Pattern Recognition 60 (2016) 72–85

Contents lists available at ScienceDirect

Pattern Recognition

journal homepage: www.elsevier.com/locate/pr

Improving semi-supervised learning through optimum connectivity

Willian P. Amorim[a], Alexandre X. Falcão[b,*], João P. Papa[c], Marcelo H. Carvalho[a]

[a] FACOM - Institute of Computing, Federal University of Mato Grosso do Sul - UFMS, University City, Cidade Universitaria - Universitaria, CEP 79070-900 Campo Grande, MS, Brazil
[b] Department of Information Systems, Institute of Computing, University of Campinas, Av. Albert Einstein, 1251, CEP 13083-852 Campinas, SP, Brazil
[c] Department of Computing, São Paulo State University, Av. Eng. Luiz Edmundo Carrijo Coube, 14-01, CEP 17033-360 Bauru, SP, Brazil

ARTICLE INFO

Article history:
Received 6 July 2015
Received in revised form
2 February 2016
Accepted 28 April 2016
Available online 20 May 2016

Keywords:
Semi-supervised learning
Optimum-path forest classifiers

ABSTRACT

The annotation of large data sets by a classifier is a problem whose challenge increases as the number of labeled samples used to train the classifier reduces in comparison to the number of unlabeled samples. In this context, semi-supervised learning methods aim at discovering and labeling informative samples among the unlabeled ones, such that their addition to the correct class in the training set can improve classification performance. We present a semi-supervised learning approach that connects unlabeled and labeled samples as nodes of a minimum-spanning tree and partitions the tree into an optimum-path forest rooted at the labeled nodes. It is suitable when most samples from a same class are more closely connected through sequences of nearby samples than samples from distinct classes, which is usually the case in data sets with a reasonable relation between number of samples and feature space dimension. The proposed solution is validated by using several data sets and state-of-the-art methods as baselines.

1. Introduction

Data organization, prediction, and retrieval are crucial components in many applications, which can be greatly facilitated when data sets are fully annotated. In pattern recognition, it is assumed that a supervised classifier can annotate a data set, when an expert provides labeled samples from all classes. However, as the data sets grow in size, due to the advances in data acquisition and storage systems, the design of the pattern classifier becomes more sensitive to the limited size of the training set and the choice of its manually annotated samples. In this context, a question that naturally arises is: can we improve the effectiveness of the classifier by exploiting the larger amount of unlabeled data? Semi-supervised learning approaches have tried to answer this question.

Existing methods for semi-supervised learning either (a) propagate labels to unlabeled training samples, one by one, via supervised classification [1–4], or (b) explore the spatial distribution of labeled and unlabeled training samples in the feature space for label propagation [5,6]. In both categories, the label propagation process can repeat a few times and a final classifier is created from the completely labeled set (e.g., from its most confidently labeled samples).

In this paper, we propose a significant improvement of our previous approach, named OPFSEMI, for semi-supervised learning [5]. The method, named OPFSEMI$_{mst}$, also exploits optimum connectivity between labeled and unlabeled samples for label propagation, but it can design the final classifier from a single iteration of label propagation. OPFSEMI$_{mst}$ is considerably more efficient and more accurate than OPFSEMI, as we will demonstrate in this work.

We have adopted the Optimum-Path Forest (OPF) methodology for the design of unsupervised, supervised, and semi-supervised pattern classifiers. OPF was initially proposed for image processing [7], and subsequently extended to clustering [8] and supervised classification [9]. This methodology consists in choosing three main components: (a) the training samples, (b) some adjacency relation, and (c) a suitable connectivity function. The training samples can be labeled, unlabeled, or both, characterizing the supervised, unsupervised, and semi-supervised learning processes, respectively. The adjacency relation aims at linking training samples in the feature space as nodes of a graph, in order to explore optimum connectivity between them. The connectivity function defines a value to any sequence of distinct adjacent samples (simple path) in the graph, as well as to trivial paths formed by single nodes. Initially, every node defines one trivial path and the minimization of the connectivity map computes optimum paths with terminus at each node, such that (a) the roots of the paths are first derived from the minima of the connectivity map and (b) these roots conquer other nodes by offering to them optimum paths, thus partitioning the graph into an optimum-path forest (the classifier), which assigns labels to new samples by

[*] Corresponding author.
E-mail addresses: paraguassuec@gmail.com (W.P. Amorim), afalcao@ic.unicamp.br (A.X. Falcão), papa@fc.unesp.br (J.P. Papa), mhc@facom.ufms.br (M.H. Carvalho).

http://dx.doi.org/10.1016/j.patcog.2016.04.020
0031-3203/© 2016 Elsevier Ltd. All rights reserved.

