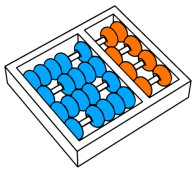
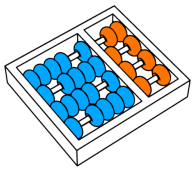


Pedro Ribeiro Mendes Júnior

"Open-Set Optimum-Path Forest Classifier"

"*Classificador Optimum-Path Forest para Cenário Aberto*"

CAMPINAS
2014



University of Campinas
Institute of Computing

Universidade Estadual de Campinas
Instituto de Computação

Pedro Ribeiro Mendes Júnior

"Open-Set Optimum-Path Forest Classifier"

Supervisor: **Prof. Dr. Anderson de Rezende Rocha**
Orientador(a):

Co-Supervisor: **Prof. Dr. Ricardo da Silva Torres**
Coorientador(a):

"Classificador Optimum-Path Forest para Cenário Aberto"

MSc Dissertation presented to the Post Graduate Program of the Institute of Computing of the University of Campinas to obtain a master's degree in Computer Science.

Dissertação de Mestrado apresentada ao Programa de Pós-Graduação em Ciência da Computação do Instituto de Computação da Universidade Estadual de Campinas para obtenção do título de Mestre em Ciência da Computação.

This volume corresponds to the final version of the Dissertation defended by Pedro Ribeiro Mendes Júnior, under the supervision of Prof. Dr. Anderson de Rezende Rocha.

Este exemplar corresponde à versão final da Dissertação defendida por Pedro Ribeiro Mendes Júnior, sob orientação de Prof. Dr. Anderson de Rezende Rocha.

Supervisor's signature / *Assinatura do Orientador(a)*

CAMPINAS
2014

Ficha catalográfica
Universidade Estadual de Campinas
Biblioteca do Instituto de Matemática, Estatística e Computação Científica
Maria Fabiana Bezerra Muller - CRB 8/6162

M522o

Mendes Júnior, Pedro Ribeiro, 1990-
Open-set optimum-path forest classifier / Pedro Ribeiro Mendes Júnior. – Campinas, SP : [s.n.], 2014.

Orientador: Anderson de Rezende Rocha.
Coorientador: Ricardo da Silva Torres.
Dissertação (mestrado) – Universidade Estadual de Campinas, Instituto de Computação.

1. Reconhecimento de padrões. 2. Reconhecimento em cenário aberto. 3. Floresta de caminhos ótimos. I. Rocha, Anderson de Rezende,1980-. II. Torres, Ricardo da Silva,1977-. III. Universidade Estadual de Campinas. Instituto de Computação. IV. Título.

Informações para Biblioteca Digital

Título em outro idioma: Classificador optimum-path forest para cenário aberto
Palavras-chave em inglês:
Pattern recognition
Open set recognition
Optimum-path forest
Área de concentração: Ciência da Computação
Titulação: Mestre em Ciência da Computação
Banca examinadora:
Anderson de Rezende Rocha [Orientador]
Gisele Lobo Pappa
Alexandre Xavier Falcão
Data de defesa: 22-08-2014
Programa de Pós-Graduação: Ciência da Computação

TERMO DE APROVAÇÃO

Defesa de Dissertação de Mestrado em Ciência da Computação, apresentada pelo(a) Mestrando(a) **Pedro Ribeiro Mendes Júnior**, aprovado(a) em **22 de agosto de 2014**, pela Banca examinadora composta pelos Professores Doutores:

Prof(a). Dr(a). Gisele Lobo Pappa
Titular

Prof(a). Dr(a). Alexandre Xavier Falcão
Titular

Prof(a). Dr(a). Anderson de Rezende Rocha
Presidente

Institute of Computing /*Instituto de Computação*
University of Campinas /*Universidade Estadual de Campinas*

Open-Set Optimum-Path Forest Classifier

Pedro Ribeiro Mendes Júnior[1]

August 22, 2014

Examiner Board/*Banca Examinadora*:

- Prof. Dr. Anderson de Rezende Rocha (Supervisor/*Orientador*)
- Prof. Dr. Alexandre Xavier Falcão
  Institute of Computing - UNICAMP
- Prof[a]. Dr[a]. Gisele Lobo Pappa
  Computer Science Department - UFMG
- Prof. Dr. Siome Klein Goldenstein
  Institute of Computing - UNICAMP (Substitute/*Suplente*)
- Prof[a]. Dr[a]. Nina Sumiko Tomita Hirata
  Institute of Mathematics and Statistics - USP (Substitute/*Suplente*)

[1]Part of this work was conducted in the context of the project "Pattern recognition and classification by feature engineering, *-fusion, open-set recognition, and meta-recognition", sponsored by Samsung Eletrônica da Amazônia Ltda., in the framework of law No. 8248/91.

Abstract

An open-set recognition scenario is the one in which there are no a priori training samples for some classes that might appear during testing. Usually, many applications are inherently open set. Consequently, the successful closed-set solutions in the literature are not always suitable for real-world recognition problems. Here, we propose a novel multiclass open-set classifier that extends upon the Optimum-Path Forest (OPF) classifier. OPF is a graph-based, simple, parameter independent, multiclass, and widely used classifier for closed-set problems. Our proposed Open-Set OPF (OSOPF) method incorporates the ability to recognize samples belonging to classes that are unknown at training time, being suitable for open-set recognition. In addition, we propose new evaluation measures for assessing the effectiveness performance of classifiers in open-set problems. In experiments, we consider six large datasets with different open-set recognition scenarios and demonstrate that the proposed OSOPF significantly outperforms its counterparts of the literature.

Resumo

Em reconhecimento de padrões, um cenário aberto é aquele em que não há amostras de treinamento para algumas classes que podem aparecer durante o teste. Normalmente, muitas aplicações são inerentemente de cenário aberto. Consequentemente, as soluções bem sucedidas da literatura para cenário fechado nem sempre são adequadas para problemas de reconhecimento na prática. Nesse trabalho, propomos um novo classificador multiclasse para cenário aberto, que estende o classificador *Optimum-Path Forest* (OPF). O OPF é um classificador de padrões baseado em grafos, simples, independente de parâmetros, multiclasse e desenvolvido para para problemas de cenário fechado. O método que propomos, o *Open-Set OPF* (OSOPF), incorpora a capacidade de reconhecer as amostras pertencentes às classes que são desconhecidas no tempo de treinamento, sendo adequado para reconhecimento em cenário aberto. Além disso, propomos novas medidas para avaliação de classificadores propostos para problemas em cenário aberto. Nos experimentos, consideramos seis grandes bases de dados com diferentes cenários de reconhecimento e demonstramos que o OSOPF proposto supera significativamente as abordagens existentes na literatura.

Sumário

Acknowledgements

I am grateful to everyone who contributed to me to spend two years in a pleasant way here in Campinas. Firstly, my girlfriend Amanda Veríssimo who, as an eximious cultivator of her own feelings, knew to keep and enrich them despite the physical distance that we have since I came here. Finding such sensitive correspondence in her being is what fills me with energy to perform all my activities, thereby expanding the projections that these activities may have in my own life and our lives.

I am grateful to my friends of the Logosophical Foundation, with whom I most can establish a true understanding. Arriving at Campinas and finding friends with whom I can exchange points of view on major topics that greatly pleases my spirit was undoubtedly a great joy. Knowing that a friendship must always be held with the ternary sympathy–trust–respect, I say that their efforts to raise their own human qualities, individually, have this ternary as a natural consequence in my heart and my mind. I do not mention names in particular because everyone is special; and to mention every name will stop the flow of reading.

I am grateful to my friend Nayara Hachich, to the company and the conversations we kept over the period we lived in the same "republic". Her willingness to maintain our friendship, when no longer we lived in the same "republic", resonated so grateful to me, even more by the understanding we reached in later times.

During the master course here in Campinas so much I apprehended regarding my academic life. I am grateful to all the members of the research team of the project we developed together over these past two years; the experience was new and rich for me. Firstly to the professor Ricardo Torres who received me in Campinas, provided guidance, and then engaged me in the project in which I had professor Anderson Rocha as advisor. Rocha's mind seems to have infinite space to hold all his mentees and I am grateful to him for always conserving the space I require from his mind.

Finally, I thank my parents for the great opportunity they gave me to bring me to life. More than that, I am grateful for the elements they left me in the form of inheritance, which are thoughts, feelings, and knowledge without which my life could be a simple waiting for death. Increasingly I realize the value that these elements have for me.

"Todo lo que permanezca ajeno al hombre es como si no existiera para él, mas no por ello deja de existir para los demás."

Carlos Bernardo González Pecotche

List of Figures

List of Tables

Chapter 1

Introduction

Typical *Pattern Classification* refers to the problem of assigning a test sample to one or more known classes. In a typical classification problem, it is not necessary to indicate that the test sample does not belong to one of the known classes. For instance, classifying an image of a digit as one out of 10 possible digits ($0 \ldots 9$). We know, by definition, that this problem has 10 classes. On the other hand, *recognition* is the task of verifying whether a test sample belongs to one of the known classes and, if so, finding to which of them the test sample belongs. In the recognition problem, the test sample can belong to none of the classes known by the classifier during training. For instance, classifying if a biometric sample belongs to one of the persons registered in the system or automatically reject it otherwise. The recognition scenario is more similar to what we call an open-set scenario, in which the classifier cannot be trained with all possible classes because the classes are ill-sampled, not sampled, or *unknown* [43].

In some problems, all classes are known a priori, leading to a closed-set scenario. For example, suppose that inside an aquarium there are only three species of fish and biologists are interested in training a classifier with all three classes. In this application, all test samples are assigned to one out of those classes because it is known that all fish that could be tested at the aquarium belong to one of those three classes. The same classifier, however, is unsuitable for being used in a new larger aquarium containing the same three species and some new ones, i.e., in an open-set scenario in which new species are *unknown*. In this case, the trained classifier will always classify an *unknown* sample as belonging to a known class because it was developed to be used in the closed-set scenario (first aquarium), leading to an undesired misclassification.

Open-set classification problems are typically a multiclass problem. The classifier must assign the label of one of the training classes or an *unknown* label to test samples. Approaches aiming at tackling this problem must avoid the following errors:

- the test sample belongs to one of the known training classes but the classifier assigns

it to a wrong class;

- the test sample belongs to one of the known training classes but the classifier assigns it to the *unknown* label (*false unknown*); and
- the test sample is *unknown* but the classifier assigns it to one of the known training classes (e.g., the aforementioned fish species recognition error).

In a closed-set classification scenario, only the first kind of error is possible. A common approach to partially handling the open-set scenario relies on the use of threshold-based classification schemes [38]. Basically, those methods verify whether the *matching score* is greater than or equal to a previously defined threshold. Phillips et al. [38], for example, used this approach to classify a sample as *unknown* when the most similar training class is not *enough* similar. In this approach, without that threshold on the matching score, an *unknown* test sample will always be assigned to one of the training classes.

Another trend relies on modifying the classification engine or objective function of Support Vector Machines (SVM) classifiers [11, 12, 43]. The traditional SVM classifier (a binary classifier) assigns a test sample to a certain class even if the test sample is far from the training samples of the class. SVM defines half-spaces [7] and does not verify how far the test sample is from the training samples. This strong generalization may not be useful in the open-set scenario given that probably the faraway test sample must be classified as *unknown*. Therefore, we can understand the binary SVM as a closed-set classifier. The One-vs-All multiclass SVM ($\text{SVM}_{\text{MCBIN}}$) [39, 40], however, can be considered an open-set classifier, as all the One-vs-All binary SVMs used in the $\text{SVM}_{\text{MCBIN}}$ procedure [7] are able to classify a test sample as *negative* and, in this case, the test sample could be considered *unknown*. Figure 1.1 illustrates this case.

An SVM is obtained by solving an optimization problem whose objective is to minimize the *empirical risk*[1] measured on training samples. In an open-set recognition scenario, the objective is to minimize the *risk of the unknown*[2] by minimizing the *open space of risk*[3] instead of minimizing only the *empirical risk*. The *open space of risk* is the region of the feature space in which a test sample would be classified as known (one of the available classes for training). As SVMs define half-spaces, it is not easy to create a bounded *open space of risk*. Every SVM extension for open set we found in the literature [11, 12, 43] maintains an unbounded *open space of risk*. Potential solutions for open-set recognition problems should minimize the *open space of risk*, preferably making it bounded.

[1] Approximation of the actual risk. It is calculated by averaging the loss function on the training set.

[2] Risk of the unknown from insufficient generalization or specialization of a recognition function f.

[3] Open space representing the learned recognition function f, outside the support of training samples [43].

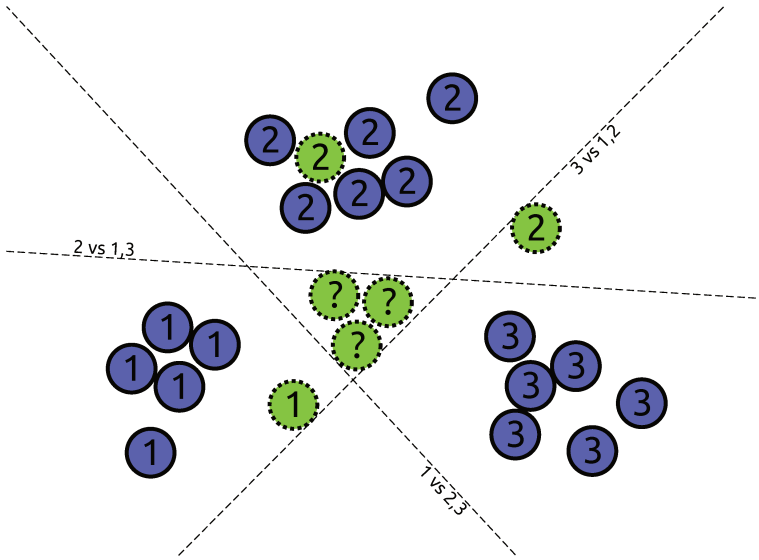


Figure 1.1: Example of samples classified as *unknown* by the multiclass Support Vector Machines using One-vs-All approach. The dotted–green samples are test samples. The question-mark "?" samples are classified as *unknown* because they are classified as *negative* by all One-vs-All binary Support Vector Machines.

In this work, we address this issue by introducing open-set versions of the Optimum-Path Forest (OPF) classifier, called the open-set OPF (OSOPF). The OPF classifier, introduced by Papa et al. [34, 36], is a graph-based classifier that was developed as a generalization of the Image Forest Transform (IFT) [16] from the image domain to the feature space. OPF is similar to the well-known k-Nearest Neighbors (kNN) algorithm [46, 47], inherently multiclass, and parameter independent. The OPF does not assume any prior knowledge about classes in the feature space [34]. It has shown good results in many classification problems, outperforming other classification engines such as SVM for some problems [13, 14, 33, 34, 35, 37, 49]. However, the current implementation of the OPF is inherently targeted at closed-set problems.

Our approaches extend upon the traditional closed-set OPF and introduces modifications to verify if a test sample can be classified as *unknown*. To the best of our knowledge, the OSOPF extension is the first version of the OPF classifier suitable for open-set recognition. Its main advantage, compared to the SVM-based and other existing approaches, is that it is inherently multiclass, i.e., the efficiency of the OSOPF is not affected as the number of *available classes* for training increases. Another major advantage is that it can create a bounded *open space of risk* for every known class therefore gracefully protecting the classes of interest and rejecting unknown classes. These two advantages make OSOPF ideal for developing solutions for novelty detection and online learning which could detect *unknown* classes on-the-fly and include them on the recognition system automatically.

This is a whole new research branch with countless applications.

In addition to the proposed open-set solutions, we have designed a special experimental protocol for benchmarking open-set solutions. In many works in the literature, in spite of the explicit discrimination between classification and recognition, authors perform tests on recognition problems considering a closed-set scenario instead of an open-set one [2, 9, 20, 26]. Consequently, the observed results are not similar to what is observed in real-world open-set applications. Another limitation of existing experimental protocols is the lack of appropriate measures to assess the quality of the open-set classifiers. Therefore, another contribution of this work is the discussion of two measures adapted to the open-set scenario: the *normalized accuracy* and the *open-set f-measure*. The purpose of such adapted measures is to evaluate the performance of classifiers when handling both *known* and *unknown* test samples.

To validate the proposed methods and to compare them with existing ones in the literature, we considered a diverse set of recognition problems, such as object recognition (*Caltech-256*, *ALOI* and *ukbench*), scene recognition (*15-Scene*), letter recognition (*letter*), and sign language recognition (*Auslan*). The number of classes in such problems vary from 15 to 2550 and the number of examples vary from a few thousands to hundreds of thousands. The experiments were performed by considering training scenarios with three, six, nine, and twelve classes, and testing scenarios with samples of the remaining classes as possible *unknown*. We compared the proposed OSOPF with a multiclass version of the SVMDBC, which is an open-set SVM extension recently proposed by Costa et al. [11, 12]. We also compared the OSOPF with the traditional OPF, the $\text{SVM}_{\text{MCBIN}}$, and the One-vs-Set Machine recently proposed by Scheirer et al. [43]. The proposed OSOPF outperformed all existing solutions, with statistical significance.

We organized the remainder of this work as follows. In Chapter 2, we present related work in open-set recognition. In Chapter 3, we describe the OPF classifier and its concepts as well as its properties by focusing on the context of open-set recognition. In Chapter 4, we introduce the first ideas for the development of the proposed method and some of its properties. Then, in the same chapter, we describe the OSOPF in its inherent-multiclass form. In Chapter 5, we show the performed experiments as well as the failing cases of the proposed method and possible future improvements. Finally, in Chapter 6, we conclude the work and outline important research directions for future work.

Chapter 2

Related work

In this chapter, we present previous approaches that somehow deal with open-set classification scenarios. Those approaches can be divided into four main categories: approaches based on one-class classifiers; approaches based on binary classifiers; approaches used in similar problems; and approaches that explicitly deal with the open-set scenario. As we will see, all previous solutions for the open-set scenario are based on SVM classifiers.

2.1 Approaches based on one-class classifiers

Schölkopf et al. [44] proposed an extension of SVM called the one-class SVM (OCSVM). This classifier is trained on just one known class. It finds the best margin with respect to the origin. This is the most reliable approach in cases where the access to a second class is very difficult or even impossible. As it focus only on the positive class of interest, this approach is suitable for the open-set scenario. Zhou and Huang [55], however, mention that the OCSVM has a limited use because it does not provide good generalization nor specialization. Several works dealing with OCSVMs try to overcome the problem of lack of generalization [6, 21, 27, 53].

Jin et al. [21] improved the OCSVM using subsets of negative samples as positive samples. They assume that training the classifier with negative classes are possible and may lead to good classification results. Despite its effectiveness, the authors present no theoretical explanation for their approach.

Strategies based on the combination of classifiers have also been proposed for one-class classification problems. One example is the cascade approach proposed by Cevikalp and Triggs [6]. The first stage of the cascade approach is a linear classifier that uses an intersection of affine hyperplanes to approximate the class of interest. Then, an OCSVM classifier based on a hypersphere model is used in the (non-kernelized) input space.

Wu and Ye [53] trained a classifier by partitioning the training data between normal

and outlier samples. A hypersphere containing most of the normal samples is constructed. This closed and tight boundary around the normal data is defined by a Gaussian kernel. The volume of such sphere is as small as possible. At the same time, the margin between the surface of this sphere and the outlier training data is as large as possible. An optimization problem is constructed to maximize the margin between the positive volume and the outliers.

Manevitz and Yousef [27] proposed a one-class classifier that selects some outliers based on a novel outlier detection approach and used the binary SVM classifier. The outlier detection approach tries to select the data points close to the origin. The authors justified that choosing the outliers as features close to the origin is a general idea, but the rationale for using the procedure is specific to the problem of document classification.

According to Scheirer et al. [43], the lack of generalization and specialization, combined with the use of testing protocols that consider only the closed-set scenario, are the main reasons for the small development of the OCSVM until now. Also, according to the same authors, although the OCSVM is inherently suitable for open-set classification problems, the potential of the binary SVM should not be neglected. They claim that if the knowledge of the negative classes is increased, a better SVM classifier can be defined. Some approaches in the literature introduced modifications on the SVM that can be, somehow, applied to open-set problems nevertheless the testing protocols used do not consider open-set classification scenarios.

2.2 Approaches based on binary classifiers

In [2], Bartlett and Wegkamp considered a binary classification problem in which a classifier can choose not to classify a sample. In that work, the *reject option* is presented to avoid a misclassification of a doubtful sample. Their work, however, does not consider the cases where test samples do not belong to a training class, i.e., the open-set scenario.

