Universidade Estadual de Campinas
Instituto de Computação

Pedro Ribeiro Mendes Júnior

Open-set recognition for different classifiers

Reconhecimento em cenário aberto para diferentes classificadores

CAMPINAS
2018

Pedro Ribeiro Mendes Júnior

Open-set recognition for different classifiers

Reconhecimento em cenário aberto para diferentes classificadores

Tese apresentada ao Instituto de Computação da Universidade Estadual de Campinas como parte dos requisitos para a obtenção do título de Doutor em Ciência da Computação.

Dissertation presented to the Institute of Computing of the University of Campinas in partial fulfillment of the requirements for the degree of Doctor in Computer Science.

Supervisor/Orientador: Prof. Dr. Anderson de Rezende Rocha

Este exemplar corresponde à versão final da Tese defendida por Pedro Ribeiro Mendes Júnior e orientada pelo Prof. Dr. Anderson de Rezende Rocha.

CAMPINAS
2018

Agência(s) de fomento e nº(s) de processo(s): CAPES; CNPq, 140468/2018-8
ORCID: https://orcid.org/0000-0001-8086-018X

Ficha catalográfica
Universidade Estadual de Campinas
Biblioteca do Instituto de Matemática, Estatística e Computação Científica
Ana Regina Machado - CRB 8/5467

M522o

Mendes Júnior, Pedro Ribeiro, 1990-
Open-set recognition for different classifiers / Pedro Ribeiro Mendes Júnior. – Campinas, SP : [s.n.], 2018.

Orientador: Anderson de Rezende Rocha.
Tese (doutorado) – Universidade Estadual de Campinas, Instituto de Computação.

1. Aprendizado de máquina. 2. Reconhecimento de padrões. 3. Reconhecimento em cenário aberto. 4. Máquina de vetores de suporte. 5. Redes neurais (Computação). I. Rocha, Anderson de Rezende, 1980-. II. Universidade Estadual de Campinas. Instituto de Computação. III. Título.

Informações para Biblioteca Digital

Título em outro idioma: Reconhecimento em cenário aberto para diferentes classificadores
Palavras-chave em inglês:
Machine learning
Pattern recognition
Open set recognition
Support vector machines
Neural networks (Computer science)
Área de concentração: Ciência da Computação
Titulação: Doutor em Ciência da Computação
Banca examinadora:
Anderson de Rezende Rocha [Orientador]
Hélio Pedrini
Fernanda Alcântara Andaló
Roberto Hirata Junior
André Carlos Ponce de Leon Ferreira de Carvalho
Data de defesa: 14-09-2018
Programa de Pós-Graduação: Ciência da Computação

Universidade Estadual de Campinas
Instituto de Computação

Pedro Ribeiro Mendes Júnior

Open-set recognition for different classifiers

Reconhecimento em cenário aberto para diferentes classificadores

Banca Examinadora:

- Prof. Dr. Anderson de Rezende Rocha
  Instituto de Computação,
  Universidade Estadual de Campinas
- Prof. Dr. Hélio Pedrini
  Instituto de Computação,
  Universidade Estadual de Campinas
- Dra. Fernanda Alcântara Andaló
  Instituto de Computação,
  Universidade Estadual de Campinas
- Prof. Dr. Roberto Hirata Junior
  Instituto de Matemática e Estatística,
  Universidade de São Paulo
- Prof. Dr. André Carlos Ponce de Leon Ferreira de Carvalho
  Instituto de Ciências Matemáticas e de Computação,
  Universidade de São Paulo

A ata da defesa, assinada pelos membros da Comissão Examinadora, consta no SIGA/Sistema de Fluxo de Dissertação/Tese e na Secretaria do Programa da Unidade.

Campinas, 14 de setembro de 2018

Ao meu querido avô que,
com seus problemas matemáticos de *Malba Tahan*,
estimulava na mente infantil de um menino da roça,
os pensamentos que viriam a mudar o seu destino.

Acknowledgements

It has been a long and nice journey. The nice part is thanks to the people along the way. And the long is how it should be when it is intense and pleasant.

My main thanks is to my advisor, Prof. Anderson Rocha, who has been attentive and available for talking about our work, showing kind aspects of tolerance and active patience, throughout this four-year period, as well as commitment and interest on the research.

I also remember Prof. Terrance Boult with admiration and thank him for the great advisement and support for the development of part of this work. It was him who encouraged me to further develop the Specialized Support Vector Machines (SSVM) as an optimization problem. With the support of his guidance, we could accomplish this work during a period in which I learned much. I also thank Prof. Boult for all the great talks we had and for sharing some of his ideas with me, which have broadened my vision as a researcher. He and his wife, Ginger Boult, were very kindly along my one-year stay in Colorado Springs. I remember with joy great moments of hiking, skiing, snowboarding, and a lot more, not to mention their sympathetic willingness on helping a foreign student in everyday situations when necessary. Also, many thanks to my colleagues in the VAST lab[†] for the eventual chats and also the skiing times.

Many thanks to my colleague and friend Bernardo Stein. SSVM was born in a late night of study in the lab. "What defines which class SVM is going to bound?"—I asked him. "Probably the bias term"—Bernardo guessed it correctly. "I don't think so"—my statement. In fact, later we confirmed the bias as the factor, which lead to the development of a prominent part of this work.

I feel quite fortunate of being part of the RECOD lab,[‡] in which I have met many friendly and cordial colleagues. This research lab have been growing in recent times and I hope it continues to improve its qualities in all aspects.

The scholarship support of CAPES and CNPq[*] is highly appreciated as a fundamental support for the normal development of this research with no hindrances. I am grateful for this opportunity. I also thank all the staff team of the Institute of Computing for the promptness, the inclination for helping when necessary, and the permanent good mood.

For the ones not mentioned in this note, which are also part of this four-year story, I prefer to avoid expressing my reverence herein, as my feelings require this discreet act. I keep them in my recollection with joy.

[†]Vision and Security Technology (VAST) lab, University of Colorado Colorado Springs (UCCS).

[‡]Reasoning for Complex Data (RECOD) lab, IC, UNICAMP.

[*]This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior—Brasil (CAPES)—Finance Code 001. Also financed in part by Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) through process № 140468/2018-8.

Resumo

Neste trabalho, estudamos e analisamos o problema de reconhecimento em cenários abertos no contexto de diversos tipos de métodos de reconhecimento de padrões: baseados em distância, geométricos e redes neurais. O problema de reconhecimento em cenário aberto apresenta particularidades extras a serem tratadas, quando comparado ao já bem estudado problema de classificação em cenários fechados. Em cenários abertos, o método de reconhecimento deve ser devidamente capaz de reconhecer e também rejeitar instâncias de classes desconhecidas, i.e., de classes não consideradas durante a etapa de treino. Por outro lado, métodos de classificação em cenários fechados assumem que qualquer instância apresentada para classificação sempre pertence a uma das classes conhecidas. Extensões triviais de métodos próprios para cenários fechados, usualmente baseadas em limiares de rejeição, não lidam bem com cenários abertos e esta é a razão principal pela qual este problema tem recebido maior atenção recentemente.

Nesta pesquisa, fizemos a hipótese de que limitar o espaço aberto classificado como conhecido seja uma propriedade requerida para um método de reconhecimento em cenários abertos. Isso significa que instâncias de teste fora do suporte das instâncias de treino, em uma região infinita do espaço de características, seriam devidamente rejeitadas como desconhecidas, sendo, consequentemente, o risco do desconhecido limitado. Nossos experimentos confirmam esta hipótese e mostramos como garantir esta propriedade em classificadores geométricos que, usualmente, definem semiespaços, i.e., potencialmente definem uma região ilimitada do espaço aberto classificada como conhecida. Além da abordagem trivial de aplicar um limiar à distância em si, também mostramos como melhor definir a região classificada como conhecida em classificadores baseados em distância. Além do mais, neste trabalho, realizamos uma análise perspicaz em redes neurais — que são inerentemente fechadas por design — com o objetivo de obter as mesmas propriedades com este tipo de classificadores em trabalhos futuros.

As análises e discussões apresentadas neste trabalho também têm o objetivo de definir conceitos e clarificar o problema de reconhecimento em cenários abertos. Há particularidades no problema às quais devemos estar atentos e que independem do tipo de classificadores empregados para resolvê-lo, como é o caso da análise de métodos de extensão de classificadores inerentemente binários para classificação multiclasse; a estratégia de busca por parâmetros própria para cenários abertos e as medidas de acurácia próprias para cenários abertos.

Abstract

In this work, we have studied and analyzed the open-set recognition problem from the context of multiple types of recognition methods, namely, distance-based, geometric and neural networks. Open-set recognition problems bring some extra particularities to handle compared to well-studied closed-set classification problems. In open-set scenarios, the recognition method must be able to properly recognize and also reject instances from unknown classes, i.e., classes never seen during training phase. On the other hand, closed-set classification methods assume that any instance presented for classification always belongs to one of the known classes. Trivial threshold-based extensions of closed-set methods do not handle well the open-set recognition scenario and that is the reason this problem has received more attention nowadays.

In the research, we had hypothesized that ensuring a bounded known-labeled open space is a required property for a recognition method in open-set scenarios. It means that test instances from outside the support of the training instances, on an infinity region of the feature space, would be properly rejected as unknown; consequently, the risk of the unknown would be limited. Our experiments confirm this hypothesis and we have shown how to accomplish this with geometric classifiers, that usually define half-spaces, i.e., possibly unbounded known-labeled open space, as well as with nearest neighbors classifiers, besides the trivial approach of thresholding the raw distance. Furthermore, in this work, we perform insightful analyses on neural networks—which is inherently closed by design—aiming at obtaining similar achievements for this type of methods in future work.

The analyses and discussion presented in this work also aim at defining concepts and clarifying the open-set recognition problem. There are peculiarities on the problem for which anyone should be attentive, independently of the type of classifiers employed for solving it, as is the case of the analysis of multiclass-from-binary extensions, open-set grid search strategy, and evaluation measures employed for open-set setups.

List of Figures

List of Tables

Acronyms

EVT	Extreme Value Theory 28, 81
FC	Fully Connected (FC) 17, 81, 84
IMQ	Inverse Multiquadric 36, 39
KLOS	Known-Labeled Open Space 9, 16, 19–22, 31–33, 39–41, 43, 46, 48, 54, 56, 59, 62–64, 69, 83, 84, 87, 99–101
MAV	Mean Activation Vector 81, 90
PLOS	Positively-Labeled Open Space 16, 19, 22, 28, 34–36, 38–41, 51, 54, 84, 100
RBF	Radial Basis Function 9, 27–29, 35–39, 42, 51, 54, 56, 99
ReLU	Rectified Linear Unit 80, 82, 84
RQ	Rational Quadratic 36, 39
TST	Generalized T-Student 36, 39

Evaluation measures

AKS	Accuracy on Known Samples 23, 24, 47, 48, 50, 51, 55, 56, 58, 59, 61, 62, 65, 66, 68, 121–125
AUS	Accuracy on Unknown Samples 23, 24, 47, 48, 50, 51, 55, 56, 58, 59, 61, 62, 65, 66, 68, 121–125
FM_M	Macro-averaging F-measure 47, 50, 51, 55, 56, 58, 59, 61, 65, 66, 68, 121–125
FM_μ	Micro-averaging F-measure 47, 50, 51, 55, 56, 58, 59, 61, 65, 66, 68, 121–125
HNA	Harmonic Normalized Accuracy 24, 47–53, 55, 57–68, 121–125
NA	Normalized Accuracy 23, 24, 28, 46, 47, 50, 51, 55, 58, 59, 61, 65, 66, 68, 121–125
$OSFM_M$	Macro-averaging Open-set F-measure 47, 50, 51, 55, 56, 58, 59, 61, 65, 66, 68, 121–125
$OSFM_\mu$	Micro-averaging Open-set F-measure 47, 50, 51, 55, 58, 59, 61, 65, 66, 68, 121–125

Recognition methods

CNN	Convolutional Neural Network 66, 81, 84, 87, 101
DBC	Decision Boundary Carving 27, 55, 56, 66, 69, 71–78, 122
DBC_C	DBC performing closed-set grid search 51, 121
DBC_O	DBC performing open-set grid search 58, 59, 66, 68, 122–125
FNN	Feedforward Neural Network 79, 80
MLP	Multilayer Perceptron 79–81, 87, 101
NN	Nearest Neighbor 16, 26, 30–32, 48
kNN	k-Nearest Neighbors 30, 31
OCSVM	One-Class Support Vector Machines 9, 16, 19, 26, 27, 34, 40, 43–45, 55, 69, 71–78, 122
$OCSVM_C$	OCSVM performing closed-set grid search 51, 121
$OCSVM_O$	OCSVM performing open-set grid search 58, 59, 66, 68, 122–125
OPF	Optimum-Path Forest 29, 30, 48
OSOPF	Open-Set Optimum-Path Forest 20, 22, 29, 48, 50, 121

OSOPF^{CV}	Open-Set Optimum-Path Forest Class Verification [48, 50, 121]
OSNN	Open-Set Nearest Neighbors [9, 11, 12, 16, 19, 20, 29–33, 46, 48–51, 56, 58, 59, 65, 69, 71–78, 99, 100, 121, 122, 124]
OSNN^{CV}	Open-Set Nearest Neighbors Class Verification [32, 33, 48, 50, 121]
$\text{OSNN}_{10}^{\lambda_r}$	Open-Set Nearest Neighbors trained with $\lambda_r = 10$ for NA [65, 124]
$\text{OSNN}_{30}^{\lambda_r}$	Open-Set Nearest Neighbors trained with $\lambda_r = 30$ for NA [65, 124]
$\text{OSNN}_{70}^{\lambda_r}$	Open-Set Nearest Neighbors trained with $\lambda_r = 70$ for NA [65, 124]
$\text{OSNN}_{90}^{\lambda_r}$	Open-Set Nearest Neighbors trained with $\lambda_r = 90$ for NA [65, 124]
OVS	1-vs-Set Machine [27, 51, 55, 56, 69, 71–78, 122]
OVS_{C}	OVS performing closed-set grid search [51, 121]
OVS_{O}	OVS performing open-set grid search [58, 59, 66, 68, 122–125]
PISVM	Support Vector Machines with Probability of Inclusion [22, 28, 29, 51, 55, 56, 66, 69, 71–78, 99, 122]
PISVM_{C}	PISVM performing closed-set grid search [51, 121]
PISVM_{O}	PISVM performing open-set grid search [58, 59, 66, 68, 122–125]
RNN	Recurrent Neural Network [80]
SSVM	Specialized Support Vector Machines [9, 11, 12, 16, 17, 19, 20, 34, 35, 38–41, 46, 48, 51–56, 59, 62, 66, 68–78, 99, 100, 115, 122]
SSVM_{C}	SSVM performing closed-set grid search [11, 12, 51, 121]
SSVM_{O}	SSVM performing open-set grid search [59, 66, 68, 123–125]
SVDD	Support Vector Data Descriptor [19, 26, 44, 55, 69, 71–78, 122]
SVDD_{C}	SVDD performing closed-set grid search [51, 121]
SVDD_{O}	SVDD performing open-set grid search [58, 59, 66, 68, 122–125]
SVM	Support Vector Machines [9, 16, 18, 21, 22, 25–27, 29, 34–44, 46, 48, 51, 54–56, 59–66, 69, 71–78, 84, 99, 100, 116, 122, 123]
SVM_{C}	SVM performing closed-set grid search [51, 121]
SVM_{O}	SVM performing open-set grid search [58, 59, 66, 68, 122–125]
SVM^{WB}	Support Vector Machines without bias term [56, 59, 61, 123]
$\text{SVM}_{6}^{\text{WB}}$	Support Vector Machines without bias term with artificial bias set to -1×10^{-6} [59, 61, 123]
$\text{SVM}_{1}^{\text{WB}}$	Support Vector Machines without bias term with artificial bias set to -1×10^{-1} [11, 12, 59, 61, 123]
SVM^{OVO}	Support Vector Machines with the one-vs-one approach [9, 62]
$\text{SVM}_{\ominus}^{\text{OVO}}$	Support Vector Machines with one-vs-one *unable* to bound the KLOS [62]
$\text{SVM}_{\oplus}^{\text{OVO}}$	Support Vector Machines with one-vs-one *able* to bound the KLOS [62]
TNN	Thresholded Nearest Neighbor [48]
TNN_{E}	Thresholded Nearest Neighbor with external grid search [48, 50, 121]
TNN_{I}	Thresholded Nearest Neighbor with internal grid search [48, 50, 121]
WSVM	Weibull-Calibrated Support Vector Machines [28, 47, 55, 56, 66, 69, 71–78, 122]
WSVM_{C}	WSVM performing closed-set grid search [51, 121]
WSVM_{O}	WSVM performing open-set grid search [58, 59, 66, 68, 122–125]

Nomenclature

ℓ_0	Unknown label.
ℓ_i	Label referring to class i.
λ	Regularization parameter.
m	Number of training instances.
m_i	Number of training instances from class i.
m_p	Number of positive training instances.
n	Number of known/training classes.
T	Decision threshold for rejection.
$\mathbf{x}$, $\mathbf{x}'$, $\mathbf{x}_i$	Training/testing samples (a.k.a. instances, data points, and examples). $\mathbf{x}_i$ represents the i-th training sample or the nearest training sample from the i-th nearest training class, depending on the context.
$\mathbb{R}^d$	d-dimensional feature space.
$d(\mathbf{x}, \mathbf{x}')$	Distance in the Euclidean space between data points $\mathbf{x}$ and $\mathbf{x}'$.
$f(\mathbf{x})$	Decision/recognition function of a classifier given a test instance $\mathbf{x}$.
K	Kernel function.
$\phi : \mathcal{X} \mapsto \mathcal{Z}$	Projection function such that $K(\mathbf{x}, \mathbf{x}') = \phi(\mathbf{x})^T \phi(\mathbf{x}')$.
$\theta(\mathbf{x})$	Ground-truth class of a training instance $\mathbf{x}$ such that $\theta(\mathbf{x}) \in \{\ell_1, \ldots, \ell_n\}$.
$\mathbf{a}^{[0]}$, $\mathbf{x}$	Input vector of a neural network.
$\mathbf{a}^{[l]}$	Activation vector of the l-th layer of a neural network.
$a_i^{[l]}$	Activation of the i-th unit of the l-th layer of a neural network.
$\hat{\mathbf{y}}$	Output vector of a neural network.
$\phi^{[l]}$	Activation function of the l-th layer of a neural network.
$L^{[l]}$	Set of units of the l-th layer of a neural network.
$w_{ji}^{[l]}$	Weight of the connection between j-th unit of layer l with i-th unit of layer $l-1$.

Contents

Chapter 1

Introduction

The literature of machine learning and pattern recognition is rich on closed-set classification methods, ranging from traditional distance-based and geometric methods to prominent neural networks, highly adopted nowadays. The open-set scenarios, inherently present in some of the real-world recognition problems, have been mainly treated in a per-instance basis instead of approaching the problem on the level of the classification methods. For that reason, the machine learning literature still has a lack of general purpose open-set recognition methods. Aiming at overcoming this deficit, this work is dedicated to a broad range of categories of recognition methods present in the literature: distance-based, geometric, and neural network classifiers.

The problem of recognition in open-set scenarios is characterized by the lack of information regarding the circumstances in which a trained recognition method would be employed. Solutions for open-set recognition must assume extraneous instances can be presented for classification, consequently, they should be properly rejected, i.e., recognized as not belonging to any of the classes with which the recognition method was trained. On the other hand, it is an inherent assumption for closed-set classifiers that any prediction can be safely assigned to the known class for which the method has more confidence about. Although it can be true in a more controlled scenario, any unexpected change on that scenario would make the classifier unreliable.

For this reason, there is a recent effort on establishing concepts that are inherent to the open-set problem, previously ignored when dealing with a strict closed-set one. One of the most important is the concept of *open space* [Scheirer et al., 2013], which refers to the region of the feature space outside the support of the training classes, i.e., region with lower probability of having a representative instance for one of the known classes. That is, the region in which no sample of any of the known classes is likely to appear. Most of the works on closed-set classifiers have neglected it, aiming at only obtaining the best possible separation on the region of the feature space among the known classes. For instance, Support Vector Machines (SVM) [Cortes and Vapnik, 1995] define half-spaces, which means that even when classifying an instance far away from any training sample, that sample will be detected as belonging to one class or the other, but not recognized as *none of them*. This SVM behavior can be considered even worse on open-set scenarios, if we consider that the further away a testing sample is from the hyperplane, the surer SVM is about its classification as belonging to a certain class. Other examples are the Neural

Networks, which are usually trained aiming at solely the highest confidence on predicting the learned instance at its target class, but neglects the possibility of the *unknown* through its learning process.

The concepts of positively-labeled open space (PLOS) and known-labeled open space (KLOS) arise to make explicit the problem of this behavior on defining half-spaces or, in a more general term, the problem of not bounding the KLOS. For binary classification, PLOS refers to the region of the open space in which a classifier ends up classifying an instance as positive. Similarly, KLOS applies to the multiclass level and refers to the region of the open space in which a recognition method would predict a test sample as belonging to one of the known classes. If a binary classifier predicts as positive any sample at an unbounded (infinity) region of the feature space, PLOS is unbounded as well, as the open space is potentially always unbounded. Despite the relationship of the concept of PLOS to binary classification, it has its main importance when considering extending binary classifiers for multiclass classification through the one-vs-all approach [Rocha and Goldenstein, 2014]. We can see that if a binary classifier is able to bound the PLOS for every classifier that composes a multiclass-from-binary classifier, then the region of the open space classified as known (KLOS) is bounded as well. We present more details regarding this point in Chapter 2.

A straightforward approach to take advantage of the amount of closed-set classifiers available in the literature for applications in open-set scenarios would be to apply thresholds on raw confidence scores calculated for each method. This approach, however, is not safe and cannot offer a good generalization. Furthermore, due to the curse of dimensionality, it is often not reliable.

Employing one-class methods is another straightforward alternative for handling the open-set recognition problem. For instance, One-Class Support Vector Machines (OCSVM) [Schölkopf et al., 2001] and Support Vector Data Descriptor (SVDD) [Tax and Duin, 1999a, 2004, Chang et al., 2013] are support vectors-based methods mainly designed for outlier detection. When employed with kernels [Boser et al., 1992], PLOS is bounded. However, employment of those methods are not well-suitable due to its specialization-generalization ability problem. As one-class methods do not consider other classes when generating their models, when employed with one-vs-all approaches to multiclass classification, it tends to generate a poor model on the decision boundaries among the known classes. Anyhow, as we shall see, recent methods targeted for open-set problems have considered one-class methods, along with the benefit of the discrimination ability of binary methods, for open-set recognition.

The methods we present in this thesis differ from those trivial approaches—as we name them—on bounding the KLOS as they try to keep a reasonable model for the known classes as well. The Open-Set Nearest Neighbors (OSNN), proposed in this work, relies its decision on a ratio of distances instead of raw distances themselves while the Specialized Support Vector Machines (SSVM), also proposed herein, optimizes a separation margin at the same time of ensuring a bounded PLOS for every binary classifier, later composed with a one-vs-all approach for multiclass classification.

Finally, we should notice an inherent problem that remains, when dealing with open-set setups, despite all the effort for bounding the PLOS/KLOS. The OSNN and the

SSVM, that will be presented in Chapters 4 and 5, respectively, are able to bound the KLOS on the given feature space. Like other methods with the same ability, they can only ensure a bounded KLOS on the feature space defined for training, which does not mean the input space—space of the raw data, e.g., images—would also be bounded. It is important to notice, mainly when contrasting the theoretical guarantees we have obtained with OSNN and SSVM with efforts for dealing with the open-set problem with neural networks. Further discussion regarding this point is presented in Chapter 7 along with the analysis for neural networks.

The OSNN we propose herein is a generalization of the Open-Set Optimum-Path Forest (OSOPF) proposed in our previous work [Mendes Júnior, 2014]. OSNN have recently been published [Mendes Júnior et al., 2017] and OSOPF has also been extended along with genetic programming [Neira et al., 2018]. Up to the date of publication of this thesis, SSVM is under review, however, preprint has been kept updated online [Mendes Júnior et al., 2018]. Furthermore, its source-code, extended from LIBSVM [Chang and Lin, 2011], is available at GitHub.[1]

As for the remaining chapters of this thesis, in Chapter 2, we present some important considerations about the open-set problem and define concepts that will be used throughout this work. In Chapter 3, we present previous work on open-set recognition. The distance-based OSNN and the geometric SSVM are presented in Chapters 4 and 5, respectively, along with additional considerations for each type of classifiers. We have decided to group the results for OSNN and SSVM methods in a single chapter, Chapter 6, due to their similarity on the experimental setup. On the other hand, for experiments with neural networks, we present them in Chapter 7 itself, along with their discussion. Finally, our general conclusion is presented in Chapter 8.

[1] SSVM source-code is available at `https://github.com/pedrormjunior/ssvm`.

Chapter 2

Considerations about the open-set scenario

In this chapter, we review important concepts related to the open-set recognition scenario that will enable a better understanding of this work. In Section 2.1, we consider one-vs-all and one-vs-one multiclass-from-binary approaches for extending binary classifiers for multiclass classification. The content of that section is important for the discussion we present in Chapter 5. When we consider forms of grid searching for parameters, as in Section 2.2, it can be applied to any type of classifiers. In Section 2.3, we consider evaluation measures properly designed for open-set scenarios.

2.1 Multiclass-from-binary approaches

From works in closed-set recognition, both one-vs-all and one-vs-one [Rocha and Goldenstein, 2009, 2014] are well-known approaches for extending inherently binary classifiers—e.g., SVMs—for multiclass classification. Each of those have its own advantage, but one-vs-one is usually preferable on closed-set scenarios due to their smaller training time and slightly improved accuracy compared to one-vs-all [Hsu and Lin, 2002]. It was for that reason one-vs-one was chosen to be implemented in LIBSVM [Chang and Lin, 2011]. However, this preference should change in an open-set scenario, as we shall consider.

The one-vs-one approach consists on decomposing the complete multiclass problem into $n(n-1)/2$ pairwise binary problems, for n training classes, so that each problem can be solved by a binary SVM. For the final decision, a *voting scheme* is employed and the most voted winning class is chosen to label the test sample. Each class appears in $n-1$ binary problems and, consequently, it can receive at most $n-1$ votes. The behavior of one-vs-one, along with the voting scheme, is closed-set and there is no straightforward extension for open-set problems. To the best of our knowledge, the idea of thresholding the number of votes for classifying a test instance as unknown has not been tested and published in any research. However, one would consider estimating probabilities for SVMs [Platt, 2000, Wu et al., 2004] and thresholding them for rejection, even with no guarantee on being able to bound the KLOS. We present more analysis regarding the one-vs-one approach with probability estimates in Section 5.3.2, however, for now, let us consider

the one-vs-all strategy for the open-set scenario.

The one-vs-all approach consists on training n binary problems. The positive class for each problem is one of the available classes for training and the negative class comprises all other $n-1$ remaining classes. This way, each class appears as positive in exactly one binary problem. A trivial extension of one-vs-all for the open-set scenario is to classify a test sample as unknown when all n binary classifiers classify a sample as negative. The rationale is that it indicates the instance is negative for every class, therefore unknown. In Chapter 6, we show some empirical evidence that justify the reasonable performance that SVM with one-vs-all approach can obtain in open-set scenarios.

Employing one-vs-all in the open-set scenario gives us further perspectives: if a binary classifier can ensure a bounded PLOS, the one-vs-all approach then can ensure a bounded KLOS. That is the main rationale of the method we present in Chapter 5: to be able to bound the PLOS.

2.2 Open-set grid search

In previous work, Jain et al. [2014] have proposed the *cross-class validation* targeted for Support Vector Machines with Probability of Inclusion (PISVM) method. Independently, in previous work [Mendes Júnior, 2014], we have defined a *parameter optimization* phase for OSOPF classifier and they both share the same principle. In summary, they consist of simulating the open-set scenario, with unknown classes on validation so that obtained fitting parameters are suitable for the open-set scenario that appears on testing.

In this work, we formalize this method—we call it open-set grid search—aiming at its general employment along with any classifier targeting open-set scenarios. First, consider two possible well-known alternatives for performing grid search for parameters. One grid searches for *individual parameters* for each binary model of a one-vs-all or one-vs-one composition. The other one grid searches for parameters so that all binary models *share the same parameters*. We call them *internal* and *external grid search*, respectively—notice they are already well-known in the literature [Chang and Lin, 2011], however with no name explicitly assigned to them as we do here. For closed-set scenarios, previous work [Chung et al., 2003, Kao et al., 2004, Chen et al., 2005] have shown no significant difference between those two methods when considering the one-vs-one approach. Notice both methods can also be employed with inherently multiclass classifiers as well, as is the case of the *reject option* of Fukunaga [1990] (external grid search) and the rejection threshold per training class of Muzzolini et al. [1998] (internal grid search).[1]

Now, we define all four possible configurations among external, internal, closed-, and open-set grid search, thus open-set grid search is defined for every case and we evince the difference compared to closed-set forms.

The *external closed-set grid search* is performed as usual: it introduces samples of all n known classes into validation set and searches for best parameters based only on the *empirical risk*. *External open-set grid search*, differently, ensures that a subset of the n known classes appears only in validation set so that samples from those classes are

[1] Reject option and rejection threshold per training class are better introduced in Chapter 3.

	External		Internal	
	Fitting	**Validation**	**Fitting**	**Validation**
Closed	n	n	n	n
Open	$\lceil n/2 \rceil$	n	$\lceil (n-1)/2 \rceil + 1$	n

Table 2.1: Comparison of open- and closed-set grid search strategies in terms of number of known classes employed in *fitting* and *validation* sets during grid search. Value n refers to the number of known classes available for training, i.e., present in training set.

unknown for the model used for evaluating parameters.

We define the internal grid search considering the one-vs-all approach. For the *internal* form, the *closed-set* variation searches for parameters of a binary model by having the *same* representative classes on fitting set—the set used for generating a model for grid search—and validation set: one of the classes is labeled as positive and all other available classes are labeled negative; however, the set of distinct classes composed by the negative class is the same on both fitting and validation sets. For the *internal open-set grid search*, it ensures the negative class of the validation set comprises extra classes compared to the set of classes composed by the fitting set. This way, some "unknown instances" appear along with the negative set for validating best parameters for the final model.

In Table 2.1, we summarize those alternatives. As shown in that table, internal open-set grid search employs $\lceil (n-1)/2 \rceil + 1$ known classes on fitting set because half of the $n-1$ classes included in negative set—in the level of a binary classifier—are selected to appear only in validation set. The other half remains on fitting set along with the additional positive class.

Those open-set approaches can be employed along with any classifier with three or more classes available for training. In this work, we have opted at employing grid search instead of other alternatives for hyperparameter optimization—e.g., random search [Solis and Wets, 1981]—to ensure a paired comparison among the recognition methods. As the open-set approach only differs from the closed-set one on the split of the training data, notice that this technique can be trivially extended to be employed along with random search as well as other hyperparameter optimization methods [Li et al., 2018]. In Chapter 6, we compare closed- with open-set grid search applied to multiple classifiers.

2.3 Evaluation measures

In previous work [Mendes Júnior, 2014], we have proposed some evaluation measures specific for assessing performance of experiments in open-set scenarios, due to the lack of appropriate measures at the time. Those measures are necessary because in some problems the proportion of known/unknown instances for testing can be very unbalanced and traditional closed-set measures could misinterpret results. As for reference, those measures are extension of the well-known macro- and micro-averaging f-measure—we call them open-set macro- and micro-averaging f-measure—and the named Normalized Accuracy (NA), that balances the Accuracy on Known Samples (AKS) and the Accuracy

on Unknown Samples (AUS), calculated separately. Since publication of those definitions, we have obtained a small improvement on the definition of open-set measures. We define them here for later use on the experiments.

As their own name indicate, AKS and AUS measures are calculated separately on the subsets of testing samples containing only the known and unknown samples, respectively. The NA of previous work balances AKS and AUS with a 50–50% weight. In certain scenarios, however, one would consider that a better accuracy in one of the sets is preferable over the other. Then, in this case, we consider the definition of NA as in Equation (2.1).

$$\text{NA} = \lambda_r \text{AKS} + (1 - \lambda_r)\text{AUS}, \tag{2.1}$$

in which λ_r, $0 < \lambda_r < 1$, is a regularization constant. This more general definition can also be employed for grid searching, while assessing accuracy on validation set. For instance, if one prefers the recognition method to be less tolerant to false acceptance of unknown samples, $\lambda_r < 0.5$ can be set for grid searching better parameters for the model. Conversely, if one wants parameters that ensure both accuracy on AKS and AUS to be reasonably well, the Harmonic Normalized Accuracy (HNA), as defined in Equation (2.2), should be employed.

$$\text{HNA} = \begin{cases} 0, \text{ if AKS} = 0 \text{ or AUS} = 0, \\ \dfrac{2}{\dfrac{1}{\text{AKS}} + \dfrac{1}{\text{AUS}}}, \text{ otherwise.} \end{cases} \tag{2.2}$$

Notice that HNA goes to 0 as either AKS or AUS goes to 0. NA, however, can stay around 0.5, in case of a no-classifier, which is not desirable in some cases, then justifying the use of HNA. In Chapter 6, we show how λ_r of NA can be calibrated for evaluation on validation set during grid search, targeted on defining a more/less restrictive behavior on classifying unknown samples.

Chapter 3

Related work

The open-set problem in inherently present in many real-world recognition problem, however, only in a recent work of Scheirer et al. [2013] it has been properly formalized with a math-grounded basis. The term, however, have been employed back in the works of Gong [2002], Deng and Hu [2003], Sivakumaran et al. [2003], Li and Wechsler [2005], Han et al. [2010], Gao et al. [2011], Güney et al. [2012], Heflin et al. [2012], Pritsos and Stamatatos [2013], and Zhao et al. [2013], predominantly on biometric recognition.

In recent years, we have we have watched an increasing attention on the open-set setup along with multiple other applications in machine learning and pattern recognition. Besides the well-known problems in biometric recognition [Kumar and Kumar, 2014, Zhang and Hao, 2014, dos Santos Junior and Schwartz, 2014, Rattani et al., 2015, Wang et al., 2016, Günther et al., 2017, Moeini et al., 2017, Vareto et al., 2017, Xie et al., 2018], which still seems to receive the greatest attention, exist works in domain analysis [Busto and Gall, 2017, Dong et al., 2019], intrusion detection [Cruz et al., 2017], camera and camera model identification [Costa et al., 2012, 2014, Bayar and Stamm, 2018], acoustic scene classification [Battaglino et al., 2016], language identification/recognition [Zhang and Hansen, 2014, 2016], web genre detection [Pritsos and Stamatatos, 2018], among others, as well as works targeted at more general-purpose solutions in multiple steps of the recognition process [Scherreik and Rigling, 2016, Zhang and Patel, 2017, Liang et al., 2018, Xiao et al., 2018, Rudd et al., 2018, Tian et al., 2018, Neira et al., 2018]. In this work, we focus on the methods tailored to general-purpose open-set recognition [Heflin et al., 2012, Pritsos and Stamatatos, 2013, Costa et al., 2014, Scheirer et al., 2013, 2014, Jain et al., 2014] as well as the ones that allow direct extension for the open-set setup [Schölkopf et al., 2001, Tax and Duin, 2004, Chang and Lin, 2011].

In this chapter, we present existing methods that somehow deal with open-set classification scenarios. We separated those approaches into two categories: in Section 3.1, we present approaches employed in problems related to open-set recognition and, in Section 3.2, we present previous work directly addressing the open-set recognition problem by means general-purpose classification methods. As we shall see, virtually all previous solutions for the open-set scenarios were based on SVM classifiers.

3.1 Approaches for similar problems

One-class classifiers—such as the OCSVM and SVDD—at first glance, seem promising for the open-set scenario, as they focus on the known class and ignore everything else. For the multiclass and open-set scenario, one-class classifiers can be applied by training a one-class classifier for each of the known classes. As those methods have the same behavior of binary classifiers, they can be directly extended for multiclass classification by employing one-vs-all multiclass-from-binary approaches.

In the case of the One-Class Support Vector Machines (OCSVM) [Schölkopf et al., 2001], for example, it finds the best margin with respect to the origin. Kernels can be applied, creating a bounded positive region around the samples of the known classes [Boser et al., 1992]. This is the most reliable approach in cases in which the access to a second class is very difficult or even impossible. It is usually employed in problems for which leaving half-spaces is undesirable [Chen et al., 2001]. OCSVM, however, has a limited use because it does not provide good generalization nor specialization. Several works dealing with OCSVM have tried to overcome the problem of lack of generalization/specialization, e.g., by introducing some few extra negative/outlier instances to better refine decision boundary [Jin et al., 2004, Tax and Duin, 1999b, Wu and Ye, 2009, Manevitz and Yousef, 2001] or by employing cascade approaches along with binary models [Cevikalp and Triggs, 2012]. All of these works can be applied to the multiclass and open-set scenario in the same way the OCSVM can be applied.

Although one-class classifiers are inherently suitable for open-set classification problems, binary classifiers (e.g., SVM) also hold potential. For example, binary classifiers can be applied to the open-set scenario (which is multiclass) using the one-vs-all [Rocha and Goldenstein, 2014] approach. The binary classifier which classifies as positive is chosen to decide the final class of the multiclass classifier. When two or more binary classifiers return positive for some test instance, the one most confident about its classification is chosen to decide the final class. When no binary classifier classifies as positive, then the test sample is classified as unknown. In this vein, all variations of the SVM [Bartlett and Wegkamp, 2008, Malisiewicz et al., 2011, Jayadeva et al., 2007, Chew et al., 2012] (which are also binary classifiers) can be applied using the one-vs-all approach.

As we mentioned before, the trivial approach to handle the open-set scenario is to define a threshold on the similarity score of the classifiers: for SVM, this threshold could be defined based on the distance from the hyperplane or the probability value; for the Nearest Neighbor (NN) classifier, it could be defined based on the distance to the nearest neighbor, for example. Establishing a threshold on the similarity score means *rejecting distant* samples from the training samples in some cases. Also, one would be interested in rejecting doubtful or ambiguous samples.

The *reject option* presented by Fukunaga [1990] is a form of postponing the decision-making process to further evaluate the test sample by other means (e.g., other classifiers). Note that in the open-set scenario, we want to classify a test sample as one of the known classes or as none of the known classes (unknown) without postponing the decision making. Chow [1970] presented a method for rejecting doubtful test samples, i.e., to avoid classifying the test sample as one of the known classes when the classifier has good similar

scores for more than one class. Later, Dubuisson and Masson [1993] extended the *ambiguity reject option* of Chow [1970] and presented the *distance reject option* in the context of statistical pattern recognition. The distance reject option is to avoid classifying the test sample "far from" the training ones in the feature space. Muzzolini et al. [1998] extended the work of Dubuisson and Masson [1993] to define better distance rejection thresholds adapted for each training class.

Works dealing with distance rejection can be applied to the open-set classification scenario because if one ensures that far away test samples are rejected (i.e., classified as unknown), then the classifier creates a bounded open space in the feature space. The problem for most of the methods dealing with rejection by thresholding the similarity score is the difficulty to define such threshold.

3.2 Approaches proposed for open-set scenarios

In this section, we review only recent work that explicitly deals with open-set scenarios. Anyhow, we note that other insights presented in many works in the literature can be somehow modified or directly used for the open-set scenario. Most of these works, however, did not perform experiments with appropriate open-set setup.

In the works of Heflin et al. [2012] and Pritsos and Stamatatos [2013], they present a multiclass SVM classifier based on OCSVM. For each of the training classes, they fit an OCSVM. In the prediction phase, the test sample is classified by all n OCSVMs, in which n is the number of available classes for training. The test sample is classified to the class in which its OCSVM classified as positive. When no OCSVM classifies as positive, the test sample is classified as unknown. Heflin et al. [2012] deal with multiple class classification, then when two or more OCSVMs classify as positive, the test sample is classified as belonging to those positive classes. Differently, Pritsos and Stamatatos [2013] choose the more confident classifier among the ones that classify as positive. In those works, the OCSVM is used with the Radial Basis Function (RBF) kernel.

The Decision Boundary Carving (DBC) [Costa et al., 2012, 2014] is an extension upon the SVM aiming at a more restrictive specialization on the positive class of the binary classifier. For this, the method moves the hyperplane a value ϵ towards the positive class (in rare cases backwards). The value ϵ is obtained by minimizing the *training data error*. For multiclass classification, the one-vs-all approach can be used.[1] The DBC was tested by the authors along with RBF kernel. The test sample is classified as unknown when no binary classifier classifies as positive and the test sample is classified as the most confident class when one or more classifiers classify as positive. The confidence is obtained based on the distance of the test sample from the hyperplane: the more distant, the more confident.

The 1-vs-Set Machine (OVS) [Scheirer et al., 2013] is a binary classifier extended upon the SVM. Similarly to the DBC, it moves the main hyperplane towards the positive class. Besides, a second hyperplane, parallel to the main one, is created such that the positive

[1] Despite dealing with a multiclass problem, Costa et al. [2014] evaluated their method in a binary fashion by obtaining the accuracy of individual binary classifiers. They did not present the multiclass version of the classifier directly. Therefore, in this work, we consider their method with the one-vs-all approach in the experiments.

class is between the two hyperplanes. This second hyperplane makes the samples "behind" the positive class to be classified as negative. Then a refinement step is performed on both hyperplanes. According to the authors, the method works better with the linear kernel.

The work of Scheirer et al. [2013] was the first effort on formalizing the open-set recognition problem. The concept of *open space* has been defined as the region of the feature space outside the support of the training samples, i.e., the region that potentially refers to classes unknown at training phase. In practice, the open space should be estimated by an open-set recognition function f and it is unknown a priori. Consider $f(\mathbf{x})$ a recognition function such that $f(\mathbf{x}) = 1$ indicates that $\mathbf{x}$ is known and $f(\mathbf{x}) = 0$ indicates $\mathbf{x}$ as unknown. Assuming a bounded positively-labeled open space $\mathcal{O}$, to define the open-space risk $R_{\mathcal{O}}$, Scheirer et al. have considered a large ball S_o containing both $\mathcal{O}$ and the known training samples such that

$$R_{\mathcal{O}}(f) = \frac{\int_{\mathcal{O}} f(\mathbf{x})d\mathbf{x}}{\int_{S_o} f(\mathbf{x})d\mathbf{x}}. \tag{3.1}$$

We see in Equation (3.1) that the greater the PLOS the greater the risk of the unknown.

Usual closed-set classifiers optimize the *empirical risk* R_ϵ, usually measured on the training data, e.g., by grid searching for the best parameters for the model. An open-set problem, however, as formalized by Scheirer et al., requires minimizing both R_ϵ and $R_{\mathcal{O}}$:

$$\arg\min_{f\in\mathcal{H}} \left\{\lambda_o R_\epsilon(f) + R_{\mathcal{O}}(f)\right\}, \tag{3.2}$$

in which λ_o is a regularization constant. In practice, the open-space risk $R_{\mathcal{O}}$ is difficult to obtain, as it strongly depends on unknown data not available at training phase. As a side note, the open-set grid search we have formalized in Chapter 2 is a form of estimation of $R_{\mathcal{O}}$. Also, notice that λ_o of Equation (3.2) has a direct correspondence to λ_r for the NA in Equation (2.1). Later in Chapter 6 we will show the effectiveness of both employing open-set grid search and adjusting the value of λ_r for the NA during grid search.

