Nearest neighbors distance ratio open-set classifier

Pedro R. Mendes Júnior ・Roberto M. de Souza・Rafael de O. Werneck ・Bernardo V. Stein ・Daniel V. Pazinato ・Waldir R. de Almeida ・Otávio A. B. Penatti ・Ricardo da S. Torres ・Anderson Rocha

摘要

近年来，开放集识别（Open Set Recognition，OSR）迅速发展成为热点领域，大量研究工作围绕 OSR 展开。其中，针对开放集识别问题，出现了基于支持向量机（Support Vector Machine，SVM）、基于稀疏表示、基于距离、基于边缘分布以及基于深度学习等开放集识别方法。本文课程的论文复现工作使用最近邻距离比 (Nearest Neighbors Distance Ratio，NNDR)，提出了开集最近邻（Open-Set Nearest-Neighbor，OSNN）识别方法，并主要针对基于 SVM 及基于距离的相关方法，分析 OSNN 方法的性能及其可行性，并对 OSNN 方法进行改进。

关键词：开放集识别；最近邻距离比；开放集识别评价指标

1 引言

随着机器学习技术的发展，机器学习被大量应用于识别领域。但无论是传统的机器学习方法还是深度学习方法，在实际识别和分类任务中，在训练识别器或分类器时需要大量标记样本作为支撑，而标记样本耗时耗力，加上真实场景变化莫测，通常很难收集包含所有类别的训练样本。受这些客观因素的限制，在训练过程中得到所有类别是不现实的。而在测试过程中一些没有见过的情况会出现，即开放环境。为了解决这个问题，开放集识别研究应运而生。

开放集识别不仅要求模型能正确分类已知类（减少经验风险），同时能够准确识别出新出现的类的开放集识别（减少开放空间风险）[1]，开放集识别问题更贴近真实场景，具有研究意义。目前，针对开放集识别问题，出现了基于支持向量机、基于稀疏表示、基于距离、基于边缘分布以及基于深度学习等开放集识别方法[2]。

本文课程的论文复现工作针对基于距离的开放集识别方法，使用最近邻距离比 NNDR 进行开放集识别，研究最近邻开集识别方法 OSNN 的可行性。

2 相关工作

此部分对本文中提及的基于 SVM 的开放集识别方法以及基于最近邻（Nearest Neighbor，NN）的开放集识别方法进行概括和描述。

2.1 基于 SVM 的开放集识别方法

Scheirer 等人[3]在定义开放集风险的基础上提出了一种新的机制 1-Versus-Set（oneVS，1VS）。该算法基于带有线性核函数 SVM 算法，通过约束已知类占据的空间减小开放空间风险。具体地，该算法在核空间中构造另一个超平面，该超平面与 SVM 形成的超平面平行，两个超平面之间会形成一定距离的空间。在这种情况下，出现在两个超平面之间的测试样本将被标记为对应的已知类，否则，它将被视为非目标类或被拒绝，这取决于它位于两个超平面的哪一侧。

Scheirer 等人[4]尝试将非线性核函数引入算法，通过用有限测度对样本进行标记来进一步限制开放空间风险。具体地，针对 OSR 问题提出紧凑衰减概率（Compact Abating Probability，CAP）模型，在该模型中，越靠近开放空间的样本属于此已知类的概率越低。同时，利用 CAP 模型和 EVT 理论进行概率估计，提出韦伯校准 SVM（Weibull-calibrated SVM，WSVM)。该方法由一元 SVM 和二元 SVM 组成，样本首先经过第一个结合 CAP 的一元 SVM 模型，产生一个后验概率。如果这个概率小于阈值，样本就会被拒绝；否则，样本会经过第二个结合 CAP 的二元 SVM，产生一个对应已知类正样本的后验概率（基于韦伯分布）和一个已知类负样本的后验概率（基于逆韦伯分布），最后样本会被识别为两个概率乘积最大对应的类别。

2.2 基于 NN 的开放集识别方法

Bishop[5]针对闭集环境，提出了最近邻分类器 NN，用于将样本识别为最近邻样本对应的类别。NN 方法只能用于闭集环境，为了实现开放集识别，TNN（NN using Threshold）方法在 NN 的基础上设置了一个阈值，当待测样本与最近邻样本的距离大于该阈值时，则将该待测样本划分为未知类。

Junior 等人在提出 OSNN 之前提出了 OSNN_CV 方法，该方法的思想是在预测阶段选出待测样本的两个最近邻，如果两个具有相同的标签，则把对应的标签分给该待测样本，否则该待测样本会被划分为未知类。

3 本文方法

3.1 本文方法概述

由于在 OSNN_CV 算法中，如果待测样本离训练样本很远，则无法被正确地识别为未知类，因此 Junior 等人提出了 OSNN 方法。该方法的特点在于不是直接对某一最相似类的相似性得分设置阈值，而是对两个最相似类的相似性得分比率设置阈值。分别计算待测样本到两个不同的最近邻 t 和 u 的欧式距离 $d(s,t)$ 及 $d(s,u)$，得到距离比 $R=\frac{d(s,t)}{d(s,u)}$，如果 R 小于阈值，则 s 被分为和 t 相同的标签，否则就会被划分为未知类，详见 4.2 节具体实现细节 Procedure 3。

