Optimizing Optimum-Path Forest Classification for Huge Datasets

João P. Papa
Department of Computing
UNESP - Univ Estadual Paulista
Bauru, Brazil
papa@fc.unesp.br

Fábio A. M. Cappabianco, Alexandre X. Falcão
Institute of Computing
University of Campinas
Campinas, Brazil
fcappabianco@gmail.com,afalcao@ic.unicamp.br

Abstract—Traditional pattern recognition techniques can not handle the classification of large datasets with both efficiency and effectiveness. In this context, the Optimum-Path Forest (OPF) classifier was recently introduced, trying to achieve high recognition rates and low computational cost. Although OPF was much faster than Support Vector Machines for training, it was slightly slower for classification. In this paper, we present the Efficient OPF (EOPF), which is an enhanced and faster version of the traditional OPF, and validate it for the automatic recognition of white matter and gray matter in magnetic resonance images of the human brain.

Keywords-Optimum-Path Forest; Supervised Classification; Support Vector Machines; Brain Image Classification;

I. INTRODUCTION

Usual applications of pattern recognition have hundreds or thousands of samples. However, high resolution image classification may define another context of recognizing patterns. An image may define a huge dataset, generally with millions of pixels as samples (e.g, pictures, remote sensing and tomographic images). In this context, the recognition of pixels's attributes can be performed in a very fast manner. Some of these applications with huge datasets require learning techniques, and also the recognition task can be performed as an interactive or automatic approach.

The interactive recognition (supervised and semi-supervised learning) requires, for instance, a retraining after each iteration (e.g., for marking background and foreground image pixels) in several applications of broad interest, such as interactive segmentation of natural images and medical magnetic resonance (MR) images of the human brain. In this context, some challenges need to be pointed out: (a) to minimize the user time and interaction in the process; (b) to provide controlling for the user for that the segmentation/classification after learning does not affect the results obtained and already accepted; and (c) to maximize the accuracy and precision of the segmentation/classification. In such a way, the retraining of millions of pixels (huge datasets) followed by the classification of the remaining ones needs to be performed in a few seconds.

The learning of a classifier is the most expensive step regarding the computational effort. Traditional and widely used techniques such as Artificial Neural Networks (ANN),

Support Vector Machines (SVM) and Bayesian classifiers try to fit their parameters or make assumptions about shape and the training samples separability in the feature space, aiming to minimize the classification error on that set. An ANN with Multilayer Perceptrons (ANN-MLP), ANN with Radial Basis Function (ANN-RBF) and Kohonen Self Organizing Maps (SOM), for instance, can easily handle linear, piecewise linear and nonlinear separable problems. Bayesian classifiers can also address these situations.

However, more complicated situations, such as non separable problems, require a more detailed study about the classifier to be used. Although algorithms based on neural networks have been extensively used, they are not adequate for handling huge datasets, due to their high computational burden for training. The choice of their parameters, such as the neural network architecture for ANN-MLP and ANN-RBF, and the size of the neural lattice for SOM, for instance, has a intrinsic relation with their accuracies. Hence, it not guaranteed that a more complex neural network architecture will effectively solve the problem [1].

Support Vector Machines (SVM) have been proposed to overcome the problem, by assuming linearly separable classes in a higher-dimensional feature space. However, its computational cost rapidly increases with the training set size and the number of support vectors. As a binary classifier, multiple SVM are required to solve a multi-class problem. Hence, the assumption of a separability of the mapped feature space, in which the SVM theory are based, may also not be valid in any space of finite dimension.

The above scenario evidences that the traditional pattern recognition techniques are not adequate to handle huge datasets with both efficiency and effectiveness. To achieve high recognition rates, they pay the price of a high computational burden for training patterns. However, a novel framework for designing graph-based classifiers that reduce the pattern recognition problem as an optimum-path forest (OPF) computation in the feature space induced by a graph was recently presented in a supervised version by Papa et al. [2], and the unsupervised approach was addressed by Rocha et al. [3].

