基于知识线记忆的多分类器集成算法

于思皓[1),2)] 郭嘉丰[1),2)] 范意兴[1)] 兰艳艳[1),2)] 程学旗[3)]

[1)](中国科学院计算技术研究所网络数据科学与技术重点实验室 北京 100190)

[2)](中国科学院大学 北京 100190)

[3)](烟台中科网络技术研究所 山东 烟台 264005)

摘 要 多分类器系统作为混合智能系统的分支,集成了具有多样性的分类器集合,使整体得到更优的分类性能.结果融合是该领域中的一个重要问题,在相同分类器成员下,好的融合策略可以有效提升系统整体的分类正确率.随着模型安全性得到重视,传统融合策略可解释性差的问题凸显.本文基于心理学中的知识线记忆理论进行建模,参考人类决策过程,提出了一种拥有较好可解释性的启发式多分类器集成算法,称为知识线集成算法.该算法模拟人类学习与推断的行为,组织多分类器结果的融合.在训练中,模型收集给定分类器集合的不同子集,构建不同特征空间到解空间的映射,构成知识线.在推断时,模型启发式地激活知识线,进行选择性结果集成,得到推断结果.知识线集成使用样本驱动的模式,易于进行中间过程与最终结果的分析.以决策树作为分类器的实验表明,在相同的决策树集合下,知识线集成算法分类正确率与随机森林相仿.在此基础之上,知识线集成算法可量化问题不同粒度下的难易程度,且在推断时能提供相关训练样本作为依据.

关键词 多分类器;知识线记忆理论;启发式;样本驱动;可解释性

中图法分类号 TP393 **DOI 号** 10.11897/SP.J.1016.2021.00462

Multi Classifier Ensemble Algorithm Based on Knowledge-Line Memory

YU Si-Hao[1),2)] GUO Jia-Feng[1),2)] FAN Yi-Xing[1)] LAN Yan-Yan[1),2)] CHENG Xue-Qi[3)]

[1)](*Key Lab of Network Data Science and Technology, Institute of Computing Technology, Chinese Academy of Sciences, Beijing* 100190)

[2)](*University of Chinese Academy of Sciences, Beijing* 100190)

[3)](*Institute of Network Technology ICT(YANTAI) CAS, Yantai, Shandong* 264005)

Abstract Multi-classifier System, a branch technology of Hybrid Intelligent System, integrates many classifiers to approach higher accuracy. Because of the limitation of computing resource and the quality of classifiers, classifiers fusion is an important problem in Multi-classifier System. Better fusion strategy can reach higher performance of whole Multi-classifier System under the same well-trained classifier members. The traditional methods had tried many fusion strategies such as normal voting, weighted voting and fusion function. As the models developed, the classification accuracy went higher. But these models only paid attention to classification accuracy and paid little attention to interpretability which is an inevitable problem when safety of model was concerned. This paper takes a view of human decision making and presents a new multi-classifier ensemble algorithm named knowledge-line ensemble which based on knowledge-line memory theory describing the process of human decision making with memory. In order to get the

收稿日期:2019-10-10;在线发布日期:2020-09-15.本课题得到国家自然科学基金项目(61722211,61872338,61902381)、北京智源人工智能研究院(BAAI2019ZD0306)、中国科学院青年创新促进会(20144310)、国家重点研发计划(2016QY02D0405)、联想-中科院联合实验室青年科学家项目、王宽诚教育基金会、重庆市基础科学与前沿技术研究专项项目(重点)(cstc2017jcjyBX0059)和泰山学者工程专项经费(ts201511082)资助.**于思皓**,博士研究生,主要研究方向为强化学习、集成学习、自适应网络.E-mail:yusihao@ict.ac.cn.**郭嘉丰**,博士,研究员,中国计算机学会(CCF)会员,主要研究领域为信息检索、数据挖掘.**范意兴**,博士,助理研究员,主要研究方向为信息检索、自然语言处理.**兰艳艳**,博士,研究员,中国计算机学会(CCF)会员,主要研究领域为机器学习、排序学习、信息检索.**程学旗**,博士,研究员,中国计算机学会(CCF)会员,主要研究领域为网络科学、网络与信息安全、互联网搜索与数据挖掘.

interpretability like human decision making, knowledge-line ensemble algorithm imitates the learning and inference processes of human according to the psychological theory description. In training, the model tries to create memory called knowledge-line like human to store memory about solving different problems and forget memory like human in order to avoid sinking into special bad cases. Knowledge-line and training sample are one-to-one correspondence. Knowledge-line is a subset of given well-trained classifiers which can result in right classification on the corresponding sample. Different samples result in creating different knowledge-lines, so after training, the model stores varied knowledge-lines. These knowledge-lines create a set of mappings which are used to map feature space to answer space. In inference, the model chooses a subset of existing knowledge-lines to activate depending on heuristics rules. These active knowledge-lines will work, and vote to get a result. Knowledge-line ensemble algorithm is a kind of sample driven method, when inferring a new case, only the knowledge-lines born with familiar samples will be activated. It seems that human beings think of solution in memory when suffering from troubles. So knowledge-line ensemble algorithm is using sampled data to make decisions. Specially, because the process that the knowledge-line memory theory uses computing units to construct knowledge lines is similar to adding elements to sets, in order to describe the calculation process of the algorithm better, this paper uses matrices to model this process. The connection relationship between the knowledge-lines and the computing units can be represented by an adjacency matrix, the results of different classifiers can be stored by a classification matrix, and the activation of the knowledge-lines can be completed in the form of the inner product of the results of all knowledge-lines and the activation vectors. So the final classification result can be expressed in the form of matrix multiplication. On this basis, the goal and convergence of the algorithm are explained. In the experiments, this paper used decision trees as the given classifiers. Under the same given classifiers, experiments showed that knowledge-line ensemble algorithm had comparable accuracy with random forest which uses normal voting as its coordinating strategy. More importantly, knowledge-line ensemble algorithm can discriminate the difficulty of inference cases according to the active situation of knowledge-lines and give specific training cases to support the inference which makes its results more convinced.

Keywords　multi-classifier; knowledge-line memory theory; heuristics; sample driven; interpretability

1　引　言

随着大数据时代的推进，数据所蕴含的模式多元化，机器学习算法需要解决的任务愈发困难. 在多变的任务中，模型结构趋于复杂，参数量愈发庞大. 但是“没有免费的午餐”原理[1]是一个无法打破的枷锁，它论证了单个模型能力的局限性. 若要有所突破，多个模型的合作势在必行.

正如在多个器官的共同作用下，人类得以生存. 擅长不同任务的智能体合理地组成一个系统，就可以解决更多样化的问题. 混合智能系统[2]也是在这样的构想下被提出的. 在机器学习任务中，分类问题与回归问题是重要的基础问题. 针对分类问题，多分类器系统作为混合智能系统的分支在文献[3]中被提出. 多分类器系统重点在于采用“分而治之”的理念. 它将复杂的分类问题分解成多个简单的子问题，分别使用单模型逐个击破后，再合理地将这些模型组合以得到原问题的解决方案.

如今，多分类器集成算法在各种任务中扮演着重要角色，也是机器学习竞赛中提升成绩的重要手段. 但是在金融、安全等任务上，仅有分类正确率是不够的，即使模型在测试集上的正确率达到 100%，模型也依旧具有极大的可能在新的样本上给出荒谬的结果. 原因是，仅靠类似正确率的一个指标，只能做出现实世界中大多数任务的不完整描述[4]. 模型做出决策的原因是不能忽略的.

现有的多分类器系统所使用的集成策略，在推

断时无法给出做决策的具体原因，无法像 K 近邻[5]、协同过滤[6]等模型一样显式的给出推断时起作用的训练样本.事实上，在心理学的研究中，知识线记忆理论[7]说明了人在决策时会激活过往数据产生的记忆，用旧例子作为依据来推测新问题的答案.本文的贡献主要有以下几点：

(1)本文用矩阵对知识线记忆理论的计算框架进行了数学建模.

(2)本文结合心理学中的知识线记忆理论提出了一种新的多分类器集成策略，称为知识线集成算法.该算法具有良好可解释性，且分类正确率与现有集成分类算法保持在同一水平.

(3)该算法为使用者提供了丰富、简单的模型分析手段.可以量化类别推断难度，估计类别、样本之间产生混淆的概率.

2 背景介绍与相关工作

本文根据心理学中的知识线记忆理论，设计了一套启发式多分类器集成算法，本节将介绍多分类器系统的相关工作(参考文献[8-9])与知识线记忆理论的背景知识.