3.2 参数优化

为了得到 OSNN 方法中的阈值，通过在训练阶段模拟开放集环境，即选择训练集中一部分可用训练类为“未知类”，并基于给定的阈值取值范围及在该开放集环境中识别的最优性能估计阈值，详见 4.2 节具体实现细节 Procedure 3。

4 复现细节

4.1 与已有开源代码对比

本复现工作参考了《A Nearest Neighbor Open-Set Classifier based on Excesses of Distance Ratios》相关源代码。具体来说，本复现工作参考了源代码中分批次处理数据集的方法，以及在 OSNN 的改进方法 KOSNN 的基础上进行了修改补充。此外，本复现工作完成了 NN、TNN、OSNN、OSNN_CV 方法以及各评价指标的计算的具体实现，并实现了 OSNN 方法的改进方法 OSNN_GPD。

4.2 具体实现细节

Procedure 1 构造开集环境

Input: training set *Train,* test set *Test,* the number of known classes *n*

Output: training set after relabeled *train,* test set after relabeled *test,* known classes *ref*

1: Randomly select n known classes from the training set.
2: Remove samples that do not belong to the selected known class from the training set.
3: According to the selected known classes, remap the labels of training set samples to obtain the training set $train$ that contains only known classes.
4: According to the selected known classes, remap the labels of test set samples to obtain the test set $test$ that the unknown samples are relabeled as -99999 .

Procedure 2 计算识别评价指标

Input: the label set *label,* the prediction label set *prediction,* known classes *ref,* regulation constant *lamda*

Output: the accuracy on known samples *AKS,* the error rate on recognition of known samples as other known classes *MIS,* the error rate on recognition of known samples as unknown classes *FU,* the accuracy on unknown samples *AUS,* the normalized accuracy *NA,* micro-averaging open-set f-measure *micrF1,* macro-averaging open-set f-measure *macroF1*

1: Compute $AKS = \frac{\text{the total number of correctly identified samples for all known classes}}{\text{the number of the known classes samples}}$.
2: Compute $MIS = \frac{\text{the total number of incorrectly identified as other known classes}}{\text{the number of the the known classes samples}}$.
3: Compute $FU = \frac{\text{the total number of incorrectly identified as unknown classes}}{\text{the number of the the known classes samples}}$.
4: Remove samples that do not belong to the selected known class from the training set.
5: Compute $AUS = \frac{\text{the total number of correctly identified samples for unknown classes}}{\text{the number of the unknown classes samples}}$.
6: According to the $lamda$, Compute $NA = lamda \times AKS + (1 - lamda) \times AUS$.
7: **for** *each known class* i **do**
8: Compute the number of correctly identified samples $TP[i]$.
9: Compute the number of incorrectly identified samples $FN[i]$.
10: Compute the number of samples incorrectly identified as class i by other classes $FP[i]$.
11: Compute micro precision $micro\ precision = \frac{\sum_{i=1}^{n} TP[i]}{\sum_{i=1}^{n} TP[i] + FP[i]}$.
12: Compute micro recall $micro\ recall = \frac{\sum_{i=1}^{n} TP[i]}{\sum_{i=1}^{n} TP[i] + FN[i]}$.
13: Compute $microF1 = 2 * \frac{micro\ precision \times micro\ recall}{micro\ precision + microrecall + e^{-08}}$.
14: Compute macro precision $macro\ precision = \frac{\sum_{i=1}^{n} \frac{TP[i]}{TP[i] + FP[i]}}{\text{the number of the the known classes samples}}$.
15: Compute macro recall $macro\ recall = \frac{\sum_{i=1}^{n} \frac{TP[i]}{TP[i] + FN[i]}}{\text{the number of the the known classes samples}}$.
16: Compute $macroF1 = 2 * \frac{macro\ precision \times macro\ recall}{macro\ precision + macro\ recall + e^{-08}}$.

Procedure 3 OSNN 开放集识别方法

Input: training set *train,* test set *test,* known classes *ref,* batch size *batch_ size,* threshold range *T,* regulation constant *lamda*

Output: the accuracy on known samples *AKS(OSNN),* the accuracy on unknown samples *AUS(OSNN),* the normalized accuracy *NA(OSNN),* micro-averaging open-set f-measure *micrF1(OSNN),* macro-averaging open-set f-measure *macroF1(OSNN)*

for *each fold of k-fold cross validation* **do**
    Divide the training set into fitting set and validation set.
    for *each sample in the validation set* **do**
        Compute the distance ratio of this validation sample $distance\ ratio$.
    for *each threshold* t *in the threshold range* **do**
        if $distance\ ratio > t$ **then**
            the sample is classified as unknown.
        else
            the sample is classified with the same label of y_pred.
        According to the label and the prediction label of the validation samples, using the $Procedure2$ to compute the normalized accuracy with $lamda$.
Take the threshold corresponding to optimal average of the k NA as the optimal threshold opt_t.
for *each sample* s *in the test set* **do**
    Compute the distance ratio of s as R.
    if $R > opt\_t$ **then**
        the sample is classified as unknown.
    else
        the sample is classified with the same label of its nearest neighbor.
According to the label and the prediction label of the test samples, using the $Procedure2$ to compute the evaluation measures with $lamda$.

