Improving Optimum-Path Forest Classification Using Confidence Measures

Conference Paper · November 2015

DOI: 10.1007/978-3-319-25751-8_74

4 authors, including:

David Cox
Harvard University
56 PUBLICATIONS **3,436** CITATIONS
SEE PROFILE

João Paulo Papa
São Paulo State University
308 PUBLICATIONS **3,181** CITATIONS
SEE PROFILE

Improving Optimum-Path Forest Classification Using Confidence Measures

Silas E. N. Fernandes[1],
Walter Scheirer[2], David D. Cox[2], and
João Paulo Papa[3]

[1] Department of Computing, Federal University of São Carlos - UFSCar,
Rodovia Washington Luís, Km 235 - SP 310, São Carlos - SP, 13565-905, Brazil
silasfernandes@ieee.org,
[2] Department of Molecular & Cellular Biology, Harvard University,
52 Oxford St., Cambridge-MA, USA 02138
[3] Department of Computing, Univ Estadual Paulista - UNESP,
Av. Eng. Luiz Edmundo Carrijo Coube, 14-01, Bauru-SP, 17033-360, Brazil
papa@fc.unesp.br

Abstract. Machine learning techniques have been actively pursued in the last years, mainly due to the great number of applications that make use of some sort of intelligent mechanism for decision-making processes. In this work, we presented an improved version of the Optimum-Path Forest classifier, which learns a score-based confidence level for each training sample in order to turn the classification process "smarter", i.e., more reliable. Experimental results over 20 benchmarking datasets have showed the effectiveness and efficiency of the proposed approach for classification problems, which can obtain more accurate results, even on smaller training sets.

Keywords: Optimum-Path Forest, Supervised learning, Confidence measures

1 Introduction

Pattern recognition techniques aim at learning decision functions that separate a dataset in clusters of samples that share similar properties. Supervised techniques are known to be the most accurate, since the amount of information available about the training samples allows them to learn class-specific properties, as well as one can design more complex learning algorithms to improve the quality of the training data. The reader can refer to some state-of-the-art supervised techniques, such as Support Vector Machines (SVMs) [4], Artificial Neural Networks (ANNs) [8], Bayesian classifiers, and the well-known k-nearest neighbours (k-NN), among others. The reader can refer to Duda et al. [6] for a wide discussion about such methods.

Although we have very sophisticated and complex techniques, it is always important to keep an open mind for different approaches that may lead us to

better results. Simple ideas can improve the effectiveness of some well-known techniques. Ahmadlou and Adeli [1], for instance, proposed the Enhanced Probabilistic Neural Networks, being the idea to avoid the influence of noisy samples when computing the covariance matrix of each class. This simple idea has shown to be very effective in some situations. Later on, Guo et al. [7] presented a simple heuristic to reduce SVM computational load while maintaining its good generalization over unseen data. Their approach is based on the computation of the lowest margin instances, which are than used as support vector candidates.

Some years ago, Papa et al. [11, 10] presented a graph-based supervised pattern recognition technique called Optimum-Path Forest (OPF), which has demonstrated interesting results in terms of efficiency and effectiveness, being some of them comparable to the ones obtained by SVMs, but faster for training. The idea of OPF is to model the pattern recognition task as a graph partition problem, in which a set of key samples (*prototypes*) acts as being the rulers of this competition process. Such samples try to conquer the remaining ones offering to them optimum-path costs: when a sample is conquered, it receives the label of its conqueror. An interesting property stated by Souza et al. [12] concerns with OPF error bounds, which are the same as k-NN when all training samples are prototypes and a path-cost function that computes the maximum arc-weight along a path is employed. Such statement is very interesting, since a recent work by Amancio et al. [3] showed strong evidences that, in practice, k-NN may perform so well as SVMs.