The authors of the Weibull-Calibrated Support Vector Machines (WSVM) [Scheirer et al., 2014] classifier define the Compact Abating Probability (CAP) model for open-set recognition, which decreases the probability of a test sample to be considered as belonging to one of the known classes when it is far away from the training samples. In the WSVM, they use two steps for classification: a CAP model based on a one-class classifier and the other one based on a binary classifier allied with the Extreme Value Theory (EVT) [Coles, 2001, de Haan and Ferreira, 2007, Scheirer, 2017]. The first step aims at obtaining the probability of a test sample to belong to a positive/known class and the second step aims at obtaining the probability of a test sample to *not* belong to a negative/unknown class. The product of both probabilities is the probability of the test sample to belong to a positive/known class. The WSVM uses the RBF kernel in the work of Scheirer et al. [2014].

Jain et al. [2014] propose the PISVM, also based on the EVT. It is an algorithm for estimating the unnormalized posterior probability of class inclusion. For each known class, a Weibull distribution [Coles, 2001] is estimated based on the smallest decision values of

the positive training samples. The binary classifier for each class is an SVM with RBF kernel trained using the one-vs-all approach, i.e., the samples of all remaining classes are considered as negative samples. They introduce the idea of *cross-class validation* which is similar to the open-set grid search we formally define in our work. For a test sample, PISVM chooses the class for which the decision value produces the maximum probability of inclusion. If that maximum is below a given threshold, the input is marked as unknown.

Bendale and Boult [2015] have considered initial steps towards what they have named open-world recognition, which consists on the open-set problem along with incremental learning [Ross et al., 2008] not only on the level of instances but also on the level of classes to be included online in the system. The concepts of *known unknown* and *unknown unknown* classes are important considerations from their work. In a testing scenario—or a real scenario—samples that appear for classification that belong to none of the classes used for training a classifier are, in essence, unknown unknown. For certain applications, however, one can be interested in recognizing and classifying a limited number of classes of interest, while extra classes, that might be available for training the recognition method as well, can be employed as known unknown classes used to guide the classifier at recognizing the unknown unknown classes. In Chapter 7, we evaluate the employment of known unknown classes in the context of neural networks.

Finally, in a previous work [Mendes Júnior, 2014, Neira, Mendes Júnior, Rocha, and Torres, 2018], we have extended the graph-based Optimum-Path Forest (OPF) [Papa et al., 2007, 2012] classifier for open-set recognition by introducing the OSOPF, a distance-based method that shares the same principle of OSNN—thresholding the ratio of distances instead of raw distances, as we shall see in Chapter 4—however the latter consists on a simplification/generalization of the first.

Chapter 4

Distance-based classifiers

As discussed previously, distance-based open-set classifiers can be easily extended to open-set recognition by simply applying a threshold on the distance or the similarity value generated by the classifier. This threshold, however, is difficult to obtain and the behavior of such classifier is not reliable when applied to the raw value, due to the curse of dimensionality. Furthermore, sparseness of each of the known training classes can be different and, consequently, by choosing a single threshold will make the method unable to generalize well. Anyhow, as we shall show in Chapter 6, even by defining a threshold per known class, this method does not generalize well.

Aiming at overcoming the problem of thresholding the raw distance of distance-based methods, in previous work [Mendes Júnior, 2014], we have developed a method that learns an optimal threshold on ratio of similarity scores based on the OPF, an inherently closed-set classification method. As this ratio is ensured to be always in the interval 0–1, the threshold that trades off the empirical risk and the risk of the unknown has delimited range for search regardless the dimensionality of the feature space. In this work, we have extended this previous work to work with an even simpler classifier: the Nearest Neighbor (NN) classifier.

As for the remaining of this chapter, in Section 4.1, we present the base foundation of NN and then we introduce the Open-Set Nearest Neighbors (OSNN) in Section 4.2. In Section 4.3, we present some additional considerations regarding distance-based methods for open-set recognition.

4.1 Nearest Neighbor classifier

In this section, we first describe the more general k-Nearest Neighbors (kNN) classifier, then we present the NN classifier we use as the base classifier for OSNN. We present the kNN as described by Bishop [2006], firstly as a technique for density estimation, then turn it to the kNN classifier.

Differently to the kernel approach to density estimation, the kNN technique does not need to have a fixed parameter for the kernel width. Instead, given a fixed value k of data points to be used to infer the density estimation, with kNN technique, we obtain a volume V of the minimal sphere around data point $\mathbf{x} \in \mathbb{R}^d$ such that k data points are

inside, in which $\mathbf{x}$ is a point for which we want the density estimate:

$$p(\mathbf{x}) = \frac{k}{mV}. \tag{4.1}$$

In this case, m is the total number of data points available.[1]

To extend this density estimation for classification, we apply the kNN density estimation technique for each class ℓ_i. Consider that each class ℓ_i has m_i representative samples. To classify a sample $\mathbf{x}$, we draw a minimal sphere with volume V centered in $\mathbf{x}$ so that it contains k sample regardless the class. Consider that inside this sphere there are k_i samples of the class ℓ_i. Using Equation (4.1), we have a density estimate for each class:

$$p(\mathbf{x}|\ell_i) = \frac{k_i}{m_i V} \tag{4.2}$$

and the class priors:

$$p(\ell_i) = \frac{m_i}{m} \tag{4.3}$$

Combining the unconditional density of Equation (4.1) with Equations (4.2) and (4.3) using Bayes' theorem, we have the probability of class membership:

$$p(\ell_i|\mathbf{x}) = \frac{p(\mathbf{x}|\ell_i)p(\ell_i)}{p(\mathbf{x})} = \frac{k_i}{k}, \tag{4.4}$$

i.e., the kNN classifier assigns the sample $\mathbf{x}$ using a majority voting scheme based on the classes of the k nearest training samples of $\mathbf{x}$. The value of k defines the degree of smoothing of the classifier.

The kNN classifier does not need a fitting phase, as the training data are simply stored to be used in the prediction phase. OSNN is based on a particular case of the kNN, called NN classifier, which is equivalent to kNN for $k = 1$.

4.2 Open-Set Nearest Neighbors classifier

As for NN, OSNN is also inherently multiclass and has a simplified training step. In training phase, OSNN is simply required to store training samples and search for a decision threshold T, $0 < T < 1$, that in prediction phase is applied to a calculated ratio of distances. The rationale behind OSNN, firstly, is to bound the KLOS, and to better define the decision boundary on the region of the feature space getting far apart from training samples. That is why thresholding on ratio of distance is employed instead of thresholding the raw distance itself. The definition of OSNN is straightforward and defined as follows.

The training phase of OSNN simply requires the storage of training samples, as for NN, and the choice of T for proper rejection of instances from unknown classes. We call

[1] As presented by Bishop [2006], the validity of Equation (4.1) depends on two contradictory assumptions: (1) the region in the volume V be sufficiently small, making the density approximately constant in the region; and (2) sufficiently large so that the number k of points inside the region is enough for a binomial distribution to be sharply peaked.

the phase of choosing T as *parameter optimization* and, in essence, it relies on the open-set grid search as presented in Chapter 2.

For the prediction phase, first consider that the traditional NN obtains only the nearest neighbor training sample $\mathbf{x}_1$ and predicts that a test instance $\mathbf{x}$ belongs to class $\theta(\mathbf{x}_1)$, in which $\theta(\mathbf{x}') \in \{\ell_1, \ldots, \ell_n\}$ represents the ground-truth class of a training instance $\mathbf{x}'$ in a classification problem with n classes. OSNN also obtains $\mathbf{x}_1$ in the same way and, furthermore, obtains an additional nearest neighbor $\mathbf{x}_2$ such that $\theta(\mathbf{x}_1) \neq \theta(\mathbf{x}_2)$. Then, the ratio R is defined as

$$R = \frac{d(\mathbf{x}, \mathbf{x}_1)}{d(\mathbf{x}, \mathbf{x}_2)}, \tag{4.5}$$

in which d is the distance in the feature space. Finally, OSNN's decision function employs the threshold T previously obtained in the training phase:

$$f(\mathbf{x}) = \begin{cases} \theta(\mathbf{x}_1) & \text{if } R \leq T \\ \ell_0 & \text{if } R > T, \end{cases}$$

in which ℓ_0 indicates the test sample is classified as unknown.

OSNN is able to bound the KLOS because R approaches 1 as a test sample $\mathbf{x}$ get far away from training samples.

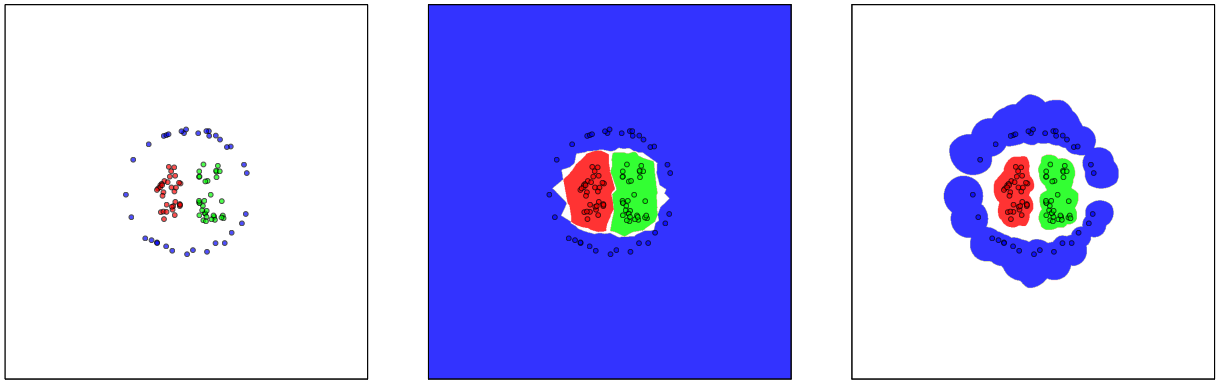
4.3 Additional considerations

The effectiveness of OSNN resides on being able to bound the KLOS. This property, however, comes with a price: R also approaches 1 for test samples in the decision frontier of two or more training classes. Anyhow, the ability of rejecting an instance in the open space compensates this undesirable behavior. Aiming at demonstrating this, we define an additional classifier based on NN that also rejects doubtful testing samples but with no ability for bounding the KLOS. We call it Open-Set Nearest Neighbors Class Verification (OSNN$^{\text{CV}}$) as its rationale is to obtain a second nearest neighbor to "verify" if the class of the main nearest neighbor should be used for classification.

OSNN$^{\text{CV}}$ does not require training, as it has no parameter learning (as for NN). In prediction phase, as for NN and OSNN, OSNN$^{\text{CV}}$ obtains the nearest neighbor $\mathbf{x}_1$. Then, a second nearest neighbor is obtained with just the constraint $\mathbf{x}_1 \neq \mathbf{x}_2$, so it can happen that $\theta(\mathbf{x}_1) = \theta(\mathbf{x}_2)$. The decision function of OSNN$^{\text{CV}}$ is defined as follows.

$$f(\mathbf{x}) = \begin{cases} \theta(\mathbf{x}_1) & \text{if } \theta(\mathbf{x}_1) = \theta(\mathbf{x}_2) \\ \ell_0 & \text{if } \theta(\mathbf{x}_1) \neq \theta(\mathbf{x}_2). \end{cases}$$

The rationale behind this is that the classifier is only sure about attributing an example to the class of the nearest neighbor if there are more than a single nearest neighbor of the same class. However, on the open space, the two nearest neighbors can continue to be on the same class *ad infinitum* and that is the reason OSNN$^{\text{CV}}$ cannot bound the KLOS. Notice that OSNN$^{\text{CV}}$ also has the uninteresting behavior of rejecting test instances among known classes, which can be problematic in overlapping regions of two or more training



Boat dataset. (a) OSNN^{CV}. (b) OSNN.

Figure 4.1: Behavior analysis of OSNN. The Boat dataset is depicted on the far left. Figure (a) depicts the behavior of OSNN^{CV}, which is not able to bound the KLOS. Figure (b) depicts the bounded KLOS left by OSNN. The small circles represent the training samples from the dataset and their colors represent their classes. A colored background represents the class in which a possible test instance in the corresponding position of the feature space would be classified. White background indicates that a possible test instance in that region would be classified as unknown.

classes.

For a proof of concept, we compare the behavior of OSNN and OSNN^{CV} on the synthetic Boat [Kuncheva and Hadjitodorov, 2004] dataset in Figure 4.1. The Boat dataset is 2-dimensional and comprises 3 classes. All samples from this dataset were employed for training the classifiers for generating Figures 4.1a,b. We observe the nicely bounded KLOS left by OSNN in Figure 4.1b while, in Figure 4.1a, we observe an unbounded KLOS for OSNN^{CV}. In Chapter 6, we compare those two classifiers to evince the importance of bounding the KLOS.

Chapter 5

Geometric classifiers

In this chapter, we analyze geometric classifiers—e.g., SVM, OCSVM—for open-set recognition. As explained in Chapter 2, we employ multiclass-from-binary one-vs-all method in all those considerations, except when otherwise stated. The main purpose here is to ensure a binary classifier to be able to bound the PLOS so that the risk of the unknown is finite. In Section 5.1, we formalize the SVM, a base foundation for the SSVM presented in Section 5.2. And, in Section 5.3, we present additional considerations regarding geometric classifiers for open-set recognition.

5.1 Support Vector Machines formalization

SVM is a binary classifier that, given a set X of training samples $\mathbf{x}_i \in \mathbb{R}^d$ and the corresponding labels $y_i \in \{-1, 1\}$, $i = 1, \ldots, m$, it finds a maximum-margin hyperplane that separates $\mathbf{x}_i$ for which $y_i = -1$ from $\mathbf{x}_j$ for which $y_j = 1$ [Cortes and Vapnik, 1995]. We consider the soft margin case with parameter C.

The primal optimization problem is usually defined as

$$\min_{\mathbf{w},b,\xi} \frac{1}{2}||\mathbf{w}||^2 + C\sum_{i=1}^{m} \xi_i,$$

$$\text{s.t. } y_i(\mathbf{w}^T\mathbf{x}_i + b) \geq 1 - \xi_i, \ \forall i, \tag{5.1}$$

$$\xi_i \geq 0, \ \forall i. \tag{5.2}$$

To solve this optimization problem, we use the Lagrangian method to create the dual optimization problem. In this case, the final Lagrangian is defined as

$$\mathcal{L}(\mathbf{w}, b, \xi, \alpha, r) = \sum_{i=1}^{m} \alpha_i - \frac{1}{2}||\mathbf{w}||^2, \tag{5.3}$$

in which $\alpha_i \in \mathbb{R}$, $r_i \in \mathbb{R}$, $i = 1, \ldots, m$, are the Lagrangian multipliers. Then, the

optimization problem now is defined as

$$\min_{\alpha} W(\alpha) = -\mathcal{L}(\mathbf{w}, b, \xi, \alpha, r) = \frac{1}{2}||\mathbf{w}||^2 - \sum_{i=1}^{m} \alpha_i, \tag{5.4}$$

$$\text{s.t. } 0 \leq \alpha_i \leq C, \ \forall i, \tag{5.5}$$

$$\sum_{i=1}^{m} \alpha_i y_i = 0. \tag{5.6}$$

The decision function of a test sample $\mathbf{x}$ comes from the constraint in Equation (5.1) and is defined as

$$f(\mathbf{x}) = \text{sign}(\mathbf{w}^T\mathbf{x} + b) = \text{sign}\left(\sum_{i=1}^{m} y_i \alpha_i \mathbf{x}_i^T \mathbf{x} + b\right).$$

Boser et al. [1992] proposed a modification in SVM for the cases in which the training data are not linearly separated in the feature space. Instead of linearly separating the samples in the original space $\mathcal{X}$ of the training samples in X, the samples are projected onto a higher dimensional space $\mathcal{Z}$ in which they are linearly separated. This projection is accomplished using the kernel trick [Mercer, 1909]. One advantage of this method is that in addition to separating non-linear data, the optimization problem of the SVM remains almost the same: instead of calculating the inner product $\mathbf{x}^T\mathbf{x}'$, it uses a kernel $K(\mathbf{x}, \mathbf{x}')$ that is equivalent to the inner product $\phi(\mathbf{x})^T\phi(\mathbf{x}')$ in a higher dimensional space $\mathcal{Z}$, in which $\phi : \mathcal{X} \mapsto \mathcal{Z}$ is a projection function. When using the kernel trick, we do not need to know the $\mathcal{Z}$ space explicitly.

Using kernels, the decision function of a test sample $\mathbf{x}$ becomes

$$f(\mathbf{x}) = \text{sign}\left(\sum_{i=1}^{m} y_i \alpha_i K(\mathbf{x}_i, \mathbf{x}) + b\right). \tag{5.7}$$

The most used kernel for SVM is the RBF kernel [Schölkopf and Smola, 2001], defined as follows.

$$K(\mathbf{x}, \mathbf{x}') = e^{-\gamma||\mathbf{x}-\mathbf{x}'||^2}. \tag{5.8}$$

It is proved that using this kernel, the projection space $\mathcal{Z}$ is an ∞-dimensional space [Schölkopf and Smola, 2001].

5.2 Specialized Support Vector Machines classifier

In this section, we show how it is possible to ensure a bounded PLOS for SVM no matter the shape of the samples in the feature space so that a limited risk of the unknown can be ensured for open-set recognition. Then, we present the SSVM, that implements an alternative optimization problem compared to SVM, aiming at ensuring a bounded PLOS for every binary classifier. As usual, SSVM can be extended to multiclass classification based on a one-vs-all approach.

5.2.1 Ensuring a bounded positively-labeled open space for Support Vector Machines

By simply employing an RBF kernel, we cannot ensure the PLOS is bounded.

Theorem 1. *Support Vector Machines (SVM) with any Radial Basis Function (RBF) kernel has a bounded positively-labeled open space (PLOS) if and only if the bias term b is negative.*[1]

Proof. We know that

$$\lim_{d\to\infty} K(\mathbf{x}, \mathbf{x}') = 0, \tag{5.9}$$

in which $K(\mathbf{x}, \mathbf{x}')$ is any RBF kernel and $d = \|\mathbf{x} - \mathbf{x}'\|$. For the cases in which a test sample $\mathbf{x}$ is far away from every support vector $\mathbf{x}_i$, we have that

$$\sum_{i=1}^{m} y_i \alpha_i K(\mathbf{x}_i, \mathbf{x})$$

also tends to 0. From Equation (5.7) it follows that

$$f(\mathbf{x}) \to \text{sign}\,(b)$$

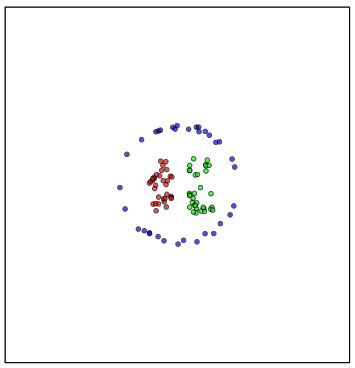
when $\mathbf{x}$ is far away from the support vectors. Therefore, for negative values of b, $f(\mathbf{x})$ is always negative for far away $\mathbf{x}$ samples. That is, samples in an bounded region of the feature space will be classified as positive. For the only if direction, let b be positive. Then there will exist a distance d such that $\forall i : \|\mathbf{x}_i - \mathbf{x}\| > d \implies f(\mathbf{x}) = \text{sign}(b) > 0$, i.e., positively classified samples will be in an unbounded region of the feature space. □

Theorem 1 can be applied not only to the RBF kernel of Equation (5.8) but to any radial basis function [Buhmann, 2003] kernel satisfying Equation (5.9), e.g., Generalized T-Student (TST) kernel, Rational Quadratic (RQ) kernel, and Inverse Multiquadric (IMQ) kernel [Souza, 2010].

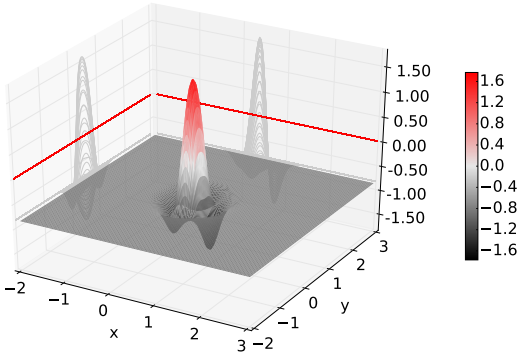
Figure 5.1 depicts the rationale behind Theorem 1 on a 2-dimensional synthetic dataset. The z axis represents the decision values for which possible 2-dimensional test samples (x, y) would have for different regions of the feature space. Training samples are normalized between 0 and 1. Note in the subfigures that for possible test samples far away from the training ones, e.g., $(2, 2)$, the decision value approaches the bias term b. Note in Figure 5.1c that an unbounded region of the feature space would have samples classified as positive. Consequently, all those samples would be classified as class 3 by the final multiclass-from-binary classifier. In general SVM usage, both positive and negatives biases occur as b depends on the training data.

Theorem 1 also provides a solution to the problem of unbounded PLOS. We can ensure a bounded PLOS by simply employing an RBF kernel and ensuring a negative b.

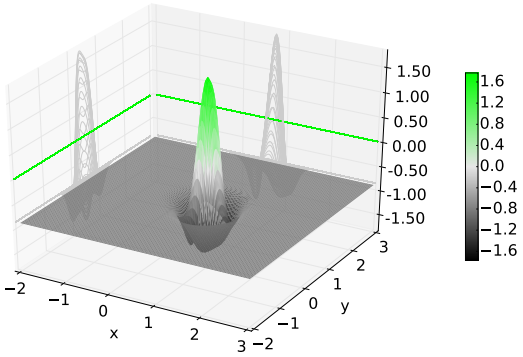
[1] In some implementations, including the LIBSVM library [Chang and Lin, 2011], the decision function is defined as $f(\mathbf{x}) = \text{sign}(\mathbf{w}^T\mathbf{x} - \rho)$. In that case, instead of ensuring a negative bias term b, one must ensure a positive bias term ρ to bound the PLOS.



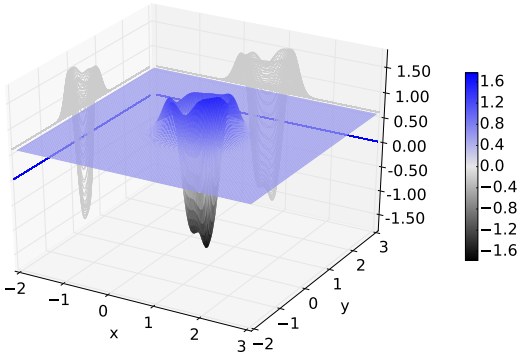
Boat dataset with 3 classes: red (the central class to the left), green (the central class to the right), and blue (the class with the ring shape).



(a) Class 1 (red). $b = -0.832$.



(b) Class 2 (green). $b = -0.86$.



(c) Class 3 (blue). $b = +0.594$.

Figure 5.1: Behavior analysis of SVM with a RBF kernel. Image on the top-left depicts the Boat dataset. Figures (a)–(c) correspond to the red, green and blue classes of the Boat dataset, respectively. The x, y axes in Figures (a)–(c) represent the two features of the Boat dataset, used for training. The training data is normalized between 0 and 1 for each feature. The z axis shows the value of the SVM decision function $\sum_{i=1}^{m} y_i \alpha_i K(\mathbf{x}_i, \mathbf{x}) + b$ (Equation 5.7 without sign function) and the colored lines in the walls depict the point 0, that separates the positive class from the negative one (equivalent to the sign function of Equation 5.7). Note in Figure (c) that an unbounded region of the feature space remains in the positive side, as $b > 0$ and $f(x, y) \approx b$ for $(x,\ y)$ points far away from support vectors.

In Section 5.2.2, we present a new SVM optimization objective that optimizes the margin while ensuring the bias term b is negative.

A corollary from Theorem 1 is that either the positively-labeled or the negatively-labeled regions of the feature space is bounded while either of them is unbounded as well, when SVM is employed with an RBF kernel.

5.2.2 Specialized Support Vector Machines optimization problem

As we discussed in Section 5.2.1, we must ensure a negative b to obtain a bounded PLOS. For this, we define the SSVM optimization problem as

$$\min_{\mathbf{w},b,\xi} \frac{1}{2}||\mathbf{w}||^2 + C\sum_{i=1}^{m}\xi_i + \lambda b, \tag{5.10}$$

subject to the same constraints defined in Equations (5.1) and (5.2), in which λ is a regularization parameter that trades off between the empirical risk and the risk of the unknown.

From Equation (5.10), the dual formulation has the same Lagrangian defined in Equation (5.3). Consequently, we have to optimize the same function as defined in Equation (5.4) with the constraint in Equation (5.5). However, the constraint in Equation (5.6) is replaced by the constraint

$$\sum_{i=1}^{m}\alpha_i y_i = \lambda. \tag{5.11}$$

The same Sequential Minimal Optimization (SMO) algorithm proposed by Platt [1998], with the Working Set Selection (WSS) proposed by Fan et al. [2005], for optimizing ensuring the constraint in Equation (5.6) can be applied to this optimization containing the constraint of the Equation (5.11). As the main idea of the SMO algorithm is to ensure that $\sum \alpha_i y_i$ remains the same from one iteration to the other, before the optimization starts, we initialize α_i such that $\sum \alpha_i y_i = \lambda$. For this, we let $\alpha_i = \lambda/m_p$, $\forall i$ such that $y_i = 1$, in which m_p is the number of positive training samples.

Proposition 1. *For the Support Vector Machines (SVM) with soft margin, the maximum valid value for λ is Cm_p.*

Proof. From Equation (5.5), $0 \leq \alpha_i \leq C$. The maximum value $\lambda = \sum \alpha_i y_i$ is thus obtained by setting $\alpha_i = C$ for i such that $y_i = 1$ and setting $\alpha_i = 0$ for i such that $y_i = -1$. This yields $\lambda \leq Cm_p$ □

During optimization, we must ensure $\lambda \leq Cm_p$ given that if $\lambda > Cm_p$, the constraint in Equation (5.5) would be broken for some α_i.

Despite Proposition 1 saying that it is allowed $\lambda = Cm_p$, when it happens, we have that $\alpha_i = C$ for $y_i = 1$ and $\alpha_i = 0$ for $y_i = -1$, and there will be no optimization. In this case, despite satisfying the constraints, there is no flexibility for changing values of α_i because, for each pair α_i, α_j selected by the WSS algorithm, we must update $\alpha_i = \alpha_i + \nabla_\alpha$, $\alpha_j = \alpha_j + \nabla_\alpha$ when $y_i \neq y_j$ and $\alpha_i = \alpha_i - \nabla_\alpha$, $\alpha_j = \alpha_j + \nabla_\alpha$ when $y_i = y_j$. For any

$\nabla_\alpha \neq 0$, the constraint $0 < \alpha_i < C$ would break for either α_i or α_j, for any selected pair. Then, in practice, we grid search λ in the interval $0 \leq \lambda < Cm_p$.

Proposition 2. *There exists some λ such that we can obtain a bias term $b < 0$ for the Specialized Support Vector Machines (SSVM) with a Radial Basis Function (RBF) kernel K such that $0 < K(\mathbf{x}, \mathbf{x}') \leq 1$ when $C \geq 1$.*

Proof. See Appendix B. □

In Proposition 2, we considered a very extreme case for the proof. For example, in Case (1)—for i such that $y_i = 1$—we considered $K(\mathbf{x}_i, \mathbf{x}_j) = 1$ for j such that $y_j = -1$ and $K(\mathbf{x}_i, \mathbf{x}_j) \approx 0$ for j such that $y_j = 1$. It means that all negative samples have the same feature vector of sample $\mathbf{x}_i$ under consideration and all positive samples are far away from sample $\mathbf{x}_i$. In practice, we do not have the λ nearly as constrained as in the proof to ensure a negative bias term. Moreover, in our experiments with the SVM, we observed that oftentimes the bias term is negative for a binary classifier trained with the one-vs-all approach, i.e., it is often the case that even with $\lambda = 0$ the bias will be negative. More details about this behavior is shown in Section 6.

Notice that the proof of Proposition 2 is restricted to RBF kernels such that $0 < K(\mathbf{x}, \mathbf{x}') \leq 1$. That is the case for Gaussian kernel of Equation (5.8) as well as Generalized T-Student (TST) and Rational Quadratic (RQ) kernels. As in practice the recognition scenario is not as constrained as in the proof, we believe the statement of Proposition 2 holds true even when the RBF does not satisfy that property, e.g., for Inverse Multiquadric (IMQ) kernel.

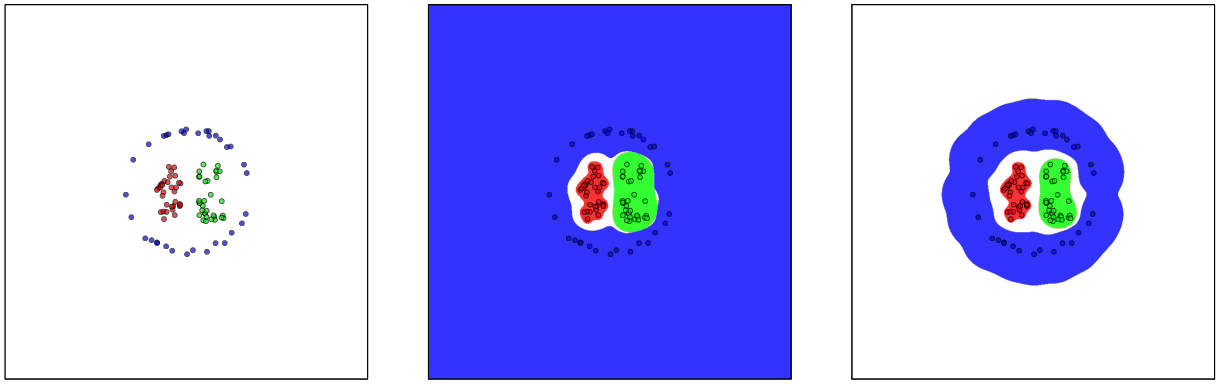
In Appendix A, we present the complete formulation of the optimization problem for the SSVM classifier.[2]

Choosing the λ parameter for the SSVM

Proposition 2 states that we can find a λ parameter that ensures a bounded PLOS for the optimization problem presented above. To ensure this, models with a non-negative bias term receive accuracy of $-\infty$ on the validation set, during the grid search. Nevertheless, we cannot ignore that, in special circumstances, certain λ values allow a negative bias term during the grid search but not for training in the whole set of training samples. In this case, once the parameters are obtained by grid search, if the obtained λ does not ensure a negative bias term for the whole training set, one would need to retrain the classifier with an increased value for λ, until a negative bias term is obtained for the final model. However, for grid search, we assume the distribution of the validation set, a subset of the training set, represents the distribution of the training set; that is one possible explanation as for why in our experiments we did not need to retrain the classifier with a value of λ larger than the one obtained during grid search, as all values of λ obtained during grid search were able to ensure a negative bias term for all binary classifiers.

As for intuitive and empirical evidence of SSVM behavior, Figure 5.2 depicts the behavior of SSVM compared to SVM. As we can see in Figure 5.2b, SSVM gracefully bounds the KLOS around training samples.

[2] SSVM source-code is available at https://github.com/pedrormjunior/ssvm.



Boat dataset. (a) SVM. (b) SSVM.

Figure 5.2: Behavior analysis of SSVM. The Boat dataset is depicted on the far left. Figure (a) depicts the behavior of the SVM with a one-vs-all approach as is. As previously evinced in Figure 5.1c, the model for the ring-shaped blue class obtains a positive bias term b and, consequently, SVM leaves an unbounded PLOS for that class. Figure (b) depicts the behavior of SSVM being able to bound the PLOS for every binary classifier and, consequently, the KLOS. All figures were generated with closed-set grid search. The small circles represent the training samples from the dataset and their colors represent their classes. A colored background represents the class in which a possible test instance in the corresponding position of the feature space would be classified. White background indicates that a possible test instance in that region would be classified as unknown.

5.3 Additional considerations

The work on SSVM has allowed us to observe other support vector classifiers in a different perspective. For this reason, in this section, we present additional considerations regarding classification with geometric classifiers other than binary SVM with the one-vs-all strategy. In Section 5.3.1, we visit the formulation of SVM without the bias term and analyze its behavior on open-set scenarios. In Section 5.3.2, we analyze the employment of the one-vs-one approach for multiclass extension along with SVM. Finally, we describe a straightforward adaptation of OCSVM that better takes advantage of possible extra classes available for training, when generating its model in an open-set scenario.

5.3.1 Support Vector Machines without bias term

The theoretical foundation of SSVM indicates that other extensions of geometric classifiers for open-set recognition can be obtained taking into account the factor that determines a bounded/unbounded open-space risk, as shown in Section 5.2.1. For instance, consider SVM without explicit bias term [Vogt, 2002, Kecman et al., 2005], for which $b = 0$ is implicit. Its decision function is similar to the one of SVM with bias term; as shown in Equation (5.12); compared to Equation (5.7), only the bias term is missing.

$$f(\mathbf{x}) = \text{sign}\left(\sum_{i=1}^{m} y_i \alpha_i K(\mathbf{x}_i, \mathbf{x})\right). \tag{5.12}$$

For test samples far away from support vectors, we have that $\sum_{i=1}^{m} y_i \alpha_i K(\mathbf{x}_i, \mathbf{x})$ converges to 0 from the bottom or from the above, depending on the training samples. Consequently, a bounded PLOS cannot be ensured in all cases for this method as is. However, in practice, as the Gaussian bell precipitates to 0 and due to limitations on float point representation, PLOS is bounded if the sign function considers only values *strictly greater* than 0 as positive.

The main difference from the SVM without bias term to the traditional SVM is that the constraint in Equation (5.6) does not exist in the dual formulation. SSVM optimizes the risk of the unknown, that is equivalent to say to minimize the bias term b, with a substitute to that constraint, as defined on Equation (5.11). Consequently, SVM without bias term cannot optimize the risk of the unknown as performed by SSVM, however, a straightforward way an SVM without bias term can bound the PLOS is by introducing an *artificial bias term* ϵ with a negative value on the decision function of Equation (5.12) at prediction time, i.e., after the model is obtained. The same Theorem 1 applies to SVM without bias term if the artificial bias term $\epsilon < 0$ is introduced, as in Equation (5.13). It is not as elegant as the SSVM optimization problem that takes into account the risk of the unknown during optimization, however, equivalent experimental results might be obtained if ϵ is properly grid searched. We leave the problem of grid searching optimal ϵ as future work. However, in Chapter 6 we present results with fixed ϵ.

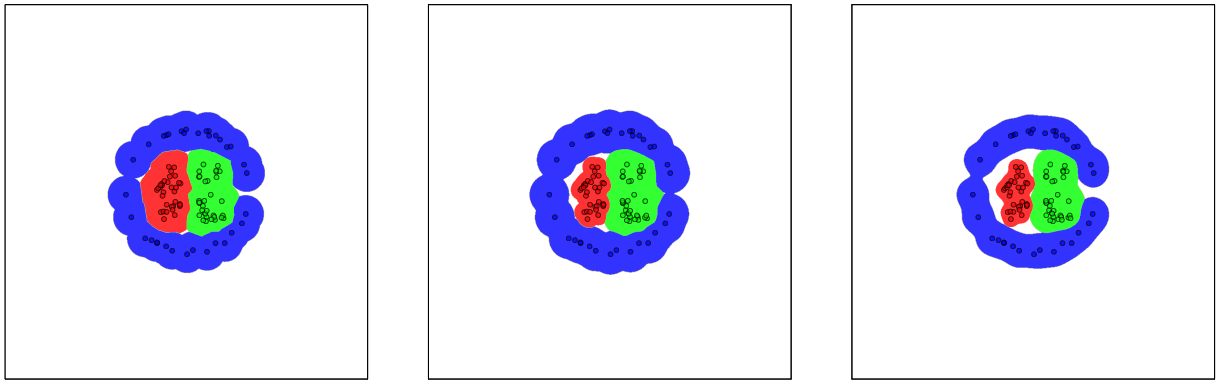
$$f(\mathbf{x}) = \text{sign}\left(\sum_{i=1}^{m} y_i \alpha_i K(\mathbf{x}_i, \mathbf{x}) + \epsilon\right) \tag{5.13}$$

For better gaining an intuition about the SVM without bias term, in Figure 5.3, we present its behavior with and without the artificial bias term. As we have considered the sign function to classify as positive only for values strictly greater than 0, Figure 5.3a presents a bounded KLOS for SVM without bias.

5.3.2 Support Vector Machines with one-vs-one multiclass-from-binary approach

Usually, at least for closed-set classification problems, the one-vs-one approach is preferable over the one-vs-all, for multiclass extension of SVMs. It creates $n(n-1)/2$ smaller problems compared to the n larger problems of one-vs-all approach, so that in practice, one-vs-one usually runs faster. Then, for final decision, it uses a *voting scheme* for choosing the class of a test sample. Differently than one-vs-all, the one-vs-one approach does not have a direct criteria for allowing the multiclass-from-binary SVM to classify a sample as unknown. A straightforward approach, however, used in practice, is to estimate probabilities, combine them in the multiclass level, and establish a threshold on the probability to the most probable class.[3] A natural question that arises here is about the minimal threshold such that KLOS would be bounded or, in other words, what is the probability

[3] We have talked with some authors of previous work on open-set recognition and, in fact, they seem to use this approach very often as baseline, as the one-vs-one approach is the only one implemented in libraries like LIBSVM and, furthermore, those implementations already calculate probability estimates.



(a) SVM without bias.

(b) SVM without bias term with artificial bias set as -1×10^{-6}.

(c) SVM without bias term with artificial bias set as -1×10^{-1}.

Figure 5.3: Behavior analysis of SVM without bias term. Figures were generated for the Boat dataset. Figure (a) depicts the behavior of SVM without bias term as is, i.e., without an artificial bias term introduced after training. Figures (b) and (c) show the behavior of SVM without bias term by introducing two different values of artificial bias term. All figures were generated with closed-set grid search. As grid search was performed, parameters to fit each of those classifiers differ from image to image. The small circles represent the training samples from the dataset and their colors represent their classes. A colored background represents the class in which a possible test instance in the corresponding position of the feature space would be classified. White background indicates that a possible test instance in that region would be classified as unknown.

in the open space?

As we have analyzed the abating behavior of RBF kernel in Section 5.2.1, we can infer that the probability for any sample in the open space would approach a constant value. The same is not true for kernels that do not satisfy the property of Equation (5.9). The analysis in this section, then, considers RBF kernels.

A well-known approach for estimating probabilities in a binary problem for SVM is proposed by Platt [2000] and later improved by Lin et al. [2007]. They use a parametric model to fit the posterior $P(y = 1|f)$ as in Equation (5.14), i.e., a sigmoid form is assumed to fit the data, according to their empirical evaluation.

$$P(y = 1|f) = \frac{1}{1 + e^{Af+B}}, \tag{5.14}$$

in which A and the bias B are the parameters of the sigmoid obtained by minimizing the negative log likelihood of the training data.

Then, any method for obtaining multiclass probabilities from pairwise posteriors [Wu et al., 2004] can be employed for calculating the final probability. For instance, consider the "Second Approach" of Wu et al. [2004]. Given a test sample $\mathbf{x}$, it consists on solving

an optimization problem as follows.

$$\min_{\mathbf{p}} \sum_{i=1}^{n} \sum_{j:j\neq i} (r_{ji}p_i - r_{ij}p_j)^2, \tag{5.15}$$

$$\text{s.t.} \sum_{i=1}^{n} p_i = 1, \tag{5.16}$$

$$p_i \geq 0, \forall i, \tag{5.17}$$

in which $r_{ij} = P(y = 1|f)$ and $r_{ji} = 1 - P(y = 1|f)$, considering the parameters A^{ij} and B^{ij} obtained for the class i (as positive) vs. class j (as negative) problem.

Those methods [Platt, 2000, Lin et al., 2007, Wu et al., 2004], in fact, are implemented in LIBSVM and are commonly used. For each binary problem, parameters A and B are obtained on training phase by fitting models to smaller problems, employing cross-validation to avoid bias, to obtain the values of f. The problem of Equation (5.15) is only solved on prediction phase.

As we have observed in Section 5.2.1, for a test sample $\mathbf{x}$ in the open space, $f(\mathbf{x})$ (without sign function) approaches b. Then, we can simply estimate the probability on the open space by replacing f by b in Equation (5.14) and solve the problem of Equation (5.15) at training time. Each value p_i, $i \in \{1, \ldots, n\}$, represents the probability the model would assign, as belonging to class i, for any sample in the open space. This way,

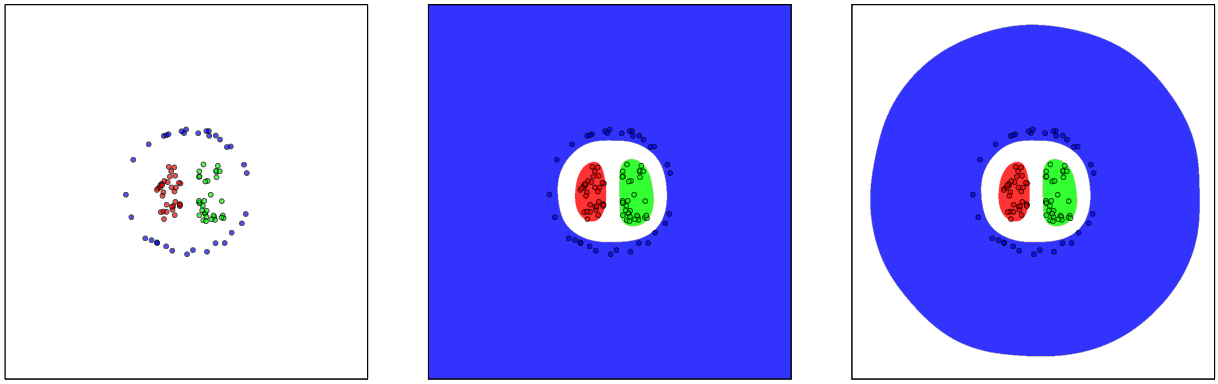
$$T = \max_i p_i \tag{5.18}$$

is the maximum threshold such that KLOS is *not* bounded and for any positive value of ϵ, a threshold of $T + \epsilon$ can ensure a bounded KLOS.

For empirical evidence of this property, in Figure 5.4, we show the decision boundaries of SVM with the one-vs-one approach when the threshold for classifying as unknown is below and above a minimum required threshold T of Equation (5.18). KLOS is unbounded/bounded when the rejection threshold is below/above T. And we can infer from Figure 5.4a that the probability estimated for some regions of the feature space—the white regions close to the training samples—is smaller than for the open space, which indicates that SVM with one-vs-one strategy also is affected by the problem of mistaking doubtful test instances with unknown ones.

5.3.3 One-Class Support Vector Machines with open-set grid search

As stated before, OCSVM has a poor specialization-generalization ability, as it neglects possible information from other available classes when fitting a model considering a certain class as positive. Consequently, another straightforward extension of a multiclass-from-binary one-vs-all implementation composed of OCSVMs is to consider those extra classes at least during the grid search procedure. This way, even when the model does not consider the separation among known classes, parameters that lead to models that disrespect this separation would be penalized. In such a way, it is more likely better parameters are



Boat dataset.

(a) SVM with one-vs-one approach and rejection threshold as $T - \epsilon$.

