Supervised Learning Using Local Analysis in an Optimal-Path Forest

Willian Paraguassu Amorim
FACOM - Institute of Computing
Federal University of Mato Grosso do Sul - UFMS
Campo Grande, Brazil
Email: paraguassuec@gmail.com

Marcelo Henriques de Carvalho
FACOM - Institute of Computing
Federal University of Mato Grosso do Sul - UFMS
Campo Grande, Brazil
Email: mhc@facom.ufms.br

***Abstract*—In this paper, we present an OPF-LA (Optimal Path Forest–Local Analysis), a new learning model proposal. OPF-LA is a heuristic that uses local information for selecting prototypes that, in turn, will be used to classify new data. It employs the main ideas of an OPF classifier, suggesting a new procedure in the data training phase. Experimental results show the advantages in efficiency and accuracy over classical learning algorithms in areas such as Support Vector Machines (SVM), Artificial Neural Networks using Multilayer Perceptrons (MP), and Optimal Path Forest (OPF), in several applications.**

***Keywords*-Supervised classifiers; Optimal-Path Forest;**

I. Introduction

Pattern recognition is a research area which aims to classify patterns into categories or classes. Given a set of c classes, $\omega_1, \omega_2, \cdots, \omega_c$, and a pattern, x, a pattern recognition system associates the pattern x to the label i of one of the classes ω_i. The pattern classification problem is divided into the following classes: (i) supervised, where each input pattern is identified as a member of a predefined class, (ii) unsupervised, where each input pattern is assigned to a class as yet unknown, and (iii) semi-supervised, where part of the input set has a predefined class [1].

The main approaches to learning and pattern classification are based on statistical analysis. Simple statistical techniques can easily handle linearly separable classes, as shown in Figure 1(a), but piecewise representations, as shown in Figure 1(b), require more robust techniques, such as Artificial Neural Networks. Unfortunately, many applications involve non linearly-separable classes, as shown in Figure 1(c). Possible solutions in these cases are Support Vector Machines (SVM) [2] and the classic k-nearest neighbours algorithm [3].

One of the most important features of sample spaces, which has not received much attention in supervised classification, is the relation of distance between samples (specially along sequence of samples). A recent research that explores this relationship has obtained promising results for supervised and unsupervised learning using the OPF (Optimal Path Forest) classifier [4][5].

OPF is a supervised pattern classification framework, particularly effective in image classification, which reduces the pattern classification problem to the problem of partitioning the vertices of a graph.

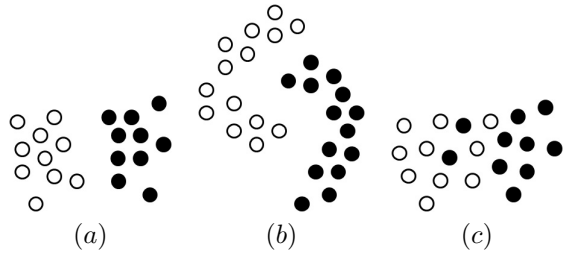

Figure 1. Examples of feature spaces: (a) Linearly separable. (b) Piecewise linearly separable. (c) Non separable.

The problem of pattern recognition can be modeled as a complete graph with positive weights on its arcs, where the nodes are the samples, represented by their respective feature vectors, and the arcs are defined by an adjacency relation between samples [6], [7], [8]. The vertices of the graph can thus be partitioned into optimal-path trees rooted at their respective prototypes (seeds) obtained in the training phase. The label of the most closely connected prototype gives the classification of a new input sample.

The calculation of the Optimal Paths for new samples is performed using the Image Forest Transform (IFT) algorithm [8]. The IFT technique is essentially Dijkstra's algorithm, modified to receive various sources and more general cost functions. It initially assigns the minimum cost function to the source nodes and propagates it to the other nodes in nondecreasing order, partitioning the graph into an Optimal-Paths Forest where the roots are the prototypes. In this paper, we present OPF-LA (Optimal Path Forest–Local Analysis), a new model of supervised classifier. It employs the main ideas of OPF classifiers, suggesting a new procedure in the training phase. We also present a generalization of the OPF-LA technique, which expands the number of prototypes representing each class, increasing the space of possibilities to control data sorting.