The OPF-based classifiers did not interpret the classification task as a hyperplanes optimization problem, but as a

1051-4651/10 $26.00 © 2010 IEEE
DOI 10.1109/ICPR.2010.1012

4162

IEEE
computer
society

polynomial combinatorial optimum-path computation from some key samples (*prototypes*) to the remaining nodes. Each prototype becomes a root from its optimum-path tree and each node is classified according to its strongly connected prototype, that defines a discrete optimal partition (influence region) of the feature space. The OPF classifiers have some advantages with respect to the traditional approaches: (i) do not assume any shape/separability of the feature space, (ii) run training phase faster and (iii) make decisions based on a global criteria.

Although the supervised OPF have been demonstrated to be faster than SVM for training [2], if the number of support vectors is small and the test set is huge, SVM tend to be a little faster than OPF for classification [4]. Given the increasing number of applications involving huge datasets, we exploit a theoretical property of the OPF to considerably reduce its classification time. In order to demonstrate the importance of this contribution, we address the problem of white matter and gray matter (WM/GM) automatic classification from MR images of the human brain. In this application, each voxel is considered as a sample for the dataset, which is composed by several millions of samples. The remainder of this paper is organized as follows. Section II introduces the OPF theory. Sections III and IV present, respectively, the new algorithm for supervised OPF efficient classification and the experimental results, in which we compared the proposed OPF (i.e., the Efficient OPF - EOPF) against the traditional OPF and Support Vector Machines. Section V states conclusions.

II. PATTERN RECOGNITION BY OPF

Let $\mathcal{Z}$ be a dataset with $|\mathcal{Z}|$ samples (voxels in this paper), and $\mathcal{Z}_1$ and $\mathcal{Z}_2$ be training and test sets, in which $\mathcal{Z} = \mathcal{Z}_1 \cup \mathcal{Z}_2$ and $\mathcal{Z}_1 \cap \mathcal{Z}_2 = \emptyset$. A graph $(\mathcal{Z}_1, \mathcal{A})$ may be defined by taking each voxel as a node and an *adjacency relation* $\mathcal{A}$ between nodes of $\mathcal{Z}_1$ to form the arcs. Let $S \subset \mathcal{Z}_1$ be a set of prototypes of all classes (i.e., key samples that best represent the classes), and let v be an algorithm which extracts n attributes from any sample $s \in \mathcal{Z}_1 \cup \mathcal{Z}_2$ and returns a vector $\vec{v}(s) \in \Re^n$. The distance $d(s, t)$ between two samples, s and t, is the one between their feature vectors $\vec{v}(s)$ and $\vec{v}(t)$.

Given a graph, a *path* π can be defined as a sequence of distinct samples $\pi = \langle s_1, s_2, \ldots, s_k \rangle$, where $(s_i, s_{i+1}) \in A$ for $s_i \in \mathcal{Z}_1$ and $1 \leq i \leq k - 1$. A path is said *trivial* if $\pi = \langle s_1 \rangle$. Let $f(\pi)$ be a *connectivity function* that associates to a path π a real value. A path π is considered optimum if $f(\pi) \leq f(\pi')$ for any other path π', where π and π' end at a same sample s_k. We also denote by $\pi \cdot \langle s, t \rangle$ the concatenation of a path π with terminus at s and an arc (s, t), which is weighted by the distance of their corresponding feature vectors.

The OPF-based algorithms assign one optimum path $P^*(s)$ from S to every sample $s \in \mathcal{Z}_1$, forming an optimum path forest P (a function with no cycles which assigns

to each $s \in \mathcal{Z}_1 \backslash S$ its predecessor $P(s)$ in $P^*(s)$ or a marker nil when $s \in S$). Let $R(s) \in S$ be the root of $P^*(s)$ which can be reached from $P(s)$. The OPF algorithms compute for each $s \in \mathcal{Z}_1$, the cost $C(s)$ of $P^*(s)$, the label $L(s) = \lambda(R(s))$, and the predecessor $P(s)$. Notice that $\lambda(s)$ means the label of sample s.

