Improving the Accuracy of the Optimum-Path Forest Supervised Classifier for Large Datasets

César Castelo-Fernández^{1,*}, Pedro J. de Rezende^{1,**},
Alexandre X. Falcão^{1,***}, and João Paulo Papa^{2,†}

¹ Institute of Computing, State University of Campinas - UNICAMP,
Campinas, Brazil,

`ccastelo@liv.ic.unicamp.br, {rezende, afalcao}@ic.unicamp.br`

² Department of Computing, São Paulo State University - UNESP,
Baurú, Brazil,
`papa@fc.unesp.br`

Abstract. In this work, a new approach for supervised pattern recognition is presented which improves the learning algorithm of the Optimum-Path Forest classifier (OPF), centered on detection and elimination of outliers in the training set. Identification of outliers is based on a penalty computed for each sample in the training set from the corresponding number of imputable false positive and false negative classification of samples. This approach enhances the accuracy of OPF while still gaining in classification time, at the expense of a slight increase in training time.

Keywords: Optimum-Path Forest Classifier, Outlier Detection, Supervised Classification, Learning Algorithm.

1 Introduction

Pattern recognition aims at the capacity of classifying a pattern based in its inherent characteristics, represented as a feature vector (i.e., a point in a higher dimensional space) [5]. Usually, this task is divided into two phases, namely, training and classification. During the training phase the classifier learns the distribution of the data in the feature space through a subset of the dataset, inferring rules which then allow for predicting the correct classes of unknown

* Supported by Capes – Coordenação de Aperfeiçoamento de Pessoal de Nível Superior – Grant # 01-P-04388/2010.

** Partially supported by CNPq – Conselho Nacional de Desenvolvimento Científico e Tecnológico – Grants # 472504/2007-0, 483177/2009-1 and FAPESP – Fundação de Amparo à Pesquisa do Estado de São Paulo – Grant #07/52015-0 and a Grant from FAEPEX/UNICAMP.

*** Partially supported by CNPq Grants # 481556/2009-5 (ARPIS), 302617/2007-8 and FAPESP Grant #2007/52015-0.

† Partially supported by CNPq Grant # 481556/2009-5 (ARPIS) and FAPESP Grant # 2009/16206-1.

data. Several methods also include a learning phase, based on two labeled sets with training and evaluating samples, which are usually selected at random, with the goal of improving the performance of the classifier. This phase employs the evaluating set as a pseudo-test of the quality of the training set and for its improvement.

Support Vector Machines (SVM) [9], a largely used classification method, is formulated as an optimization problem that seeks to determine the hyperplane which best splits the data. Also, given non-separable data, a prior mapping to a higher dimensional space is required, assuming that the mapped data becomes separable. SVM's main deficiency is that, depending on the size of the training set, too much computational time is needed before convergence to a solution occurs. This lack of efficiency for large datasets can render SVM infeasible in these cases. Furthermore, the assumption of class-separability in higher-dimensional space may not always hold [4].

The Optimum-Path Forest classifier (OPF) [8] is a graph-based technique which models classification problems as optimum-path searches in graphs derived from an adjacency relation between samples in a given feature space (a complete relation in the case of this paper). Class representatives (prototypes) are chosen among the training samples and used to classify the remaining samples based on lengths of paths on the graph. This method has as an advantage a very low computational training cost, since it does not have to optimize parameters. Furthermore, it can deal with non-separable data since its formulation is based on multiple prototypes that represent the various classes.

As showed in [8], OPF can attain an accuracy as high as SVM's, while keeping the training time much lower. Those results show that OPF is the best choice for several classification problems, mainly in the case of large datasets. However, a disadvantage of the original learning algorithm proposed in [8], is that it does not attempt to eliminate outliers in the training set which may cause classification errors.

The main contribution of the present work is the development of a learning algorithm for the OPF classifier focused on the detection and elimination of outliers from the training set. This paper describes the OPF classifier in Section 2, introduces the new method in Section 3, shows the experiments in Section 4 and Section 5 brings together the conclusions.

2 Optimum-Path Forest Classifier

For the sake of completion, the OPF classifier will be briefly described. For more details, see [8].