The results show that OPF-LA outperforms Support Vector Machines (SVM), Artificial Neural Networks using Multilayer Perceptrons (MP), and Optimal Path Forest (OPF), in the majority of applications, in accuracy, precision, and

recall metrics. We use the T-Student hypothesis test applied to the accuracy rate, recall and precision to assess the performance of the classifiers and analyze if there are differences in the application of the techniques. This paper is organized as follows: The OPF Classifier is presented in Section II, and the OPF-LA Classifier is presented in Section III. In Section IV, the experimental results are presented. Finally, in Section V, the conclusions are drawn.

II. The OPF classifier

The OPF [6], [7] technique allows to create pattern classifiers from a training set, a subset of prototypes, and a path-cost function in the training graph, and seeks to group samples with similar attributes. The objective is to partition the training graph into a forest of minimum cost paths, where each tree is rooted at a prototype and all samples of the tree are labeled with the same label of the root. The classification of a new sample is given by finding the prototype that offers the optimal path among the paths offered by all prototypes, i.e., the classification is based on the strength of connectedness between the sample and the most closely connected prototype or the most strongly connected prototype. We shall present here both OPF and OPF-LA techniques.

Let Z_1 be the training set, let Z_2 be the evaluating data set, and let Z_3 be the data used for classifying the samples. The set Z_1 is used to create the learning model, and Z_3 is used to measure the accuracy of the classified data from the learning model generated by Z_1. A pseudo-test on Z_2 is used to teach the classifier by randomly interchanging samples of Z_1 with misclassified samples of Z_2. After learning, an improvement is expected in the accuracy on Z_3. The reason for dividing the data set is to improve, if necessary, the learning from the errors found.

Let $Z = (Z_1 \cup Z_2 \cup Z_3)$, and let $\lambda(s)$ be the function that assigns the correct label i, $i = 1, 2, ..., c$, of class i to any sample $s \in Z$. Let $S \subset Z_1$ be a set of prototypes of all classes, and let v be an algorithm which extracts n attributes, such as texture, colour, geometric shape, etc., from any sample $s \in Z$ and returns a vector $\vec{v}(s)$.

The distance $d(s,t) \geq 0$, between two samples, s and t, is that between their feature vectors $\vec{v}(s)$ and $\vec{v}(t)$. One can use any distance function suitable for the extracted features. The most common is the Euclidean norm $||\ \vec{v}(t)$ - $\vec{v}(s)\ ||$ used by Papa et al.[4]. The goal is to produce a classifier that has the capacity to predict the label $\lambda(s)$ for any sample $s \in Z_3$.

The training data determines a subset $S^* \subset Z_1$ of prototypes and a discrete optimal partition of Z_1 into the feature space, with an optimum-path forest rooted in S^*. The classification of any sample $s \in Z_3$ is made by evaluating the optimum paths incrementally, as though it were part of the forest, and assigning to it the label of the most strongly connected prototype.