Procedure 4 OSNN_CV 开放集识别方法

Input: training set *train,* test set *test,* known classes *ref,* batch size *batch_ size,* threshold range *T,* regulation constant *lamda*

Output: the accuracy on known samples *AKS(OSNN_CV),* the accuracy on unknown samples *AUS(OSNN_CV),* the normalized accuracy *NA(OSNN_CV),* micro-averaging open-set f-measure *micrF1(OSNN_CV),* macro-averaging open-set f-measure *macroF1(OSNN_CV)*

for *each sample* s *in the test set* **do**
    Get the nearest neighbor sample t and the label of t as t_label.
    Get the nearest neighbor sample u of s that does not belong to t_label and the label of u as u_label.
    if $t\_label = u\_label$ **then**
        s is classified as the same class with t.
    else
        s is classified as unknown.
According to the label and the prediction label of the test samples, using the $Procedure2$ to compute the evaluation measures with $lamda$.

Procedure 5 NN 开放集识别方法

Input: training set *train,* test set *test,* known classes *ref,* batch size *batch_ size,* threshold range *T,* regulation constant *lamda*

Output: the accuracy on known samples *AKS(NN),* the accuracy on unknown samples *AUS(NN),* the normalized accuracy *NA(NN),* micro-averaging open-set f-measure *micrF1(NN),* macro-averaging open-set f-measure *macroF1(NN)*

for *each sample* s *in the test set* **do**
    Get the nearest neighbor sample t and the label of t as t_label.
    Classify s with the same class with t.
According to the label and the prediction label of the test samples, using the $Procedure2$ to compute the evaluation measures with $lamda$.

Procedure 6 TNN 开放集识别方法

Input: training set *train,* test set *test,* known classes *ref,* batch size *batch_ size,* threshold range *T,* regulation constant *lamda*

Output: the accuracy on known samples *AKS(TNN),* the accuracy on unknown samples *AUS(TNN),* the normalized accuracy *NA(TNN),* micro-averaging open-set f-measure *micrF1(TNN),* macro-averaging open-set f-measure *macroF1(TNN)*

for *each fold of k-fold cross validation* **do**
    Divide the training set into fitting set and validation set.
    for *each sample in the validation set* **do**
        Get its nearest neighbor sample and Euclidean distance $distance$.
    for *each threshold* t *in the threshold range* **do**
        if $distance > t$ **then**
            the validation sample is classified as unknown.
        else
            the sample is classified with the same label of its nearest neighbor sample.
        According to the label and the prediction label of the validation samples, using the $Procedure2$ to compute the normalized accuracy with $lamda$.
Take the threshold corresponding to optimal average of the k NA as the optimal threshold opt_t.
for *each sample* s *in the test set* **do**
    Get its nearest neighbor sample t and Euclidean distance $d(s,t)$.
    if $d(s,t) > opt\_t$ **then**
        the sample is classified as unknown
    else
        the sample is classified with the same label of t
According to the label and the prediction label of the test samples, using the $Procedure2$ to compute the evaluation measures with $lamda$.

4.3 创新点

本复现工作参考了 Matthys Lucas Steyn 等人[6]的工作，对 OSNN 方法进行改进，实现了 OSNN_GPD 方法。具体来说，OSNN_GPD 方法利用训练集计算出训练集内各样本的最近邻距离比 ratio 后，利用极值理论（Extreme Value Theory，EVT）中的广义帕累托分布（Generalized Pareto Distribution，GPD）对最近邻距离比分布 R 的右尾部（upper tail）进行建模[7]。在计算出待测样本的最近邻距离比后，根据最近邻距离比分布 R 的右尾部模型，计算出分布 R 内大于该待测样本的最近邻距离比的概率 $P(R > ratio)$，若 $P(R > ratio)$ 大于给定的概率阈值 α（人为指定），则该待测样本被识别为其最近邻样本对应的类别；否则，该样本将被识别为未知类。

其中，样本的最近邻距离比的计算方法为：计算该样本与训练集内各样本间的欧式距离；将计算得到的欧氏距离从小到大进行排序；选择该样本的最近邻及对应的欧式距离 nume；再得到该样本的不属于其最近邻对应类别的近邻样本及对应的欧式距离 denom；即可计算该样本的最近邻距离比 $ratio = \frac{nume}{denom + e^{-09}}$。

此外，OSNN_GPD 方法基于给定的超参数范围，根据广义帕累托分布的拟合优度和模型的分类性能来选择超参数 t（广义帕累托分布的建模阈值）。并基于最优超参数 t，通过迭代极大似然估计法[8]估计了广义帕累托分布的参数 τ 和 γ。

Procedure 7 OSNN_GPD 开放集识别方法

Input: training set *train,* test set *test,* known classes *ref,* batch size *batch_ size,* threshold percentile range *T,* regulation constant *lamda,* probability threshold *α,* regulation constant in objective function *penalty*

