







A Multi-Class Probabilistic Optimum-Path Forest

Silas E. Nachif Fernandes¹^a, Leandro A. Passos²^b, Danilo Jodas¹^c, Marco Akio³^d,
André N. Souza³^e and João Paulo Papa¹^f

¹Department of Computing, São Paulo State University, Bauru, Brazil

²School of Engineering and Informatics, University Wolverhampton, Wolverhampton, England, U.K.

³Department of Electrical Engineering, São Paulo State University, Bauru, Brazil

Keywords: Optimum-Path Forest, Probabilistic Classification, Multi-Class.

Abstract: The advent of machine learning provided numerous benefits to humankind, impacting fields such as medicine, military, and entertainment, to cite a few. In most cases, given some instances from a previously known domain, the intelligent algorithm is encharged of predicting a label that categorizes such samples in some learned context. Among several techniques capable of accomplishing such classification tasks, one may refer to Support Vector Machines, Neural Networks, or graph-based classifiers, such as the Optimum-Path Forest (OPF). Even though such a paradigm satisfies a wide sort of problems, others require the predicted class label and the classifier's confidence, i.e., how sure the model is while attributing labels. Recently, an OPF-based variant was proposed to tackle this problem, i.e., the Probabilistic Optimum-Path Forest. Despite its satisfactory results over a considerable number of datasets, it was conceived to deal with binary classification only, thus lacking in the context of multi-class problems. Therefore, this paper proposes the Multi-Class Probabilistic Optimum-Path Forest, an extension designed to outdraw limitations observed in the standard Probabilistic OPF.

1 INTRODUCTION


Machine learning-based approaches became essential in the twenty-first century's daily life, impacting in trivial tasks such as movie recommendations, as well as complex ones, such as safety and health condition predictions, among others. In general, most of these techniques learn patterns from data and assign each compounding sample a label, thus classifying them as a member of a specific group.


Despite the aforementioned paradigm, several problems demand a different approach concerning the classification procedure. Consider, for instance, an automotive insurance company computing the risks associated with each customer profile (Apte et al., 1999). In this scenario, a specialist usually considers the driver's age, gender, vehicle price, vehicle


age, among others (Huang and Meng, 2019), to estimate the probabilities of theft and accidents, thus implying in the final price charged by the company. A common alternative to undertaking such problems engages probabilistic models, which returns a real-valued number denoting the degree of confidence or probability of an event.


Different solutions may include Bayesian approaches, whose inference mechanism is based on a stream of probabilities, and watershed-based models, such as the Probabilistic Watershed (Sanmartin et al., 2019), which considers all possible spanning forests in a graph to compute the probability of connecting a particular seed to a node. Besides, one can consider extending traditional classifiers to create probabilistic models, e.g., the probabilistic Nearest Neighbor (Ma et al., 2020) and the probabilistic Support Vector Machines (SVM) (Platt, 1999).


Considering traditional classification techniques, a graph-based approach called Optimum-Path Forest (OPF) (Papa et al., 2009; Papa et al., 2012) obtained notorious relevance in the last years due to its outstanding results in a wide range of applications,


^a <https://orcid.org/0000-0001-7228-1364>

^b <https://orcid.org/0000-0003-3529-3109>

^c <https://orcid.org/0000-0002-0370-1211>

^d <https://orcid.org/0000-0002-8288-1758>

^e <https://orcid.org/0000-0001-9783-6311>

^f <https://orcid.org/0000-0002-6494-7514>

such as anomaly detection, data oversampling, and medical issues, to cite a few. In short, the OPF is a graph-based framework developed to tackle supervised (Papa et al., 2009; Papa et al., 2017) and unsupervised (Rocha et al., 2009) classification problems, among others. Contextualizing with pertinency-based methods, Souza et. al (Souza et al., 2019) proposed the Fuzzy OPF, a variant that considers each sample’s membership while computing its predicted label. Besides, Fernandes et. al (Fernandes et al., 2018) proposed an OPF extension to deal with probabilistic classification problems, the so-called Probabilistic Optimum-Path Forest.

The Probabilistic OPF extends the “Platt Scaling” concept (Platt, 1999) to the context of the Optimum-Path Forest classifier. In a nutshell, the variant considers the cost assigned to each sample during OPF training and classification steps to approximate the posterior class probability distribution. Even though experiments conducted over distinct scenarios demonstrated that the algorithm is suitable to attack several problems, they regard to binary classification tasks only.

Therefore, this paper proposes two main contributions to address such a drawback: (i) to propose the Multi-Class Probabilistic Optimum-Path (MCP-OPF), an extension of the Probabilistic OPF to carry out probabilistic classification issues in multi-class environments; and (ii) to promote the literature regarding graph-based learning architectures, probabilistic classification, and multi-class handling methods.

The remainder of this paper is presented as follows. Section 2 describes the supervised Optimum-Path Forest, as well as the Probabilistic OPF, while Section 3 introduces the proposed approach to tackle multi-class problems through the Probabilistic OPF. Further, Sections 4 and 5 present the methodology and experiments conducted in work, respectively. Finally, Section 6 states the conclusions and future work.