2.1 多分类器系统

多分类器系统是混合智能系统中的一个重要分支，旨在集成多个模型解决分类问题.它的拓扑结构有两种：链式结构与分布式结构.

链式结构如图 1 所示，所有分类器成员有序排列，数据从前到后逐个经过每个分类器.分类器成员在训练中逐个产生，每个新成员是在给定已有分类器与当前集成结果的条件下得到的.链式结构主要有两种运行模式.第一种为数据传递型[10-16].前置分类器接收到数据时，计算得到推断的结果并评估此结果的可信程度.若可信度不足，则把数据发送给后续的分类器，直到有分类器给出可信结果.这种方式有着明显的弊端：分类器成员的数量难以控制、可信度难以评估，被拒绝的结果对后续分类器作用有限.因此，第二种模式，合作型，也就是 Boosting[17-20]应运而生.每个分类器不再讨论难以评估的结果可信度，而是直接使用监督学习的方式找出推断错误的训练样本；分类器不再逐条数据进行训练，而是面向整个数据集，根据前置分类器的表现调整数据分布；推断结果由所有分类器的加权和得到，而不是完全由最后的分类器决定.链式结构下，分类器之间必然会产生较大相关性，而本文主要研究独立的分类器集成方法，所以此处不再对链式结构相关方法的发展进行更深入地讨论.

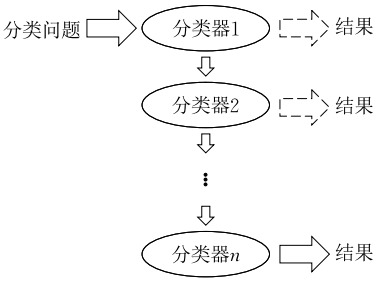

图 1 链式结构多分类器系统

分布式的结构如图 2 所示，它要求分类器成员输入的数据相同，结果独立，且分类器群体具有多样性.文献[21]从统计学出发，论证了无穷个无偏、独立分类器的结果均值与最佳贝叶斯分类器效果一致.它说明了独立的多个分类器，使用“少数服从多数”的投票策略进行决策是一种多分类器结果融合的有效思路.它对分布式多分类器系统的发展有着指导意义.分布式多分类器系统的设计主要是解决两个问题：其一，如何得到具有多样性且独立性较高的分类器集合；其二，如何将多个分类器的结果融合成一个结果.

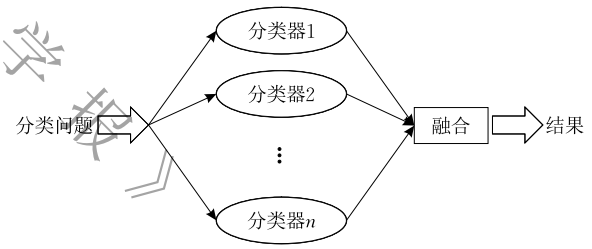

图 2 分布式结构多分类器系统

针对第一个问题，解决方案可以分为两类，数据采样与模型多样化.数据采样包括样本的随机采样，比如 Bagging[22]；特征的随机采样，比如随机森林[23]在决策树上的尝试，文献[24]在线性分类器上的尝试，文献[25]在最小距离分类器上的尝试；数据特征空间分割，比如文献[26]中所提出的模型；数据特征子集的随机投影，比如 Attribute Bagging[27]；数据标签的形式修改，比如文献[28]将标签改成多次一对多的二分类形式.模型多样化指模型在训练过程中，模型受到干预导致的多样化，比如使用不同初始化的神经网络，部分节点随机分裂的决策树等.

针对第二个问题，主要有三种解决方案：标签融合、函数融合和训练融合.标签融合是指多个分类器结果按照一定的规则合成一个结果.在文献[21]的基础上，可以证明若每个分类器成员的正确率大于

随机分类的正确率，则整体投票结果的正确率将比分类器成员正确率均值高. 可见，“少数服从多数”的结果投票是简单有效的方法，除此之外，文献[29]认为不同的分类器应有不同的重要性，所以提出了带权重的投票来组合分类器结果. 文献[30-32]利用特征信息来辅助完成结果融合. 而函数融合是把每个分类器得到的分数融合成最终结果，比如文献[33-35]使用 SoftMax 函数把多个分类器的结果重构成最终结果的后验概率，文献[36]构建结果的最优投影得到统一的结果. 以上方法都基于人为设定的规则，其实融合结果的函数也可以通过机器学习得到，即训练融合. 它可以使用决策树[37]、感知机[38]、进化算法[39]、数据包围分析[40]学习权重；使用强化学习[41]、启发式搜索[42]剪枝；使用 Stacking[43]把结果作为输入再次训练，或者将所有分类器的结果作为特征输入到一个融合分类器中进行训练，比如神经网络[44]、贝叶斯分类器[45]，来得到一个组合多分类器结果的模型. 而本文提出的方法是一种更具有可解释性的启发式剪枝方法.

多分类器系统是重要、前沿的方法，它的应用十分广泛，比如在遥感上的土地覆盖制图[46]、变化检测[47]、计算机安全上的手机通讯[48]、网络安全[49]，银行中的欺诈检测[50]、经济风险评估[51]，医药中的蛋白质折叠检测[52]、神经科学[53]以及推荐系统[54-55]等. 在众多机器学习竞赛中，集成学习、模型融合也是提高指标的重要手段. 目前的集成方法虽然能提供良好的分类性能，但同样重要的模型可解释性却都有所欠缺. 而在上述提到的众多应用中，尤其是与安全和风险有关的应用，模型的可解释性往往是更重要的需求. 因此本文从心理学中的知识线记忆理论出发，设计了一个具有良好可解释性的启发式分类器集成算法.

2.2　知识线记忆理论

知识是如何表述、存储、提取、使用的？心理学中的知识线记忆理论尝试回答了这个问题. 每当你“有一个好主意”，解决了一个问题时，你就会创建知识线来记忆它. 知识线会与被激活的思维智能体相联结，之后当你再次激活此知识线时，与这个知识线联结的智能体就会被激活，使得你进入之前解决问题时相似的“思维状态”. 这就让你在解决新的、相似的问题时，感到容易一些. 这就是知识线的基本理论.

此处引用《心智社会》[56]中提到的一个例子：当你想要维修一辆自行车，在你开始之前，先将红色油漆抹在手上. 这样你所用过的所有工具都会有红色的记号. 当你修好之后，只要记住红色标记表示“有助于修车”，下次你再修自行车的时候就可以节约时间，只需要把涂了红色标记的工具拿出来就可以了. 这里的红色就是知识线，工具就是思维智能体. 如果你用不同的颜色标记不同的工作，有些工具最后可能会有不止一种颜色. 每个智能体可以和多个知识线相联结. 当问题来临，只要激活问题相关的知识线即可.

知识线理论阐述了人类构建记忆和使用记忆的过程，是心理学中对人类行为的一种基于经验的解释，是目前比较被认同的一种猜想. 本文算法受到此理论的启发，对其计算框架进行数学建模，将知识线抽象成线性算子，构造出新的多模型集成算法. 正如知识线记忆理论可以对人类行为进行解释，类知识线的构造也赋予了本文算法较好的可解释性.

3　知识线集成算法

本节将详细介绍本文提出的知识线集成算法，首先 3.1 节用矩阵建模了知识线集成算法并给出计算框架；3.2 节针对知识线理论中未知的复杂函数，给出了知识线集成算法中的定义；3.3 节、3.4 节中具体说明了知识线训练与推断的过程，并给出了算法流程以及相关的描述与分析. 最后 3.5 节中对算法的可解释性进行了说明.

3.1　一般投票与知识线集成计算框架

给定 v 个独立的 ω 类分类器算子，构成向量：$\boldsymbol{C}=(c_1,c_2,\cdots,c_v)$，对于给定数据特征 $\boldsymbol{x}$ 有

$$\boldsymbol{C}(\boldsymbol{x})=\begin{pmatrix}c_1(\boldsymbol{x})\\c_2(\boldsymbol{x})\\\vdots\\c_v(\boldsymbol{x})\end{pmatrix}=\begin{pmatrix}c_{11}&c_{12}&\cdots&c_{1\omega}\\c_{21}&c_{22}&\cdots&c_{2\omega}\\\vdots&\vdots&\ddots&\vdots\\c_{v1}&c_{v2}&\cdots&c_{v\omega}\end{pmatrix}\tag{1}$$

其中，$c_{ij}=c_i(\boldsymbol{x})_j\in\{0,1\}$ 表示第 i 类分类器结果是否为 j，且有 $\sum\limits_j c_{ij}=1$，则分类器结果 $y_i\in\{1,2,\cdots,\omega\}$ 有

$$y_i=\arg\max_z c_{iz},\ z=1,2,\cdots,\omega\tag{2}$$

按照“少数服从多数”的一般投票方式，对每个分类器的结果进行公平的计数，最终票数最多的类别作为最终的结果：

$$y=\arg\max_z\sum_{i=1}^{v}I(y_i=z),\ z=1,2,\cdots,\omega\tag{3}$$

其中，$I(\cdot)$为示性函数，当自变量逻辑为真时结果为1，假时为0. 在式(3)中，若第 i 个分类器结果 y_i 等于 z，则结果为1，否则为0.