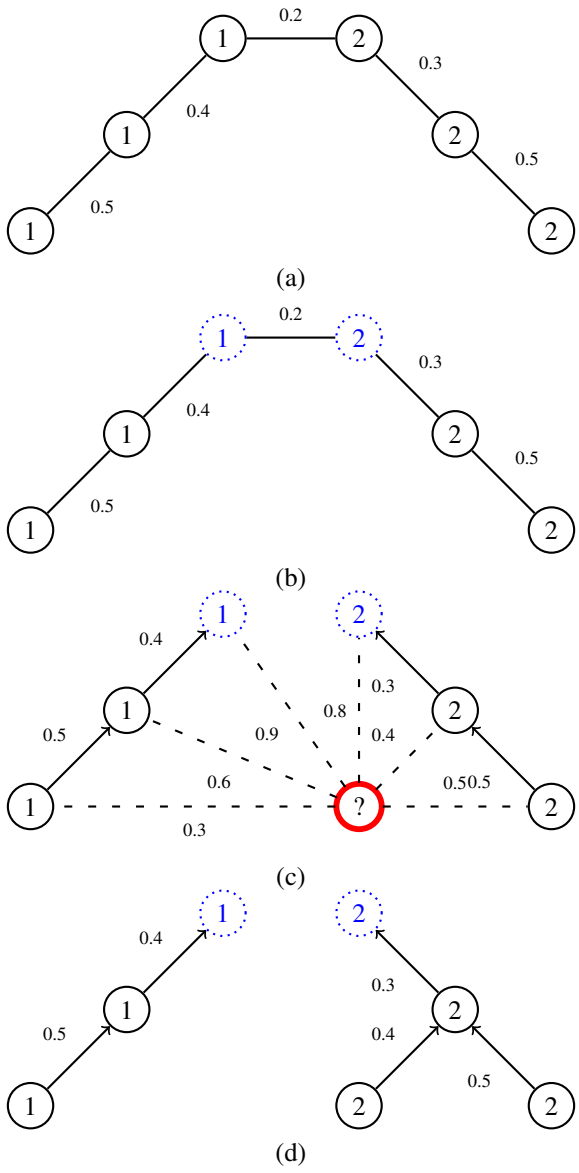

Figure 2. (a) An MST graph referring to some complete training graph. (b) Prototypes selected from the heuristic of adjacent elements in different classes. (c) An optimum-path forest rooted at the prototypes (the numbers indicate cost and label, respectively), a test sample, and its possible connections with all elements in the training graph. (d) Optimum path to the most closely connected prototype.

A. Training Phase

Let (Z_1, A) be a complete graph whose nodes are the training samples and where any pair of samples defines an arc in $A = Z_1 \times Z_1$. The training consists of finding a set S^* of prototypes, considered the most representative. Several heuristics can be adopted. However, the selected model can affect the performance of the classifier.

Papa et al.[4] suggests selecting the prototypes by exploiting the theoretical relation between the minimum-spanning

tree (MST) and the optimum-path tree.

Consider a complete graph whose nodes are all samples of Z_1, and whose arcs are undirected and weighted by the distances between adjacent samples. Now, compute an MST in this graph. Figure 2(a) illustrates the result of this process. In the MST, every pair of samples is connected by a single path which is optimum according to a function f_{max} [4]. That is, the minimum-spanning tree contains one optimum-path tree for any selected root node. The selected prototypes are the closest elements of the MST in different classes, as illustrated in Figure 2(b). By removing the arcs between different classes, the prototypes will be adjacent samples in S^*. A class may contain several prototypes, but we must always ensure that every class has at least one prototype.

Algorithm 1: Algorithm OPF

Input: Set Z_1, prototypes $S^* \subset Z_1$ and a pair of values (v, d) for feature extraction and calculation of distances.
Result: Optimal-Path Forest P, map of optimum values V, and map labels L.
Data: Priority queue Q and variable tmp

1 **foreach** $s \in Z/\ S^*$ **do**
2 $\quad V(s) \longleftarrow +\infty$,
3 **foreach** $s \in S^*$ **do**
4 $\quad V(s) \longleftarrow 0$, $P(s) \longleftarrow nil$, $L(s) \longleftarrow \lambda(s)$, and insert s in Q.
5 **while** *Q is not empty* **do**
6 $\quad$ Remove from Q a sample s such that $V(s)$ is minimal.
7 $\quad$ **foreach** *t ∈ Z such that* $t \neq s$ *and* $V(t) > V(s)$ **do**
8 $\quad\quad$ Calculate tmp $\longleftarrow max\{V(s), d(s,t)\}$.
9 $\quad\quad$ **if** $tmp < V(t)$ **then**
10 $\quad\quad\quad$ **if** *V(t)* $\neq +\infty$ **then**
11 $\quad\quad\quad\quad$ Remove t of Q.
12 $\quad\quad\quad$ $P(t) \longleftarrow s$
13 $\quad\quad\quad$ $L(t) \longleftarrow L(s)$
14 $\quad\quad\quad$ $V(t) \longleftarrow tmp$
15 $\quad\quad\quad$ Insert t in Q.
16 **return** *P,L,V*

Once we know the set S^*, Algorithm 1, the Optimal-Path Forest (OPF), propagates the labels of the prototypes for all the samples of their optimal paths trees, forming a map of labels $L(s) \in \{1, 2, ..., c\}$. Lines 1–4 initialize the maps and insert prototypes in Q. The main loop starts at line 5 and calculates optimal paths from S^* to all samples s in a non-decreasing order of minimum cost. At each iteration, a minimum cost path π_s with optimal value $V(s)$ is obtained in P when we remove its last node s from Q (line 6). Draws are resolved by FIFO policy in Q, that is, when two optimal paths reach the same sample s, this sample is associated with the first path that reached it. In line 7, $V(t) > V(s)$ is false when s can not modify the attributes of t and $V(t) \neq +\infty$ in lines 10 and 11, when only $t \in Q$. The other lines check if the path $\pi_s.\langle s, t\rangle$ is better than the current path $\pi_t \in P$. If so, update Q, $V(t)$, $L(t)$ and $P(t)$.