2 THEORETICAL BACKGROUND

In this section, we present a brief introduction to the Optimum-Path Forest classifier, as well as its extension for probabilistic classification.

2.1 Optimum-Path Forest Classifier

Let $\mathcal{D} = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \dots, (\mathbf{x}_m, y_m)\}$ be a dataset of samples such that $\mathbf{x}_i \in \mathbb{R}^n$ and $y_i \in \{-1, +1\}$. Besides, we have that $\mathcal{D} = \mathcal{D}_1 \cup \mathcal{D}_2 \cup \mathcal{D}_3$, where $\mathcal{D}_1, \mathcal{D}_2,$

and \mathcal{D}_3 denote the training, validation, and testing sets, respectively. The Optimum-path Forest classifier (Papa et al., 2009; Papa et al., 2017) is a graph-based algorithm where the nodes denote data samples, and edges represent connections between each pair of instances. Besides, the most representative samples are selected as prototypes, i.e., the nodes that compete among themselves in a conquering-like process whose objective is offering optimum-path costs to the remaining samples in the graph. Consequently, the training process is succeeded by minimizing a path-cost function f_{max} , described as follows:

$$\begin{aligned} f_{max}(\langle \mathbf{v} \rangle) &= \begin{cases} 0 & \text{if } \mathbf{v} \in \mathcal{P}, \\ +\infty & \text{otherwise} \end{cases} \\ f_{max}(\phi_{\mathbf{v}} \cdot \langle \mathbf{v}, \mathbf{t} \rangle) &= \max\{f_{max}(\phi_{\mathbf{v}}), d(\mathbf{v}, \mathbf{t})\}, \end{aligned} \quad (1)$$

where $\phi_{\mathbf{v}}$ represents a path starting from a root in \mathcal{P} and ending at sample \mathbf{v} , $d(\mathbf{v}, \mathbf{t})$ denotes the distance between samples \mathbf{v} and \mathbf{t} , and \mathcal{P} stands for the set prototypes. Moreover, $\phi_{\mathbf{v}} \cdot \langle \mathbf{v}, \mathbf{t} \rangle$ represents the concatenation between the path $\phi_{\mathbf{v}}$ and the edge $\langle \mathbf{v}, \mathbf{t} \rangle$. In short, $f_{max}(\phi_{\mathbf{v}})$ computes the maximum distance among adjacent samples in the path $\phi_{\mathbf{v}}$.

Let $\mathcal{P}^* \subseteq \mathcal{P}$ be the set of optimum prototypes¹, i.e., a set of adjacent samples with different labels discovered after computing the Minimum Spanning Tree over \mathcal{D}_1 . Such a step is accomplished by assigning an optimum cost $C_{\mathbf{t}}$ to each sample $\mathbf{t} \in \mathcal{D}_1$, i.e.:

$$C_{\mathbf{t}} = \min_{\forall \mathbf{v} \in \mathcal{D}_1} \{\max\{C_{\mathbf{v}}, d(\mathbf{v}, \mathbf{t})\}\}, \quad (2)$$

where \mathbf{v} represents the training instance that conquered \mathbf{t} . The classification step is achieved by discovering the training sample that confers the optimum-path cost to each test instance, computed through Equation 2.

2.2 Probabilistic Optimum-Path Forest

The first change imposed to OPF in order to accomplish probabilistic classification concerns approximating the posterior class probability based on the f_{max} path-cost function (Fernandes et al., 2018), performed as follows:

$$P(\hat{y}_i = y_i | \mathbf{x}_i) \approx P_{A,B}(C_{\mathbf{x}_i}) = \frac{1}{1 + \exp(Ay_i C_{\mathbf{x}_i} + B)}, \quad (3)$$

where A and B are parameters to be learned, $C_{\mathbf{x}_i}$ stands for the cost assigned to sample \mathbf{x}_i during OPF training or classification steps, and \hat{y}_i denotes the label

¹ \mathcal{P}^* denotes the set of optimum prototypes, i.e., the set of prototypes that minimizes the training error over \mathcal{D}_1 .

predicted by the classifier. The rationale behind the proposed approach is to assume the lower the cost assigned to sample \mathbf{x}_i , the higher the probability of that sample be correctly classified. Therefore, the training process considers minimizing the classification error, determined as follows:

$$F(\theta) = \sum_{i=1}^{|\mathcal{D}_2|} (t_i q_i + \log(1 + \exp(-A y_i C_{\mathbf{x}_i} - B))), \quad (4)$$

where $\theta = (A^*, B^*)$ denotes the set of parameters that optimizes the equation, and $q_i = C_{\mathbf{x}_i} + B$. Since probabilistic OPF deals with binary values only, t_i is formulated as follows:

$$t_i = \begin{cases} \frac{N_+ + 1}{N_+ + 2} & \text{if } y_i = +1 \\ \frac{1}{N_- + 2} & \text{if } y_i = -1, \end{cases} \quad (5)$$

where N_+ and N_- stand for the number of positive and negative samples, respectively. Such an approach is considered to handle unbalanced datasets.