Output: the accuracy on known samples *AKS(OSNN_GPD),* the accuracy on unknown samples *AUS(OSNN_GPD),* the normalized accuracy *NA(OSNN_GPD),* micro-averaging open-set f-measure *micrF1(OSNN_GPD),* macro-averaging open-set f-measure *macroF1(OSNN_GPD)*

for *each fold of k-fold cross validation* **do**
    Divide the training set into fitting set and validation set.
    Get the distribution of distance ratio R based on the distance ratios of fitting set.
    for *each threshold percentile in* T **do**
        Compute the modeling threshold t based on the distance ratios of fitting set.
        Compute the parameter of generalized Pareto distribution, using the iterative maximum likelihood procedure and based on the values that the distance ratios of fitting set exceeds t.
        for *j=1 to the number of samples in validation set* **do**
            Compute s_j as the value that its distance ratio r exceeds t and the class of its nearest neighbor.
            Compute $Q(\frac{j}{ne+1})$ as the quantiles of the generalized Pareto distribution at probability $\frac{j}{ne+1}$, where ne is the number of samples in validation set.
            Compute the probability $P(R > r)$ of variable greater than r in distribution R.
            if $P(R > r) < \alpha$ **then**
                the sample is classified as unknown.
            else
                the sample is classified with the same label of the class of its nearest neighbor.
        Compute $C(t)$ as the Pearson correlation coefficient between the set of s_j and $Q(\frac{j}{ne+1})$.
        Compute $E(t)$ as the classification error of the known classes after open set recognition.
        Compute objective function $O(t) = (1 - penalty) \times C(t) - penalty \times E(t)$.
Take the threshold corresponding to optimal average of the k $O(t)$ as the optimal threshold opt_t.
Compute the parameter of generalized Pareto distribution, using the iterative maximum likelihood procedure and based on opt_t.
for *each sample in the test set* **do**
    Compute r^* the value its distance ratio exceeds opt_t and the class of its nearest neighbor sample.
    Compute the probability $P(R > r^*)$ of variable greater than r^* in distribution R.
    if $P(R > r^*) < \alpha$ **then**
        the sample is classified as unknown.
    else
        the sample is classified with the same label of the class of its nearest neighbor.
According to the label and the prediction label of the test samples, using the $Procedure2$ to compute the evaluation measures with $lamda$.

5 实验结果分析

本部分对实验所得结果进行分析，详细对实验内容进行说明，实验结果进行描述并分析。

5.1 实验数据集

本实验在 Ukbench、Caltech-256、15-Scenes、ALOI、Auslan、Letter6 个数据集上进行开放集识别。

5.2 实验评价指标

本实验主要根据已知类识别准确率 AKS、已知类识别为其他已知类的错误率 MIS，已知类识别为未知类的错误率 FU，未知类识别准确率 AUS，归一化准确度 NA，微观平均开放集 F 度量 micrF1(OSFM_μ)，宏观平均开放集 F 度量 macroF1(OSFM_M)，详见 4.2 节具体实现细节 Procedure 2。

5.3 实验结果

图 1-6 描述了 1VS、WSVM、NN、TNN、OSNN_CV 以及 OSNN 方法在 15-Scenes、Caltech-256、Ukbench、ALOI、Auslan、Letter 6 个数据集上的性能。此外，图 4-6 还描述了 OSNN_GPD 方法在 ALOI、Auslan、Letter 数据集上的性能。