以上是 Bagging 中采用的做法. 根据 Bagging 方法的结论，当每个分类器的结果错误率低于随机分类错误率时，Bagging 得到结果的错误率低于单一分类器的错误率均值，且在 n 趋于无穷时 Bagging 结果的错误率趋于理论最小错误率.

从统计学的角度来看，上述方法有很好的理论保证，后续的众多研究也都是在其基础上改进的，但是这些方法都只注重最终结果的正确率，却忽视了算法的可解释性.

根据心理学中的记忆理论，人脑会根据需要，唤醒一部分智能体进行决策，而具体应该唤醒哪些智能体，由人脑之前的记忆决定. 而本文受到此理论的启发，将多分类集成的过程嵌入到知识线记忆理论的框架下，得到知识线集成算法计算框架如图3所示.

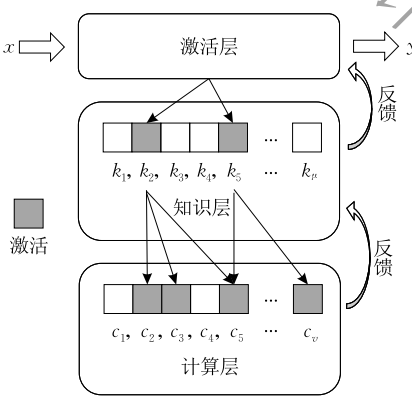

图3 知识线集成算法计算框架

计算层中的分类器算子 c_i 扮演着知识线记忆理论中的计算单元，它可以提供最基础的决策. 知识层中的 k_i 代表知识线理论中的知识线，它与计算层中的计算单元相联结，若当前存在 μ 个知识线，则它的形式为

$$\boldsymbol{K}=\begin{pmatrix}k_1\\k_2\\\vdots\\k_\mu\end{pmatrix}=\begin{pmatrix}k_{11}&k_{12}&\cdots&k_{1v}\\k_{21}&k_{22}&\cdots&k_{2v}\\\vdots&\vdots&\ddots&\vdots\\k_{\mu1}&k_{\mu2}&\cdots&k_{\mu v}\end{pmatrix} \tag{4}$$

其中，$k_{ij}\in\{0,1\}$ 表示第 i 个知识线是否激活第 j 个分类器，若 $k_{ij}=1$，则表示激活.

当接收到数据特征 $\boldsymbol{x}$ 时，根据知识线理论中的表述，只有与问题相关的知识线应该被激活. 因此激活层 $\boldsymbol{A}$ 的目标是对知识线进行激活. 它的形式表达如下：

$$\boldsymbol{A}=\begin{pmatrix}a_1\\a_2\\\vdots\\a_\mu\end{pmatrix}^{\mathrm{T}},\ a_i\in\{0,1\},\ i=1,\cdots,\mu \tag{5}$$

若 $a_i=1$ 则表示第 i 个知识线 k_i 被激活. 最终不同分类结果的分值 $S=(s_1,s_2,\cdots,s_\omega)=\boldsymbol{A}\boldsymbol{K}^*\boldsymbol{C}(\boldsymbol{x})$ 即

$$S=\begin{pmatrix}a_1\\a_2\\\vdots\\a_\mu\end{pmatrix}^{\mathrm{T}}\begin{pmatrix}k_{11}^*&k_{12}^*&\cdots&k_{1v}^*\\k_{21}^*&k_{22}^*&\cdots&k_{2v}^*\\\vdots&\vdots&\ddots&\vdots\\k_{\mu1}^*&k_{\mu2}^*&\cdots&k_{\mu v}^*\end{pmatrix}\begin{pmatrix}c_1(\boldsymbol{x})\\c_2(\boldsymbol{x})\\\vdots\\c_v(\boldsymbol{x})\end{pmatrix} \tag{6}$$

其中，$\boldsymbol{K}^*$ 为 $\boldsymbol{K}$ 每行经过标准化后的结果，且有 $k_{ij}^*=k_{ij}\Big/\sum_j k_{ij}$. 最终分类结果为

$$y=\arg\max_z S_z,\ z=1,2,\cdots,\omega \tag{7}$$

从式(6)可以看出，知识线集成算法本质上是一种加权集成的做法，但与传统加权集成算法不同的是，本算法中的权重矩阵 $\boldsymbol{K}$ 是通过模拟知识线记忆理论中记忆更新迭代的方法得到的，这使得它可以进行更丰富的可解释性方面的分析. 具体将在后文进行讨论.

3.2 知识线的计算

知识线集成算法的计算框架已经在3.1节中详细说明，但是如何计算知识线矩阵 $\boldsymbol{K}$ 中的元素 k_{ij} 以及激活向量 $\boldsymbol{A}$ 中的元素 a_i 还未定义. 实际上知识线理论对于知识线的激活以及计算单元的激活问题也只给出了逻辑表述而缺乏具体算法，本文本着计算简单有效且符合知识线理论中相关表述的原则，对知识线这部分的具体内容与计算方法进行了设计.

3.2.1 计算层激活

计算层中计算单元的激活由与其联结的知识线控制，若第 i 个知识线与第 j 个分类器联结则有 $k_{ij}=1$，否则 $k_{ij}=0$. 根据记忆理论，当遇到无法解决的问题时，大脑不断尝试激活不同的计算单元子集，直到找到解决该问题的子集后，使用一个智能体与本次激活的计算单元相联结，从而构建一个知识线. 即找到一个集合 $C'\sqsubseteq\{c_1,c_2,\cdots,c_v\}$ 使得以下条件成立：

$$y^*=\arg\max_z\sum_{c_i\in C'}I(z=\arg\max_{z'}c_i(\boldsymbol{x})_{z'}) \tag{8}$$

其中，$z,z'=1,2,\cdots,\omega$，y^* 为正确的类别. 因为所有分类器的集合较大，且随机采样得到的 C' 不能保证结果正确性，所以此处令 $C'=C^*$，且对于 $\forall c\in C^*$，$y^*=\arg\max_z c(\boldsymbol{x})_z$. 这样即可保证结果的正确性，从

而避免低效的重复采样.

3.2.2　知识层激活

当使用知识线集成算法进行推断时，激活层将选取部分知识层中的知识线进行激活，即计算 a_i. 根据知识线理论的描述，知识线是根据某个特定问题产生的，之后若遇到类似问题，此知识线将被激活.

在本文算法中，当知识线 k_i 为了记忆样本 $\boldsymbol{x}_{k_i}$ 而产生时，此样本的类别 y_{k_i} 也同时被记忆. 当对新的样本 $\boldsymbol{x}'$ 进行推断时有：

$$a_i = I(y_{k_i} = \arg\max_z (k_i \boldsymbol{C}(\boldsymbol{x}'))_z) \tag{9}$$

其中，$I(\cdot)$ 为示性函数，当自变量逻辑为真时结果为 1，假时为 0. 在式(9)中，若知识线 k_i 判定 $\boldsymbol{x}'$ 与 $\boldsymbol{x}_{k_i}$ 有相同的标签则被激活. 知识线 k_i 所联结的分类器构成了类别 y_{k_i} 的印象，若在同样的映射下，$\boldsymbol{x}'$ 得到相同的结果，说明 $\boldsymbol{x}'$ 与 $\boldsymbol{x}_{k_i}$ 具有相似性. 因此，式(9)的是符合知识线激活描述的一种激活方法.

3.3　记忆的产生

知识线集成算法主要包含三部分，激活矩阵 $\boldsymbol{A}$，知识线矩阵 $\boldsymbol{K}$，分类算子向量 $\boldsymbol{C}$，其中 $\boldsymbol{C}$ 如式(1)的形式，是提前训练完成的；$\boldsymbol{A}$ 是基于 $\boldsymbol{K}$ 得到的，而 $\boldsymbol{K}$ 中参数需要通过学习获得. 记忆的产生即知识线的更新，也就是 $\boldsymbol{K}$ 的训练，其具体算法如下：

算法 1.　知识线矩阵参数学习.