Finally, one can attribute a label $+1$ to a sample whose probability $P(\hat{y}_{\mathbf{x}} = 1 | \mathbf{x}) > P(\hat{y}_{\mathbf{x}} = -1 | \mathbf{x})$. Otherwise, the sample is labeled with -1 .

3 Multi-Class Probabilistic Optimum-Path Forest Algorithm

Even though the Probabilistic OPF was designed to tackle binary classification problems in its original formulation (Fernandes et al., 2018), several techniques have been developed to decompose multi-class problems into a variety of simple multi-class problems (Madzarov et al., 2009). Among them, the one-against-all (OvA) (Vapnik, 1999) approach obtained notorious popularity due to its simplicity and effective results. Therefore, such a method was adopted to extend the binary version of the Probabilistic OPF.

To deal with problems composed of K -classes, where $K > 2$, the Multi-Class Probabilistic OPF instantiates K versions of the binary Probabilistic OPF, such that the k -th model is trained to classify k labeled samples as positive, and the remaining as negative, such that $k \in \{1, 2, \dots, K\}$. Further, the testing procedure considers presenting each testing sample to all K binary Probabilistic OPFs and attributing the sample to a label whose respective classifier provided the maximum output among all others.

Algorithm 1 implements the proposed approach. Lines 2 – 8 construct a new k -class set for training and validation purposes. Lines 9 and 10 execute the OPF

training and classification algorithm according to section 2.1. Line 11 is in charge of optimizing parameters A and B , i.e., they aim at computing the best set of parameters using Newton's method with backtracking line search proposed by Platt et al. (Platt, 1999) and further improved by Lin et al. (Lin et al., 2007). Lines 12 – 13 repeats the OPF training and testing algorithm using the original training set. Parameters A and B for the k -th model are then used to compute the probability of each test sample in Lines 14 – 20. The steps mentioned above are repeated during K times to obtain an array of probabilities for each sample in \mathcal{D}_3 . Finally, the loop presented in Lines 21 – 22 assigns each test sample the label whose score obtained the highest probability.

The additional training step (Line 11) provides the parameters A and B by solving the regularized maximum likelihood problem, according to Equation 4. Even though any optimization algorithm could be employed for the task, the optimization approach proposed by Platt et al. (Platt, 1999) and further improved by Lin et al. (Lin et al., 2007), has been proved to be a simple and robust solution, and it has been integrated into LibSVM² source code. Consider the works proposed by Platt et al. (Platt, 1999) and Lin et al. (Lin et al., 2007)³ for more details about the optimization method.

The main drawback related to the OvA approach regards its increasingly training complexity when the number of training samples is large. Such behavior is expected since each of the K classifiers is trained considering the whole training set. Figure 1 illustrates the OvA probabilistic classification idea. Notice the probability of the class pertinence presented in each image denote the values obtained by the Probabilistic OPF after normalizing the outputs obtained through OvA.

4 METHODOLOGY

This section describes the datasets employed in the work. Further, it also presents the setup considered in the experiments.

4.1 Datasets

The experiments performed in this paper were conducted over seven datasets, described as follows:

- **Gases05** (Lupi Filho, 2012): it comprises 1,201

²<https://www.csie.ntu.edu.tw/~cjlin/libsvm/>

³<https://www.csie.ntu.edu.tw/~cjlin/papers/plattprob.pdf>

Algorithm 1: Probabilistic Optimum-Path Forest Algorithm.

Input: A λ -labeled training $\mathcal{G}^{tr} = (\mathcal{D}_1, \mathcal{A})$ and validating $\mathcal{G}^{vl} = (\mathcal{D}_2, \mathcal{A})$ sets, unlabeled test $\mathcal{G}^{ts} = (\mathcal{D}_3, \mathcal{A})$ set, and the number of classes K .

Auxiliary: Optimum-path forest P , cost map C , and label map L .