Let Z_1 , Z_2 and Z_3 be the disjoint training, evaluating and testing sets, randomly chosen from a dataset Z , such that $Z = Z_1 \cup Z_2 \cup Z_3$. Let $\lambda(s)$ be a function which maps each sample $s \in Z$ to its correct class $i, i = 1, \dots, c$. Let $\vec{v}(s)$ denote the feature vector of sample s , computed by some feature extractor v and $d(s, t)$ denote the distance between s and t (e.g. euclidean distance).

A specially chosen subset $S \subset Z_1$, whose elements are called *prototypes*, contains the samples responsible for classifying each $t \in Z \setminus S$. This classification

is accomplished by determining, for each t , which $s \in S$ is the most closely (strongly) connected to t and setting the class of t to be that of s .

The strategy of the OPF classifier is to first obtain an optimum discrete partition of the feature space of Z_1 , so that it becomes possible to classify every $t \in Z_2 \cup Z_3$ based on that partition, in an incremental way.

Let (Z_1, A) be the complete graph which represents the training set, where to each sample $s \in Z_1$ corresponds a node and such that to each edge $\langle s, t \rangle$ is assigned a cost equal to $d(s, t)$. Let $f(\pi_t^{s_1})$ be a function which evaluates the cost of the path $\pi_t^{s_1} = \langle s_1 = t_1, t_2, \dots, t_n = t \rangle$, starting at some $s_1 \in S$ and ending at $t \in Z_1$. A path π_t^s is considered optimum if, for any other path $\pi_t^{s'}$, $f(\pi_t^s) \leq f(\pi_t^{s'})$. The union of optimum-paths for all $t \in Z_1$ forms a forest (an acyclic graph).

The optimum path forest is computed using the Image Foresting Transform algorithm [6] by basically solving the following optimization problem:

$$C(t) = \min_{\forall s \in S} \{f_{max}(\pi_t^s)\} \quad (2.1)$$

which gives an optimum path-cost $C(t)$ for each $t \in Z_1$ of an optimum path, denoted π_t^* for short, which starts at some $s \in S$. The function $f_{max}(\pi_t^s)$ computes the maximum edge cost along the path π_t^s .

Algorithm 1 shows the general procedure to compute the Optimum-Path Forest [8].

Algorithm 1. – OPF-ALGORITHM

INPUT: Training set Z_1 , λ -labeled prototypes $S \subset Z_1$.

OUTPUT: Optimum-Path Forest P (predecessor map), path-cost map C and label map L and a list Z'_1 of the training nodes ordered by their path-cost.

AUXILIARY: Priority queue Q and cost variable cst .

```

1. For each  $s \in Z_1 \setminus S$  do  $C(s) \leftarrow +\infty$ .
2. For each  $s \in S$  do
3.    $C(s) \leftarrow 0$ ,  $P(s) \leftarrow nil$ 
4.    $L(s) \leftarrow \lambda(s)$ , insert  $s$  into  $Q$ .
5. While  $Q$  is not empty do
6.   Remove from  $Q$  a sample  $s$  with minimum  $C(s)$  and insert  $s$  in  $Z'_1$ .
7.   For each  $t \in Z_1$  such that  $C(t) > C(s)$  do
8.      $cst \leftarrow \max\{C(s), d(s, t)\}$ .
9.     If  $cst < C(t)$  then
10.      If  $C(t) \neq +\infty$  then remove  $t$  from  $Q$ .
11.       $P(t) \leftarrow s$ ,  $L(t) \leftarrow L(s)$  and  $C(t) \leftarrow cst$ .
12.      Insert  $t$  in  $Q$ .
```

The training process (i.e., finding the set S of prototypes) consists of computing a Minimum Spanning Tree (MST) on Z_1 and, for each edge $\{s, t\}$ of the MST, if $\lambda(s) \neq \lambda(t)$ then s and t are marked as prototypes.

In the classification phase, for every sample $t \in Z_3$ all possible paths from each $s \in Z_1$ to t are computed and the optimum one, π_t^* , is chosen. This path can easily be identified by incrementally evaluating the optimum cost $C(t)$ as follows:

$$C(t) = \min_{\forall s \in Z_1'} \{ \max \{ C(s), d(s, t) \} \}. \quad (2.2)$$

The (first) role of Z_1' is to speed up the evaluation of Equation 2.2 which can halt when $\max \{ C(s), d(s, t) \} < C(p)$, for a node p whose position in Z_1' succeeds the position of s [7].