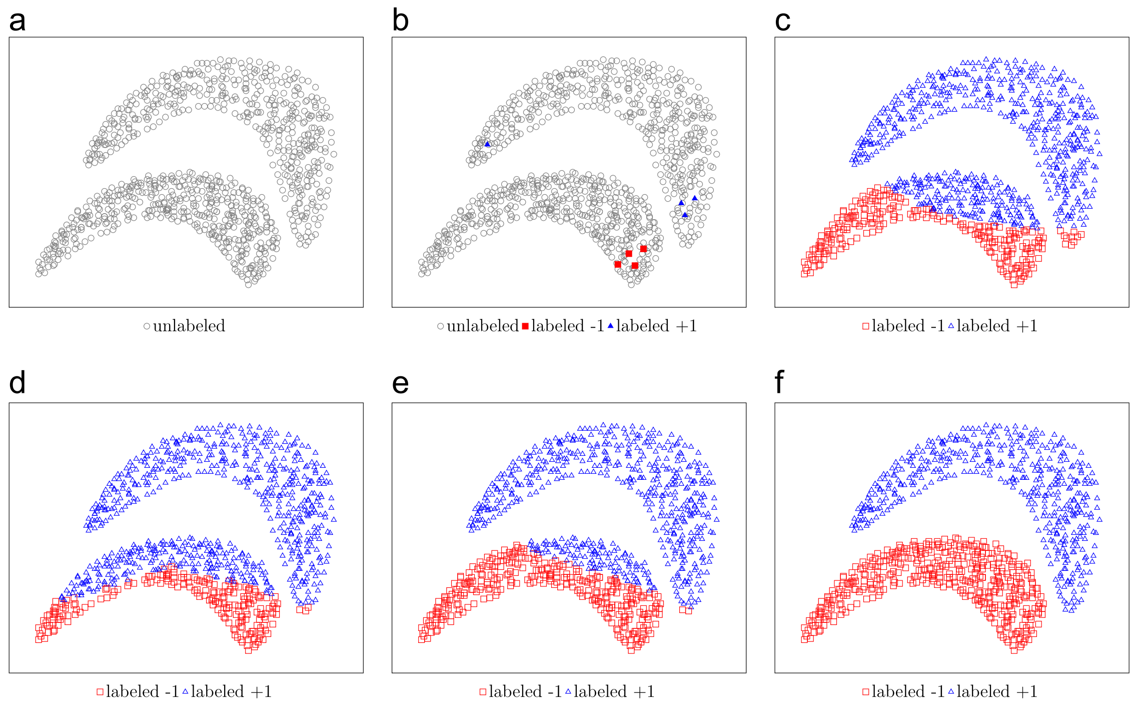

Fig. 1. An example of a training set with two half-moon-shaped classes. (a) Unlabeled samples, (b) a few manually labeled samples in each class. Label propagation to the remaining samples is computed by (c) SVM with RBF, (d) 1-NN, (e) OPFSUP, and (f) OPFSEMI$_{mst}$ (the same for OPFSEMI).

evaluating extended paths to them.

For supervised learning [10], a training set with only labeled samples is interpreted as a complete graph (the adjacency relation connects all training samples to each other) and the connectivity function forces the roots of the forest to be the closest samples between distinct classes (the most informative ones). In OPFSEMI [5], the training set consists of labeled and unlabeled samples, and the classifier requires two executions of the OPF algorithm. In the first execution, the roots of the forest are forced to be the closest labeled samples between distinct classes and, in the second execution, the root set is improved by adding informative samples (previously unlabeled and now labeled) to it. In both methods, the execution times to estimate those informative samples (roots of the forest) and to compute the optimum-path forest are $O(n^2)$, for a complete graph with n nodes. Therefore, OPFSEMI requires four executions of time cost $O(n^2)$, where n is the number of training samples.

In OPFSEMI$_{mst}$, the adjacency relation is defined as the set of arcs of a Minimum-Spanning Tree (MST) of the complete graph, whose nodes are the labeled and unlabeled samples, being computed in $O(n^2)$ for n nodes. We then simplify the choice of the forest roots to be all labeled samples and the classifier is created from a single execution of the OPF algorithm, in time $O(n \log n)$, on the topology of the MST. One may say that the supervised classifier [10], named OPFSUP, could also be used to propagate labels to the unlabeled samples, one by one, without exploring the spatial distribution of the unlabeled samples (see category (a) above), and then be retrained from the completely labeled set. However, the accuracies of OPFSEMI and OPFSEMI$_{mst}$, as representatives of the category (b) above, are much higher than the one of OPFSUP for label propagation. Fig. 1 illustrates this aspect by comparing the performance in label propagation of OPFSEMI$_{mst}$ (which is exactly the same of OPFSEMI in this simple example) with the performance of methods from category (a), by using three popular supervised classification models.

Fig. 1 a shows the training set wherein a few labeled samples are presented in Fig. 1b. Such a set of labeled samples is actually plausible, given that an expert must select and annotate samples for it from the initially unlabeled data with unknown distribution in the feature space.[1] The results of label propagation to the remaining samples of Fig. 1b are shown in Fig. 1c–e for the classifiers, SVM with Radial Basis Function (RBF) kernel [11], 1-NN [12], and OPFSUP [10], respectively. For OPFSEMI$_{mst}$, the connectivity between labeled and unlabeled samples allows us to propagate labels with no errors in this case (Fig. 1f).

The results of OPFSEMI and OPFSEMI$_{mst}$ should be equivalent in theory, because any path between two nodes in a minimum-spanning tree is optimum according to the connectivity function used by both. However, their choices of the roots of the forest are not the same, and due to that, in practice, we have found that OPFSEMI$_{mst}$ is more accurate than OPFSEMI. Additionally, the OPF-based semi-supervised methods are (a) free of parameters, (b) treat multi-class problems in a natural way, (c) do not make assumptions about the shapes of the classes, and (d) can handle some overlapping between classes, as long as the roots of the forest can protect their respective classes.

This paper is organized as follows. Section 2 presents the related works, with our choice of baselines for the experiments. Section 3 describes OPFSEMI$_{mst}$. Section 4 shows the experiments and their results are discussed in Section 5. Section 6 draws conclusions and provides directions to future work.

2. Related works

The methods OPFSEMI$_{mst}$ and its previous conference version, OPFSEMI [5], are graph-based approaches for semi-supervised learning. In this category, we should mention a popular approach based on harmonic functions and Gaussian fields [13], which has been implemented in SemiL[2] – a tool for solving large-scale transductive inference problems. This method has been

[1] It is common to select uniformly distributed samples from each class for the training set, when comparing classifiers. Note that this option is not often possible in practice, since the acquired data are initially unlabeled.