B. Classification Phase

For classifying a new sample t, consider all nodes in the training set connected to t, as though t were part of the training graph (Figure 2(c)). From all possible paths S^* to t, the goal is to find the optimum path $P^*(t)$ from S^* to t, and label t with the class $\lambda(R(t))$ of its most strongly connected prototype $R(t) \in S^*$. The path can be found incrementally by evaluating the optimum cost $V(t)$ as in Equation 1.

$$V(t) = min\{max\{V(s), d(s,t)\}\}. \quad (1)$$

Let $s^* \in Z_1$ be a node that satisfies Equation 1 (predecessor $P(t) = s^*$). Since $L(s^*) = \lambda(R(t))$, the classification assigns the label $L(s^*)$ as the class of t (Figure 2(d)).

III. The OPF-LA Classifier

The new approach differs the OPF technique in the training phase. The OPF training phase uses a complete graph to estimate the frontier prototypes of classes. The OPF-LA estimates the prototypes at the points of high concentration samples. The main motivation of this paper is to demonstrate that regions of higher concentration of samples per class can offer better candidates to prototypes, besides being a new methodology for supervised classifiers based on Optimum-path Forest using other adjacency relationships and a different way to find prototypes.

We propose a change in the training phase of the Optimum-path forest (OPF) classifier by analysing the data locally in each class, fixing a number $k > 0$, and selecting k prototypes which have the smallest shortest-path accumulated distance to the other vertices of the graph. More precisely, the k selected prototypes of each class will be those that have the k smallest average shortest-path distances to the other vertices of the class. The aim of this technique is to select the k most 'central' samples of each class as prototypes.

In order to perform a local analysis of each class, we modify the data representation format suggested by OPF [4], by using an adjacency relation for each pair of samples, providing a complete graph. The cumulative shortest-path distance between a vertex and other vertices in the graph is obtained by using Dijkstras algorithm (i.e., the IFT algorithm with additive path-cost function), well known in graph theory. The proposed technique is called optimum-path forest local analysis (OPF-LA).

The training process starts by fixing a number $k > 0$ and selecting k prototypes from each class using the following procedure:

1) Fix a class C and consider the (complete) graph G induced by the vertices of C;
2) For each vertex v in C, use the Dijkstra algorithm to calculate the sum of the shortest-path distances between v and all other vertices in its class;
3) The k prototypes selected are those that have the k shortest-path accumulated distances;
4) Add a new vertex x (dummy node) to the graph and add k edges of weight 0 connecting x to the k selected prototypes;
5) Apply Dijkstra's algorithm again, starting with the new vertex x, to determine the distance from x to all the other vertices in its class. The application of Dijkstra returns a tree, T, containing the shortest paths from x to any other vertex of its class;
6) Remove x from T. As a result, we have k trees for each class, whose roots are the prototypes.

We define the best value of k^* as the value that maximizes the classification accuracy of the Z_3 data on Z_1 until it stabilizes at best value, given by Equation 2. The evaluating data set Z_2 is not used. The classification phase is the same as the OPF classifier.

$$k^* = max(accuracy(Z_1, k)), \tag{2}$$

and $k = 1, 2, ..., C_{min}$. The value C_{min} is the sample size of the smaller class.

IV. Experimental Results

We evaluate the effectiveness of OPF-LA using some combinations of public datasets, such as MPEG-7 (using shapes-contours of images, illustrated in Figure 3) and public datasets that use the (x,y) coordinates of 2D points: CONES-TORUS (Figure 4(a)), SATURN (Figure 4(b)), PETALS (Figure 4(c)) and BOAT (Figure 4(d)) [9]. More details about the MPEG-7 feature extraction algorithm can be found in [10]. We also use the problem of classification of defects in cowhide. These databases allow us to evaluate the performance of classifiers in accordance with descriptors of shape, color and texture. The main reason for selecting samples of defects is the great complexity of evaluation, especially in areas close to the vicinity of different defects. A set of five types of regions of interest in the Wet-Blue[1] processing stage were selected, namely scabies, ticks, hot-iron, cut, and regions without defect (Figure 5). The images were obtained in real leather classification in Brazil, and experts performed their classifications.