The learning phase consists of performing, after the training process, the classification of each sample in Z_2 through the method just described and, subsequently, swapping each incorrectly classified sample from Z_2 for a randomly chosen sample from $Z_1 \setminus S$.

3 Optimum-Path Forest Classifier with Outliers Detection: OPF-OD

Outliers in the training set might be identified through two kinds of errors: when a sample is attributed an incorrect label (false positive - FP) and, when a classifying sample does not identify a sample of its own class (false negative - FN). Samples that cause such errors negatively impact the accuracy in the classification phase unless they are detected during the learning phase (through incorrect classifications) and properly dealt with, as will be described later.

Additionally, it becomes necessary to compute the number of correct classifications (true positives - TP) and of correct rejections (true negatives - TN) to be contrasted with the false positives and false negatives in order to deduce the usefulness of a sample in the training set.

To understand this bookkeeping, let $s \in Z_1'$ be the sample in Equation 2.2 which classifies a sample $t \in Z_2$.

- The false positive counter for s , FP_s , is incremented if $L(t) \neq \lambda(t)$, otherwise, the true positive counter for s , TP_s , is incremented.

Now, proceeding upwards (i.e., as $C(\cdot)$ decreases) on Z_1' , let p_1 and p_2 be the first samples found such that $L(p_1) = \lambda(t)$ and $L(p_2) \neq \lambda(t)$.

- The false negative counter for p_1 , FN_{p_1} , and the true negative counter for p_2 , TN_{p_2} , are incremented.

After computing FP, TP, FN and TN for all samples in Z_1 , a penalty pen_s is calculated for each $s \in Z_1$ so that a sample s can be considered an outlier whenever $pen_s > \varepsilon$, for some threshold $0 < \varepsilon < 1$.

Let

$$E_{s+} = \frac{FP_s}{TP_s} \quad E_{s-} = \frac{FN_s}{TN_s},$$

where E_{s+} and E_{s-} are defined as *acceptance* and *rejection rates* for s , which are then normalized in ξ_{s+} and ξ_{s-} , respectively, by their maximum possible values.

Define pen_s as follows:

$$pen_s = \mu \cdot (\xi_{s+}) + (1 - \mu) \cdot (\xi_{s-}) \quad (3.3)$$

where $0 \leq \mu \leq 1$, and hence, $0 \leq pen_s \leq 1$.

In other words, this penalty corresponds to the convex combination of the normalized acceptance and rejection rates, weighted by a parameter μ which allows for a user-controlled balance between FP and FN, that can be application dependent.

Once the penalties are computed for all samples in Z_1 , any sample s identified as an outlier (i.e., $pen_s > \varepsilon$) is swapped for a random sample from Z_2 . However, care is taken to prevent s from being swapped back into Z_1 , if it happens to be misclassified in Z_2 during the learning phase (see Section 2). This will guarantee that previously identified outliers are immovable from Z_2 so as not to negatively impact the accuracy of the classification of Z_3 .

Algorithm 2 explains the proposed approach.

Algorithm 2. – OPF-LEARNING-WITH-OUTLIERS-DETECTION

INPUT: λ -labeled training and evaluating sets Z_1 and Z_2 , number T of iterations, and threshold ε .

OUTPUT: Instance of the OPF-OD classifier with best accuracy over Z_2 .

AUXILIARY: List LM of misclassified samples and Acc .

```

1. For each iteration  $i \leftarrow 1, 2, \dots, T$  do
2.   Compute  $S$  from  $Z_1$ .
3.    $(P, C, L, Z'_1) \leftarrow \text{Algorithm-1}(Z_1, S)$ .
4.   For each sample  $t \in Z_2$  do  $\triangleright \triangleright \triangleright$  classify all samples in  $Z_2$ 
5.     Use the classifier computed in Line 3 to classify  $t$  with label  $L(t)$ .
6.     If  $\lambda(t) \neq L(t)$  then  $LM \leftarrow LM \cup t$ .  $\triangleright \triangleright \triangleright$  save misclassified samples
7.     Update FP, TP, FN, TN for the corresponding samples (Section 3).
8.   For each sample  $s \in Z_1$  do  $\triangleright \triangleright \triangleright$  compute all penalties
9.     Compute  $pen_s$  using Equation 3.3
10.    If  $pen_s > \varepsilon$  then Swap  $s$  for a sample from  $Z_2$ .
11.  Compute accuracy  $Acc$  and save the actual instance of the classifier if
12.  its accuracy is maximum so far.
13.   $k \leftarrow |Z'_1|$ .
14.  While  $LM \neq \emptyset$  and  $k > 0$  do  $\triangleright \triangleright \triangleright$  swap misclassified samples
15.    Choose a random sample  $t$  from  $LM$ ,  $LM \leftarrow LM \setminus t$ 
16.    While  $Z'_1[k]$  is a prototype and  $k > 0$  do  $k \leftarrow k - 1$ .
17.    If  $k > 0$  then Swap  $t$  for sample  $Z'_1[k]$ .
18.     $k \leftarrow k - 1$ 
19. Return instance of the classifier with highest accuracy.