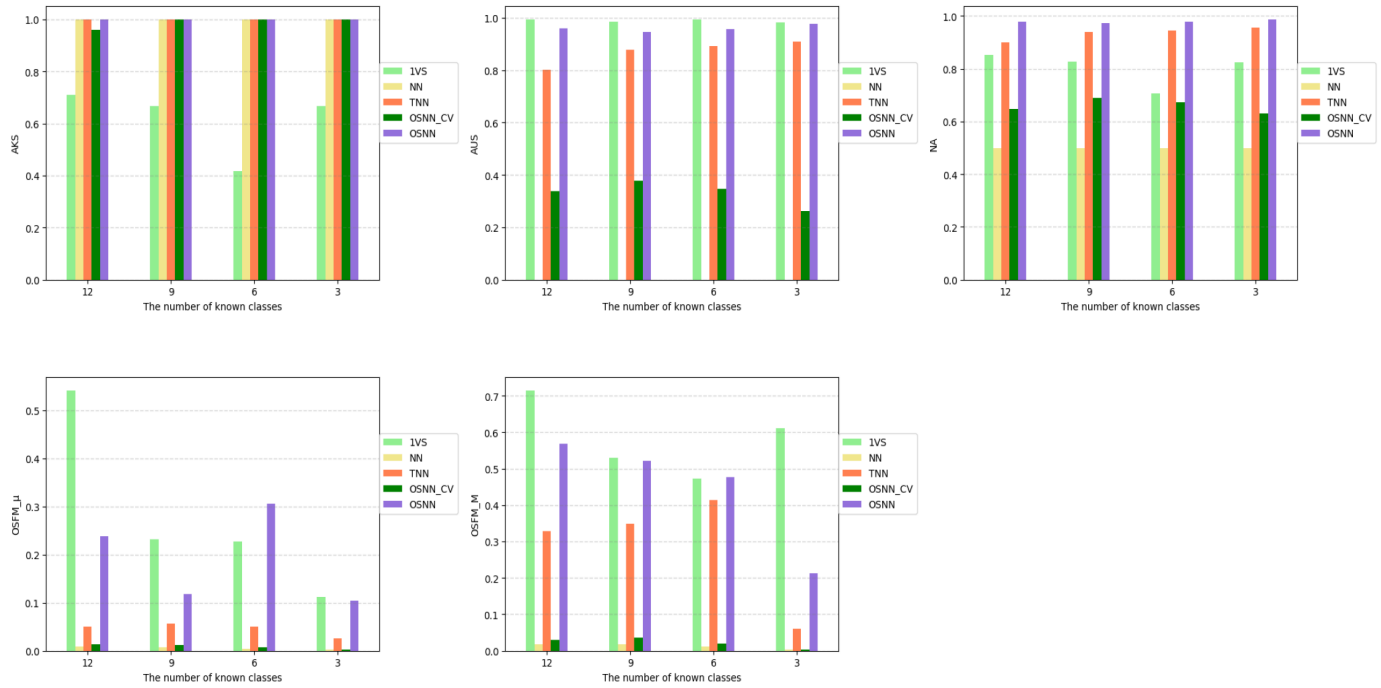

图 1: Ukbench 数据集上各算法的实验结果

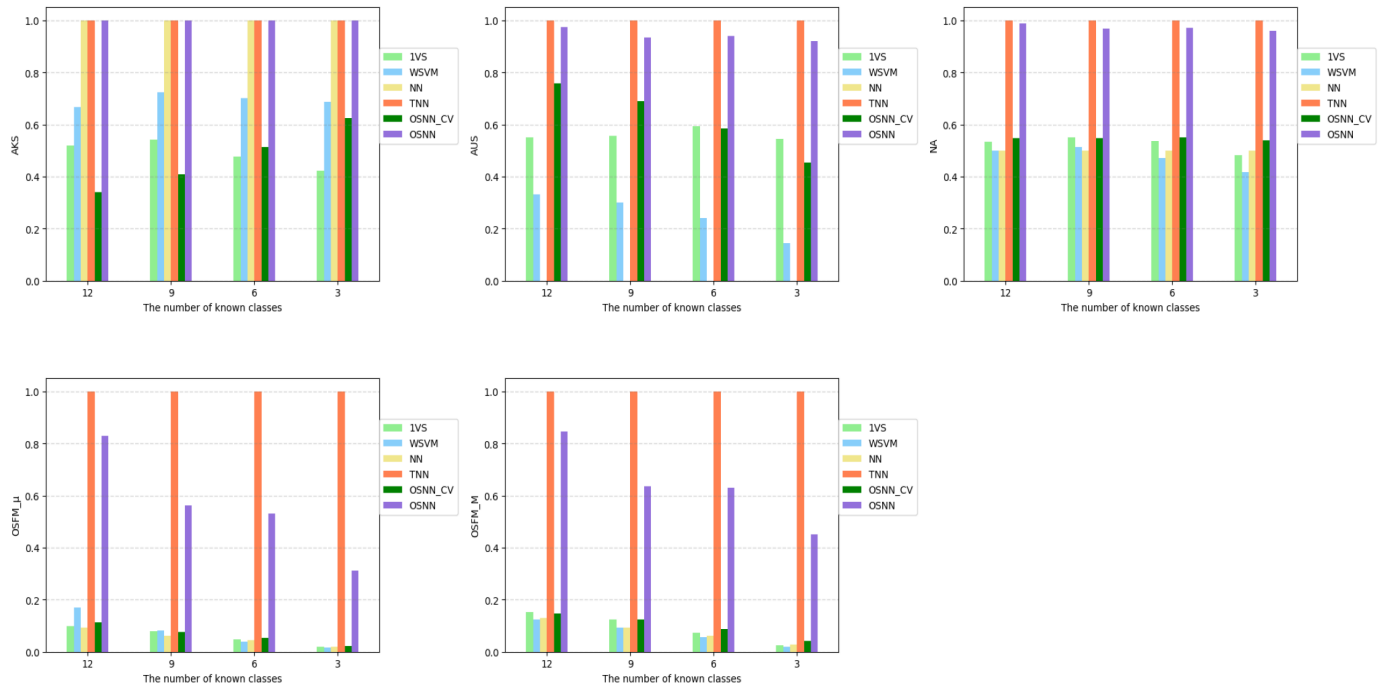

图 2: Caltech-256 数据集上各算法的实验结果

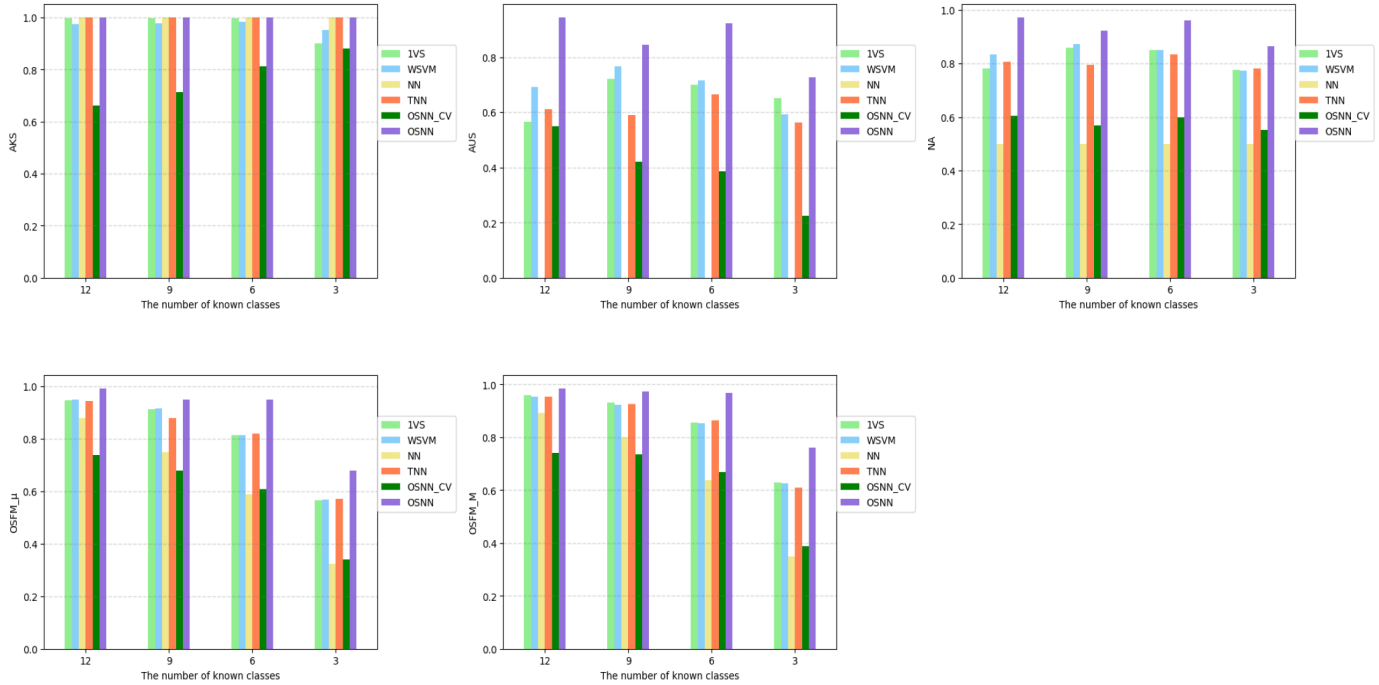

图 3: 15-Scenes 数据集上各算法的实验结果

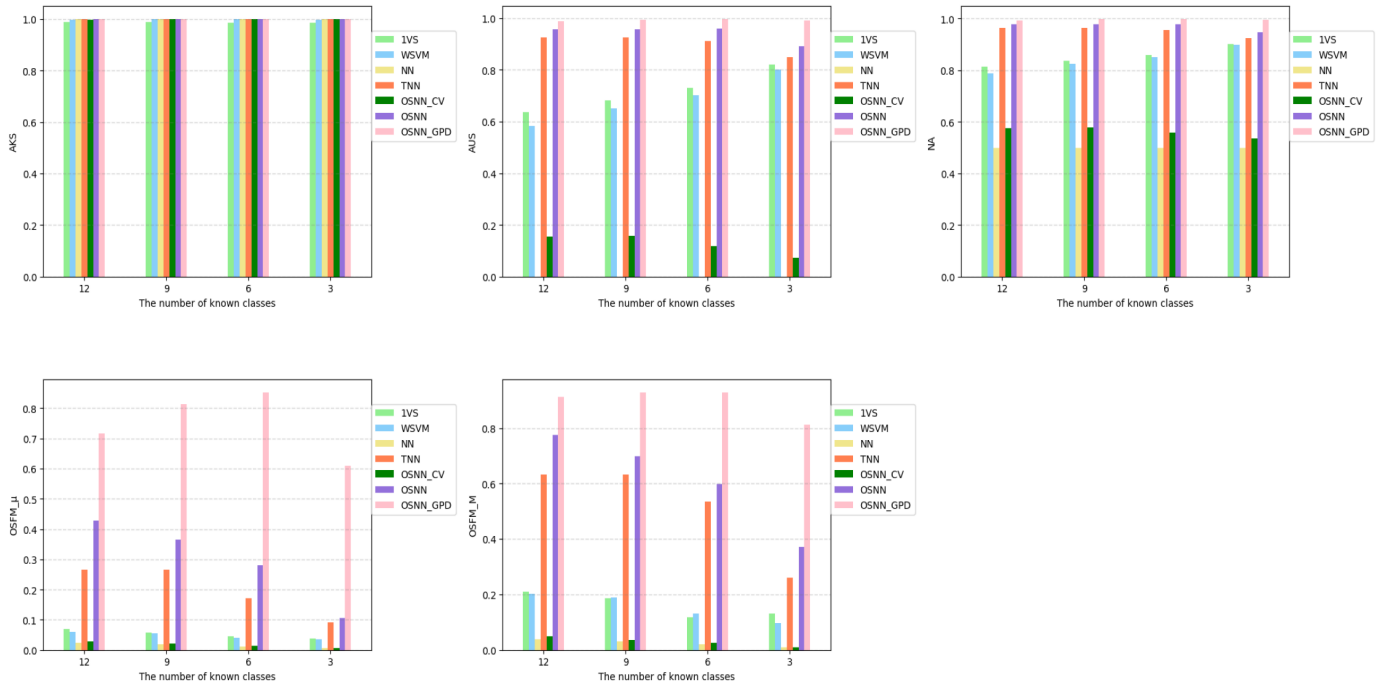

图 4: ALOI 数据集上各算法的实验结果

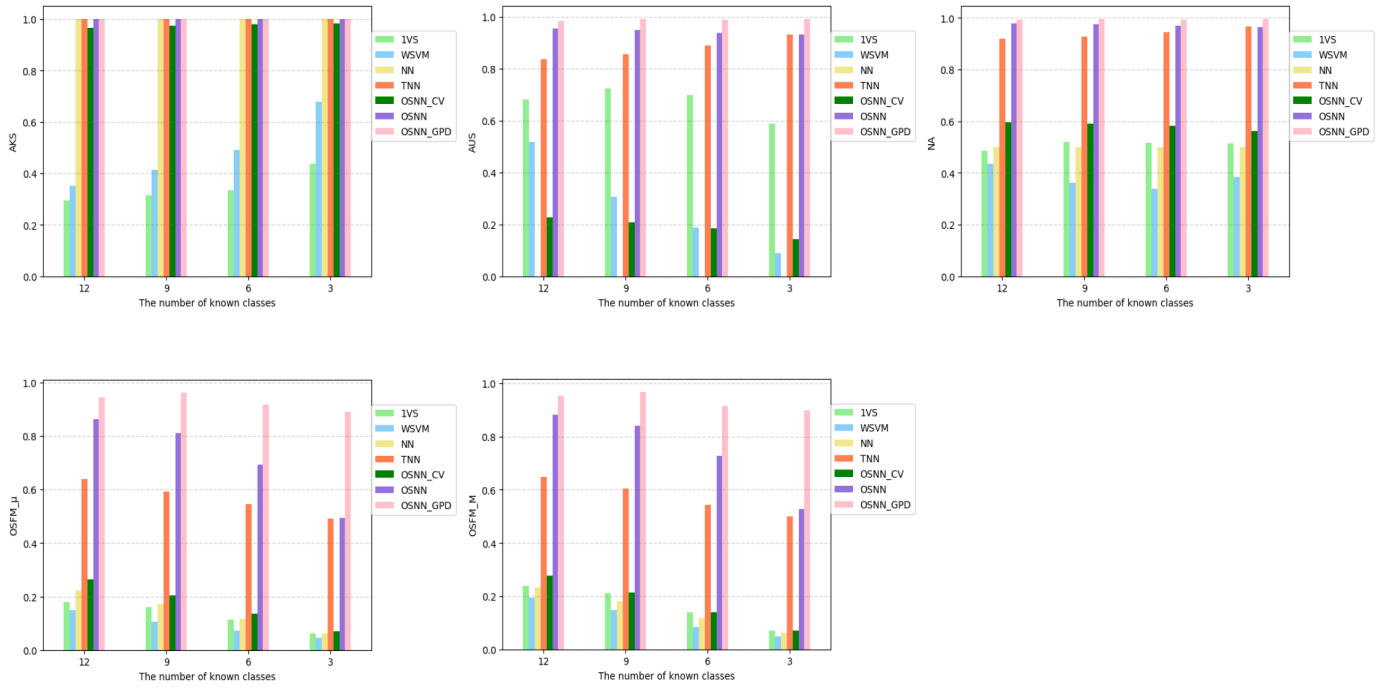

图 5: Auslan 数据集上各算法的实验结果

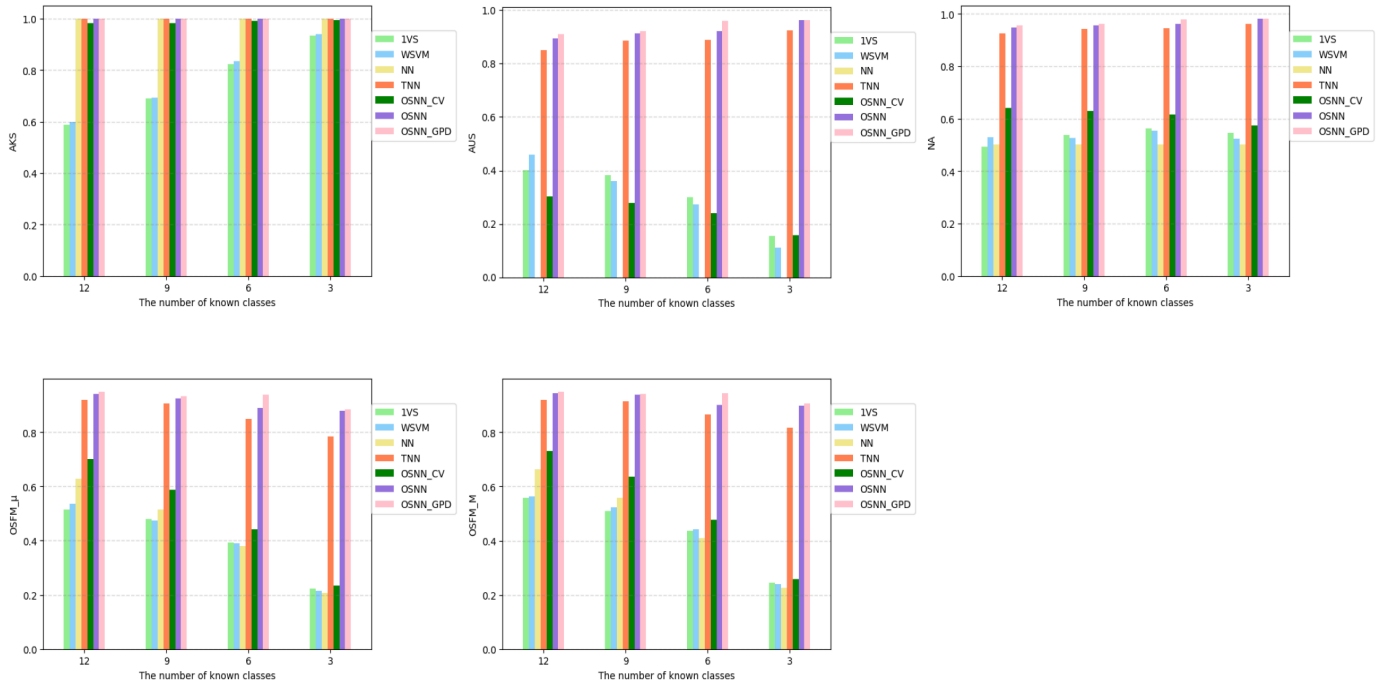

图 6: Letter 数据集上各算法的实验结果

总的来说，从图 1 可以看出，在 Ukbench 数据集上，OSNN 的方法相对于其他的基准算法在归一化准确度 NA 上性能要更好，在微观平均开放集 F 度量以及宏观平均开放集 F 度量上性能要相对 1VS 方法要差；从图 2 可以看出，在 Caltech-256 数据集上，OSNN 的方法的性能不如 TNN 算法；从图 3-6 可以看出，在 15-Scenes、ALOI、Auslan、Letter 数据集上，OSNN 的方法能比其他的基准算法的性能要好。