The approach proposed by Papa et al. [11, 10] elects the prototype nodes as being the nearest samples from different classes, which can be found out through a Minimum Spanning Tree (MST) computation over the training graph: the connected samples in the MST are marked as being the prototype nodes. In case of multiple MSTs in large datasets, the current OPF implementation, although the values of the possible optimum-paths that are going to be offered for a given graph node may be the same from samples from different classes, the one which reaches that node first will conquer it. The main problem concerns with the "tie-regions", i.e., the regions in which we have a set of training samples that offer the same optimum-path cost to a given node. Therefore, this scenario may lead OPF to be more prone to errors in the training set.

In this paper, we propose to consider not only the optimum-path value from a given sample in the classification process, but also its *confidence value*, which is measured by means of a score index computed through a learning process over a validating set. The idea is to penalize the training samples that do not have "reliable" confidence values. We have shown this approach can overcome traditional OPF in several datasets, even when we learn on smaller training sets, as well as it can perform training faster than its na "ive version when using the same amount of data.

The remainder of the paper is organized as follows. Sections 2 and 3 present the OPF background theory and the proposed approach for score-based confidence computation, respectively. Section 4 describes the methodology and the

experimental results. Finally, conclusions and future works are stated in Section 5.

2 Optimum-Path Forest

Let $\mathcal{D}(\mathcal{X}, \mathcal{Y})$ be a dataset, in which $\mathcal{X}$ and $\mathcal{Y}$ stand for the set of samples (feature vectors) and the set of their labels, respectively. The OPF classifier models D as being a weighted graph $G(\mathcal{V}, \mathcal{A}, d)$, such that the set of samples are now the graph nodes, i.e., $\mathcal{V} = \mathcal{X}$, and the arcs are defined by the adjacency relation $\mathcal{A}$. In addition, the arcs are weighted by a distance function $d : \mathcal{V} \times \mathcal{V} \to \Re^+$.

Similarly to the process of ordered communities generation, in which group of individuals are originated based on the connectivity relations among their leaders, the OPF classifier employs a competition process among some key samples in order to partition the graph into optimum-path trees (OPTs) according to a predefined path-cost function. Analogously, the population is partitioned into communities, where each individual belongs to a group that has offered him the best reward.

Besides, the dataset $\mathcal{D}$ can be partitioned in two or three subsets according to the set of possible approaches. In the situation we need two subsets, we have that $\mathcal{D} = \mathcal{D}_1 \cup \mathcal{D}_2$, in which $\mathcal{D}_1$ and $\mathcal{D}_2$ stand for the training and testing sets, respectively. Therefore, the graph-based formulations of the training and testing sets are given by $G_1(\mathcal{V}_1, \mathcal{A}_1, d)$ and $G_2(\mathcal{V}_2, \mathcal{A}_2, d)$, respectively. However, without loss of generality, OPF usually uses the same adjacency relation for both sets. Thus, we can redefine both graphs as $G_1(\mathcal{V}_1, \mathcal{A}, d)$ and $G_2(\mathcal{V}_2, \mathcal{A}, d)$. Notice the standard OPF classifier uses a complete graph, which means all pairs of nodes are connected.

Let π_s be a path in the graph $\mathcal{G}_1$ with terminus in the sample $s \in \mathcal{D}_1$, and $(\pi_s \cdot \langle s, t \rangle)$ be the concatenation between π_s and the arc $\langle s, t \rangle$, such that $t \in \mathcal{D}_1$. Let $\mathcal{S} \subseteq \mathcal{V}_1$ be the set of prototype nodes from all classes. Roughly speaking, the idea of OPF is to minimize $f(\pi_t)$, $\forall t \in \mathcal{D}_1$, where $f(\cdot)$ is defined as the path-cost function given by:

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

in which $d(s, t)$ denotes the distance between nodes s and t. Particularly, an optimal set of prototypes $\mathcal{S}^*$ can be found exploiting the theoretical relation between the MST and the minimum spanning forest generated by OPF using $f(\cdot)$, as stated by Alléne et al. [2]. By computing an MST in G_1, we obtain an acyclic graph whose nodes are the samples in $\mathcal{D}_1$ and the arcs are non-directed and also weighted by the distance function d. Besides that, every pair of nodes in the MST is connected by a simple path, which is optimum with respect to $f(\cdot)$. In addition, this minimum spanning tree encodes an optimum-path tree for

each root (prototype) node. Thus, the optimum prototypes are defined as the nearest elements in the MST with different labels in $\mathcal{D}_1$.