[2] http://www.support-vector.ws/html/semil.html

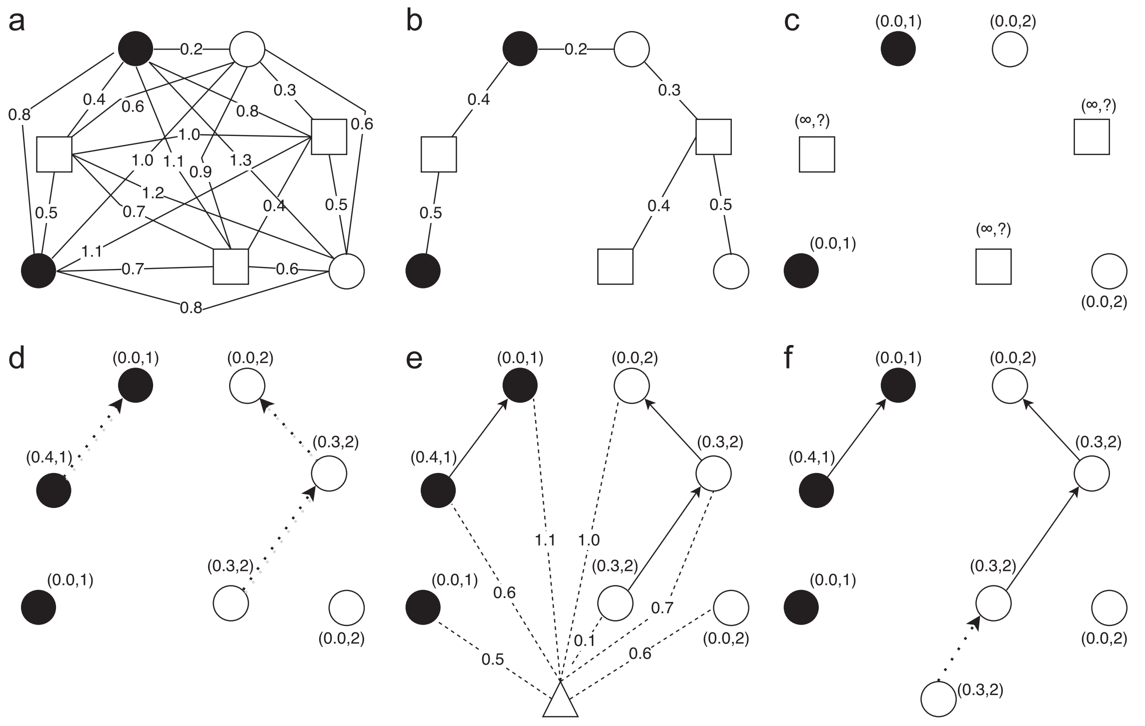

Fig. 2. (a) Complete and weighted graph for a simple training set (• labeled samples of class 1, ○ labeled samples of class 2 and □ unlabeled samples). (b) A minimum-spanning tree from (a) and (c) a trivial path-value map. Labeled samples are forced to be the minima (cost 0) of the map, and unlabeled samples are assigned to infinite cost. The entries (*x*, *y*) over the nodes are, respectively, the cost and the label of the samples. (d) An optimum-path forest after path and label propagation on (b) by starting from the path-value map in (c). The directed arcs indicate the predecessor nodes in the optimum path. (e) A new sample △ and (f) its classification by extending the optimum path from its the most closely connected root.

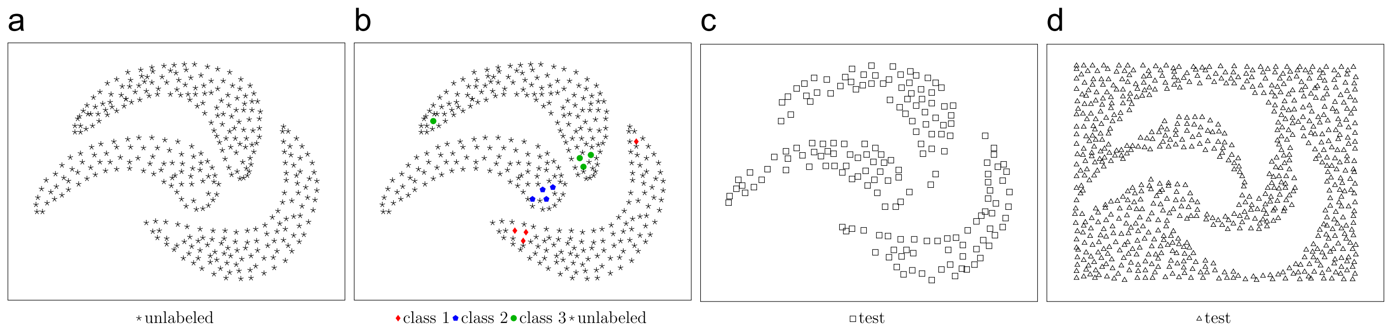

Fig. 3. A data set with (a) unlabeled samples. (b) A training set with unlabeled and a few labeled samples, (c) test samples inside the classes, and (d) test samples outside the classes.

successfully used in several domains [14–16].

Basu et al. [1] proposed two semi-supervised learning methods based on the *k*-means clustering algorithm. In both methods, labeled samples are used to estimate the initial *k* cluster representatives. The remaining samples, including the unlabeled ones, are assigned to the cluster of their closest representatives. The representatives are recomputed and the process repeats until convergence. As difference, in one of the methods, the labeled samples must remain in their initial clusters. It is not clear in these methods how they guarantee at least one cluster per class, but alternatives to the problem have been proposed in [17].