输入：分类算子向量 $\boldsymbol{C}$；数据集 Data

输出：知识线矩阵 $\boldsymbol{K}$

1. 初始化 $\boldsymbol{K}=(0,0,\cdots,0)$，$y_{\boldsymbol{K}}=(0)$
2. FOR $(\boldsymbol{x},y)$ in Data DO
3. $\quad y'=Inference(\boldsymbol{C},\boldsymbol{K},y_{\boldsymbol{K}},\boldsymbol{x})$
4. $\quad$ IF $y'\neq y$ THEN
5. $\quad\quad k'=(I(c_1(\boldsymbol{x})\rightarrow y),\cdots,I(c_v(\boldsymbol{x})\rightarrow y))^{\mathrm{T}}$
6. $\quad\quad \boldsymbol{K}=(\boldsymbol{K}^{\mathrm{T}}|k')^{\mathrm{T}}$，$y_{\boldsymbol{K}}=(y_{\boldsymbol{K}}|y)$
7. $\quad\quad$ IF need forget THEN
8. $\quad\quad\quad k_i=\arg\min_{k\in\boldsymbol{K}^{\Delta}} Precision(k)$
9. $\quad\quad\quad \boldsymbol{K}=(k_1,\cdots,k_{i-1},k_{i+1},\cdots,k_{\boldsymbol{K}_{\mathrm{row}}})^{\mathrm{T}}$
10. $\quad\quad\quad y_{\boldsymbol{K}}=(y_{k_1},\cdots,y_{k_{i-1}},y_{k_{i+1}},\cdots,y_{k_{K_{\mathrm{row}}}})$
11. $\quad\quad$ END IF
12. $\quad$ END IF
13. END FOR
14. RETURN $\boldsymbol{K}$

训练伊始，模型不存在记忆，此时 $\boldsymbol{K}$ 中不包含有效信息，当遇到问题时，若此时依靠知识线无法推断出正确答案，则需要参考正确答案. 算法 1 第 5 行中的 $I(\cdot)$ 为示性函数，$I(c_i(\boldsymbol{x})\rightarrow y)=1$ 则表示第 i 个分类器结果正确. 这样得到的 k' 即可满足式(8)的要求. 将 k' 添加到 $\boldsymbol{K}$ 的最后一行并记录知识线 k' 所对应的类型 y，即完成了一次知识线的更新. 经过一段时间的迭代后，知识线矩阵中已经储存了一定信息，此时若推断错误有两个原因：其一，现有知识线尚未覆盖当前问题，所以依旧需要执行上述更新记忆的操作；其二，现有知识线中存在精准率较低的特例知识线，它严重影响了整体集成的效果，这个问题则需要通过遗忘来解决.

在遗忘过程中，模型会按照给定概率 p 删除一条知识线如下：

$$k'=\arg\min_{k\in\boldsymbol{K}^{\Delta}} \frac{\sum_{i=1}^{N} I(y_i=y_k, y_{k,i}=y_k)}{\sum_{i=1}^{N} I(y_{k,i}=y_k)} \tag{10}$$

其中，N 为样本总数，$I(\cdot)$ 为示性函数，$\boldsymbol{K}^{\Delta}$ 表示被错误激活的知识线集合即集合内元素被激活但所对应的类别是错误的，y_k 表示知识线 k 对应的标签，$y_{k,i}$ 表示知识线 k 对第 i 个数据判断的结果，y_i 表示第 i 个数据的真实标签. 式(10)可以更直观的表述为

$$k'=\arg\min_{k\in\boldsymbol{K}^{\Delta}} Precision(k) \tag{11}$$

即在犯错的知识线中找到精准率最低的知识线进行删除. 不妨设 k' 在知识线矩阵 $\boldsymbol{K}$ 的第 i 行. 所以经过遗忘之后的知识线矩阵为

$$\boldsymbol{K}=(k_1,\cdots,k_{i-1},k_{i+1},\cdots,k_{|\boldsymbol{K}|})^{\mathrm{T}} \tag{12}$$

由于激活操作的存在，本算法实际上使用了二分类器集合来判断样本是否属于某特定类别，并通过投票解决多分类问题，因此当解决 ω 分类问题时，目标函数可设置为最大化 R：

$$R=\sum_{i=1}^{\omega} precision_i+recall_i \tag{13}$$

对于类别为 j 的单个知识线，它只对所属类别的精准率即 $precision_{i=j}$ 以及其他类别的召回率即 $recall_{i\neq j}$ 起作用. 此知识线精准率越高则本身所属类别精准率越高，且对其他类别的召回率负面影响越小. 特别地，当精准率为 100% 时，此知识线仅对自身类别样本的推断提供正确信息且完全不影响其他类别. 单个知识线的高召回率可以有效减少知识线的必要数量，但并不是单个知识线的必要目标. 精准率是单个知识线唯一需要考虑的目标，且精准率越高效果越好，所以在遗忘知识线时采用贪心算法，留下精准率更高的知识线. 在保证高精准率的情况下，增加知识线的过程则可近似成用贪心法解决集合覆盖问题的过程. 无法正确推断的样本相当于未

覆盖的元素，模型添加至少能解决此样本的知识线，相当于覆盖问题中增加一个至少包含此未覆盖元素的集合.因此随着训练迭代，知识线集成召回率将逐步提高.

3.4 记忆的使用

不管是使用知识线集成完成测试，还是训练中判断记忆是否可以解决问题，都需要使用知识线完成数据到类标签的映射.记忆的使用即推断的过程，具体步骤如下：

算法 2. 知识线集成推断 *Inference* 函数.

输入：分类算子向量 $\boldsymbol{C}$；知识线矩阵 $\boldsymbol{K}$，知识线类别标签 $y_{\boldsymbol{K}}$；数据特征 $\boldsymbol{x}$

输出：推断结果 y'

1. $a_i=I(y_{k_i}=\arg\max_z(k_i\boldsymbol{C}(\boldsymbol{x}))_z)$，$i=1,2,\cdots,\boldsymbol{K}_{\text{row}}$
2. $\boldsymbol{A}=(a_1,a_2,\cdots,a_{\boldsymbol{K}_{\text{row}}})$
3. $\boldsymbol{K}^*\leftarrow$ row normalizing $\boldsymbol{K}$
3. $S=\boldsymbol{A}\boldsymbol{K}^*\boldsymbol{C}(\boldsymbol{x})$
4. $y'=\arg\max_i s_i$，$i=1,2,\cdots,S_{\text{col}}$
5. RETURN y'

根据式(9)得到激活矩阵 $\boldsymbol{A}$，根据式(6)，得到不同类别的分值 S，其中分值最高的类别则为推断结果.若存在多个类别分值相同则随机选择其中一类输出.

3.5 可解释性

在知识线矩阵训练的过程中，可以记录产生记忆时被激活的分类算子集合 C_x、被记忆数据的特征 $\boldsymbol{x}$ 和标签 y，这些是知识线可解释性分析的要素，因为 $\forall c_i\in C_x$，$c_i(\boldsymbol{x})=y$，若 c_i 连续，则有 $c_i(\boldsymbol{x}+\boldsymbol{\varepsilon})=y$ 当 $\boldsymbol{\varepsilon}\rightarrow 0$ 时成立，所以知识线包含了“形如 $\boldsymbol{x}$ 的数据标签为 y”的信息.当一个新的数据 $\boldsymbol{x}'$ 需要被推断时，知识线会使用 C_x 来判断 $\boldsymbol{x}'$ 的标签是否为 y.这实际上是一种类似谱聚类[57]的过程，如果 C_x 可以把 $\boldsymbol{x}'$ 映射到标签 y，则说明 $\boldsymbol{x}'$ 和 $\boldsymbol{x}$ 在 C_x 关注的特征上距离较近，所以 $\boldsymbol{x}'$ 和 $\boldsymbol{x}$ 之间存在着一定的相似性.

因为在记忆中存在着和新数据 $\boldsymbol{x}'$ 相似的数据 $\boldsymbol{x}$，所以模型做出了 $\boldsymbol{x}'$ 的标签可能是 y 的推测.由于记忆是丰富的，可能有多个知识线被同时激活，所以最终的结果由知识线投票产生，而结果的票数则可以反映 $\boldsymbol{x}'$ 是每个类别的可能性.被激活的知识线也代表着曾经出现过的与 $\boldsymbol{x}'$ 相似的样本，最后的结果可以认为是立足于样本进行的投票，而不是像已有的方法是立足于模型进行的投票.由于类似的样本大部分是某个标签，所以算法推断样本是这个标签.知识线集成算法把学习和推断的过程显式的表达了出来.