```

The samples to be swapped from Z_1 in Line 10 are not randomly chosen as in [8], but in decreasing order of path-cost using Z'_1 , because samples with higher path-cost have less probability of correctly classifying samples from Z_2 . Therefore, this gives a third important role for the ordered list Z'_1 .

4 Experiments

In this section, the datasets used in the tests, the computational environment, the classifiers employed and the overall approach of the experiments are described.

4.1 Datasets Used

To validate the proposed approach, the following datasets from the UC Irvine Repository [3] were used:

- The “Adult” dataset, with about 45000 samples, 14 attributes and 2 classes.
- The “King-Rook vs. King” dataset (krk), with about 30000 samples, 6 attributes and 18 different classes.
- The “Gamma telescope” dataset, with about 20000 samples, 11 attributes and 2 possible classes.

Also, the “Mpeg 7” dataset [1] was used, which consists of about 1400 samples, 70 classes and whose features were extracted using a shape descriptor called Bean Angle Statistics (BAS) [2]. The distance between feature vectors was measured using euclidean distance and optimal correspondence subsequence (OCS) which is more appropriate for the BAS descriptor.

The classifiers used were the OPF with the traditional learning algorithm and the proposed OPF-OD method which refines OPF.

The accuracy measure used to evaluate the classifiers is computed as follows [8]:

$$e_{i,1} = \frac{FP(i)}{|Z_3| - |Z_3(i)|} \quad e_{i,2} = \frac{FN(i)}{|Z_3(i)|}, i = 1, \dots, c$$

$$E(i) = e_{i,1} + e_{i,2} \quad Acc = 1 - \frac{\sum_{i=1}^c E(i)}{2c}$$

where $FP(i)$ and $FN(i)$ are the number of false positives and false negatives for class i and $Z_3(i)$ is the number of samples from class i which belong to Z_3 .

4.2 Sizes of Z_1 and Z_2

Due to the fact that OPF-OD is based on a learning phase, the evaluating set Z_2 must be significantly larger than the training set Z_1 to allow for the computation of meaningful penalties for each sample in Z_1 . To ensure a good trade-off between accuracy and learning time, the most beneficial ratio $|Z_2|/|Z_1|$ was experimentally determined to be 5.

4.3 Results

In this section, comparisons between OPF and OPF-OD are presented. All the experiments reported here were run on an IntelTM XeonTM 4-core 2.50 GHz processor, with 8 GB of RAM, under Linux but *without* multi-thread execution.

Firstly, the Adult dataset was used to test 10 different sizes for the training set Z_1 averaged over 10 executions for each size. Figure 1 presents the accuracy obtained, showing the average value and the standard deviation for each size. Note that in *all* cases, OPF-OD obtained higher accuracy than standard OPF.

Fig. 1. Accuracy for OPF and OPF-OD calculated over the classification of Z_3 using different sizes for Z_1 . Average of ten runs and standard deviation are shown. Here $|Z_2|/|Z_1| = 5$.

Considering the extra time required for the learning phase, the training time for OPF-OD was, as expected, higher than for OPF, as shown in Figure 2.(a) for the Adult dataset.

However, swapping out the outliers from Z_1 leads to some gain, time-wise, in the classification phase, as it can be seen in Figure 2.(b). This is due to the faster convergence of the optimization process that searches, for each sample in Z_3 , for its closest sample from Z_1 , as per Equation 2.2, since fewer samples have to be tested when outliers are no longer present.

Fig. 2. (a) Training time, and (b) Classification time in seconds for OPF and OPF-OD using different sizes for Z_1 . Average of ten runs is shown. Here $|Z_2|/|Z_1| = 5$.

