在不同 (k, k_b) 参数对时两种方法下的错误率

接下来比较了 kTLNN 分类器采用方法 2 时得出的分类结果和论文中给出的结果，如表 2所示, 其中括号内数字为当前 $k_b = rate * k$ 情况下，最低错误率时 k 的取值。由表 2可知，在对数据集 Dermatology 和 Glass 进行分类时，我们的错误率低于论文中给出的错误率，在对数据集 Ionosphere 进行分类时，我们的错误率和论文中的错误率接近，最大差距不高于 1%。因此, 我们较为成功的复现了原文给出的算法。同时也可以从表中看出，当不同数据集有着最低的错误率时，相应的 (k, k_b) 参数对是不同的，且没有规律，这和原文中有着一样的结论。

表 2. 与原文的实验结果对比

数据集	Dermatology		Glass		Ionosphere	
	Ours	kTLNN	Ours	kTLNN	Ours	kTLNN
$k_b = 1.0 * k$	**7.26(3)**	8.35(4)	**25.70(3)**	27.05(7)	**11.40(12,13)**	12.22(18)
$k_b = 1.2 * k$	**6.98(3)**	8.37(2)	**26.17(1,17)**	26.25(2)	11.40(12)	**11.02(18)**
$k_b = 1.4 * k$	**7.82(2)**	8.37(2)	**26.17(1,2)**	26.25(2)	11.11(10)	**10.46(19)**
$k_b = 1.6 * k$	**7.82(3)**	8.92(1)	**25.70(1)**	27.05(12)	11.11(10)	**10.19(17)**
$k_b = 1.8 * k$	**7.54(2)**	8.92(1)	**25.70(1)**	27.16(2)	11.11(10)	**10.74(15)**
$k_b = 2.0 * k$	**7.54(2)**	8.92(1)	**25.23(1)**	27.16(2)	11.40(10,11)	**10.46(15)**

6 总结与展望

本人对论文中提出的 kTLNN 分类器进行了复现，并发现了原 TLNN 规则的不足之处并提出了解决办法，通过实验结果证明了原 TLNN 规则存在的问题并表明了提出方法的有效

性。该算法在对较大的数据集进行分类时，存在花费时间较长的问题，即该方法的时间复杂度较之于 NN 规则有着明显的提升，因此未来可以通过减少访问训练集中的实例数等方法进行时间成本的降低，这是该算法未来最需要通过研究去进行改进的方向。

参考文献

[1] Nicolás García-Pedrajas, Juan A. Romero del Castillo, and Gonzalo Cerruela-García. A proposal for local k values for k -nearest neighbor rule. *IEEE Transactions on Neural Networks and Learning Systems*, 28(2):470–475, 2017.

[2] Jianping Gou, Taisong Xiong, Yin Kuang, et al. A novel weighted voting for k-nearest neighbor rule. *J. Comput.*, 6(5):833–840, 2011.

[3] Sankha Subhra Mullick, Shounak Datta, and Swagatam Das. Adaptive learning-based k -nearest neighbor classifiers with resilience to class imbalance. *IEEE Transactions on Neural Networks and Learning Systems*, 29(11):5713–5725, 2018.

[4] José Salvador Sánchez, Filiberto Pla, and Francesc J Ferri. On the use of neighbourhood-based non-parametric classifiers. *Pattern Recognition Letters*, 18(11-13):1179–1186, 1997.

[5] T. Wagner. Convergence of the nearest neighbor rule. *IEEE Transactions on Information Theory*, 17(5):566–571, 1971.

[6] Shichao Zhang, Xuelong Li, Ming Zong, Xiaofeng Zhu, and Ruili Wang. Efficient knn classification with different numbers of nearest neighbors. *IEEE Transactions on Neural Networks and Learning Systems*, 29(5):1774–1785, 2018.