以手写数字识别为例，模型通过见识各种不同的数字，并记下曾经不认识的形状应当是什么数字.不仅如此，不同的人写字的风格不一样，当模型无法用标准的 0～9 进行判断时，也会逐渐学会各种不同风格的同一个数字.根据经验，1 和 7 经常容易混淆，那么假设当模型经过足够训练后，现在需要推断一个长得又像 1 又像 7 的图片到底是哪个数字，已有的集成方法给出解决方案却不会给出原因，而知识线集成可以提供很多类似的图片，并通过统计不同类别图片出现的频次反馈给用户结果.虽然知识线集成也是将分类器进行集成，但是中间过程却可以抽取出样本作为推断依据，让整个过程有理有据.

就推断过程而言，知识线集成有着与 K 近邻算法相似的可解释性.但是知识线集成可以进行更丰富的分析.知识线数量作为模型的参数，可以量化问题不同粒度下的难度.比如，单个类别的难度可以由不同类别的知识线出现频率量化.容易混淆的类别可以使用被遗忘知识线的混淆情况量化.每一个测试样本的难易程度，可以用被激活知识线的种类个数量化.结果的可信度可以用知识线激活的类别占比量化.由于引入了知识线，这些原本难以直接通过模型参数评估的指标，都可以使用最基本的古典概型诠释.

4 实 验

此章节对本文实验所用的数据集、实验的方法做出了介绍，并对实验结果进行了分析.

4.1 数据集

Wine、Statlog.(Heart)、Wall-Following Robot Navigation Data、Ecoli、Glass Identification、Balance Scale、Iris、Seeds、Contraceptive Method Choice、Connectionist Bench(Sonar, Mines vs. Rocks)均是加州大学欧文分校机器学习数据库中经典的分类数据集.它们提供样本多维特征以及相应类型标签，可用于测试分类模型算法性能.实验是在随机划分数据集的 80％作为训练集，20％作为测试集下进行的.在下文中，Statlog.(Heart)简称为 Heart，Wall-Following Robot Navigation Data 简称为 Robot，Glass Identification 简称为 Glass，Balance Scale 简

称为 Balance，Contraceptive Method Choice 简称为 CMC，Connectionist Bench（Sonar，Mines vs. Rocks）简称为 Sonar.

MNIST 数据集是一个常用的手写识别数据集. 它的每条数据是 784 维的特征，用于表示一副 28×28 尺寸图片每个像素的灰度值；标签为 0～9 的数字，用来表示图片对应的手写阿拉伯数字. 此数据集拥有 70000 个图片样本，其中训练集 60000 个，测试集 10000 个.

Fashion MNIST 数据集是一个时尚用品类别识别数据集，以下简称 Fashion. 它的维度，尺寸，数据集大小与 MNIST 完全一致，总共十类：0 表示 T 恤/上衣，1 表示裤子，2 表示套头衫，3 表示连衣裙，4 表示大衣，5 表示凉鞋，6 表示衬衣，7 表示运动鞋，8 表示包，9 表示高帮鞋. 其中每个类别的样本数量相同.

4.2　实验设置

本文模型仅对内存有一定要求，数据集越大，分类难度越高，所需要的知识线存储空间越大. 本文实验需求至少 16 GB 内存.

用于对比的 K 近邻、朴素贝叶斯[58]、逻辑回归[59]是传统的非集成分类模型，随机森林、AdaBoost、GBDT[60]是经典的集成学习分类模型，OO 集成[61]是选择性集成的典型做法，它根据结果方向为不同样本选择不同分类器子集进行决策.

特别地，OO 集成和知识线集成在所有数据集上均使用与随机森林相同的决策树集合作为基础. 对于 Fashion 数据集上的可解释性相关实验，知识线集成了 100 个决策树分类器，且所有决策树均为随机抽取 Fashion 数据集中的 20000 个随机样本的 100 维随机特征训练得到的.

另外，知识线集成中的遗忘概率对实验结果有一定影响，本文基于大量调优实验，使用待删除知识线的精准率作为放弃遗忘的概率.

4.3　分类性能实验与分析

知识线集成算法适用于任何种类的分类器成员. 在本文实验中，仅对决策树作为分类器成员进行了验证与讨论.

4.3.1　正确率对比

知识线集成算法作为一种新的多分类器集成算法，在不同数据集上与传统分类算法以及经典集成算法的正确率对比如表 1 所示.

表 1　分类正确率

数据集	K 近邻	朴素贝叶斯	逻辑回归	AdaBoost	GBDT	随机森林	OO	知识线
Wine	0.6389	0.9722	0.9444	0.9722	0.9722	0.9653	**0.9792**	0.9722
Heart	0.6292	0.8148	0.7778	**0.8519**	0.6852	**0.8519**	0.8333	**0.8519**
Robot	0.8608	0.5339	0.6914	**0.9918**	**0.9918**	0.9890	0.9908	**0.9918**
Ecoli	**0.8676**	0.7059	0.7353	0.6618	0.7941	0.8382	0.8235	0.8235
Glass	0.6977	0.4419	0.5116	0.5116	0.7209	0.8140	0.8372	**0.8605**
Balance	0.7680	0.8400	0.8080	**0.8720**	0.8320	0.7920	0.7920	0.7680
Iris	0.9000	0.8667	0.8000	**0.9000**	0.8667	0.8667	0.8778	0.8667
Seeds	**0.8810**	**0.8810**	**0.8810**	0.6429	0.8571	0.8631	**0.8810**	**0.8810**
CMC	0.5627	0.4881	0.5153	0.5390	**0.5864**	0.5559	0.5763	0.5424
Sonar	0.8095	0.5476	0.7619	0.7857	**0.9048**	0.8691	0.8333	0.8452
MNIST	**0.9668**	0.5558	0.9173	0.7299	0.9487	0.9640	0.9638	0.9632
Fashion	0.8577	0.5856	0.8374	0.5425	0.8682	**0.8715**	0.8686	0.8710

由结果可见，知识线集成算法在多个数据集上表现最佳，且在大部分数据集上不存在显著不足. 可以认为知识线集成算法在不讨论可解释性的情况下，其分类正确率与其它集成算法表现在同一水平线上. 在此基础之上，引入知识线概念为知识线集成算法增添的可解释性成为了其相比于其他算法的优势，这将在后文重点讨论.

4.3.2　记忆的作用

随着模型遭遇无法解决的问题，知识线被建立，模型能力逐渐提高，图 4 所示为 Fashion 数据集上的实验结果.

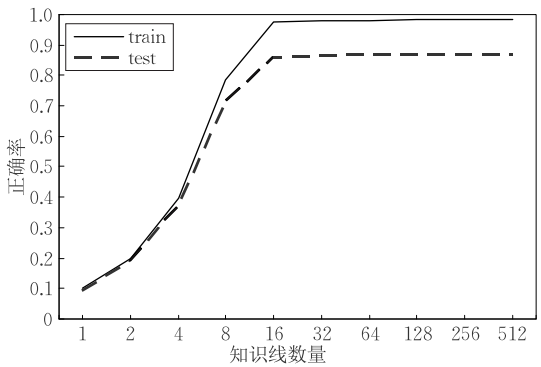

图 4　知识线数量与正确率的关系

随着知识线增加,模型正确率不断提高,且趋于平稳.当知识线数量小于类别数时,新增知识线总对应尚未接触过的类别,所以模型能力增长较快.当知识线数量超过类别数后,正确率增长缓慢.此时模型已经掌握不同类别的大致情况,想要进一步提高正确率变得困难,需要大量的知识线来刻画更多细节.512 个知识线相对于 16 个知识线,训练集上正确率提高了 0.68%,测试集上正确率提高了 1.08%.

4.3.3 收敛过程

分类问题中精准率和召回率是一组存在矛盾的指标.当使用贪心决策,若希望得到尽量高的精准率,召回率则不可避免的变低.反之若希望得到尽量高的召回率,精准率则会受损.而知识线集成算法将这两个指标分割到两个不同的部分,作为各自的主要优化目标,在一定程度上缓解了这一矛盾.知识线个体作为解决问题的核心单元,它的目标是拥有尽量高的精准率.类似于人脑中的记忆,当人类面临一个问题,并不会激活所有记忆,而是激活能切实解决问题的记忆.知识线集成算法与此是一致的,当知识线"十拿九稳"时才被激活,即知识线分类的精准率要高.图 5 所示为在 Fashion 数据集的训练集上的实验结果.