Output: Probability output p for each sample in \mathcal{D}_3 .

```

1 for each  $k \in \{1, \dots, K\}$  do
    // Build a new  $\lambda$ -labeled
    // training and validation
    // sets
2    $\mathcal{D}_1^k \leftarrow \emptyset$  and  $\mathcal{D}_2^k \leftarrow \emptyset$ ;
3   for each  $d \in \{\mathcal{D}_1, \mathcal{D}_2\}$  do
4     for each  $v \in \mathcal{D}_d$  do
5       if  $\lambda(v) = k$  then
6          $\mathcal{D}_d^k \leftarrow (v, +1)$ ;
7       else
8          $\mathcal{D}_d^k \leftarrow (v, -1)$ ;
9    $P_1 \leftarrow$  OPF Training( $\mathcal{D}_1^k$ );
10   $[C_2, L_2] \leftarrow$  OPF Testing( $P_1, \mathcal{D}_2^k$ );
    // Newton's method with a
    // backtracking line search, a
    // Platt's Probabilistic
    // Output with an improvement
    // from Lin et al. (Lin
    // et al., 2007).
11   $[A^k, B^k] \leftarrow$ 
    Sigmoid Training( $\mathcal{D}_1^k, C_2, L_2$ )
12   $P_1 \leftarrow$  OPF Training( $\mathcal{D}_1^k \cup \mathcal{D}_2^k$ );
13   $[C_3, L_3] \leftarrow$  OPF Testing( $P_1, \mathcal{D}_3$ );
14  for each  $i \in \mathcal{D}_3$  do
15     $fACpB \leftarrow A^k L_3^i C_3^i + B^k$ ;
16     $fACmB \leftarrow A^k L_3^i C_3^i - B^k$ ;
    // Compute sigmoid
    // probability
17    if  $(fACpB) \geq 0$  then
18       $p_i^k \leftarrow \frac{\exp(-fACmB)}{1 + \exp(-fACmB)}$ ;
19    else
20       $p_i^k \leftarrow \frac{1}{1 + \exp(fACpB)}$ ;
21 for each  $i \in \mathcal{D}_3$  do
22   $[k, p_i] \leftarrow \arg \max_{k \in \{1, \dots, K\}} (p_i^k)$ 
    
```

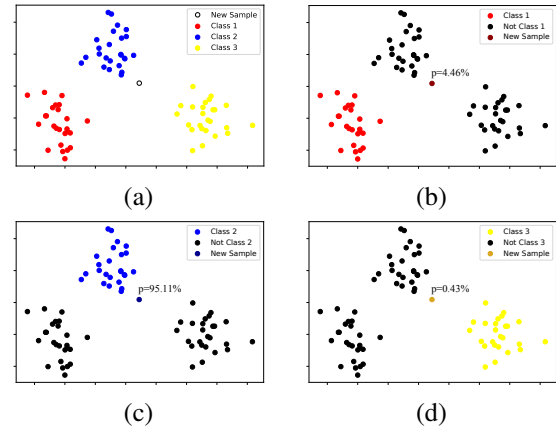


Figure 1: Illustration of the OvA procedure. (a) the 2-D plot considers three distinct classes and a new sample to be classified. Further, frames (b), (c), and (d) depict the normalized probability of this sample belonging to classes 1 (4.46%), 2 (95.11%), and 3 (0.43%), respectively, compared against all other classes.

instances with 5 features each describing dissolved gas analysis in oil-filled power transformers. The dataset is composed of 3 classes, representing normal behavior, thermal faults, and electrical faults.

- **Gases07** (Lupi Filho, 2012): similar to the Gases05 dataset. The difference lies in the number of samples, i.e., 1,144 instances, and the number of features, comprising 7 distinct gasses, instead of 5.
- **Scene** (Boutell et al., 2004): it is composed of 2,407 semantic scene classification samples represented by 294 extracted features each, divided into 6 classes, i.e., Beach, Sunset, Fall foliage, Field, Mountain, and Urban.
- **Sonar** (Gorman and Sejnowski, 1988): it is composed of 208 instances with 60 features each, this dataset is employed to discriminate sonar signals bounced off a metal cylinder or cylindrical rocks.
- **Synthetic**: dataset generated by sampling over a Gaussian distribution. Comprises 1,000 samples with 6 features each. The dataset is divided into 3 classes.
- **Synthetic01** and **Synthetic02**: similar to Synthetic dataset, Synthetic01 and Synthetic02 were generated by sampling over Gaussian distributions. Both of them comprise 1,000 samples with 2 features each. Both are also divided into 2 classes.

4.2 Experimental Setup

The experiments conducted in this paper compare the Multi-Class Probabilistic OPF against six baselines: the standard OPF (Papa et al., 2009), as well as five other probabilistic classifiers adopted, i.e., the probabilistic SVM (Platt, 1999), Naive Bayes (Kuncheva, 2006), Decision Tree (Nagabhushan and Pai, 1999), Linear Discriminant Analysis (LDA) (Al-Dulaimi et al., 2019), and Logistic Regression (Yang and Loog, 2018). The methodology considered for the evaluation employs a data pre-processing procedure using the z-score normalization, described as follows:

$$\mathbf{t}' = \frac{\mathbf{t} - \mu}{\rho}, \quad (6)$$

where μ denotes the mean and ρ stands for the standard deviation. Besides, t and t' correspond to the original and normalized features, respectively.

Further, a cross-validation process with 30 runs is performed for statistical analysis based on the Wilcoxon signed-rank test (Wilcoxon, 1945) with a significance of 0.05. In this scenario, the best method over each dataset (i.e., the one that obtained the highest F1 score) is compared against all other algorithms individually. Furthermore, a post hoc analysis is conducted using the Nemenyi test (Nemenyi, 1963) with $\alpha = 0.05$, which exposes the critical difference (CD) among all techniques. Finally, the runs mentioned above are divided into three groups of 10 runs each, such that the datasets are randomly split as follows:

1. 10 runs using 70% of the samples for training and 30% for testing;
2. 10 runs using 80% of the samples for training and 20% for testing; and
3. 10 runs using 90% of the samples for training and 10% for testing.