Malisiewicz et al. [26] took into account the cases where the access to the positive samples is limited and there are many negative samples. The method is based on training a separate linear SVM classifier for every positive sample in the training set. Each of the so-called Exemplar-SVMs is defined by a single positive sample and multiple negative ones. According to them, the use of several Exemplar-SVMs leads to a good generalization. Malisiewicz et al. [26] do not address the case in which training with certain negative classes is not possible. Another drawback of the method is that training many SVMs can be very expensive when implementing a multiclass SVM classifier based on the One-vs-All approach (the approach that allows *unknown* classification).

Jayadeva et al. [20] presented the Twin SVM (TWSVM), an approach that defines two nonparallel hyperplanes such that each hyperplane is close to samples of one of the two

classes and is distant from samples of the other class. According to the authors, TWSVMs yield better results than SVM. Similarly, Chew et al. [9] also handle parallel-hyperplanes constraints and propose the Modified Correlation Filters (MCF) to the problem of facial expression recognition. The MCF is a supervised binary classification algorithm inspired by correlation filters and SVMs. According to the authors, the resulting classifier can be accommodated to be much more robust against noise and outlying training samples.

2.3 Approaches used in similar problems

In the machine learning literature, the concept of zero-shot learning can be understood as the procedure of learning a classifier that must predict novel classes that are not present in the training set [32]. To cover this problem, researchers explore *knowledge transfer* approaches between object classes. As the information of the test classes cannot be learned at training phase, this information often must be added to the system by human effort [24].

The *knowledge transfer* between object classes has been accepted as a promising research venue towards scalable recognition. As presented by Rohrbach et al. [42], this is done by reusing acquired knowledge in the context of newly posed, but related recognition tasks. The zero-shot learning is also a way to deal with the need for increasing amounts of training data. As mentioned by Scheirer et al. [43], these approaches have formal definitions, but without constraints on smoothness or data accuracy.

Open-set recognition problems differ from the unsupervised and semi-supervised learning techniques as well. In the open-set scenario, it is necessary to classify the test sample as belonging to either one of the known (trained) classes or to the *unknown* "class." On the other hand, in an unsupervised learning procedure, the objective is only to group similar samples. The semi-supervised learning technique is also very different because in the open-set scenario, we are not interested in propagating labels from the known classes to the *unknown* ones. In the case of using the open-set solutions as a semi-supervised learning technique, the evaluation procedure must be significantly different, as the evaluation protocol must simulate the open-set scenario.

2.4 Approaches proposed for open-set problems

Some recent approaches have turned the attention to open-set problems directly and extended the SVM classifier to deal with the modified constraints of the open-set scenario [11, 12, 43]. As the original SVM's *risk minimization* is based only on the known classes (*empirical risk*), it can misclassify the *unknown* classes that can appear in the

testing phase. Differently, possible open-set solutions need to minimize the *risk of the unknown* [43].

Costa et al. [11, 12] presented a source camera attribution algorithm considering the open-set scenario and developed an extension of the SVM classifier. According to the authors, their method, called SVM with Decision Boundary Carving (SVMDBC), is suitable for recognition in open-set scenario. As the SVMDBC is a binary classifier extension and the authors presented no multiclass-from-binary version of the SVM using it, we can state that Costa et al. [11, 12] proposed a method that minimizes the *risk of false positive* instead of the *risk of the unknown*. The minimization of the *risk of false positive* of the SVMDBC method is done by moving the decision hyperplane found by the traditional SVM by a value ϵ inwards (possibly outwards) the positive class. The value of ϵ is defined by an exhaustive search to minimize the *training data error*. In our work, we present a multiclass-from-binary version of the SVMDBC using the One-vs-All approach for a comparison purpose. We call this multiclass-from-binary version as $\text{SVMDBC}_{\text{MCBIN}}$.

Scheirer et al. [43] introduced the 1-vs-Set Machine with a linear kernel formulation that can be applied to both binary and one-class SVMs. Also, the idea is to minimize the *risk of the unknown*, but in reality only the *risk of the false positive* is minimized, as the 1-vs-Set Machine is a binary classifier.

Similarly to Costa et al. [11, 12], Scheirer et al. [43] also move the original SVM hyperplane inwards the positive class, but now adding a parallel hyperplane "after" the positive samples aiming at decreasing the *open space of risk* of the positive class. The hyperplanes are initialized to contain all the positive samples between them. Then, a refinement step is performed to adjust the hyperplane to generalize or specialize the classifier according to the user *parameter pressures*. As noted by the authors, better results are usually obtained when the original SVM hyperplane is near to the positive boundary seeking a specialization and the added hyperplane is adjusted seeking generalization. Although the *open space of risk* is minimized by the second hyperplane for the positive class, the *open space* remains unbounded.

Scheirer et al. [43] also did not present a multiclass-from-binary for the classifier. Their experiments considered the partial results of binary classification problems only. Here, we present a multiclass-from-binary extension of the 1-vs-Set Machine using the One-vs-All approach for comparison purposes from now referred to as $\text{SVM1VS}_{\text{MCBIN}}$.

2.5 Remarks

Most of the discussed related work, can be somehow extended or adapted to be used for open-set recognition problems with some effort. The state-of-the-art methods, however, contain some drawbacks (e.g., the unbounded *open space of risk* and the limitation to

binary classification) that must be avoided in real-world open-set classification problems.

Chapter 3

Optimum-path forest classifier

The Optimum-Path Forest (OPF) classifier, originally proposed by Papa et al. [34, 36], is a graph-based classifier that was developed as a generalization of the Image Foresting Transform (IFT) [16] from the image domain to the feature space. Afterwards, the Optimum-Path Forest was proposed as a methodology and allowed the development of unsupervised [41] and semi-supervised [1] classification methods. However, in this work we refer to OPF as the supervised classifier [34, 36].

OPF classifier is similar to the well-known k-Nearest Neighbors (kNN) [4] algorithm for $k = 1$ [46, 47], inherently multiclass, and parameter independent. The OPF makes no assumption about the shapes of the classes and can support some degree of class overlapping [34]. It has shown good results in many classification problems [13, 14, 33, 34, 35, 37, 49].

In the OPF supervised classifier, the training samples are labeled graph nodes linked by arcs whose weights are distances between samples in the feature space. Each label is associated with a different class. The OPF has two phases: *fitting* and *prediction*. In the *fitting* phase, OPF finds a set of *prototypes* (which are training samples) and creates an optimum-path forest rooted in these *prototypes*. In the *prediction* phase, a test sample receives the same label of the *prototype* that offers the lowest cost (the most strongly connected *prototype*), according to some cost function defined a priori. In the following sections, we describe these two phases in more detail.

3.1 Fitting phase

Let Z be a set of m labeled training samples divided into n classes, as follows:

$$Z = \{(s_1, \lambda(s_1)), (s_2, \lambda(s_2)), \ldots, (s_m, \lambda(s_m))\}, \tag{3.1}$$

where s_i, $i = 1, \ldots, m$, is a feature vector and $\lambda(s_i)$ is the actual class of s_i, i.e., $\lambda(s_i) \in \mathcal{L} = \{\ell_1, \ell_2, \ldots, \ell_n\}$, where $\mathcal{L}$ is a set of labels representing n distinct classes.

The weight $w(s,t)$ of the arc (s,t) in the complete graph $A = (Z \times Z)$ is the distance (the Euclidean distance, in this work) between samples s and t in the feature space. For computing the set S of *prototypes*, the supervised classifier computes the Minimum Spanning Tree (MST) M of A. Every sample s is considered a *prototype* when s is connected to some sample t in M such that $\lambda(s) \neq \lambda(t)$. The k *prototypes* form the set $S \subset Z$ of *prototypes*:

$$S = \{(s_1^p, \lambda(s_1^p)), \ldots, (s_k^p, \lambda(s_k^p))\}, \tag{3.2}$$

An important difference of OPF when compared to other classifiers relies on the *path cost function* used. A path $\pi_{s,t} = \langle s = s_1, s_2, \ldots, s_p = t \rangle$ with length p from source node s to target node t is defined as a sequence of nodes, such that there is an arc between s_i and s_{i+1}. A path $\pi_{s,s} = \langle s \rangle$ is a trivial path. The OPF *path cost function* f is given by the maximum arc weight w in the path, represented by:

$$f(\langle s \rangle) = \begin{cases} 0 & \text{if } s \in S \\ +\infty & \text{otherwise,} \end{cases}$$

$$f(\pi_{s,u} \cdot (u,t)) = \max\{f(\pi_{s,u}), w(u,t)\}. \tag{3.3}$$

where $\pi_{s,u} \cdot (u,t)$ denotes the concatenation of a path $\pi_{s,u}$ ending at u and an arc (u,t).

A path $\pi_{s,t}$ with terminus t is said optimum if $f(\pi_{s,t}) \leq f(\pi_{u,t})$ for all $u \in Z$. Based on the *path cost function* f, the supervised classifier assigns an optimum path $P^*(t)$ for every sample $t \in Z$. Note that every optimum path is from $s \in S$, $S \subset Z$, according to the Equation 3.3. Then, every sample t holds its minimum cost $C(t)$:

$$C(t) = f(P^*(t)) \tag{3.4}$$

Algorithm 1 is an extension of the IFT algorithm [16] from the image domain to the feature space specialized for f. The optimum-path forest P^* and the cost map C are obtained using Algorithm 1. This algorithm also returns the label map L that maintains the information of the label that each sample $s \in Z$ holds in the trained classifier. When the set S of *prototypes* is obtained based on the MST, as described earlier, it is guaranteed that $L(s) = \lambda(s)$ for all $s \in Z$ [34]. Based on P^*, the root $\mathcal{R}(t) \in S$ of a sample t is obtained by following the predecessors backwards along the path. Note that it is guaranteed that $L(t) = \lambda(\mathcal{R}(t))$ for all $t \in Z$.

3.2 Prediction phase

The classification of a test sample s is accomplished by creating arcs linking s to all nodes of the optimum-path forest (OPF). The *prediction* phase consists in an optimization

Algorithm 1 OPF algorithm.

Require: Set of training samples Z
Require: Set of prototypes S
Ensure: $S \subset Z$
Output: Optimum-path forest P
Output: Cost map C
Output: Label map L

1: **for all** $s \in Z \backslash S$ **do**
2: $C(s) \leftarrow +\infty$
3: **end for**
4: **for all** $s \in S$ **do**
5: $C(s) \leftarrow 0$; $P(s) \leftarrow nil$; $L(s) \leftarrow \lambda(s)$
6: Insert s in Q
7: **end for**
8: **while** Q is not empty **do**
9: Remove s from Q such that $C(s) \leq C(t)$ for all $t \in Q$, $s \neq t$
10: **for all** $t \in Z$ such that $s \neq t$ **do**
11: $cst \leftarrow max\{C(s), w(s,t)\}$
12: **if** $cst < C(t)$ **then**
13: **if** $C(t) \neq +\infty$ **then**
14: Remove t from Q
15: **end if**
16: $P(t) \leftarrow s$; $L(t) \leftarrow L(s)$; $C(t) \leftarrow cst$
17: Insert t in Q
18: **end if**
19: **end for**
20: **end while**

problem in which the objective is to find the *prototype* s^p whose path to s offers the lowest cost, i.e.,

$$s^p = \arg\min_{\forall s^p \in S}\{f(\pi_{s^p,s})\}, \tag{3.5}$$

or simply

$$s^p = \mathcal{R}(\arg\min_{\forall t \in Z}\{\max\{C(t), w(s,t)\}\}). \tag{3.6}$$

Note that for the prediction, not all training samples need to be verified if they are maintained sorted by its costs [35].

Given the found *prototype* s^p, the class prediction process consists in assigning the label of s^p to s, i.e.,

$$L(s) = \lambda(s^p). \tag{3.7}$$

For a better understanding, Figure 3.1 depicts OPF's operation in a toy–example scenario of five training samples: s_1 and s_2 from the blue class; and s_3, s_4, and s_5 from the red class. Figure 3.1a depicts the complete graph A. In Figure 3.1b, the MST M is calculated and s_2 and s_3 are chosen as *prototypes*. These steps correspond to the *fitting* phase. In the *prediction* phase illustrated in Figure 3.1c, the test sample s is assigned to the tree rooted in s_3 because it offers the lowest cost, based on Equation 3.3. Node s is therefore classified as belonging to the red class.

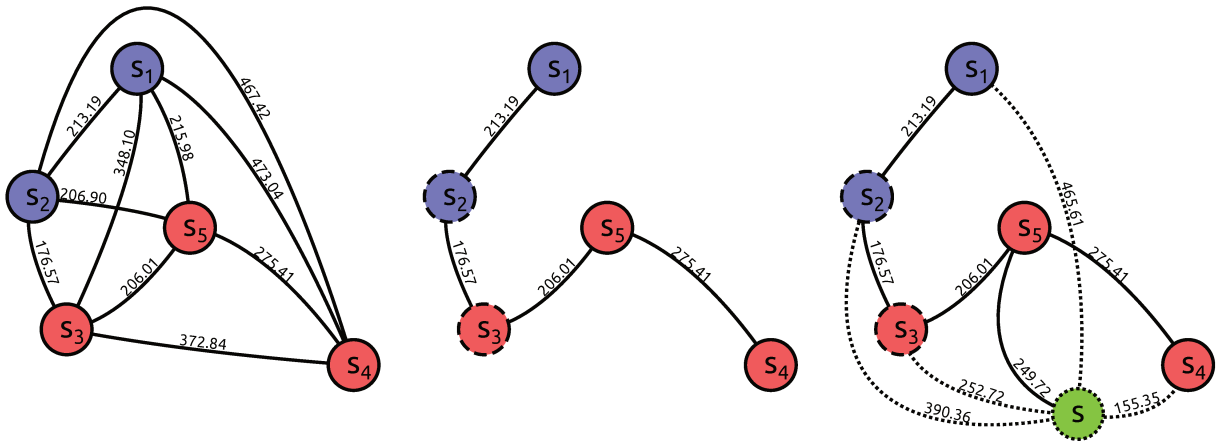
3.3 Properties of the optimum-path forest classifier

In this section, we present some properties of the OPF when analyzing it for the open-set scenario. There are important properties of the OPF we need to highlight for a better understanding of the proposed OSOPF, and its main differences when compared to other available open-set approaches such as $\text{SVM}_{\text{MCBIN}}$, $\text{SVMDBC}_{\text{MCBIN}}$, and $\text{SVM1VS}_{\text{MCBIN}}$.

First of all, suppose a simplified version of OPF that is simply the OPF itself as a binary classifier. We call this classifier as OPF_{BIN}. The OPF_{BIN} can be obtained by simply using the OPF restricted to two classes. For reference, we use OPF_{BIN} only to explicit that we are using the OPF with only two classes for training.

Let $\text{OPF}_{\text{MCBIN}}$ be the multiclass-from-binary version of the OPF. That is, the $\text{OPF}_{\text{MCBIN}}$ decomposes the general multiclass problem into n binary ones, where n is the number of *available classes* for training. In this case, we suppose the $\text{OPF}_{\text{MCBIN}}$ using the One-vs-All approach.

The One-vs-All approach for the $\text{OPF}_{\text{MCBIN}}$ can be accomplished in the same way it is done for the $\text{SVM}_{\text{MCBIN}}$: (1) if none of the OPF_{BIN} classifies an input as *positive*, then the $\text{OPF}_{\text{MCBIN}}$ classifies it as *unknown*; (2) if one or more OPF_{BIN} classifies an input as



(a) (b) (c)

Figure 3.1: Depiction of the Optimum-Path Forest classifier. (a)-(b) *fitting* phase. (c) *prediction* phase. The numeric values on the arcs indicate the distance between samples in the feature space. The blue samples (s_1 and s_2) and the red samples (s_3, s_4, and s_5) refer to different classes. (a) Complete graph computation based on training samples. (b) Choice of the *prototypes* based on the Minimum Spanning Tree. Dashed samples are *prototypes*. (c) Classification of the green-dotted test sample s. The dotted arcs depict the arcs that create possible paths from a *prototype* to s. The non-dotted arc adjacent to s belongs to the best path. In this example, s is assigned to the red class.

positive, then the $\text{OPF}_{\text{MCBIN}}$ uses the best OPF_{BIN}, according to the *path cost function* of Equation 3.3, to decide the positive label of that input test sample.

Now, let us present the aforementioned properties of the OPF classifier. We start with a proposition:

Proposition 1. *In the original formulation of the OPF, no test sample is classified as* unknown.

Proof. The proof is trivially obtained from Equations 3.5 and 3.7. Together they show that one of the known trained *prototypes* is chosen and its label is always assigned to a given test sample in the *prediction* phase. □

As the OPF is an inherently closed-set classifier, the $\text{OPF}_{\text{MCBIN}}$ is also a closed-set classifier. In the following propositions, we show that OPF and $\text{OPF}_{\text{MCBIN}}$ are equivalent.

Proposition 2. *Exactly one OPF_{BIN} of the OPF_{MCBIN} classifies a test sample as* positive.

Proof. First, we are going to prove that at most one OPF_{BIN} classifies an input as *positive*. Then we are going to prove that at least one OPF_{BIN} classifies an input as *positive*.

Let $\text{OPF}^{\text{n}}_{\text{BIN}}$ be the OPF_{BIN} of the $\text{OPF}_{\text{MCBIN}}$ trained with the n^{th} class as a *positive* class. Let the classification result of the test sample s with $\text{OPF}^{\text{n}}_{\text{BIN}}$ be *positive*. Then, $L(\mathcal{R}(s))$ is *positive*. We call this *prototype* node $\mathcal{R}(s)$ as s^p. We are going to show that $L(\mathcal{R}(s))$ is *positive* for no other OPF_{BIN}. The OPF behavior creates a Voronoi tessellation in the feature space, as OPF is based on the *path cost function* of Equation 3.3 (that is based on distances). The *positive* class of $\text{OPF}^{\text{n}}_{\text{BIN}}$ is a *negative* class in other OPF_{BIN}s when using the One-vs-All approach. For any other OPF_{BIN}, there are two possible cases: (1) s^p continues to be a *prototype* or (2) s^p is no longer a *prototype*. Analyzing case (1): If s^p continues to be a *prototype*, the best path of the OPF_{BIN} under consideration is equal to the best path of the $\text{OPF}^{\text{n}}_{\text{BIN}}$. But now $L(s^p)$ is *negative* and s is classified as *negative*. Note that the Minimum Spanning Tree (MST) of training phase of all OPF_{BIN}s are equal, as all available training samples are used. Analyzing case (2): If s^p is no longer a *prototype*, the best path continues passing through s^p to reach a new *prototype*. In the OPF_{BIN} under consideration, s^p is *negative* and to reach a *positive* subgraph, the path needs to pass through a *negative prototype* and then a *positive prototype*. As the *negative prototype* is reached before, $\mathcal{R}(s)$ is the *negative prototype* and the OPF_{BIN} under consideration classifies the input as *negative*.

Now we must show that at least one OPF_{BIN} classifies an input as *positive*. Let the classification result of the test sample s with $\text{OPF}^{\text{n}}_{\text{BIN}}$ be *negative*. Then, $L(\mathcal{R}(s))$ is *negative*. We are going to show that $L(\mathcal{R}(s))$ is *positive* for at least one OPF_{BIN} other than $\text{OPF}^{\text{n}}_{\text{BIN}}$. The best path π_b from a (*negative*) *prototype* to s in $\text{OPF}^{\text{n}}_{\text{BIN}}$ passes through samples of one or more training classes (considering the real class information). In some cases, the path reaches the same training class more than once, but it does not influence our proof. Let s^n be the neighbor of s in π_b. Let s^p be the last sample in π_b with the same label of s^n such that all samples between s^n and s^p have the same label of s^n and s^p. Let π_s be the subpath of π_b from s to s^p. Now, it is easy to see that when the class $\lambda(s^n)$ is the *positive* class of one OPF_{BIN}, π_s is the best path and s^p is a *positive prototype*. Then, s is classified as *positive* by one OPF_{BIN}. □

Figure 3.2 depicts the Proposition 2 situation. A test sample in any possible point in the feature space is in the *positive* region of some OPF_{BIN}.

Proposition 3. *The OPF_{MCBIN} classifier is equivalent to the OPF classifier.*

Proof. For any test sample s, exactly one binary classifier in the $\text{OPF}_{\text{MCBIN}}$ will classify s as *positive* (see Proposition 2). The positive class is the same class to which the OPF classifier would assign s because OPF behavior creates a Voronoi tessellation in the feature space. □

According to the Propositions 2 and 3, Figure 3.2 can also depict the inherently multiclass OPF classifier.