[1]Wet-Blue leather is an intermediate stage between untanned and finished leather.

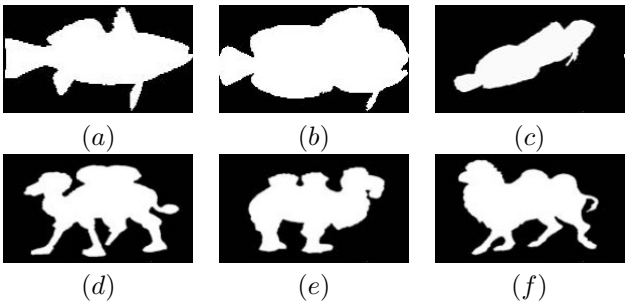

Figure 3. Examples of shapes of MPEG-7 with classes (a)-(c) fishes and (d)-(f) camels.

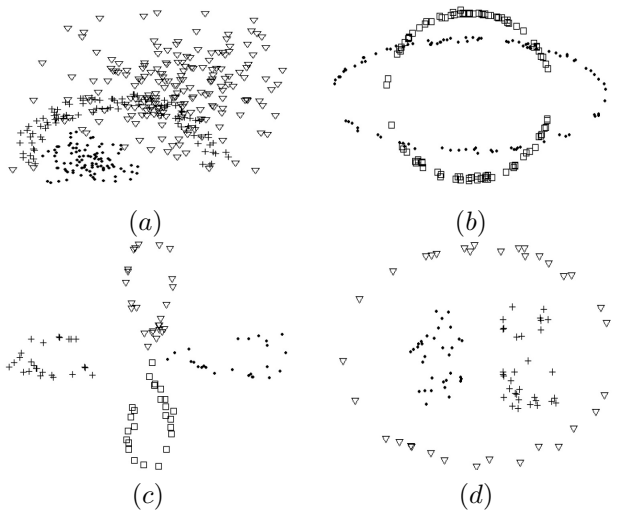

Figure 4. Datasets of 2D points: (a) Cones-torus, (b) Saturn, (c) Petals, and (d) Boat.

Figure 5. Images of a defect in wet-blue. (a) Scabies, (b) Tick, (c) Hot-iron, (d) Cut, (e) without defect.

The defects were manually segmented with the help of the tool DTCOURO[2]. The segmentation process generated samples proportional to the area of each defect in the range of 40x40 pixels, and resulted in 1,690 samples: 369 ticks, 338 hot-iron, 400 cut, 369 scabies, and 214 samples without defect. For each sample, 160 attributes were extracted as follows: 12 colour attributes (HSB and RGB space), 7 from Interaction Maps, 126 from co-occurrence matrices, and 15 from Gabor Filter Banks. Table I contains the number of samples and classes of the six datasets used in the experiments.

[2]Automatic classification system of leather defects – http://trac.gpec.ucdb.br/wiki/site_dtcouro.

Table I
NUMBER OF SAMPLES AND CLASSES OF THE DATASETS

Dataset	Samples	Classes
COWHIDE	1690	5
MPEG-7	1400	70
CONES-TORUS	400	3
SATURN	200	2
PETALS	100	4
BOAT	100	3

A. Experimental Setup

WEKA[3] was used to perform the experiments. WEKA is open source software that presents a collection of machine learning algorithms for data mining tasks. We implemented both OPF and OPF-LA in the WEKA classification format[4]. The implementation of both approaches in this format will facilitate the generation of new experiments and dissemination of the work presented here. For analysis of each algorithm, we use fivefold cross validation and two validations to get better time statistics. From the total samples, 80% were taken for learning, and the remaining 20% were used for the validation tests. This was repeated twice for each of the five validation groups. From these ten results, an average was then evaluated to obtain a hit ratio percentage.

The experiment session of WEKA was used to measure the OPF and OPF-LA classifiers performance compared with other supervised learning classifiers available in WEKA: SVM and MP. Configuration used for SVM: complexity parameter (1.0), epsilon for round-off error (1.0E-12), filter type (Normalize traning data) and kernel (Polynomial kernel). Configuration used for MP: backpropagation, hidden layers of the neural network ((attribs + classes)/2), value of updating the weights (0.3), momentum applied to the weights during updating (0.2) and normalize attributes.