Papa et al. [2] and Rocha et al. [3] proposed, respectively, the supervised and unsupervised versions of the OPF classifier. The main difference between them rely on the training set $\mathcal{Z}_1$, which are $\lambda-$labeled in the former approach, in the connectivity function, in the methodology to find prototypes and in the adjacency relation. However, both methodologies are based on the Image Foresting Transform (IFT) algorithm [5], which is the core of the OPF framework. The IFT is a tool for graph-based image processing, and the OPF essentially extends the IFT to the feature space domain. One interesting characteristic of IFT algorithm concerns with the possibility of changing just one parameter (connectivity function, the methodology to find prototypes, the adjacency relation and the domain of the problem), and one can obtain a new algorithm (e.g., the IFT was already applied for several different applications, such as image segmentation, morphological reconstruction and Medial Axis Transform computation [5]). The OPF framework algorithm is described as follows.

Algorithm 1: – OPF FRAMEWORK ALGORITHM

INPUT: *Graph $(\mathcal{N}, \mathcal{A})$ and connectivity function f.*
OUTPUT: *Optimum-path forest P, its connectivity cost map C and its root map R.*
AUXILIARY: *Priority queue Q and variable tmp.*

1. **For each** $t \in \mathcal{N}$, **do**
2. $P(t) \leftarrow nil$, $R(t) \leftarrow t$ and $C(t) \leftarrow f(\langle t \rangle)$.
3.   **If** $C(t) \neq +\infty$, **then** insert t in Q.
4. **While** $Q \neq \emptyset$, **do**
5.   Remove s from Q such that $C(s)$ is minimum.
6.   **For each** $t \in \mathcal{A}(s)$, such that $C(t) > C(s)$, **do**
7.   Compute $tmp \leftarrow f(P^*(s) \cdot \langle s, t \rangle)$.
8.   **If** $tmp < C(t)$, **then**
9.   **If** $C(t) \neq +\infty$, **then** remove t from Q.
10.   Set $P(t) \leftarrow s$, $R(t) \leftarrow R(s)$.
11. $C(t) \leftarrow tmp$, $L(t) \leftarrow \lambda(s)$
12.   Insert t in Q.

Lines 1–3 initialize maps for trivial paths. The minima of the initial map C compete with each other and some of them become roots of the forest. They are pixels with optimum trivial-path values, which are inserted in queue Q. The main loop computes an optimum path $P^*(s)$ from the roots to every node s in a non-decreasing order of value (Lines 4–12). At each iteration, a path $P^*(s)$ of minimum value $C(s)$ is obtained in P when we remove its last pixel s from Q (Line 5). Ties are broken in Q using first-in-first-out policy. The remaining lines evaluate if the path $P^*(s) \cdot \langle s, t \rangle$ that reaches an adjacent pixel t through s is cheaper than the current path with terminus t and update Q, $C(t)$, $R(t)$, and $P(t)$ accordingly.

III. Optimizations for fast classification

In the supervised OPF, the graph $(\mathcal{Z}_1, \mathcal{A})$ defined in the previous section is modeled as a complete graph, whose any pair of samples defines an arc in $\mathcal{A} = \mathcal{Z}_1 \times \mathcal{Z}_1$. One can note that the arcs do not need to be stored and so the graph does not need to be explicitly represented. As aforementioned, any OPF-based classifier tries to partition the graph into optimum-path trees according to some optimality criteria. This version uses the f_{max} as connectivity function, given by:

$$
\begin{aligned}
f_{max}(\langle s \rangle) &= \begin{cases} 0 & \text{if } s \in S \\ +\infty & \text{otherwise,} \end{cases} \\
f_{max}(\pi \cdot \langle s, t \rangle) &= \max\{f_{max}(\pi), d(s,t)\}, \quad (1)
\end{aligned}
$$

such that $f_{max}(\pi)$ computes the maximum distance between adjacent samples in π, when π is not a trivial path.