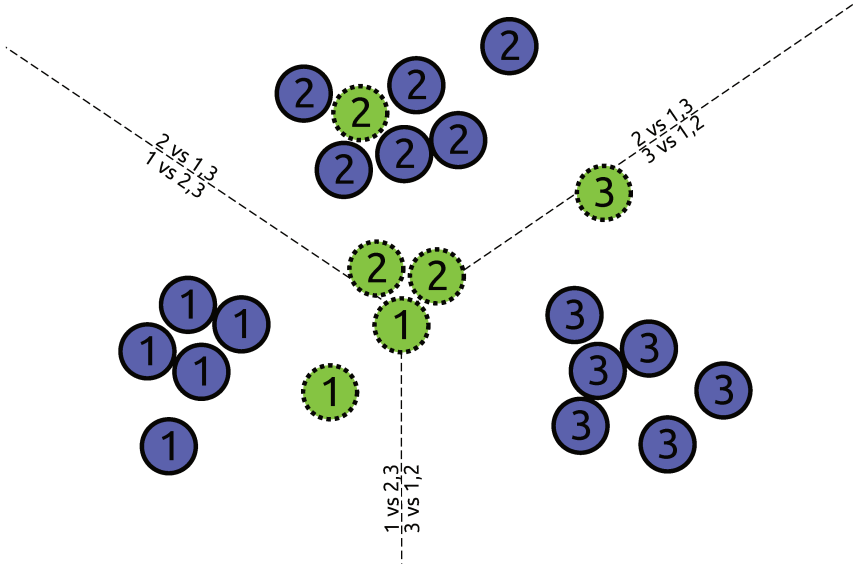


Figure 3.2: Example of the multiclass Optimum-Path Forest (OPF) classifier using One-vs-All approach ($\text{OPF}_{\text{MCBIN}}$). The multiclass-from-binary $\text{OPF}_{\text{MCBIN}}$ is equivalent to the original multiclass OPF. The dotted–green samples are test samples. A test sample is never classified as *unknown*. Notice that the decision boundaries for a binary classifier is not linear in practice.

Here we presented the $\text{OPF}_{\text{MCBIN}}$ for a better understanding of the OPF classifier's behavior and also for a better understanding of our proposed OSOPF classifier for dealing with open-set problems, which we describe in details in the next chapter.

Chapter 4

Open-set optimum-path forest classifiers

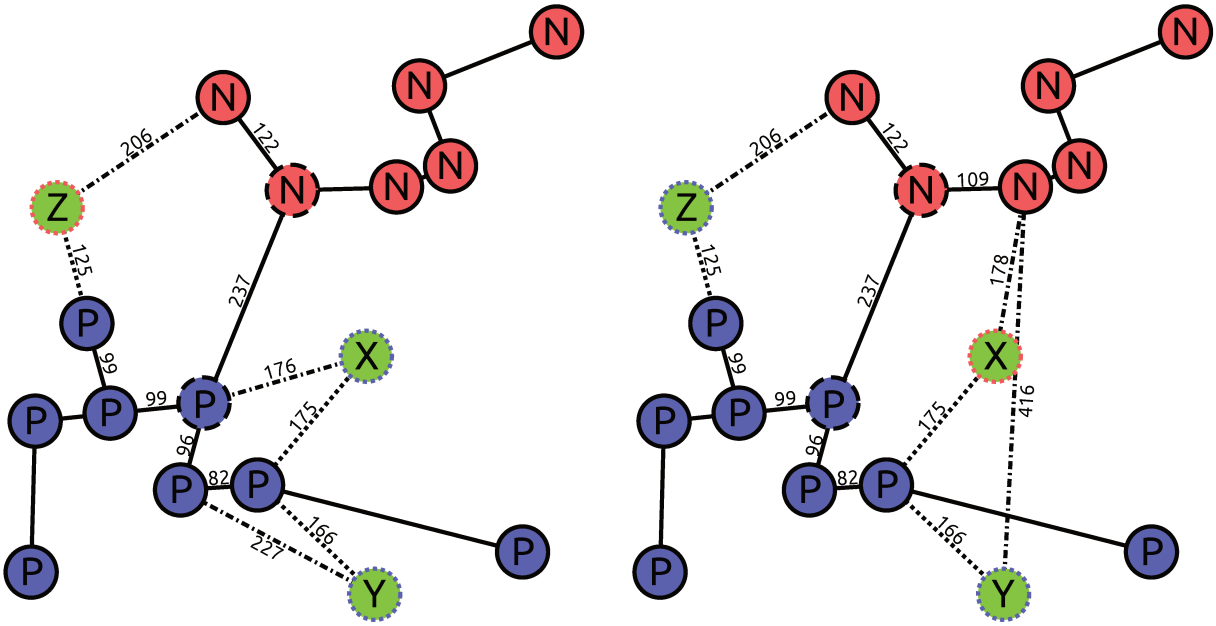
In this work, OSOPF stands for the Open-set Optimum-Path Forest classifier. In this chapter, we present two proposed open-set classifiers. Both methods are similar in their structure, but different in their behavior. We present the methods as multiclass-from-binary classifiers. Then, in Section 4.3, we show how the multiclass-from-binary methods are extended to inherently multiclass ones.

First, we present the binary version of the two proposed classifiers: the $\text{OSOPF}^1_{\text{BIN}}$ and $\text{OSOPF}^2_{\text{BIN}}$. Both binary classifiers are similar to the OPF_{BIN}, but with an additional verification step.

4.1 Proposed binary open-set classifiers

Given a test sample s, the $\text{OSOPF}^1_{\text{BIN}}$ computes the paths from *prototypes* with the best and second best classification costs (see cost function in Equation 3.3). Next, it checks the labels of the identified *prototypes* in each case. If both paths yield *positive prototypes*, then s is classified as *positive*; if some path is rooted in a *negative prototype*, s is classified as *negative*. Figure 4.1a illustrates the $\text{OSOPF}^1_{\text{BIN}}$ classifier. $\text{OSOPF}^1_{\text{BIN}}$ avoids classifying as *positive* the "ambiguous" samples, i.e., the samples that are relatively not close enough to *prototypes* of the *positive* class.

The $\text{OSOPF}^2_{\text{BIN}}$, in turn, computes the paths with the best classification costs from *positive* and *negative prototypes*. Next, it computes the ratio R between the best *positive* classification cost and the best *negative* classification cost. If R is less than or equal to a specified threshold T, for $0.0 < T < 1.0$, the test sample s is classified as *positive*. Otherwise, s is classified as *negative*. The R ratio can be seen as the uncertainty level of the classification. The smaller the value of R, the more confident is the classifier in



(a) The *prediction* phase of the $\text{OSOPF}^1_{\text{BIN}}$. The test samples X and Y are classified as *positive* because both acquired paths are rooted in *positive prototypes*. The sample Z is classified as *negative* because the path of the second best classification cost has a *negative prototype*.

(b) The *prediction* phase of the $\text{OSOPF}^2_{\text{BIN}}$. Supposing the threshold T of $\text{OSOPF}^2_{\text{BIN}}$ equals to 0.80, the test sample X is classified as *negative* because its ratio R is $0.98 > T$. The samples Y and Z are classified as *positive* because their R is 0.40 and 0.61, respectively, i.e., both are less than T.

Figure 4.1: Binary version of the Open-set Optimum-Path Forest classifiers: (a) The $\text{OSOPF}^1_{\text{BIN}}$ and (b) The $\text{OSOPF}^2_{\text{BIN}}$. Red samples Ns are *negative* samples. Blue samples Ps are *positive* samples. The green dotted samples X, Y, and Z are test samples. Dashed samples are prototypes. The border's color of the test samples indicates in which class it is classified by the method: blue for *positive* and red for *negative*. The dotted arcs adjacent to a test sample s are arcs to the neighbor sample of s in the path of the best classification cost for Figure (a) and the path of the best *positive* classification cost for Figure (b). The dot-dash arcs are adjacent to the neighbor of s in the path of the second best classification cost for Figure (a) and the path of the best *negative* classification cost for Figure (b). Normal arcs depict the arcs acquired in the *fitting* phase of the OPF.

assigning the sample to the *positive* class. Figure 4.1b illustrates the $\text{OSOPF}^2_{\text{BIN}}$ binary classifier.

In the case of the $\text{OSOPF}^2_{\text{BIN}}$, a test sample s faraway from the training samples is also classified as *negative*. That happens because R tends to 1 as both the cost from the closest *positive* and the cost from the closest *negative prototypes* increase. Probably this is the main reason that a multiclass-from-binary extension using this proposed method

obtained better results, as we will see in Chapter 5.

Proposition 4. *All test samples that would be classified as* negative *by the OPF_{BIN} are classified as* negative *by the $OSOPF^1_{BIN}$ and $OSOPF^2_{BIN}$.*

Proof. Let s be a test sample that would be classified as *negative* by the OPF_{BIN}. For the $\text{OSOPF}^1_{\text{BIN}}$: if s would be classified as *negative* by the OPF_{BIN}, the best path acquired by the $\text{OSOPF}^1_{\text{BIN}}$ is rooted in a *negative prototype*. As either the best path or the second best path acquired by the $\text{OSOPF}^1_{\text{BIN}}$ is *negative*, s is classified as *negative* by the $\text{OSOPF}^1_{\text{BIN}}$. For the $\text{OSOPF}^2_{\text{BIN}}$: if s would be classified as *negative* by the OPF_{BIN}, the path to the *negative* prototype has the lowest cost. As *positive* cost is greater than or equal to the *negative* cost, R is greater than or equal to 1. As $0.0 < T < 1.0$, then $R > T$. Consequently, $\text{OSOPF}^2_{\text{BIN}}$ classifies s as *negative*. □

Proposition 4 shows that $\text{OSOPF}^1_{\text{BIN}}$ and $\text{OSOPF}^2_{\text{BIN}}$ increase only the *true negative* compared to the OPF_{BIN}. In Section 4.2, we show that this property helps the multiclass-from-binary classifier to correctly identify *unknown* samples.

4.2 Multiclass-from-binary proposed classifiers

In this section, we present the following multiclass-from-binary classifiers: $\text{OSOPF}^1_{\text{MCBIN}}$ and $\text{OSOPF}^2_{\text{MCBIN}}$ that uses the $\text{OSOPF}^1_{\text{BIN}}$ and $\text{OSOPF}^2_{\text{BIN}}$, respectively.

In Section 3.3, we showed that exactly one OPF_{BIN} of the $\text{OPF}_{\text{MCBIN}}$ classifies an input sample as *positive*. In our proposed methods, all $\text{OSOPF}_{\text{BIN}}$ classifiers that compose the $\text{OSOPF}_{\text{MCBIN}}$ (for both $\text{OSOPF}^1_{\text{MCBIN}}$ and $\text{OSOPF}^2_{\text{MCBIN}}$) can classify a sample as *negative*. While the $\text{OPF}_{\text{MCBIN}}$ is inherently closed set (no test sample is classified as *unknown*), it is possible to the $\text{OSOPF}_{\text{MCBIN}}$ to classify input samples as *unknown*. Note that, similarly to the $\text{OPF}_{\text{MCBIN}}$, in the $\text{OSOPF}^1_{\text{MCBIN}}$, at most one $\text{OSOPF}^1_{\text{BIN}}$ classifies an input sample as *positive*. The same is not true for the $\text{OSOPF}^2_{\text{MCBIN}}$, because in some special cases more than one $\text{OSOPF}^2_{\text{BIN}}$ can classify the input as *positive* (see Section 4.3.2 for explanation).

Proposition 5. *If the OPF_{MCBIN} classifies s with class label ℓ_n, the $OSOPF^1_{MCBIN}$ and $OSOPF^2_{MCBIN}$ classify s with class label ℓ_n or* unknown.

Proof. Let $\text{OPF}^{\text{n}}_{\text{BIN}}$ be the OPF_{BIN} of the $\text{OPF}_{\text{MCBIN}}$ trained with the n^{th} class as a *positive* class. Let $\text{OSOPF}^{\text{n}}_{\text{BIN}}$ be the $\text{OSOPF}_{\text{BIN}}$ of the $\text{OSOPF}_{\text{MCBIN}}$ trained with the n^{th} class as a *positive* class. There are two possible results in classification of s by the $\text{OSOPF}_{\text{MCBIN}}$: (1) *unknown* or (2) one of the *available classes*. Analyzing for (1): the proposition is trivially valid. Analyzing for (2): we must show that $L(s) = \ell_n$ for $\text{OPF}_{\text{MCBIN}}$ implies

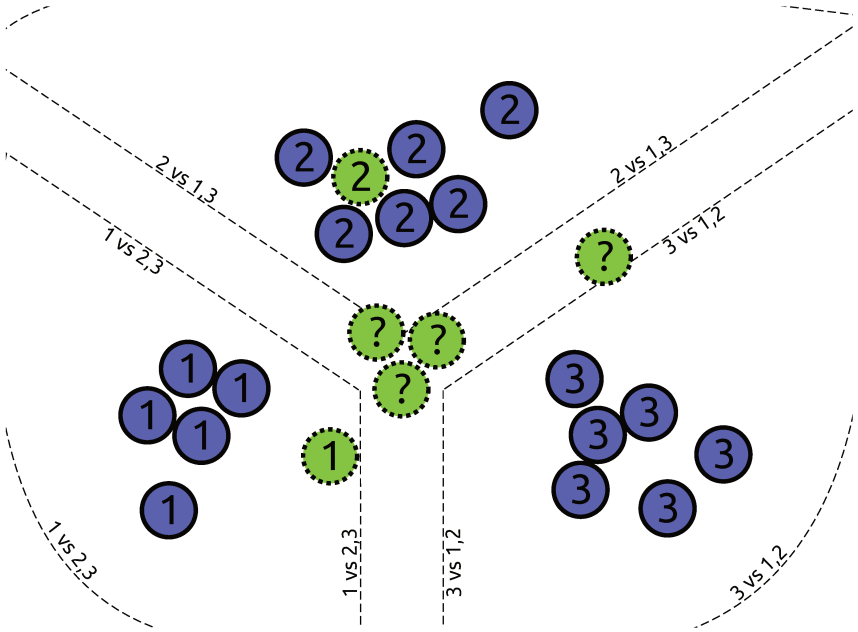


Figure 4.2: Example of the multiclass-from-binary Open-set Optimum-Path Forest ($\text{OSOPF}_{\text{MCBIN}}$) classifier with One-vs-All approach. The $\text{OSOPF}_{\text{MCBIN}}$ allows for *unknown* classification. The curved boundaries only exist in the case of the $\text{OSOPF}^{2}_{\text{MCBIN}}$.

$L(s) = \ell_n$ for $\text{OSOPF}_{\text{MCBIN}}$. Let $\text{OSOPF}^{\text{n}}_{\text{BIN}}$ be the classifier with *positive* result (with better cost in case of $\text{OSOPF}^{2}_{\text{BIN}}$). We know that $\text{OPF}_{\text{MCBIN}}$ classified s as ℓ_n, then it is sure that the $\text{OSOPF}^{\text{n}}_{\text{BIN}}$ of the $\text{OSOPF}_{\text{MCBIN}}$ has the smallest cost for the *positive class*. As we know that $\text{OSOPF}_{\text{MCBIN}}$ classified s as one of the available classes, we know that this class is the *positive* class of the $\text{OSOPF}^{\text{n}}_{\text{BIN}}$, i.e., this class is ℓ_n. □

The general behavior of the $\text{OSOPF}_{\text{MCBIN}}$ is presented in Figure 4.2. Note that the curve boundaries in Figure 4.2 only exist in case of the $\text{OSOPF}^{2}_{\text{MCBIN}}$, as mentioned in Section 4.1 that for $\text{OSOPF}^{2}_{\text{BIN}}$ faraway samples are classified as *negative*.

4.3 Inherently multiclass version of the proposed classifiers

The implementation of a multiclass-from-binary classifier has an inconvenience: as the number n of *available classes* increases, the number of binary classifiers also increases and, consequently, the efficiency of the classification process is affected. Rocha and Goldenstein [40] showed that it becomes computationally intensive to use multiclass binary-based classifiers when n is high, unless error correcting output codes (ECOC) solutions

are used. These solutions, however, rely on random partitions of classes and not always lead to good classification results.

In this section, we present two proposed inherently multiclass Open-set Optimum-Path Forest classifiers: OSOPF^1 and OSOPF^2. The OSOPF^1 has the same behavior of the multiclass-from-binary OSOPF$^1_{\text{MCBIN}}$ and the OSOPF^2 has a similar behavior of the multiclass-from-binary OSOPF$^2_{\text{MCBIN}}$ previously introduced. The OSOPF^1 and OSOPF^2 are extended from the binary OSOPF$^1_{\text{BIN}}$ and OSOPF$^2_{\text{BIN}}$, respectively.

4.3.1 OSOPF^1

OSOPF^1 is based on the agreement of the labels of the two best paths. The fitting phase of the OSOPF^1 is identical to the fitting phase of the OPF. Its *prediction* phase works as follows: it computes the paths with the best and second best classification costs and checks the labels of the associated *prototypes*. If both paths lead to the same label, this label is assigned to the test sample s. Otherwise, s is classified as *unknown*. We present the OSOPF^1 in Algorithm 2.

Algorithm 2 Fitting and prediction phases of the OSOPF^1.

Require: Set of training samples Z
Require: Test sample s
Ensure: $s \notin Z$
1: Let ℓ_0 be the *unknown* label
2: $OPF \leftarrow$ Traditional OPF classifier using Z
3: $\pi^1 \leftarrow$ Best path to s in OPF according to f
4: $s^1 \leftarrow \mathcal{R}(\pi^1)$
5: $\pi^2 \leftarrow$ Best path to s in OPF such that its root is not equal to s^1
6: $s^2 \leftarrow \mathcal{R}(\pi^2)$
7: **if** $L(s^1) = L(s^2)$ **then**
8: $\quad L(s) \leftarrow L(s^1)$
9: **else**
10: $\quad L(s) \leftarrow \ell_0$
11: **end if**

Proposition 6. *OSOPF^1 and OSOPF$^1_{MCBIN}$ have the same behavior.*

Proof. We must prove that (1) if OSOPF$^1_{\text{MCBIN}}$ classifies a test sample s as *unknown*, then OSOPF^1 also classifies s as *unknown* and (2) if OSOPF$^1_{\text{MCBIN}}$ classifies s as ℓ_n, then OSOPF^1 also classifies s as ℓ_n.

In OSOPF^1, let s^p be the *prototype* of the best path to s. Let $L(s^p)$ be ℓ_n. Let OSOPF$^{\text{n}}_{\text{BIN}}$ be the OSOPF$^1_{\text{BIN}}$ of the OSOPF$^1_{\text{MCBIN}}$ trained with the class ℓ_n as a *positive*

class. Analyzing for (1): as $\text{OSOPF}^1_{\text{MCBIN}}$ classifies s as *unknown*, every $\text{OSOPF}^1_{\text{BIN}}$ of the $\text{OSOPF}^1_{\text{MCBIN}}$ classifies s as *negative*, i.e., for each $\text{OSOPF}^1_{\text{BIN}}$, either the best path or the second best path to s is rooted in a *negative prototype*. Then, for the $\text{OSOPF}^{\text{n}}_{\text{BIN}}$ either the best path or the second best path to s is *negative*. As we known that the best path of $\text{OSOPF}^{\text{n}}_{\text{BIN}}$ is rooted in s^p such that $L(s^p) = \ell_n$ (that is the *positive* class of $\text{OSOPF}^{\text{n}}_{\text{BIN}}$), we known that the second best path is rooted in a *negative prototype*. Consequently, the second best path to s in OSOPF^1 is rooted in a class other than ℓ_n, leading to an *unknown* classification, according to Algorithm 2. Analyzing for (2): as $\text{OSOPF}^1_{\text{MCBIN}}$ classifies s as ℓ_n, both the best path and the second best path of $\text{OSOPF}^{\text{n}}_{\text{BIN}}$ is *positive*. Consequently, the second best path to s in OSOPF^1 is rooted in the same class of the best path, i.e., ℓ_n. Then OSOPF^1 classifies s as ℓ_n. □

4.3.2 OSOPF^2

OSOPF^2 is based on the cost ratio of the two best paths of different classes. The fitting phase of the OSOPF^2 is identical to the fitting phase of the OPF, except for the *parameter optimization* phase used to find the best value for the threshold T. Similarly to the $\text{OSOPF}^2_{\text{BIN}}$, for the OSOPF^2 we compute the ratio R between the best classification cost for test sample s and the best classification cost to the second nearest class (according to the *path cost function* of Equation 3.3). If R is less than or equal to the specified threshold T, $0.0 < T < 1.0$, s is classified with the same label of the *prototype* s^p of the path that yields the best classification cost. Otherwise, it is classified as *unknown*, i.e.,

$$L(s) = \begin{cases} L(s^p) & \text{if } R \leq T \\ \ell_0 & \text{if } R > T. \end{cases}$$

where ℓ_0 is the *unknown* label. We present the OSOPF^2 in Algorithm 3.