Li et al. [2] introduced *SVM-KNN* – a hybrid semi-supervised approach based on Support Vector Machines (SVM) and *k*-Nearest Neighbors (*k*-NN). First, an SVM classifier, trained from the labeled samples, propagates labels to the unlabeled data. The samples near the boundary between classes are called *boundary vectors*. By construction, the SVM classification of such informative samples may not be correct. Then boundary vectors are reclassified by the *k*-NN algorithm, using the remaining samples labeled by SVM as a training set.

Rosenberg et al. [3] presented a strategy based on self-training [18]. First, the classifier trained from the labeled samples assigns labels to the unlabeled data. The samples classified with higher confidence are added to the training set and the process starts over until it achieves a convergence criterion. Blum et al. [4] presented the Co-Training method, in which labeled samples are divided into two subsets. Two supervised classifiers are trained on each subset and each classifier makes its prediction on the unlabeled samples, thus teaching the other classifier their labels. Samples classified with higher confidence are used to increase the training set of the other classifier, and the retraining process starts over until it achieves a convergence criterion.

Joachims [6] proposed the Transductive Support Vector Machines (TSVM), in which the SVM hyperplane of maximum separation between classes must also satisfy a second criterion of being far away from unlabeled samples. The idea does not work well for a high number of unlabeled samples [19], which is usually the case, but subsequently Collobert et al. [20] introduced the concave–convex optimization procedure (CCP) to improve TSVM [5,21–24].

Semi-supervised learning has been a topic of continuous progress over the last 10 years, with methods based on Laplacian

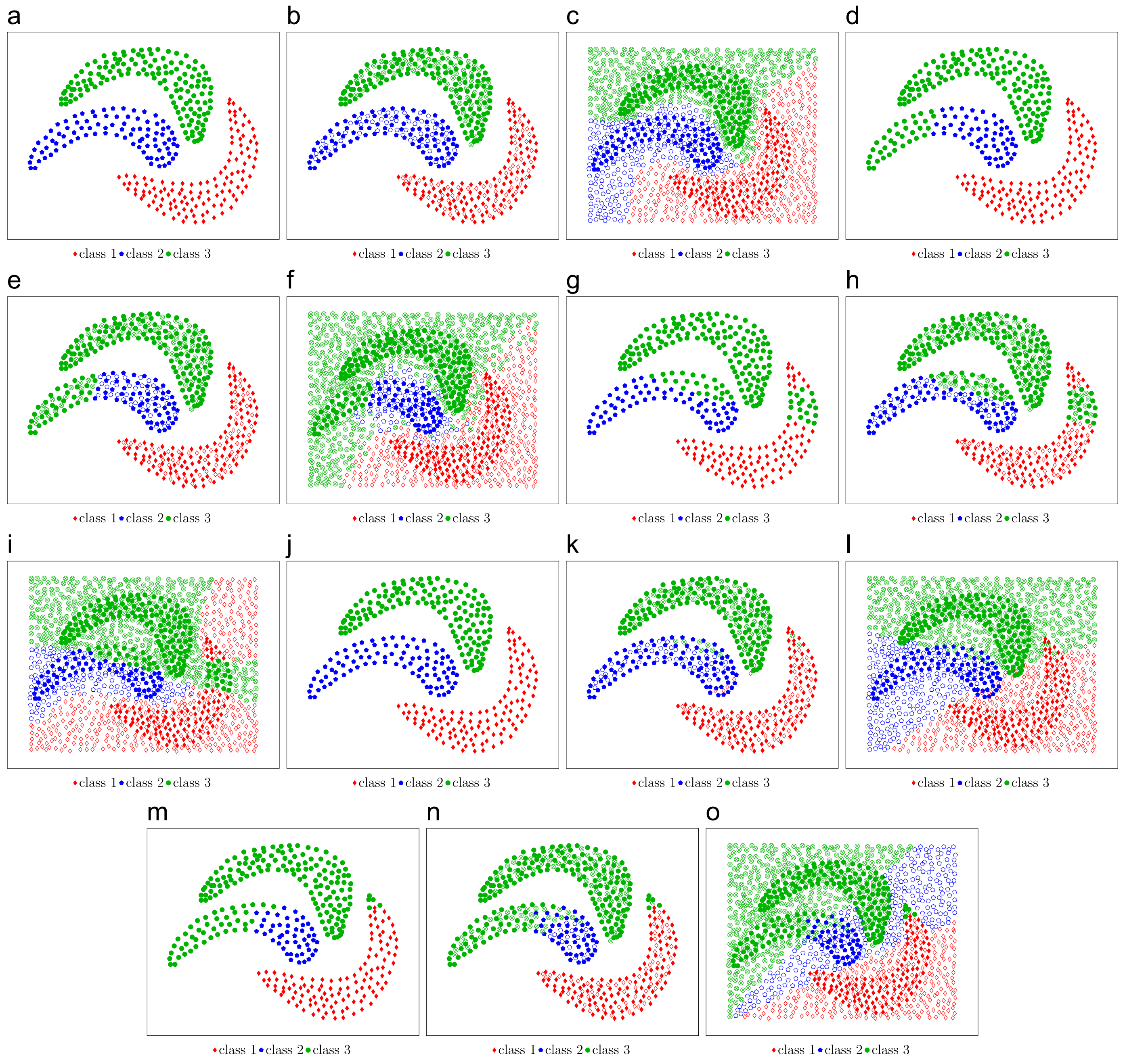

Fig. 4. Label propagation, classification of test samples inside the classes, classification for test samples inside and outside the classes for (a–c) OPFSEMI$_{mst}$ (the same for OPFSEMI), (d–f) SemiL, (g–i) TSVM, (j–l) LapSVM and (m–o) SSELM, respectively.

regularization [25], Extreme Learning Machine (ELM) [26,27], and manifold regularization [28], just to name a few. Manifold regularization seems to be among the most actively pursued [29–32]. However, in [33], the authors proposed USELM and SSELM – extensions of ELMs to handle unsupervised and semi-supervised learning problems, respectively, by using manifold regularization to cope with the absence or scarcity of labeled data.