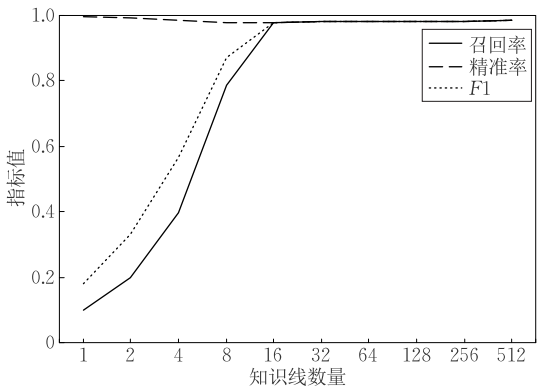

图 5 Fashion 训练集分类评价指标

正如上文所述,高精准率是基础,从图 5 可见知识层中的知识线精准率均值一直处于较高水平.在知识线数量较少时,随着知识线数量的增加,精准率有下降趋势.但由于遗忘机制的存在,精准率较差的知识线将被删除,所以后续整体的精准率又有所提升.

就召回率而言,随着知识线数量的增加,更多不同种类的问题被解决,整体的召回率水平逐渐提升.最终模型的召回率、精准率、$F1$ 在训练集上非常接近.

图 6 展示了同一个实验中测试集上的表现.可以看出,结论与训练集上保持一致.需要注意的是,当知识线数量大于类别总数后,召回率与精准率是有小幅增幅的.512 个知识线相对于 16 个知识线,在训练集上召回率增长了 1.08%,精准率增长了 1.01%;在测试集上召回率增长了 0.68%,精准率增长了 0.54%,实验中的具体数值可在附录 1 中查看.

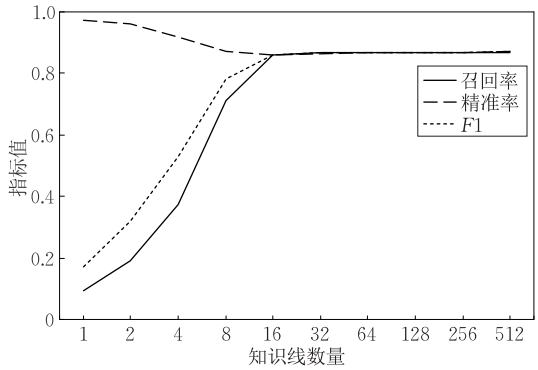

图 6 Fashion 测试集分类评价指标

4.4 可解释性实验与分析

由于知识线集成算法中的知识线是基于心理学中的知识线记忆理论设计的,这为我们理解模型、解释结果提供了思路.

4.4.1 样本难点提取

多个知识线同时被激活时,最终结果存在以下 4 种情况:

(1)没有知识线被激活,表示模型认为没有见过类似的样本.

(2)所有被激活的知识线投票一致,这种情况得到的结果有更高的置信度.

(3)所有被激活的知识线投票不一致,但是有某个类别胜出.

(4)所有被激活的知识线投票不一致,且最终出现至少两个类别平票的情况.

以 Fashion 数据集上的实验为例.对于训练集和测试集,最终这 4 类情况发生的分布如图 7 所示.

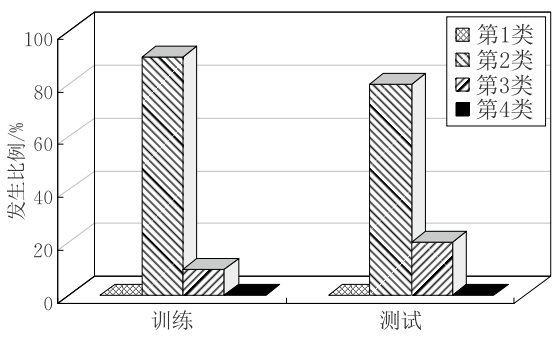

图 7 4 类情况发生频率

其中第 1 类情况在实验中没有发生,第 4 类仅在测试中发生了 1 例,所以后续不再进行讨论.为了进一步探讨知识线集成提炼问题难点的能力,本文对属

于不同区域的样本在知识线集成与随机森林中的表现进行了对比，如表 2 所示.

表 2　特定类别正确率

	随机森林			知识线集成	
	第 2 类/%	第 3 类/%		第 2 类/%	第 3 类/%
训练	99.96	81.39	训练	99.96	**84.19**
测试	94.97	**56.37**	测试	94.97	56.16

根据之前的定义，第 2 类表示激活的知识线在决策时答案是一致的，而第 3 类则说明有多种类别的知识线被激活. 直觉上来说，第 3 类的样本难度是比第 2 类高的. 表 2 中的实验结果也与直觉相符，在第 3 类样本上，随机森林与知识线集成出现了较大的问题. 可见本文方法使用激活知识线的情况为测试样本划分类别，可以提炼出更有难度的样本即第 3 类样本.

4.4.2　类别难点量化

以 Fashion 数据集上的实验为例，表 3 中展示了训练完毕后每个类别的知识线数量与不同算法在测试集每个类别上的正确率.

表 3　各类别知识线数量与正确率

类别	知识线数量	知识线集成/%	随机森林/%	K 近邻/%
0	282	82.9	85.6	88.6
1	289	95.7	95.9	97.7
2	471	78.9	81.6	81.6
3	281	89.6	89.9	87.1
4	431	80.4	81.3	81.4
5	321	95.2	94.8	80.2
6	1360	65.2	56.5	52.8
7	366	94.4	94.2	96.6
8	239	96.0	96.9	96.7
9	409	94.6	94.8	96.3

第 6 类知识线数量相比于其他类别明显更大，而随机森林和知识线集成在第 6 类上的正确率水平明显低于其他类别，可见这一类别难度较高. 而更进一步的计算相关系数，知识线数量与知识线集成各类别正确率的相关系数为 -0.82，知识线数量与随机森林各类别正确率的相关系数为 -0.92. 此处使用的知识线集成与随机森林是由一样的 100 个决策树作为分类器成员得到的，为排除成员本身质量的影响，此处还对比了直接由样本进行推断的 K 近邻方法. 表中结果为 Cosine 距离在 $K=4$ 时的结果，此结果是遍历了 K 等于 1 至 100，分别使用 Cosine 距离与欧式距离测试后得到的最好结果. 经过计算，知识线数量与 K 近邻分类器各类别正确率的相关系数为 -0.88，可见知识线数量有一定量化类别难度的能力.

值得一提的是知识线数量与知识线集成结果的相关度更低，这表明知识线集成算法在发现问题难点后会努力将其解决，因此第 6 类知识线数量较大的同时知识线集成算法在第 6 类上的正确率也显著高于其他方法.

4.4.3　易混淆难点量化

知识线集成算法在训练过程中存在记忆遗忘的机制. 根据遗忘的规则，被遗忘的知识线必然存在将两类混淆的情况. 虽然遗忘的过程具有一定的随机性，但是若假设类 A 与类 B 混淆的概率大于类 A 与类 C 混淆的概率. 那么混淆 A 与 B 的知识线数量将大概率大于混淆 A 与 C 的知识线数量. 若个体被遗忘概率相同，混淆 A 与 B 的知识线被遗忘的概率期望大于混淆 A 与 C 的知识线. 因此被遗忘知识线的犯错情况可以用来量化问题中容易混淆的类别.

以 Fashion 数据集上的实验为例，统计知识线被遗忘时的犯错原因并进行可视化得到图 8，其中坐标 (i,j) 的灰度值表示将第 i 类错判成第 j 类的犯错相对频率，颜色越深表示频率越高.

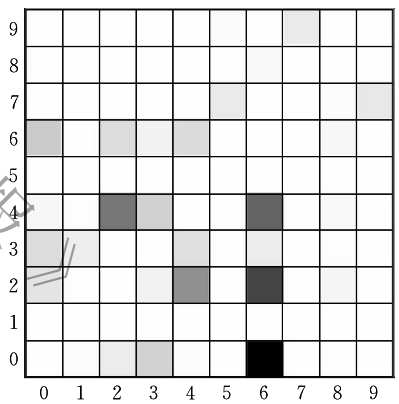

图 8　类间混淆率可视化

从图 8 中可以发现模型容易把 6-衬衫混淆为 0-T 恤、2-套头衫、4-大衣，由此也可以解释在 4.4.2 节中发现的第 6 类难度较大的原因. 另外，用遗忘知识线的犯错情况来量化类别间易混淆程度，结果符合直观预期. 0、2、3、4、6 之间容易混淆，因为这些类别都是衣服，它们不容易和 5、7、9 这些类混淆，因为要分辨衣服和鞋子是容易的. 而这之中 1-裤子，8-包和其他类别想要区分开直观上也是容易的. 这也是第 1 类分类正确率高达 95%，第 8 类分类正确率高达 96%的原因.