Differently from the OSOPF^1, the OSOPF^2 is not equivalent to its multiclass-from-binary version. In fact, the behavior is similar, but in specific regions of the feature space the results differ depending on the threshold T and the shape of the graph created in the OPF classifier. In Figure 4.3 we depict a case in which a test sample that would be classified as *unknown* by the OSOPF^2 would be classified as positive by two binary classifiers that compose the $\text{OSOPF}^2_{\text{MCBIN}}$. In Figure 4.3a, for the OSOPF^2, the class 1 is the best for the test sample and class 3 is the second best one. In that figure, we consider that the test sample would be classified as *unknown* by the OSOPF^2, as the cost to the best class and the cost to the other best class is similar. Supposing that the OSOPF^2 and $\text{OSOPF}^2_{\text{MCBIN}}$ would have the same behavior, then every binary classifiers of the $\text{OSOPF}^2_{\text{MCBIN}}$ should classify the test sample as negative. By analyzing the Figures 4.3b and 4.3d we cannot ensure that. In Figure 4.3b, the cost to the positive class is the same

Algorithm 3 Fitting and prediction phases of the OSOPF^2.

Require: Set of training samples Z
Require: Test sample s
Ensure: $s \notin Z$
1: Let ℓ_0 be the *unknown* label
2: $T \leftarrow$ Threshold from *parameter optimization* procedure using Z
3: $OPF \leftarrow$ Traditional OPF classifier using Z
4: $\pi^1 \leftarrow$ Best path to s in OPF according to f
5: $\pi^2 \leftarrow$ Best path to s in OPF such that its root's class is not equal to $L(\mathcal{R}(\pi^1))$
6: $R \leftarrow f(\pi^1)/f(\pi^2)$
7: **if** $R \leq T$ **then**
8: $\quad L(s) \leftarrow L(\mathcal{R}(\pi^1))$
9: **else**
10: $\quad L(s) \leftarrow \ell_0$
11: **end if**

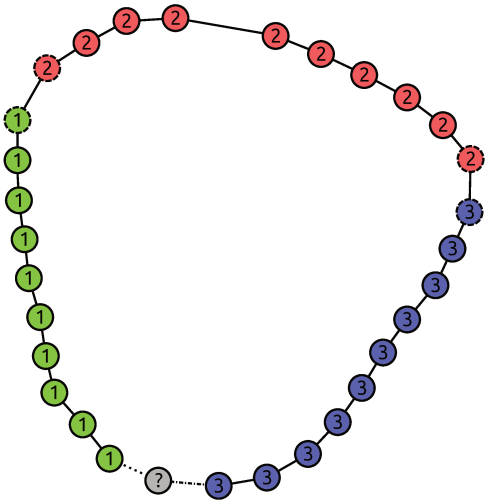
of the cost to the best class in Figure 4.3a. But the cost to the negative class is greater than the cost to the other best class of Figure 4.3a. Consequently, the ratio R for the binary classifier of the Figure 4.3b is smaller than ratio R of the OSOPF^2 and can be smaller than threshold T. The same analysis can be accomplished to the binary classifier of Figure 4.3d.

In fact, we experimentally compared the OSOPF^2 and OSOPF$^2_{\text{MCBIN}}$ and generated the *decision boundaries* for these classifiers. The *decision boundary* of a class defines the region in which a possible test sample will be classified as belonging to that class. In Figure 4.4 we present the *decision boundaries* comparing OSOPF^2 and OSOPF$^2_{\text{MCBIN}}$. Comparing Figures 4.4a and 4.4b we can see a small difference in the *decision boundaries* of the two classifiers in the bottom part of the region between the green and blue classes for $T = 0.5$. The difference in the same region happens when the threshold is 0.8 (compare Figures 4.4c and 4.4d). For $T = 0.8$, we can see in Figure 4.4e that in a small region of the feature space (the gray region) the test sample would be classified as positive by two binary classifiers of the OSOPF$^2_{\text{MCBIN}}$.

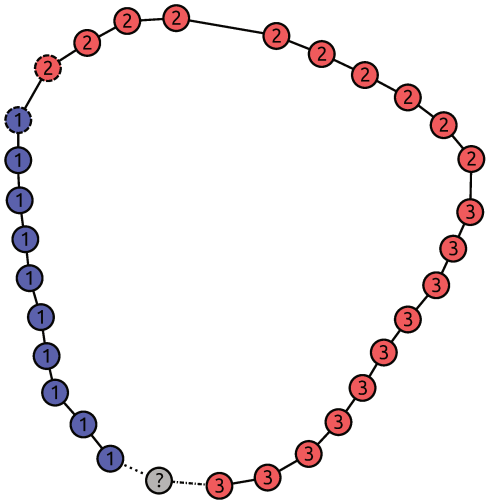
Despite the difference between OSOPF^2 and OSOPF$^2_{\text{MCBIN}}$ in a small region of the feature space, we can ensure that OSOPF^2 and OSOPF$^2_{\text{MCBIN}}$ have the same behavior when the test sample is faraway from the training samples in the feature space.

Proposition 7. *OSOPF^2 and OSOPF$^2_{MCBIN}$ have the same behavior when the distance from the test sample to the nearest training sample is greater than or equal to the highest arc within the Optimum-Path Forest.*

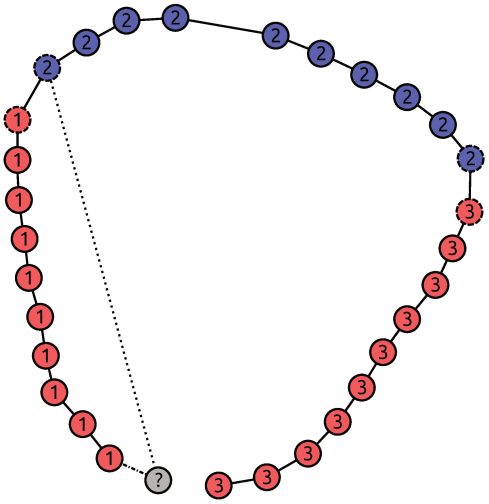
Proof. We must prove that (1) if OSOPF$^2_{\text{MCBIN}}$ classifies a test sample s as *unknown*,



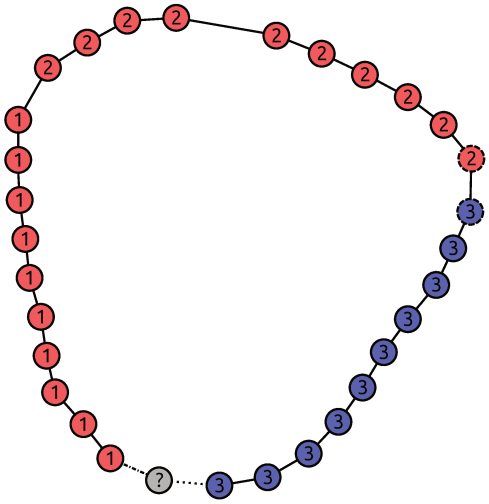
(a) OSOPF^2.



(b) Binary classifier of the OSOPF$^2_{\text{MCBIN}}$ trained with class 1 as positive.

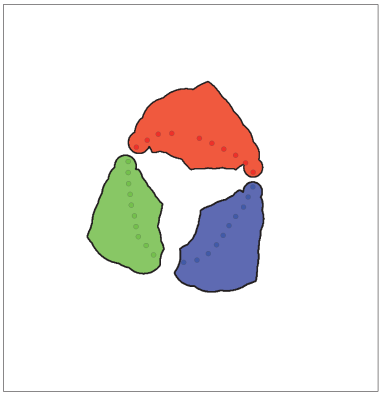


(c) Binary classifier of the OSOPF$^2_{\text{MCBIN}}$ trained with class 2 as positive.

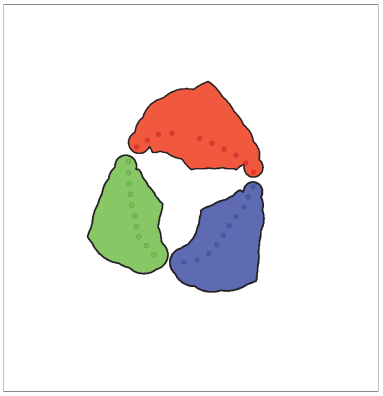


(d) Binary classifier of the OSOPF$^2_{\text{MCBIN}}$ trained with class 3 as positive.

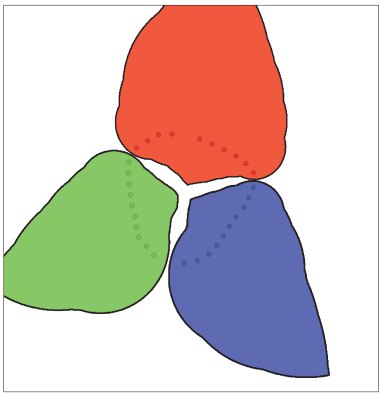
Figure 4.3: Depiction of the difference between OSOPF^2 and OSOPF$^2_{\text{MCBIN}}$. The dashed samples are *prototypes*. Figure (a) depicts the classification of the test sample by the OSOPF^2 classifier. The dotted arc indicate the best path from a *prototype*. The dot-dash arc indicate the best path from a *prototype* of a class different to the first one. Figures (b)-(d) depict the classification of the test sample by the binary classifiers that compose the OSOPF$^2_{\text{MCBIN}}$. In each of these figures, the dotted arc indicate the best path from a positive *prototype* and the dot-dash arc indicate the best path from a negative *prototype*.



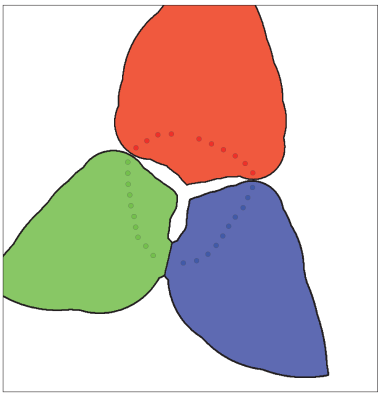
(a) OSOPF^2 for $T = 0.5$



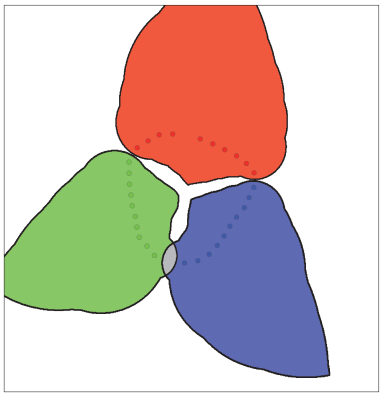
(b) OSOPF$^2_{\text{MCBIN}}$ for $T = 0.5$



(c) OSOPF^2 for $T = 0.8$



(d) OSOPF$^2_{\text{MCBIN}}$ for $T = 0.8$



(e) OSOPF$^2_{\text{MCBIN}}$ for $T = 0.8$

Figure 4.4: Decision boundaries for the synthetic dataset. The non-white regions represent the region in which a test sample would be classified as belonging to the same class of the samples with the same color. All samples in the white regions would be classified as *unknown*. The gray region in (e) is the region in which the test sample would be classified as positive by two binary classifiers.

then OSOPF^2 also classifies s as *unknown* and (2) if OSOPF$^2_{\text{MCBIN}}$ classifies s as ℓ_n, then OSOPF^2 also classifies s as ℓ_n, when the distance from the test sample to the nearest training sample is greater than or equal to the highest arc within the Optimum-Path Forest.

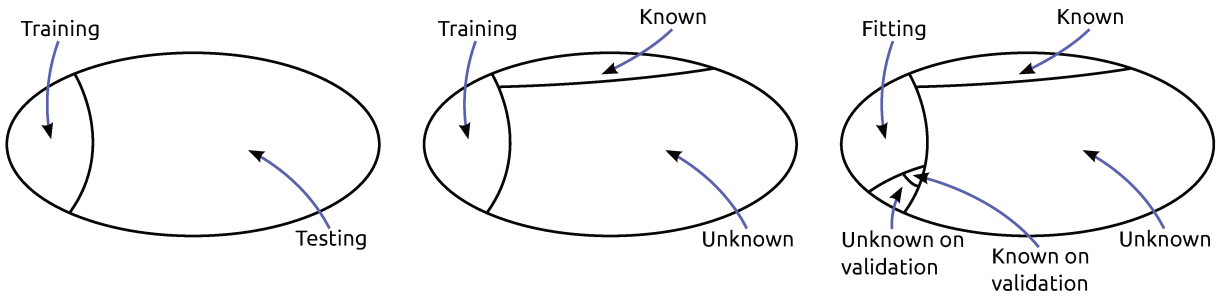
In OSOPF^2, let ℓ_n be the class of the *prototype* in the best path to s. Let OSOPF$^{\text{n}}_{\text{BIN}}$ be the OSOPF$^2_{\text{BIN}}$ of the OSOPF$^2_{\text{MCBIN}}$ trained with the class ℓ_n as a *positive* class. Analyzing for (1): as OSOPF$^2_{\text{MCBIN}}$ classifies s as *unknown*, every OSOPF$^2_{\text{BIN}}$ of the OSOPF$^2_{\text{MCBIN}}$ classifies s as *negative*, i.e., for each OSOPF$^2_{\text{BIN}}$, the ratio R between the cost of the *positive* path and the cost of the *negative* path is greater than T. Then, for the OSOPF$^{\text{n}}_{\text{BIN}}$, $R > T$ too. Consequently, the ratio R in OSOPF^2 is also greater than T because the cost of the

best path rooted in a class other than ℓ_n, according to Algorithm 3, is the same cost of the *negative* path in $\text{OSOPF}^{\text{n}}_{\text{BIN}}$, as the arc weight from s to the nearest training sample is the cost (see Equation 3.3). Then OSOPF^2 classifies s as *unknown*. Analyzing for (2): as $\text{OSOPF}^2_{\text{MCBIN}}$ classifies s as ℓ_n, the ratio R between the cost of the *positive* path and the cost of the *negative* path is less than or equal to T. Consequently, in OSOPF^2, R is also less than or equal to T because the cost of the best path in OSOPF^2 is the same cost of the *positive* path in $\text{OSOPF}^{\text{n}}_{\text{BIN}}$ and the cost of the best path rooted in a class other than ℓ_n is the same cost of the *negative* path in $\text{OSOPF}^{\text{n}}_{\text{BIN}}$. Then, OSOPF^2 classifies s as ℓ_n. □

4.3.3 Parameter optimization of OSOPF²

For the OSOPF^2, we perform a *parameter optimization* phase adapted for the open-set scenario to find the best value for T. In this work, the samples were divided into *training* and *testing* sets (see Figure 4.5a). In an open-set scenario, the *testing* set is the union of the *known* set and the *unknown* set, as there are classes for which samples are not available for *training* (see Figure 4.5b). In our *parameter optimization* phase, we divide the samples of the *training* set into *fitting* set (samples used to effectively train a classifier) and *validation* set (samples used to verify the accuracy based on a value of T), according to the following: (1) only half of the *available classes* have representative samples in the *fitting* set (the remaining is on the *validation* set) and (2) for each class considered in *fitting* set, half of its samples is on the *fitting* set (the remaining is on the *validation* set; see Figure 4.5c).

Finally, an OSOPF^2 classifier is fitted based on the samples of the *fitting* set and a traditional *grid search* [3] procedure is performed to find the best T value based on the samples in the *validation* set. Step (1) is performed to simulate the open-set scenario on the *parameter optimization* phase. Notice that the *parameter optimization* phase requires at least three available classes in the *training* set because at least one of those classes completely belongs to the *validation* set, as presented in Step (1), and the *fitting* set must contain samples of two different classes, as shown in Section 3.1.



(a) (b) (c)

Figure 4.5: Overview of data partitioning in the experiments and the *parameter optimization* of the OSOPF^2: (a) a dataset is divided into *training* and *testing* sets as a real scenario would be. (b) Most of the samples in *testing* set are *unknown* but need to be properly classified during testing operation of the classifier. (c) Partitioning of the *training* set by simulating an open-set scenario for *parameter optimization* of the OSOPF^2.

Chapter 5

Experiments

We compare the proposed OSOPF classifiers with the OPF, the multiclass SVM using One-vs-All approach ($\text{SVM}_{\text{MCBIN}}$), the multiclass SVMDBC [11, 12] using One-vs-All approach ($\text{SVMDBC}_{\text{MCBIN}}$), and the multiclass version of the 1-vs-Set Machine [43] also using the One-vs-All approach ($\text{SVM1VS}_{\text{MCBIN}}$) in different scenarios in terms of *openness* (see Section 5.2.3).

In this section, we present the evaluation measures, the experimental setup, including: details of the implementation of the classifiers $\text{SVM}_{\text{MCBIN}}$, $\text{SVMDBC}_{\text{MCBIN}}$, and $\text{SVM1VS}_{\text{MCBIN}}$, the datasets used for validation, and the experimental protocol. Then, we present the experiments and the obtained results. The details of the $\text{SVM}_{\text{MCBIN}}$ method are presented because its implementation differ from the implementation in the LibSVM [7]. Similarly, we present details of $\text{SVMDBC}_{\text{MCBIN}}$ and $\text{SVM1VS}_{\text{MCBIN}}$ because we extended the original methods to multiclass open-set scenarios for comparison purposes.

5.1 Evaluation measures

For evaluating classifiers in an open-set scenario, we should be aware of the *unknown* classes. Most of the existing evaluation measures, like the *macro-* and *micro-averaging f-measure* [45], the *average accuracy* [45], and the traditional *classification accuracy* [7], do not take into account the *unknown*. Therefore, another contribution of this work is the adaptation of two measures to assess the quality of open-set classifiers.

In the literature, the following classes of evaluation measures are found: (1) Measures for closed-set binary problems (traditional accuracy, *f-measure*, etc.); (2) Measures for closed-set multiclass problems (traditional accuracy, multiclass version of the *f-measure*, etc); (3) Measures for open-set binary problems, presented in the work of Costa et al. [11, 12] (the open-set version of the *average accuracy*). Measures in (1) and

(2) are not appropriate as they do not consider open-set scenarios, which usually lead to the overestimation of the performance of evaluated classifiers. The measures adopted in (3), in turn, do not consider open-set multiclass classification problems. The measures proposed in this work define a new class of evaluation measures, as they are suitable for open-set multiclass classification problems.

Here, we call *known samples* as the samples belonging to one of the *available classes* for training. The *unknown samples* belong to classes for which no representative sample is used during training.

5.1.1 Normalized accuracy

For a better picture regarding the effectiveness of the classifier in open-set scenarios, we compute the results as a *normalized accuracy* that takes into account both the *accuracy of known samples* (AKS) and the *accuracy of unknown samples* (AUS). The *normalized accuracy* was considered because it avoids overestimating the performance of biased classifiers, i.e., classifiers that occasionally classify almost all samples as belonging to the most frequent class. This is important because the more open the scenario, the greater the amount of *unknown* samples.

5.1.2 F-measure

Besides using the *normalized accuracy* to assess the quality of results of classifiers in open-set scenarios, we also use the *macro-* and *micro-averaging f-measure* because these measures can give us fine-grained analysis of the behavior of the evaluated methods. The definition and the properties of *f-measure* are presented by Sokolova and Lapalme [45]. The following equation describes the traditional *f-measure*:

$$f\text{-}measure = \frac{2 \times precision \times recall}{precision + recall} \tag{5.1}$$

In our work, we adapt the traditional definition of *f-measure* to assess the quality results of an open-set testing protocol.

A trivial extension of *f-measure* to open-set scenario could be to consider the *unknown* as one simple class and obtain the *f-measure* value in the same way it is accomplished for the closed-set scenario. But this trivial extension of the *f-measure* is not appropriate to evaluate tests in open-set scenarios because all correct classification of *unknown* test samples are going to be considered *true positive* classifications. These classification results cannot be considered *true positive* because it does not make sense to consider the *unknown* classes as one single positive class, since we have no representative samples of *unknown* classes to train the classifier.