The supervised OPF training phase consists, essentially, into finding the prototypes and to execute the algorithm to generate the optimum-path forest. The prototypes are chosen as been the elements close to the boundary of the classes. More precisely, we are interested in the nearest elements from different classes. For that, we compute the Minimum Spanning Tree (MST) on the graph $(\mathcal{Z}_1, \mathcal{A})$ and we select the connected nodes with different labels [2]. The classification phase is done by taking one sample from $\mathcal{Z}_2$ and connecting it to all nodes from $\mathcal{Z}_1$ and to evaluate the node $s \in \mathcal{Z}_1$ that offers the minimum cost according to f_{max}, say that s^*. Hence, this procedure takes $O(|\mathcal{Z}_1||\mathcal{Z}_2|)$ for the whole test set.

In this paper, we explore a theoretical property of OPF that makes its classification phase faster. As aforementioned, the OPF tries to find the sample s^* that offers the minimum cost to a given sample $t \in \mathcal{Z}_2$. Thus, it is reasonable to assume that the sample s^* will be the one with low cost (not necessarily with the lowest one). In such a way, if we have an ordered set of training nodes sorted by increasing value of their costs, we could make the classification phase faster, because the probability of one of the first nodes to be s^* would be very high. The main problem now is to obtain this ordered set without compromising the computational complexity of OPF. Thus, this set can be obtained in the OPF training phase. As we can see in Line 5 of Algorithm 1, the OPF algorithm removes from queue Q the node with minimum cost at that moment. Hence, we can take this sample and store it in a set $\mathcal{T}$ with complexity $\theta(1)$.

The next step, after training, is to execute the test phase, by evaluating each node $k_i \in \mathcal{T}$, $i = 1, 2, 3 \dots, |\mathcal{T}|$. Although the reader notes that, in atypical cases (unbalanced classes with elongated shapes, for instance), the complexity of this new algorithm will be the same as the traditional one, we would like to stress that, in the average case, the complexity can be done by $O(|\mathcal{Z}_1|(p+1))$, i.e., $O(|\mathcal{Z}_1|p)$, in which $p \in o(|\mathcal{Z}_2|)$ is the number of samples evaluated in $\mathcal{T}$ before s^*. Algorithm 2 presents this new approach.

The main loop (Lines 1–10) performs the classification phase. For each sample $t \in \mathcal{Z}_2$, we begin evaluating the first sample from the ordered set $\mathcal{T}$, i.e., k_1 (Lines 2–4). The inner loop in Lines 5–10 evaluates the remaining samples in $\mathcal{T}$. If the cost of sample t is greater than the cost of the next sample in $\mathcal{T}$, i.e., k_{i+1} (Line 5), there exists a possibility of k_{i+1} to offer a better cost to t. In the affirmative case, the variable $mincost$ and the label of t are updated in Lines 7–9. Otherwise, we increment index i in Line 10.

Algorithm 2: – Proposed OPF Classification Algorithm

Input:	Ordered set $\mathcal{T}$ of samples and their respective labels assigned by function λ, connectivity cost map C, and test set $\mathcal{Z}_2$.
Output:	Labeled test set $\mathcal{Z}_2$.
Auxiliary:	Variables tmp and $mincost$.

1. **For each** $t \in \mathcal{Z}_2$, **do**
2. $i \leftarrow 1$.
3. $mincost \leftarrow \max\{C(k_i), d(k_i, t)\}$.
4. $L(t) \leftarrow \lambda(k_i)$.
5.   **While** $i < |\mathcal{Z}_1|$ and $C(t) > C(k_{i+1})$, **do**
6. $tmp \leftarrow \max\{C(k_{i+1}, d(k_{i+1}, t))\}$.
7.   **If** $tmp < mincost$, **then**
8. $mincost \leftarrow tmp$.
9. $L(t) \leftarrow \lambda(k_{i+1})$.
10. $i \leftarrow i + 1$

IV. Experiments and Results

In this section we evaluate our proposed approach to enhance the OPF algorithm classification step, and we validate it for the automatic white matter and gray matter (WM/GM) recognition in magnetic resonance images of the brain. The experiments involved the Internet Brain Segmentation Repository (IBSR) [1], which is composed of 18 T1 weighted images with 1.5mm slice thickness, but in this paper we used a subset containing 5 images. In this dataset, each voxel is considered as a sample for training and classification, and each image contains more than 1 million of voxels. The feature vector of each voxel is composed of its intensity value and the five intensities around the median value of the voxels within its 26-neighborhood.