Therefore, in view of consolidating OPFSEMI$_{mst}$ among the most popular semi-supervised learning approaches, and also taking into account the available codes, we have selected the following methods for comparison: its previous version, OPFSEMI [5], TVSM with CCP (as implemented in UniverSVM [20],[3]), SemiL [13] (another graph-based approach), the LapSVM algorithm by Belkin et al. [28] (manifold regularization), the SSELM algorithm by Huang et al. [33] (extreme learning machine with manifold regularization) and two methods based on supervised classification, OPFSUP [10] and SVM with RBF [20].

[3] UniverSVM seems to be the strongest competitor, with publicly available code, in running time and classification performance.

3. The improved semi-supervised optimum-path forest classifier

Let $\mathcal{Z}$ be a data set whose samples $s \in \mathcal{Z}$ are represented by feature vectors $\vec{v}(s) \in \Re^n$. We randomly divide $\mathcal{Z}$ into subsets $\mathcal{Z}_1$ for the design of the classifier (training) and $\mathcal{Z}_2$ for testing its generalization ability. Set $\mathcal{Z}_1 = \mathcal{Z}_1^l \cup \mathcal{Z}_1^u$ also consists of labeled $\mathcal{Z}_1^l$ and unlabeled $\mathcal{Z}_1^u$ subsets of samples. Additionally, let $\lambda(s) \in \{1, 2, \ldots, c\}$ for c classes be the true label of each sample $s \in \mathcal{Z}$ and $d(s,t) \geq 0$ be a symmetric distance function between samples, according to their feature vectors, such as $d(s,t) = \| \vec{v}(t) - \vec{v}(s) \|$. For training, the method must connect samples from $\mathcal{Z}_1$ into a graph and propagate labels to $\mathcal{Z}_1^u$ such that the classifier will be an optimum-path forest rooted at $\mathcal{Z}_1^l$. The classification of new samples from $\mathcal{Z}_2$ is performed by evaluating extended optimum paths. The algorithms for each step are described next.

3.1. Training

In order to train the semi-supervised optimum-path forest classifier, OPFSEMI$_{mst}$, we first consider an adjacency relation $\mathcal{A} = \mathcal{Z}_1 \times \mathcal{Z}_1$, which defines a complete and weighted graph $(\mathcal{Z}_1, \mathcal{A}, d)$, and compute from $(\mathcal{Z}_1, \mathcal{A}, d)$ a minimum spanning-tree $(\mathcal{Z}_1, \mathcal{B}, d)$, to be used as input graph for optimum-path forest computation. The arcs in $\mathcal{B}$ already connect the closest labeled and unlabeled samples, but a node $t \in \mathcal{Z}_1^u$ can still be reached by paths from nodes of $\mathcal{Z}_1^l$ with distinct labels. Therefore, labeled nodes will compete with each other and the label $L_1(t) \leftarrow \lambda(s)$ assigned to t will come from its most closely connected node $s \in \mathcal{Z}_1^l$. A label propagation error occurs when $L_1(t) \neq \lambda(t)$.

The Optimum-Path Forest algorithm is a variant of Dijkstra's algorithm for multiple sources and more general connectivity functions [7]. Differently from the original conditions, its correctness requires constraints applied to only optimum paths. For a given graph, a path π_t with terminus t is simple when it is a sequence $\langle s_1, s_2, \ldots, t \rangle$ of distinct adjacent nodes and, when $\pi_t = \langle t \rangle$, it is said to be trivial. A path π_t is optimum for a given connectivity function when $f(\pi_t) \leq f(\tau_t)$ for any other path τ_t with terminus t in the graph, irrespective of its origin. We write $\pi_t = \pi_s \cdot \langle s, t \rangle$ to indicate the extension of a path π_s by the arc (s,t) in the graph.

For training, we first consider the graph $(\mathcal{Z}_1, \mathcal{A}, d)$ with connectivity function f_{mst}, which generates $(\mathcal{Z}_1, \mathcal{B}, d)$ – a Minimum-Spanning Tree: an acyclic and connected graph, where $\sum_{\forall (s,t) \in \mathcal{B} \subseteq \mathcal{A}} \{d(s,t)\}$ is minimum. We then consider the MST as input graph with connectivity function $f_{\max}$ for optimum-path computation purpose:

$$f_{mst}(\langle s \rangle) = \begin{cases} 0 & \text{for one arbitrary } s \in \mathcal{Z}_1, \\ +\infty & \text{otherwise,} \end{cases}$$
$$f_{mst}(\pi_s \cdot \langle s, t \rangle) = d(s,t), \tag{1}$$

$$f_{\max}(\langle s \rangle) = \begin{cases} 0 & \text{if } s \in \mathcal{Z}_1^l, \\ +\infty & \text{otherwise,} \end{cases}$$
$$f_{\max}(\pi_s \cdot \langle s, t \rangle) = \max\{f_{\max}(\pi_s), d(s,t)\}. \tag{2}$$

For f_{mst}, the OPF algorithm is equivalent to Prim's algorithm, as follows, by transforming the complete graph in Fig. 2a, for example, into the minimum-spanning tree of Fig. 2b.