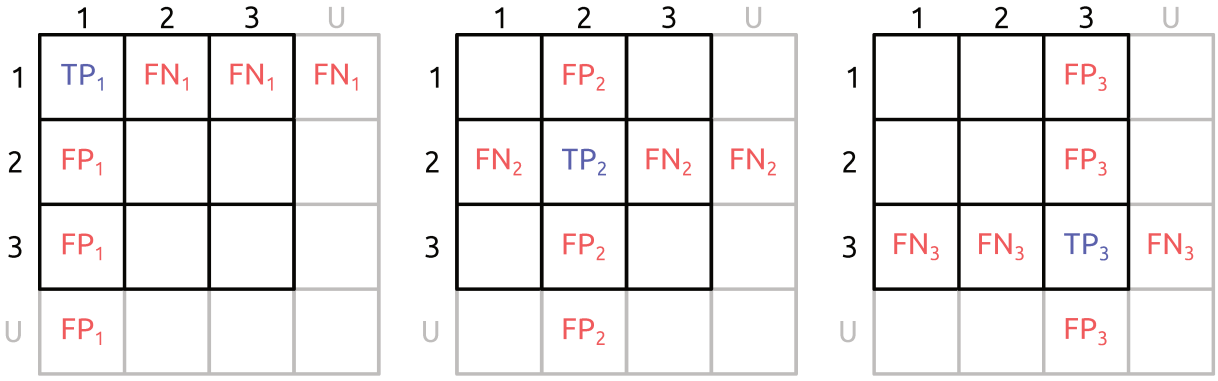


Figure 5.1: Open-set confusion matrix. Example related to the computation of the *precision* and *recall* measures from an open-set confusion matrix with three *available classes* and *unknown* samples (U). The FN_i in the column U and FP_i in the row U account for *false unknown* and *false known*, respectively. The cell in the intersection of column U and row U is *not* regarded as *true positive* in the same way it would be considered by the multiclass closed-set *f-measure*.

Our open-set modifications to the *f-measure* are related to how precision and recall are computed. Equations 5.2 and 5.3 are used to compute the *macro-averaging f-measure* and *micro-averaging f-measure*, respectively. The measures $precision_M$ and $recall_M$ stand for the macro precision and the macro recall, respectively. The measures $precision_\mu$ and $recall_\mu$, in turn, stand for the micro precision and the micro recall, respectively. The following equations detail the proposed modified measures, which are the basis for the *f-measure* computed by Equation 5.1:

$$precision_M = \frac{\sum_{i=1}^{l-1} \frac{TP_i}{TP_i+FP_i}}{l-1}, \qquad recall_M = \frac{\sum_{i=1}^{l-1} \frac{TP_i}{TP_i+FN_i}}{l-1} \tag{5.2}$$

$$precision_\mu = \frac{\sum_{i=1}^{l-1} TP_i}{\sum_{i=1}^{l-1}(TP_i+FP_i)}, \qquad recall_\mu = \frac{\sum_{i=1}^{l-1} TP_i}{\sum_{i=1}^{l-1}(TP_i+FN_i)} \tag{5.3}$$

where $l = n+1$ is the size of the confusion matrix and n is the number of *available classes* for training. TP, FP, and FN stand for *true positive*, *false positive*, and *false negative* samples, respectively. For these formulas, the last column and the last row refer to the *unknown*.

Note in Equations 5.2 and 5.3 that in spite of computing the *precision* and *recall* only for the n available classes, the FN and FP take into account the *false unknown* and *false known*, respectively, as illustrated in Figure 5.1.

The *f-measure* adopted in traditional multiclass classification is invariant to the *true negative* [45], i.e., the *f-measure* does not take into account the samples truly rejected as

belonging to the class under consideration. Similarly, the proposed multiclass *f-measure* for open-set scenarios is invariant to the *true unknown*. But both the *false unknown* (*known samples* incorrectly classified as *unknown*) and the *false known* (*unknown samples* incorrectly classified as *known*) are considered in the *f-measure*. We adopted this strategy for two reasons: (1) *f-measure* must give importance to classified *known samples*; and (2) the *unknown* is not a single class but possibly several ones.

5.2 Experimental setup

In this section, we present the baselines (Section 5.2.1) and the datasets (Section 5.2.2) considered in the experiments. The experimental protocol adopted is presented in Section 5.2.3.

5.2.1 Baselines

According to Chang and Lin [7], the current implementation of the LibSVM [7] for multiclass classification uses the One-vs-One approach, i.e., the multiclass problem for n classes is divided into $\frac{n\times(n-1)}{2}$ One-vs-One binary problems. The class of a test sample s is determined according to the classification of s by all binary classifiers. A voting scheme is employed, according to which s is assigned to the most voted class. When two or more classes receive the highest number of votes the tie-break policy consists in choosing the smallest label. We can see that using One-vs-One approach, the $\text{SVM}_{\text{MCBIN}}$ would be a closed-set classifier, i.e., at least one class will be the most voted and a test sample will always be classified as one of the known (training) classes.

In our work, we use the One-vs-All approach for the $\text{SVM}_{\text{MCBIN}}$. In this approach, the multiclass problem with n classes is divided into n binary problems. As shown in Figure 1.1, using the One-vs-All approach it is possible to classify a test sample s as *unknown*: when the n binary classifiers classify a test sample as *negative*. This approach does not use the voting scheme because each class appears only once as positive, while the negative side of the binary classifiers contains $n-1$ classes. In cases in which two or more binary classifiers classify s as positive, the chosen class is based on the distance of the tested sample to the decision hyperplane. The binary classifier, according to which s is more distant to the hyperplane, is used to define the class of s. We used the Radial Basis Function (RBF) kernel in the $\text{SVM}_{\text{MCBIN}}$.

According to Chang and Lin [7], there are two possible ways to accomplish the *grid search* for a binary-based multiclass classifier like $\text{SVM}_{\text{MCBIN}}$: (1) the *external* and (2) the *internal grid search*[1]. In the *external* approach, the *grid search* is performed in the

[1]We defined these names.

multiclass level forcing all the binary classifiers to share the same parameters. On the other hand, in the *internal grid search*, each binary classifier performs its own *grid search*. According to the analysis of Chen et al. [8] considering the One-vs-One approach, the *external* approach obtains parameters not uniformly good to every binary classifier. Still, it considers the overall accuracy of the multiclass classifier. Also, the *internal grid search* can over-fit the classifier.

We used both *external* and *internal* approaches in comparison to the proposed method. We could verify that both $\text{SVM}^{\text{ext}}_{\text{MCBIN}}$ (using *external grid search*) and $\text{SVM}_{\text{MCBIN}}$ (using *internal grid search*) have no statistical difference between them when using the One-vs-All approach. These results are in accordance with the analysis of Chen et al. [8] and Chung et al. [10] when using One-vs-One approach as they stated that there are no relevant difference between the two approaches for *grid search*. In open-set scenario, one advantage of using the *internal grid search* would be not to deal with a "simulation" of the open-set scenario[2] in the *external grid search*. In fact, even keeping the traditional closed-set grid search the *internal* and *external* approaches are equivalent. All other baselines use the *internal grid search*.

Costa et al. [11, 12] presented a binary classifier named SVM with Decision Boundary Carving (SVMDBC). In their work, it was not explained how to extend the SVMDBC for multiclass classification. Although the source camera attribution problem addressed by Costa et al. [11, 12] requires a specific decision among the *available classes*, the results were obtained based only on the binary classifications. For example, supposing a scenario with $n = 25$ classes, but with only four *available classes* for training, the authors in [11, 12] computed the individual results of the four binary classifiers and merged them (e.g., using the mean). However, a more suitable way to acquire the results in this case is to verify the final multiclass classification, i.e., the test sample must be classified as one of the four classes or *unknown*.

In our work, we trivially extend the SVMDBC for multiclass classification in the same way it was performed for the SVM: we use the One-vs-All approach with decision based on the distance from the hyperplane in cases in which two or more binary classifiers classify a test sample as *positive*. To be clear in the text, we call that trivial multiclass-from-binary extension, created for a fair comparison purpose, as $\text{SVMDBC}_{\text{MCBIN}}$. Our implementation of the $\text{SVMDBC}_{\text{MCBIN}}$ uses the RBF kernel.

The 1-vs-Set Machine proposed by Scheirer et al. [43] is also a binary classifier. Our multiclass-from-binary trivial extension, referenced by $\text{SVM1VS}_{\text{MCBIN}}$, follows the same structure of $\text{SVM}_{\text{MCBIN}}$ and $\text{SVMDBC}_{\text{MCBIN}}$: it uses the One-vs-All approach and the *internal grid search*. The decision between two binary classifiers that classify *positive* is

[2]The "simulation" of the open-set scenario can be understood as a reproduction of the real testing scenario using only the training samples.

Table 5.1: General characteristics of the classifiers used in the experiments.

Method	Approach	Open-set	Kernel	Grid search
SVM_{MCBIN}	One-vs-All	Yes	RBF	Internal
SVM^{ext}_{MCBIN}	One-vs-All	Yes	RBF	External
$SVMDBC_{MCBIN}$	One-vs-All	Yes	RBF	Internal
$SVM1VS_{MCBIN}$	One-vs-All	Yes	Linear	Internal
OPF	Multiclass	No	–	–
$OSOPF^1$	Multiclass	Yes	–	–
$OSOPF^2$	Multiclass	Yes	–	External

accomplished based on the distance from the main hyperplane of the classifier. We used the linear kernel instead of the RBF, as the authors presented best results with a linear kernel in their work [43]. We used the implementation of the binary 1-vs-Set Machine released by Scheirer et al. [43].

In Table 5.1, we summarize the evaluated methods. The method is classified as open set when it allows somehow the classification of a test sample as *unknown*.

5.2.2 Datasets

In this work, we performed experiments considering six datasets: *15-Scene* [25], *letter* [17, 29], *Auslan* [22], *Caltech-256* [19], *ALOI* [18], and *ukbench* [31]. Those datasets represent applications of object recognition (*Caltech-256*, *ALOI*, *ukbench*), scene recognition (*15-Scene*), sign language recognition (*Auslan*), and letter image recognition (*letter*). Ahead, we present the description for each dataset.

- In the *15-Scene* dataset, with 15 classes, the 4485 images were represented by a bag-of-visual-word vector created with soft assignment [50] and max pooling [5], based on a codebook of 1000 SIFT codewords.

- The 26 classes of the *letter* dataset represent the letters of the English alphabet (black-and-white rectangular pixel displays). The 20000 samples contain 16 attributes.

- The *Auslan* dataset contains 95 classes of Australian Sign Language (*Auslan*) signs collected from a volunteer native Auslan signer [22]. Data was acquired using two Fifth Dimension Technologies (5DT) gloves hardware and two Ascension Flock-of-Birds magnetic position trackers. There are 146949 samples represented with 22 features (x, y, z positions, bend measures, etc).

Table 5.2: General characteristics of the datasets used in the experiments.

Dataset	# samples	# classes	# features	# samples/class
15-Scene	4485	15	1000	299
letter	20000	26	16	769
Auslan	146949	95	22	1546
Caltech-256	29780	256	1000	116
ALOI	108000	1000	128	108
ukbench	10200	2550	128	4

- The *Caltech-256* dataset comprises 256 object classes. The feature vectors consider a bag-of-visual-words characterization approach and contain 1000 features, acquired with dense sampling, SIFT descriptor for the points of interest, hard assignment [50], and average pooling [5]. In total, there are 29780 samples.
- The *ALOI* dataset has 1000 classes and 108 samples for each class (108000 in total). The features were extracted with the BIC descriptor [48] and contain 128 dimensions.
- The *ukbench* dataset of recognition benchmark images consists of 2550 classes of four images. In our work, the images were represented with BIC descriptor [48] (128 dimensions).

In Table 5.2, we present the overall characteristics of the datasets we used in the experiments. Note that we did not try to find the best characterization approach for each dataset since this is not the focus of this work. We relied on characteristics that presented good results according to prior work in the literature. In addition, all of the used features are freely available[3].

5.2.3 Experimental protocol

We performed experiments on all these datasets by training each classifier with $n = 3$, 6, 9, and 12 classes available for training among the total number of classes of each dataset. Each experiment consists of a combination of a classifier, a dataset, and a set of n *available classes*. For each experiment, we

1. randomly choose n *available classes* for training;
2. consider half of the *known samples* in each of the n classes for testing;

[3] http://dx.doi.org/10.6084/m9.figshare.1097614 (As of Oct. 2014)

Table 5.3: *Openness* of all experiments according to Equation 5.4. The column *ac* indicates the number of *available classes* for training. For each dataset we also present the total number of classes.

ac	*15-Scene* 15	*letter* 26	*Auslan* 95	*Caltech-256* 256	*ALOI* 1000	*ukbench* 2550
3	0.55	0.66	0.82	0.89	0.94	0.96
6	0.36	0.51	0.74	0.84	0.92	0.95
9	0.22	0.41	0.69	0.81	0.90	0.94
12	0.10	0.32	0.64	0.78	0.89	0.93

3. consider the samples of the other classes as *unknown* for testing; and

4. acquire results based on the previously mentioned measures (see Section 5.1).

Scheirer et al. defined the *openness* of a problem[4] measured based on Equation 5.4 [43]:

$$openness = 1 - \sqrt{\frac{|\text{training classes}|}{|\text{testing classes}|}}, \tag{5.4}$$

In Table 5.3, we present the *openness* of the classification scenarios considered in the experiments. Note that the more classes available for training, the less open the classification problem. The considered classification scenarios are "very open", except for the *15-Scene* and the *letter* datasets.

We performed the Analysis of Variance (ANOVA) statistical test and used the post-test Tukey "Honest Significant Differences" (HSD) method [30, 54] to confirm the statistical differences (95% of confidence) on the results. For each experiment (a combination of classifier, dataset, and number of *available classes*), we ran 10 times with different sets of *available classes*.

5.3 Results

In this section, we present the performed experiments to validate the proposed methods. First, in Section 5.3.1, we show a comparison of the classification performance of seven methods in six different datasets. In Section 5.3.2, we analyze the *decision boundaries* of the proposed open-set classifier in synthetic datasets. Finally, in Section 5.3.3, we examine the impact of parameter T in the OSOPF^2 classifier.

[4]The *openness* measure serves only for evaluating open-set solutions in academic terms since in practice it might not be possible to even estimate the actual number of classes.

5.3.1 Classification performance

Results are presented in Figures 5.2, 5.3, and 5.4 for the *letter*, *Auslan*, and *ALOI* datasets, respectively, considering the classifiers trained with 3, 6, 9, and 12 classes and comparing $\text{SVM}_{\text{MCBIN}}$, $\text{SVM}^{\text{ext}}_{\text{MCBIN}}$, $\text{SVMDBC}_{\text{MCBIN}}$, $\text{SVM1VS}_{\text{MCBIN}}$, OPF, OSOPF^1, and OSOPF^2 classifiers. In Figures 5.2a, 5.3a, and 5.4a, we present the AKS, i.e., the accuracy obtained based on the samples whose class is known. In Figures 5.2b, 5.3b, and 5.4b, we present the results related to the other measure that composes our proposed *normalized accuracy*: the AUS. We also present the results of the two proposed multiclass measures for the open-set scenario: the *normalized accuracy* (see Figures 5.2c, 5.3c, and 5.4c) and the *micro-averaging f-measure* (see Figures 5.2d, 5.3d, and 5.4d).

In spite of the OSOPF^2 having the lowest AKS, as seen in Figures 5.2a, 5.3a, and 5.4a, we can see in Figures 5.2b, 5.3b, and 5.4b that based on the AUS, it outperforms other classifiers. The AUS points out how well the *unknown* samples are identified at testing phase.

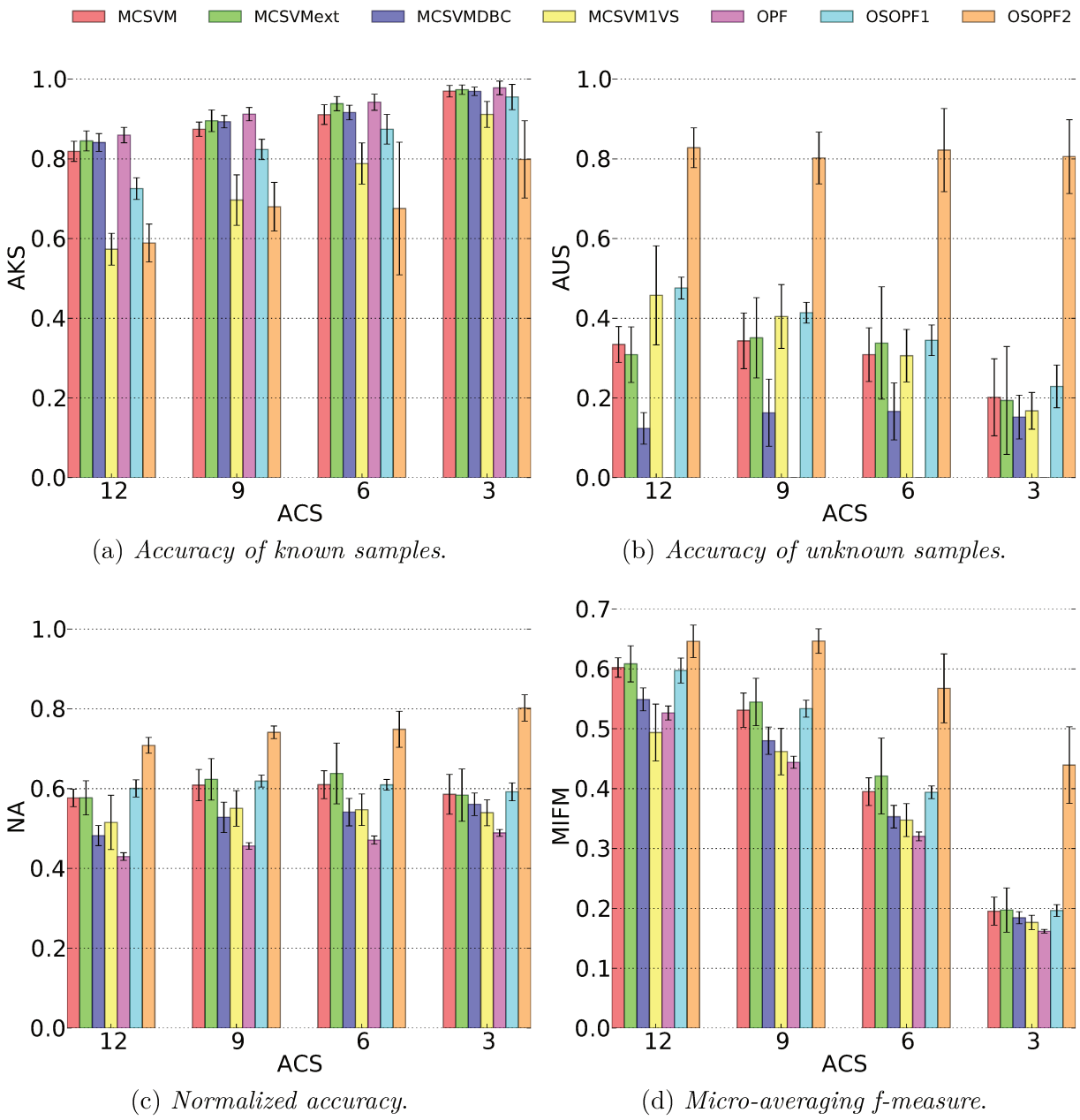
The cause of the lower AKS of the OSOPF^2 presented in Figures 5.2a, 5.3a, and 5.4a is that all doubtful test samples are classified as *unknown*. In this vein, all samples in an *overlapping region* are going to be misclassified as *unknown* instead of one of the overlapping classes. *Overlapping regions* refer to the regions with some density of training samples of two or more different training classes.

In AUS, NA, and MIFM measures in Figures 5.2, 5.3, and 5.4, we can see that the open-set classifiers of the literature (the $\text{SVMDBC}_{\text{MCBIN}}$ and $\text{SVM1VS}_{\text{MCBIN}}$) did not obtain a good performance. Even the multiclass-from-binary version of the traditional SVM (the $\text{SVM}_{\text{MCBIN}}$) obtained better results.

We can see in Figures 5.2b, 5.3b, and 5.4b that OPF has no AUS. This is because of the first property of the OPF presented in Section 3.3: the standard OPF never classifies a test sample as *unknown*. On the other hand, the OPF obtained the best results based on the AKS.