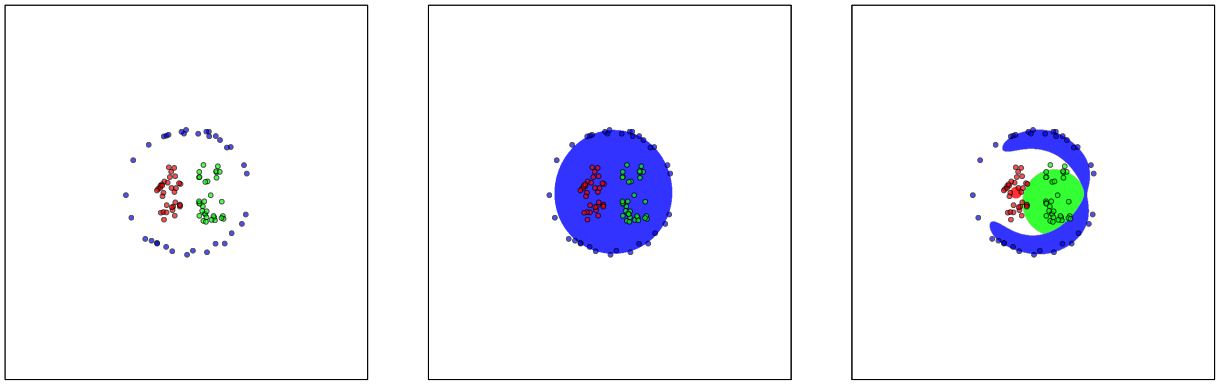
(b) SVM with one-vs-one approach and rejection threshold as $T + \epsilon$.

Figure 5.4: Behavior analysis of SVM with one-vs-one approach. The Boat dataset is depicted on the far left. Figure (a) depicts the behavior of the classifier when the threshold for rejection is *below* a minimum required threshold. Figure (b) depicts the behavior of the classifier when the threshold for rejection is *above* a minimum required threshold. T is obtained according to Equation (5.18). This behavior is obtained by fixing $C = 1$, $\gamma = 2^4$, and $\epsilon = 1 \times 10^{-6}$. In this example, $T = 0.875\,697$. The small circles represent the training samples from the dataset and their colors represent their classes. A colored background represents the class in which a possible test instance in the corresponding position of the feature space would be classified. White background indicates that a possible test instance in that region would be classified as unknown.

obtained for the multiclass model.

The same idea can be employed along with any one-class classifier, e.g., the SVDD. As a matter of fact, those implementations are present on the open-set grid search versions of those classifiers for the experiments in Chapter 6. In Chapter 6, we show the effectiveness of this simple approach.

Figure 5.5 depicts the difference of behavior of OCSVM with those two possible implementations. We observe in Figure 5.5a that the model generated for one of the classes (the ring-shaped blue class) predominates over the other, as it does not take into account how well it would predict when considering extra classes. Differently, when using other available classes for validation during grid search, as depicted in Figure 5.5b, models generated for each of the known classes allow a better separation among training classes. However, we still observe a highly specialized behavior of this classifier.



Boat dataset.

(a) OCSVM with closed-set grid search.

(b) OCSVM with open-set grid search.

Figure 5.5: Behavior analysis of OCSVM with closed- and open-set grid search. The Boat dataset is depicted on the far left. Figure (a) depicts the behavior of OCSVM when it uses only a single class on validation, during grid search. Figure (b) depicts the behavior of OCSVM when additional known classes not employed for generating the one-class model for grid search are included in the validation set so that obtained parameters generalize better. The small circles represent the training samples from the dataset and their colors represent their classes. A colored background represents the class in which a possible test instance in the corresponding position of the feature space would be classified. White background indicates that a possible test instance in that region would be classified as unknown.

Chapter 6

Experiments with distance-based and geometric classifiers

In this chapter, we present experiments performed for both the OSNN and SSVM methods compared to state-of-the-art baselines. Those are the first experiments we perform, in Sections 6.1 and 6.2, respectively, aiming at demonstrating the effectiveness of the proposed methods. In Section 6.3, we perform some experimental evaluation of SVM with the one-vs-all approach aiming at showing empirical evidence for an explanation why SVM with the one-vs-all approach, as is, performs reasonably well in open-set scenarios. For the experiments in Section 6.4, we have implemented open-set grid search on the state-of-the-art baselines aiming at showing the effectiveness of employing it in general methods for open-set recognition. In Section 6.5, we perform comparison among the best alternatives. In Sections 6.6 and 6.7, we present strong empirical evidence of the hypothesis we have carried along this work: bounding the KLOS is required for open-set recognition. For experiments in Section 6.6, we employ the SVM without bias term and, in Section 6.7, we employ SVM with the one-vs-one approach and play with its minimal required threshold on probability estimates, as derived in Section 5.3.2, for ensuring a bounded KLOS. Finally, in Section 6.8, we employ OSNN for showing how λ_r of Normalized Accuracy (NA) can be properly defined during grid search aiming at training the method to be more or less restrictive to the false acceptance of unknown samples.

For comparison of methods, we have defined an experimental setup in which 3, 6, 9, and 12 training classes are considered to be available for the methods. For statistical evaluation, for each number n of available classes, we have performed 10 paired experiments in which n classes of each dataset are chosen at random. We have employed both Binomial and Wilcoxon statistical tests along with Holm method to control the family-wise error rate when accounting for multiple comparisons [Demšar, 2006]. Along this chapter, we present tables of statistical tests only for Binomial tests and the equivalent for Wilcoxon tests are presented in Appendix C.

For the experiments, we have employed seven datasets from multiple domains. In the 15-Scenes [Lazebnik et al., 2006] dataset, images from a scene classification problem are represented by a bag-of-visual-word vector created with soft assignment [van Gemert et al., 2010] and max pooling [Boureau et al., 2010], based on a codebook of 1000 Scale Invariant Feature Transform (SIFT) codewords [Lowe, 2004]. The KRKOPT [Olson et al.,

Dataset	# classes	# samples	# features	# samples/class		
				mean	min	max
15-Scenes	15	4485	1000	299	210	410
KRKOPT	18	28,056	6	1559	27	4553
Letter	26	20,000	16	769	734	813
KDDCUP	32[1]	10,237	41	320	11	500
Auslan	95	146,949	22	1547	1390	1938
Caltech-256	256	29,780	1000	116	80	800
ALOI	1000	108,000	128	108	108	108

Table 6.1: General characteristics of the datasets employed for the experiments.

2017, Bain, 1994] is a dataset of chess endgames representing white king and rook against black king (KRK) in which the outcome represents optimal depth-of-win for white in 0–16 moves or draw. The Letter [Frey and Slate, 1991, Michie et al., 1994] dataset represents letters of the English alphabet (black-and-white rectangular pixel displays). The KDDCUP [Stolfo et al., 2000] dataset represents an intrusion detection problem on a military network environment and its feature vectors combine continuous and symbolic features. In the Auslan [Kadous, 2002] dataset, for a sign language recognition problem, the data was acquired using two Fifth Dimension Technologies (5DT) gloves hardware and two Ascension Flock-of-Birds magnetic position trackers. In the Caltech-256 [Griffin et al., 2007] dataset, comprising an object recognition problem, feature vectors consider a bag-of-visual-words characterization approach, with features acquired with dense sampling, SIFT descriptor for the points of interest, hard assignment [van Gemert et al., 2010], and average pooling [Boureau et al., 2010]. Finally, for the ALOI [Geusebroek et al., 2005] dataset—also an object recognition problem—features were extracted with the Border/Interior (BIC) descriptor [Stehling et al., 2002]. Those datasets or other datasets could be used with different characterizations. However, in this work, we focus on the learning part of the problem rather than on the feature characterization one. In Table 6.1, we summarize the main features of the considered datasets in terms of number of samples, number of classes, dimensionality, and approximate number of samples per class.

Besides the evaluation measures defined in Section 2.3, in this chapter, we employ macro- (OSFM_M) and Micro-averaging Open-set F-measure (OSFM_μ) of previous work [Mendes Júnior, 2014] as well as traditional macro- (FM_M) and Micro-averaging F-measure (FM_μ) [Sokolova and Lapalme, 2009]. Throughout this chapter, we refer to NA, Harmonic Normalized Accuracy (HNA), OSFM_M, OSFM_μ, FM_M, and FM_μ as *global measures*, as they consider accuracies on both known and unknown instances of the test set. On the other hand, we refer to Accuracy on Known Samples (AKS) and Accuracy on Unknown Samples (AUS) as *partial measures*.

[1] Aiming at keeping the same setup across all datasets, for KDDCUP, we have joined training and testing datasets and partitioned the data into those sets for the experiments. As WSVM cannot fit the model with classes with few samples, aiming at a paired experiment, we have kept only the classes with 10 or more samples.

6.1 Open-Set Nearest Neighbors versus baselines

In this section, we compare OSNN with distance-based baselines. We have included inherently closed-set NN and OPF for drawing the worst case scenario. We have also considered NN by employing a threshold on the distance to the nearest neighbor. We refer to those implementations as Thresholded Nearest Neighbor (TNN). We have considered a straightforward implementation TNN_E that performs the external grid search and establishes a single threshold for the multiclass problem [Fukunaga, 1990] and we have also considered TNN_I, that performs internal grid search and establishes a per-class threshold [Muzzolini et al., 1998].

In Figures 6.1 and 6.2, we present results regarding HNA for all datasets of Table 6.1.[2] With exception of 15-Scenes and Caltech-256 datasets, OSNN obtains better results than its baseline OSNN^CV, which is only able to reject doubtful samples but not to bound the KLOS. The same observation applies to OSOPF of previous work [Mendes Júnior, 2014, Neira et al., 2018] compared to its baseline OSOPF^CV. With few exceptions, in general, OSNN improves over its more prominent baseline, the OSOPF.

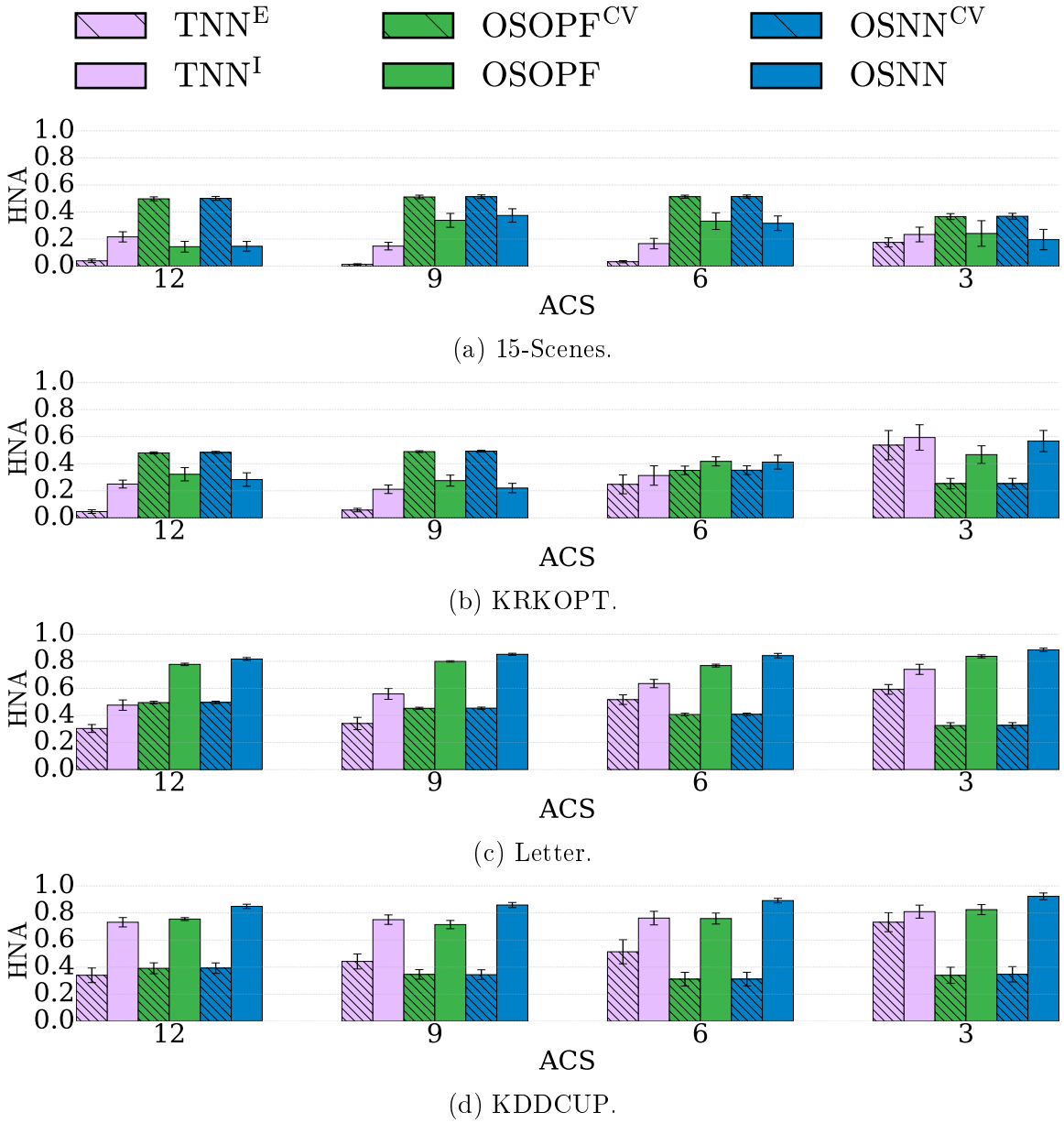
We think the exceptions for 15-Scenes and Caltech-256 datasets happen due to their high dimensional feature space. For those datasets, AKS for OSNN suffer while AUS improves compared to OSNN^CV. It indicates that in high dimensional spaces, the size of the "intermediate region" among known classes becomes more significative (in terms of the defined threshold). As mentioned in Section 4.3, it becomes more problematic for datasets with a high overlapping among the known classes, as OSNN—and also OSOPF—tends to reject any instance in those regions, as the ratio in Equation (4.5) approaches 1.

With those results, we also observe the effectiveness of establishing a threshold per class, as of TNN_I of Muzzolini et al. [1998], instead of a single global rejection threshold on the distance, as implemented in TNN_E [Fukunaga, 1990], for classifying as unknown. Both Binomial and Wilcoxon statistical tests present more than 99% of confidence evincing the superiority of TNN_I compared to TNN_E for global measures and for AUS, however, AKS for TNN_E is better than for TNN_I, also with more than 99% confidence. Anyhow, as presented in Table 6.2, OSNN outperforms TNN_I for virtually all measures, also with 99% of confidence. In fact, as seen in Table 6.2, except for AKS, OSNN performs better than any baseline with more than 99% of confidence.

6.2 Specialized Support Vector Machines versus baselines

In this section, we compare SSVM with its SVM-based baselines. General results for HNA are presented in Figures 6.3 and 6.4. In these figures, all methods perform closed-set grid search: the open-set grid search was ignored here to avoid introducing an extra factor in the analysis. Specific analysis for the influence of open-set grid search is presented in

[2] We have excluded NN and OPF from Figures 6.1 and 6.2 because their HNA is 0 for every dataset (their AUS is always 0), as they are closed-set methods. Anyhow, we consider those methods on the statistical tests for other measures.



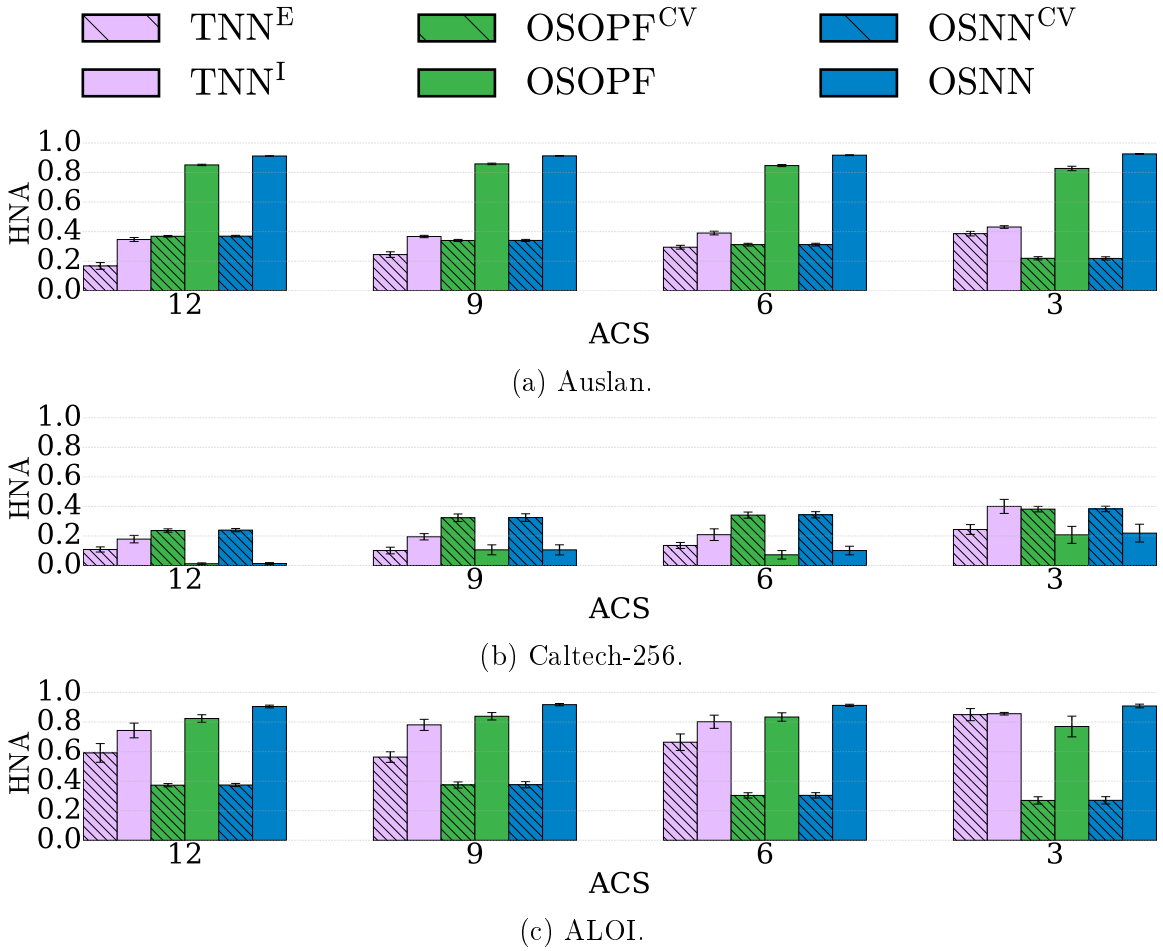
(a) 15-Scenes.

(b) KRKOPT.

(c) Letter.

(d) KDDCUP.

Figure 6.1: Comparison of OSNN with baselines (part I). Results for 15-Scenes, KRKOPT, Letter, and KDDCUP datasets regarding HNA considering 3, 6, 9, and 12 available classes (ACS).



(a) Auslan.

(b) Caltech-256.

(c) ALOI.

Figure 6.2: Comparison of OSNN with baselines (part II). Results for Auslan, Caltech-256, and ALOI datasets regarding HNA considering 3, 6, 9, and 12 available classes (ACS).

Measure	TNN_E	TNN_I	$OSOPF^{CV}$	OSOPF	$OSNN^{CV}$
NA	**<.0001***	**<.0001***	**<.0001***	**<.0001***	**<.0001***
HNA	**<.0001***	**<.0001***	**<.0001***	**<.0001***	**<.0001***
$OSFM_M$	**<.0001***	**<.0001***	**<.0001***	**<.0001***	**<.0001***
$OSFM_\mu$	**<.0001***	**<.0001***	**<.0001***	**<.0001***	**<.0001***
FM_M	**<.0001***	**0.0033***	**<.0001***	**<.0001***	**<.0001***
FM_μ	**<.0001***	**<.0001***	**<.0001***	**<.0001***	**<.0001***
AKS	*<.0001**	*<.0001**	*<.0001**	**<.0001***	*<.0001**
AUS	**<.0001***	**<.0001***	**<.0001***	**0.0049***	**<.0001***

Table 6.2: Binomial statistical tests comparing the OSNN with baselines. Each cell compares results for all datasets considering all number of available classes. **Bold** means there is statistical difference with 95% of confidence. "*" indicates the statistical difference is with 99% of confidence. And **<.0001*** indicates the statistical difference is with 99.99% of confidence. *Emphasized* indicates the method in the column obtains better performance for the measure associated with that row.

Measure	SVM_C	$OCSVM_C$	DBC_C	OVS_C	$WSVM_C$	$PISVM_C$	$SVDD_C$
NA	**<.0001***	**<.0001***	**<.0001***	**<.0001***	**<.0001***	**<.0001***	**<.0001***
HNA	**<.0001***	**<.0001***	**<.0001***	**<.0001***	**<.0001***	**<.0001***	**<.0001***
$OSFM_M$	**<.0001***	**<.0001***	**<.0001***	**<.0001***	**<.0001***	**<.0001***	**<.0001***
$OSFM_\mu$	**<.0001***	**<.0001***	**<.0001***	**<.0001***	**<.0001***	**<.0001***	**<.0001***
FM_M	**<.0001***	**<.0001***	**<.0001***	**<.0001***	**<.0001***	**<.0001***	**<.0001***
FM_μ	**<.0001***	**0.0015***	**<.0001***	**<.0001***	**<.0001***	**<.0001***	**<.0001***
AKS	*<.0001**	**<.0001***	*<.0001**	**0.0002***	*<.0001**	*<.0001**	**<.0001***
AUS	**<.0001***	1.0000	**<.0001***	**<.0001***	**<.0001***	**<.0001***	**<.0001***

Table 6.3: Binomial statistical tests comparing the $SSVM_C$ with baselines. Each cell compares results for all datasets considering all number of available classes. **Bold** means there is statistical difference with 95% of confidence. "*" indicates the statistical difference is with 99% of confidence. And **<.0001*** indicates the statistical difference is with 99.99% of confidence. *Emphasized* indicates the method in the column obtains better performance for the measure associated with that row.

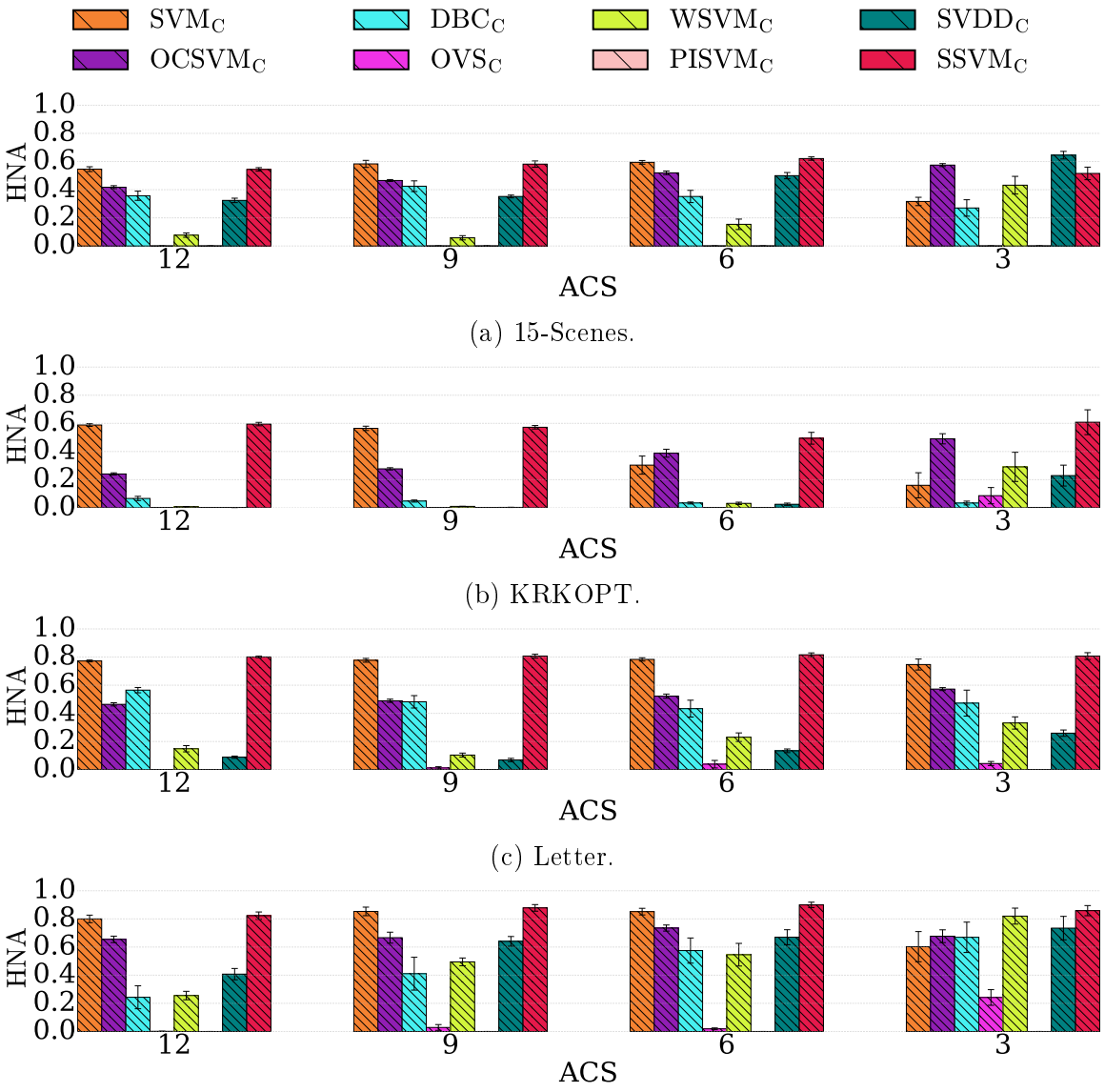
Section 6.4 and a comparison of SSVM (and OSNN) with baselines performing open-set grid search is presented in Section 6.5.

We observe in Figures 6.3 and 6.4 that SSVM outperforms baselines in most cases and is robust across datasets. We also observe the low accuracy of the OVS method. We attribute this low-accuracy behavior of OVS to its strictly-linear constraint. PISVM obtains HNA equals to 0 for all cases. It is due to its low performance on unknown samples, which leads to AUS equal to 0. We should remember here that PISVM was proposed along with the cross-class validation, which is a form of open-set grid search. Then, that is probably the reason the authors of PISVM have proposed it along with cross-class validation: it works better for estimating its parameters. All other methods, except those, perform reasonably well. Anyhow, with the statistical tests presented in Table 6.3, we confirm that SSVM clearly outperforms its baselines when all of them are employed with closed-set grid search.

6.3 Behavior analysis of Support Vector Machines

We have observed in Section 6.2, Figures 6.3 and 6.4, that traditional SVM has performed reasonably well, even with the simple closed-set grid search, when employed with the one-vs-all approach. We hypothesize that SVM with RBF kernel employing a one-vs-all strategy, as is, is able to bound the PLOS in most cases. An intuitive—and informal, however—explanation is that, when training a binary problem of a single positive class versus a negative class comprising a set of $n-1$ distinct classes, it is more likely that samples from the negative classes will be "around" the samples of the positive class, hence creating the non-linear separation hyperplane that bounds the PLOS.

For example, consider Figure 6.5a, in which the behavior of SVM is presented for



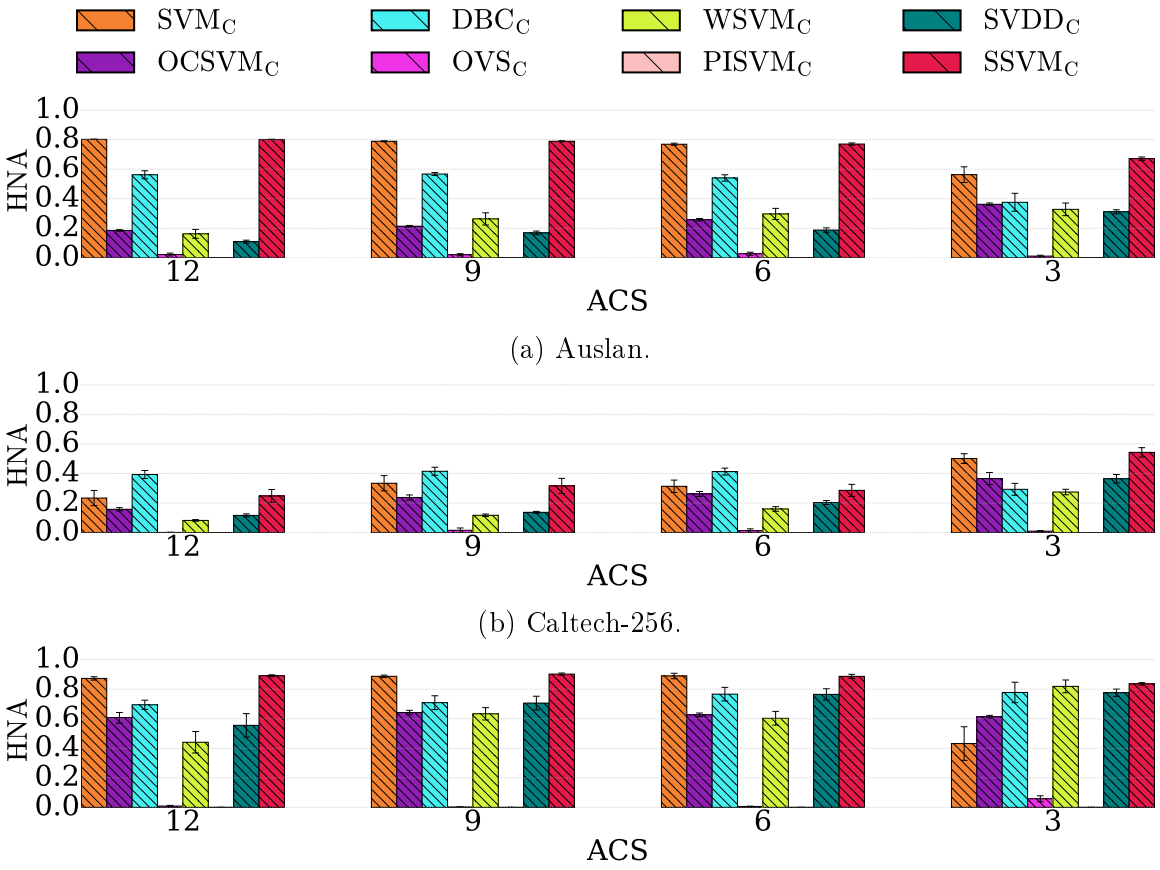
(a) 15-Scenes.

(b) KRKOPT.

(c) Letter.

(d) KDDCUP.

Figure 6.3: Comparison of SSVM with baselines (part I). Results for 15-Scenes, KRKOPT, Letter, and KDDCUP datasets regarding HNA considering 3, 6, 9, and 12 available classes (ACS).

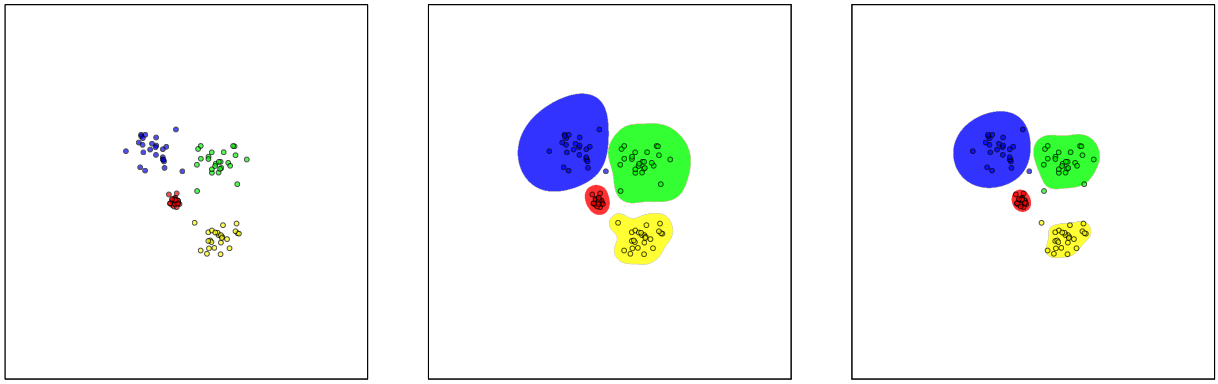


(a) Auslan.

(b) Caltech-256.

(c) ALOI.

Figure 6.4: Comparison of SSVM with baselines (part II). Results for Auslan, Caltech-256, and ALOI datasets regarding HNA considering 3, 6, 9, and 12 available classes (ACS).



Four-Gauss dataset. (a) SVM. (b) SSVM.

Figure 6.5: Behavior analysis of SVM with one-vs-all approach. The Four-Gauss dataset is depicted on the far left. The behavior presented in Figures (a) and (b) was obtained with SVM and SSVM, respectively, both with RBF kernel along with a one-vs-all strategy for multiclass-from-binary extension. Figure (a) shows that SVM is able to create a bounded KLOS, which means every binary SVM generates a negative bias term. As expected, SSVM in Figure (b), also generates a bounded KLOS and, compared to SVM, it creates a more specialized behavior. The small circles represent the training samples from the dataset and their colors represent their classes. A colored background represents the class in which a possible test instance in the corresponding position of the feature space would be classified. White background indicates that a possible test instance in that region would be classified as unknown.

Four-Gauss [Kuncheva and Hadjitodorov, 2004] dataset.[3] Consider that SVM in that figure has employed the one-vs-all approach so that for every class we have a binary classifier trained considering that class as positive an all other classes as a single negative class. For instance, the less-sparse red class versus other classes would more likely create a decision hyperplane tending to bound around the red class instead of bounding around the negative class. Recall that we know from Theorem 1 that, when RBF kernel is used, either the region classified as positive or the region classified as negative is bounded and, also, either of them is unbounded. For the red class of Figure 6.5, intuitively, we would say the positive class is the one to be bounded. It is equivalent to say that the bias term of that binary classifier is likely to be negative. As a side note, notice in Figure 6.5b that SSVM also bounds the KLOS and additionally presents a more specialized behavior than SVM.

Aiming at confirming this hypothesis, we have analyzed the percentage of cases for which binary SVM classifiers "correctly" obtains a negative bias term after the optimization process, when employed by performing the one-vs-all strategy. For comparison purposes, we also have considered the SVM along with the one-vs-one approach, so that both positive and negative classes of each binary classifier would comprise a single known class. In this case, there would be no preference on bounding the positively-labeled space over the negatively-labeled one. In fact, as shown in Table 6.4, in general, over 97% of the binary SVMs that compose the one-vs-all strategy is able to properly bound the PLOS.

[3] The Four-Gauss dataset comprises a 4-classes problem and—similarly to the Boat dataset employed since Section 4.3—also contains 2-dimensional data, proper for behavior visualization of the classifiers.

Dataset	one-vs-all	one-vs-one
15-Scenes	99.00%	56.92%
KRKOPT	95.67%	40.25%
Letter	99.67%	55.75%
KDDCUP	99.33%	49.79%
Auslan	99.00%	42.50%
Caltech-256	98.00%	48.75%
ALOI	97.67%	52.50%

Table 6.4: Percentage of binary classifiers with negative bias term, for each dataset, obtained by the Support Vector Machines trained with one-vs-all and one-vs-one strategies.

Measure	SVM	OCSVM	DBC	OVS	WSVM	PISVM	SVDD	SSVM
NA	*0.0101*	**<.0001***	**<.0001***	0.4373	**<.0001***	**<.0001***	**<.0001***	**0.0004***
HNA	*0.0101*	**<.0001***	**<.0001***	**<.0001***	**<.0001***	**<.0001***	*<.0001**	*0.0829*
$OSFM_M$	0.0829	**0.0006***	**<.0001***	0.0637	**<.0001***	**<.0001***	*<.0001**	**<.0001***
$OSFM_\mu$	**0.0101**	**<.0001***	**<.0001***	*0.2561*	**<.0001***	**<.0001***	*<.0001**	**<.0001***
FM_M	0.3701	**<.0001***	**0.0101**	**0.0022***	**0.0015***	**<.0001***	*<.0001**	**0.0006***
FM_μ	**<.0001***	**<.0001***	**<.0001***	**<.0001***	**<.0001***	**<.0001***	**<.0001***	**<.0001***
AKS	*<.0001**	**0.0003***	*<.0001**	*<.0001**	*<.0001**	*<.0001**	*<.0001**	*<.0001**
AUS	**<.0001***	**<.0001***	**<.0001***	**<.0001***	**<.0001***	**<.0001***	**<.0001***	**<.0001***

Table 6.5: Binomial statistical tests for the pairwise comparison between closed- and open-set grid search implementations of the methods. Each cell compares results for all datasets considering all number of available classes. **Bold** means there is statistical difference with 95% of confidence. "*" indicates the statistical difference is with 99% of confidence. And **<.0001*** indicates the statistical difference is with 99.99% of confidence. *Emphasized* means the version with closed-set grid search obtains better performance for the measure associated with that row.

On the other hand, as expected, for the one-vs-one approach, it happens only with around 50% of the cases.

6.4 Effectiveness of open-set grid search

Aiming at a per-factor analysis, in this section, we analyze the influence of open-set grid search over SSVM as well as all SVM-based baselines. We have employed all datasets listed in Table 6.1 and the same setup described in the beginning of this chapter—10 experiments per number of available classes, for 3, 6, 9, and 12 available classes—for both closed- and open-set grid search variations for each method. From Table 6.5, we confirm that, for most of the methods, their versions with open-set grid search perform better than their counterparts with closed-set grid search. For SVM and SVDD, however, it seems that the best grid search alternative changes from cases to case—or from measure to measure.

6.5 Comparison among best alternatives

In Section 6.1, we have compared OSNN only with its distance-based baselines and, in Section 6.2, we have compared SSVM only against geometric classifiers. Also, notice that open-set grid search has not been employed for the results presented in those sections. In this section, we compare OSNN and SSVM with baseline methods employing open-set grid search. Recall, from Section 6.2, that PISVM has performed poorly because it was not using the cross-class validation. In this section, we perform a fair comparison with PISVM, as the open-set grid search employed here has the same principle of the cross-class validation proposed by the authors of PISVM.

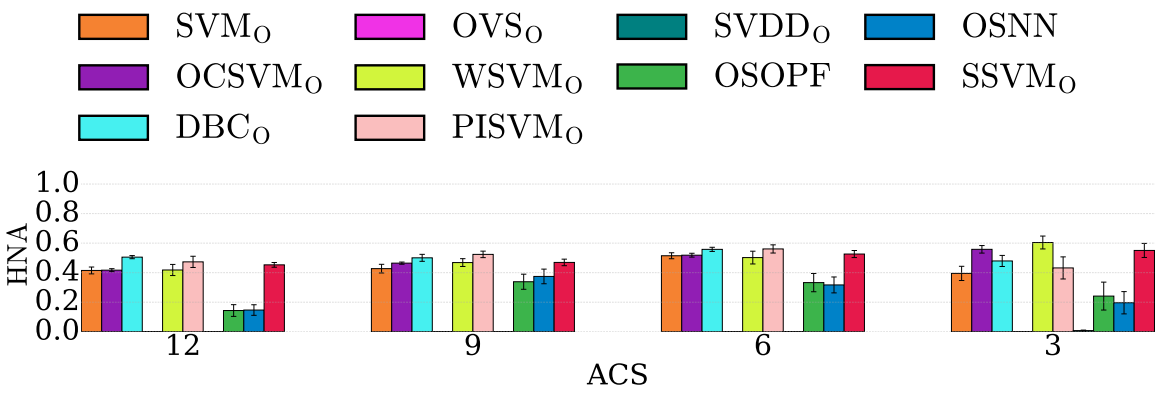
We can observe in Figures 6.6 and 6.7 that both OSNN and SSVM present the highest classification accuracies, with few exceptions. In general, SSVM shows a more robust behavior than OSNN, as it can keep a higher accuracy in the cases for which OSNN behavior suffers, e.g., for 15-Scenes, KRKOPT, and Caltech-256. On the other hand, there are a few datasets for which OSNN obtains a slightly better accuracy than SSVM, e.g., Letter, Auslan, and ALOI. Overall, we conclude that SSVM performs slightly better than OSNN, as we notice by comparing Table 6.6 with Table 6.7.

In Table 6.6, we compare OSNN with the baselines. We observe in this table that OSNN outperforms its baselines in most cases, however, PISVM performs better than OSNN for OSFM_M and FM_M even though OSNN surpasses PISVM for FM_μ. We also observe that OSNN, compared to baselines proposed for open-set scenarios—DBC, OVS, WSVM, PISVM—has a more restrictive behavior against accepting unknown samples, as points out the statistical difference for AUS. On the other hand, AKS is worse, also with statistical significance, compared to those open-set methods, as usually AUS and AKS trade off. Anyhow, global metrics indicate an improved behavior for the OSNN compared to its baselines, in general.

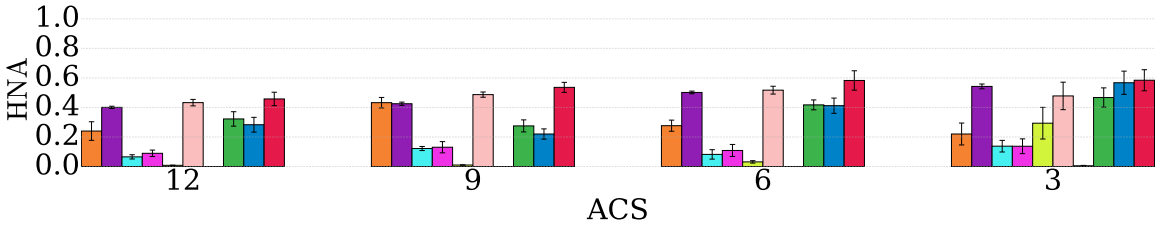
When comparing SSVM with its baselines, as in Table 6.7, results improve. Still, the more competing baseline is PISVM. However, in this case, we observe a favorable performance for SSVM: out of the six global measures—AKS and AUS assess only partial performance—SSVM improves with statistical significance of 95% for four of them, when compared to PISVM. As for OSNN, we also observe that SSVM has a more restrictive behavior on accepting unknown instances as known, compared to most of the baselines.

6.6 Assessing Support Vector Machines without bias term

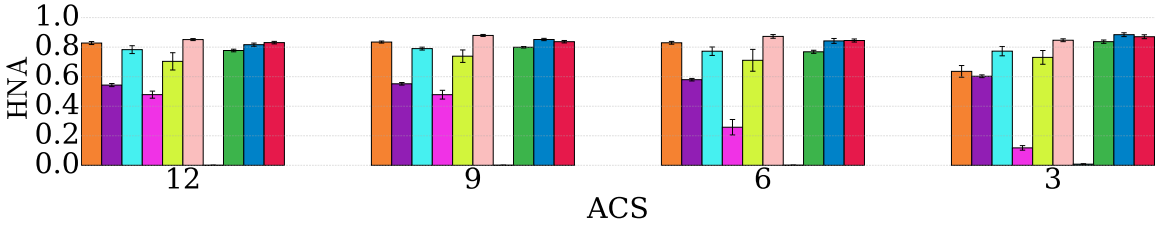
In this section—and in Section 6.7, as we shall see—our objective is to evince the importance of not only bounding the KLOS but also decreasing it as much as possible. There usually is a trade off between decreasing the KLOS and properly recognizing known samples. To evince the importance of decreasing the KLOS—here, we assume the one-vs-all strategy is employed—we use the Support Vector Machines without bias term (SVM^{WB}). As presented in Section 5.3.1, SVM^{WB} is only able to bound the KLOS due to the fast convergence of RBF kernel to 0—in function of distance, as in Equation (5.9)—and when



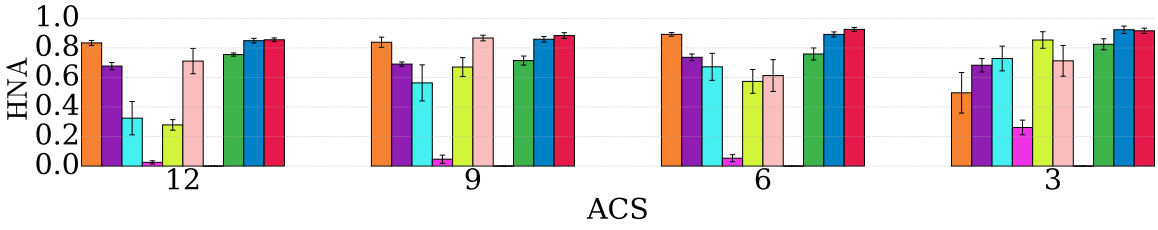
(a) 15-Scenes.