4.4.4　推断证据提供

知识线集成算法是一种从样本推断样本的算法模型，所以不论结果正确与否，模型都可以提供依据.

由之前的实验可以发现知识线算法易于分析,可以有效的将问题难点提炼.以下,使用实验中 Fashion 数据集上的一个真实例子来更为直观的展示知识线集成算法提供推断证据的能力.

正如 3.5 节中所讨论的,知识线集成算法可以显式的呈现判断的过程.当图 9 的特征输入模型,有 162 个知识线被成功激活,其中 41 个关于 2-套头衫的知识线被激活,27 个关于 4-大衣的知识线被激活,94 个关于 6-衬衫的知识线被激活.被激活的知识线对应的样本每类抽取了 3 个,如图 10 所示.

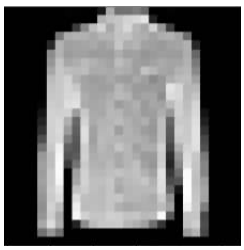

图 9 被推断的图片(标签为 6-衬衫)

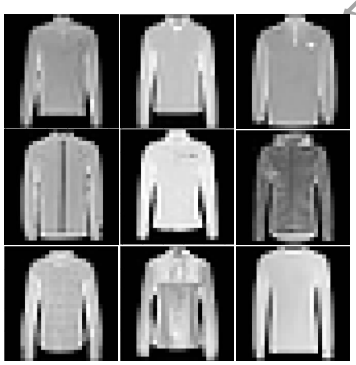

图 10 被激活的知识线对应的样本
(三行分别为 2-套头衫,4-大衣,6-衬衫)

可以看出,这些记忆和输入的图片是有一定相似性的,也就是说,当测试集向模型展示新的图片时,模型回忆了过去所遇到过的类似图片.因为类似的图片大部分都是 6-衬衫,所以根据经验,图 9 的标签也应该是 6,且概率为 94/162.

这种从训练样本中找依据的做法,和 K 近邻的做法相似.但是知识线集成算法中的知识线越多则效果越好,且经过训练,知识线的数量也会趋于稳定,而 K 近邻则需要选择合适的 K.另外,知识线集成算法相对于 K 近邻有着更强的分类能力,在 Fashion 数据集上,知识线集成算法测试集上的分类正确率可以达到 87.31%,而 K 近邻在 K 取 1~100 中,使用 Cosine 距离所能达到的最高值 85.90%在 $K=4$ 时取得,使用欧式距离能达到的最高值 85.77%在 $K=4$ 时取得.K 近邻算法的具体表现详情可参考附录 2.

5 总 结

本文针对多分类器系统中的集成策略进行了研究,结合心理学中的知识线记忆理论,提出了一种拥有较强可解释性的多分类器集成算法,称为知识线集成算法.此算法根据历史解决问题的记忆构建知识线记忆矩阵,最终使用样本相关的记忆解决问题.推断新问题时,此算法可以找到训练样本中和此问题相似问题的解决方案,显式地呈现集成模型推断的过程并给出结论的依据.知识线集成算法不仅拥有良好的分类性能,还可以通过知识线的创建、遗忘、激活情况提炼问题的难点,进行更具可解释性的分析实验与数据相关性挖掘.

参 考 文 献

[1] Wolpert D H, Macready W G. No free lunch theorems for optimization. IEEE Transactions on Evolutionary Computation, 1997, 1(1): 67-82

[2] Neumann J V. The computer and the brain. Annals of the History of Computing, 1958, 11(3): 161-163

[3] Chow C K. Statistical independence and threshold functions. IEEE Transactions on Electronic Computers, 2006, EC-14(1): 66-68

[4] Doshi-Velez F, Kim B. Towards a rigorous science of interpretable machine learning. arXiv, 2017, 1050: 2

[5] Cover T, Hart P. Nearest neighbor pattern classification. IEEE Transactions on Information Theory, 1967, 13(1): 21-27

[6] Su X, Khoshgoftaar T M. A survey of collaborative filtering techniques. Advances in Artificial Intelligence, 2009, 2009: 1-19

[7] Minsky M. *K*-Lines: A theory of memory. Cognitive Science, 1980, 4(2): 117-133

[8] Woźniak M, Graña M, Corchado E. A survey of multiple classifier systems as hybrid systems. Information Fusion, 2014, 16: 3-17

[9] Sagi O, Rokach L. Ensemble learning: A survey. Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, 2018, 8(4): e1249

[10] Lam L. Classifier combinations: Implementations and theoretical issues//Proceedings of the International Workshop on Multiple Classifier Systems. Berlin, Germany: Springer, 2000: 77-86

[11] Rahman A F R, Fairhurst M C. Serial combination of multiple experts: A unified evaluation. Pattern Analysis & Applications, 1999, 2(4): 292-311

[12] Fumera G, Pillai I, Roli F. A two-stage classifier with reject option for text categorisation//Proceedings of the Joint IAPR International Workshops on Statistical Techniques in Pattern Recognition (SPR) and Structural and Syntactic Pattern Recognition(SSPR). Berlin, Germany: Springer, 2004: 771-779

[13] Bartlett P L, Wegkamp M H. Classification with a reject option using a hinge loss. Journal of Machine Learning Research, 2008, 9(8): 1823-1840

[14] Termenon M, Graña M. A two stage sequential ensemble applied to the classification of Alzheimer's disease based on MRI features. Neural Processing Letters, 2012, 35(1): 1-12

[15] Clark P, Niblett T. The CN2 induction algorithm. Machine Learning, 1989, 3(4): 261-283

[16] Rivest R L. Learning decision lists. Machine Learning, 1987, 2(3): 229-246

[17] Freund Y. Boosting a weak learning algorithm by majority. Information and Computation, 1995, 121(2): 256-285

[18] Freund Y, Schapire R E. A decision-theoretic generalization of on-line learning and an application to boosting//Proceedings of the European Conference on Computational Learning Theory. Berlin, Germany: Springer, 1995: 23-37

[19] Schapire R E. The strength of weak learnability. Machine Learning, 1990, 5(2): 197-227

[20] Kivinen J, Warmuth M K. Boosting as entropy projection//Proceedings of the 12th Annual Conference on Computational Learning theory. Santa Cruz, USA, 1999: 134-144

[21] Tumer K, Ghosh J. Analysis of decision boundaries in linearly combined neural classifiers. Pattern Recognition, 1996, 29(2): 341-348

[22] Breiman L. Bagging predictors. Machine Learning, 1996, 24(2): 123-140

[23] Breiman L. Random forests. Machine Learning, 2001, 45(1): 5-32

[24] Skurichina M, Duin R P W. Bagging, boosting and the random subspace method for linear classifiers. Pattern Analysis & Applications, 2002, 5(2): 121-135

[25] Tremblay G, Sabourin R, Maupin P. Optimizing nearest neighbour in random subspaces using a multi-objective genetic algorithm//Proceedings of the 17th International Conference on Pattern Recognition. Cambridge, UK, 2004, 1: 208-211

[26] Ting K M, Wells J R, Tan S C, et al. Feature-subspace aggregating: Ensembles for stable and unstable learners. Machine Learning, 2011, 82(3): 375-397

[27] Bryll R, Gutierrez-Osuna R, Quek F. Attribute bagging: Improving accuracy of classifier ensembles by using random feature subsets. Pattern Recognition, 2003, 36(6): 1291-1302

[28] Duan K, Keerthi S S, Chu W, et al. Multi-category classification by soft-max combination of binary classifiers//Proceedings of the International Workshop on Multiple Classifier Systems. Berlin, Germany: Springer, 2003: 125-134

[29] Kuncheva L I. Combining Pattern Classifiers: Methods and Algorithms. John Wiley & Sons, 2014

[30] Raudys Š. Trainable fusion rules. I. Large sample size case. Neural Networks, 2006, 19(10): 1506-1516

[31] Raudys Š. Trainable fusion rules. II. Small sample-size effects. Neural Networks, 2006, 19(10): 1517-1527

[32] Inoue H, Narihisa H. Optimizing a multiple classifier system //Proceedings of the Pacific Rim International Conference on Artificial Intelligence. Berlin, Germany: Springer, 2002: 285-294