In Figures 5.2c, 5.3c, and 5.4c we can see that the OSOPF^2 and the other classifiers are somehow stable as the scenario opens. Still, the *micro-averaging f-measure* illustrated in Figures 5.2d, 5.3d, and 5.4d indicates that all classifiers are impacted as the scenario gets more open.

Similar graphs were obtained for the *ukbench* dataset: the AKS of the OSOPF^2 is a little bit worse compared to the AKS of the other classifiers; based on the AUS, *normalized accuracy*, and *micro-averaging f-measure*, the OSOPF^2 is always better than the other classifiers. For the *Caltech-256* and *15-Scene* datasets, the AKS and AUS are also worse and better, respectively, compared to the other classifier, but the OSOPF^2 did not obtain the best *micro-averaging f-measure*. In *Caltech-256*, the OSOPF^2 obtained the best *normalized accuracy* for 3, 6, 9, and 12 *available classes*, but its *normalized accuracy*



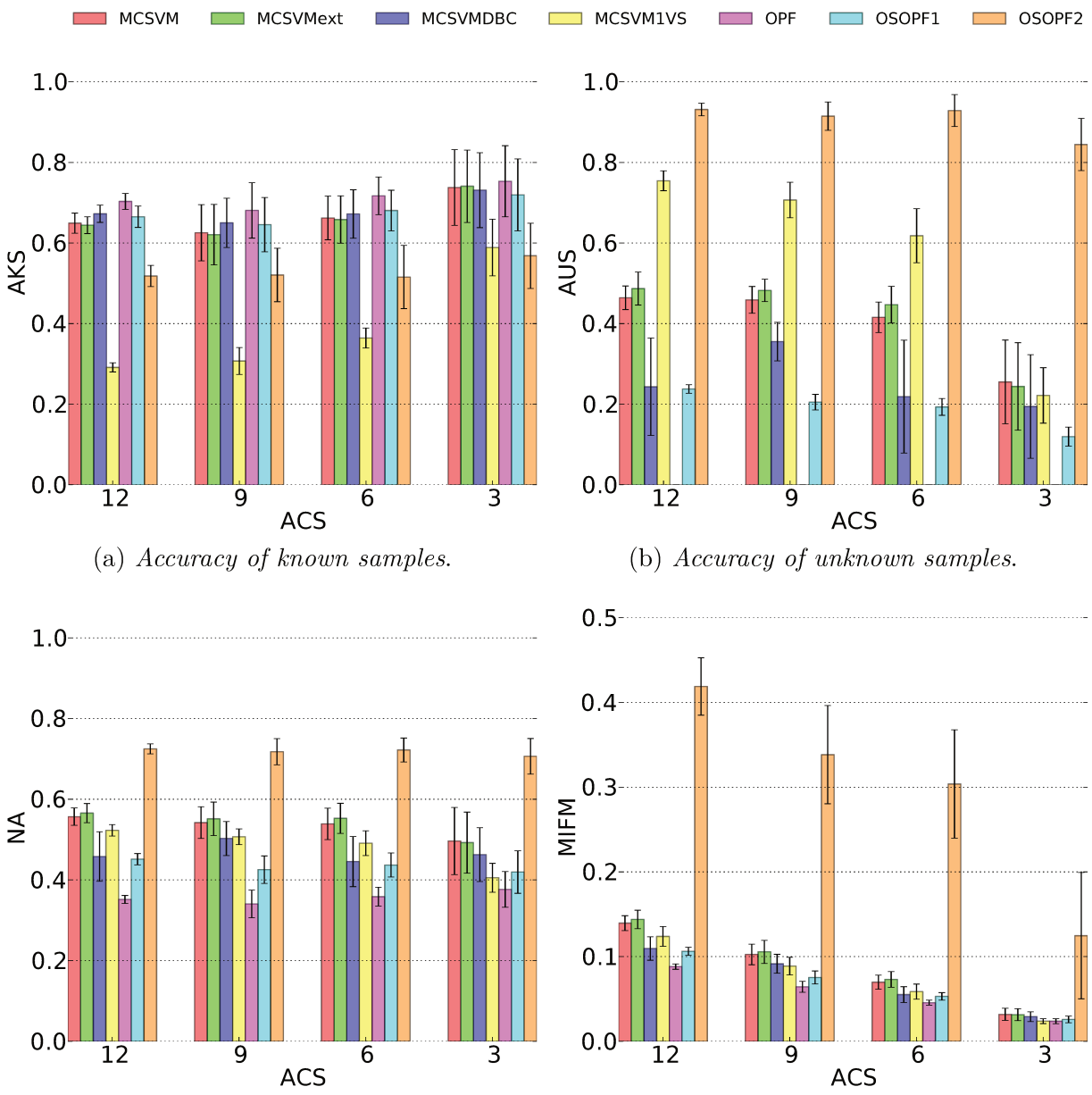
(a) *Accuracy of known samples.*

(b) *Accuracy of unknown samples.*

(c) *Normalized accuracy.*

(d) *Micro-averaging f-measure.*

Figure 5.2: Results for 3, 6, 9, and 12 *available classes* (ACS) for the *letter* dataset. Figures (a), (b), (c), and (d) depict the accuracy measures for all classifiers. Figure (a) depicts the accuracy obtained on test samples belonging to one of the *available classes* for training (AKS). Figure (b) depicts the accuracy in the samples whose class is *unknown* at training phase (AUS). Figures (c) and (d) depict the results of the proposed *normalized accuracy* (NA) and *micro-averaging f-measure* (MIFM) measures, respectively.



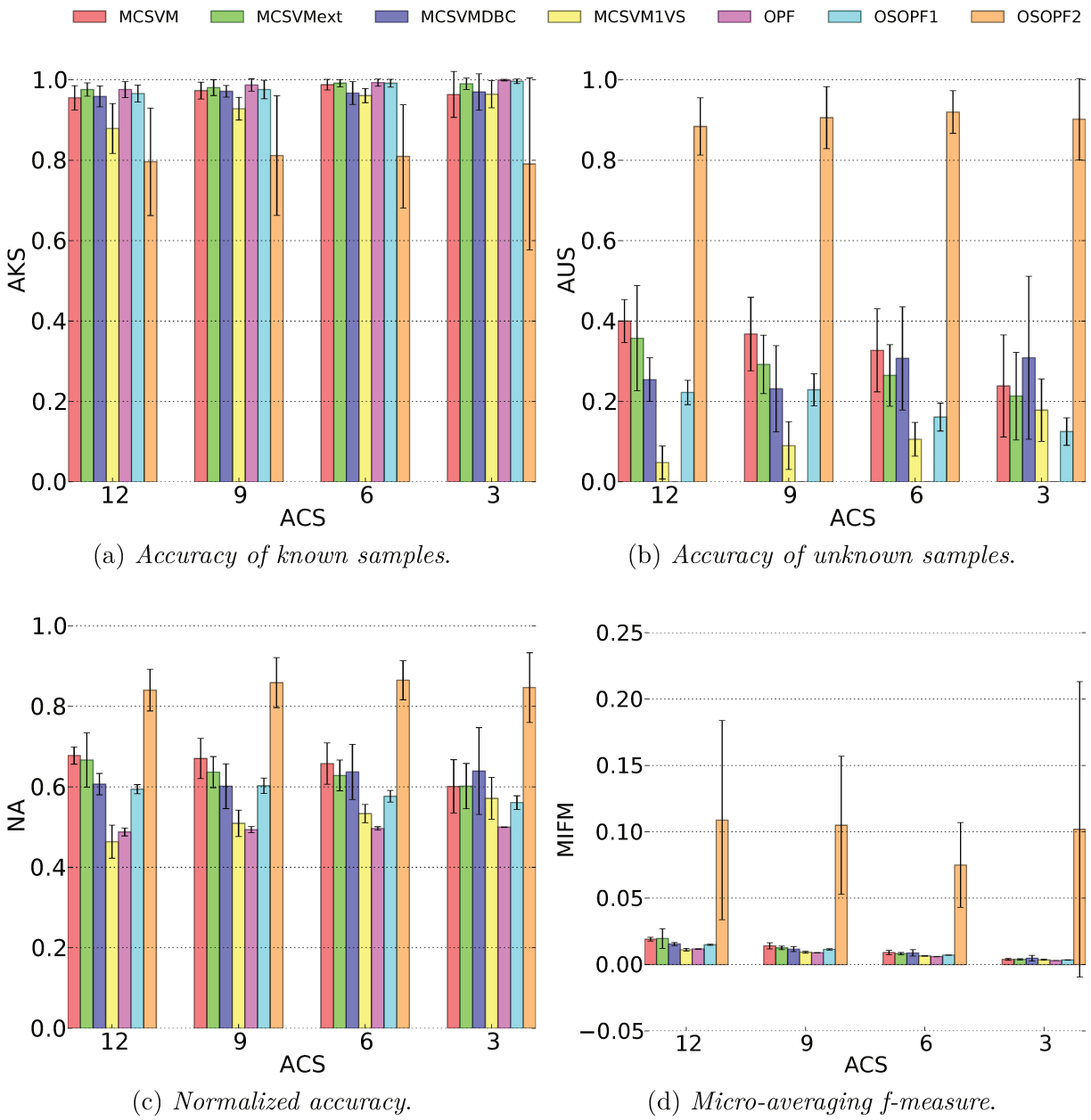
(a) *Accuracy of known samples.*

(b) *Accuracy of unknown samples.*

(c) *Normalized accuracy.*

(d) *Micro-averaging f-measure.*

Figure 5.3: Results for 3, 6, 9, and 12 *available classes* (ACS) for the *Auslan* dataset. Figures (a), (b), (c), and (d) depict the accuracy measures for all classifiers. Figure (a) depicts the accuracy obtained on test samples belonging to one of the *available classes* for training (AKS). Figure (b) depicts the accuracy in the samples whose class is *unknown* at training phase (AUS). Figures (c) and (d) depict the results of the proposed *normalized accuracy* (NA) and *micro-averaging f-measure* (MIFM) measures, respectively.



(a) *Accuracy of known samples.*

(b) *Accuracy of unknown samples.*

(c) *Normalized accuracy.*

(d) *Micro-averaging f-measure.*

Figure 5.4: Results for 3, 6, 9, and 12 *available classes* (ACS) for the *ALOI* dataset. Figures (a), (b), (c), and (d) depict the accuracy measures for all classifiers. Figure (a) depicts the accuracy obtained on test samples belonging to one of the *available classes* for training (AKS). Figure (b) depicts the accuracy in the samples whose class is *unknown* at training phase (AUS). Figures (c) and (d) depict the results of the proposed *normalized accuracy* (NA) and *micro-averaging f-measure* (MIFM) measures, respectively.

on the *15-Scene* was not the better one.

We believe that worst OSOPF^2's performance on the *Caltech-256* and *15-Scene* is because these datasets are not very well separable, i.e., there are a considerable number of *overlapping regions*.

In Tables 5.4 and 5.5, for each pair of methods (the intersection between the row and the column) the arrow indicates the winner method as verified with the ANOVA and Tukey post-hoc tests. An empty cell indicates that the difference between the pair of methods is not statistically significant. The column *ac* refers to the number of *available classes*. We present the results for 3, 6, 9, and 12 *available classes*.

In Table 5.4 we present the results of Tukey HSD post-test for *all* datasets together to compare the overall behavior of the methods. Note that we performed the statistical test of all datasets as the post-test Tukey HSD method allows this kind of testing.

In Table 5.5, for each pair of methods, we present separated results for six datasets: *15-Scene* in the first cell, *letter* in the second cell, *Auslan* in the third cell, *Caltech-256* in the fourth cell, *ALOI* in the fifth cell, and *ukbench* in the last cell. We can see that the proposed OSOPF^2 is superior to the other classifiers in most of the cases.

Table 5.4: Statistical tests for all datasets. The arrows point to the winner methods for each pair of compared methods (row and column). Empty cells indicate there is no statistical difference between the pair of methods. The column *ac* indicates the number of *available classes* during training.

	ac	SVM_{MCBIN}	SVM^{ext}_{MCBIN}	$SVMDBC_{MCBIN}$	$SVM1VS_{MCBIN}$	OPF	$OSOPF^1$	$OSOPF^2$
SVM_{MCBIN}	3				←	←		↑
	6			←	←	←	←	↑
	9			←	←	←	←	↑
	12			←	←	←		↑
SVM^{ext}_{MCBIN}	3				←	←		↑
	6			←	←	←		↑
	9			←	←	←	←	↑
	12			←	←	←		↑
$SVMDBC_{MCBIN}$	3					←		↑
	6	↑	↑			←	↑	↑
	9	↑	↑		←	←	↑	↑
	12	↑	↑			←	↑	↑
$SVM1VS_{MCBIN}$	3	↑	↑			←		↑
	6	↑	↑			←	↑	↑
	9	↑	↑	↑		←	↑	↑
	12	↑	↑			←	↑	↑
OPF	3	↑	↑	↑	↑		↑	↑
	6	↑	↑	↑	↑		↑	↑
	9	↑	↑	↑	↑		↑	↑
	12	↑	↑	↑	↑		↑	↑
$OSOPF^1$	3					←		↑
	6	↑		←	←	←		↑
	9	↑	↑	←	←	←		↑
	12			←	←	←		↑
$OSOPF^2$	3	←	←	←	←	←	←	
	6	←	←	←	←	←	←	
	9	←	←	←	←	←	←	
	12	←	←	←	←	←	←	

Table 5.5: Statistical tests for every dataset. d_1, d_2, d_3, d_4, d_5, and d_6 refer to the following datasets, respectively: *15-Scene*, *letter*, *Auslan*, *Caltech-256*, *ALOI*, and *ukbench*. The arrows point to the winner methods for each pair of compared methods (row and column). Empty cells indicate there is no statistical difference between the pair of methods. The column *ac* indicates the number of *available classes* for training.

		SVM_{MCBIN}						SVM^{ext}_{MCBIN}						$SVMDBC_{MCBIN}$						$SVM1VS_{MCBIN}$						OPF						$OSOPF^1$						$OSOPF^2$					
	ac	d_1	d_2	d_3	d_4	d_5	d_6	d_1	d_2	d_3	d_4	d_5	d_6	d_1	d_2	d_3	d_4	d_5	d_6	d_1	d_2	d_3	d_4	d_5	d_6	d_1	d_2	d_3	d_4	d_5	d_6	d_1	d_2	d_3	d_4	d_5	d_6	d_1	d_2	d_3	d_4	d_5	d_6
SVM_{MCBIN}	3																			←		←				←	←	←	←	←				←					↑	↑		↑	
	6													←	←	←	←			←	←	←	←	←		←	←	←	←	←	←	←		←		←			↑	↑		↑	
	9													←	←	←	←	←	←	←	←	←	←	←	←	←	←	←	←	←	←			←		←			↑	↑		↑	↑
	12													←	←	←	←	←	←	←	←		←	←	←	←	←	←	←	←	←			←		←			↑	↑		↑	
SVM^{ext}_{MCBIN}	3																			←		←				←	←	←	←	←				←	←				↑	↑		↑	
	6													←	←	←	←			←	←	←	←	←		←	←	←	←	←	←	←		←		←			↑	↑		↑	↑
	9													←	←	←	←		←	←	←	←	←	←	←	←	←	←	←	←	←			←	←				↑	↑		↑	
	12													←	←	←	←	←	←	←	←	←	←	←	←	←	←	←	←	←	←			←	←	←			↑	↑		↑	
$SVMDBC_{MCBIN}$	3																			←						←	←	←	←	←									↑	↑		↑	
	6	↑	↑	↑	↑			↑	↑	↑	↑									←				←		←	←	←	←	←		↑	↑		↑	←	↑	↑	↑	↑	↑	↑
	9	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑		↑							←			←	←		←	←	←	←	←		↑	↑	←	↑		↑	↑	↑	↑	↑	↑
	12	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑							←		↑	←			←	←	←	←	←		↑	↑		↑		↑	↑	↑	↑	↑	↑
$SVM1VS_{MCBIN}$	3	↑		↑				↑		↑				↑															←			↑	↑	↑				↑	↑	↑	↑	↑	↑
	6	↑	↑	↑	↑	↑		↑	↑	↑	↑	↑		↑				↑									←	←	←			↑	↑	←	↑			↑	↑	↑	↑	↑
	9	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑			↑	↑									←	←				↑	↑	←	↑	↑	↑	↑	↑	↑	↑	↑
	12	↑	↑		↑	↑	↑	↑	↑	↑	↑	↑	↑	↑		←	↑	↑								↑	←	←				↑	↑	←	↑	↑	↑	↑	↑	↑	↑	↑
OPF	3	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑				↑									↑	↑		↑		↑	↑	↑	↑	↑	↑
	6	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑		↑	↑	↑									↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑
	9	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑		↑	↑	↑									↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑
	12	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	←	↑	↑	↑									↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑	↑
$OSOPF^1$	3			↑						↑	↑									←				←		←	←												↑	↑		↑	
	6	↑		↑		↑		↑		↑		↑		←	←		←	↑	←	←	←	↑	←			←	←	←	←	←	←								↑	↑	↑	↑	
	9			↑		↑				↑	↑			←	←	↑	←		←	←	←	↑	←	←	←	←	←	←	←	←	←								↑	↑	↑	↑	↑
	12			↑		↑				↑	↑	↑	↑	←	←		←		←	←	←	←	←	←	←	←	←	←	←	←	←								↑	↑	↑		
$OSOPF^2$	3		←	←		←			←	←		←		←	←		←		←	←	←	←	←	←	←	←	←	←	←	←	←		←	←		←						
	6		←	←		←			←	←		←	←	←	←	←	←	←	←	←	←	←	←	←	←	←	←	←	←	←	←		←	←	←	←						
	9		←	←		←	←		←	←		←		←	←	←	←	←	←	←	←	←	←	←	←	←	←	←	←	←	←		←	←		←	←					
	12		←	←		←			←	←		←		←	←	←	←	←	←	←	←	←	←	←	←	←	←	←	←	←	←		←	←	←	←						

Based on Tables 5.4 and 5.5, we can note the effectiveness of the proposed methods. Our more prominent open-set version of the OPF, the OSOPF^2, obtained results better than or equal to SVM$_{\text{MCBIN}}$, SVM$^{\text{ext}}_{\text{MCBIN}}$, SVMDBC$_{\text{MCBIN}}$, SVM1VS$_{\text{MCBIN}}$, OPF, and OSOPF^1 in all experiments. OSOPF^2 yields strictly better results than other classifiers in *letter*, *Auslan*, and *ALOI* datasets. Also, note that impressive results obtained with *ukbench* which shows that the proposed methods can deal with open-set problems even when presented with just a few training samples per available class (see Table 5.2).