Finally, Table 1 shows the accuracy and time results for the other datasets. Two different sizes for Z_1 (depending on the dataset) were considered. The accuracy of OPF-OD for these datasets was higher than OPF's, while OPF-OD's performance was slightly worse in the training phase and better in the classification phase.

Table 1. Accuracy (mean $\pm$ standard deviation), Training time (mean) and Classification time (mean) in seconds for several datasets. Best values are in bold.

Datasets	$ Z_1 $	$ Z_2 $	$ Z_3 $	Accuracy		Training time		Classification time	
				OPF	OPF-OD	OPF	OPF-OD	OPF	OPF-OD
krk	280	1400	26320	58.66 ± 0.64	58.90 ± 0.90	0.155	0.328	0.468	0.460
krk	2800	14000	11200	69.24 ± 0.47	67.68 ± 0.54	26.773	29.367	1.976	1.915
gamma	190	950	17860	63.66 ± 1.24	67.36 ± 0.74	0.115	0.190	0.261	0.228
gamma	1900	9500	7600	67.24 ± 0.72	70.74 ± 0.75	11.589	15.997	1.109	0.982
mpeg7	70	350	980	68.09 ± 1.76	71.53 ± 1.02	0.039	0.105	0.0194	0.0192
mpeg7	140	700	560	78.09 ± 1.49	79.68 ± 1.49	0.145	0.348	0.0223	0.0221
mpeg7-OCS	70	350	980	75.57 ± 0.82	78.40 ± 1.65	0.013	0.039	0.0007	0.0008
mpeg7-OCS	140	700	560	89.93 ± 0.66	90.92 ± 0.65	0.022	0.065	0.0010	0.0011

Note that the accuracy grows higher for all datasets as the training set becomes larger. For instance, with a training set of 140 samples for the mpeg7-OCS dataset, the accuracy was 11 percentage points higher than with half as many samples. Furthermore, it is critical to highlight that accuracy depends on other factors such as the distance function, as was the case of the mpeg7 dataset computed with euclidean distance vs. OCS distance.

5 Conclusions

In this work, a novel learning algorithm was presented that improves the time performance and accuracy of the supervised Optimum Path Forest classifier. This algorithm is essentially based on an ingenious method of detecting outliers in the training set and their subsequent swapping for new random samples from the evaluating set leading to a refined training set. As substantiation, notice that the attained improvement in accuracy was as high as 3.7 percentage points in the case of the Gamma dataset. This is mostly due to the resulting samples in the training set being better representatives of the classes present in the testing set than if outliers had remained in it, causing negative impact in accuracy. Despite the fact that the new learning algorithm performs extra work to identify outliers, the overall training time is only slightly increased. However, for most applications, this is well compensated by the lower classification time which results from the faster convergence of the search process through the elements of the training set when outliers have been swapped out. Hence, the forest computed by OPF-OD over the training set is more effective and efficient than OPF's. For instance, for the Adult dataset, the behavior of training time vs. classification time is illustrated in Figures 2.(a) and 2.(b).

References

1. MPEG-7: The generic multimedia content description standard, part 1. IEEE MultiMedia 9, 78–87 (2002)
2. Arica, N., Vural, F.T.Y.: BAS: a perceptual shape descriptor based on the beam angle statistics. Pattern Recognition Letters 24(9-10), 1627–1639 (2003)

3. Asuncion, A., Newman, D.J.: UCI machine learning repository (2007)
4. Boser, B.E., Guyon, I.M., Vapnik, V.N.: A training algorithm for optimal margin classifiers. In: Proceedings of the 5th Workshop on Computational Learning Theory, pp. 144–152. ACM Press, New York (1992)
5. Duda, R.O., Hart, P.E., Stork, D.G.: Pattern Classification, 2nd edn. Wiley-Interscience, Hoboken (2000)
6. Falcão, A.X., Stolfi, J., Lotufo, R.A.: The image foresting transform: Theory, algorithms, and applications. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 26(1), 19–29 (2004)
7. Papa, J.P., Cappabianco, F.A.M., Falcão, A.X.: Optimizing optimum-path forest classification for huge datasets. In: Proceedings of The 20th International Conference on Pattern Recognition (to appear, 2010)
8. 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, 120–131 (2009)
9. Vapnik, V.N.: An overview of statistical learning theory. *IEEE Transactions on Neural Networks* 10(5), 988–999 (1999)