In the classification phase, for each sample $r \in \mathcal{D}_2$, we consider all arcs connecting r to every $s \in \mathcal{D}_1$. If we take into account all possible paths from $\mathcal{S}^*$ to r, we can find the optimum path π_r^*, i.e., the one that minimizes $f(r)$ as follows:

$$f(r) = \min_{\forall s \in \mathcal{D}_1} \{\max\{f(s), d(s,r)\}\}. \tag{2}$$

Let $s^* \in \mathcal{D}_1$ be the sample that satisfies Equation 2. The OPF classification step simply assigns the label of s^* as being that of r.

3 Learning Score-based Confidence Levels

The classification using the confidence level supports the idea of assigning a score to all training nodes by means of a learning process over a validation set. In order to extract the confidence level, we need to partition the dataset $\mathcal{D}$ in three subsets, say that $\mathcal{D} = \mathcal{D}_1 \cup \mathcal{D}_v \cup \mathcal{D}_2$, in which $\mathcal{D}_1$, $\mathcal{D}_v$ and $\mathcal{D}_2$ stand for the training, validation and testing sets, respectively.

The proposed approach for learning scores aims at training OPF classifier over $\mathcal{D}_1$ for further classification of $\mathcal{D}_v$, using the same methodology described in Section 2. The main difference now is that we associate to each training sample a *reliability* level $\phi(\cdot)$, which is computed by means of its individual performance in terms of its recognition rate over the validation set. However, considering the aforementioned approach, a sample $t \in \mathcal{D}_1$ that did not participate from any classification process, would be scored as $\phi(t) = 0$, and may be penalized, since the higher the score the most reliable that sample is. Therefore, for such samples we have set $\phi(t) \to 1$ to give them a chance to perform a good job during the classification over the unseen (test) data. Thus, at the and of the classification process over the validation set $\mathcal{D}_v$, we have a score measure $\phi(s) \in [0, 1], \forall s \in \mathcal{D}_1$, which can be used as a *confidence level* of that sample. In short, there are three possible confidence levels:

- $\phi(s) = 0$: it means sample s did not perform a good work on classifying samples, since it has misclassified all samples. Therefore, samples with score equals to 0 *may not be reliable*;
- $0 < \phi(s) < 1$: it means sample s has misclassified samples, as well as it has also assigned correct labels to some of them. Notice the larger the errors, the lower is a sample's reliability. Samples with scores that fall in this range, *may be reliable*; and
- $\phi(s) = 1$: it means either sample s did not participate in any classification process, or s assigned the correct label to all its conquered samples, which means s is a *reliable sample* according to our definition.

After learning the confidence levels for each training sample, one needs to modify the naïve OPF classification procedure in order to consider this information during the label assignment. In order to fulfill this purpose, we proposed a modification in the OPF classification procedure (Equation 2) as follows:

$$f(r) = \min_{\forall s \in \mathcal{D}_1} \left\{ \left(\frac{1}{\phi(s) + \epsilon} \right) * \max\{f(s), d(s,r)\} \right\}, \tag{3}$$

where $\epsilon = 10^{-4}$ is employed to avoid numerical instabilities. Therefore, the idea of the first term in the above equation is to penalize samples with *low confidence* values by increasing their costs. In short, the amount of penalty is inversely proportional to a sample's confidence level.

4 Methodology and Experimental Results

In order to evaluate the efficiency and effectiveness of the proposed confidence-based approach for OPF classifier, we perform experiments over 20 classification datasets (real and synthetic datasets)[4][5][6][7]. Due to the lack of space, instead of showing characteristics individually for these datasets, we append in Table 1 which also presents the mean accuracies. The choice of these datasets was motivated by their level of complexity (overlapped samples), which turns the classification process more sensible to misclassification. The experiments were conducted on a computer with a Pentium Intel Core $i3^{®}$ 3.07Ghz processor, 4 GB of memory RAM and Linux Ubuntu Desktop LTS 12.04 as the operational system.