5.3.2 Analysis of decision boundaries

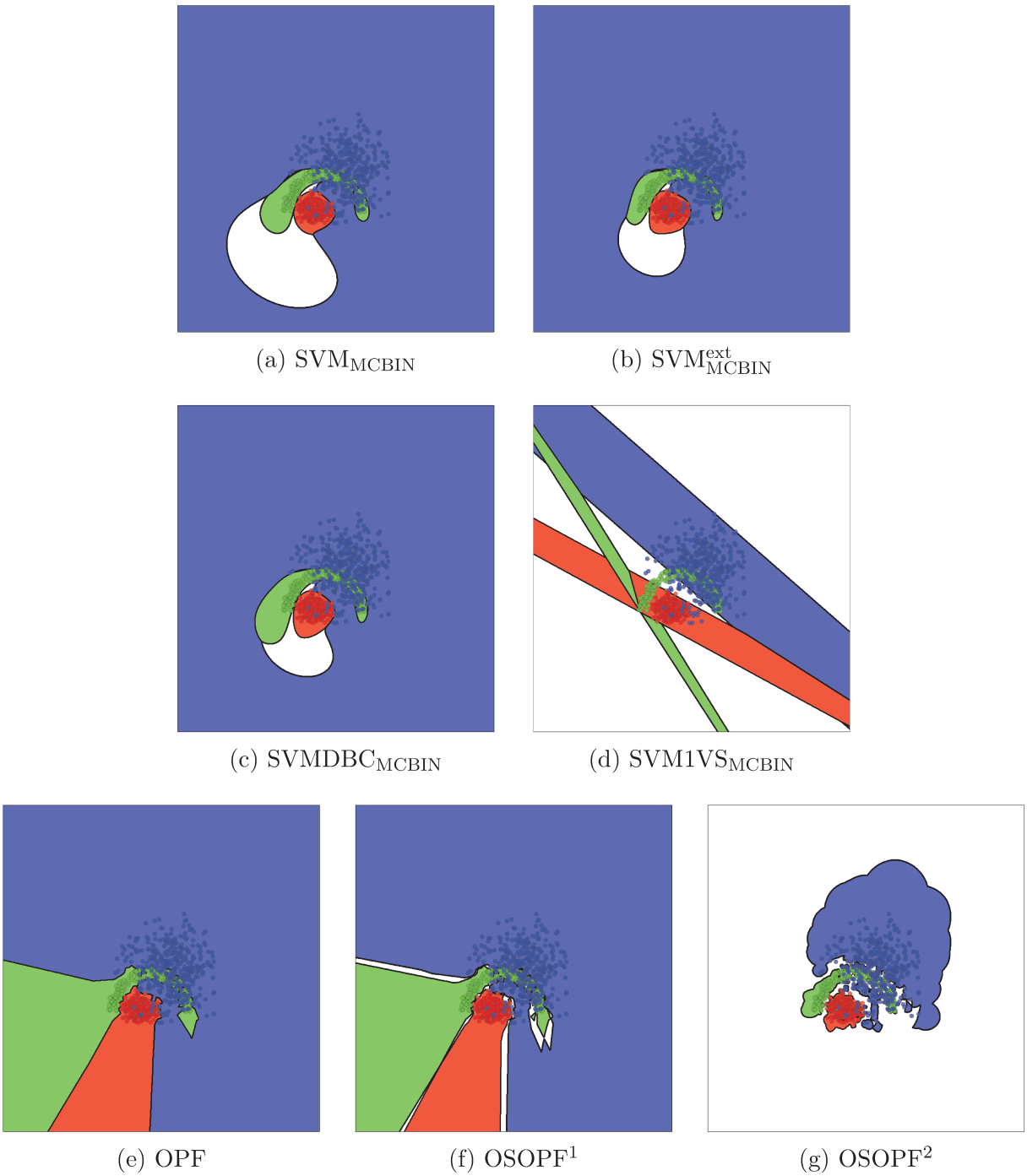
Aiming at visually understanding the different behavior of the classifiers, we also performed tests on 2-dimensional synthetic datasets. We used the *Cone-torus*, *Four-gauss* [23], and *R15* [51] datasets. We trained the classifiers using all samples of the dataset to plot the *decision boundaries* for each class. The *decision boundary* of a class defines the region in which an possible test sample will be classified as belonging to that class.

In Figures 5.5, 5.6, and 5.7, we present the *decision boundaries* for the *Cone-torus*, *Four-gauss*, and *R15* datasets, respectively. The non-white regions represent the region in which a test sample would be classified as belonging to the same class of the samples with the same color. All samples in the white regions would be classified as *unknown*.

Based on these figures, we can note that OSOPF^2 successfully classifies test samples as *unknown* when necessary. While the SVM$_{\text{MCBIN}}$ are able to classify as *unknown* only the doubtful samples among the *available classes*, OSOPF^2 also avoids recognizing the faraway samples. In this work, our proposed OSOPF^2 is the only classifier able to create a bounded *open space of risk*.

5.3.3 The impact of threshold T

In the previous experiments, we performed the *parameter optimization* phase presented in Section 4.3.3 to obtain the best value for the threshold T. In this section, we evaluate the impact of the threshold T on the performance of the OSOPF^2. We performed experiments simulating five *available classes* for each dataset, ranging T from 0.0 (all samples are classified as *unknown*) to 1.0 (no sample is classified as *unknown*) stepping by 0.005. The obtained results are plotted in *known–unknown curves* (see Figure 5.8) that show the trade-off between the AKS and the AUS. For each T, we ran 10 experiments with different sets of five *available classes*. In the curve, it is shown that OSOPF^2 is well-behaved despite parameter changing, i.e., a reasonable T estimation ensures the definition of a suitable open-set classifier for usage in an operational scenario. The cross–red point in the curves indicate the point with the best *normalized accuracy* in the testing. The x-axis (AKS) for

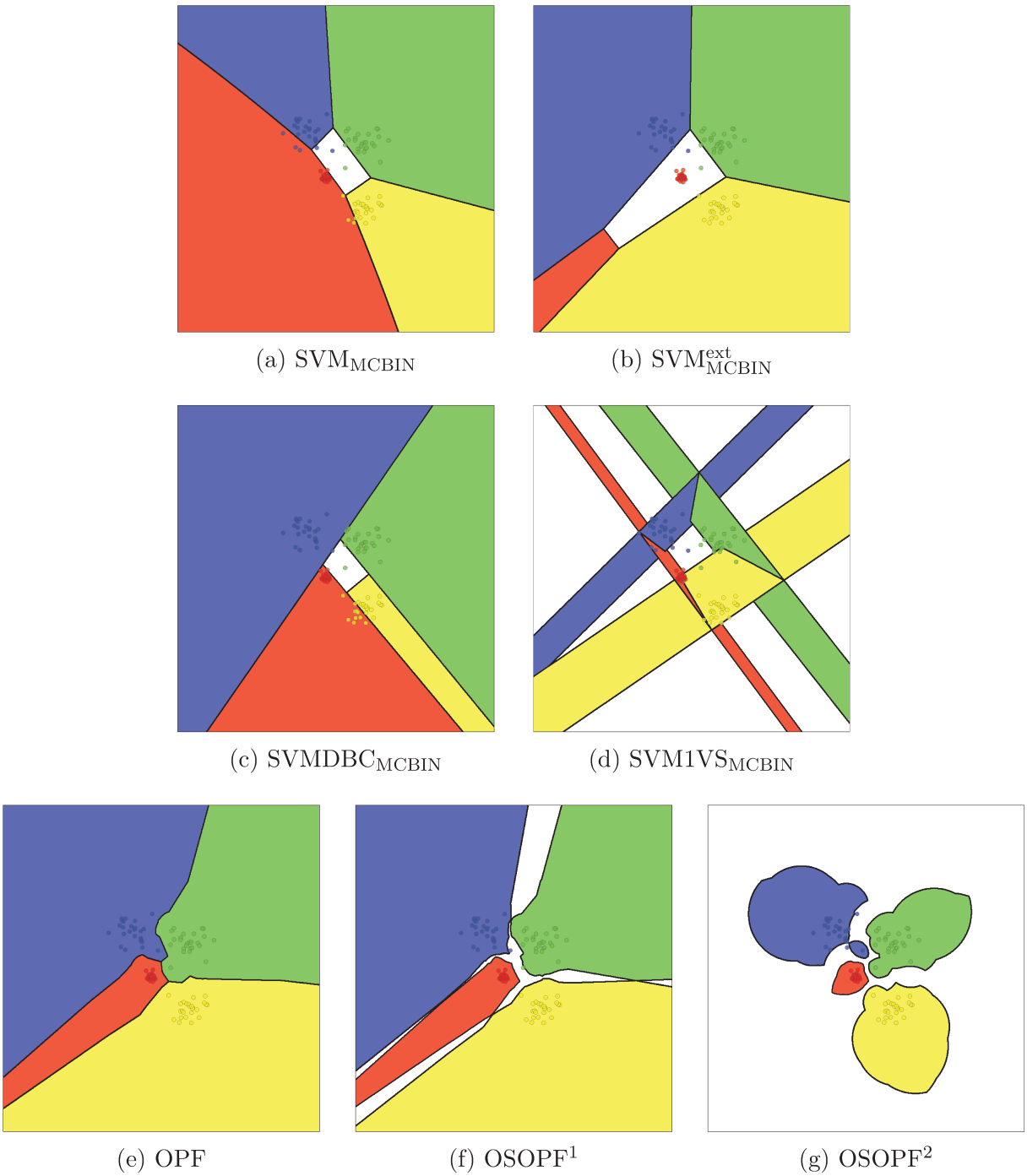


(a) $\text{SVM}_{\text{MCBIN}}$ (b) $\text{SVM}^{\text{ext}}_{\text{MCBIN}}$

(c) $\text{SVMDBC}_{\text{MCBIN}}$ (d) $\text{SVM1VS}_{\text{MCBIN}}$

(e) OPF (f) OSOPF^1 (g) OSOPF^2

Figure 5.5: Decision boundaries for the *Cone-torus* dataset. The non-white regions represent the region in which a test sample would be classified as belonging to the same class of the samples with the same color. All samples in the white regions would be classified as *unknown*.

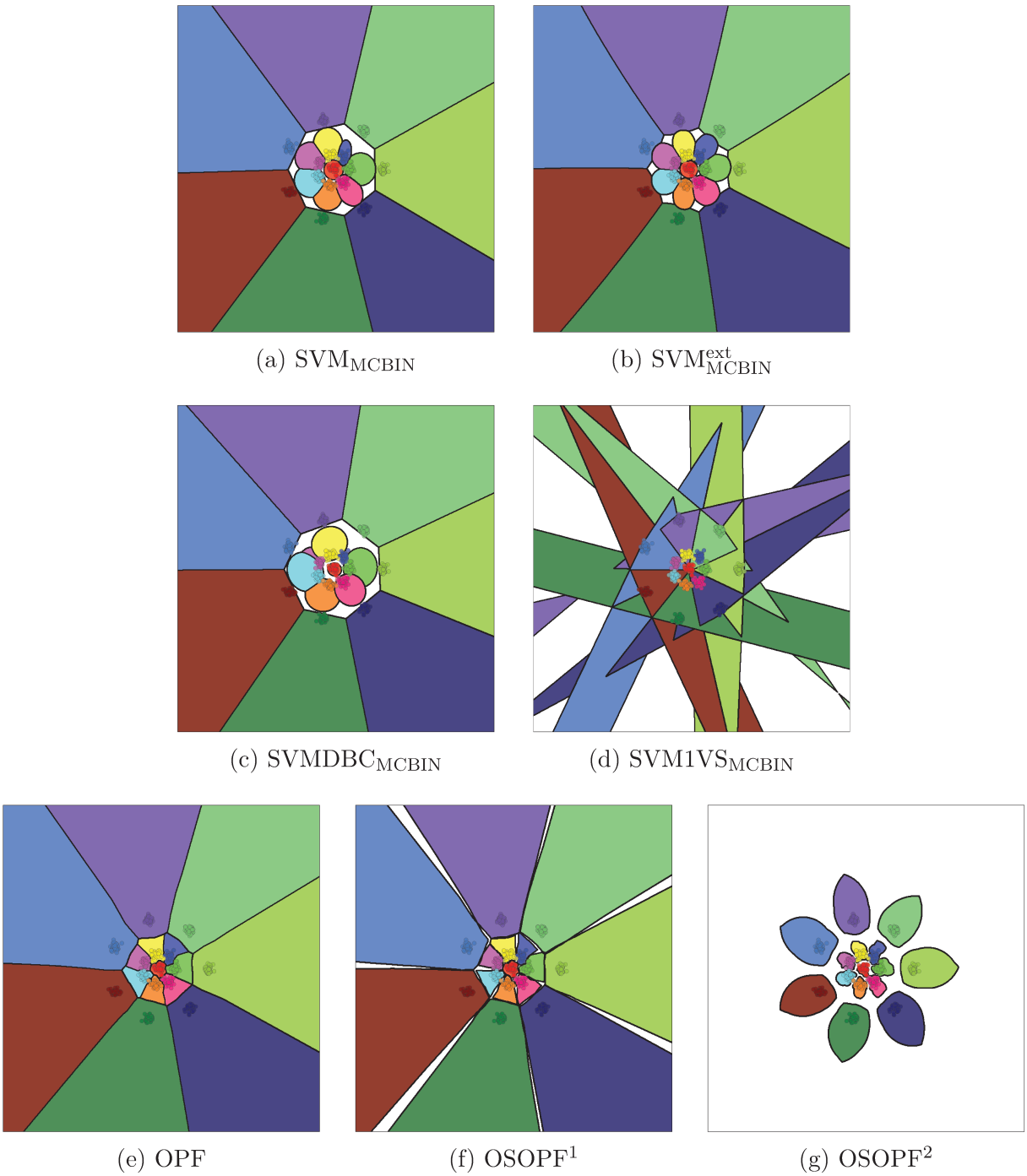


(a) $\text{SVM}_{\text{MCBIN}}$ (b) $\text{SVM}^{\text{ext}}_{\text{MCBIN}}$

(c) $\text{SVMDBC}_{\text{MCBIN}}$ (d) $\text{SVM1VS}_{\text{MCBIN}}$

(e) OPF (f) OSOPF^1 (g) OSOPF^2

Figure 5.6: Decision boundaries for the *Four-gauss* dataset. The non-white regions represent the region in which a test sample would be classified as belonging to the same class of the samples with the same color. All samples in the white regions would be classified as *unknown*.

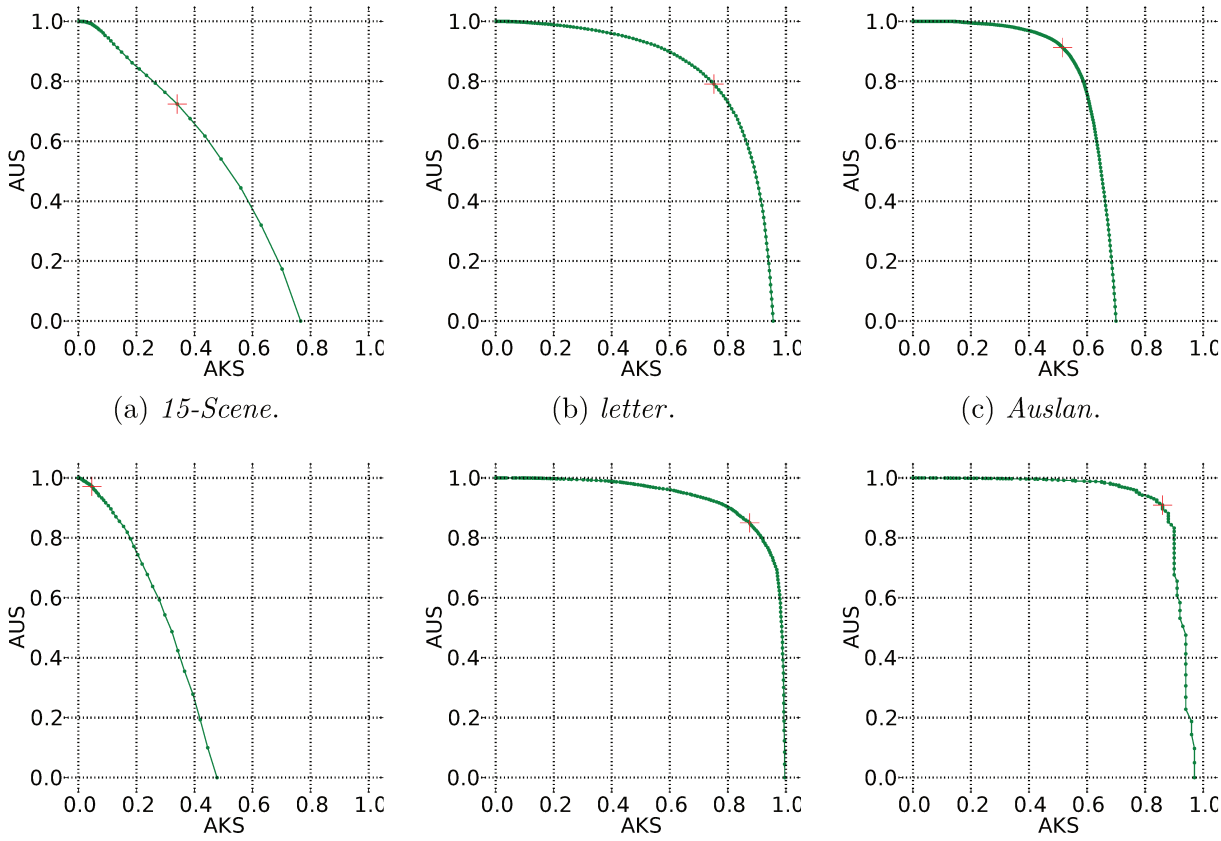


(a) $\text{SVM}_{\text{MCBIN}}$ (b) $\text{SVM}^{\text{ext}}_{\text{MCBIN}}$

(c) $\text{SVMDBC}_{\text{MCBIN}}$ (d) $\text{SVM1VS}_{\text{MCBIN}}$

(e) OPF (f) OSOPF^1 (g) OSOPF^2

Figure 5.7: Decision boundaries for the *R15* dataset. The non-white regions represent the region in which a test sample would be classified as belonging to the same class of the samples with the same color. All samples in the white regions would be classified as *unknown*.



(a) *15-Scene.* (b) *letter.* (c) *Auslan.*

(d) *Caltech-256.* (e) *ALOI.* (f) *ukbench.*

Figure 5.8: *Known–unknown curves* showing the *accuracy of known samples* (AKS; x-axis) and the *accuracy of unknown samples* (AUS; y-axis) used to obtain the *normalized accuracy*. Curve obtained in an open-set classification with 5 *available classes*. The cross–red point has the best value of *normalized accuracy*.

the case in which AUS is 0 indicates the best possible accuracy the OSOPF^2 can obtain when classifying only the samples among the 5 *available classes*.

5.4 Failing case

In Section 5.3, we could see the effectiveness of the proposed OSOPF^2 method and, in this section, we present the failing case of the OSOPF^2. A noticeable failing case happens into the *overlapping regions* of the dataset, i.e., the regions in which two or more classes have some density of training samples. In such cases, the OSOPF^2 likely will wrongly classify an input sample as *unknown* instead of as one of the two or more *overlapping classes*. It increases the *false unknown* of the classifier.

5.4.1 Trying to minimize the *false unknown*

Here, we present a possible alternative to minimize the *false unknown* of the OSOPF^2. However, this modification (called OSOPF^{mc}) did not yield good results. We also present the possible reason for the low performance of the OSOPF^{mc} and infer a possible general principle leading to unsuccessful modifications. We call this modification as OSOPF^{mc} because it takes into account the intraclass *maximum cost*.

The OSOPF^{mc} verifies if the test sample s is in an *overlapping region*. The pertinence verification of s to an *overlapping region* is accomplished when $R > T$, *i.e.*, when s is going to be classified as *unknown*. For the class of the best path and the class of the second best path (in the same way it is obtained for the OSOPF^2), we obtain the maximum costs c_1 and c_2, respectively, among all samples of the class based on OPF's cost function of Equation 3.3. Obtaining the maximum cost of the samples of a class is equivalent to obtaining the maximum arc intraclass. We obtain the neighbor s_1 of s in the first path and the neighbor s_2 of s in the second path. If $w(s, s_1) \leq c_1$ or $w(s, s_2) \leq c_2$, s is classified as belonging to the same class of the *prototype* of the first path, as it would accomplish if $R \leq T$. Otherwise, if $w(s, s_1) > c_1$ and $w(s, s_2) > c_2$, s is classified as *unknown* (see Equation 5.5).

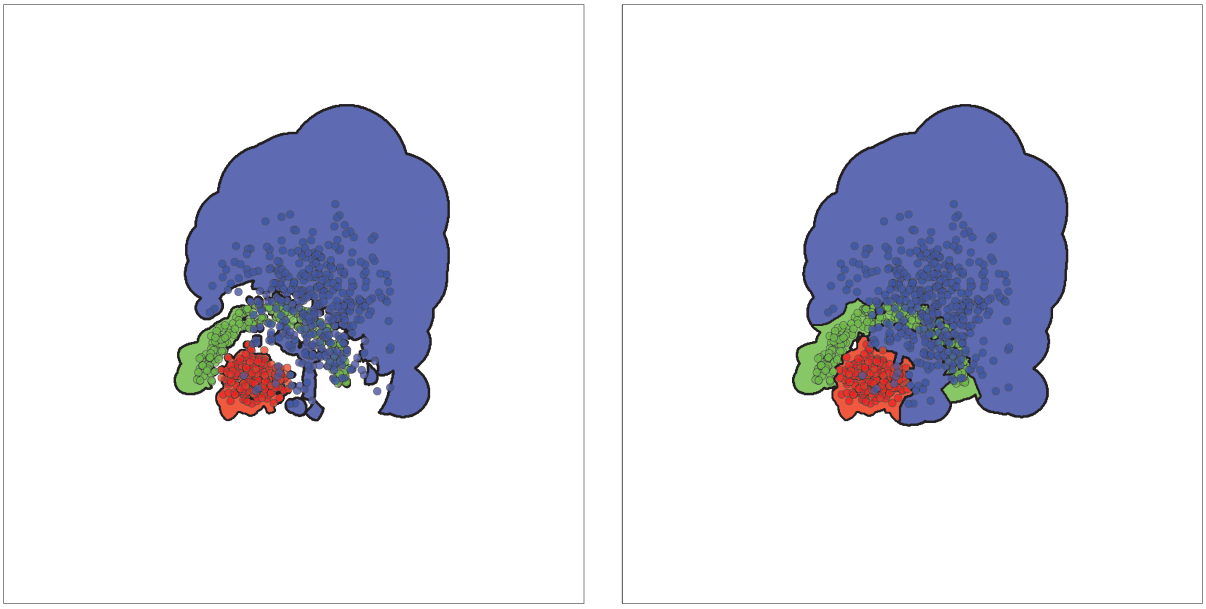
$$L(s) = \begin{cases} L(s_1) & \text{if } R \leq T \text{ or } w(s, s_1) \leq c_1 \text{ or } w(s, s_2) \leq c_2 \\ \ell_0 & \text{if } w(s, s_1) > c_1 \text{ and } w(s, s_2) > c_2 \end{cases} \tag{5.5}$$

where ℓ_0 is the *unknown* label, and $w(s, t)$ is the distance between s and t.