The evaluation procedure focuses on finding the set of hyperparameters that maximizes the models' accuracy. In this context, both MCP-OPF and Probabilistic SVM were optimized using Newton's method with a backtracking line search comprising a minimal step of $1e^{-7}$, Hessian's partial derivatives $\sigma = 1e^{-7}$, stopping criteria $\eta = 1e^{-3}$, and the maximal number of iterations equal to $t_{max} = 100$. At the same time, the probabilistic SVM hyperparameters C and γ , and the remaining techniques were optimized using a grid search. The optimization process was performed through a 5-fold cross-validation procedure, i.e., for each fold, 80% of the training set was used to train the model, while the remaining 20% was used for validation purposes. Finally, Table 1 presents the parameter configuration. Notice the probabilistic SVM employs a Radial Basis Function kernel. Additionally,

the standard OPF and Naive Bayes have no hyperparameters to be tuned.

Table 1: Parameter configuration.

Algorithm	Parameters
Decision Tree	criterion \in {'gini', 'entropy'} max_depth \in [2, 15]
LDA	Solver \in {'svd', 'lsqr', 'eigen'} tol \in [$1.0e^{-5}$, $1.0e^{-1}$]
Logistic Regression	Solver \in {'newton-cg', 'lbfgs', 'liblinear', 'sag', 'saga'} $C \in$ {0.01, 0.1, 1.0, 10.0, 100.0}
Probabilistic SVM	$C \in$ {1, 10, 100, 1,000} $\gamma \in$ {0.001, 0.01, 0.1, 1}

Finally, the experiments were conducted over the C-based library LibOPF⁴ for both OPF and MCP-OPF, as well as the well-known Python library Scikit-learn (Pedregosa et al., 2011) considering other techniques. Besides, the computational environment comprises a 3.9 GHz Intel i7 processor with 16 GB of RAM, running over an Ubuntu 16.04 Linux machine.

5 EXPERIMENTS

This section discusses the experimental results considering the proposed Multi-Class Probabilistic OPF classification effectiveness and computational burden. Results presented in bold denote the best values according to the Wilcoxon signed-rank test.

5.1 Effectiveness

Table 2 presents the F1 score and the accuracy⁵ values obtained over the aforementioned datasets⁶. Even though MCP-OPF did not obtain the best results in some cases, it is worth noting some interesting particularities. Regarding the standard OPF, even though its absolute value outperformed the ones obtained by the proposed approach in four-out-seven datasets, i.e., SCENE, SONAR, SYNTHETIC01, and SYNTHETIC02, the accuracy difference is minimal (approximately 0.5%). On the other hand, MCP-OPF is capable of producing results considerably higher than the ones obtained by OPF in some cases, as observed over the GASES05 dataset, where MCP-OPF performed 36% better.

Concerning the remaining techniques, one can observe that MCP-OPF obtained the best results in two-out-of-seven datasets, i.e., SYNTHETIC01 and SYN-

⁴<https://github.com/jppbsi/LibOPF>

⁵By accuracy, this work considers the number of corrected classified instances over the total number of testing samples.

⁶Results presented in bold denote the best values according to the Wilcoxon signed-rank test.

THETIC02, as well as similar statistical results over SONAR datasets and a relatively small accuracy difference over GASES05 and GASES07 ($\approx 2\%$). In fact, SVM presents a considerable upper hand over two datasets, i.e., SCENE and SYNTHETIC. However, such an advantage comes at a price of 9 to 49 times the computational burden cost demanded in the learning process, as presented in the next section.

Notice the purpose of the present work is not to outperforming the standard OPF classifier, but to provide a suitable alternative considering the context of multi-class probabilistic classification. Nevertheless, the Probabilistic OPF obtained results comparable to the standard OPF in the worst case and much better ones in others, e.g., Gases05. Further, together with the standard OPF, the model also outperformed the baselines considering different cases, such as Synthetic01 and Synthetic02, confirming the contribution's relevance.

5.2 Computational Burden

Table 3 exhibits the execution time for the learning (i.e., training + evaluating), expressed in seconds. Since both standard OPF and Naive Bayes do not perform an evaluation step, i.e., they do not present any hyperparameter to be tuned, they present the lowest computational costs. Moreover, the standard Optimum-Path Florest is incorporated as a procedural step executed multiple times in the Multi-Class Probabilistic OPF, as described in Algorithm 1. Therefore, it is convenient to accept a considerable difference in their computational burden. Even though this difference is proportionally significant, the algorithm can provide a substantial gain in some cases, such as the 36% over the GASES05 dataset (Table 2), in an efficient manner if compared to SVM or Logistic Regression, for instance. Indeed, the probabilistic SVM is the most onerous technique in terms of computational resources since its learning time demanded, on average, 256 and 24 times slower than OPF and MCP-OPF, respectively.