针对 OSNN 的改进方法 OSNN_GPD，从图 4-6 可以看出，其性能在 ALOI、Auslan、Letter 数据集上相对于 OSNN 方法有所改进。

	1VS	WSVM	NN	TNN	OSNN_CV	OSNN_GPD
AKS	w w w w	w w w w	. . . .	. . . .	w w w w	l l l l
AUS	w w w w	w w w w	w w w w	w w w w	w w w w	l l l l
NA	w w w w	w w w w	w w w w	w w w w	w w w w	l l l l
OSFM_μ	w w w w	w w w w	w w w w	w w w l	w w w w	l l l l
OSFM_M	w w w w	w w w w	w w w w	w w w l	w w w w	l l l l

表 1: 对所有数据集进行各评价指标的统计检验，并将 OSNN 方法与其他基准方法进行比较。其中每一个单元格显显示了 12、9、6 和 3 个已知类的情况。w 表示 OSNN 有较好的性能，l 表示 OSNN 有较差的性能，. 表示 OSNN 方法与列表中对应的方法没有统计学差异。

从表 1 可以看出，从总体来看，OSNN 方法在 AKS、AUS、NA、OSFM_μ、OSFM_M 上高于所有的基准方法。另一方法，作为改进，OSNN_GPD 方法则在各性能上均优于 OSNN 方法。

6 总结与展望