The idea of this verification is to recognize when the test sample s is into an *overlapping region* by comparing the distance from s to its path's nearest neighbor to the maximum intraclass distance of the path's class. When the distance from its neighbor is greater than the intraclass distance, this indicates s is faraway from the path's class and possibly is really *unknown*. When the distance from its neighbor is smaller than the intraclass distance, possibly s is in the influence of the path's class (considering the two best paths) and then it must be classified as the most probable class instead of *unknown* even when $R > T$.

Despite the *boundary images* for simple cases (see Figure 5.9) of the OSOPF^{mc} have some sense, we did not obtain good results with this classifier in high-dimensional spaces. See the statistical comparison in Table 5.6. In fact, its results are very similar to the ones of the traditional classifiers, *e.g.*, $\text{SVM}_{\text{MCBIN}}$, therefore not handling the open-set nature of the problems.

In the OSOPF^2, defining a threshold based on the ratio of similarity scores instead of the similarity score itself led to a good adaptation to the known classes aiming to avoid the *unknown* ones. We believe that the use of the raw similarity score (the OPF's cost



(a) OSOPF^2 (b) OSOPF^{mc}

Figure 5.9: Decision boundaries for the *Cone-torus* dataset comparing the OSOPF^2 and the modification OSOPF^{mc}. The non-white regions represent the region in which a test sample would be classified as belonging to the same class of the samples with the same color. All samples in the white regions would be classified as *unknown*.

Table 5.6: Statistical tests for every dataset. d_1, d_2, d_3, d_4, d_5, and d_6 refer to the following datasets, respectively: *15-Scene*, *letter*, *Auslan*, *Caltech-256*, *ALOI*, and *ukbench*. The arrows point to the winner methods for each pair of compared methods (row and column). Empty cells indicate there is no statistical difference between the pair of methods. The column *ac* indicates the number of *available classes* for training.

		OSOPF^2						OSOPF^{mc}					
	ac	d_1	d_2	d_3	d_4	d_5	d_6	d_1	d_2	d_3	d_4	d_5	d_6
OSOPF^2	3							←	←	←	←		
	6							←	←	←	←	←	
	9							←	←	←	←		
	12							←	←	←	←		
OSOPF^{mc}	3	↑	↑	↑	↑								
	6	↑	↑	↑	↑	↑							
	9	↑	↑	↑	↑								
	12	↑	↑	↑	↑								

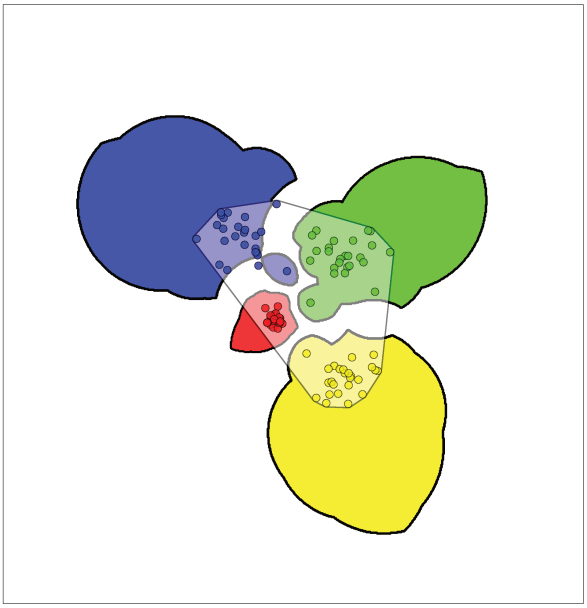


Figure 5.10: Decision boundaries for the *Four-gauss* dataset depicting the *internal open space of risk*. The non-white regions represent the region in which a test sample would be classified as belonging to the same class of the samples with the same color. The non-white regions under the bright region (representing the convex hull) represent the *internal open space of risk*. The other non-white regions outside the bright region represent the *external open space of risk*. All samples in the white regions would be classified as *unknown* and are not *open space of risk*.

function itself) instead of some ratio of similarity scores is the main reason the OSOPF^{mc} fails. Further extending OSOPF^2 to avoid unnecessary *unknown* classification, possibly in the *overlapping regions*, requires further study and is left as future work.

5.5 Further improvement

A possible improvement regards the reduction of OSOPF^2 *external open space of risk*. We refer to *external open space of risk* as the *open space of risk* that is not among the training samples, i.e., the *open space of risk* outside the *convex hull* [15] of the training samples. Note that, in our current implementation, the *internal open space of risk* is already reduced (see Section 5.4). In Figure 5.10, we depict the *internal* and *external open space of risk*. We use the concept of *convex hull* [15] to define the *internal* and *external open space of risk*, however, note that in high dimensional spaces it could not be appropriate to use the concept of *convex hull* to define these expressions due to the *curse of dimensionality* [52].

Throughout this work, we presented the requirement of a bounded *open space of risk* for open-set classification. Our solution to the open-set problem in fact creates a bounded *open space of risk*. However, the *external open space of risk* is not so tight as it would be according to our subjective analysis of the *boundary images* (simple cases; see Figures 5.5g, 5.6g, and 5.7g). We believe that further reducing the *external open space of risk* will improve the *accuracy of unknown samples* of the OSOPF^2 in some open-set problems.

A faraway test sample is classified as *unknown* by the OSOPF^2 when the ratio R between the cost of the best class and the cost of the second best class approaches 1.0. For faraway test samples, R approaches 1.0 as both the cost of the best class and the cost of the second best class are high. The ratio R also approaches 1.0 for a doubtful test sample: when it is between two known classes. We must note that, compared to the doubtful test samples, the ratio R for faraway test samples approaches 1.0 more slowly. That is the cause why the *external open space of risk* is not so tight as the *internal* one.

Chapter 6

Conclusions

There is a lack of open-set classifiers in the literature. Usually, experiments are performed considering that all classes of the problem are *available classes* for training, i.e., a closed-set scenario. However, in real-world situations, the amount of classes during test is many times larger than the known classes. That means that real systems must be able to deal with *unknown* elements that appear only during the system use and not during its development. In this work, we have two main contributions:

- the introduction of two new graph-based open-set classifiers extending upon the original formulation of the closed-set Optimum-Path Forest classifier.
- two new evaluation measures to assess the quality of methods in multiclass open-set classification problems and

To the best of our knowledge, the proposed evaluation measures are the first in the literature for dealing with multiclass and open-set scenarios.

The two proposed open-set classifiers (OSOPF^1 and OSOPF^2) have the advantage of being inherently multiclass (non-binary–based), while the state-of-the-art methods are multiclass from binary. As more classes are available, multiclass-from-binary classifiers get slower, while the proposed classifiers remain with the same efficiency.

Based on the results obtained with one of the proposed measures, we showed that the proposed OSOPF^2 is better than (or equivalent to) OSOPF^1 and the baseline classifiers evaluated ($\text{SVM}_{\text{MCBIN}}$, $\text{SVM}^{\text{ext}}_{\text{MCBIN}}$, $\text{SVMDBC}_{\text{MCBIN}}$, $\text{SVM1VS}_{\text{MCBIN}}$, and OPF) for several datasets: *15-Scene*, *letter*, *Auslan*, *Caltech-256*, *ALOI*, and *ukbench* datasets. We confirmed the superiority of the proposed method using the Analysis of Variance (ANOVA) and the post-test Tukey "Honest Significant Differences" (HSD). As we can see in Figures 5.5g, 5.6g, and 5.7g, only the proposed OSOPF^2 is able to limit the *open space of risk* [43].

Future work will be dedicated to reducing the *external open space of risk* of the OSOPF^2. We refer to *external open space of risk* as the *open space of risk* that is not among the training samples, i.e., the *open space of risk* outside the *convex hull* [15] of the training samples. Note that, in our current implementation, the *internal open space of risk* is already reduced.

Another important improvement for the OSOPF^2 is to reduce the *false unknown* rate. We note that all test samples in the *overlapping regions* are going to be classified as *unknown* because the OSOPF^2 classifies doubtful and faraway samples as *unknown*. *Overlapping regions* refers to the regions with some density of training samples of two or more different training classes. The improvement is to identify if a test sample s is in an *overlapping region* and classify s as the most probable overlapping class instead of *unknown*. It is not a trivial task in high dimensional spaces, as we could see in Section 5.4.

Another future work consists in using the proposed *parameter optimization* for the OSOPF^2 as a general open-set *grid search* procedure and investigating whether this novel *grid search* procedure obtains better parameter estimation than the traditional *grid search* for general classifiers in the open-set scenario.

It is also worth to investigate the use of the OSOPF^2 for *novelty detection*. The *novelty detection* refers to the problem of inferring new classes and aggregate them to the training model when a substantial set of input samples classified as *unknown* is obtained [28]. As the OSOPF^2 creates a bounded *open space of risk* for each class, we believe it can be somehow successfully applied to the problem of *novelty detection*.

Bibliography

[1] Willian P. Amorim, Alexandre X. Falcão, and Marcelo H. Carvalho. Semi-supervised pattern classification using optimum-path forest. In *Conference on Graphics, Patterns and Images*, page to appear. Conference Publishing Services, 2014.

[2] Peter L. Bartlett and Marten H. Wegkamp. Classification with a reject option using a hinge loss. *Journal of Machine Learning Research*, 9(0):1823–1840, 2008.

[3] James Bergstra and Yoshua Bengio. Random search for hyper-parameter optimization. *Journal of Machine Learning Research*, 13(1):281–305, 2012.

[4] Christopher M. Bishop. *Pattern Recognition and Machine Learning*. Information Science and Statistics. Springer, 2006.

[5] Y-Lan Boureau, Francis Bach, Yann LeCun, and Jean Ponce. Learning mid-level features for recognition. In *International Conference on Computer Vision and Pattern Recognition*, pages 2559–2566, San Francisco, CA, USA, June 2010. IEEE Press.

[6] Hakan Cevikalp and Bill Triggs. Efficient object detection using cascades of nearest convex model classifiers. In *International Conference on Computer Vision and Pattern Recognition*, pages 3138–3145, Providence, Rhode Island, 2012. IEEE Press.

[7] Chih-Chung Chang and Chih-Jen Lin. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2(3):1–27, April 2011.

[8] 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, 2005.

[9] S. W. Chew, S. Lucey, P. Lucey, S. Sridharan, and J. F. Cohn. Improved facial expression recognition via uni-hyperplane classification. In *International Conference on Computer Vision and Pattern Recognition*, pages 2554–2561, Providence, Rhode Island, 2012. IEEE Press.

[10] Kai-Min Chung, Wei-Chun Kao, Tony Sun, and Chih-Jen Lin. Decomposition methods for linear support vector machines. *International Conference on Acoustics, Speech, and Signal Processing*, 4:IV–868–871, 2003.

[11] Filipe de O. Costa, Ewerton Silva, Michael Eckmann, Walter J. Scheirer, and Anderson Rocha. Open set source camera attribution and device linking. *Pattern Recognition Letters*, 39:92–101, April 2014.

[12] 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, 2012. IEEE Press.

[13] André Tavares da Silva, Alexandre Xavier Falcão, and Léo Pini Magalhães. Active learning paradigms for CBIR systems based on optimum-path forest classification. *Pattern Recognition*, 44(12):2971–2978, December 2011.

[14] André Tavares da Silva, Jefersson Alex dos Santos, Alexandre Xavier Falcão, Ricardo da S. Torres, and Léo Pini Magalhães. Incorporating multiple distance spaces in optimum-path forest classification to improve feedback-based learning. *Computer Vision and Image Understanding*, 116(4):510–523, April 2012.

[15] Mark de Berg, Otfried Cheong, Marc van Kreveld, and Mark Overmars. *Computational Geometry*, volume 4. Springer-Verlag, Berlin Heidelberg, November 2008.

[16] Alexandre Xavier Falcão, Jorge Stolfi, and Roberto de Alencar Lotufo. The image foresting transform: Theory, algorithms, and applications. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(1):19–29, January 2004.

[17] Peter W. Frey and David J. Slate. Letter recognition using holland-style adaptive classifiers. *Machine Learning*, 6(2):161–182, 1991.

[18] Jan-Mark Geusebroek, Gertjan J. Burghouts, and Arnold W. M. Smeulders. The Amsterdam Library of Object Images. *International Journal of Computer Vision*, 61 (1):103–112, January 2005.

[19] Greg Griffin, Alex Holub, and Pietro Perona. Caltech-256 object category dataset. Technical report, California Institute of Technology, 2007.

[20] 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.

[21] Hongliang Jin, Qingshan Liu, and Hanqing Lu. Face detection using One-class-based support vectors. In *International Conference on Automatic Face and Gesture Recognition*, pages 457–462, Seoul, Korea, 2004. IEEE Press.

[22] Mohammed Waleed Kadous. *Temporal classification: Extending the classification paradigm to multivariate time series*. PhD thesis, The University of New South Wales, 2002.

[23] Ludniila I. Kuncheva and Stefan T. Hadjitodorov. Using diversity in cluster ensembles. In *International Conference on Systems, Man, and Cybernetics*, pages 1214–1219, The Hague, Netherlands, 2004. IEEE Press.

[24] C.H. Lampert, H. Nickisch, and S. Harmeling. Learning to detect unseen object classes by between-class attribute transfer. In *International Conference on Computer Vision and Pattern Recognition*, pages 951–958, Miami, Florida, USA, June 2009. IEEE Press.

[25] S. Lazebnik, C. Schmid, and J. Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *International Conference on Computer Vision and Pattern Recognition*, volume 2, pages 2169–2178, New York, NY, USA, 2006. IEEE Press.

[26] Tomasz Malisiewicz, Abhinav Gupta, and Alexei A. Efros. Ensemble of exemplar-SVMs for object detection and beyond. In *International Conference on Computer Vision*, pages 89–96, Barcelona, Spain, November 2011. IEEE Press.

[27] Larry M. Manevitz and Malik Yousef. One-class SVMs for document classification. *Journal of Machine Learning Research*, 2(0):139–154, 2001.

[28] Markos Markou and Sameer Singh. Novelty detection: a review—part 1: statistical approaches. *Signal Processing*, 83(12):2481–2497, December 2003.

[29] D. Michie, D. J. Spiegelhalter, and C. C. Taylor, editors. *Machine learning, neural and statistical classification*. Ellis Horwood Upper Saddle River, NJ, USA, 1995.

[30] Rupert G. Miller-Jr. Normal Univariate Techniques. In *Simultaneous Statistical Inference*, Springer Series in Statistics, pages 37–108. Springer New York, 1981.

[31] David Nist and Henrik Stew. Scalable recognition with a vocabulary tree. In *International Conference on Computer Vision and Pattern Recognition*, pages 2162–2168, Providence, RI, USA, 2006. IEEE Press.

[32] Mark Palatucci, Dean Pomerleau, Geoffrey Hinton, and Tom M. Mitchell. Zero-shot learning with semantic output codes. *Advances in Neural Information Processing Systems*, 22:1–9, 2009.

[33] J. P. Papa, A. A. Spadotto, A. X. Falcão, and J. C. Pereira. Optimum Path Forest classifier applied to laryngeal pathology detection. In *International Conference on Systems, Signals and Image Processing*, pages 249–252, Bratislava, Slovak Republic, 2008. IEEE Press.

[34] João P. Papa, Alexandre X. Falcão, and Celso T. N. Suzuki. Supervised pattern classification based on optimum-path forest. *International Journal of Imaging Systems and Technology*, 19(2):120–131, June 2009.

[35] João P. Papa, Alexandre X. Falcão, Victor Hugo C. de Albuquerque, and João Manuel R.S. Tavares. Efficient supervised optimum-path forest classification for large datasets. *Pattern Recognition*, 45(1):512–520, January 2012.

[36] 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 *International Symposium on Mathematical Morphology and its Applications to Signal and Image Processing*, pages 337–348, Rio de Janeiro, Brazil, 2007. MCT/INPE.

[37] João Paulo Papa, Alexandre Xavier Falcão, Greice Martins de Freitas, and Ana Maria Heuminski de Ávila. Robust pruning of training patterns for optimum-path forest classification applied to satellite-based rainfall occurrence estimation. *IEEE Geoscience and Remote Sensing Letters*, 7(2):396–400, 2010.

[38] P. Jonathon Phillips, Patrick Grother, and Ross Micheals. Evaluation methods in face recognition. In Stan Z. Li and Anil K. Jain, editors, *Handbook of Face Recognition*, pages 551–574. Springer, London, 2011.

[39] Anderson Rocha and Siome Goldenstein. Multi-class from binary: divide to conquer. In *International Conference on Computer Vision Theory and Applications*, pages 1–8, Lisboa, Portugal, 2009. IEEE Press.

[40] 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):1–13, 2014.

[41] Leonardo Marques Rocha, Fábio A. M. Cappabianco, and Alexandre Xavier Falcão. Data clustering as an optimum-path forest problem with applications in image analysis. *International Journal of Imaging Systems and Technology*, 19(2):50–68, June 2009.

[42] Marcus Rohrbach, Michael Stark, and Bernt Schiele. Evaluating knowledge transfer and zero-shot learning in a large-scale setting. In *International Conference on Computer Vision and Pattern Recognition*, pages 1641–1648, Colorado Springs, CO, USA, 2011. IEEE Press.

[43] Walter Scheirer, Anderson Rocha, Archana Sapkota, and Terrance Boult. Towards open set recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(7):1757 – 1772, July 2013.

[44] Bernhard Schölkopf, John C. Platt, John Shawe-Taylor, Alex J. Smola, and Robert C. Williamson. Estimating the support of a high-dimensional distribution. Technical report, Microsoft Research, 1999.

[45] Marina Sokolova and Guy Lapalme. A systematic analysis of performance measures for classification tasks. *Information Processing and Management*, 45(4):427–437, July 2009.

[46] Roberto Souza, Roberto Lotufo, and Leticia Rittner. A Comparison between Optimum-Path Forest and k-Nearest Neighbors Classifiers. In *Conference on Graphics, Patterns, and Images*, pages 260–267, Ouro Preto, MG, Brazil, August 2012. IEEE Computer Society.

[47] Roberto Souza, Letícia Rittner, and Roberto Lotufo. A comparison between k-Optimum Path Forest and k-Nearest Neighbors supervised classifiers. *Pattern Recognition Letters*, 39(0):2–10, September 2014.

[48] Renato O. Stehling, Mario A. Nascimento, and Alexandre X. Falcão. A compact and efficient image retrieval approach based on border/interior pixel classification. In *International Conference on Information and Knowledge Management*, pages 102–109, McLean, VA, USA, 2002. ACM Press.

[49] Celso T. N. Suzuki, Jancarlo F. Gomes, Alexandre X. Falcão, João P. Papa, and Sumie Hoshino-Shimizu. Automatic segmentation and classification of human intestinal parasites from microscopy images. *IEEE Transactions on Biomedical Engineering*, 60(3):803–812, March 2013.

[50] 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):1271–83, July 2010.

[51] 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.

[52] Roger Weber, Hans-J. Schek, and Stephen Blott. A quantitative analysis and performance study for similarity-search methods in high-dimensional spaces. In *International Conference on Very Large Data Bases*, pages 194–205, San Francisco, CA, USA, 1998. ACM Press.

[53] 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.

[54] B. S. Yandell. *Practical Data Analysis for Designed Experiments*. Chapman & Hall statistics textbook series. Chapman & Hall, 1997.

[55] Xiang Sean Zhou and Thomas S. Huang. Relevance feedback in image retrieval: A comprehensive review. *Multimedia Systems*, 8(6):536–544, April 2003.