Regarding classifiers, we compare EOPF against the traditional OPF[2] and Support Vector Machines [3] with Radial Basis Functions. We used the same training and test sets for all classifiers, and the results show the average accuracy under 10 runnings with randomly generated sets. Table I displays the recognition rates and the training and classification times in seconds at the up and the bottom of the table

---

[1]URL: www.cma.mgh.harvard.edu/ibsr/
[2]URL: www.ic.unicamp.br/ afalcao/libopf/
[3]URL: www.csie.ntu.edu.tw/ cjlin/libsvm/

cells, respectively. We used 0.01% of the whole dataset for training and the remaining 99.99% for classification.

Table I
AUTOMATIC WM/GM RECOGNITION RESULTS FOR EOPF, OPF AND SVM-RBF.

Image #	EOPF	OPF	SVM-RBF
1	89.04% $(0.001 - 7.84)$	89.07% $(0.002 - 13.33)$	88.29% $(1.91 - 9.12)$
2	87.00% $(0.001 - 6.74)$	87.20% $(0.002 - 13.93)$	90.55% $(2.53 - 10.87)$
3	87.44% $(0.001 - 4.74)$	87.47% $(0.002 - 9.07)$	89.82% $(1.68 - 9.73)$
4	86.82% $(0.001 - 6.29)$	86.82% $(0.001 - 11.89)$	88.00% $(1.72 - 9.76)$
5	84.65% $(0.001 - 7.57)$	85.01% $(0.001 - 10.04)$	83.82% $(2.48 - 10.30)$

The experimental results show that the accuracy of EOPF, OPF and SVM-RBF are similar for all images, but the former are 1.75 times and 1.5 times faster than OPF and SVM for classification, respectively. Regarding training, the EOPF are 3628 times faster than SVM-RBF. Notice that we used only 0.01% for training due to the expensive SVM-RBF computational time for training patterns. We can also see that SVM-RBF is faster than traditional OPF in all experiments for classification, but always slower than EOPF.

V. CONCLUSIONS

We propose a new algorithm called EOPF that explores a theoretical property of traditional OPF to speed up its classification time. Experiments among EOPF, OPF and SVM-RBF for automatic WM/GM recognition in magnetic resonance images of the human brain demonstrated that the proposed algorithm can speed up traditional OPF and SVM-RBF classifiers, and also to maintain its accuracy recognition results. We can also see that EOPF did not affect the traditional OPF training phase computational time, which is much faster than SVM-RBF[4].

REFERENCES

[1] L. Reyzin and R. Schapire, "How boosting the margin can also boost classifier complexity," in *Proceedings of the 23th ICML*, 2006, pp. 753–760.

[2] J. P. Papa, A. X. Falcão, and C. T. N. Suzuki, "Supervised pattern classification based on optimum-path forest," *Intl. J. of Imaging Systems and Tech.*, vol. 19, no. 2, pp. 120–131, 2009.

[3] L. M. Rocha, F. A. M. Cappabianco, and A. X. Falcão, "Data clustering as an optimum-path forest problem with applications in image analysis," *Intl. J. of Imaging Systems and Tech.*, vol. 19, no. 2, pp. 50–68, 2009.

[4] J. P. Papa, V. H. C. de Albuquerque, A. X. Falcão, and J. M. R. S. Tavares, "Fast automatic microstructural segmentation of ferrous alloy samples using optimum-path forest," in *CompIM-AGE 2010*, vol. 6026, 2010, pp. 210–220.

[5] A. X. Falcão, J. Stolfi, and R. A. Lotufo, "The image foresting transform theory, algorithms, and applications," *IEEE Transactions on PAMI*, vol. 26, no. 1, pp. 19–29, 2004.