We evaluated accuracy (classification rate) in the format $x \pm y(z)$, where x, y and z are, respectively, the mean accuracy, its standard deviation, and the mean execution time (efficiency in training and classification) in seconds. We also measured precision (capturing the effect of the high number of negative examples in the performance of algorithms), recall (representing the rate of true positives) and application of hypothesis testing T-students with significance level 5%. The WEKA tool calculates the T-Students. So, when the result of an experiment is obtained, a circle can be present next to the result of the classifier. This circle shows that the classifier won or lost the OPF-LA classifier of the tables of results. Empty circle says that the classifier won the OPF-LA.

B. Numerical Results

To choose the number of prototypes for OPF-LA for each problem, an experiment was performed, varying k, starting from 1 to the total number of samples of the lower class. As the value was stabilized, the value of k with the first measurement found with better classification rate was selected. Table II presents the values of k^* for each problem.

Table II
BEST VALUE OF k FOR EACH PROBLEM

Dataset	k^*
COWHIDE	200
MPEG-7	10
CONES-TORUS	50
SATURN	40
PETALS	11
BOAT	13

Table III presents the mean accuracy and the standard deviation. Figure 6 presents a graph of the normalized execution times for each problem, comparing the SVM algorithms, MP, OPF, and OPF-LA with the best value of k. Precision and recall metrics vary from 0 to 1, with 0 meaning the worst and 1 the best performance. The results for these metrics are shown in Tables IV and V.

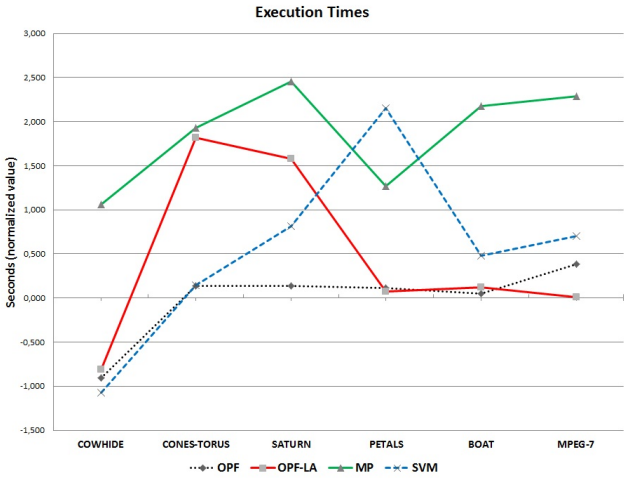

Figure 6. Execution Times in seconds (normalized value)

We consider the overall performance of the OPF-LA classifier satisfactory. Only in the problem of classification of cowhide did the MP technique achieved better results,

Table IV
PRECISION (○, ● STATISTICALLY SIGNIFICANT IMPROVEMENT OR DEGRADATION).

Dataset	OPF	OPF-LA	MP	SVM
COWHIDE	0.999	0.999	**1.000**	0.840
CONES-TORUS	**0.908**	0.871	0.707 ●	0.706 ●
SATURN	**0.905**	0.829	0.739	0.393
PETALS	0.981	**1.000**	0.981	**1.000**
BOAT	**0.949**	**0.949**	0.715	0.682 ●
MPEG-7	0.810	0.810	**0.834**	**0.834**

[3]http://www.cs.waikato.ac.nz/ml/weka
[4]http://code.google.com/p/opf-weka

Table III
MEAN ACCURACY (○, ● STATISTICALLY SIGNIFICANT IMPROVEMENT OR DEGRADATION).

Dataset	OPF	OPF-LA	MP	SVM
COWHIDE	$95.74 \pm 1.04(7.78)$	$96.30 \pm 1.02(12.01)$	**99.97±0.06(89.60)**	$94.53 \pm 0.57(1.02)$
CONES-TORUS	$83.38 \pm 0.48(0.33)$	**83.88±0.85(4.31)**	$70.75 \pm 1.44(4.57)$ ●	$73.88 \pm 1.80(0.34)$ ●
SATURN	**87.50±5.57(0.06)**	$84.50 \pm 4.20(0.70)$	$81.00 \pm 4.08(1.09)$	$46.25 \pm 9.07(0.36)$
PETALS	$99.50 \pm 1.00(0.08)$	**100.00±0.00(0.05)**	$99.50 \pm 1.00(0.88)$	$98.50 \pm 1.91(1.49)$
BOAT	**97.00±2.58(0.02)**	$96.50 \pm 3.00(0.05)$	$75.50 \pm 4.43(0.91)$	$70.50 \pm 5.26(0.20)$ ●
MPEG-7	$78.43 \pm 0.84(110.4)$	**78.64±0.86(2.95)**	$71.39 \pm 1.10(652.23)$ ●	$73.82 \pm 0.62(201.62)$ ●

Table V
RECALL (○, ● STATISTICALLY SIGNIFICANT IMPROVEMENT OR DEGRADATION).