本部分对整个文档的内容进行归纳并分析目前实现过程中的不足以及未来可进一步进行研究的方向。OSNN 算法的阈值是由人为提供参数的取值范围，通过遍历取值范围内的每一可能取值，根据在该取值下算法的识别性能选择最优的参数，在这种情况下选定的阈值没有概率解释。

作为改进，OSNN_GPD 给定了区分已知类和未知类的概率阈值 α，如果一个样本来自未知类的概率低于 α，则将其划分为未知类，这为区分已知类和未知类提供了一种直观的方法。

但是，另一方面，算法内参数的选择可以进行自适应，而不是由人工提供参数范围。

参考文献

[1] 高菲，杨柳，李晖. 开放集识别研究综述[J]. 南京大学学报（自然科学）, 2022, 58(1): 115-134.

[2] GENG C, et al. Recent Advances in Open Set Recognition: A Survey[J]. Transactions on Pattern Analysis and Machine Intelligence, 2020, 43(10): 3614-3631.

[3] SCHEIRER W J, et al. Towards open set recognition[J]. Transactions on Pattern Analysis and Machine Intelligence, 2013, 35(7): 1757-1772.

[4] SCHEIRER W J, et al. Probability models for open set recognition[J]. Transactions on Pattern Analysis and Machine Intelligence, 2014, 36(11): 2317-2324.

[5] BISHOP C M. Pattern recognition and machine learning (1st ed.)[M]. Springer, 2006.

[6] STEYN M L, et al. A Nearest Neighbor Open-Set Classifier based on Excesses of Distance Ratios[J]. Journal of Computational and Graphical Statistics, 2022: 1-10.

[7] VIGNOTTO E, ENGELKE S. Extreme Value Theory for Open Set Classification –GPD and GEV Classifiers[Z]. https://arxiv.org/abs/2110.11334. 2021.

[8] DEL CASTILLO J, SERRA I. Likelihood inference for generalized Pareto distribution[J]. Computational Statistics & Data Analysis, 2015, 83: 116-128.