Such an in-depth analysis of the results considering both the accuracy and the computational burden shows that MCP-OPF poses itself as an efficient alternative for the task of multi-class probabilistic classification, providing a balance between reasonable accuracies at the cost of an acceptable execution time.

5.3 Statistical Analysis

Despite the statistical analysis performed in Section 5.1, which compares all techniques against the one that obtained the highest F1 score value consid-

ering the Wilcoxon signed-rank test, this section provides an alternative statistical analysis considering the Nemenyi test. Such an approach verifies the presence of critical difference among all techniques and depicts the results in a diagram representing each method's average rank in a horizontal bar (Demšar, 2006), presented in Figure 2. Notice lower ranks denote the better techniques, and methods connected do not significantly differ between themselves.

One can notice that MCP-OPF obtained the best results in four-out-of-seven datasets, i.e., Gases07, Sonar, Synthetic01, and Synthetic02, performing better than most of the techniques. Considering a solo comparison against the standard OPF classifier, MCP-OPF obtained statistically similar or better results. Regarding the other techniques, the Nemenyi test could not provide explicitly correlated patterns among themselves, with exception to Logistic regression and the LDA, which were considered similar in all situations. Such results confirm the suitability of the MCP-OPF for multi-class probabilistic classification tasks, once it provided a satisfactory classification performance in a reasonably efficient fashion.

6 CONCLUSION

This paper proposes the Multi-Class Probabilistic Optimum-Path Forest, a variant of the OPF classifier designed to deal with probabilistic classification problems over datasets composed of more than two classes. Experiments conducted over seven datasets showed the method is capable of outperforming the standard OPF with considerable difference in some cases or obtaining pretty approximate results in others. Concerning the other techniques, the proposed MCP-OPF obtained the best results over four datasets considering the Nemenyi test, i.e., Gases07, Sonar, Synthetic01, and Synthetic02, as well as comparative effectiveness considering Gases05 dataset. Moreover, the proposed approach showed itself computationally efficient, performing way faster than traditional techniques, such as the Probabilistic SVM and Logistic Regression, for instance.

ACKNOWLEDGEMENTS

The authors are grateful to FAPESP grants #2013/07375-0, #2014/12236-1, #2017/02286-0, #2018/21934-5, #2019/07665-4, #2019/18287-0, and #2020/12101-0, CNPq grants #307066/2017-7, and #427968/2018-6, as well as the UK Engineering and

Table 2: F1 scores and accuracies of the techniques considered in the work.

Algorithm	Metric	MCP-OPF	OPF	Decision Tree	LDA	Logistic Regression	Naive Bayes	Probabilistic SVM
Gases05	F1	0.9207 ± 0.0107	0.6771 ± 0.3113	0.9462 ± 0.0162	0.9117 ± 0.0016	0.9121 ± 0.0024	0.9315 ± 0.0112	0.9392 ± 0.0090
	Accuracy	89.5210 ± 2.5432	68.8818 ± 30.0493	94.5281 ± 1.5203	86.1681 ± 1.0248	87.2736 ± 1.0018	91.7969 ± 1.2873	93.1041 ± 1.2255
Gases07	F1	0.9424 ± 0.0092	0.9256 ± 0.0222	0.9491 ± 0.0141	0.9338 ± 0.0050	0.9349 ± 0.0024	0.9390 ± 0.0129	0.9515 ± 0.0075
	Accuracy	92.5401 ± 1.8654	92.3230 ± 1.9651	94.7790 ± 1.3552	89.6237 ± 1.2499	90.2258 ± 0.9331	92.8461 ± 1.1565	94.2133 ± 1.0482
Scene	F1	0.6809 ± 0.0224	0.6861 ± 0.0223	0.6045 ± 0.0247	0.6990 ± 0.0228	0.7559 ± 0.0178	0.6526 ± 0.0295	0.8069 ± 0.0164
	Accuracy	68.3735 ± 2.2019	68.8663 ± 2.1772	60.7388 ± 2.4528	70.1881 ± 2.2343	75.6337 ± 1.7203	65.3087 ± 2.7866	80.6974 ± 1.5899
Sonar	F1	0.8444 ± 0.0458	0.8454 ± 0.0456	0.7155 ± 0.0613	0.7159 ± 0.0657	0.7525 ± 0.0542	0.6572 ± 0.0577	0.8597 ± 0.0501
	Accuracy	84.3362 ± 4.5961	84.4404 ± 4.5781	71.2518 ± 6.2225	71.3648 ± 6.5491	75.0477 ± 5.4414	65.1180 ± 6.3343	85.9040 ± 5.0213
Synthetic	F1	0.8376 ± 0.0217	0.8360 ± 0.0228	0.8793 ± 0.0199	0.8808 ± 0.0213	0.8896 ± 0.0206	0.8873 ± 0.0190	0.8966 ± 0.0182
	Accuracy	83.7388 ± 2.2380	83.6064 ± 2.3354	87.9310 ± 1.9876	87.9698 ± 2.2083	88.8961 ± 2.0998	88.6087 ± 1.9734	89.5588 ± 1.8843
Synthetic01	F1	0.6180 ± 0.0263	0.6211 ± 0.0280	0.5892 ± 0.0215	0.4749 ± 0.0401	0.5184 ± 0.0601	0.5260 ± 0.0350	0.5657 ± 0.0268
	Accuracy	61.7222 ± 2.6775	62.0444 ± 2.8281	58.3556 ± 2.2055	47.3667 ± 3.9752	48.7444 ± 2.9331	51.5833 ± 3.5535	56.3778 ± 3.0218
Synthetic02	F1	0.9029 ± 0.0212	0.9089 ± 0.0182	0.8917 ± 0.0207	0.4768 ± 0.1017	0.5510 ± 0.0712	0.4649 ± 0.0989	0.8448 ± 0.0308
	Accuracy	90.2833 ± 2.1201	90.8889 ± 1.8188	89.1556 ± 2.0794	47.5944 ± 10.1727	52.8389 ± 6.7255	45.9389 ± 10.1153	84.4667 ± 3.0817