(b) KRKOPT.

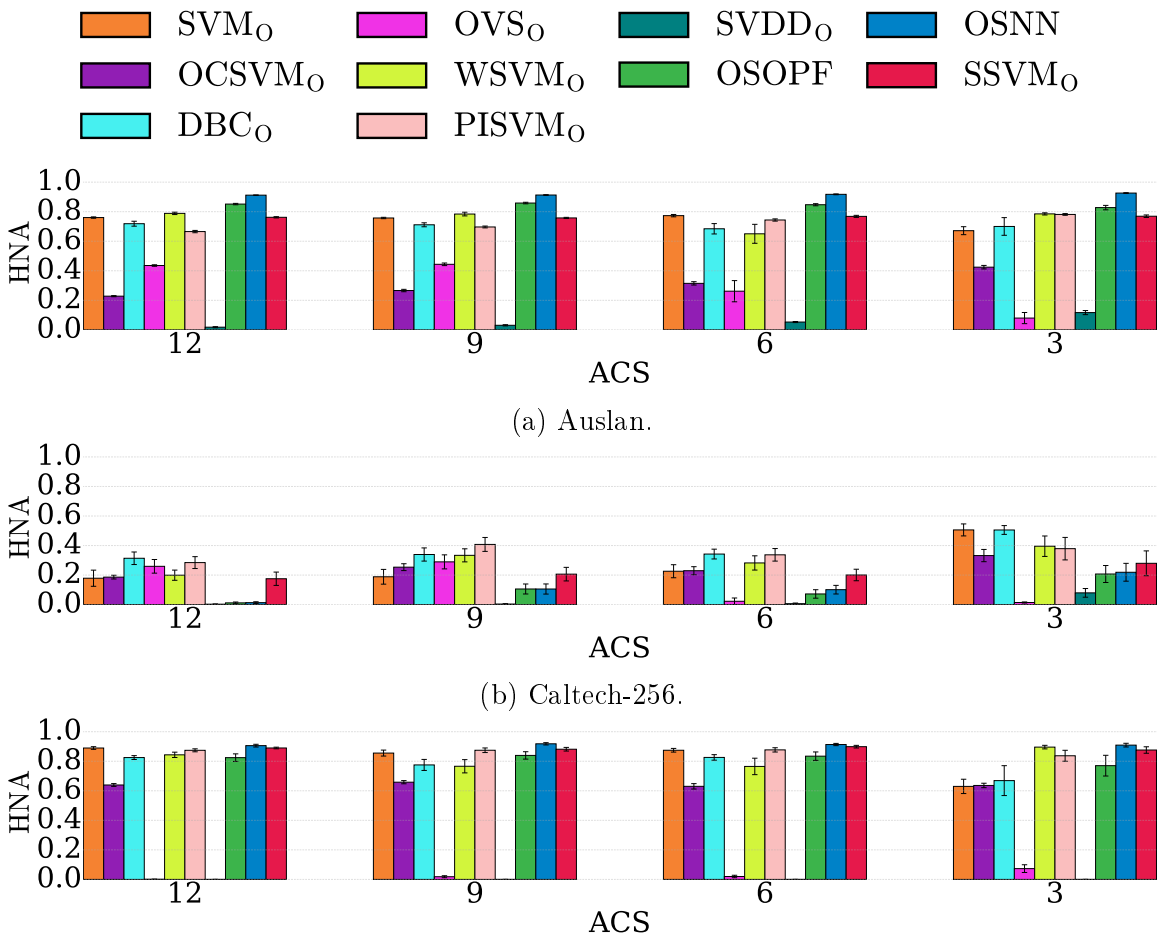


(c) Letter.



(d) KDDCUP.

Figure 6.6: Comparison among best methods (part I). Results for 15-Scenes, KRKOPT, Letter, and KDDCUP datasets regarding HNA considering 3, 6, 9, and 12 available classes (ACS).



(a) Auslan.

(b) Caltech-256.

(c) ALOI.

Figure 6.7: Comparison among best methods (part II). Results for Auslan, Caltech-256, and ALOI datasets regarding HNA considering 3, 6, 9, and 12 available classes (ACS).

Measure	SVM$_O$	OCSVM$_O$	DBC$_O$	OVS$_O$	WSVM$_O$	PISVM$_O$	SVDD$_O$
NA	**0.0030***	**<.0001***	**<.0001***	**<.0001***	**<.0001***	0.3701	**<.0001***
HNA	**0.0392**	**<.0001***	**<.0001***	**<.0001***	**<.0001***	*0.2094*	**<.0001***
OSFM$_M$	*0.2700*	**0.0013***	**0.0133**	**<.0001***	0.5110	***0.0147***	**<.0001***
OSFM$_\mu$	0.3383	**0.0302**	**<.0001***	**<.0001***	**0.0060***	0.9524	**<.0001***
FM$_M$	1.0000	**0.0112**	**0.0112**	**<.0001***	1.0000	***0.0112***	**<.0001***
FM$_\mu$	**<.0001***	**<.0001***	**<.0001***	**<.0001***	**<.0001***	**<.0001***	*0.7651*
AKS	1.0000	**<.0001***	***<.0001****	***0.0283***	***<.0001****	***0.0019****	**<.0001***
AUS	**<.0001***	**<.0001***	**<.0001***	**<.0001***	**<.0001***	**<.0001***	***<.0001****

Table 6.6: Binomial statistical tests comparing the OSNN with best baselines. Each cell compares results for all datasets considering all number of available classes. **Bold** means there is statistical difference with 95% of confidence. "*" indicates the statistical difference is with 99% of confidence. And **<.0001*** indicates the statistical difference is with 99.99% of confidence. *Emphasized* indicates the method in the column obtains better performance for the measure associated with that row.

Measure	SVM_O	$OCSVM_O$	DBC_O	OVS_O	$WSVM_O$	$PISVM_O$	$SVDD_O$
NA	**<.0001***	**<.0001***	**<.0001***	**<.0001***	**<.0001***	**<.0001***	**<.0001***
HNA	**<.0001***	**<.0001***	**<.0001***	**<.0001***	**0.0001***	0.2561	**<.0001***
$OSFM_M$	**<.0001***	**<.0001***	**<.0001***	**<.0001***	**<.0001***	**0.0101**	**<.0001***
$OSFM_\mu$	**<.0001***	**<.0001***	**<.0001***	**<.0001***	**<.0001***	**<.0001***	**<.0001***
FM_M	**<.0001***	**<.0001***	**<.0001***	**<.0001***	**<.0001***	0.0829	**<.0001***
FM_μ	**<.0001***	**<.0001***	**<.0001***	**<.0001***	**<.0001***	**<.0001***	**0.0268**
AKS	*0.1658*	**<.0001***	***<.0001****	*0.1658*	***<.0001****	***<.0001****	**<.0001***
AUS	**<.0001***	**<.0001***	**<.0001***	**<.0001***	**<.0001***	**<.0001***	***<.0001****

Table 6.7: Binomial statistical tests comparing the $SSVM_O$ with best baselines. Each cell compares results for all datasets considering all number of available classes. **Bold** means there is statistical difference with 95% of confidence. "*" indicates the statistical difference is with 99% of confidence. And **<.0001*** indicates the statistical difference is with 99.99% of confidence. *Emphasized* indicates the method in the column obtains better performance for the measure associated with that row.

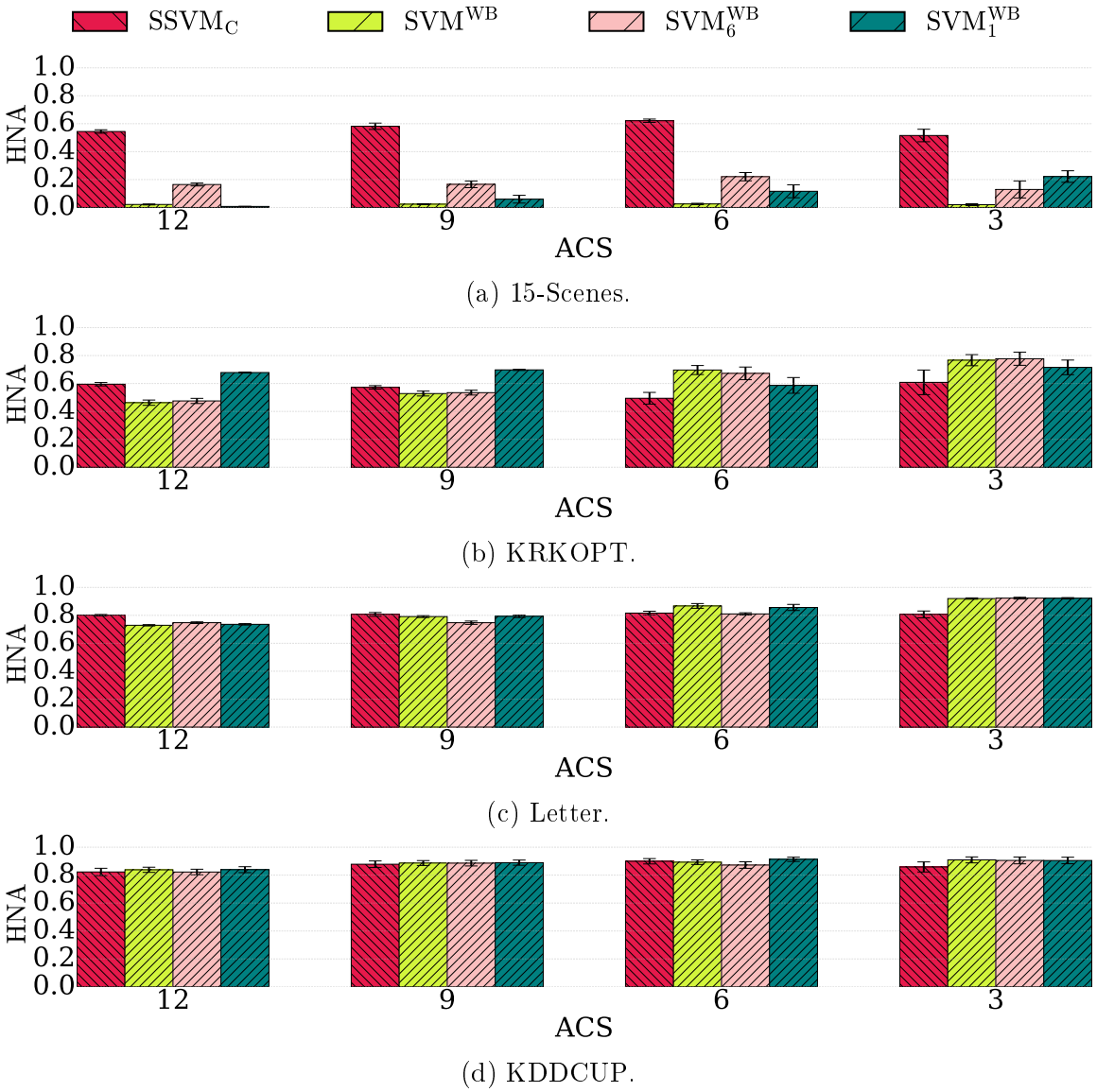
the sign function considers as positive values strictly greater than 0. However, notice that a theoretically correct way of bounding the KLOS for SVM^{WB} is to establish an *artificial bias term* $\epsilon < 0$ on prediction time, after SVM^{WB} model is obtained. This way, as soon as the decision value for a test instance approaches 0, it is considered to be on the open space.

For the purpose of those experiments, we introduce the artificial bias term ϵ on the decision function, as in Equation (5.13), aiming at a more restrictive KLOS. We have established $\epsilon = -1 \times 10^{-6}$ in SVM_6^{WB} and an even more restrict behavior, with $\epsilon = -1 \times 10^{-1}$, in SVM_1^{WB}. The experiments we have performed here consider closed-set grid search. In Figures 6.8 and 6.9, we present results for those alternatives—and also for SSVM as well, for comparison purposes. In general, we observe that SVM_1^{WB} improves the HNA over SVM^{WB} and slightly improves over SVM_6^{WB}. Statistical evaluation of this improvement is present in Table 6.8, where we check that, in fact, SVM_1^{WB} outperforms SVM^{WB} with statistical significance of 95% for most evaluation measures. However, for the global measures, there is no evidence that SVM_1^{WB} outperforms SVM_6^{WB}. Anyhow, by analyzing AKS and AUS, we confirm that SVM_1^{WB} is more restrict than SVM_6^{WB} on accepting unknown instances.

Those experiments show that by decreasing the KLOS, we improve accuracy on open-set scenarios.

6.7 Assessing the importance of bounding the known-labeled open space

The rationale of SSVM—and OSNN as well—is to bound the KLOS for better performance on open-set scenarios. Besides bounding the KLOS, the regularization parameter λ of SSVM also minimizes the risk of the unknown. This double-factor difference over traditional SVM does not allow us to claim that being able to bound the KLOS is "essential"



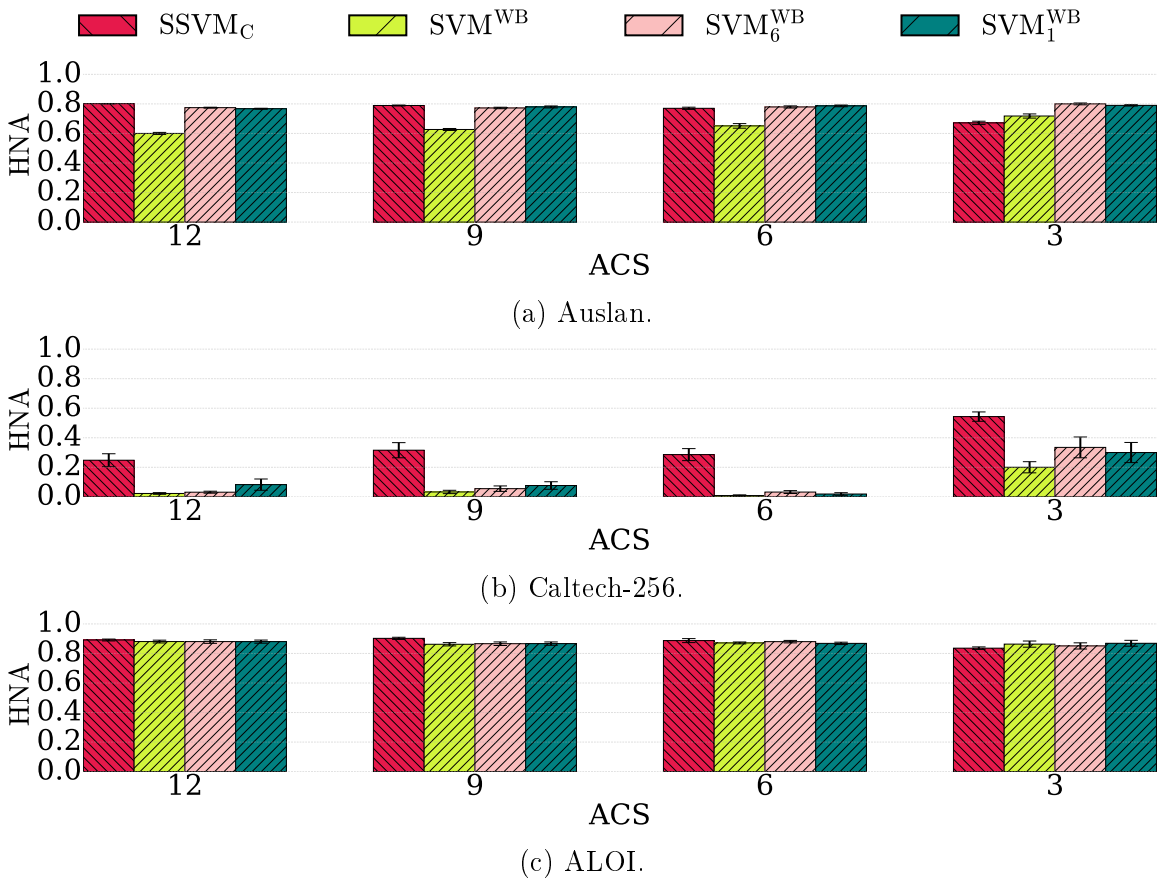
(a) 15-Scenes.

(b) KRKOPT.

(c) Letter.

(d) KDDCUP.

Figure 6.8: Performance of SVM without bias term (part I). Results for 15-Scenes, KRKOPT, Letter, and KDDCUP datasets regarding HNA considering 3, 6, 9, and 12 available classes (ACS).



(a) Auslan.

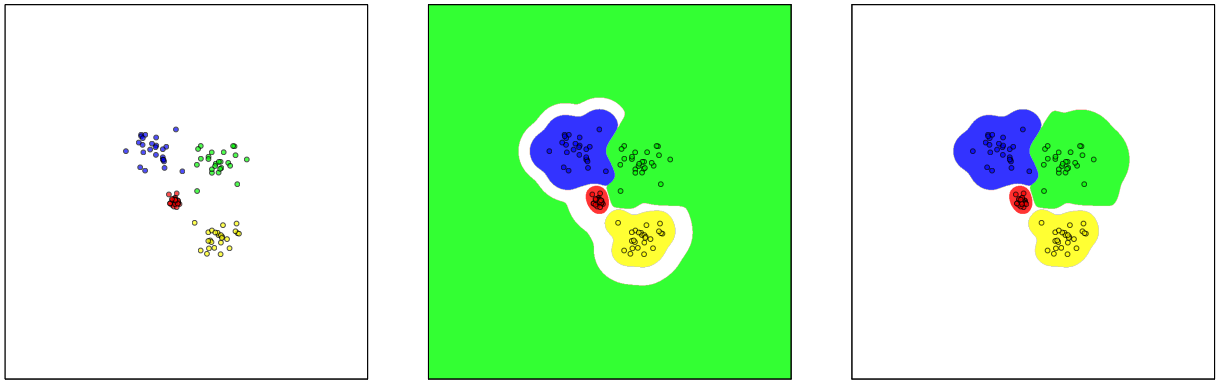
(b) Caltech-256.

(c) ALOI.

Figure 6.9: Performance of SVM without bias term (part II). Results for Auslan, Caltech-256, and ALOI datasets regarding HNA considering 3, 6, 9, and 12 available classes (ACS).

Measure	SVM^{WB}	SVM$_6^{WB}$
NA	0.1275	*0.4373*
HNA	**<.0001***	0.7651
OSFM$_M$	**0.0003***	0.7651
OSFM$_\mu$	**<.0001***	0.1350
FM$_M$	**0.0001***	0.8578
FM$_\mu$	**0.0392**	0.4373
AKS	*<.0001**	*0.0010**
AUS	**0.0196**	**0.0020***

Table 6.8: Binomial statistical tests comparing the SVM$_1^{WB}$ with SVM without bias term alternatives. Each cell compares results for all datasets considering all number of available classes. **Bold** means there is statistical difference with 95% of confidence. "*" indicates the statistical difference is with 99% of confidence. And **<.0001*** indicates the statistical difference is with 99.99% of confidence. *Emphasized* indicates the method in the column obtains better performance for the measure associated with that row.



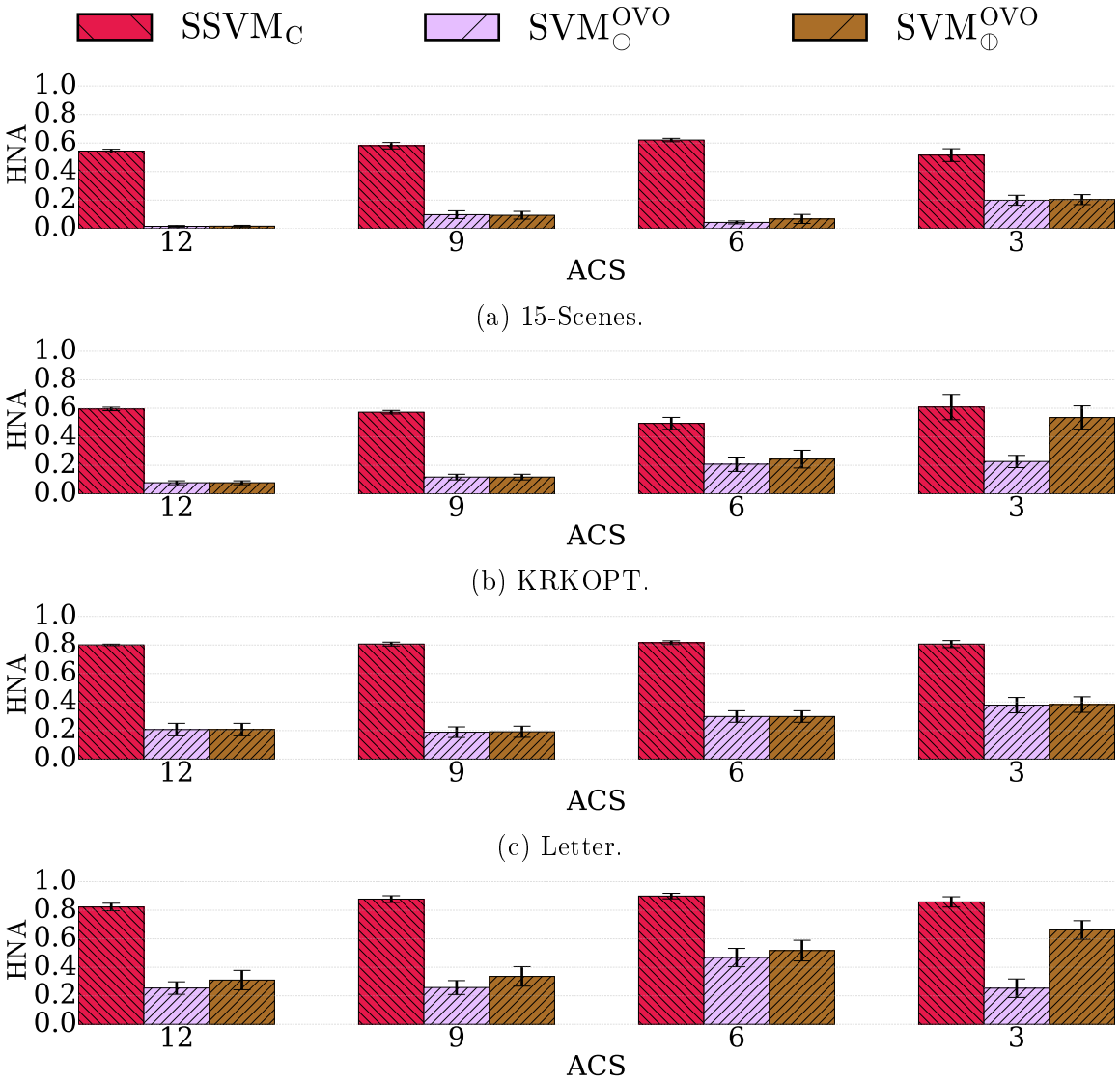
Four-Gauss dataset. (a) $\text{SVM}_{\ominus}^{\text{OVO}}$. (b) $\text{SVM}_{\oplus}^{\text{OVO}}$.

Figure 6.10: Behavior analysis of SVM^{OVO} with minimal threshold. The Four-Gauss dataset is depicted on the far left. Figure (a) depicts the behavior of the classifier when the threshold for rejection is $T - 1 \times 10^{-6}$, i.e., *below* a minimum required threshold. Figure (b) depicts the behavior of the classifier when the threshold for rejection is $T + 1 \times 10^{-6}$, i.e., *above* a minimum required threshold. T is obtained according to Equation (5.18). Closed-set grid search was employed on generating those figures. The small circles represent the training samples from the dataset and their colors represent their classes. A colored background represents the class in which a possible test instance in the corresponding position of the feature space would be classified. White background indicates that a possible test instance in that region would be classified as unknown.

for open-set recognition, as one factor can influence the other. However, the Support Vector Machines with the one-vs-one approach (SVM^{OVO}) presented in Section 5.3.2, along with its minimum required threshold T for bounding the KLOS, allows us to analyze this factor individually. As shown in that section, the multiclass probability estimate for the open space approaches T, as defined in Equation (5.18). Consequently, the KLOS of SVM^{OVO} is bounded if and only if the threshold for rejection is greater than T. Aiming at showing the importance of bounding the KLOS, we have established $\text{SVM}_{\ominus}^{\text{OVO}}$ and $\text{SVM}_{\oplus}^{\text{OVO}}$ implementations. $\text{SVM}_{\ominus}^{\text{OVO}}$ establishes its rejection threshold as $T - 1 \times 10^{-6}$ and $\text{SVM}_{\oplus}^{\text{OVO}}$ establishes its rejection threshold as $T + 1 \times 10^{-6}$. The former is *unable* to bound the KLOS and the latter is *able* to, as we can see in Figure 6.10.

In Figure 6.10, each implementation has performed its own grid search, which would potentially allow them to obtain distinct fitting parameters. However, notice that the difference on thresholds between $\text{SVM}_{\ominus}^{\text{OVO}}$ and $\text{SVM}_{\oplus}^{\text{OVO}}$ is small. And, as SVM models for both implementations are obtained based on known samples, those are potentially similar, which we can confirm by analyzing the decision boundaries among known classes in Figures 6.10a and 6.10b. For those reasons, SVM^{OVO} is appropriate for verifying our hypothesis on the requirement of bounding the KLOS.

In Figures 6.11 and 6.12, we present HNA results for $\text{SVM}_{\ominus}^{\text{OVO}}$ and $\text{SVM}_{\oplus}^{\text{OVO}}$—as well as for SSVM, for comparison purposes. We can observe that in virtually all cases, $\text{SVM}_{\oplus}^{\text{OVO}}$ outperforms $\text{SVM}_{\ominus}^{\text{OVO}}$, which confirms our hypothesis. Both Binomial and Wilcoxon statistical tests present more than 99% of confidence on those results for global measures and for AUS. However, AKS for $\text{SVM}_{\ominus}^{\text{OVO}}$ is better than for $\text{SVM}_{\oplus}^{\text{OVO}}$, also with more than 99% confidence.



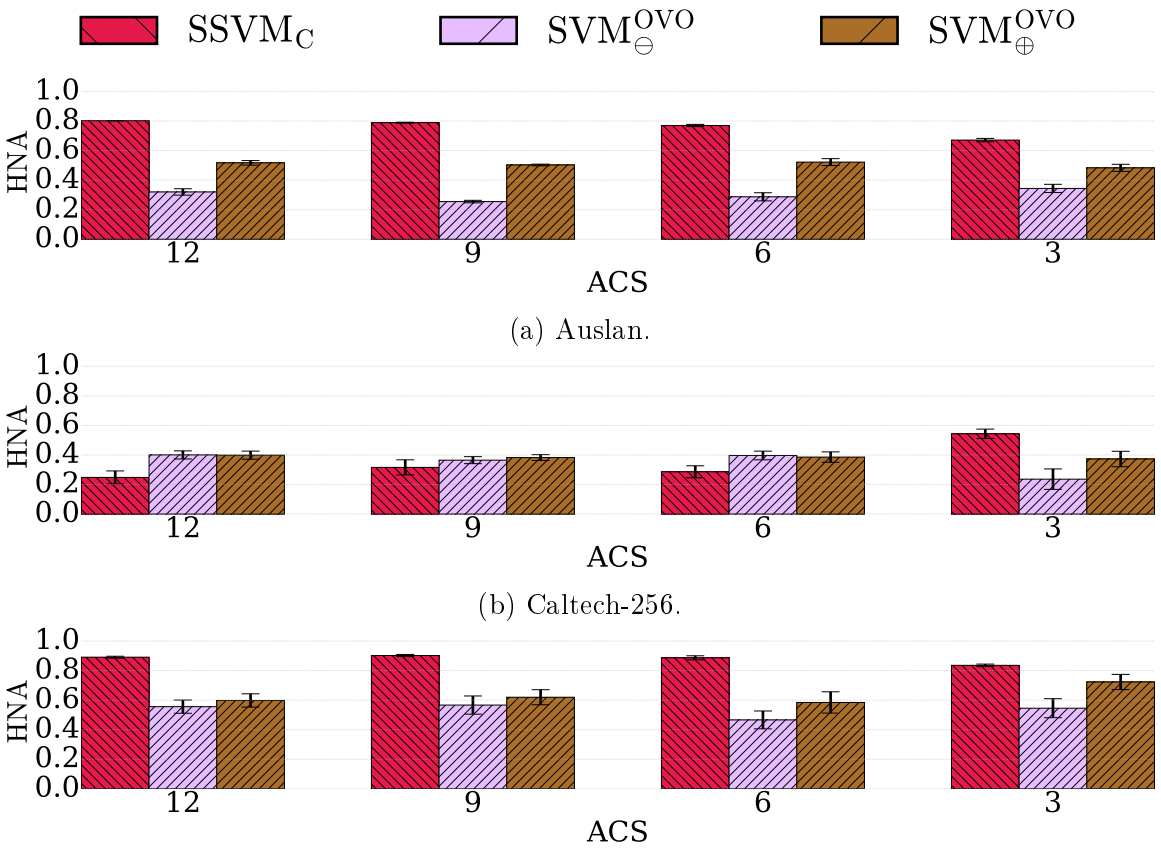
(a) 15-Scenes.

(b) KRKOPT.

(c) Letter.

(d) KDDCUP.

Figure 6.11: Comparison of SVM with unbounded/bounded KLOS (part I). Results for 15-Scenes, KRKOPT, Letter, and KDDCUP datasets regarding HNA considering 3, 6, 9, and 12 available classes (ACS).



(a) Auslan.

(b) Caltech-256.

(c) ALOI.

Figure 6.12: Comparison of SVM with unbounded/bounded KLOS (part II). Results for Auslan, Caltech-256, and ALOI datasets regarding HNA considering 3, 6, 9, and 12 available classes (ACS).

Measure	$\text{OSNN}_{10}^{\lambda_r}$	$\text{OSNN}_{30}^{\lambda_r}$	$\text{OSNN}_{70}^{\lambda_r}$	$\text{OSNN}_{90}^{\lambda_r}$
NA	1.0000	**<.0001***	1.0000	**<.0001***
HNA	**0.0020***	0.8578	*<.0001**	**<.0001***
OSFM_M	**0.0005***	**<.0001***	**0.0001***	**0.0196**
OSFM_μ	**<.0001***	**<.0001***	**0.0141**	**0.0001***
FM_M	**0.0001***	**<.0001***	*<.0001**	0.2094
FM_μ	*<.0001**	*<.0001**	**<.0001***	**<.0001***
AKS	**<.0001***	**0.0268**	*<.0001**	*<.0001**
AUS	*<.0001**	*<.0001**	**<.0001***	**<.0001***

Table 6.9: Binomial statistical tests comparing the OSNN with OSNN alternatives. Each cell compares results for all datasets considering all number of available classes. **Bold** means there is statistical difference with 95% of confidence. "*" indicates the statistical difference is with 99% of confidence. And **<.0001*** indicates the statistical difference is with 99.99% of confidence. *Emphasized* indicates the method in the column obtains better performance for the measure associated with that row.

6.8 Variable metric on grid search

In this section, we analyze the influence of the regularization constant λ_r of NA—see Equation (2.1)—when employing that measure during grid search. For this purpose, we have established the following extra implementations of OSNN: $\text{OSNN}_{10}^{\lambda_r}$, $\text{OSNN}_{30}^{\lambda_r}$, $\text{OSNN}_{70}^{\lambda_r}$, and $\text{OSNN}_{90}^{\lambda_r}$. Respectively, they perform grid search based on NA with λ_r set to 0.1, 0.3, 0.7, and 0.9. In Table 6.9, we check if there is statistical significance for each evaluation measure when compared to OSNN, which uses $\lambda_r = 0.5$. As most measures check for overall performance, including f-measure alternatives, we observe that OSNN with $\lambda_r = 0.5$ still performs the best in general. AKS and AUS in Table 6.9 present the most important information in this analysis, however. With more than 95% confidence, we have the following observations. Regarding AKS, OSNN outperforms $\text{OSNN}_{10}^{\lambda_r}$ and $\text{OSNN}_{30}^{\lambda_r}$ but not $\text{OSNN}_{70}^{\lambda_r}$ and $\text{OSNN}_{90}^{\lambda_r}$. Regarding AUS, OSNN outperforms $\text{OSNN}_{70}^{\lambda_r}$ and $\text{OSNN}_{90}^{\lambda_r}$ but not $\text{OSNN}_{10}^{\lambda_r}$ and $\text{OSNN}_{30}^{\lambda_r}$. It indicates that, in fact, NA with certain values for λ_r can be employed during grid search to make the recognition methods more or less restrictive on accepting false unknown rates. Furthermore, a trade off is unavoidable in this case.

6.9 Performance with deep features

In Chapter 7, we will formalize neural networks and present analyses regarding their behavior in open-set scenarios. Beforehand, we have decided to include results with deep features in this chapter, as neural networks were employed solely for feature extraction, hence, the handling of the open-set problem at network's level is not analyzed here. We consider, for the experiments with deep features, the SVM-based classifiers previously employed.

In this analysis, we have extracted the features in two distinct setups: (1) Open-set

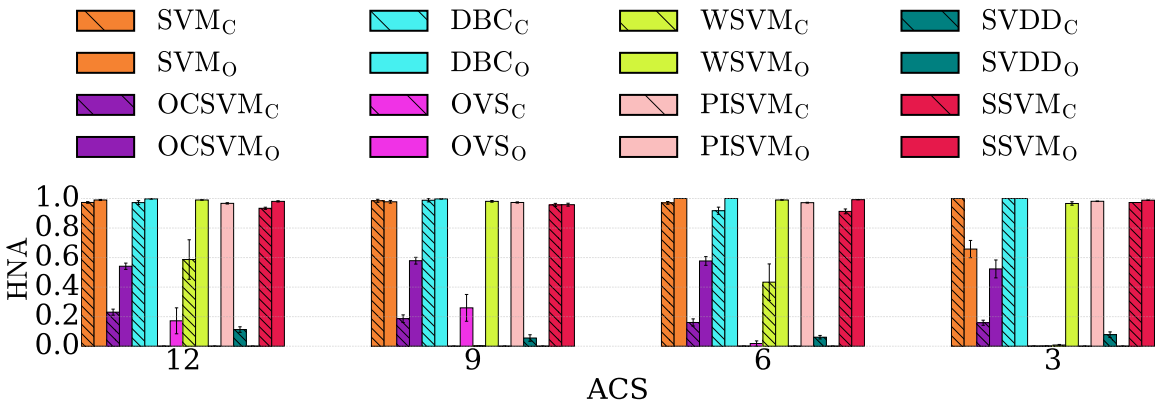
Measure	SVM_O	$OCSVM_O$	DBC_O	OVS_O	$WSVM_O$	$PISVM_O$	$SVDD_O$
NA	*0.0067**	**<.0001***	*<.0001**	**<.0001***	0.8746	**0.0067***	**<.0001***
HNA	*0.0067**	**<.0001***	*<.0001**	**<.0001***	0.8746	**0.0067***	**<.0001***
$OSFM_M$	*0.0129*	**<.0001***	*<.0001**	**<.0001***	0.8746	**0.0067***	**<.0001***
$OSFM_\mu$	*0.0193*	**<.0001***	*<.0001**	**<.0001***	*0.8746*	0.8592	**<.0001***
FM_M	*0.0129*	**<.0001***	*<.0001**	**<.0001***	0.8746	**0.0067***	**<.0001***
FM_μ	*0.0257*	**<.0001***	*<.0001**	**<.0001***	*0.8746*	0.8592	0.1154
AKS	*<.0001**	**<.0001***	*<.0001**	**<.0001***	**<.0001***	**<.0001***	**<.0001***
AUS	*0.0129*	**0.0005***	*<.0001**	**<.0001***	*<.0001**	*0.0166*	*<.0001**

Table 6.10: Binomial statistical tests comparing the $SSVM_O$ with baselines in ImageNet. **Bold** means there is statistical difference with 95% of confidence. "*" indicates the statistical difference is with 99% of confidence. And **<.0001*** indicates the statistical difference is with 99.99% of confidence. *Emphasized* indicates the method in the column obtains better performance for the measure associated with that row.

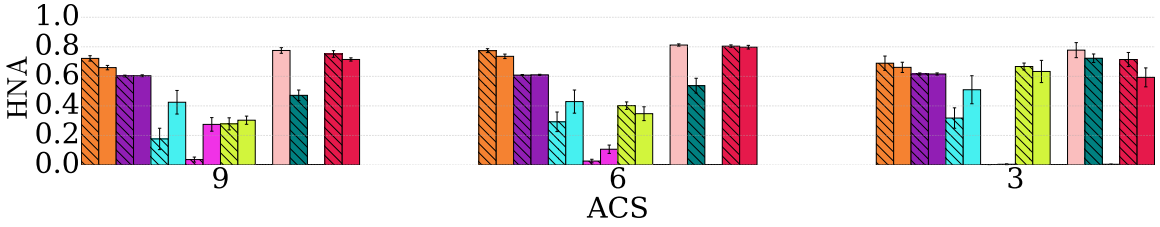
network: the network used for feature extraction is not trained with the classes employed in the open-set experiment and (2) Closed-set network: the network used for feature extraction is trained on all classes employed in the experiment.

We have employed ImageNet [Deng et al., 2009] dataset along with *open-set network* setup. For this, a Convolutional Neural Network (CNN) was trained on ImageNet 2012 dataset, which contains 1000 classes. Then, experiments where performed on a subset of ImageNet 2010 with 360 classes, which has been reported in Bendale and Boult [2016] and Russakovsky et al. [2015] to have no overlapping with the classes of version 2012 of the dataset. In Figure 6.13a, we present results for ImageNet for methods with both closed- and open-set grid search. We see that SVM and open-set methods DBC, WSVM, PISVM, and SSVM have achieved near 100% accuracy, which means that both AKS and AUS for those methods are also near 100%. Aiming at verifying if there are statistical differences among those methods, we have selected only the versions with open-set grid search. Binomial statistical tests for ImageNet are presented in Table 6.10, in which we can see that both SVM and DBC have outperformed SSVM for this dataset. It is interesting to notice that DBC and SVM have not been competing methods to SSVM in previous experiments, however, in this case, they have presented outstanding performance.

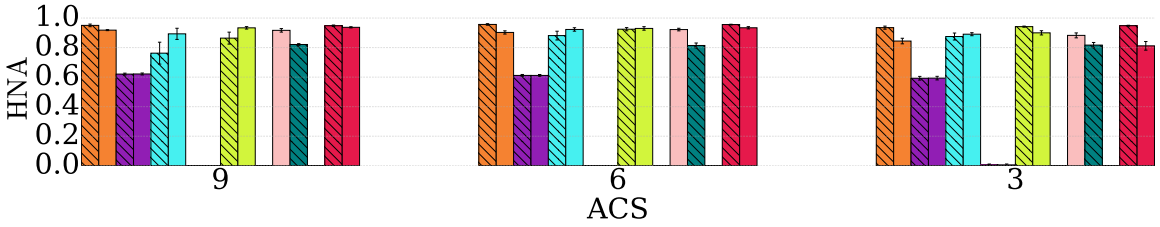
We have also experimented with CIFAR-10 [Krizhevsky and Hinton, 2009] and MNIST [LeCun et al., 1998] datasets along with the *closed-set network* setup. Networks employed for both datasets are publicly available [Tensorflow.org, 2018a,b]. Both datasets comprise 10-class problems. CIFAR-10 represents an object classification problem with classes of vehicles and animals. MNIST is a digit classification problem whose classes are 0–9 digits. In Figures 6.13b and 6.13c, we present results for CIFAR-10 and MNIST datasets, respectively. For CIFAR-10, in Figure 6.13b, SSVM, PISVM, and SVM seem to have performed best, with a highlight for PISVM. In fact, in Table 6.11, we confirm the superiority of PISVM with Binomial tests. In this case, however, SSVM has outperformed the baselines SVM and DBC of the previous experiment, for the global measures. As for the MNIST, WSVM has excelled, as indicated with statistical significance for FM_M and FM_μ. DBC and PISVM seem to have slightly outperformed SSVM, however, no statis-



(a) ImageNet 2010.



(b) CIFAR-10.



(c) MNIST.

Figure 6.13: Comparison among best methods with deep features. Results for ImageNet 2010, CIFAR-10, and MNIST datasets. Experiments regarding HNA considering 3, 6, 9, and 12 available classes (ACS) for ImageNet 2010 and 3, 6, and 9 ACS for CIFAR-10 and MNIST datasets.

Measure	SVM_O	$OCSVM_O$	DBC_O	OVS_O	$WSVM_O$	$PISVM_O$	$SVDD_O$
NA	**0.0072***	**0.0161**	**0.0072***	**<.0001***	**0.0072***	***0.0104***	**<.0001***
HNA	**0.0104**	**0.0016***	**0.0104**	**<.0001***	**0.0057***	***0.0057****	**<.0001***
$OSFM_M$	**0.0209**	**0.0086***	**0.0209**	1.0000	0.1975	***0.0086****	**<.0001***
$OSFM_\mu$	**0.0209**	**0.0019***	**0.0209**	0.5847	0.1975	***0.0019****	**<.0001***
FM_M	**0.0157**	**0.0072***	**0.0072***	0.3616	0.1975	***0.0019****	**<.0001***
FM_μ	**<.0001***	**0.0057***	**0.0003***	0.0987	**0.0104**	***0.0057****	**<.0001***
AKS	1.0000	**0.0016***	1.0000	***<.0001****	*1.0000*	*0.0645*	**<.0001***
AUS	**<.0001***	*0.1283*	**<.0001***	**<.0001***	0.1975	0.2005	***<.0001****

Table 6.11: Binomial statistical tests comparing the $SSVM_O$ with baselines in CIFAR-10. **Bold** means there is statistical difference with 95% of confidence. "*" indicates the statistical difference is with 99% of confidence. And **<.0001*** indicates the statistical difference is with 99.99% of confidence. *Emphasized* indicates the method in the column obtains better performance for the measure associated with that row.