For each dataset, we conducted a cross-validation procedure with 15 runnings, being each of them partitioned as follows: 30% of the samples were used to compose the training set, being the validation and testing sets ranged from $10\% - 60\%, 20\% - 50\%, \ldots, 50\% - 20\%$. These percentages have been empirically chosen, being more intuitive to provide a larger validation set for *confidence learning*.

In Table 1 is included average accuracy over all datasets. In order to provide a robust analysis, we performed the non-parametric Friedman test, which is used to rank the algorithms for each dataset separately. In case of Friedman test provides meaningful results to reject the null-hypothesis (h_0: all techniques are equivalent), we can perform a post-hoc test further. For this purpose, we conducted the Nemenyi test, proposed by Nemenyi [9] and described by Demšar [5], which allows us to verify whether there is a critical difference (CD) among techniques or not. Due to the lack of space, instead of showing all diagrams for each dataset, we highlighted the best techniques in bold according to Nemenyi test.

We can observe OPFc has obtained the best results in 7 out 20 datasets, and with results very close to the best ones in other 7 datasets. The very worst

[4] http://mldata.org
[5] http://archive.ics.uci.edu/ml
[6] http://pages.bangor.ac.uk/ mas00a/activities/artificial_data.htm
[7] http://lrs.icg.tugraz.at/research/aflw

Table 1. Mean accuracy results: the bold values stand for the most accurate techniques. The recognition rates were computed according to [11], which consider unbalanced datasets.

Dataset	OPF	OPF*	OPFc	# samples	# features	# classes
a1a	65.74	65.59	**69.05**	32,561	123	2
aloi	95.31	**96.92**	95.09	108,000	128	1,000
connect-4	**63.32**	63.05	63.10	67,557	126	3
synthetic1	50.69	**50.78**	50.72	100,000	100	1,000
synthetic2	85.29	85.56	**87.33**	100,000	4	4
synthetic3	89.55	89.70	**91.14**	100,000	4	4
synthetic4	53.05	52.44	**56.14**	500	2	2
dmoz-web-directory-topics	59.16	**62.06**	56.72	1,329	10,629	5
dna	83.80	**88.99**	85.02	5,186	180	3
duke-breast-cancer	80.37	**91.15**	79.46	86	7,129	2
ijcnn1	93.78	**96.46**	94.13	191,681	22	2
Statlog-Letter	97.31	**98.58**	97.58	35,000	16	26
Leukemia	71.47	**76.90**	69.63	72	7,129	2
mushrooms	93.68	92.61	**96.93**	8,124	112	2
scene-classification	66.04	**67.78**	66.60	2,407	294	15
shuttle	94.48	**97.25**	95.09	101,500	9	7
usps	97.24	**97.93**	97.28	9,298	256	10
w1a	80.54	80.15	**80.68**	49,749	300	4
yahoo-web-directory-topics	50.54	51.77	**56.36**	1,106	10,629	4
aflw	88.00	**89.48**	88.93	8,193	4,096	2

results were obtained over "duke-breast-cancer" and "Leukemia", since these are small datasets, thus providing a validation set that was not enough to learn good confidence levels. However, even in these datasets, OPFc recognition rate was close to standard OPF one. As OPF* has employed bigger datasets, it was expected more accurate results.

It was not possible to establish some specific situation (considering the dataset configuration, such as the number of classes and the number features, for instance) in which OPFc might be better than OPF and OPF*, although it seems the proposed approach has obtained the top results in high-dimensional datasets, except for "dmoz-web-directory-topics". If we consider an error margin of around 3%, the proposed approach obtained similar results in 17 out 20 datasets, thus being considered a very suitable approach to improve OPF classifier.