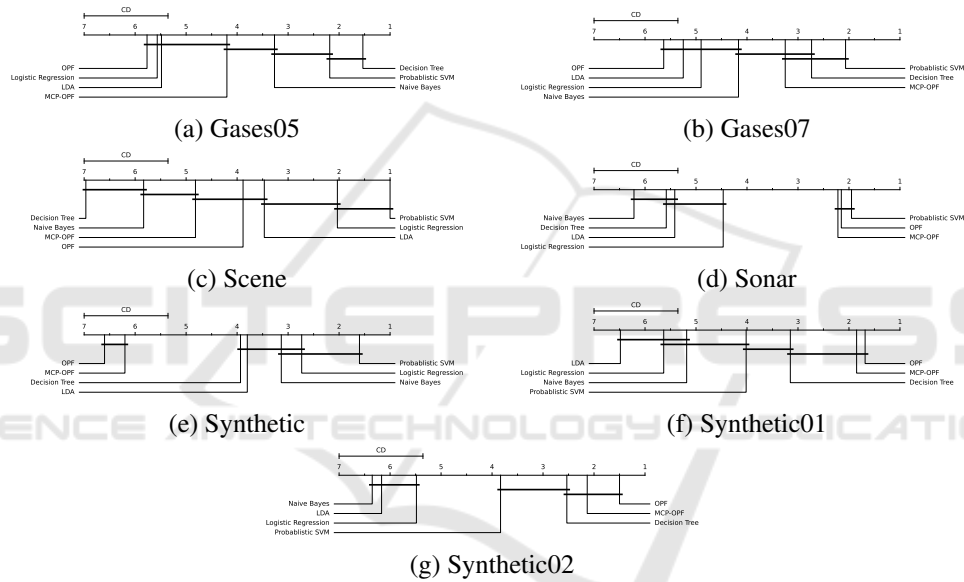


Figure 2: Comparison of Nemenyi test concerning all techniques over (a) Gases05, (b) Gases07, (c) Scene, (d) Sonar, (e) Synthetic, (f) Synthetic01, and (g) Synthetic02 datasets. Groups connected does not present a critical difference.

Table 3: Learning (training + evaluating) time, in seconds.

Algorithm	MCP-OPF	OPF	Decision Tree	LDA	Logistic Regression	Naive Bayes	Probabilistic SVM
Gases05	0.3228 ± 0.0965	0.0257 ± 0.0071	0.1148 ± 0.0135	0.0285 ± 0.0053	1.2868 ± 0.1762	0.0012 ± 0.0004	4.4331 ± 1.0348
Gases07	0.4669 ± 0.1281	0.0426 ± 0.0143	0.1491 ± 0.0120	0.0344 ± 0.0088	1.4851 ± 0.2446	0.0012 ± 0.0003	4.2941 ± 1.0569
Scene	32.4470 ± 6.5194	1.0816 ± 0.2626	49.1499 ± 15.2594	4.7377 ± 1.3074	133.1621 ± 13.4707	0.0084 ± 0.0012	291.4601 ± 51.9725
Sonar	0.0273 ± 0.0060	0.0029 ± 0.0007	0.2563 ± 0.0314	0.0962 ± 0.0497	0.8907 ± 0.2161	0.0010 ± 0.0002	1.3286 ± 0.3268
Synthetic	0.3734 ± 0.0708	0.0344 ± 0.0126	0.2427 ± 0.0419	0.0326 ± 0.0081	1.6555 ± 0.2443	0.0012 ± 0.0002	6.2055 ± 1.0522
Synthetic01	0.2278 ± 0.0611	0.0261 ± 0.0105	0.1426 ± 0.0176	0.0252 ± 0.0057	0.3356 ± 0.0609	0.0011 ± 0.0002	8.9314 ± 1.8717
Synthetic02	0.2679 ± 0.0672	0.0321 ± 0.0098	0.1269 ± 0.0166	0.0264 ± 0.0057	0.3369 ± 0.0589	0.0010 ± 0.0002	8.5503 ± 1.8026

Physical Sciences Research Council (EPSRC) Grant Ref. EP/T021063/1.