Measure	SVM_O	$OCSVM_O$	DBC_O	OVS_O	$WSVM_O$	$PISVM_O$	$SVDD_O$
NA	0.1711	**<.0001***	*1.0000*	**<.0001***	*1.0000*	*1.0000*	**<.0001***
HNA	0.3949	**<.0001***	*1.0000*	**<.0001***	*1.0000*	*1.0000*	**<.0001***
$OSFM_M$	0.1283	**<.0001***	*0.1975*	**<.0001***	*0.0645*	*0.1975*	**<.0001***
$OSFM_\mu$	0.0645	**<.0001***	*0.7232*	**<.0001***	*0.0645*	*0.8555*	**<.0001***
FM_M	**0.0484**	**<.0001***	*1.0000*	**<.0001***	***0.0209***	*1.0000*	**<.0001***
FM_μ	**0.0057***	**<.0001***	*1.0000*	**<.0001***	***0.0157***	*1.0000*	**<.0001***
AKS	*0.8555*	**<.0001***	***<.0001****	***0.0010****	***0.0002****	***0.0104***	**<.0001***
AUS	**0.0029***	***<.0001****	**0.0010***	**<.0001***	**0.0002***	0.0987	***<.0001****

Table 6.12: Binomial statistical tests comparing the $SSVM_O$ with baselines in MNIST. **Bold** means there is statistical difference with 95% of confidence. "*" indicates the statistical difference is with 99% of confidence. And **<.0001*** indicates the statistical difference is with 99.99% of confidence. *Emphasized* indicates the method in the column obtains better performance for the measure associated with that row.

tical significance is evinced in this case. It is worth noticing that for those two datasets, SSVM has performed better with its variant performing closed-set grid search.

Those results with deep features have shown us that depending on the problem, one classifier can be more suitable than the other. Besides not maintaining the best performance in all cases, SSVM have shown a robust behavior, as it outperforms each baseline for at least one of the datasets.

6.10 Behavior analysis of the classifiers

Aiming at obtaining an intuition of the proposed classifiers, as well as of the baselines employed in this work, in this section, we present a behavior analysis of those methods. We employ 2-dimensional synthetic datasets for training the classifiers, then we use the generated models to predict their behavior in the feature space. To generate the images

depicting the behavior of the classifiers, test samples comprise points in grid on the 2-dimensional space.

We have employed the following 2-dimensional synthetic datasets: Boat, Four-Gauss, Petals, Regular, R15, Seven-Gauss, Half-Ring, and Cone-Torus.[4] Respectively, they are depicted in Figures 6.14–6.21. In those figures, the small circles represent training samples from the dataset. Their colors represent their classes. The background color represents the class in which a possible test instance in that position of the feature space would be classified. White background indicates that a possible test instance in that region would be classified as unknown. We can see that a method is able to bound the KLOS when the colored region is bounded.

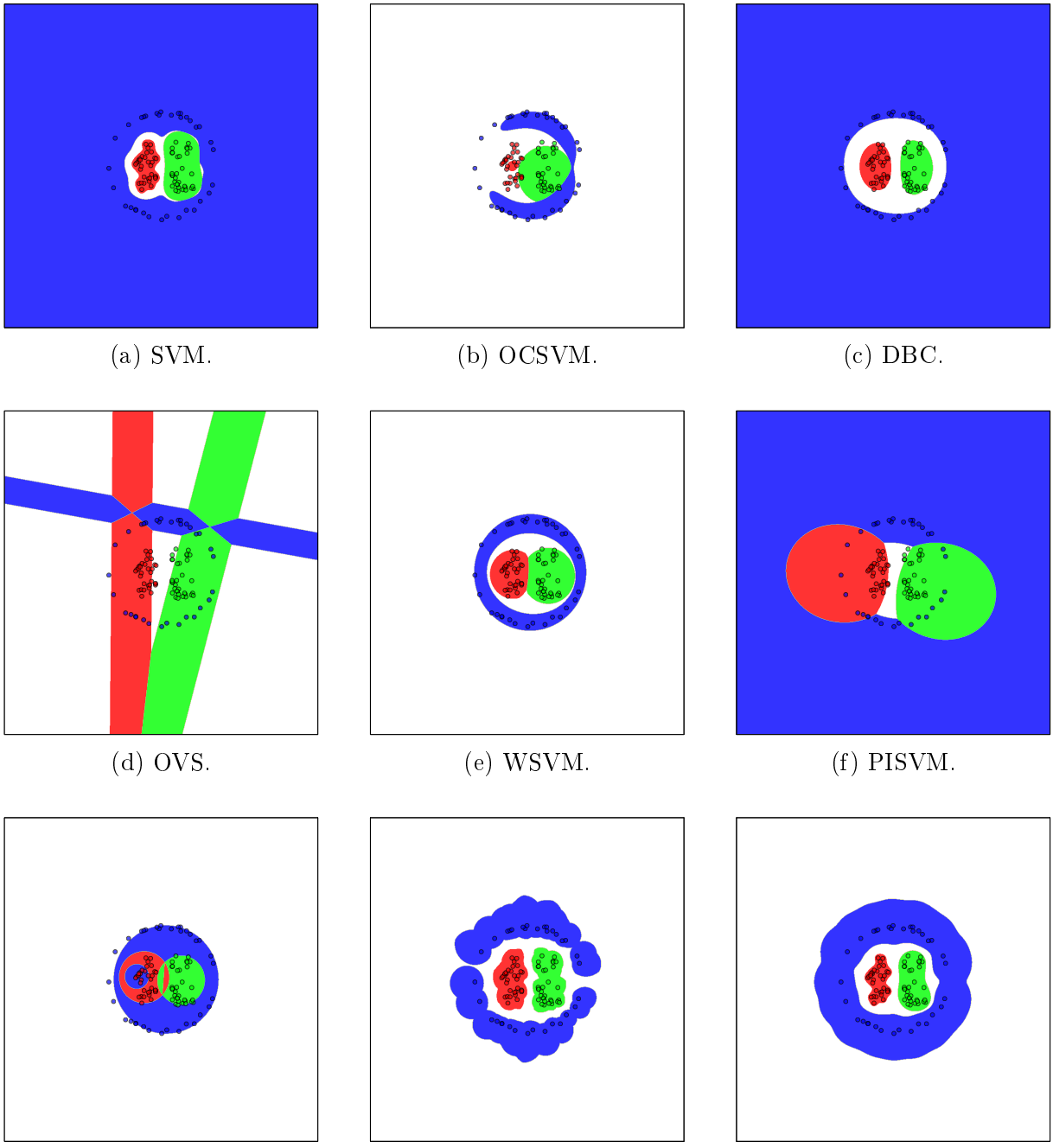
Subfigures labeled (a) show that SVM is able to bound the KLOS however not in all cases, as can be seen in Figures 6.14a, 6.20a, and 6.21a. Compared to SVM, DBC performs a finer adjustment of the decision hyperplane, aiming at a more specialized behavior, however, it does not ensure a bounded KLOS, as evinced in Figures 6.14c and 6.21a. Anyhow, when the translation of the hyperplane performed by DBC obtains a negative bias term for every binary classifier, DBC is able to bound the KLOS, as can be seen in Figure 6.20c compared to SVM in Figure 6.20a. For the one-class classifiers OCSVM and SVDD, in Subfigures labeled (b) and (g), respectively, we observe that, in fact, they are able to always obtain a bounded KLOS at the expense of a highly-specialized behavior. Due to its linear kernel, OVS is never able to bound the KLOS, as observed in Subfigures labeled (d). In those figures, we clearly observe the slabs this method creates aiming at decreasing the KLOS. WSVM employs one-class models in its formulation and that is the reason it can bound the KLOS, as depicted in Subfigures labeled (e)[5], however, its behavior is not as specialized as the behavior of other one-class models. It is well depicted in Figure 6.14e how the binary model employed by WSVM creates a good separation among the known classes and avoids the influence of the one-class models for the separation. For PISVM in Subfigures labeled (f), we also observe that it is not always able to bound the KLOS, as in Figures 6.14f and 6.20f. OSNN and SSVM are always able to bound the risk of the unknown, as seen in Subfigures labeled (h) and (i). In general, SSVM presents a more specialized behavior than OSNN and gracefully bounds the KLOS. SSVM also avoids the extra KLOS obtained by OSNN in some cases. Finally, we observe that SSVM does not suffer from the problem of rejecting doubtful test samples that might appear in the overlapping region of two or more classes, as OSNN does. Clearly, we see that in Cone-Torus dataset by comparing Figures 6.21h and 6.21i.

Although in high-dimensional spaces the behavior of the classifiers can differ, with those images, we obtain an intuition of what to expect from each classifier. For instance, consider the PISVM: it is not always the case PISVM is able to bound the KLOS—as there is no mechanism to ensure that—although it happens in most situations. By contrast,

[4] R15 dataset was made available by Veenman et al. [2002], Seven-Gauss datasets was generated by us, and all other synthetic datasets are from Kuncheva and Hadjitodorov [2004].

[5] In Figure 6.20e, WSVM is not presenting a bounded-KLOS behavior due to a bug found in the source code provided by Scheirer et al. [2014]: when only two classes are available, instead of obtaining one model per class, the method is fitting a single model, which might leave an unbounded KLOS for one of the classes. However, it does not affect the experiments we have presented in previous sections as we consider at least 3 available classes on the open-set setup.

SSVM does that by considering the correct signal for the bias term. With those images, we gain a better view that, in a critical open-set application, one should seriously consider the implications of employing a classifier with no guaranty of limited open-space risk.

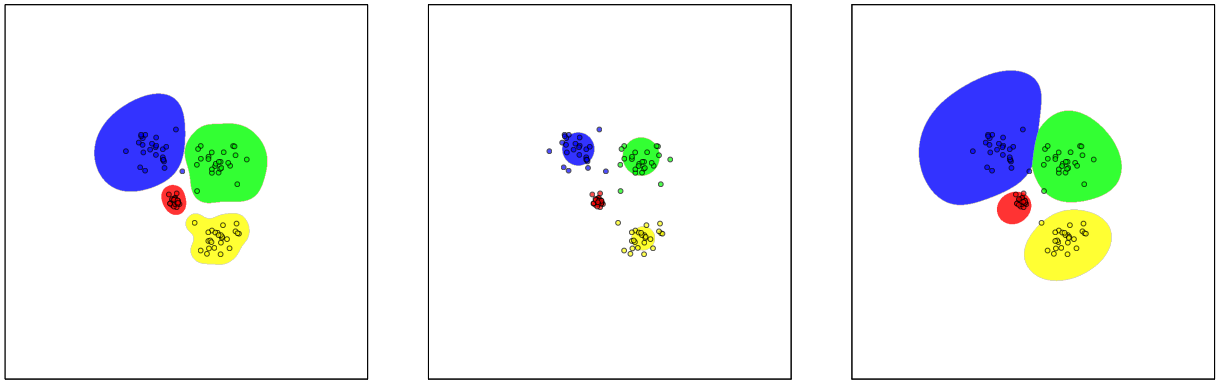


(a) SVM. (b) OCSVM. (c) DBC.

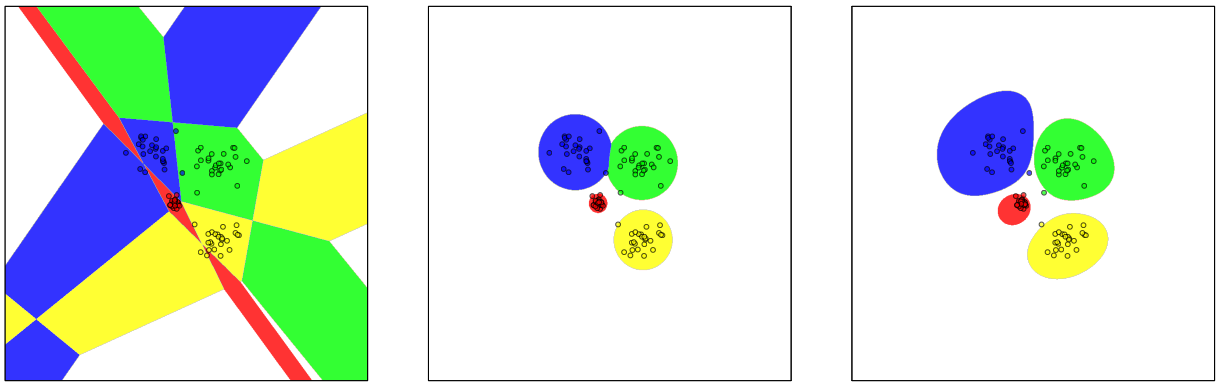
(d) OVS. (e) WSVM. (f) PISVM.

(g) SVDD. (h) OSNN. (i) SSVM.

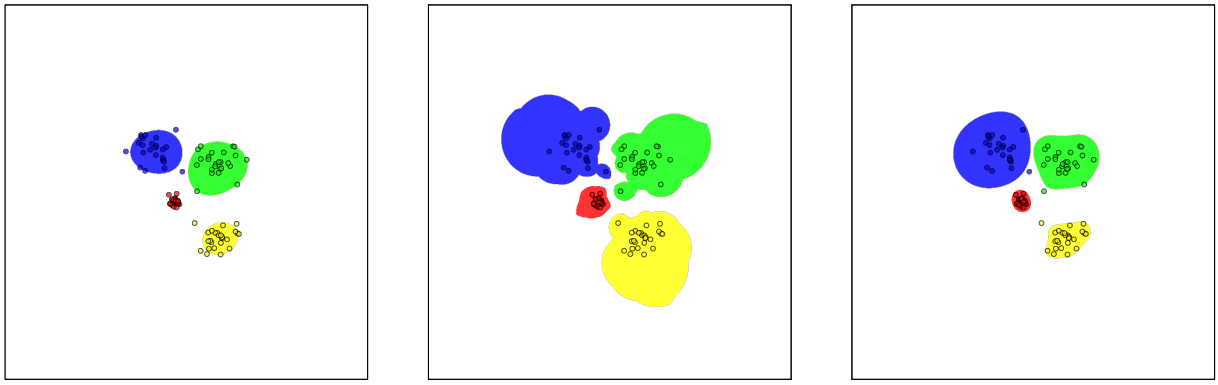
Figure 6.14: Decision boundaries on the Boat dataset. Behavior of the classifiers considered for the experiments. The small circles represent the training samples from the dataset and their colors represent their classes. A colored background represents the class in which a possible test instance in the corresponding position of the feature space would be classified. White background indicates that a possible test instance in that region would be classified as unknown.



(a) SVM. (b) OCSVM. (c) DBC.

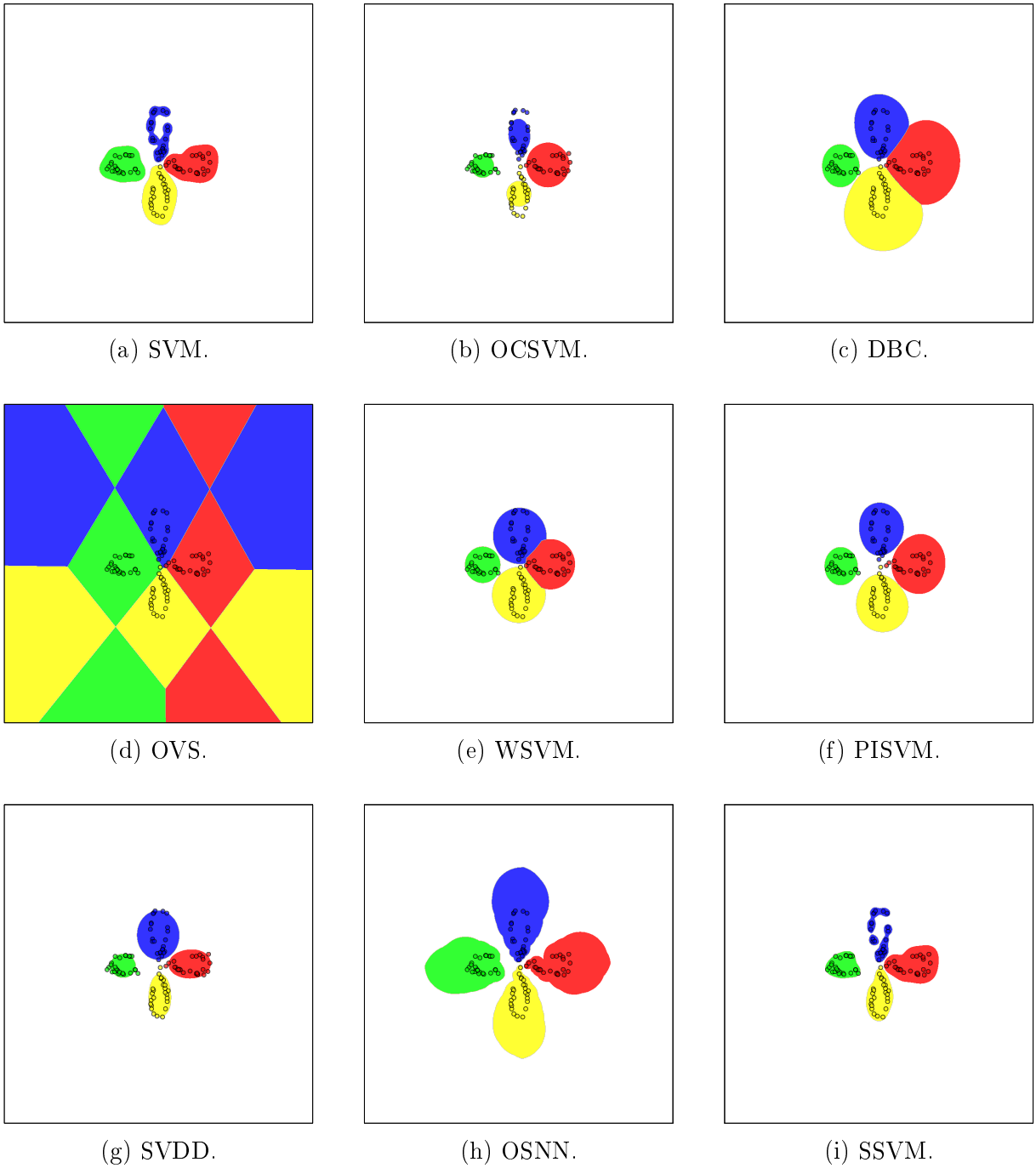


(d) OVS. (e) WSVM. (f) PISVM.



(g) SVDD. (h) OSNN. (i) SSVM.

Figure 6.15: Decision boundaries on the Four-Gauss dataset. Behavior of the classifiers considered for the experiments. The small circles represent the training samples from the dataset and their colors represent their classes. A colored background represents the class in which a possible test instance in the corresponding position of the feature space would be classified. White background indicates that a possible test instance in that region would be classified as unknown.

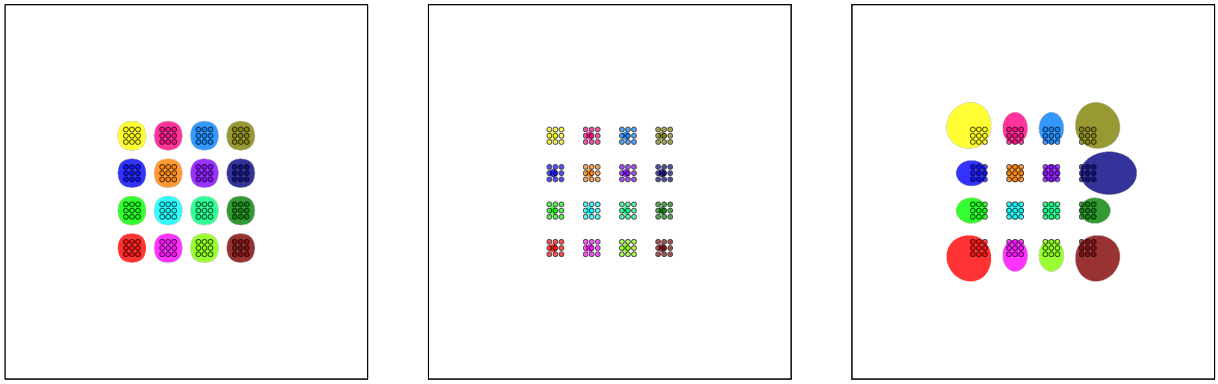


(a) SVM. (b) OCSVM. (c) DBC.

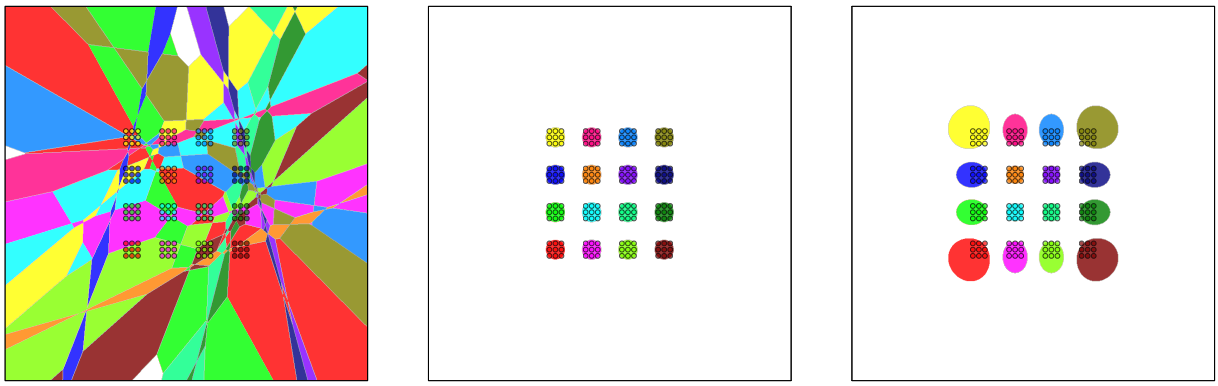
(d) OVS. (e) WSVM. (f) PISVM.

(g) SVDD. (h) OSNN. (i) SSVM.

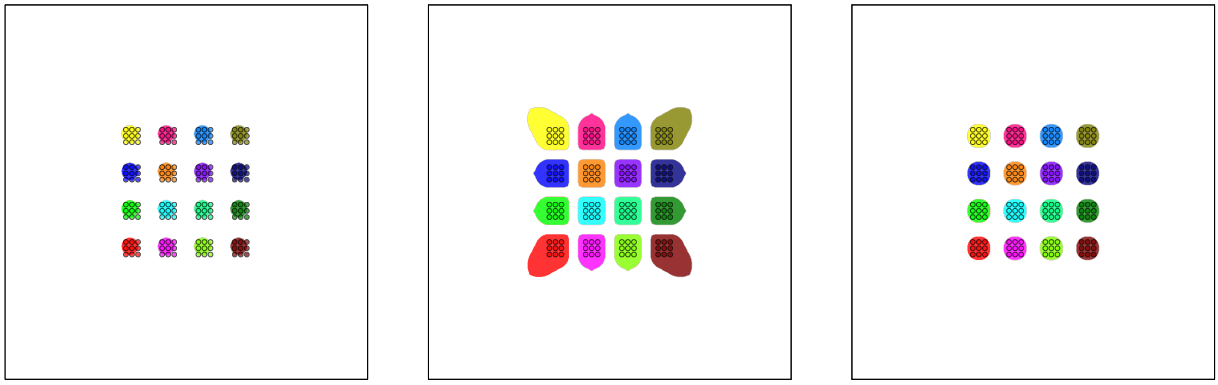
Figure 6.16: Decision boundaries on the Petals dataset. Behavior of the classifiers considered for the experiments. The small circles represent the training samples from the dataset and their colors represent their classes. A colored background represents the class in which a possible test instance in the corresponding position of the feature space would be classified. White background indicates that a possible test instance in that region would be classified as unknown.



(a) SVM. (b) OCSVM. (c) DBC.

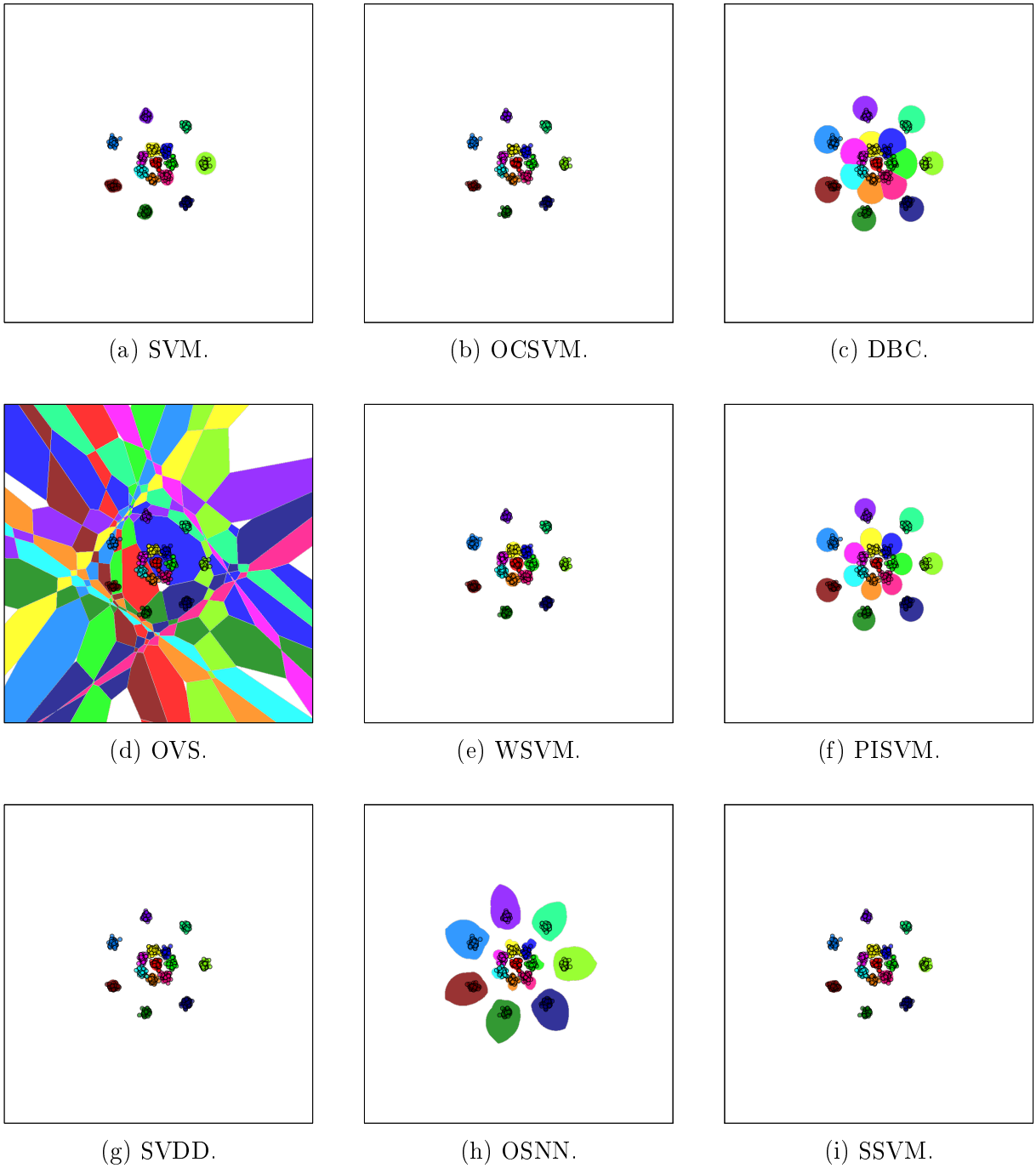


(d) OVS. (e) WSVM. (f) PISVM.



(g) SVDD. (h) OSNN. (i) SSVM.

Figure 6.17: Decision boundaries on the Regular dataset. Behavior of the classifiers considered for the experiments. The small circles represent the training samples from the dataset and their colors represent their classes. A colored background represents the class in which a possible test instance in the corresponding position of the feature space would be classified. White background indicates that a possible test instance in that region would be classified as unknown.

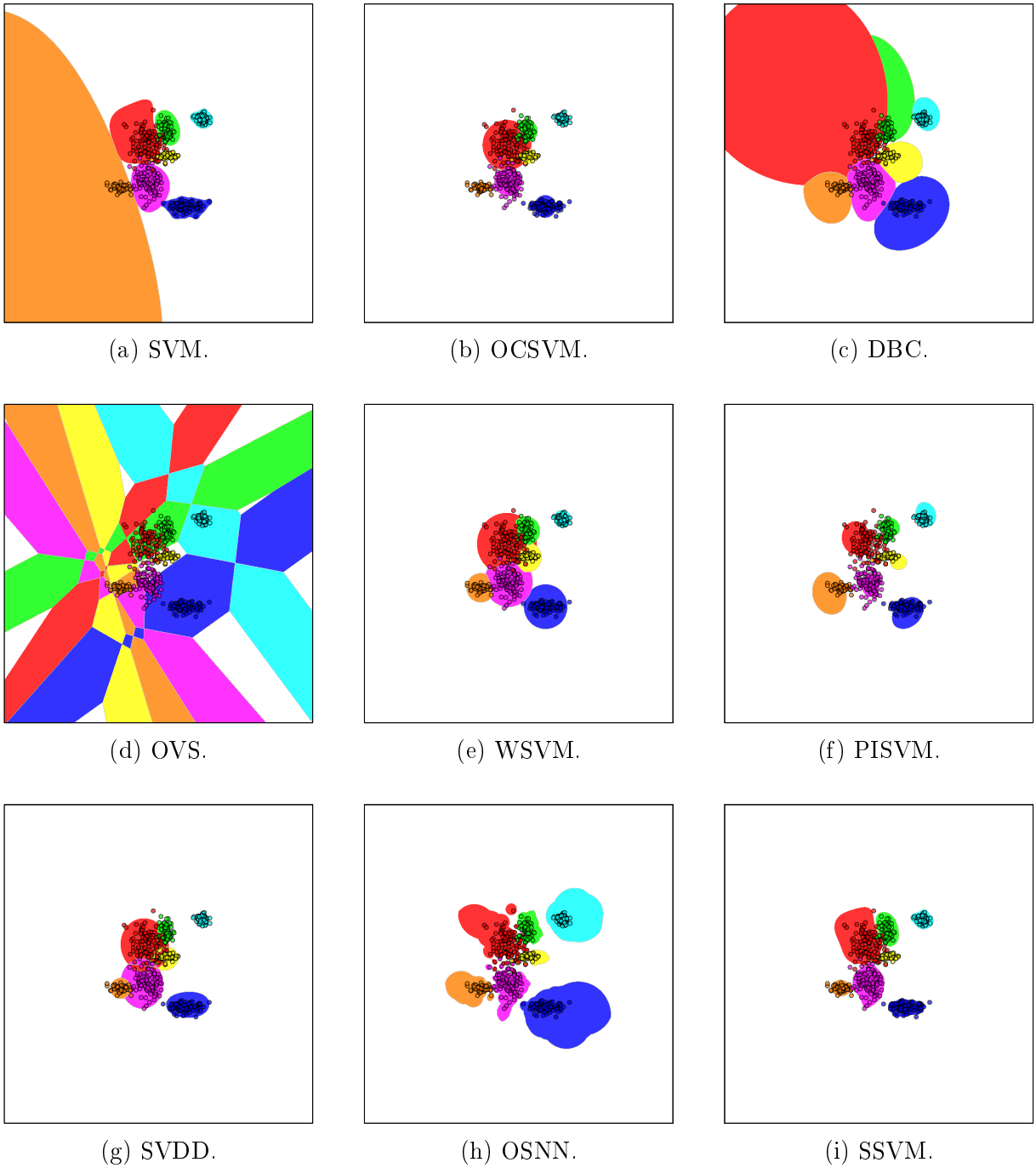


(a) SVM. (b) OCSVM. (c) DBC.

(d) OVS. (e) WSVM. (f) PISVM.

(g) SVDD. (h) OSNN. (i) SSVM.

Figure 6.18: Decision boundaries on the R15 dataset. Behavior of the classifiers considered for the experiments. The small circles represent the training samples from the dataset and their colors represent their classes. A colored background represents the class in which a possible test instance in the corresponding position of the feature space would be classified. White background indicates that a possible test instance in that region would be classified as unknown.

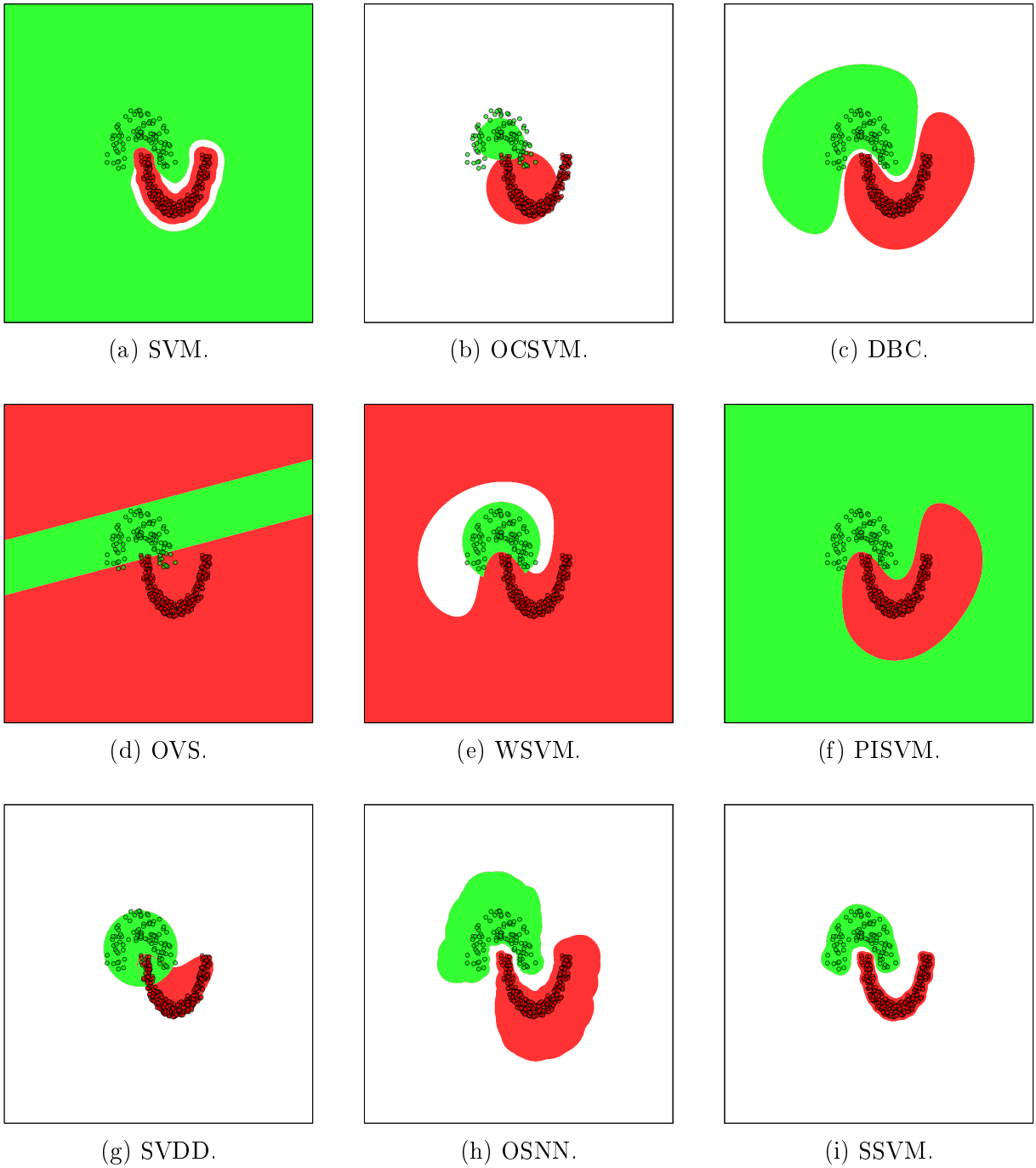


(a) SVM. (b) OCSVM. (c) DBC.

(d) OVS. (e) WSVM. (f) PISVM.

(g) SVDD. (h) OSNN. (i) SSVM.

Figure 6.19: Decision boundaries on the Seven-Gauss dataset. Behavior of the classifiers considered for the experiments. The small circles represent the training samples from the dataset and their colors represent their classes. A colored background represents the class in which a possible test instance in the corresponding position of the feature space would be classified. White background indicates that a possible test instance in that region would be classified as unknown.

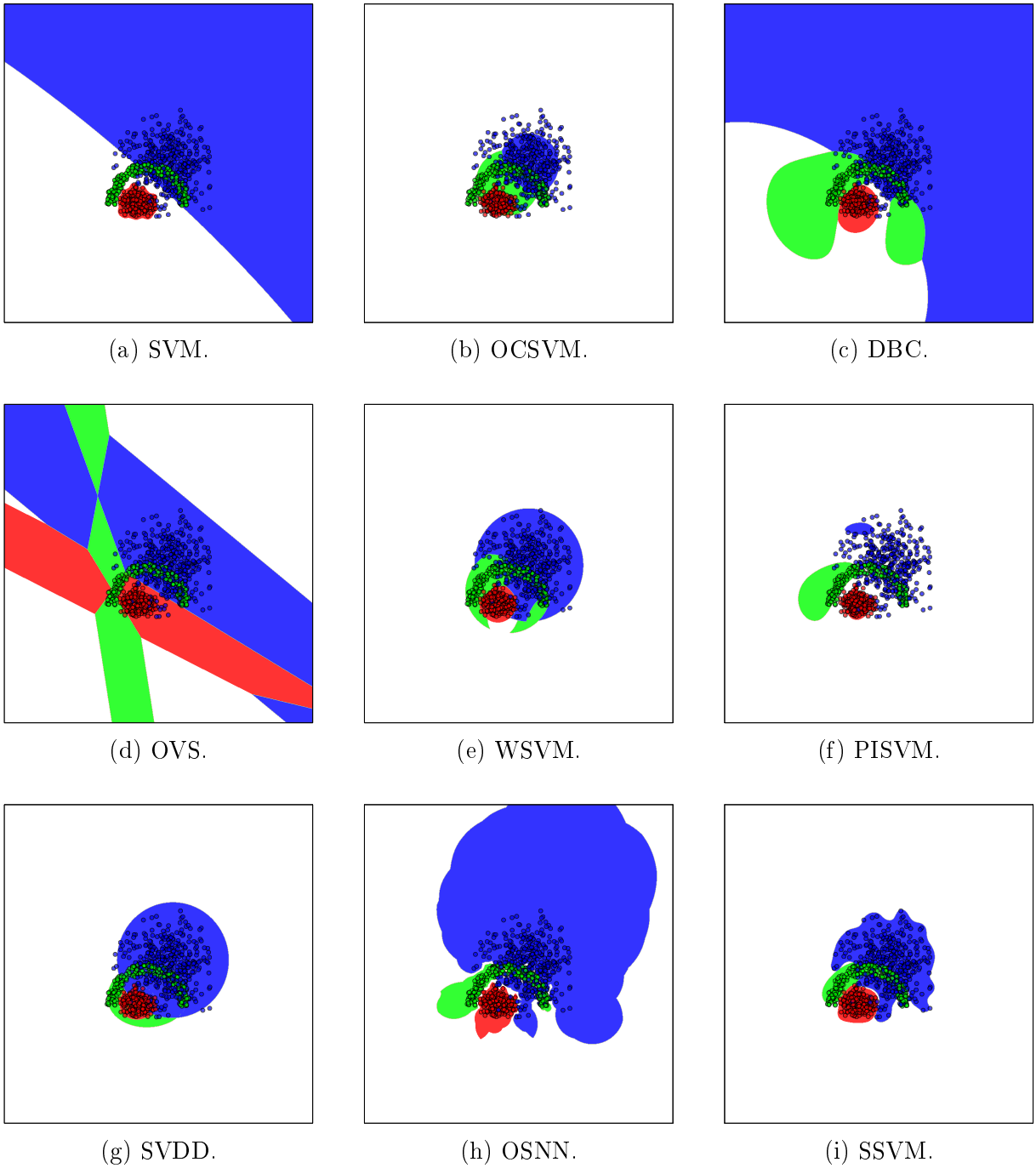


(a) SVM. (b) OCSVM. (c) DBC.

(d) OVS. (e) WSVM. (f) PISVM.

(g) SVDD. (h) OSNN. (i) SSVM.

Figure 6.20: Decision boundaries on the Half-Ring dataset. Behavior of the classifiers considered for the experiments. The small circles represent the training samples from the dataset and their colors represent their classes. A colored background represents the class in which a possible test instance in the corresponding position of the feature space would be classified. White background indicates that a possible test instance in that region would be classified as unknown.



(a) SVM. (b) OCSVM. (c) DBC.

(d) OVS. (e) WSVM. (f) PISVM.

(g) SVDD. (h) OSNN. (i) SSVM.

Figure 6.21: Decision boundaries on the Cone-Torus dataset. Behavior of the classifiers considered for the experiments. The small circles represent the training samples from the dataset and their colors represent their classes. A colored background represents the class in which a possible test instance in the corresponding position of the feature space would be classified. White background indicates that a possible test instance in that region would be classified as unknown.

Chapter 7

Neural networks

Neural networks—along with recent advances on deep learning [LeCun et al., 2015]—have obtained state-of-the-art performance in a broad range of machine-learning tasks and fields, e.g., computer vision, object detection, object segmentation, speech recognition, etc., and their development have been for both supervised and unsupervised classification problems. In addition, neural networks allow a straightforward employment, as they do not require previous feature extraction and raw data can directly be used as input data.

In this work, we focus on neural networks for classification problems. Despite all advancements reported in the literature, neural networks have been tested predominantly on closed-set scenarios, which makes us wonder if their employment on open-set scenarios will continue to present the expected behavior. This doubt is strengthened with results obtained with *adversarial images* [Szegedy et al., 2014, Goodfellow et al., 2015], which are able to trick the network on classifying a testing instance with high confidence to the incorrect class. Furthermore, networks are known to be susceptible to *fooling images* [Nguyen et al., 2015], which are images whose content is unrecognizable for humans and make a network to classify to certain classes also with high confidence.

More than bringing final conclusions and/or defining ready-to-use methods for open-set recognition with neural networks, our purpose here is to present results aiming at driving the intuition for dealing with open-set scenarios along with this type of classifiers. First, in Section 7.1, we present a mathematical formulation of neural networks that will help us on the analyses we present in Sections 7.2 and 7.3. In Section 7.2, we inspect the behavior of neural networks in low-dimensional input data aiming at assessing factors that could empower the network to properly handle the open space. In special, we consider the method of Bendale and Boult [2016] in this analysis. Finally, in Section 7.3, we avoid defining thresholds on networks' output and, instead, we evaluate the influence known unknown data have on final performance. The contribution of our work on neural networks resides on those experimental evaluations.

7.1 Neural networks for classification problems

For the purpose of our work, we consider the *multilayer perceptron* (MLP), a form of *feed-forward neural network* (FNN) [Bishop, 2006] for multiclass classification highly employed

nowadays. It consists of small processing *units* connected one to another by weighted edges. In case of an FNN, differently than a *Recurrent Neural Network* (RNN) [Graves, 2012], the connected units form a directed graph with no cycle, hence its name. The MLP is arranged in form of *layers* and each layer is composed of a set of units. Connections happen among units of consecutive layers however there is not intra-layer nor non-consecutive layer connections. The *input layer* represents the input data so that each of its units becomes a value of the input vector $\mathbf{x} \in \mathbb{R}^d$. The last layer of this arrangement is called the *output layer* and represents the prediction of the neural network. All other layers are called *hidden layers* because they are intermediate layers on the chain of computations. The number of layers of an MLP comprises the number of hidden layers plus one (the output layer), hence an input vector $\mathbf{x}$ can also be represented by $\mathbf{a}^{[0]}$ and the output of a l-layer network by $\mathbf{a}^{[l]} = \hat{\mathbf{y}}$. Except for units on the input layer, each one of them calculates a weighted sum and then applies an *activation function* to obtain its final activation. In summary, the final activation $a_j^{[l]}$ of a unit j on layer l is given by

$$a_j^{[l]} = \phi^{[l]}(z_j^{[l]} + b_j^{[l]}),$$

in which $\phi^{[l]}$ is an activation function employed on layer l, e.g., linear, sigmoid, hyperbolic tangent, *rectified linear unit* (ReLU), etc., $b_j^{[l]}$ is a bias term, and $z_j^{[l]}$ is the weighted sum:

$$z_j^{[l]} = \sum_{i=1}^{|L^{[l-1]}|} w_{ji}^{[l]} a_i^{[l-1]},$$

in which $L^{[l]}$ represents the set of units on layer l and $w_{ji}^{[l]}$ represents the weight of the connection between unit j of layer l with unit i of previous layer $l-1$.