[33] Alexandre L A, Campilho A C, Kamel M. Combining independent and unbiased classifiers using weighted average//Proceedings of the 15th International Conference on Pattern Recognition. Barcelona, Spain, 2000, 2: 495-498

[34] Biggio B, Fumera G, Roli F. Bayesian analysis of linear combiners//Proceedings of the International Workshop on Multiple Classifier Systems. Berlin, Germany: Springer, 2007: 292-301

[35] Kittler J, Alkoot F M. Sum versus vote fusion in multiple classifier systems. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2003, 25(1): 110-115

[36] Rao N S V. A generic sensor fusion problem: Classification and function estimation//Proceedings of the International Workshop on Multiple Classifier Systems. Berlin, Germany: Springer, 2004: 16-30

[37] Shlien S. Multiple binary decision tree classifiers. Pattern Recognition, 1990, 23(7): 757-763

[38] Wozniak M. Experiments with Trained and Untrained Fusers. Innovations in Hybrid Intelligent Systems. Berlin, Germany: Springer, 2007: 144-150

[39] Wozniak M. Evolutionary approach to produce classifier ensemble based on weighted voting//Proceedings of the 2009 World Congress on Nature & Biologically Inspired Computing. Kochi, India, 2009: 648-653

[40] Zheng Z, Padmanabhan B. Constructing ensembles from data envelopment analysis. INFORMS Journal on Computing, 2007, 19(4): 486-496

[41] Partalas I, Tsoumakas G, Vlahavas I. Pruning an ensemble of classifiers via reinforcement learning. Neurocomputing, 2009, 72(7-9): 1900-1909

[42] Ruta D, Gabrys B. Classifier selection for majority voting. Information Fusion, 2005, 6(1): 63-81

[43] Wolpert D H. Stacked generalization. Neural networks, 1992, 5(2): 241-259

[44] Hashem S. Optimal linear combinations of neural networks. Neural Networks, 1997, 10(4): 599-614

[45] Duan Z, Wang L. *K*-dependence Bayesian classifier ensemble. Entropy, 2017, 19(12): 651

[46] Mahdianpari M, Salehi B, Mohammadimanesh F, et al. Random forest wetland classification using ALOS-2 L-band, RADARSAT-2 C-band, and TerraSAR-X imagery. ISPRS Journal of Photogrammetry and Remote Sensing, 2017, 130: 13-31

[47] Maghsoudi Y, Collins M, Leckie D G. Polarimetric classification of Boreal forest using nonparametric feature selection and multiple classifiers. International Journal of Applied Earth Observation and Geoinformation, 2012, 19(Complete): 139-150

[48] Siami M, Naderpour M, Lu J. A choquet fuzzy integral vertical bagging classifier for mobile telematics data analysis// Proceedings of the 2019 IEEE International Conference on Fuzzy Systems. New Orleans, USA, 2019: 1-6

[49] Koay A, Chen A, Welch I, et al. A new multi classifier system using entropy-based features in DDoS attack detection// Proceedings of the 2018 International Conference on Information Networking. Chiang Mai, Thailand, 2018: 162-167

[50] Ala'Raj M, Abbod M. Classifiers consensus system approach for credit scoring. Knowledge-Based Systems, 2016, 104: 89-105

[51] Tsai C F. Combining cluster analysis with classifier ensembles to predict financial distress. Information Fusion, 2014, 16(1): 46-58

[52] Ibrahim W, Abadeh M S. Protein fold recognition using deep kernelized extreme learning machine and linear discriminant analysis. Neural Computing and Applications, 2018, (4): 1-14

[53] Malik F, Farhan S, Fahiem M A. An ensemble of classifiers based approach for prediction of Alzheimer's disease using fmri images based on fusion of volumetric, textural and hemodynamic features. Advances in Electrical & Computer Engineering, 2018, 18(1): 61-70

[54] Anyosa S C, Vinagre J, Jorge A M. Incremental matrix co-factorization for recommender systems with implicit feedback //Proceedings of the 2018 World Wide Web Conference. Lyon, France, 2018: 1413-1418

[55] Logesh R, Subramaniyaswamy V, Malathi D, et al. Enhancing recommendation stability of collaborative filtering recommender system through bio-inspired clustering ensemble method. Neural Computing and Applications, 2018, (5): 1-24

[56] Minsky M. The society of mind. Personalist Forum, 1987, 3(1): 19-32

[57] Ng A Y, Jordan M I, Weiss Y. On spectral clustering: Analysis and an algorithm//Proceedings of the Advances in Neural Information Processing Systems. Vancouver, Canada, 2002: 849-856

[58] Rish I. An empirical study of the naive Bayes classifier. Journal of Universal Computer Science, 2001, 1(2): 127

[59] Fan R E, Chang K W, Hsieh C J, et al. LIBLINEAR: A library for large linear classification. Journal of Machine Learning Research, 2008, 9(9): 1871-1874

[60] Friedman J H. Stochastic gradient boosting. Computational Stats & Data Analysis, 2002, 38(4): 367-378

[61] Martínez-Muñoz G, Suárez A. Pruning in ordered bagging ensembles//Proceedings of the Machine Learning, Twenty-Third International Conference. Pittsburgh, USA, 2006: 609-616

附录 1.

知识线数量与分类指标变化具体情况见表 4.

表 4 Fashion 数据集参数 20000-100 知识线数量与分类指标详情

知识线数量	精准率	召回率	*F*1
1	0.9967	0.0993	0.1806
2	0.9937	0.1989	0.3315
4	0.9836	0.3963	0.5650
8	0.9771	0.7874	0.8720
16	0.9785	0.9771	0.9778
32	0.9806	0.9805	0.9806
128	0.9818	0.9817	0.9818
256	0.9827	0.9827	0.9827
512	0.9830	0.9829	0.9830

附录 2.

K 近邻算法正确率具体情况见表 5.

表 5 Fashion 数据集 *K* 近邻表现具体情况

K 取值	Cosine 距离/%	欧式距离/%
1	85.67	84.97
2	85.41	84.60
3	85.64	85.41
4	95.90	85.77
5	85.78	85.54
6	85.80	85.44
7	85.59	85.40
8	85.42	85.34
9	85.16	85.19
10	85.29	85.15
11	84.76	84.95
12～100	＜85.10	＜85.00

YU Si-Hao, Ph. D. candidate. His research interests include reinforcement learning, ensemble learning and adaptive networks.

GUO Jia-Feng, Ph. D., professor. His research interests include information retrieval and data mining.

FAN Yi-Xing, Ph. D., assistant professor. His research interests include information retrieval and natural language processing.

LAN Yan-Yan, Ph. D., professor. Her research interests include machine learning, learning to rank and information retrieval.

CHENG Xue-Qi, Ph. D., professor. His research interests include network science, network and information security, Web search and data mining.

Background

Ensemble learning has always been an important branch of machine learning. Just as under the cooperation of multiple organs, human beings can survive. Agents who are good at different tasks can form a reasonable system to solve more diverse problems. For classification problems, the multi-classifier system focuses on the "divide and conquer" concept. It decomposes the complex classification problem into multiple simple sub-problems, and uses a single model to break them one by one, and then reasonably combines these models to obtain a solution to the original problem.

Nowadays, multi-classifier ensemble algorithms play an important role in various tasks and are important methods to improve performance in machine learning competitions. However, in financial, security and some other tasks, the inference result is not convincing only by relying on the incomplete description like accuracy. The reason for the model's decision cannot be ignored.

The ensemble strategy used by the existing multi-classifier system cannot give specific reasons for decision-making during inference, and cannot explicitly give training samples that are effective in inference like *K*-nearest neighbors, collaborative filtering [and other models]. In fact, in the research of psychology, the knowledge-line memory theory explains that people will activate the memory generated by past data when making decisions, and use old examples as a basis to guess the answers to new questions. This paper explores this psychological process, and proposes a new multi-classifier ensemble strategy based on the knowledge-line memory theory, called the knowledge-line ensemble algorithm.

This algorithm has better interpretability than *K*-nearest neighbor algorithm on the basis of guaranteeing the classification evaluation accuracy. In inference, the model can provide similar samples in training as the basis for this inference. Not only that, during the training of the model, the process of increasing complexity and the changes in various aspects of capabilities are all explicitly displayed. In the training process, the model can extract the difficult points of the problem, such as a certain class of sample that is difficult to do right, and some subsets of categories that are easy to be confused. Due to the introduction of the knowledge-line, these difficult quantification is now available to be described by the activation, forgetting, and creation frequency of the knowledge-lines. These indicators improve the interpretability of the model, allowing users to conduct a more specific analysis of the problem.