Dataset	OPF	OPF-LA	MP	SVM
COWHIDE	0.958	0.962	**1.000**	0.996
CONES-TORUS	0.913	**0.984**	0.924 ●	0.978 ●
SATURN	0.850	0.880	**0.965**	0.415
PETALS	**1.000**	**1.000**	**1.000**	0.960
BOAT	**1.000**	**1.000**	**1.000**	**1.000**
MPEG-7	0.825	0.825	0.850	**0.925**

with a classification rate of 99.97%. For the other five problems, the OPF and the new proposed OPF-LA obtained better results.

The precision and recall values demonstrate the convenience of using supervised learning algorithms to deal with the problem of classification. The OPF-LA and OPF runtimes was much lower than those of the MP and SVM algorithms. However, we believe that further optimizations and simplifications of the algorithm can accelerate even more the learning and classification processes. We also affirm by T-Students hypothesis test that OPF and OPF-LA classifiers do not show statistical difference in the classification of data in the presented problems.

V. CONCLUSIONS

In this paper, we presented a new heuristic, OPF-LA, for supervised learning in pattern recognition. We also presented data classification results comparing OPF-LA with OPF and other techniques for the problems of analysis and classification of defects in cowhide and classifications and combinations of public datasets. The results showed that the new proposed OPF-LA, with adjustments to the parameter k, can obtain satisfactory classification results compared to classical algorithms in the area. A major advantage of using optimal-path forest techniques in the learning process is that it explores the relation of connectedness between samples, which might give an optimization gain on the generated learning space, and thus yields better results.

We have also developed a WEKA-OPF library containing the OPF classifiers with complete graph and the new proposed OPF-LA in order to disseminate the use of Optimal-Path Forest in the learning process and data classification. We are currently conducting experiments in biotechnology, in applications such as the monitoring of larvae, the counting and classification of yeast in the generation of ethanol, the classification of pollen grains, and face recognition, to better assess the effectiveness of OPF-LA in different contexts.

REFERENCES

[1] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification*, 2nd ed. New York: Wiley, 2001.

[2] B. E. Boser et al., "A training algorithm for optimal margin classifiers," in *5th annual ACM workshop on computational learning theory*. ACM Press, 1992, pp. 144–152.

[3] J. Laaksonen and E. Oja, "Classification with learning k-nearest neighbors," in *ICNN 96: The 1996 IEEE International Conference on Neural Networks - IEEE*, vol. 3, 1996, pp. 1480–1483.

[4] J. P. Papa, A. X. Falcão, and C. T. M. Suzuki, "Supervised pattern classification based on optimum-path forest," *International Journal of Imaging Systems and Technology*, pp. 120–131, 2009.

[5] L. M. Rocha, F. A. M. Cappabianco, and A. X. Falcã, "Data clustering as an optimum-path forest problem with applications in image analysis," *International Journal of Imaging Systems and Technology*, vol. 19, pp. 50–68, 2009.

[6] J. P. Papa, A. X. Falcão, P. A. V. Miranda, C. T. N. Suzuki, and N. D. A. Marcarenhas, "Design of robust pattern classifier based on optimum-path forest," in *Mathematical Morphology and its Applications to Signal and Image Processing (ISMM)*, 2007, pp. 337–348.

[7] J. P. Papa and A. X. Falcão, "A new variant of the optimum-path forest classifier," in *4th Internation Symposium on Visual Computing*, 2008.

[8] A. X. Falcão, J. Stolfi, and R. A. Lotufo, "The image foresting transform: Theory, algorithms, and applications," *IEEE Transactions on Patterns Analysis and Machine Intelligence*, vol. 26, no. 1, pp. 19–29, Jan 2004.

[9] R. Kohavi, "A study of cross-validation and bootstrap for accuracy estimation and model selection," *IJCAI*, p. 11371145, 1995.

[10] J. Papa, "Supervised pattern classification based on optimum-path forest," Ph.D. dissertation, Institute of Computing, University of Campinas, 2008.