Usually, for binary problems, a single unit is required on the output layer, along with a sigmoid activation function such that the final output is in $[0, 1]$, and can be used to represent the probability for the positive class. For multiclass problems, however, it is a well-established convention to have a unit per training class on the output layer, i.e., n units in total. In this case, a softmax function—a.k.a. softmax layer—as defined in Equation (7.1), is usually employed to obtain per-class probabilities [Bridle, 1990].

$$P(\ell_j|\mathbf{x}) = \frac{e^{\hat{y}_j}}{\sum_{i=1}^{|L^{[l]}|} e^{\hat{y}_i}}, \tag{7.1}$$

in which ℓ_i, $i = 1, \ldots, n$, indicate each of the training classes. Then, the final decision is simply given as ℓ_i, for

$$i = \arg\max_i P(\ell_i|\mathbf{x}).$$

It is proven that MLPs with a sufficient number of hidden units can approximate any continuous function [Hornik et al., 1989, Nielsen, 2015]. For training a network, i.e., adjusting its weights to approximate the desired function, the *backpropagation* technique has been a staple in the literature. In essence, it consists on propagating backwards the error calculated by the loss function through successive applications of partial derivatives

with respect to any weight and bias, by means of the chain rule. *Gradient descent*—or any of its alternative forms [Graves, 2012]—can then be employed to minimize any differentiable loss function. The gradient for updating the parameters of the network can be calculated on the entire training set (batch learning), however, when training set is large enough to make this approach unfeasible, *mini-batch* gradient descent should be employed. Mini-batch gradient descent consists of updating weights and biases based on the gradient calculated for a subset of training samples.

For the experiments in Section 7.2, we consider an MLP as defined here. In Section 7.3, we use a Convolutional Neural Network (CNN) [LeCun et al., 2015] for the experiments, however, as the theoretical foundation of CNNs is not required for the analysis we perform, we abstain to formalize it here.

7.2 Behavior analysis of fully-connected networks

The purpose of the experiments we present in this section is to assess the behavior of a neural network on the *input space* when decisions are accomplished on one of the feature spaces of the last layers of the network, e.g., considering the trivial extension of a neural network to open-set recognition by thresholding its probability score (softmax layer) for certain classes.

First, let us define what would be that trivial extension. One can think that if the network is not confident about its classification—i.e., if the probability score calculated by the softmax layer is not high enough, based on a threshold—then the test instance can be rejected as unknown. This rationale is basically implemented by Equation (7.2).

$$f(\mathbf{x}) = \begin{cases} \ell_i & \text{if } \max\limits_i P(\ell_i|\mathbf{x}) > T_s \\ \ell_0 & \text{otherwise,} \end{cases} \tag{7.2}$$

in which ℓ_0 is the unknown label and T_s, $0 \leq T_s < 1$, is some previously obtained threshold to be applied on the softmax probability estimate.

Recently, Bendale and Boult [2016] have proposed a more elaborated method for extending neural networks for open-set recognition. In essence, their method works as follows. The network is trained as it is usually accomplished in a closed-set scenario. Then, the purpose of their method is to estimate if an input test instance is from an unknown class. They first generate what they call the Mean Activation Vectors (MAVs), one per training class. The MAV $\mathbf{m}_k$ for a class k is calculated by extracting the activations $\mathbf{a}^{[l-1](i)}$ on the penultimate layer (before softmax) for every correctly-classified training sample i from class k. Their main hypothesis is that the MAV of a class represents how a sample of that class activates the penultimate layer. Then, by employing EVT, the method consists of estimating a Weibull distribution [Coles, 2001] per class k based on the largest distances $\|\mathbf{m}_k - \mathbf{a}^{[l-1](i)}\|$, for $i \in S_k$, in which S_k is the set of correctly-classified training samples from class k. On prediction phase, a test instance $\mathbf{x}$ is predicted to some class k by the closed-set neural network and its activation vector $\mathbf{a}^{[l-1]}$ on the penultimate layer is acquired for further verification. Then, the final decision is performed

as in Equation (7.3).

$$f(\mathbf{x}) = \begin{cases} \ell_k & \text{if } \max\limits_k P_k(y = k|d) > T_u \\ \ell_0 & \text{otherwise,} \end{cases} \tag{7.3}$$

in which P_k is the posterior of the previously-estimated Weibull distribution for class k, T_u is an uncertainty threshold, and $d = \|\mathbf{m}_k - \mathbf{a}^{[l-1]}\|$. The network layer that performs the verification described in Equation (7.3) is called the openmax layer.

For the purpose of the experiments we present in this section, we have considered a simplified version of openmax: instead of using a Weibull distribution, we have employed the well-known normal distribution, calibrated such that approximately 95% of the correctly-classified training instances are non-outliers on the penultimate-layer feature space. We have preferred the normal distribution to facilitate the interpretation of the behavior. As for T_s in Equation (7.2), we have considered $T_s = 0.97$.

In these experiments, our aim is to visualize the behavior of the network in a 2-dimensional input space. For this, we have defined a 3-layer neural network with 2 input units. From the first hidden layer to the output layer, this network has 384, 192, and n units, respectively, in which n is the number of training classes. For the first two layers, ReLU [Glorot et al., 2011] activation function was employed. Throughout those experiments, a mini-batch of size 40 was used, unless otherwise stated. For each dataset we will present, the network was trained in 1 000 000 steps (feedforward and weights update), although less steps would be enough to obtain an appropriate model. The same network model, along with softmax with rejection threshold of Equation (7.2), was also used as the base model for the openmax method.

Training data are from 2-dimensional synthetic datasets, always normalized in the interval $[0, 1]$. Examples of the datasets we used for training are the Boat, Four-Gauss, Petals, Regular, Saturn, Cone-Torus [Kuncheva and Hadjitodorov, 2004], and R15 [Veenman et al., 2002] datasets. We have also created an additional dataset similar to Four-Gauss, named Four-Gauss-Full, that has a similar shape compared to Four-Gauss, however, with more training samples. We have also generated the Seven-Gauss dataset—not as dense as Four-Gauss-Full—with seven known classes. All those datasets are depicted in Figure 7.1.

The network, as previously described, was trained on each of the datasets presented in Figure 7.1. Their decision boundaries for each dataset are presented in Figure 7.2. We can notice in this figure that generated models are able to separate well the samples in most cases. In general, we also observe a tendency on generating linear decision frontiers when possible, as can be seen between the two middle classes of Boat dataset in Figure 7.2a and between classes in Petals, Regular, Half-Ring, R15, and Four-Gauss-Full datasets. In special, we notice linear decision frontiers among classes in Half-Ring and Four-Gauss-Full, which do not allow proper separation of the entire training set. We conjecture it is due to small size of mini-batch compared to the entire training set. One intuitive explanation is that for each mini-batch of randomly selected samples, it is likely for those datasets that a linear decision boundary can separate them and from step to step the model for each dataset keeps those frontiers linear. After presenting the decision boundaries for softmax

with rejection threshold and for openmax, we will return to this point.

Firstly, in Figure 7.3, we present the decision boundaries for the same neural networks (same weights) with the openmax layer included. As we can see, in most cases, openmax gracefully bounds the KLOS on the input space. Interestingly, the cases with non-linear-shaped boundaries from Figure 7.2, makes the openmax more likely to bound the KLOS. Bendale and Boult [2016] have proved that the KLOS (i.e., the open-space risk) is bounded. However, their proof applies only to the feature space of the penultimate layer. It does not avoid that an unbounded region on the feature space of previous layers—including the input space—is mapped to similar activations on the penultimate layers, making the KLOS on previous layers unbounded. In fact, Figures 7.3g, 7.3h, and 7.3i seem to confirm it. It is a question due to debate whether the known-labeled space needs to be bounded *de facto* on the input space. We leave it for future analyses as we do not have the final answer on that matter, however, we should notice that leaving an unbounded KLOS, as in Figure 7.3g, seems unreasonable and unsafe, as the behavior of the network outside the support of the training samples seems unpredictable. For instance, in Figure 7.4, we analyze the cases of Figure 7.3 with (seemingly) unbounded KLOS in a larger portion of the feature space. As we can see, the openmax layer (apparently) still leaves an unbounded KLOS for Half-Ring, as shown in Figure 7.4d. For Cone-Torus and Four-Gauss-Full, Figures 7.4e and 7.4f evince openmax is still not able to bound the KLOS in the range $[-10, 11]$ of the input space, however, it is not clear whether it might be able to bound the KLOS at some point of that feature space.

In Figure 7.5, we show the decision frontiers for the method defined in Equation (7.2). As we can see, establishing a threshold on softmax only makes doubtful testing samples to be rejected while a great part of the open space is still labeled as known. It indicates that even by increasing the rejection threshold T_s, it would only make the decision frontier to be tighter among known classes but the high confidence region on the open space would still remain. Anyhow, by thresholding softmax, in some cases it makes the KLOS bounded further away in the feature space, as can be noticed on the corners of Figure 7.5e and evinced for several other datasets in Figure 7.6, in which we present the behavior in a broader region of the input space (for the range $[-10, 11]$). The reason is that a point further away in the open space starts having similar probabilities for every class, hence there will be no highly activated class for a point sufficiently far in the open space. However, Figures 7.5g, 7.5h, and 7.5i present no indication it might happen at some point.

Those results help us understand the results obtained by Nguyen et al. [2015]: it evinces the possibility of obtaining fooling images with high confidence for certain classes—sometimes with more confidence than for training samples themselves. As the method of Bendale and Boult [2016] bounds the KLOS on the input space *in some cases*—by bounding the KLOS on the penultimate-layer space—those results we have presented are an explanation why the openmax is also able to correctly reject some fooling images as well as *rubbish images* [Goodfellow et al., 2015], as reported by the authors [Bendale and Boult, 2016]. We also visually confirm the linearity problem of neural networks as stated by Goodfellow et al. [2015]: it is likely to leave an unbounded KLOS, as we could see.

Previously, we have observed the linear behavior of the network on Four-Gauss-Full

dataset. We have hypothesized it is due to the small mini-batch size compared to the total number of the training samples. Aiming at further checking and evincing this hypothesis, we have trained the same network with a larger mini-batch size of 800. In this case, the generated model better separated training instances non-linearly, as shown in Figure 7.7. As we can see in Figure 7.7b, now openmax is able to bound the KLOS on the input space, as decision frontiers are no longer linear. However, softmax still continues to yield high confidence scores for possible test instances that might appear on the feature space outside the support of the training samples.

7.3 Partial knowledge of the unknown

In this section, we analyze the performance of neural networks when simulating an "unknown class". In this case, instead of establishing a threshold for rejection, the networks are trained with *known unknown* classes [Bendale and Boult, 2015]. The known unknown classes comprise the semantic classes of any instance that can be acquired at training phase but for which we have no interest in recognizing them. For this purpose, it is not mandatory defining the label for each of those instances as long as we can assure they do not belong to any of the classes of interest. On the other hand, *unknown unknown* in an open-set setup refers to the classes for which representative samples are not available for training.

Our objective is to understand the impact of the assumption that by including on training of a neural network as many known unknown samples as possible would make the network to learn to recognize unknown classes. We have seen in Section 6.3 that, for an SVM with the one-vs-all strategy, when the negative class comprises multiple known classes, it makes the SVM more likely to bound the PLOS, generating a model suitable for open-set scenarios. The factor we want to analyze here is similar, however, with an empirical approach, i.e., through experiments with multiple configurations.

For the purpose of those experiments, we have employed a publicly available CNN [Tensorflow.org, 2018b]. This network is targeted for closed-set digit classification on MNIST dataset and achieves a classification accuracy of approximately 99.2% in the closed-set setup among 10 classes. The input layer is 28×28 pixels, followed by 7 layers: two convolutional layers interchanged with two max pooling layers and two fully-connected (FC) layers at the end followed by the softmax layer. ReLU is employed along with each convolutional layer and the first FC layer. For regularization, this network performs a dropout [Srivastava et al., 2014] on the first FC layer. We have employed a mini-batch of size 50.

For the setup of those experiments, we have split the 10 MNIST classes into K, K_u, and U sets such that $|K| + |K_u| + |U| = |K \cup K_u \cup U| = 10$. For each experiment, K is the set of known classes used for training (the classes of interest); K_u is the set of known unknown classes, i.e., classes not of interest however used to aid the network model at recognizing the unknown; and U is the set of unknown classes that appear only on prediction time.

Trained networks have $n+1$ units at the two last layers: $n = |K|$ for each of the known

classes and +1 for the (known) unknown class. This way, we can assess the performance of the network when partial knowledge of the unknown is included in the trained model.

For testing, besides MNIST, we have also employed Chars74K [de Campos et al., 2009] dataset, which consists of digits and letters in a distinct domain. K^d represents the set of known digits; K_u^d represents the set of known unknown digits; U^d represents the set of unknown digits; and U^l represents the set of (unknown) letters from Chars74K. The instances from Chars74K were resized to be 28×28 and converted to grayscale.

Furthermore, we have generated two additional datasets: R consists of instances whose pixels have received a random intensity from the range $[0, 255]$ and R^p consists of test instances from MNIST dataset with their pixels shuffled. Both R and R^p represent unknown samples from the point of view of every trained network in those experiments.

In Table 7.1, we present results when training the networks only with data from the MNIST dataset. In Table 7.2, networks were trained also with samples for known classes from Chars74K datasets. In Table 7.3, networks do *not* use known samples from Chars74K, however, it uses samples from Chars74K for the known unknown set of classes. Finally, in Table 7.4, trained networks use instances from Chars74K for both known and known unknown classes. In every case, known and known unknown classes from MNIST are used. Samples that appear for training a network, do not appear for testing in any other network and vice-versa. Obtained results for each $(|K|, |K_u|, |U|)$ configuration denote the mean of 10 experiments with distinct randomly selected classes.

The main result we analyze here is the one obtained in MNIST's unknown unknown set. As we are not dealing with the problem of domain adaptation, results on Chars74K dataset are extra considerations. For instance, we would not expect a network not trained with samples from Chars74K to be able to perform well on the set K^d of known classes from Chars74K. However, we would expect an open-set classifier to be able to reject the unknown classes from Chars74K.

First, let us analyze the performance only for the MNIST's K, K_u, and U test sets. In Table 7.1, we observe that results on K and K_u are similar to the ones for closed-set, i.e., around 99.2%, evincing that the network continues to fit well on the available classes, as expected. Furthermore, as the number of classes of interest considered on the experiments in Table 7.1 is smaller than the equivalent closed-set experiments (always 10 classes), we observe a slight improvement. The K_u in Table 7.1—and in the other tables as well—comprises the test set referring to the known unknown classes. Results on K_u is consistently better than results on K set, and they improve as the ratio $|K_u|/|K|$ increases.

Those two results show that by including a set of known unknown classes for training the network, its performance is not affected in those two sets. However, it does not make the network able to recognize true unknown samples, as shown by the accuracy obtained on the U test set. Anyhow, by increasing the size of K_u, compared to the size of K, the likelihood of rejecting true unknown samples increases. At first glance, it might indicate that by introducing as much data as possible as known unknown would solve the open-set problem on neural networks, however, from further analyses, this conclusion cannot be drawn, as we shall see ahead.

By comparing the performance on K, K_u, and U sets across Tables 7.1–7.4, we observe

that the introduction of data from Chars74K in training—as in Tables 7.2–7.4—does not change the performance neither for good nor for bad. Anyhow, it is interesting to notice that by introducing data from a distinct domain does not change its behavior on the main domain.

That last observation indicates us that the main factor for the improved performance on U set, as the size of K_u increases, is the increase of the ratio $|K_u|/|K|$. It makes the network simply more likely to reject instances in general, which does not mean it is taking into account the content of the samples for rejecting them.[1] Anyhow, the accuracy on K does not decrease as the ratio $|K_u|/|K|$ increases, hence we can also infer that the introduction of known unknown samples from the same domain of the classes of interest can help the network to recognize at least a part of the unknown world without affecting the performance on recognizing the classes of interest.

We have observed across Tables 7.1–7.4 that by including samples from Chars74K in the training, the performance on MNIST data was not affected much. On the other hand, as more known unknown classes from MNIST are included on training, the better the accuracy on unknown samples from Chars74K dataset—as observed across rows of Table 7.1—although the accuracy on the set K^d of known samples from Chars74K suffers. Observe, however, that in this case—across the rows of the same table—the configuration of those test sets changes, which is not true when comparing the same cases across tables.

Test sets R and R^p are kept the same across rows on those tables and, in fact, they can indicate that the model better rejects unknown samples as the ratio $|K_u|/|K|$ increases. As saw before, it happens without affecting the performance on known samples in K, from the main domain. However, it is a casual behavior of the network, as the network becomes more likely to reject unknown instances in general as that ratio increases. The results for R and R^p in Table 7.2 strongly evinces this statement. By introducing known samples from Chars74K on training, it makes the network to misclassify the set R almost entirely as well as significantly decrease the accuracy on R^p. Furthermore, the inclusion of those samples on training disturb the behavior on the sets K_u^d, U^d, and U^l of unknown samples from Chars74K. Remember that R comprises samples whose pixels are randomly generated and R^p was created based on images from MNIST with their pixels shuffled, which makes those results and the following ones unexpected.

In Table 7.3—with known unknown training samples from Chars74K—the scenario reverses: accuracy on R reaches 100% in multiple cases and accuracy on R^p is reasonable. As expected, results for K^d, K_u^d, U^d, and U^l are also reversed, compared to Table 7.2.

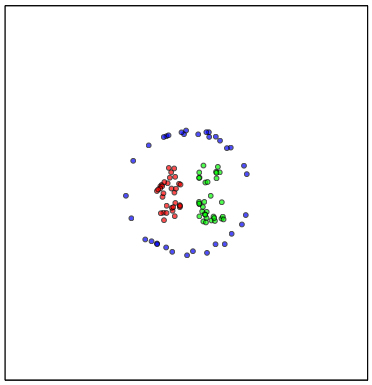
Finally, in Table 7.4—when considering samples from Chars74K as both known and known unknown training data—we observe that the network performs as if two separated models where trained. We have observed before—by comparing results for K, K_u, U across the tables—that the model for MNIST data is not affected by the introduction of data from Chars74K on training. Now, we observe for Chars74K test sets a similar behavior we have previously observed for MNIST: results on known and known unknown sets—K^d and K_u^d, respectively—maintains a reasonable accuracy while the accuracy on the true unknown data is usually worse.

[1] In this exploration, we have not focused too much on the balancing of the training classes for each mini-batch, as this was not the main topic of research by itself.

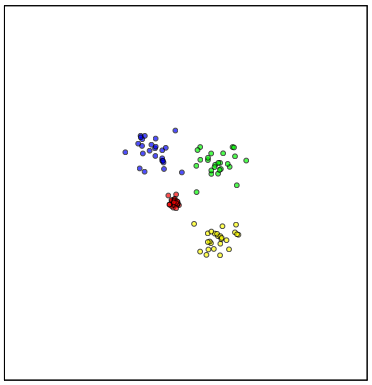
7.4 Final considerations

With those analyses and results we have shown, at present, it is not reasonable to expect a secure behavior from neural networks in the open-set scenario by simply giving them a massive amount of training data, even when a partial representation of the unknown is included among the known unknown data being used. Neural networks are data-driven methods and that is their very advantage over other classifiers, however, we have seen that undesirable behavior can happen under certain circumstances, as their behavior in the open space usually cannot be inferred based solely on known data or on the obtained model. The results we have obtained for R and R^p—along with the analysis in the previous section—are an important indication of those unexpected behaviors. As unknown data is not available for training and, furthermore, in some applications the type of input data cannot be predicted a priori, those analyses indicate the need of understanding the particularities that would allow us to make neural networks more robust and reliable in the open space.

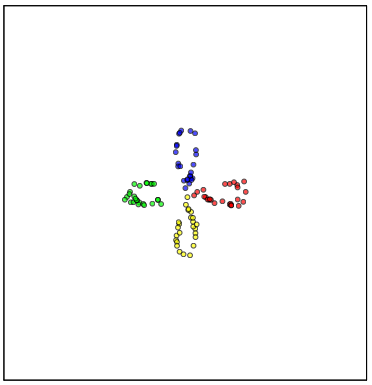
The previous analyses we have presented broaden the view of the field of open-set recognition along with neural networks, as neural networks for open-set scenarios should optimize not only the empirical risk but also the open-space risk. As for other classifiers, we argue its behavior and properties should be analysed in their essence so that theoretical guarantees should be provided. For instance, consider the analysis of openmax layer, which is able to obtain a bounded KLOS on the input space for some cases. For the cases for which KLOS is bounded, it means that network's model ensures that distinct configurations from previous layers distinctly activates further layers so that bounding the KLOS on the last layer ensures a bounded KLOS on the input space. If one can guarantee it happens from layer to layer, independently from the training data, then the same open-set properties we have guaranteed for other classifiers would be guaranteed for MLP. Furthermore, further studies can also be accomplished along with CNN aiming at the same objective.



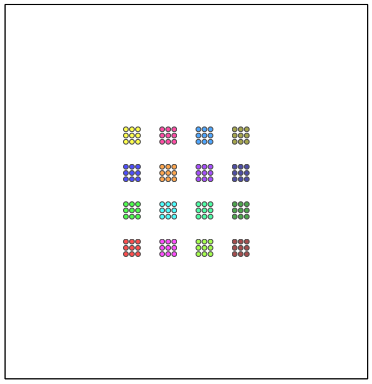
(a) Boat (100 instances).



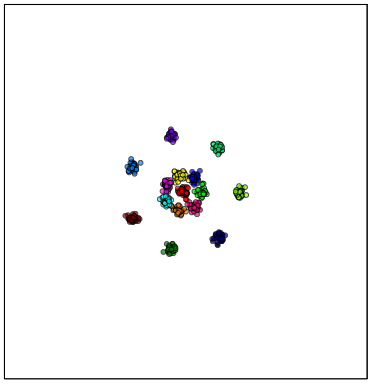
(b) Four-Gauss (100 instances).



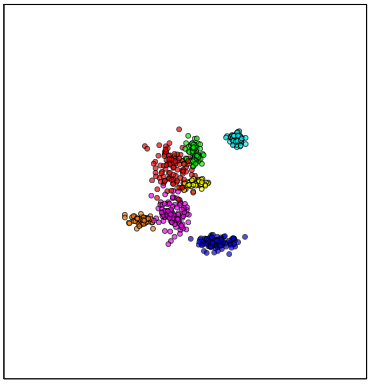
(c) Petals (100 instances).



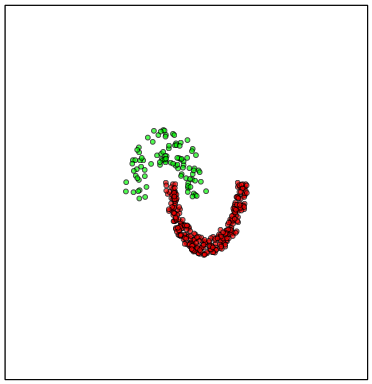
(d) Regular (144 instances).



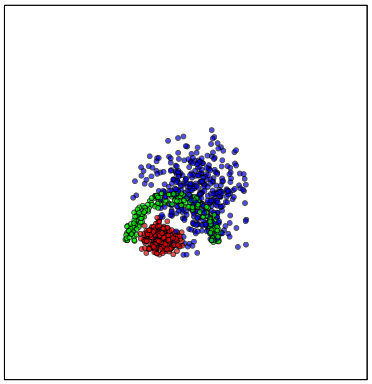
(e) R15 (600 instances).



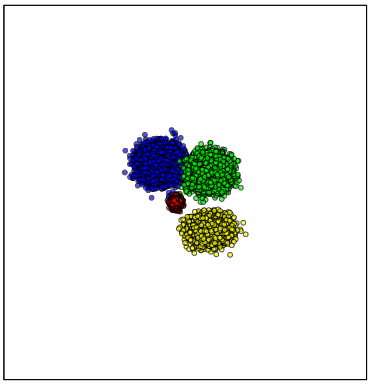
(f) Seven-Gauss (570 instances).



(g) Half-Ring (373 instances).

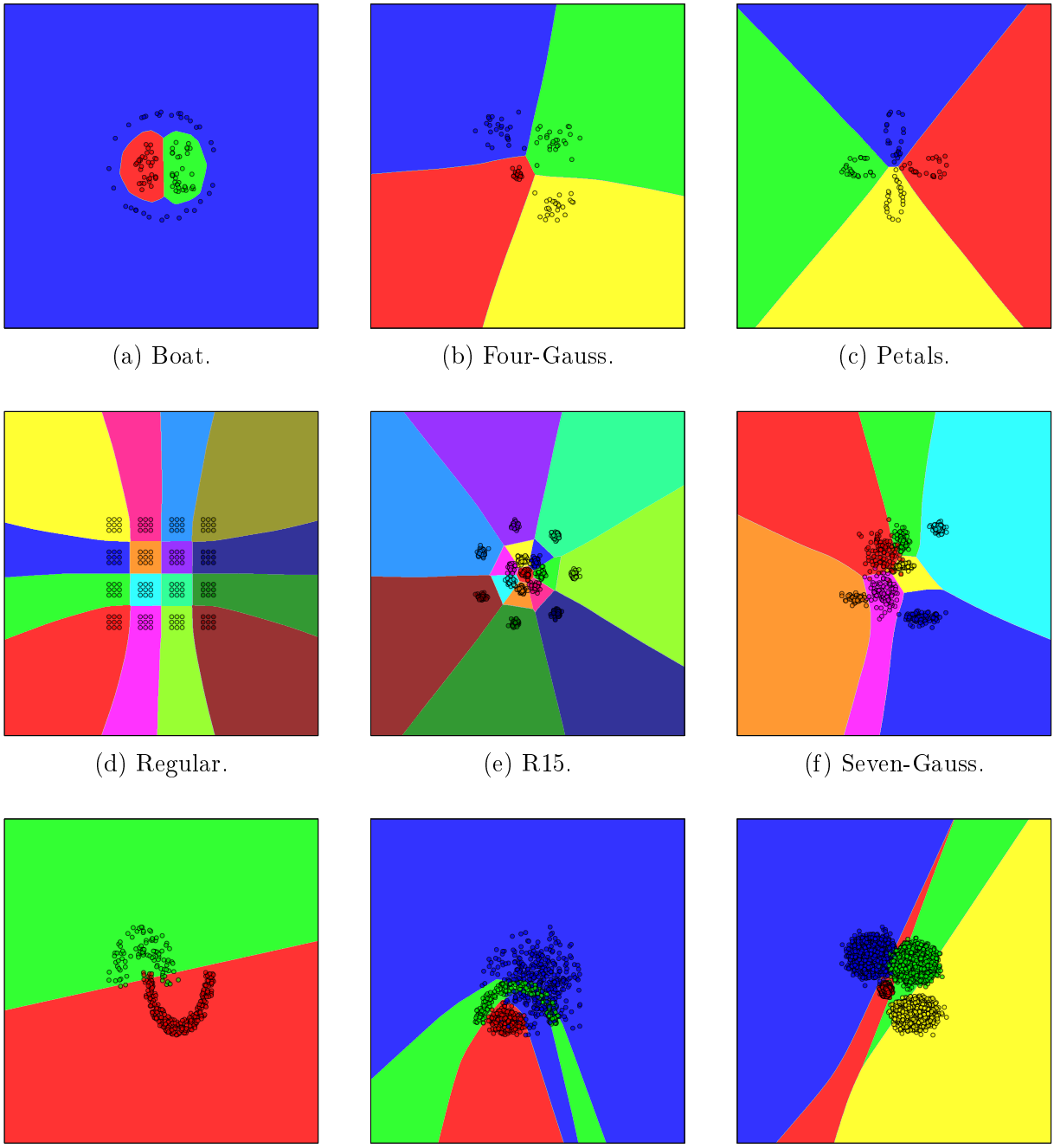


(h) Cone-Torus (800 instances).



(i) Four-Gauss-Full (20000 instances).

Figure 7.1: 2-dimensional datasets employed on the behavior analysis of neural networks. Colored points represent training samples for the neural network. Training data are normalized in the interval $[0, 1]$ for each feature. Each image shows the range $[-1, 2]$ in the feature space for each feature. The small circles represent the training samples from the dataset and their colors represent their classes.

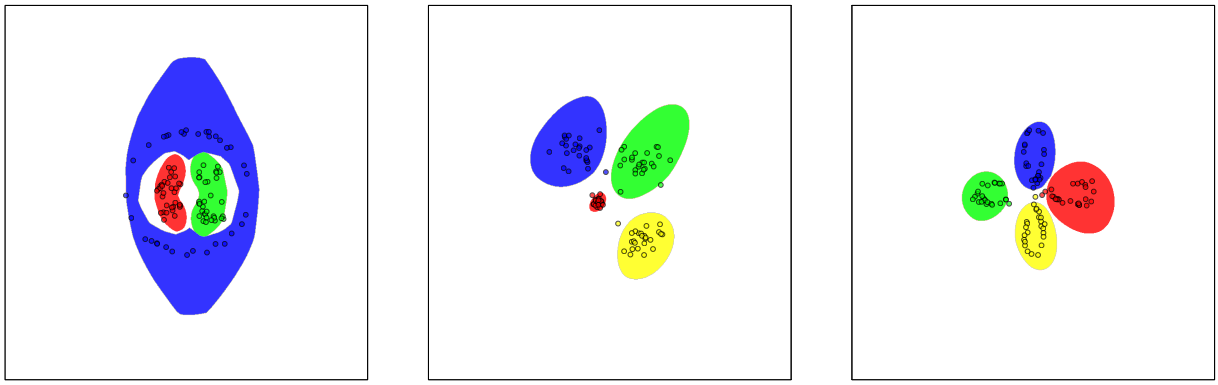


(a) Boat. (b) Four-Gauss. (c) Petals.

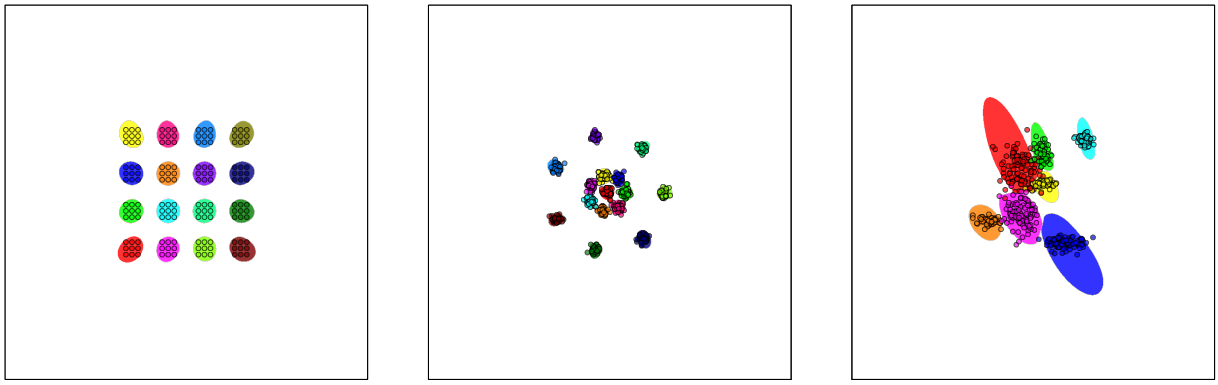
(d) Regular. (e) R15. (f) Seven-Gauss.

(g) Half-Ring. (h) Cone-Torus. (i) Four-Gauss-Full.

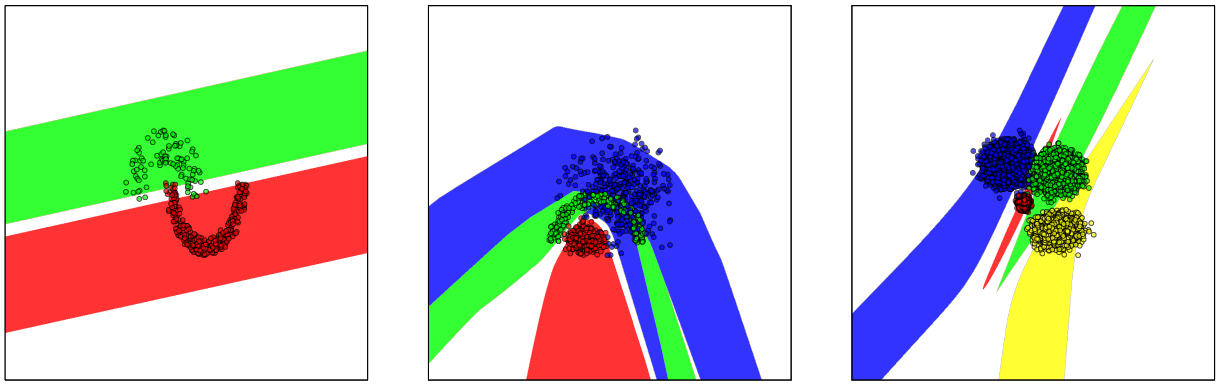
Figure 7.2: Behavior analysis of the closed-set neural network. Images generated without employing any kind of rejection criteria, hence, a closed-set behavior is presented. The small circles represent the training samples from the dataset and their colors represent their classes. A colored background represents the class in which a possible test instance in the corresponding position of the feature space would be classified. White background indicates that a possible test instance in that region would be classified as unknown.



(a) Boat. (b) Four-Gauss. (c) Petals.

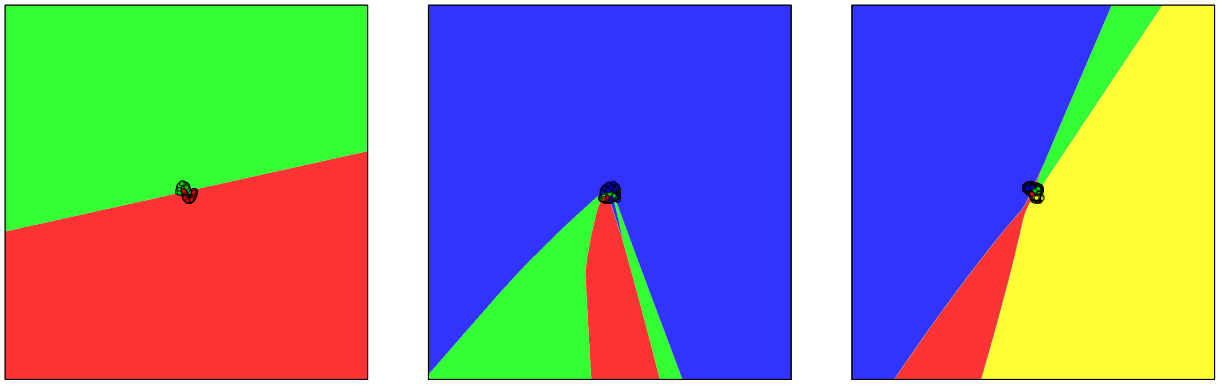


(d) Regular. (e) R15. (f) Seven-Gauss.



(g) Half-Ring. (h) Cone-Torus. (i) Four-Gauss-Full.

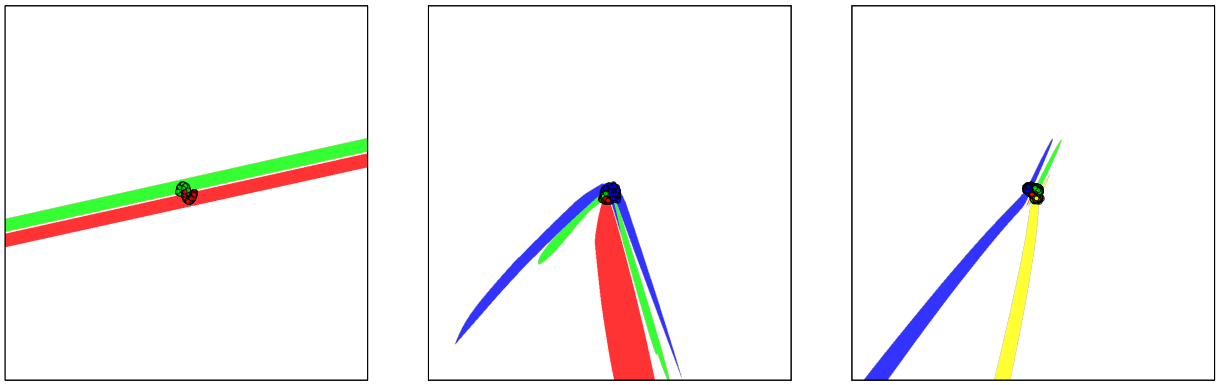
Figure 7.3: Behavior analysis of the neural network with openmax rejection layer. Openmax layer rejects a test instance when its activation vector on the penultimate layer is dissimilar to the Mean Activation Vector of the predicted class. The small circles represent the training samples from the dataset and their colors represent their classes. A colored background represents the class in which a possible test instance in the corresponding position of the feature space would be classified. White background indicates that a possible test instance in that region would be classified as unknown.



(a) Closed-set. Half-Ring.

(b) Closed-set. Cone-Torus.

(c) Closed-set. Four-Gauss-Full.

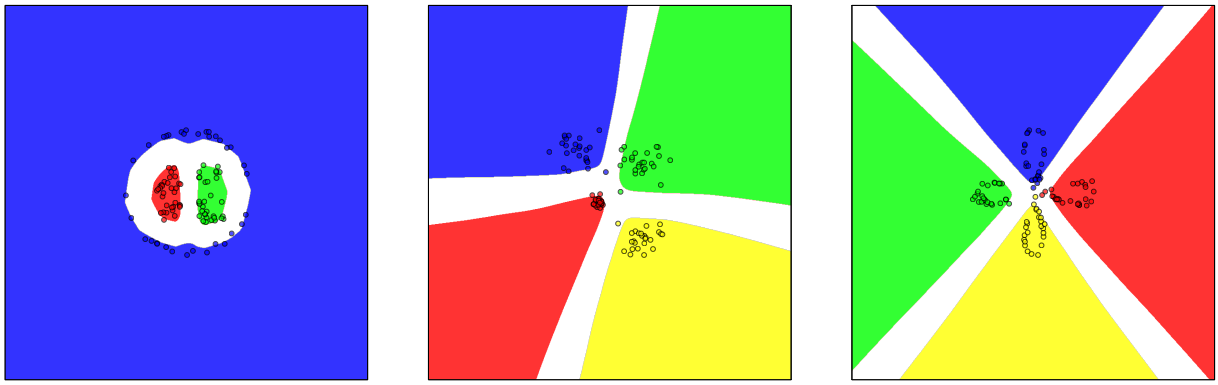


(d) Openmax. Half-Ring.

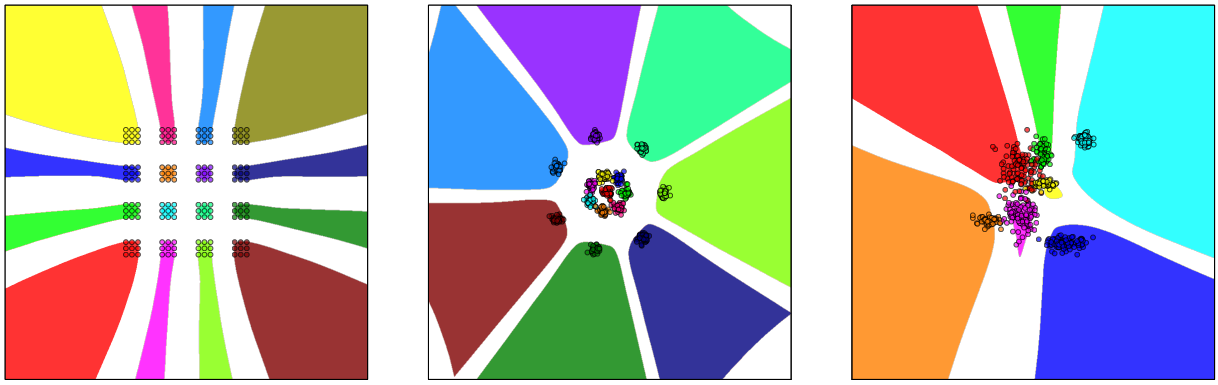
(e) Openmax. Cone-Torus.

(f) Openmax. Four-Gauss-Full.

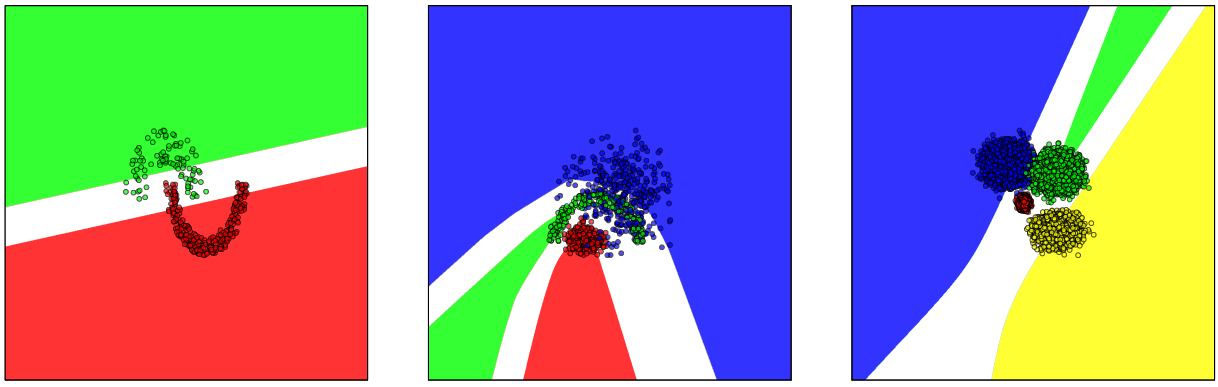
Figure 7.4: Behavior analysis of the neural network with openmax rejection layer far from training samples. Neural networks behavior for the open space far from training samples. Depiction of decision boundaries for closed-set neural network, openmax layer, and softmax layer with threshold. Images represent the 2-dimensional input space in the range $[-10, 11]$. The small circles represent the training samples from the dataset and their colors represent their classes. A colored background represents the class in which a possible test instance in the corresponding position of the feature space would be classified. White background indicates that a possible test instance in that region would be classified as unknown.



(a) Boat. (b) Four-Gauss. (c) Petals.