The above assumption can be strengthened if we consider the computational effort of the techniques. As expected, standard OPF has been faster than OPFc and OPF* with respect to the training (training+learning scores) step, since it does not need to compute the confidence level for every training sample. However, the Nemenyi statistical test pointed out OPFc has been faster than OPF* for training (Figure 1a), being similar to it with respect to the classification step, as displayed in Figure 1b. On average, i.e., considering all 20 datasets, standard OPF has been about 2.108 times faster than OPFc and OPF*.

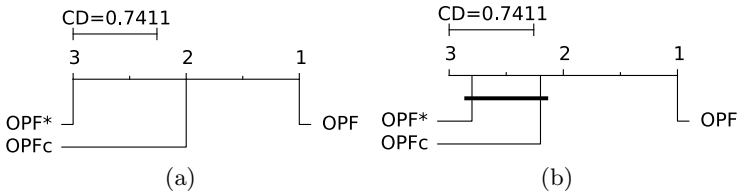

Fig. 1. Nemenyi statistical test regarding the (a) training (training + learning scores) and (b) testing computational load. Groups of similar approaches are connected to each other.

5 Conclusions and future works

In this work, we introduced a confidence-based learning algorithm to improve OPF classification results. The idea is to penalize training samples that misclassify others in a classification process over a validation set. The proposed algorithm aims at learning confidence levels for each training sample to be further used in a modified version of the standard classification procedure employed by OPF.

Experiments over 20 datasets showed the robustness of the proposed approach, which obtained the best results in 7 datasets, as well as very close recognition rates in other 7 datasets. Additionally, OPFc can improve standard OPF results even with smaller training sets, being also faster than OPF trained over training+validation sets.

Acknowledgments The authors would like to thank CAPES for their financial support, and FAPESP grants #2013/20387-7 and #2014/16250-9, as well as CNPq grants #47057162013-6, #303182/2011-3 and #306166/2014-3.

References

1. Ahmadlou, M., Adeli, H.: Enhanced probabilistic neural network with local decision circles: A robust classifier. Integrated Computer-Aided Engineering 17(3), 197–210 (2010)
2. Allène, C., Audibert, J.Y., Couprie, M., Keriven, R.: Some links between extremum spanning forests, watersheds and min-cuts. Image and Vision Computing 28(10), 1460–1471 (2010)
3. Amancio, D.R., Comin, C.H., Casanova, D., Travieso, G., Bruno, O.M., Rodrigues, F.A., Costa, L.F.: A systematic comparison of supervised classifiers. PLoS ONE 9(4), e94137 (2014)
4. Cortes, C., Vapnik, V.: Support-vector networks. Machine Learning 20(3), 273–297 (1995)
5. Demšar, J.: Statistical comparisons of classifiers over multiple data sets. J. Mach. Learn. Res. 7, 1–30 (Dec 2006), http://dl.acm.org/citation.cfm?id=1248547.1248548

6. Duda, R.O., Hart, P.E., Stork, D.G.: Pattern Classification (2nd Edition). Wiley-Interscience (2000)
7. Guo, L., Boukir, S.: Fast data selection for SVM training using ensemble margin. Pattern Recognition Letters 51(0), 112–119 (2015)
8. Haykin, S.: Neural Networks: A Comprehensive Foundation (3rd Edition). Prentice-Hall, Inc., Upper Saddle River, NJ, USA (2007)
9. Nemenyi, P.: Distribution-free Multiple Comparisons. Princeton University (1963)
10. Papa, J.P., Falcão, A.X., Albuquerque, V.H.C., Tavares, J.M.R.S.: Efficient supervised optimum-path forest classification for large datasets. Pattern Recognition 45(1), 512–520 (2012)
11. Papa, J.P., Falcão, A.X., Suzuki, C.T.N.: Supervised pattern classification based on optimum-path forest. International Journal of Imaging Systems and Technology 19(2), 120–131 (2009)
12. Souza, R., Rittner, L., Lotufo, R.A.: A comparison between k-optimum path forest and k-nearest neighbors supervised classifiers. Pattern Recognition Letters 39(0), 2–10 (2014), advances in Pattern Recognition and Computer Vision