Algorithm 1. The OPF ALGORITHM for f_{mst}.

```
    INPUT:      A complete and weighted graph $(\mathcal{Z}_1, \mathcal{A}, d)$.
    OUTPUT:     A minimum-spanning tree $(\mathcal{Z}_1, \mathcal{B}, d)$.
184 AUXILIARY:  Priority queue $Q$, cost variable $cst$, path-cost map $C_1$, predecessor map
                $P_1$, and $color(s)$ that is: $white$ when $s$ has never been inserted in $Q$;
                $gray$ when $s \in Q$; and $black$ when $s$ has been removed from $Q$.

1.  Set $\mathcal{B} \leftarrow \emptyset$.
2.  For each node $t \in \mathcal{Z}_1$, set $C_1(t) \leftarrow +\infty$ and $color(t) \leftarrow white$.
3.  Select any node $s \in \mathcal{Z}_1$, set $C_1(s) \leftarrow 0$, $P_1(s) \leftarrow nil$, $color(s) \leftarrow gray$, and insert $s$ in $Q$.
4.  While $Q$ is not empty, do
5.      Remove from $Q$ a sample $s$ such that $C_1(s)$ is minimum and set $color(s) \leftarrow black$.
6.      If $P_1(s) \neq nil$ then $\mathcal{B} \leftarrow \mathcal{B} \cup \{(s, P_1(s)), (P_1(s), s)\}$.
7.      For each $t \in \mathcal{A}(s)$ do
8.          If $color(t) \neq black$, then
9.              Set $cst \leftarrow d(s,t)$.
10.             If $cst < C_1(t)$, then
11.                 Set $P_1(t) \leftarrow s$ and $C_1(t) \leftarrow cst$.
12.                 If $color(t) = gray$, then update position of $t$ in $Q$
13.                 Else insert $t$ in $Q$ and set $color(t) \leftarrow gray$.
14. Return a minimum-spanning tree $(\mathcal{Z}_1, \mathcal{B}, d)$.
```

For input $(\mathcal{Z}_1, \mathcal{B}, d)$ with connectivity function $f_{\max}$, the OPF algorithm will minimize a path-cost map C_1 (connectivity map) by considering all possible paths with terminus t and assign to t the cost $C_1(t)$ of an optimum path π_t, which can be obtained backwards from a predecessor map P_1 – Optimum-Path Forest: an acyclic map defined for all nodes in $\mathcal{Z}_1$ as $P_1(t) = nil$, when $t \in \mathcal{Z}_1^l$ is a root of the map, or $P_1(t) = s \in \mathcal{Z}_1$, when s is the predecessor of t in the optimum path π_t. Each sample $s \in \mathcal{Z}_1^l$ will be root of one optimum-path tree and each class will be represented by its root samples. The true labels $\lambda(s)$ of the roots $s \in \mathcal{Z}_1^l$ can be propagated to create a label map L_1, where unlabeled samples $t \in \mathcal{Z}_1^u$ will be assigned to the label $L_1(t) \leftarrow \lambda(s)$ of its most closely connected root $s \in \mathcal{Z}_1^l$, as follows.

Algorithm 2. The OPF algorithm for $f_{\max}$.

```
INPUT:      Graph $(\mathcal{Z}_1, \mathcal{B}, d)$.
OUTPUT:     Maps of the optimum-path forest and its attributes $[P_1, C_1, L_1, \mathcal{Z}_1']$.
198 AUXILIARY:  Priority queue $Q$, cost variable $cst$, $color(s)$: $white$ when $s$ has never
            been inserted in $Q$; $gray$ when $s \in Q$; and $black$ when $s$ has been
            removed from $Q$.

1.  For each node $t \in \mathcal{Z}_1$, do
2.      set $C_1(t) \leftarrow +\infty$ and $color(t) \leftarrow white$.
3.      If $t \in \mathcal{Z}_1^l$, then
4.          set $C_1(t) \leftarrow 0$, $P_1(t) \leftarrow nil$, $L_1(t) \leftarrow \lambda(t)$, and $color(t) \leftarrow gray$.
5.          insert $t$ in $Q$.
6.  While $Q$ is not empty, do
7.      Remove from $Q$ a sample $s$ such that $C_1(s)$ is minimum and set $color(s) \leftarrow black$.
8.      Insert $s$ in $\mathcal{Z}_1'$.
9.      For each $t \in \mathcal{B}(s)$ do
10.         If $color(t) \neq black$, then
11.             Set $cst \leftarrow \max\{C_1(s), d(s,t)\}$.
12.             If $cst < C_1(t)$, then
13.                 Set $P_1(t) \leftarrow s$, $L_1(t) \leftarrow L_1(s)$, and $C_1(t) \leftarrow cst$.
14.                 If $color(t) = gray$, then update position of $t$ in $Q$
15.                 Else insert $t$ in $Q$ and set $color(t) \leftarrow gray$.
16. Return optimum-path forest and attributes $[P_1, C_1, L_1, \mathcal{Z}_1']$.
```

The MST computation from $(\mathcal{Z}_1, \mathcal{A}, d)$ has time complexity $O(|\mathcal{Z}_1|^2)$, since the graph is complete, while the time complexity of the optimum-path forest from $(\mathcal{Z}_1, \mathcal{B}, d)$ is $O(|\mathcal{Z}_1| \log |\mathcal{Z}_1|)$, since $|\mathcal{B}| \lll |\mathcal{Z}_1| \log |\mathcal{Z}_1|$. The nodes t of the forest in $\mathcal{Z}_1'$ are stored in their non-decreasing order of optimum path values $C_1(t)$. This is useful to speed-up classification as presented next.