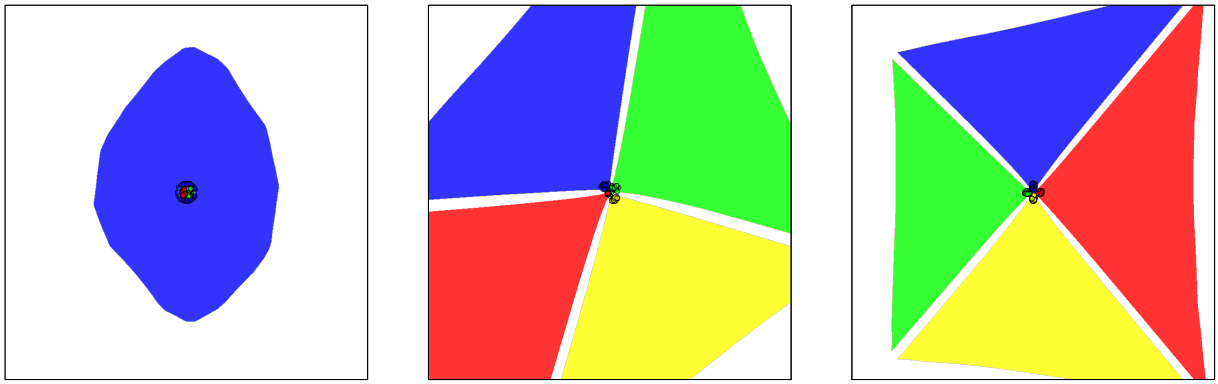


(d) Regular. (e) R15. (f) Seven-Gauss.

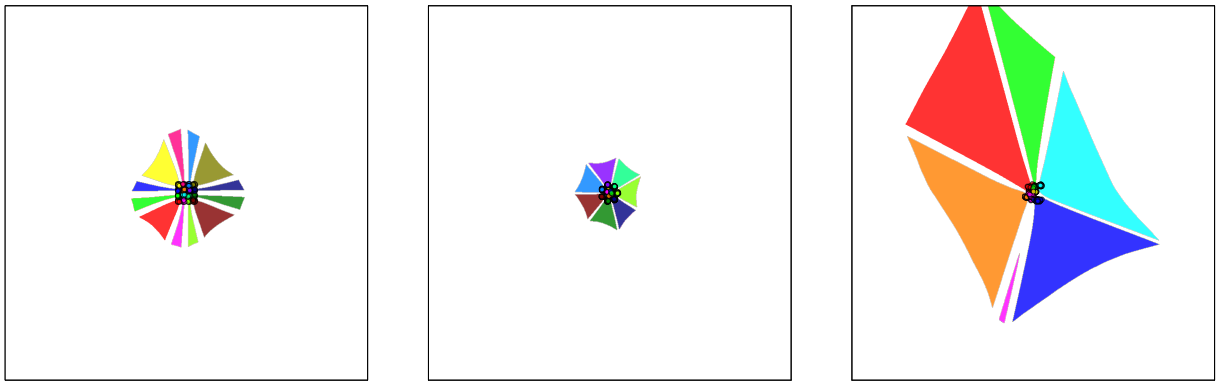


(g) Half-Ring. (h) Cone-Torus. (i) Four-Gauss-Full.

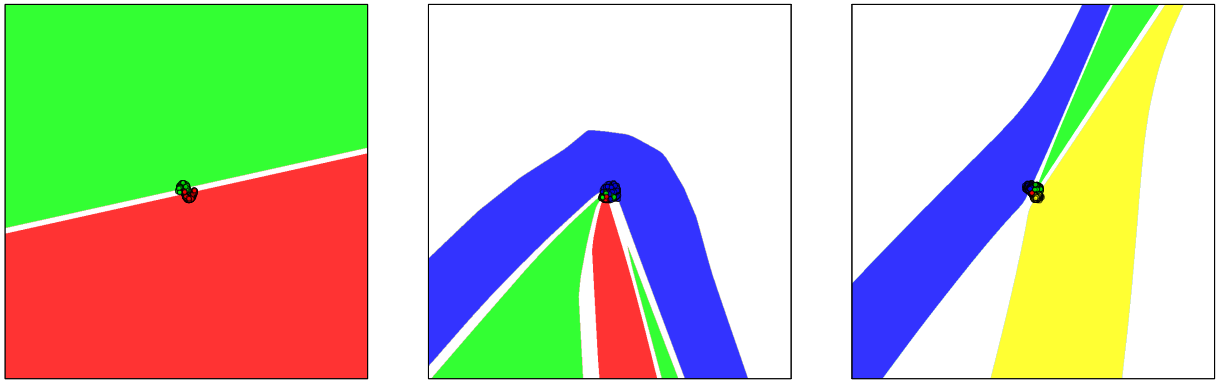
Figure 7.5: Behavior analysis of the neural network by establishing a rejection threshold on the softmax layer. When the probability to the most probable class is not high enough, the test instance is rejected. The small circles represent the training samples from the dataset and their colors represent their classes. A colored background represents the class in which a possible test instance in the corresponding position of the feature space would be classified. White background indicates that a possible test instance in that region would be classified as unknown.



(a) Boat. (b) Four-Gauss. (c) Petals.

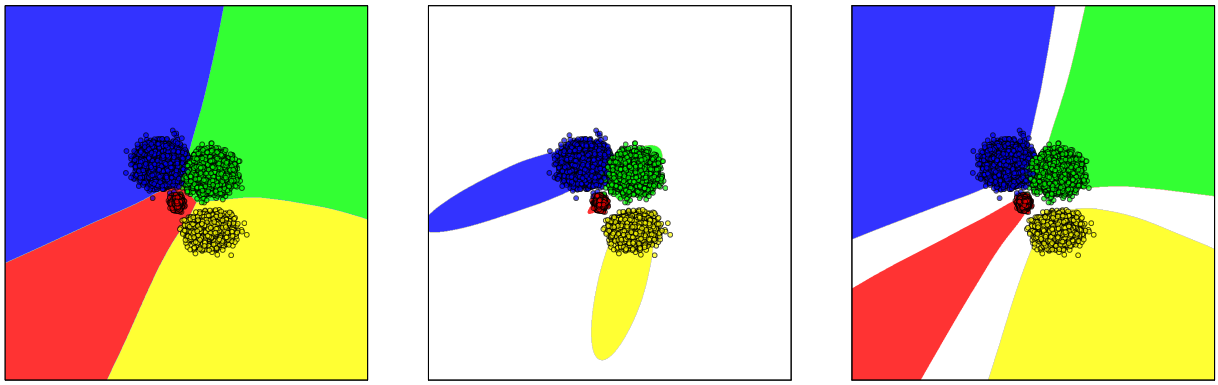


(d) Regular. (e) R15. (f) Seven-Gauss.



(g) Half-Ring. (h) Cone-Torus. (i) Four-Gauss-Full.

Figure 7.6: Behavior analysis of the neural network by establishing a rejection threshold on the softmax layer far from training samples. When the probability to the most probable class is not high enough, the test instance is rejected. Images represent the 2-dimensional feature space in the range $[-10, 11]$. The small circles represent the training samples from the dataset and their colors represent their classes. A colored background represents the class in which a possible test instance in the corresponding position of the feature space would be classified. White background indicates that a possible test instance in that region would be classified as unknown.



(a) Closed-set. Four-Gauss-Full.

(b) Openmax. Cone-Torus.

(c) Softmax. Cone-Torus.

Figure 7.7: Behavior analysis of the neural network for Four-Gauss-Full dataset with mini-batch of size 800. Depiction of decision boundaries for closed-set neural network, openmax layer, and softmax layer with threshold. Images represent the 2-dimensional feature space in the range $[-1, 2]$. The small circles represent the training samples from the dataset and their colors represent their classes. A colored background represents the class in which a possible test instance in the corresponding position of the feature space would be classified. White background indicates that a possible test instance in that region would be classified as unknown.

$(\|K\|, \|K_u\|, \|U\|)$	K	K_u	U	K^d	K_u^d	U^d	U^l	R	R^p
$(6, 2, 2)$	0.9935	0.9939	0.1480	0.4029	0.5069	0.2977	0.3624	0.3296	0.2453
$(5, 2, 3)$	0.9942	0.9948	0.3176	0.2790	0.6579	0.4514	0.5038	0.5400	0.5580
$(6, 3, 1)$	0.9916	0.9947	0.4797	0.2652	0.7418	0.6490	0.6261	0.4997	0.5348
$(4, 2, 4)$	0.9941	0.9944	0.3003	0.3469	0.6866	0.4739	0.5265	0.4466	0.3871
$(5, 3, 2)$	0.9932	0.9953	0.5229	0.2492	0.8281	0.7613	0.7442	0.8585	0.6245
$(3, 2, 5)$	0.9951	0.9968	0.4258	0.2029	0.8931	0.7635	0.7677	0.7802	0.7051
$(4, 3, 3)$	0.9922	0.9974	0.5069	0.2292	0.8689	0.7218	0.7635	0.8564	0.5341
$(5, 4, 1)$	0.9918	0.9955	0.5360	0.1834	0.8748	0.8404	0.8154	0.7573	0.6387
$(4, 4, 2)$	0.9910	0.9954	0.6064	0.1133	0.9524	0.9185	0.8805	0.8562	0.8713
$(3, 3, 4)$	0.9943	0.9970	0.5076	0.2639	0.8765	0.7448	0.7960	0.7992	0.7011
$(2, 2, 6)$	0.9941	0.9984	0.4962	0.4527	0.7592	0.6049	0.6445	0.6727	0.7080
$(4, 5, 1)$	0.9891	0.9966	0.6648	0.1327	0.9781	0.9367	0.9563	0.9639	0.9608
$(3, 4, 3)$	0.9924	0.9969	0.6655	0.1554	0.9469	0.9200	0.9332	0.9568	0.7998
$(2, 3, 5)$	0.9940	0.9983	0.6822	0.2067	0.9826	0.9008	0.9202	0.9517	0.9134
$(3, 5, 2)$	0.9907	0.9970	0.8213	0.1139	0.9588	0.9253	0.9256	0.9555	0.8220
$(3, 6, 1)$	0.9910	0.9969	0.8389	0.0815	0.9881	0.9938	0.9511	0.9886	0.8008
$(2, 4, 4)$	0.9929	0.9982	0.7403	0.2999	0.9018	0.8685	0.8622	0.9146	0.8157
$(2, 5, 3)$	0.9921	0.9978	0.8098	0.0997	0.9944	0.9663	0.9744	0.9999	0.8124
$(2, 6, 2)$	0.9879	0.9982	0.8847	0.1100	0.9895	0.9615	0.9815	0.9957	0.8195

Table 7.1: Results on MNIST and Chars74K datasets with networks trained with known and known unknown classes of MNIST. Each line represents the mean of 10 experiments with randomly selected $(|K|, |K_u|, |U|)$ digits. Lines are sorted by $|K_u|/|K|$. Red background indicates low accuracy (for visualization purposes).

$(\|K\|, \|K_u\|, \|U\|)$	K	K_u	U	K^d	K_u^d	U^d	U^l	R	R^p
$(6, 2, 2)$	0.9932	0.9949	0.1708	0.8402	0.1208	0.0000	0.0352	0.0011	0.0551
$(5, 2, 3)$	0.9938	0.9936	0.3030	0.8311	0.1292	0.0367	0.0220	0.0002	0.2562
$(6, 3, 1)$	0.9919	0.9949	0.4353	0.8383	0.1198	0.0534	0.0382	0.0007	0.1697
$(4, 2, 4)$	0.9945	0.9952	0.3206	0.8813	0.1177	0.0248	0.0392	0.0011	0.3235
$(5, 3, 2)$	0.9932	0.9944	0.4695	0.8526	0.1085	0.0275	0.0415	0.0011	0.2818
$(3, 2, 5)$	0.9948	0.9967	0.3966	0.8865	0.2067	0.0491	0.0330	0.0005	0.3105
$(4, 3, 3)$	0.9927	0.9967	0.4863	0.8367	0.1577	0.0692	0.0622	0.0010	0.2335
$(5, 4, 1)$	0.9908	0.9952	0.5347	0.8195	0.1483	0.0866	0.0698	0.0020	0.3124
$(4, 4, 2)$	0.9916	0.9949	0.5970	0.8523	0.1928	0.0864	0.0521	0.0039	0.4767
$(3, 3, 4)$	0.9935	0.9972	0.5305	0.8945	0.1362	0.0487	0.0836	0.0027	0.4126
$(2, 2, 6)$	0.9958	0.9970	0.4552	0.8927	0.1583	0.0480	0.0588	0.0028	0.3610
$(4, 5, 1)$	0.9882	0.9967	0.7300	0.8349	0.1225	0.0767	0.0901	0.0004	0.5658
$(3, 4, 3)$	0.9928	0.9960	0.6586	0.8574	0.1371	0.0900	0.0762	0.0000	0.3695
$(2, 3, 5)$	0.9953	0.9978	0.6374	0.8544	0.1904	0.0407	0.0669	0.0000	0.5080
$(3, 5, 2)$	0.9901	0.9969	0.8426	0.8559	0.1472	0.1194	0.0792	0.0000	0.4118
$(3, 6, 1)$	0.9909	0.9975	0.8592	0.8768	0.1747	0.1282	0.0783	0.0000	0.4786
$(2, 4, 4)$	0.9931	0.9980	0.7216	0.9273	0.1670	0.1214	0.0907	0.0005	0.4299
$(2, 5, 3)$	0.9917	0.9979	0.7951	0.8848	0.1726	0.0952	0.0949	0.0120	0.6134
$(2, 6, 2)$	0.9886	0.9981	0.8564	0.8791	0.1652	0.1366	0.1088	0.0000	0.5057

Table 7.2: Results on MNIST and Chars74K datasets with networks trained with known and known unknown classes of MNIST and known classes of Chars74K. Each line represents the mean of 10 experiments with randomly selected $(|K|, |K_u|, |U|)$ digits. Lines are sorted by $|K_u|/|K|$. Red background indicates low accuracy (for visualization purposes).

$(\|K\|, \|K_u\|, \|U\|)$	K	K_u	U	K^d	K_u^d	U^d	U^l	R	R^p
$(6, 2, 2)$	0.9936	0.9948	0.1553	0.0538	0.9984	0.9825	0.9639	1.0000	0.7266
$(5, 2, 3)$	0.9937	0.9953	0.3404	0.0512	1.0000	0.9908	0.9642	1.0000	0.9370
$(6, 3, 1)$	0.9925	0.9938	0.4670	0.0382	0.9988	0.9818	0.9766	1.0000	0.8778
$(4, 2, 4)$	0.9953	0.9957	0.3362	0.0422	0.9970	0.9828	0.9739	1.0000	0.8202
$(5, 3, 2)$	0.9928	0.9949	0.4718	0.0330	0.9957	0.9868	0.9789	1.0000	0.8899
$(3, 2, 5)$	0.9949	0.9963	0.4173	0.0319	0.9986	0.9850	0.9730	1.0000	0.8067
$(4, 3, 3)$	0.9930	0.9960	0.4815	0.0389	0.9958	0.9776	0.9848	1.0000	0.8570
$(5, 4, 1)$	0.9913	0.9949	0.5162	0.0212	0.9991	0.9958	0.9870	1.0000	0.9280
$(4, 4, 2)$	0.9912	0.9953	0.5992	0.0157	0.9991	0.9948	0.9876	1.0000	0.9772
$(3, 3, 4)$	0.9941	0.9975	0.4969	0.0368	0.9963	0.9941	0.9864	1.0000	0.9009
$(2, 2, 6)$	0.9966	0.9977	0.4401	0.0878	0.9976	0.9696	0.9734	0.9998	0.8148
$(4, 5, 1)$	0.9904	0.9957	0.6717	0.0149	0.9969	1.0000	0.9931	1.0000	0.9885
$(3, 4, 3)$	0.9925	0.9965	0.6660	0.0204	0.9991	0.9962	0.9944	1.0000	0.9199
$(2, 3, 5)$	0.9953	0.9979	0.6420	0.0476	0.9987	0.9929	0.9901	1.0000	0.9498
$(3, 5, 2)$	0.9900	0.9972	0.8523	0.0031	1.0000	0.9986	0.9959	1.0000	0.9485
$(3, 6, 1)$	0.9905	0.9971	0.8651	0.0078	1.0000	1.0000	0.9922	1.0000	0.9392
$(2, 4, 4)$	0.9921	0.9975	0.6975	0.0289	1.0000	0.9935	0.9892	1.0000	0.9320
$(2, 5, 3)$	0.9918	0.9979	0.7984	0.0096	0.9993	0.9988	0.9956	1.0000	0.9682
$(2, 6, 2)$	0.9891	0.9981	0.8793	0.0207	0.9995	0.9921	0.9977	1.0000	0.9295

Table 7.3: Results on MNIST and Chars74K datasets with networks trained with known and known unknown classes of MNIST and known unknown classes of Chars74K. Each line represents the mean of 10 experiments with randomly selected $(|K|, |K_u|, |U|)$ digits. Lines are sorted by $|K_u|/|K|$. Red background indicates low accuracy (for visualization purposes).

$(\|K\|, \|K_u\|, \|U\|)$	K	K_u	U	K^d	K_u^d	U^d	U^l	R	R^p
$(6, 2, 2)$	0.9934	0.9945	0.1488	0.8304	0.8605	0.2209	0.3200	0.3650	0.2725
$(5, 2, 3)$	0.9932	0.9949	0.3416	0.7904	0.9336	0.3870	0.3308	0.5529	0.6060
$(6, 3, 1)$	0.9920	0.9951	0.4300	0.7790	0.8930	0.5907	0.4709	0.5352	0.5547
$(4, 2, 4)$	0.9951	0.9953	0.3033	0.8114	0.9005	0.3902	0.4309	0.6351	0.5495
$(5, 3, 2)$	0.9926	0.9952	0.5132	0.7906	0.8311	0.4568	0.4216	0.6941	0.6415
$(3, 2, 5)$	0.9955	0.9967	0.4226	0.7847	0.9322	0.4538	0.4524	0.4971	0.6703
$(4, 3, 3)$	0.9928	0.9962	0.4844	0.7793	0.9161	0.5472	0.5396	0.6084	0.6046
$(5, 4, 1)$	0.9902	0.9959	0.5535	0.7398	0.9407	0.5863	0.6001	0.6718	0.5940
$(4, 4, 2)$	0.9912	0.9951	0.5893	0.7537	0.9572	0.7776	0.6478	0.9108	0.8865
$(3, 3, 4)$	0.9943	0.9971	0.5448	0.7636	0.9454	0.6908	0.7016	0.7799	0.7172
$(2, 2, 6)$	0.9954	0.9985	0.4696	0.7259	0.9274	0.6147	0.6470	0.6554	0.7442
$(4, 5, 1)$	0.9908	0.9963	0.6861	0.7277	0.9442	0.7340	0.7754	0.7895	0.9336
$(3, 4, 3)$	0.9923	0.9966	0.6781	0.7028	0.9781	0.7985	0.7679	0.8322	0.7294
$(2, 3, 5)$	0.9953	0.9983	0.6558	0.7349	0.9587	0.6896	0.7420	0.8659	0.8490
$(3, 5, 2)$	0.9894	0.9965	0.8215	0.7128	0.9500	0.7409	0.7119	0.7555	0.7519
$(3, 6, 1)$	0.9894	0.9974	0.8578	0.7343	0.9784	0.7249	0.7389	0.9508	0.8227
$(2, 4, 4)$	0.9936	0.9977	0.7098	0.7932	0.9599	0.7800	0.7537	0.9559	0.8653
$(2, 5, 3)$	0.9899	0.9985	0.8351	0.7024	0.9736	0.8561	0.8250	0.9383	0.9300
$(2, 6, 2)$	0.9895	0.9984	0.8736	0.6801	0.9767	0.8538	0.8918	0.9622	0.8442

Table 7.4: Results on MNIST and Chars74K datasets with networks trained with known and known unknown classes of MNIST and known and known unknown classes of Chars74K. Each line represents the mean of 10 experiments with randomly selected $(|K|, |K_u|, |U|)$ digits. Lines are sorted by $|K_u|/|K|$. Red background indicates low accuracy (for visualization purposes).

Chapter 8

Conclusions and future work

The main hypothesis we have carried out along this work is that being able to bound the known-labeled open space (KLOS) is essential for properly handling the open-set recognition problem. The methods we have proposed—the Open-Set Nearest Neighbors (OSNN) and the Specialized Support Vector Machines (SSVM)—both are capable of keeping the KLOS bounded in the feature space of the description, and their superior performance in most of the experiments highlight this requirement. The properties of Support Vector Machines (SVM) along with Radial Basis Function (RBF) kernel have allowed us to better assess this factor to prove the requirement of a bounded KLOS for open-set recognition. Furthermore, we have shown the effectiveness of employing the—now formalized—open-set grid search as a general grid search strategy, which can be applied to any parametric classifier that has the ability to reject unknown samples. Finally, we have enlarged the set of options for evaluation measures specially targeted at assessing accuracy in an open-set setup.

Some baselines from the literature have shown competitive results with our proposed methods—a special highlight for Support Vector Machines with Probability of Inclusion (PISVM)—and even the straightforward SVM, when properly configured with one-vs-all strategy, obtains reasonable results. However, most of those methods have no theoretical guaranty of being able to bound the risk of the unknown by bounding the KLOS. In fact, we have shown throughout our experiments that in certain cases, those methods leave an unbounded KLOS, which might be an undesirable characteristic for certain critical and sensitive applications. For instance, consider a forensic scenario in which suspects shall be judged for certain crimes and experts should employ a recognition method for acquiring evidence for the verdict: as they are simply suspects, we would expect a recognition method to avoid obtaining positive and highly-confident outputs on the open-space, i.e., to avoid being highly confident of its correctness when it incorrectly predicts that one of the suspects has committed the crime. It implies that a bounded KLOS should be ensured, otherwise the behavior of the method for instances from unknown classes—a.k.a. the suspects, if they have not committed the crime *de facto*—would be unexpected.

Some particularities should be taken into account when dealing with open-set scenarios in order to facilitate handling the problem and avoiding mistakes. We have shown, for instance, that the one-vs-all as a multiclass-from-binary strategy is suitable for open-set scenarios as it allows the straightforward employment of binary classifiers that only need

to satisfy simpler properties, e.g., bounding the positively-labeled open space (PLOS). This is required for bounding the KLOS and decrease the risk of the unknown. The same is true with the open-set grid search we have formalized in this work, which allows a general employment and was shown to improve performance of multiple recognition methods.

The lack of research on open-set recognition until recent years makes us consider that, when required to handle the problem, trivial approaches—as the ones described along this work—can be unduly employed in real world, as they seem reasonable at first glance. However, appearances deceive, and we have shown, for instance, that thresholding SVM's probabilities aiming at identifying unknown samples need to be employed with caution to avoid unexpected behavior in an open-set setup. The same concern applies to the straightforward approach of thresholding softmax probabilities on neural networks: although it seems reasonable to reject not-so-confident classifications, only a small portion of the open space is in fact handled.

Due to the properties obtained along the development of SSVM, we have also touched some particularities of SVM without bias term and probability estimates for SVM with one-vs-one strategy. While SVM without bias term is a method simpler to extend to the open-set recognition setup—by introducing what we have named the *artificial bias term*—it seems that the requirement of a minimum threshold on probability estimate has been ignored in previous work. We have shown that a minimal required threshold can be calculated for SVM with one-vs-one approach such that, when applied to SVM's probability, it will ensure a bounded KLOS. Future research can be accomplished not only on the search for optimal rejection threshold but also on proper ways of estimating probabilities for open-set scenarios. For instance, consider that Platt's probability for a binary problem, as is, can obtain smaller probability for a positive training sample than for the open space. Then, by simply employing the minimum threshold to ensure a bounded PLOS would incorrectly reject positive samples. It shows that Platt's probability estimate does not consider the open-set scenario. A proper probability estimate for the open-set scenario, for individual binary problems, needs to ensure higher probability for the positive instances than for the open space.

Regarding SVM for open-set scenario, all variants we have evaluated in this work consider the traditional binary version—the only formalization known until the works of Weston and Watkins [1998] and Crammer and Singer [2001], who have extended SVM for multiclass classification by means of a single optimization problem. In this work, we have not considered the employment of inherently multiclass SVMs, as their performance over traditional multiclass-from-binary extensions has not been evinced [Hsu and Lin, 2002, Rifkin and Klautau, 2004, Mathur and Foody, 2008]. Furthermore, starting the research from the binary formulation was the natural path of investigation that we could employ. We highlight, however, the promising research topic of open-set recognition along with those multiclass formulations.

OSNN has shown promising results for open set by relying on ratio of distances. The current implementation only employs the two nearest classes and its simplicity can be a plus in many scenarios. However, all other trained classes are neglected but might hold important information for better decisions. OSNN suffers from the problem of rejecting

known instances that appear on the overlapping region of two or more training classes instead of classifying them as one of the doubtful classes. This undesirable behavior might be overcome by employing extra ratios of distances to other known classes for the final decision. It is not a trivial extension to accomplish but worth investigating in future work. More elaborated techniques, as the meta-recognition proposed by Scheirer et al. [2012], can be employed to avoid dealing with multiple thresholds and better performing the final decision.

We have shown multiple nuances of neural networks when considering it for recognition in open-set scenarios. It is not enough to verify for lower confidence scores to properly identify unknown samples. Furthermore, employing a huge amount of known unknown data is not feasible—as all the universe of the unknown cannot be represented—and that does not tackle the problem in its root. We have touched the open-set problem along with neural networks aiming at gaining intuition on how to solve it and much can be explored in future work. For instance, we have observed the consequence of the linear behavior of the learned decision function, which shares some conclusions with works on adversarial images [Goodfellow et al., 2015], and it defines an intersection of research areas already evinced in previous work [Bendale and Boult, 2016] that is worth investigating.

Finally, in Chapter 7, we have saw that a Multilayer Perceptron employed with openmax layer is able to bound the KLOS of the input space, however, it is not guaranteed to be true for every case, as it depends on the shape of the dataset. Future work is worth investigating on finding out the properties of a neural network that might define bounded/unbounded KLOS at the feature space of some of the network's layers. Furthermore, analyses similar to the ones we have presented before should be accomplished for Convolutional Neural Network as well, as this model have been highly employed nowadays and its behavior with inputs from unknown classes have not received dedicated studies. Neural networks have been receiving attention mainly in the point of view of closed-set scenarios and still the possibility of the unknown have been ignored in many of the works that claim state-of-the-art results in classification problems. And we have shown that the straightforward approach of thresholding softmax probabilities is theoretically unreasonable, and training with known unknown instances is an insufficient alternative. It is an open field of research on how to make network methods optimizing the open-space risk besides the empirical risk, taking advantage of its data-driven characteristic and taking into account the unknown.

Bibliography

Michael Bain. *Learning Logical Exceptions in Chess*. PhD thesis, University of Strathclyde, 1994. URL https://tinyurl.com/Bain1994. 47

Peter L. Bartlett and Marten H. Wegkamp. Classification with a reject option using a hinge loss. *Journal of Machine Learning Research*, 9:1823–1840, August 2008. ISSN 1532-4435/1533-7928. URL http://www.jmlr.org/papers/v9/bartlett08a.html. 26

Daniele Battaglino, Ludovick Lepauloux, and Nicholas Evans. The open-set problem in acoustic scene classification. In *IEEE Intl. Workshop on Acoustic Signal Enhancement*, pages 1–5, Xi'an, China, September 2016. DOI 10.1109/IWAENC.2016.7602939. 25

Belhassen Bayar and Matthew C. Stamm. Towards open set camera model identification using a deep learning framework. In *IEEE Intl. Conference on Acoustics, Speech, and Signal Processing*, pages 2007–2011, Calgary, Alberta, Canada, April 2018. DOI 10.1109/ICASSP.2018.8462383. 25

Abhijit Bendale and Terrance E. Boult. Towards open world recognition. In *IEEE Intl. Conference on Computer Vision and Pattern Recognition*, pages 1893–1902, Boston, MA, USA, June 2015. DOI 10.1109/CVPR.2015.7298799. 29, 84

Abhijit Bendale and Terrance E. Boult. Towards open set deep networks. In *IEEE Intl. Conference on Computer Vision and Pattern Recognition*, pages 1563–1572, Las Vegas, NV, USA, June 2016. DOI 10.1109/CVPR.2016.173. 66, 79, 81, 83, 101

Christopher M. Bishop. *Pattern Recognition and Machine Learning*. Information Science and Statistics. Springer-Verlag New York, 1st edition, 2006. URL https://www.springer.com/us/book/9780387310732. 30, 31, 79, 117

Bernhard E. Boser, Isabelle M. Guyon, and Vladimir N. Vapnik. A training algorithm for optimal margin classifiers. In *ACM Annual Workshop on Computational Learning Theory*, pages 144–152, Pittsburgh, PA, USA, July 1992. DOI 10.1145/130385.130401. 19, 26, 35

Y-Lan Boureau, Francis Bach, Yann LeCun, and Jean Ponce. Learning mid-level features for recognition. In *IEEE Intl. Conference on Computer Vision and Pattern Recognition*, pages 2559–2566, San Francisco, CA, USA, June 2010. DOI 10.1109/CVPR.2010.5539963. 46, 47

John S. Bridle. Probabilistic interpretation of feedforward classification network outputs, with relationships to statistical pattern recognition. In Françoise Fogelman Soulié and Jeanny Hérault, editors, *Neurocomputing: Algorithms, Architectures and Applications*, volume 68 of *NATO ASI Series*, pages 227–236. Springer, Berlin, Heidelberg, 1990. DOI 10.1007/978-3-642-76153-9_28. 80

Martin D. Buhmann. *Radial Basis Functions: Theory and Implementations*. Cambridge Monographs on Applied and Computational Mathematics. Cambridge University Press, 1st edition, July 2003. DOI 10.1017/CBO9780511543241. 36

Pau Panareda Busto and Juergen Gall. Open set domain adaptation. In *IEEE Intl. Conference on Computer Vision*, pages 754–763, Venice, Italy, October 2017. DOI 10.1109/ICCV.2017.88. 25

Hakan Cevikalp and Bill Triggs. Efficient object detection using cascades of nearest convex model classifiers. In *IEEE Intl. Conference on Computer Vision and Pattern Recognition*, pages 3138–3145, Providence, RI, USA, June 2012. DOI 10.1109/CVPR.2012.6248047. 26

Chih-Chung Chang and Chih-Jen Lin. LIBSVM: A library for Support Vector Machines. *ACM Transactions on Intelligent Systems and Technology*, 2(3):27:1–27:27, April 2011. ISSN 2157-6904. DOI 10.1145/1961189.1961199. 20, 21, 22, 25, 36

Wei-Cheng Chang, Ching-Pei Lee, and Chih-Jen Lin. A revisit to Support Vector Data Description. Technical report, National Taiwan University of Science and Technology, Taipei, Taiwan, 2013. URL https://tinyurl.com/Chang2013. 19

Pai-Hsuen Chen, Chih-Jen Lin, and Bernhard Schölkopf. A tutorial on nu-Support Vector Machines. *Applied Stochastic Models in Business and Industry*, 21(2):111–136, March 2005. ISSN 1526-4025. DOI 10.1002/asmb.537. 22

Yunqiang Chen, Xiang Sean Zhou, and T. S. Huang. One-class SVM for learning in image retrieval. In *IEEE Intl. Conference on Image Processing*, volume 1, pages 34–37, Thessaloniki, Greece, October 2001. DOI 10.1109/ICIP.2001.958946. 26

Sien W. Chew, Simon Lucey, Patrick Lucey, Sridha. Sridharan, and Jeffrey F. Cohn. Improved facial expression recognition via uni-hyperplane classification. In *IEEE Intl. Conference on Computer Vision and Pattern Recognition*, pages 2554–2561, Providence, RI, USA, June 2012. DOI 10.1109/CVPR.2012.6247973. 26

C. K. Chow. On optimum recognition error and reject tradeoff. *IEEE Transactions on Information Theory*, 16(1):41–46, January 1970. ISSN 0018-9448/1557-9654. DOI 10.1109/TIT.1970.1054406. 26, 27

Kai-Min Chung, Wei-Chun Kao, Tony Sun, and Chih-Jen Lin. Decomposition methods for linear Support Vector Machines. In *IEEE Intl. Conference on Acoustics, Speech, and Signal Processing*, volume 4, pages IV–868–IV–871, Hong Kong, China, April 2003. DOI 10.1109/ICASSP.2003.1202781. 22

Stuart Coles. *An Introduction to Statistical Modeling of Extreme Values*. Springer Series in Statistics. Springer, London, 1st edition, 2001. DOI 10.1007/978-1-4471-3675-0. 28, 81

Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Springer Machine Learning*, 20(3):273–297, September 1995. ISSN 0885-6125/1573-0565. DOI 10.1007/BF00994018. 18, 34

Filipe de Oliveira Costa, Michael Eckmann, Walter J. Scheirer, and Anderson Rocha. Open set source camera attribution. In *Conference on Graphics, Patterns, and Images*, pages 71–78, Ouro Preto, MG, Brazil, August 2012. IEEE Press. DOI 10.1109/SIBGRAPI.2012.19. 25, 27

Filipe de Oliveira Costa, Ewerton Silva, Michael Eckmann, Walter J. Scheirer, and Anderson Rocha. Open set source camera attribution and device linking. *Elsevier Pattern Recognition Letters*, 39:92–101, April 2014. ISSN 0167-8655. DOI 10.1016/j.patrec.2013.09.006. 25, 27

Koby Crammer and Yoram Singer. On the algorithmic implementation of multiclass kernel-based vector machines. *Journal of Machine Learning Research*, 2:265–292, December 2001. ISSN 1532-4435/1533-7928. URL http://www.jmlr.org/papers/v2/crammer01a.html. 100

Steve Cruz, Cora Coleman, Ethan M. Rudd, and Terrance E. Boult. Open set intrusion recognition for fine-grained attack categorization. In *IEEE Intl. Symposium on Technologies for Homeland Security*, pages 1–6, Waltham, MA, USA, April 2017. DOI 10.1109/THS.2017.7943467. 25

Teófilo E. de Campos, Bodla Rakesh Babu, and Manik Varma. Character recognition in natural images. In *Intl. Conference on Computer Vision Theory and Applications*, pages 1–8, Lisbon, Portugal, February 2009. URL https://tinyurl.com/deCampos2009. 85

Laurens de Haan and Ana Ferreira. *Extreme Value Theory: An Introduction*. Springer Series in Operations Research and Financial Engineering. Springer, New York, NY, 1st edition, 2007. DOI 10.1007/0-387-34471-3. 28

Janez Demšar. Statistical comparisons of classifiers over multiple data sets. *Journal of Machine Learning Research*, 7:1–30, January 2006. ISSN 1532-4435/1533-7928. URL http://www.jmlr.org/papers/v7/demsar06a.html. 46

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A large-scale hierarchical image database. In *IEEE Intl. Conference on Computer Vision and Pattern Recognition*, pages 248–255, Miami, FL, USA, June 2009. DOI 10.1109/CVPR.2009.5206848. 66

Jiuqing Deng and Qixiu Hu. Open set text-independent speaker recognition based on set-score pattern classification. In *IEEE Intl. Conference on Acoustics, Speech, and Signal Processing*, pages II–73–II–76, Hong Kong, China, April 2003. DOI 10.1109/ICASSP.2003.1202297. 25

Hanze Dong, Yanwei Fu, Leonid Sigal, Sung Ju Hwang, Yu-Gang Jiang, and Xiangyang Xue. Learning to separate domains in generalized zero-shot and open set learning: A probabilistic perspective. In *Intl. Conference on Learning Representations*, pages 1–13, New Orleans, LA, USA, May 2019. URL https://arxiv.org/abs/1810.07368. 25

Cassio Elias dos Santos Junior and William Robson Schwartz. Extending face identification to open-set face recognition. In *Conference on Graphics, Patterns, and Images*, pages 188–195, Rio de Janeiro, RJ, Brazil, August 2014. IEEE Press. DOI 10.1109/SIBGRAPI.2014.23. 25

Bernard Dubuisson and Mylène Masson. A statistical decision rule with incomplete knowledge about classes. *Elsevier Pattern Recognition*, 26(1):155–165, January 1993. ISSN 0031-3203. DOI 10.1016/0031-3203(93)90097-G. 27

Rong-En Fan, Pai-Hsuen Chen, and Chih-Jen Lin. Working Set Selection using second order information for training Support Vector Machines. *Journal of Machine Learning Research*, 6:1889–1918, December 2005. ISSN 1532-4435/1533-7928. URL http://www.jmlr.org/papers/v6/fan05a.html. 38

Peter W. Frey and David J. Slate. Letter recognition using Holland-style adaptive classifiers. *Springer Machine Learning*, 6(2):161–182, March 1991. ISSN 0885-6125/1573-0565. DOI 10.1007/BF00114162. 47

Keinosuke Fukunaga. Hypothesis testing. In *Introduction to Statistical Pattern Recognition*, chapter 3, pages 51–123. Academic Press, 2nd edition, October 1990. DOI 10.1016/C2009-0-27872-X. 22, 26, 48

Chao Gao, Guruprasad Saikumar, Amit Srivastava, and Premkumar Natarajan. Open-set speaker identification in broadcast news. In *IEEE Intl. Conference on Acoustics, Speech, and Signal Processing*, pages 5280–5283, Prague, Czech Republic, May 2011. DOI 10.1109/ICASSP.2011.5947549. 25

Jan-Mark Geusebroek, Gertjan J. Burghouts, and Arnold W. M. Smeulders. The Amsterdam library of object images. *Springer Intl. Journal of Computer Vision*, 61(1):103–112, January 2005. ISSN 0920-5691/1573-1405. DOI 10.1023/B:VISI.0000042993.50813.60. 47

Xavier Glorot, Antoine Bordes, and Yoshua Bengio. Deep sparse rectifier neural networks. In Geoffrey Gordon, David Dunson, and Miroslav Dudík, editors, *Intl. Conference on Artificial Intelligence and Statistics*, volume 15, pages 315–323, Fort Lauderdale, FL, USA, April 2011. Proceedings of Machine Learning Research. URL http://proceedings.mlr.press/v15/glorot11a.html. 82

Yifan Gong. Noise-robust open-set speaker recognition using noise-dependent Gaussian Mixture classifier. In *IEEE Intl. Conference on Acoustics, Speech, and Signal Processing*, pages I–133–I–136, Orlando, FL, USA, May 2002. DOI 10.1109/ICASSP.2002.5743672. 25

Ian Goodfellow, Jon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In *Intl. Conference on Learning Representations*, pages 1–11, San Juan, Puerto Rico, May 2015. URL https://arxiv.org/abs/1412.6572. 79, 83, 101

Alex Graves. *Supervised Sequence Labelling with Recurrent Neural Networks*, volume 385 of *Studies in Computational Intelligence*. Springer-Verlag Berlin Heidelberg, 1st edition, 2012. DOI 10.1007/978-3-642-24797-2. 80, 81

Greg Griffin, Alex Holub, and Pietro Perona. Caltech-256 object category dataset. Technical report, California Institute of Technology, May 2007. URL http://resolver.caltech.edu/CaltechAUTHORS:CNS-TR-2007-001. 47

Fatma Güney, Nuri Murat Arar, and Hazım Kemal Ekenel. Open-set face recognition system. In *Signal Processing and Communications Applications Conference*, pages 1–2, Mugla, Turkey, April 2012. IEEE Press. DOI 10.1109/SIU.2012.6204819. 25

M. Günther, P. Hu, C. Herrmann, C. H. Chan, M. Jiang, S. Yang, A. R. Dhamija, D. Ramanan, J. Beyerer, J. Kittler, M. Al Jazaery, M. I. Nouyed, G. Guo, C. Stankiewicz, and T. E. Boult. Unconstrained face detection and open-set face recognition challenge. In *IEEE Intl. Joint Conference on Biometrics*, pages 697–706, Denver, CO, USA, October 2017. DOI 10.1109/BTAS.2017.8272759. 25

Zhongkai Han, Chi Fang, and Xiaoqing Ding. Discriminative prototype learning in open set face recognition. In *Intl. Conference on Pattern Recognition*, pages 2696–2699, Istanbul, Turkey, August 2010. IEEE Press. DOI 10.1109/ICPR.2010.661. 25

Brian Heflin, Walter J. Scheirer, and Terrance E. Boult. Detecting and classifying scars, marks, and tattoos found in the wild. In *IEEE Intl. Conference on Biometrics: Theory, Applications and Systems*, pages 31–38, Arlington, VA, USA, September 2012. DOI 10.1109/BTAS.2012.6374555. 25, 27

Kurt Hornik, Maxwell Stinchcombe, and Halbert White. Multilayer feedforward networks are universal approximators. *Elsevier Neural Networks*, 2(5):359–366, March 1989. ISSN 0893-6080. DOI 10.1016/0893-6080(89)90020-8. 80

Chih-Wei Hsu and Chih-Jen Lin. A comparison of methods for multiclass Support Vector Machines. *IEEE Transactions on Neural Networks*, 13(2):415–425, March 2002. ISSN 1045-9227/1941-0093. DOI 10.1109/72.991427. 21, 100

Lalit P. Jain, Walter J. Scheirer, and Terrance E. Boult. Multi-class open set recognition using probability of inclusion. In David Fleet, Tomas Pajdla, Bernt Schiele, and Tinne Tuytelaars, editors, *European Conference on Computer Vision*, volume 8691, part III of *Lecture Notes in Computer Science*, pages 393–409, Zurich, Switzerland, September 2014. Springer, Cham. DOI 10.1007/978-3-319-10578-9_26. 22, 25, 28

Jayadeva, R. Khemchandani, and Suresh Chandra. Twin Support Vector Machines for pattern classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(5):905–910, May 2007. ISSN 0162-8828/2160-9292/1939-3539. DOI 10.1109/TPAMI.2007.1068. 26

Hongliang Jin, Qingshan Liu, and Hanqing Lu. Face detection using one-class-based support vectors. In *IEEE Intl. Conference on Automatic Face and Gesture Recognition*, pages 457–462, Seoul, Korea, May 2004. DOI 10.1109/AFGR.2004.1301575. 26

Mohammed Waleed Kadous. *Temporal classification: Extending the classification paradigm to multivariate time series*. PhD thesis, The University of New South Wales, New South Wales, Australia, October 2002. URL https://dl.acm.org/citation.cfm?id=1037668. 47

Wei-Chun Kao, Kai-Min Chung, Chia-Liang Sun, and Chih-Jen Lin. Decomposition methods for linear support vector machines. *Neural Computation*, 16(8):1689–1704, August 2004. ISSN 0899-7667/1530-888X. DOI 10.1162/089976604774201640. 22

Vojislav Kecman, Te Ming Huang, and Michael Vogt. Iterative single data algorithm for training kernel machines from huge data sets: Theory and performance. In Lipo Wang, editor, *Support Vector Machines: Theory and Applications*, volume 177 of *Studies in Fuzziness and Soft Computing*, pages 255–274. Springer, Berlin, Heidelberg, April 2005. DOI 10.1007/10984697_12. 40

Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images. Master's thesis, University of Toronto, April 2009. URL https://tinyurl.com/Krizhevsky2009. 66

Amioy Kumar and Ajay Kumar. Adaptive security for human surveillance using multimodal open set biometric recognition. In *Intl. Conference on Pattern Recognition*, pages 405–410, Stockholm, Sweden, August 2014. IEEE Press. DOI 10.1109/ICPR.2014.78. 25

Ludmila I. Kuncheva and Stefan T. Hadjitodorov. Using diversity in cluster ensembles. In *IEEE Intl. Conference on Systems, Man, and Cybernetics*, volume 2, pages 1214–1219, The Hague, Netherlands, October 2004. DOI 10.1109/ICSMC.2004.1399790. 33, 54, 69, 82

Svetlana Lazebnik, Cordelia Schmid, and Jean Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *IEEE Intl. Conference on Computer Vision and Pattern Recognition*, volume 2, pages 2169–2178, New York, NY, USA, June 2006. DOI 10.1109/CVPR.2006.68. 46

Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, November 1998. ISSN 0018-9219/1558-2256. DOI 10.1109/5.726791. 66

Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521(7553): 436–444, May 2015. ISSN 0028-0836/1476-4687. DOI 10.1038/nature14539. 79, 81

Fayin Li and H. Wechsler. Open set face recognition using transduction. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(11):1686–1697, November 2005. ISSN 0162-8828/2160-9292/1939-3539. DOI 10.1109/TPAMI.2005.224. 25

Lisha Li, Kevin Jamieson, Giulia DeSalvo, Afshin Rostamizadeh, and Ameet Talwalkar. Hyperband: A novel bandit-based approach to hyperparameter optimization. *Journal of Machine Learning Research*, 18(185):1–52, April 2018. ISSN 1532-4435/1533-7928. URL http://www.jmlr.org/papers/v18/16-558.html. 23

Zhizheng Liang, Lei Zhang, Jin Liu, and Yong Zhou. Adaptively weighted learning for twin support vector machines via Bregman divergences. *Springer Neural Computing and Applications*, pages 1–14, November 2018. ISSN 0941-0643/1433-3058. DOI 10.1007/s00521-018-3843-0. 25

Hsuan-Tien Lin, Chih-Jen Lin, and Ruby C. Weng. A note on platt's probabilistic outputs for Support Vector Machines. *Springer Machine Learning*, 68(3):267–276, October 2007. ISSN 0885-6125/1573-0565. DOI 10.1007/s10994-007-5018-6. 42, 43

David Lowe. Distinctive image features from scale-invariant keypoints. *Springer Intl. Journal of Computer Vision*, 60(2):91–110, November 2004. ISSN 0920-5691/1573-1405. DOI 10.1023/B:VISI.0000029664.99615.94. 46

Tomasz Malisiewicz, Abhinav Gupta, and Alexei A. Efros. Ensemble of Exemplar-SVMs for object detection and beyond. In *IEEE Intl. Conference on Computer Vision*, pages 89–96, Barcelona, Spain, November 2011. DOI 10.1109/ICCV.2011.6126229. 26

Larry M. Manevitz and Malik Yousef. One-class SVMs for document classification. *Journal of Machine Learning Research*, 2:139–154, December 2001. ISSN 1532-4435/1533-7928. URL http://www.jmlr.org/papers/v2/manevitz01a.html. 26

A. Mathur and G. M. Foody. Multiclass and binary SVM classification: Implications for training and classification users. *IEEE Geoscience and Remote Sensing Letters*, 5(2): 241–245, April 2008. ISSN 1545-598X/1558-0571. DOI 10.1109/LGRS.2008.915597. 100

Pedro Ribeiro Mendes Júnior. Open-Set Optimum-Path Forest classifier. Master's thesis, Institute of Computing, University of Campinas, Campinas, SP, Brazil, August 2014. URL http://repositorio.unicamp.br/jspui/handle/REPOSIP/275530. 20, 22, 23, 29, 30, 47, 48

Pedro Ribeiro Mendes Júnior, Roberto Medeiros de Souza, Rafael de Oliveira Werneck, Bernardo Vecchia Stein, Daniel Vatanabe Pazinato, Waldir Rodrigues de Almeida, Otávio Augusto Bizetto Penatti, Ricardo da Silva Torres, and Anderson de Rezende Rocha. Nearest neighbors distance ratio open-set classifier. *Springer Machine Learning*, 106(3):359–386, March 2017. ISSN 0885-6125/1573-0565. DOI 10.1007/s10994-016-5610-8. 20

Pedro Ribeiro Mendes Júnior, Terrance E. Boult, Jacques Wainer, and Anderson de Rezende Rocha. Specialized Support Vector Machines for open-set recognition. Under review. Preprint available, October 2018. URL http://arxiv.org/abs/1606.03802. 20

James Mercer. Functions of positive and negative type, and their connection the theory of integral equations. *Philosophical Transactions of the Royal Society A*, 209(441–458): 415–446, January 1909. ISSN 1364-503X/1471-2962. DOI 10.1098/rsta.1909.0016. 35

Donald Michie, David J. Spiegelhalter, and Charles C. Taylor. *Machine Learning, Neural and Statistical Classification*. Ellis Horwood Series in Artificial Intelligence. Prentice Hall, Upper Saddle River, NJ, USA, July 1994. URL http://www1.maths.leeds.ac.uk/~charles/statlog. 47

Ali Moeini, Karim Faez, Hossein Moeini, and Armon Matthew Safai. Open-set face recognition across look-alike faces in real-world scenarios. *Image and Vision Computing*, 57:1–14, January 2017. ISSN 0262-8856. DOI 10.1016/j.imavis.2016.11.002. 25

Russell Muzzolini, Yee-Hong Yang, and Roger Pierson. Classifier design with incomplete knowledge. *Elsevier Pattern Recognition*, 31(4):345–369, April 1998. ISSN 0031-3203. DOI 10.1016/S0031-3203(97)00056-3. 22, 27, 48

Manuel Alberto Córdova Neira, Pedro Ribeiro Mendes Júnior, Anderson Rocha, and Ricardo da Silva Torres. Data-fusion techniques for open-set recognition problems. *IEEE Access*, 6:21242–21265, April 2018. ISSN 2169-3536. DOI 10.1109/ACCESS.2018.2824240. 20, 25, 29, 48

A. Nguyen, J. Yosinski, and J. Clune. Deep neural networks are easily fooled: High confidence predictions for unrecognizable images. In *IEEE Intl. Conference on Computer Vision and Pattern Recognition*, volume 1, pages 427–436, Boston, MA, USA, June 2015. DOI 10.1109/CVPR.2015.7298640. 79, 83

Michael Nielsen. A visual proof that neural nets can compute any function. In *Neural Networks and Deep Learning*, chapter 4. Determination Press, 2015. URL http://neuralnetworksanddeeplearning.com/chap4.html. 80

Randal S. Olson, William La Cava, Patryk Orzechowski, Ryan J. Urbanowicz, and Jason H. Moore. PMLB: A large benchmark suite for machine learning evaluation and comparison. *BioData Mining*, 10(36):1–13, December 2017. ISSN 1756-0381. DOI 10.1186/s13040-017-0154-4. 46

João Paulo Papa, Alexandre Xavier Falcão, Paulo A. V. Miranda, Celso T. N. Suzuki, and Nelson D. A. Mascarenhas. Design of robust pattern classifiers based on Optimum-Path Forests. In *Intl. Symposium on Mathematical Morphology*, volume 1, pages 337–348, Rio de Janeiro, RJ, Brazil, October 2007. MCT/INPE. URL http://urlib.net/dpi.inpe.br/ismm@80/2007/04.13.23.19. 29

João Paulo Papa, Alexandre Xavier Falcão, Victor Hugo C. de Albuquerque, and João Manuel R. S. Tavares. Efficient supervised optimum-path forest classification for large datasets. *Elsevier Pattern Recognition*, 45(1):512–520, January 2012. ISSN 0031-3203. DOI 10.1016/j.patcog.2011.07.013. 29

John C. Platt. Fast training of Support Vector Machines using Sequential Minimal Optimization. In Bernhard Schölkopf, Christopher J. C. Burges, and Alexander Johannes Smola, editors, *Advances in Kernel Methods: Support Vector Learning*, chapter 12, pages 185–208. MIT Press, 1st edition, December 1998. URL https://www.researchgate.net/publication/234786663. 38

John C. Platt. Probabilities for SV Machines. In Alexander Johannes Smola, Peter Bartlett, Bernhard Schölkopf, and Dale Schuurmans, editors, *Advances in Large-Margin Classifiers*, Neural Information Processing series, chapter 5, pages 61–74. MIT Press, September 2000. URL https://mitpress.mit.edu/books/advances-large-margin-classifiers. 21, 42, 43, 100

Dimitrios Pritsos and Efstathios Stamatatos. Open set evaluation of web genre identification. *Language Resources and Evaluation*, 52(4):949–968, December 2018. ISSN 1574-020X. DOI 10.1007/s10579-018-9418-y. 25

Dimitrios A. Pritsos and Efstathios Stamatatos. Open-set classification for automated genre identification. In Pavel Serdyukov, Pavel Braslavski, Sergei O. Kuznetsov, Jaap Kamps, and Stefan Rüger, editors, *European Conference on Information Retrieval*, volume 7814 of *Lecture Notes in Computer Science*, pages 207–217, Moscow, Russia, March 2013. Springer, Berlin, Heidelberg. DOI 10.1007/978-3-642-36973-5_18. 25, 27

Ajita Rattani, Walter J. Scheirer, and Arun Ross. Open set fingerprint spoof detection across novel fabrication materials. *IEEE Transactions on Information Forensics and Security*, 10(11):2447–2460, November 2015. ISSN 1556-6013/1556-6021. DOI 10.1109/TIFS.2015.2464772. 25

Ryan Rifkin and Aldebaro Klautau. In defense of one-vs-all classification. *Journal of Machine Learning Research*, 5:101–141, January 2004. ISSN 1532-4435/1533-7928. URL http://www.jmlr.org/papers/v5/rifkin04a.html. 100

Anderson Rocha and Siome Goldenstein. Multi-class from binary: Divide to conquer. In *Intl. Conference on Computer Vision Theory and Applications*, pages 1–8, Lisbon, Portugal, February 2009. URL https://tinyurl.com/Rocha2009. 21

Anderson Rocha and Siome Goldenstein. Multiclass from binary: Expanding one-vs-all, one-vs-one and ECOC-based approaches. *IEEE Transactions on Neural Networks and Learning Systems*, 25(2):289–302, February 2014. ISSN 2162-237X/2162-2388. DOI 10.1109/TNNLS.2013.2274735. 19, 21, 26

David A. Ross, Jongwoo Lim, Ruei-Sung Lin, and Ming-Hsuan Yang. Incremental learning for robust visual tracking. *Springer Intl. Journal of Computer Vision*, 77(1–3):125–141, May 2008. ISSN 0920-5691/1573-1405. DOI 10.1007/s11263-007-0075-7. 29

Ethan M. Rudd, Lalit P. Jain, Walter J. Scheirer, and Terrance E. Boult. The Extreme Value Machine. *IEEE Transactions on Pattern Analysis and Machine In-*

telligence, 40(3):762–768, March 2018. ISSN 0162-8828/2160-9292/1939-3539. DOI 10.1109/TPAMI.2017.2707495. 25

Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. Imagenet large scale visual recognition challenge. *Springer Intl. Journal of Computer Vision*, 115(3):211–252, December 2015. ISSN 0920-5691/1573-1405. DOI 10.1007/s11263-015-0816-y. 66

Walter J. Scheirer. *Extreme Value Theory-Based Methods for Visual Recognition*. Synthesis Lectures on Computer Vision. Morgan & Claypool Publishers, 1st edition, February 2017. DOI 10.2200/S00756ED1V01Y201701COV010. 28

Walter J. Scheirer, Anderson de Rezende Rocha, Jonathan Parris, and Terrance E. Boult. Learning for meta-recognition. *IEEE Transactions on Information Forensics and Security*, 7(4):1214–1224, August 2012. ISSN 1556-6013/1556-6021. DOI 10.1109/TIFS.2012.2192430. 101

Walter J. Scheirer, Anderson de Rezende Rocha, Archana Sapkota, and Terrance E. Boult. Towards open set recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(7):1757–1772, July 2013. ISSN 0162-8828/2160-9292/1939-3539. DOI 10.1109/TPAMI.2012.256. 18, 25, 27, 28

Walter J. Scheirer, Lalit P. Jain, and Terrance E. Boult. Probability models for open set recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(11):2317–2324, November 2014. ISSN 0162-8828/2160-9292/1939-3539. DOI 10.1109/TPAMI.2014.2321392. 25, 28, 69

Matthew D. Scherreik and Brian D. Rigling. Open set recognition for automatic target classification with rejection. *IEEE Transactions on Aerospace and Electronic Systems*, 52(2):632–642, April 2016. ISSN 0018-9251/1557-9603. DOI 10.1109/TAES.2015.150027. 25

Bernhard Schölkopf and Alexander Johannes Smola. *Learning with Kernels*. Adaptive Computation and Machine Learning series. MIT Press, 1st edition, December 2001. URL https://mitpress.mit.edu/books/learning-kernels. 35

Bernhard Schölkopf, John C. Platt, John Shawe-Taylor, Alexander Johannes Smola, and Robert C. Williamson. Estimating the support of a high-dimensional distribution. *Neural Computation*, 13(7):1443–1471, July 2001. ISSN 0899-7667/1530-888X. DOI 10.1162/089976601750264965. 19, 25, 26

P. Sivakumaran, J. Fortuna, and Aladdin M. Ariyaeeinia. Score normalisation applied to open-set, text-independent speaker identification. In *European Conference on Speech Communication and Technology*, pages 2669–2672, Geneva, Switzerland, September 2003. URL https://www.isca-speech.org/archive/eurospeech_2003/e03_2669.html. 25

Marina Sokolova and Guy Lapalme. A systematic analysis of performance measures for classification tasks. *Information Processing & Management*, 45(4):427–437, July 2009. ISSN 0306-4573. DOI 10.1016/j.ipm.2009.03.002. 47

Francisco J. Solis and Roger J.-B. Wets. Minimization by random search techniques. *Mathematics of Operations Research*, 6(1):19–30, February 1981. ISSN 0364-765X/1526-5471. DOI 10.1287/moor.6.1.19. 23

César R. Souza. Kernel functions for machine learning applications, March 2010. URL http://crsouza.com/2010/03/17/kernel-functions-for-machine-learning-applications. Accessed: November 14, 2018. 36

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15:1929–1958, June 2014. ISSN 1532-4435/1533-7928. URL http://jmlr.org/papers/v15/srivastava14a.html. 84

Renato O. Stehling, Mario A. Nascimento, and Alexandre Xavier Falcão. A compact and efficient image retrieval approach based on border/interior pixel classification. In *ACM Intl. Conference on Information and Knowledge Management*, pages 102–109, McLean, VA, USA, November 2002. DOI 10.1145/584792.584812. 47

S. J. Stolfo, Wei Fan, Wenke Lee, A. Prodromidis, and P. K. Chan. Cost-based modeling for fraud and intrusion detection: Results from the JAM project. In *DARPA Information Survivability Conference and Exposition*, volume 2, pages 130–144, Hilton Head, SC, USA, January 2000. IEEE Press. DOI 10.1109/DISCEX.2000.821515. 47

Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian J. Goodfellow, and Rob Fergus. Intriguing properties of neural networks. In *Intl. Conference on Learning Representations*, pages 1–10, Banff, Canada, April 2014. URL https://arxiv.org/abs/1312.6199. 79

David M. J. Tax and Robert P. W. Duin. Support vector domain description. *Elsevier Pattern Recognition Letters*, 20(11):1191–1199, November 1999a. ISSN 0167-8655. DOI 10.1016/S0167-8655(99)00087-2. 19

David M. J. Tax and Robert P. W. Duin. Data domain description using support vectors. In *European Symposium on Artificial Neural Networks*, pages 251–256, Bruges, Belgium, April 1999b. URL https://www.elen.ucl.ac.be/esann/proceedings/papers.php?ann=1999. 26

David M. J. Tax and Robert P. W. Duin. Support vector data description. *Springer Machine Learning*, 54(1):45–66, January 2004. ISSN 0885-6125/1573-0565. DOI 10.1023/B:MACH.0000008084.60811.49. 19, 25

Tensorflow.org. Advanced Convolutional Neural Networks, October 2018a. URL https://www.tensorflow.org/tutorials/deep_cnn. Accessed: November 14, 2018. 66

Tensorflow.org. Deep MNIST for experts, May 2018b. URL http://docs.w3cub.com/tensorflow~guide/get_started/mnist/pros. Backup version. Accessed: November 14, 2018. 66, 84

Ye Tian, Zili Wang, Lipin Zhang, Chen Lu, and Jian Ma. A subspace learning-based feature fusion and open-set fault diagnosis approach for machinery components. *Advanced Engineering Informatics*, 36:194–206, April 2018. ISSN 1474-0346. DOI 10.1016/j.aei.2018.04.006. 25

Jan C. van Gemert, Cor J. Veenman, Arnold W. M. Smeulders, and Jan-Mark Geusebroek. Visual word ambiguity. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(7):1271–1283, July 2010. ISSN 0162-8828/2160-9292/1939-3539. DOI 10.1109/TPAMI.2009.132. 46, 47

Rafael Vareto, Samira Silva, Filipe Costa, and William Robson Schwartz. Towards open-set face recognition using hashing functions. In *IEEE Intl. Joint Conference on Biometrics*, pages 634–641, Denver, CO, USA, October 2017. DOI 10.1109/BTAS.2017.8272751. 25

Cor J. Veenman, Marcel J. T. Reinders, and Eric Backer. A maximum variance cluster algorithm. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(9):1273–1280, September 2002. ISSN 0162-8828/2160-9292/1939-3539. DOI 10.1109/TPAMI.2002.1033218. 69, 82

Michael Vogt. SMO algorithms for Support Vector Machines without bias term. Technical report, Institute of Automatic Control, Technische Universität Darmstadt, Darmstadt, Germany, July 2002. URL https://tinyurl.com/Vogt2002. 40

Hanxiao Wang, Xiatian Zhu, Tao Xiang, and Shaogang Gong. Towards unsupervised open-set person re-identification. In *IEEE Intl. Conference on Image Processing*, pages 769–773, Phoenix, AZ, USA, September 2016. DOI 10.1109/ICIP.2016.7532461. 25

Jason Weston and Chris Watkins. Multi-class support vector machines. Technical report, Royal Holloway, University of London, Egham, Surrey, England, May 1998. URL https://tinyurl.com/Weston1998. 100

Mingrui Wu and Jieping Ye. A small sphere and large margin approach for novelty detection using training data with outliers. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(11):2088–2092, November 2009. ISSN 0162-8828/2160-9292/1939-3539. DOI 10.1109/TPAMI.2009.24. 26

Ting-Fan Wu, Chih-Jen Lin, and Ruby C. Weng. Probability estimates for multi-class classification by pairwise coupling. *Journal of Machine Learning Research*, 5:975–1005, August 2004. ISSN 1532-4435/1533-7928. URL http://www.jmlr.org/papers/v5/wu04a.html. 21, 42, 43

Han Xiao, Jun Sun, Xiaoyi Yu, and Liuan Wang. Compact binary feature for open set recognition. In *IAPR Intl. Workshop on Document Analysis Systems*, pages 235–238, Vienna, Austria, April 2018. IEEE Press. DOI 10.1109/DAS.2018.81. 25

Hao Xie, Yunyan Du, Huapeng Yu, Yongxin Chang, Zhiyong Xu, and Yuanyan Tang. Open set face recognition with deep transfer learning and extreme value statistics. *Intl. Journal of Wavelets, Multiresolution and Information Processing*, 16(4):1–25, July 2018. ISSN 0219-6913/1793-690X. DOI 10.1142/S0219691318500340. 25

Bailing Zhang and Hong Hao. Open-set face recognition by transductive kernel associative memory. In *Intl. Congress on Image and Signal Processing*, pages 633–638, Dalian, China, October 2014. IEEE Press. DOI 10.1109/CISP.2014.7003856. 25

H. Zhang and V. M. Patel. Sparse representation-based open set recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(8):1690–1696, August 2017. ISSN 0162-8828/2160-9292/1939-3539. DOI 10.1109/TPAMI.2016.2613924. 25

Qian Zhang and John H. L. Hansen. Training candidate selection for effective rejection in open-set language identification. In *Spoken Language Technology Workshop*, pages 384–389, South Lake Tahoe, NV, USA, December 2014. IEEE Press. DOI 10.1109/SLT.2014.7078605. 25

Qian Zhang and John H. L. Hansen. Unsupervised k-means clustering based out-of-set candidate selection for robust open-set language recognition. In *Spoken Language Technology Workshop*, pages 324–329, San Diego, CA, USA, December 2016. IEEE Press. DOI 10.1109/SLT.2016.7846284. 25

Xuran Zhao, Nicholas Evans, and Jean-Luc Dugelay. Open-set semi-supervised audio-visual speaker recognition using co-training LDA and sparse representation classifiers. In *IEEE Intl. Conference on Acoustics, Speech, and Signal Processing*, pages 2999–3003, Vancouver, BC, Canada, May 2013. DOI 10.1109/ICASSP.2013.6638208. 25

Appendix A

Complete Specialized Support Vector Machines formulation

In this appendix, we present the complete formulation of SSVM, i.e., the details regarding the derivation of the dual problem from the primal one.

The optimization problem for the SSVM classifier is defined as

$$\begin{aligned} \min_{\mathbf{w},b,\xi} \ & \frac{1}{2}||\mathbf{w}||^2 + C\sum_{i=1}^{m}\xi_i + \lambda b, \\ \text{s.t. } & y_i\left(\mathbf{w}^T\mathbf{x}_i + b\right) - 1 + \xi_i \geq 0, \\ & \xi_i \geq 0, \end{aligned}$$

as we want to minimize the value of b aiming at minimizing the risk of the unknown.

Using the Lagrangian method, we have the Lagrangian defined as

$$\begin{aligned} \mathcal{L}(\mathbf{w}, b, \xi, \alpha, r) = & \frac{1}{2}||\mathbf{w}||^2 + C\sum_{i=1}^{m}\xi_i + \lambda b - \sum_{i=1}^{m} r_i\xi_i \\ & - \sum_{i=1}^{m}\alpha_i\left[y_i\left(\mathbf{w}^T\mathbf{x}_i + b\right) - 1 + \xi_i\right], \end{aligned} \tag{A.1}$$

in which $\alpha_i \in \mathbb{R}$ and $r_i \in \mathbb{R}$, $i = 1, \ldots, m$, are the Lagrangian multipliers.

First we want to minimize with respect to $\mathbf{w}$, b, and ξ_i, then we must ensure

$$\nabla_{\mathbf{w}} = \frac{\partial}{\partial b}\mathcal{L} = \frac{\partial}{\partial \xi_i}\mathcal{L} = 0.$$

Consequently, we have

$$w - \sum_{i=1}^{m} \alpha_i y_i \mathbf{x}_i = 0 \implies w = \sum_{i=1}^{m} \alpha_i y_i \mathbf{x}_i, \tag{A.2}$$

$$\lambda - \sum_{i=1}^{m} \alpha_i y_i = 0 \implies \sum_{i=1}^{m} \alpha_i y_i = \lambda, \tag{A.3}$$

$$C - \alpha_i - r_i = 0 \implies r_i = C - \alpha_i. \tag{A.4}$$

As the Lagrangian multipliers α_i, r_i must be greater than 0, from Equation (A.4) we have the constraint $0 \leq \alpha_i \leq C$ as a consequence in the dual problem of the soft margin formulation. This is the same constraint we have in the traditional formulation of the SVM classifier.

Using Equations (A.2)–(A.4) to simplify the Lagrangian in Equation (A.1), we have

$$\mathcal{L}(\mathbf{w}, b, \xi, \alpha, r) = \sum_{i=1}^{m} \alpha_i - \frac{1}{2}||\mathbf{w}||^2,$$

i.e., the same Lagrangian of the traditional SVM optimization problem. The optimization of the bias term b relies on the constraint in Equation (A.3).

Therefore, the dual optimization problem is defined as

$$\begin{aligned} &\min_{\alpha} W(\alpha) = -\mathcal{L}(\mathbf{w}, b, \xi, \alpha, r) = \frac{1}{2}||\mathbf{w}||^2 - \sum_{i=1}^{m} \alpha_i, \\ &\text{s.t. } 0 \leq \alpha_i \leq C, \ \forall i, \\ &\quad \sum_{i=1}^{m} \alpha_i y_i = \lambda. \end{aligned}$$

Appendix B

Proof of Proposition 2

From the Karush-Kuhn-Tucker (KKT) [Bishop, 2006] conditions, the bias term is defined as

$$\begin{aligned} b &= y_i - \sum_{j=1}^{m} \alpha_j y_j K\left(\mathbf{x}_i, \mathbf{x}_j\right) \\ &= y_i - \sum_{\substack{j=1: \\ y_j=1}}^{m} \alpha_j K\left(\mathbf{x}_i, \mathbf{x}_j\right) + \sum_{\substack{j=1: \\ y_j=-1}}^{m} \alpha_j K\left(\mathbf{x}_i, \mathbf{x}_j\right), \end{aligned}$$

for any i such that $0 < \alpha_i < C$. Now, let us consider two possible cases: (1) $y_i = 1$ and (2) $y_i = -1$. For **Case (1)**, we have

$$b = 1 - \alpha_i - \sum_{\substack{j=1: \\ y_j=1, \\ j\neq i}}^{m} \alpha_j K\left(\mathbf{x}_i, \mathbf{x}_j\right) + \sum_{\substack{j=1: \\ y_j=-1}}^{m} \alpha_j K\left(\mathbf{x}_i, \mathbf{x}_j\right),$$

as $K\left(\mathbf{x}_i, \mathbf{x}_i\right) = 1$. Note that $0 < K(\mathbf{x}, \mathbf{x}') \leq 1$. To show that there exists some λ such that $b < 0$, we analyze the worst case, i.e., when the kernel in the second summation—for negative training samples—is 1. Then, we have

$$b = 1 - \alpha_i - \sum_{\substack{j=1: \\ y_j=1, \\ j\neq i}}^{m} \alpha_j K\left(\mathbf{x}_i, \mathbf{x}_j\right) + \sum_{\substack{j=1: \\ y_j=-1}}^{m} \alpha_j.$$

From Equation (5.11), we have

$$\sum_{\substack{j=1: \\ y_j=-1}}^{m} \alpha_j = \sum_{\substack{j=1: \\ y_j=1}}^{m} \alpha_j - \lambda, \tag{B.1}$$

then

$$b = 1 - \sum_{\substack{j=1: \\ y_j=1, \\ j\neq i}}^{m} \alpha_j K\left(\mathbf{x}_i, \mathbf{x}_j\right) + \sum_{\substack{j=1: \\ y_j=1 \\ j\neq i}}^{m} \alpha_j - \lambda$$

Analyzing the worst case again, considering $\alpha_j = C$ for positive training samples, with $j \neq i$, we have

$$\begin{aligned} b &= 1 - C \sum_{\substack{j=1: \\ y_j=1, \\ j\neq i}}^{m} K\left(\mathbf{x}_i, \mathbf{x}_j\right) + C\left(m_p - 1\right) - \lambda \\ &= 1 + Cm_p - C - C \sum_{\substack{j=1: \\ y_j=1, \\ j\neq i}}^{m} K\left(\mathbf{x}_i, \mathbf{x}_j\right) - \lambda. \end{aligned}$$

To ensure $b < 0$ it is sufficient to let

$$\lambda > 1 + Cm_p - C \left(1 + \sum_{\substack{j=1: \\ y_j=1, \\ j\neq i}}^{m} K\left(\mathbf{x}_i, \mathbf{x}_j\right) \right).$$

Given a $C \geq 1$, it is always possible to obtain some λ such that $\lambda < Cm_p$.

For **Case (2)**, we have

$$b = -1 - \sum_{\substack{j=1: \\ y_j=1}}^{m} \alpha_j K\left(\mathbf{x}_i, \mathbf{x}_j\right) + \sum_{\substack{j=1: \\ y_j=-1}}^{m} \alpha_j K\left(\mathbf{x}_i, \mathbf{x}_j\right).$$

Considering the worst case for the values of the kernel for negative samples and using the equality in Equation (B.1), we have

$$b = -1 - \sum_{\substack{j=1: \\ y_j=1}}^{m} \alpha_j K\left(\mathbf{x}_i, \mathbf{x}_j\right) + \sum_{\substack{j=1: \\ y_j=1}}^{m} \alpha_j - \lambda.$$

Considering the highest possible value for b, by setting $\alpha_j = C$ for positive samples, we have

$$b = -1 - C \sum_{\substack{j=1: \\ y_j=1}}^{m} K\left(\mathbf{x}_i, \mathbf{x}_j\right) + Cm_p - \lambda.$$

In this case, to ensure $b < 0$ it is sufficient to let

$$\lambda > Cm_p - 1 - C \sum_{\substack{j=1: \\ y_j=1}}^{m} K\left(\mathbf{x}_i, \mathbf{x}_j\right),$$

which is possible to obtain for any value of C. $\square$

Appendix C

Additional statistical tests

In this appendix, we present the Wilcoxon statistical tests for the same experiments we have presented the Binomial statistical tests throughout Chapter 6. In Table C.1, we summarize the correspondence of the tables with Binomial results to the tables with Wilcoxon results.

Binomial (Chapter 6)	Wilcoxon (this appendix)
Table 6.2	Table C.2
Table 6.3	Table C.3
Table 6.5	Table C.4
Table 6.6	Table C.5
Table 6.7	Table C.6
Table 6.8	Table C.7
Table 6.9	Table C.8
Table 6.10	Table C.9
Table 6.11	Table C.10
Table 6.12	Table C.11

Table C.1: Correspondence of Wilcoxon statistical tests for the previously presented Binomial statistical tests.

Measure	TNN_E	TNN_I	$OSOPF^{CV}$	OSOPF	$OSNN^{CV}$
NA	**<.0001***	**<.0001***	**<.0001***	**<.0001***	**<.0001***
HNA	**<.0001***	**0.0039***	**0.0039***	**0.0005***	**0.0039***
$OSFM_M$	**0.0106**	0.1042	**0.0106**	**0.0001***	**0.0106**
$OSFM_\mu$	**0.0106**	**0.0146**	**0.0106**	**0.0001***	**0.0106**
FM_M	**0.0220**	0.1569	**0.0176**	**<.0001***	**0.0176**
FM_μ	**<.0001***	**<.0001***	**<.0001***	**<.0001***	**<.0001***
AKS	*<.0001**	*<.0001**	*<.0001**	**0.0002***	*<.0001**
AUS	**<.0001***	**<.0001***	**<.0001***	**<.0001***	**<.0001***

Table C.2: Wilcoxon statistical tests comparing the OSNN with baselines. Each cell compares results for all datasets considering all number of available classes. For each number of available classes and dataset, the mean of the 10 experiments was taken before the statistical test. **Bold** means there is statistical difference with 95% of confidence. "*" indicates the statistical difference is with 99% of confidence. And **<.0001*** indicates the statistical difference is with 99.99% of confidence. *Emphasized* indicates the method in the column obtains better performance for the measure associated with that row.

Measure	SVM_C	$OCSVM_C$	DBC_C	OVS_C	$WSVM_C$	$PISVM_C$	$SVDD_C$
NA	**0.0001***	**<.0001***	**<.0001***	**<.0001***	**<.0001***	**<.0001***	**<.0001***
HNA	**0.0001***	**<.0001***	**<.0001***	**<.0001***	**<.0001***	**<.0001***	**<.0001***
$OSFM_M$	**0.0007***	**0.0002***	**<.0001***	**<.0001***	**<.0001***	**<.0001***	**<.0001***
$OSFM_\mu$	**0.0004***	**0.0061***	**<.0001***	**<.0001***	**<.0001***	**<.0001***	**<.0001***
FM_M	**0.0010***	**<.0001***	**0.0002***	**<.0001***	**0.0002***	**<.0001***	**<.0001***
FM_μ	**<.0001***	0.0946	**<.0001***	**<.0001***	**<.0001***	**<.0001***	**<.0001***
AKS	*<.0001**	**<.0001***	*<.0001**	0.0735	*<.0001**	*<.0001**	**<.0001***
AUS	**<.0001***	0.7282	**<.0001***	**<.0001***	**<.0001***	**<.0001***	**<.0001***

Table C.3: Wilcoxon statistical tests comparing the $SSVM_C$ with baselines. Each cell compares results for all datasets considering all number of available classes. For each number of available classes and dataset, the mean of the 10 experiments was taken before the statistical test. **Bold** means there is statistical difference with 95% of confidence. "*" indicates the statistical difference is with 99% of confidence. And **<.0001*** indicates the statistical difference is with 99.99% of confidence. *Emphasized* indicates the method in the column obtains better performance for the measure associated with that row.

Measure	SVM	OCSVM	DBC	OVS	WSVM	PISVM	SVDD	SSVM
NA	*0.0924*	**<.0001***	**<.0001***	**0.0001***	**<.0001***	**<.0001***	**<.0001***	**0.0006***
HNA	***0.0011****	**<.0001***	**<.0001***	**<.0001***	**<.0001***	**<.0001***	***<.0001****	***0.0102***
$OSFM_M$	**0.0004***	**<.0001***	**<.0001***	**0.0010***	**<.0001***	**<.0001***	***<.0001****	**<.0001***
$OSFM_\mu$	**0.0009***	**<.0001***	**<.0001***	*0.8303*	**<.0001***	**<.0001***	***<.0001****	**<.0001***
FM_M	**0.0108**	**<.0001***	**0.0011***	**<.0001***	**<.0001***	**<.0001***	***<.0001****	**<.0001***
FM_μ	**<.0001***	**<.0001***	**<.0001***	**<.0001***	**<.0001***	**<.0001***	**<.0001***	**<.0001***
AKS	***<.0001****	**<.0001***	***<.0001****	***<.0001****	***<.0001****	***<.0001****	***<.0001****	***<.0001****
AUS	**<.0001***	**<.0001***	**<.0001***	**<.0001***	**<.0001***	**<.0001***	**<.0001***	**<.0001***

Table C.4: Wilcoxon statistical tests for the pairwise comparison between closed- and open-set grid search implementation for the methods. Each cell compares results for all datasets considering all number of available classes. For each number of available classes and dataset, the mean of the 10 experiments was taken before the statistical test. **Bold** means there is statistical difference with 95% of confidence. "*" indicates the statistical difference is with 99% of confidence. And **<.0001*** indicates the statistical difference is with 99.99% of confidence. *Emphasized* means the version with closed-set grid search obtains better performance for the measure associated with that row.

Measure	SVM_O	$OCSVM_O$	DBC_O	OVS_O	$WSVM_O$	$PISVM_O$	$SVDD_O$
NA	0.1800	**<.0001***	**0.0002***	**<.0001***	**0.0004***	0.2016	**<.0001***
HNA	0.9157	0.1237	0.9157	**<.0001***	0.5025	*0.9157*	**<.0001***
$OSFM_M$	*1.0000*	0.6281	0.6281	**0.0005***	0.7078	1.0000	**<.0001***
$OSFM_\mu$	0.6322	0.5127	0.0551	**<.0001***	0.1430	0.6322	**<.0001***
FM_M	1.0000	0.5726	0.5726	**<.0001***	1.0000	1.0000	**<.0001***
FM_μ	0.0527	**0.0095***	**<.0001***	**<.0001***	**<.0001***	**0.0001***	0.9911
AKS	*1.0000*	**0.0057***	***0.0017****	*1.0000*	***0.0005****	*0.3599*	**<.0001***
AUS	**0.0104**	**0.0047***	**<.0001***	**<.0001***	**<.0001***	**<.0001***	***0.0104***

Table C.5: Wilcoxon statistical tests comparing the OSNN with best baselines. Each cell compares results for all datasets considering all number of available classes. For each number of available classes and dataset, the mean of the 10 experiments was taken before the statistical test. **Bold** means there is statistical difference with 95% of confidence. "*" indicates the statistical difference is with 99% of confidence. And **<.0001*** indicates the statistical difference is with 99.99% of confidence. *Emphasized* indicates the method in the column obtains better performance for the measure associated with that row.

Measure	SVM_O	$OCSVM_O$	DBC_O	OVS_O	$WSVM_O$	$PISVM_O$	$SVDD_O$
NA	**0.0001***	**<.0001***	**<.0001***	**<.0001***	**<.0001***	**0.0010***	**<.0001***
HNA	**0.0008***	**0.0001***	**0.0103**	**<.0001***	**0.0103**	0.3161	**<.0001***
$OSFM_M$	**0.0004***	**0.0002***	**0.0011***	**<.0001***	**0.0019***	0.1375	**<.0001***
$OSFM_\mu$	**0.0001***	**0.0013***	**<.0001***	**<.0001***	**<.0001***	**0.0013***	**<.0001***
FM_M	**0.0007***	**0.0001***	**0.0006***	**<.0001***	**0.0026***	0.1256	**<.0001***
FM_μ	**<.0001***	**0.0008***	**<.0001***	**<.0001***	**<.0001***	**<.0001***	0.1375
AKS	*1.0000*	**<.0001***	***<.0001****	*1.0000*	***<.0001****	***0.0005****	**<.0001***
AUS	**0.0001***	**0.0058***	**<.0001***	**<.0001***	**<.0001***	**<.0001***	***0.0103***

Table C.6: Wilcoxon statistical tests comparing the $SSVM_O$ with best baselines. Each cell compares results for all datasets considering all number of available classes. For each number of available classes and dataset, the mean of the 10 experiments was taken before the statistical test. **Bold** means there is statistical difference with 95% of confidence. "*" indicates the statistical difference is with 99% of confidence. And **<.0001*** indicates the statistical difference is with 99.99% of confidence. *Emphasized* indicates the method in the column obtains better performance for the measure associated with that row.

Measure	SVM^{WB}	SVM_6^{WB}
NA	0.2185	0.6136
HNA	**0.0069***	0.7793
$OSFM_M$	**0.0112**	0.5369
$OSFM_\mu$	**0.0013***	0.7621
FM_M	0.0634	0.7282
FM_μ	1.0000	*1.0000*
AKS	*0.7240*	0.7240
AUS	0.7352	*0.8489*

Table C.7: Wilcoxon statistical tests comparing the SVM_1^{WB} with SVM without bias term alternatives. Each cell compares results for all datasets considering all number of available classes. For each number of available classes and dataset, the mean of the 10 experiments was taken before the statistical test. **Bold** means there is statistical difference with 95% of confidence. "*" indicates the statistical difference is with 99% of confidence. And **<.0001*** indicates the statistical difference is with 99.99% of confidence. *Emphasized* indicates the method in the column obtains better performance for the measure associated with that row.

Measure	$\text{OSNN}_{10}^{\lambda_r}$	$\text{OSNN}_{30}^{\lambda_r}$	$\text{OSNN}_{70}^{\lambda_r}$	$\text{OSNN}_{90}^{\lambda_r}$
NA	0.1899	0.2260	**0.0044***	**<.0001***
HNA	**0.0010***	**0.0010***	***0.0129***	**0.0001***
OSFM_M	1.0000	1.0000	*1.0000*	0.3971
OSFM_μ	1.0000	1.0000	*1.0000*	0.1609
FM_M	1.0000	1.0000	*1.0000*	0.2941
FM_μ	***0.0003****	***0.0002****	**0.0001***	**<.0001***
AKS	**<.0001***	**<.0001***	***<.0001****	***<.0001****
AUS	***<.0001****	***<.0001****	**<.0001***	**<.0001***

Table C.8: Wilcoxon statistical tests comparing the OSNN with OSNN alternatives. Each cell compares results for all datasets considering all number of available classes. For each number of available classes and dataset, the mean of the 10 experiments was taken before the statistical test. **Bold** means there is statistical difference with 95% of confidence. "*" indicates the statistical difference is with 99% of confidence. And **<.0001*** indicates the statistical difference is with 99.99% of confidence. *Emphasized* indicates the method in the column obtains better performance for the measure associated with that row.

Measure	SVM_O	OCSVM_O	DBC_O	OVS_O	WSVM_O	PISVM_O	SVDD_O
NA	0.9264	**<.0001***	***<.0001****	**<.0001***	*0.9264*	**0.0057***	**<.0001***
HNA	0.9264	**<.0001***	***<.0001****	**<.0001***	*0.9264*	**0.0063***	**<.0001***
OSFM_M	0.9429	**<.0001***	***<.0001****	**<.0001***	0.9429	**0.0004***	**<.0001***
OSFM_μ	1.0000	**<.0001***	***<.0001****	**<.0001***	*1.0000*	1.0000	**<.0001***
FM_M	0.9429	**<.0001***	***<.0001****	**<.0001***	0.9429	**0.0004***	**<.0001***
FM_μ	1.0000	**<.0001***	***<.0001****	**<.0001***	*0.9689*	*1.0000*	*1.0000*
AKS	***<.0001****	**<.0001***	***<.0001****	**<.0001***	**<.0001***	**<.0001***	**<.0001***
AUS	0.2804	**<.0001***	***0.0013****	**<.0001***	*0.2571*	*0.2804*	***0.0007****

Table C.9: Wilcoxon statistical tests comparing the SSVM_O with baselines in ImageNet. **Bold** means there is statistical difference with 95% of confidence. "*" indicates the statistical difference is with 99% of confidence. And **<.0001*** indicates the statistical difference is with 99.99% of confidence. *Emphasized* indicates the method in the column obtains better performance for the measure associated with that row.

Measure	SVM_O	$OCSVM_O$	DBC_O	OVS_O	$WSVM_O$	$PISVM_O$	$SVDD_O$
NA	**0.0062***	**0.0044***	**0.0001***	**<.0001***	**0.0001***	***0.0055****	**<.0001***
HNA	0.1048	**0.0116**	**0.0018***	**<.0001***	**0.0002***	***0.0037****	**<.0001***
$OSFM_M$	*0.4259*	0.2024	0.2024	*0.4771*	0.3724	***0.0005****	**<.0001***
$OSFM_\mu$	0.0810	**0.0128**	**0.0161**	0.3085	**0.0436**	***0.0008****	**<.0001***
FM_M	0.1840	0.1173	0.0602	0.1173	0.1173	***0.0007****	**<.0001***
FM_μ	**<.0001***	**0.0001***	**<.0001***	**0.0006***	**0.0006***	***0.0010****	**<.0001***
AKS	*1.0000*	**0.0014***	*1.0000*	***<.0001****	1.0000	***0.0002****	**<.0001***
AUS	**<.0001***	***0.0064****	**<.0001***	**<.0001***	**0.0034***	0.0656	***<.0001****

Table C.10: Wilcoxon statistical tests comparing the $SSVM_O$ with baselines in CIFAR-10. **Bold** means there is statistical difference with 95% of confidence. "*" indicates the statistical difference is with 99% of confidence. And **<.0001*** indicates the statistical difference is with 99.99% of confidence. *Emphasized* indicates the method in the column obtains better performance for the measure associated with that row.

Measure	SVM_O	$OCSVM_O$	DBC_O	OVS_O	$WSVM_O$	$PISVM_O$	$SVDD_O$
NA	0.1457	**<.0001***	*1.0000*	**<.0001***	*0.7100*	*1.0000*	**<.0001***
HNA	0.6318	**<.0001***	*1.0000*	**<.0001***	*0.6318*	*1.0000*	**<.0001***
$OSFM_M$	0.1981	**<.0001***	*0.0930*	**<.0001***	***0.0038****	*0.1333*	**<.0001***
$OSFM_\mu$	0.0625	**<.0001***	*0.5787*	**<.0001***	***0.0081****	*0.5787*	**<.0001***
FM_M	0.1491	**<.0001***	*0.5069*	**<.0001***	***0.0035****	*0.5069*	**<.0001***
FM_μ	**0.0002***	**<.0001***	1.0000	**<.0001***	***0.0031****	1.0000	**<.0001***
AKS	*0.3184*	**<.0001***	***<.0001****	***0.0006****	***<.0001****	***0.0044****	**<.0001***
AUS	**<.0001***	***<.0001****	**<.0001***	**<.0001***	**<.0001***	**0.0006***	***<.0001****

Table C.11: Wilcoxon statistical tests comparing the $SSVM_O$ with baselines in MNIST. **Bold** means there is statistical difference with 95% of confidence. "*" indicates the statistical difference is with 99% of confidence. And **<.0001*** indicates the statistical difference is with 99.99% of confidence. *Emphasized* indicates the method in the column obtains better performance for the measure associated with that row.