REFERENCES

Al-Dulaimi, K., Chandran, V., Nguyen, K., Banks, J., and Tomeo-Reyes, I. (2019). Benchmarking hep-2 speci-

- men cells classification using linear discriminant analysis on higher order spectra features of cell shape. *Pattern Recognition Letters*, 125:534–541.
- Apte, C., Grossman, E., Pednault, E. P., Rosen, B. K., Tipu, F. A., and White, B. (1999). Probabilistic estimation-based data mining for discovering insurance risks. *IEEE Intelligent Systems and their Applications*, 14(6):49–58.
- Boutell, M. R., Luo, J., Shen, X., and Brown, C. M. (2004). Learning multi-label scene classification. *Pattern recognition*, 37(9):1757–1771.
- Demšar, J. (2006). Statistical comparisons of classifiers over multiple data sets. *The Journal of Machine Learning Research*, 7:1–30.
- Fernandes, S. E., Pereira, D. R., Ramos, C. C., Souza, A. N., Gastaldello, D. S., and Papa, J. P. (2018). A probabilistic optimum-path forest classifier for non-technical losses detection. *IEEE Transactions on Smart Grid*, 10(3):3226–3235.
- Gorman, R. P. and Sejnowski, T. J. (1988). Analysis of hidden units in a layered network trained to classify sonar targets. *Neural networks*, 1(1):75–89.
- Huang, Y. and Meng, S. (2019). Automobile insurance classification ratemaking based on telematics driving data. *Decision Support Systems*, 127:113156.
- Kuncheva, L. I. (2006). On the optimality of naïve bayes with dependent binary features. *Pattern Recognition Letters*, 27(7):830–837.
- Lin, H.-T., Lin, C.-J., and Weng, R. C. (2007). A note on platt’s probabilistic outputs for support vector machines. *Machine Learning*, 68(3):267–276.
- Lupi Filho, G. (2012). *Comparação entre os critérios de diagnósticos por análise cromatográfica de gases dissolvidos em óleo isolante de transformador de potência*. PhD thesis, Universidade de São Paulo.
- Ma, J., Xiao, B., and Deng, C. (2020). Graph based semi-supervised classification with probabilistic nearest neighbors. *Pattern Recognition Letters*, 133:94–101.
- Madzarov, G., Gjorgjevikj, D., and Chorbev, I. (2009). A multi-class svm classifier utilizing binary decision tree. *Informatica*, 33(2).
- Nagabhushan, P. and Pai, R. M. (1999). Modified region decomposition method and optimal depth decision tree in the recognition of non-uniform sized characters—an experimentation with kannada characters. *Pattern Recognition Letters*, 20(14):1467–1475.
- Nemenyi, P. (1963). *Distribution-free Multiple Comparisons*. Princeton University.
- Papa, J. P., Falcão, A. X., Albuquerque, V. H. C., and Tavares, J. M. R. S. (2012). Efficient supervised optimum-path forest classification for large datasets. *Pattern Recognition*, 45(1):512–520.
- Papa, J. P., Falcão, A. X., and Suzuki, C. T. N. (2009). Supervised pattern classification based on optimum-path forest. *International Journal of Imaging Systems and Technology*, 19(2):120–131.
- Papa, J. P., Fernandes, S. E. N., and Falcão, A. X. (2017). Optimum-Path Forest based on k-connectivity: Theory and Applications. *Pattern Recognition Letters*, 87:117–126.
- Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.
- Platt, J. C. (1999). Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. In *Advances in Large Margin Classifiers*, pages 61–74. MIT Press.
- Rocha, L. M., Cappabianco, F. A. M., and Falcão, A. X. (2009). Data clustering as an optimum-path forest problem with applications in image analysis. *International Journal of Imaging Systems and Technology*, 19(2):50–68.
- Sanmartin, E. F., Damrich, S., and Hamprecht, F. A. (2019). Probabilistic watershed: Sampling all spanning forests for seeded segmentation and semi-supervised learning. In *Advances in Neural Information Processing Systems*, pages 2776–2787.
- Souza, R. W. R., Oliveira, J. V. C., Passos, L. A., Ding, W., Papa, J. P., and Albuquerque, V. H. (2019). A novel approach for optimum-path forest classification using fuzzy logic. *IEEE Transactions on Fuzzy Systems*.
- Vapnik, V. N. (1999). An overview of statistical learning theory. *IEEE transactions on neural networks*, 10(5):988–999.
- Wilcoxon, F. (1945). Individual comparisons by ranking methods. *Biometrics Bulletin*, 1(6):80–83.
- Yang, Y. and Loog, M. (2018). A benchmark and comparison of active learning for logistic regression. *Pattern Recognition*, 83:401–415.