3.2. Classification

For classification, optimum paths π_s for $s \in \mathcal{Z}_1'$ must be extended to new samples $t \in \mathcal{Z}_2$ by considering

$$C_2(t) = \min_{\forall s \in \mathcal{Z}_1'} \{\max\{C_1(s), d(s,t)\}\}, \tag{3}$$

and assigning to t the label $L_2(t) \leftarrow L_1(s^*)$ of the sample $s^* \in \mathcal{Z}_1'$ for which $\pi_{s^*} \cdot \langle s^*, t\rangle$ is optimum. Note that classification considers that t is connected to all nodes in $\mathcal{Z}_1$, rather than t being an additional node of the MST. Therefore, classification is based on the same rule used for the supervised OPF classifier [10], OPFSUP, but now using a much larger set $\mathcal{Z}_1'$ than the set $\mathcal{Z}_1^l$. It is also the same classification procedure used in OPFSEMI. By following the order of nodes in $\mathcal{Z}_1'$, the evaluation of $C_2(t)$ can halt whenever $C_1(s) \geq \max\{C_1(s^*), d(s^*, t)\}$ for some previous $s^* \in \mathcal{Z}_1'$. Fig. 2e and f illustrates the classification process, which can be implemented as follows.

Algorithm 3. OPF Classification Algorithm.

```
1.  For $t \in \mathcal{Z}_2$, do
2.      $Set\ i \leftarrow 1,\ mincost \leftarrow max\{C_1(s_i), d(s_i, t)\},\ s_i \in \mathcal{Z}_1'$.
3.      $L_2(t) \leftarrow L_1(s_i)\ and\ P_2(t) \leftarrow s_i$.
4.      While $i < |\mathcal{Z}_1'|\ and\ C_1(s_{i+1}) < mincost$, do
5.          $Compute\ cst \leftarrow \max\{C_1(s_{i+1}, d(s_{i+1}, t)\},\ s_{i+1} \in \mathcal{Z}_1'$.
6.          If $cst < mincost$, then
7.              $mincost \leftarrow cst$.
8.              $L_2(t) \leftarrow L(s_{i+1})\ and\ P_2(t) \leftarrow s_{i+1}$.
9.          $i \leftarrow i + 1$.
10. Return $[L_2, P_2]$.
```

Fig. 3 shows a comparison among semi-supervised methods on a simple data set with unlabeled samples (Fig. 3a), labeled and unlabeled training samples (Fig. 3b), test samples inside (Fig. 3c) and outside (Fig. 3d) the classes. The results of label propagation to $\mathcal{Z}_1^u$ and classification of $\mathcal{Z}_2$ with test samples inside and outside the classes are presented for OPFSEMI$_{mst}$ (they are the same for OPFSEMI) (Fig. 4a–c), SemiL [13] (Fig. 4d–f), TSVM [20] (Fig. 4g–i), LapSVM [28] (Fig. 4j–l), and SSELM [33] (Fig. 4m–o), respectively. The connectivity between labeled and unlabeled samples in OPFSEMI$_{mst}$ and manifold regularization in LapSVM can considerably reduce classification errors in $\mathcal{Z}_1^u$ and $\mathcal{Z}_2$, as compared to SSELM, SemiL and TSVM.

In the case of overlapping among classes, OPFSEMI$_{mst}$ becomes sensitive to the choice of the training samples for manual labeling. If they are selected in the overlapped class regions of the feature space, they can protect their classes, reducing label propagation and classification errors (Fig. 5a–c). However, when this is not the case, such errors might deteriorate its performance (Fig. 5d–f). This essentially suggests the use of OPFSEMI$_{mst}$ with an active learning approach, where the objective is to identify and select such informative samples for label correction/confirmation by an expert.

4. Experiments

The experiments involved the comparison among supervised and semi-supervised methods on several data sets with a variety of feature space dimensions. Among the semi-supervised approaches, OPFSEMI$_{mst}$ was compared with OPFSEMI, its previous conference version [5], the Transductive Support Vector Machines (TSVM) with Concave–Convex Optimization (CCP), as implemented in UniverSVM [20], the method based on harmonic functions and Gaussian fields, SemiL [13], the manifold regularization approach [28] in LapSVM[4] and the Semi-Supervised approach using Extreme Learning Machine [33] (SSELM[5]). Among the supervised methods, OPFSEMI$_{mst}$ was compared with the most popular version of the supervised optimum-path forest classifiers, OPFSUP [10], and SVM with Radial Basis Function (RBF) kernel [20] (also used as implemented in UniverSVM). All approaches based on optimum-path forest used the C library, named LibOPF[6] for their implementation. We will make available OPFSEMI$_{mst}$ in version 3.0 of LibOPF.

Next, we present the data sets, the evaluation methodology, and the methods used for parameter optimization of the supervised and semi-supervised evaluated approaches.

4.1. Data sets

Table 1 presents the selected data sets with their number of samples, classes, and attributes. The first six data sets are synthetic and publicly available. The KddCup is a data set composed by hundreds of thousands of samples. In this paper, we used a reduced data set, which is composed by 10% of the original data set size. The last two (Cowhide and Parasites) were obtained from real applications.

The Cowhide data set is composed of five types of regions of interest in the Wet-Blue[7] processing stage, namely: scabies, ticks, hot-iron, cut, and regions without defect (Fig. 6a–e). The main reason for selecting samples of cowhide defects is the challenge of the problem, especially in areas close to the vicinity of different defects.

The Parasites data set contains samples from 15 species of protozoa and helminths. These objects were obtained through faecal exams, by segmentation of optical microscopy images. The unbalance of samples across classes is a challenge, since the number of samples per class varies from 33 to 163 (Fig. 6f displays examples from all species in the data set).

[4] http://manifold.cs.uchicago.edu/manifold_regularization/

[5] http://www.ntu.edu.sg/home/egbhuang/elm_codes.html

[6] www.ic.unicamp.br/afalcao/libopf/

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

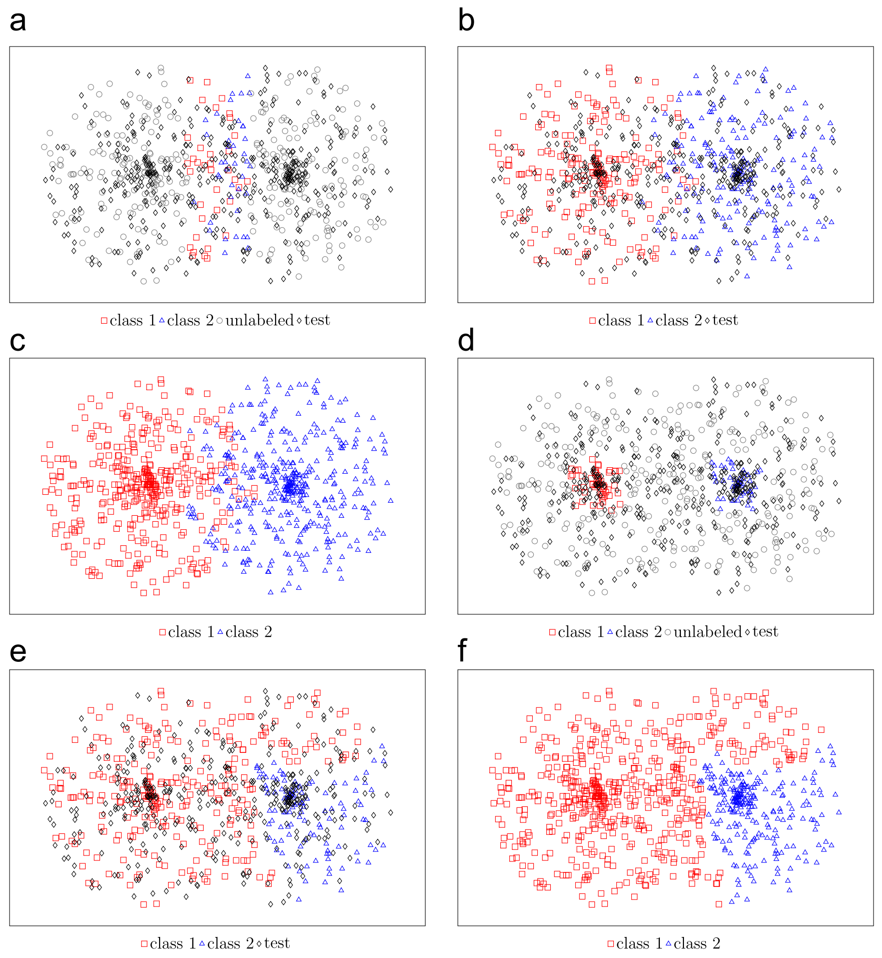

Fig. 5. Importance of informative sample selection. (a) A data set with overlapped classes, showing training (labeled and unlabeled) and test samples. Results of (b) label propagation and (c) classification for OPFSEMI$_{mst}$, when labeled samples are in the overlapped class regions. (d) When this is not the case, errors in (e) label propagation and (f) classification increase.

Table 1
Number of samples, attributes and classes of the data sets.

Data set	Samples	Attributes	Classes
Statlog [34]	2.310	19	7
Spambase [35]	4.601	57	2
Faces [36]	1.864	162	54
Pendigits [37]	10.992	16	10
KddCup [38]	48.898	41	23
Letter [39]	20.000	16	26
Cowhide [40]	1.690	160	5
Parasites [41]	1.660	262	15

4.2. Evaluation methodology

Each data set was randomly divided into two parts: 70% for the training set $\mathcal{Z}_1$ and 30% for the test set $\mathcal{Z}_2$. We also evaluated the methods for different proportions of random samples selected for the labeled set $\mathcal{Z}_1^l$ and the unlabeled set $\mathcal{Z}_1^u$, with $\mathcal{Z}_1^l \cup \mathcal{Z}_1^u = \mathcal{Z}_1$. The sizes of $\mathcal{Z}_1^l$ and $\mathcal{Z}_1^u$ ranged from 1–99% and 10–90% to 50–50% with respect to the size of $\mathcal{Z}_1$. The supervised approaches trained on $\mathcal{Z}_1^l$ and the classifiers were tested on $\mathcal{Z}_2$, while the semi-supervised methods first propagated labels from $\mathcal{Z}_1^l$ to $\mathcal{Z}_1^u$, trained on $\mathcal{Z}_1$, and then the classifiers were tested on $\mathcal{Z}_2$. The performance of the classifiers in accuracy was measured by giving higher weights to classes with lower number of samples, as suggested in [9].

For statistical analysis, the above procedure was repeated 100 times and we applied Friedman test [42] on the results. Friedman's test is a non-parametric test for testing the differences among multiple classifiers, and is an alternative for repeated measure analysis of variance, which is used when the same parameter has been measured under different conditions on the same subjects. When the difference in performance is statistically significant, the next step is a post hoc test to detect between which algorithms those differences appear. We adopted the Nemenyi test in this case. The difference in performance for two classifiers is considered statistically significant when their average ranks differ by more than a critical distance.

4.3. Parameter optimization

TSVM introduces several hyperparameters that need to be tuned. In our experiments, we tune $C \in \{10^{-5}, 10^{-3}, 10^{-1}, 10, 10^3, 10^5\}$ and $C_* \in \{10^{-5}, 10^{-3}, 10^{-1}, 0, 10\}$ – i.e., the cost parameters concerning the labeled and unlabeled data, respectively. We also employed the RBF kernel for both SVM, TSVM and LapSVM with $\gamma \in \{10^{-5}, 10^{-3}, 10^{-1}, 1, 10\}$ optimized by a 5-fold cross validation. For SSELM, we used the sigmoid function and the number of hidden neurons was fixed at 2000. The trade-off parameters C and γ were selected from the exponential sequence $\{10^{-6}, 10^{-5}, \ldots, 10^{6}\}$ (value set proposed in [33]). The remaining parameters used their default values. In SemiL, the weight matrices W were calculated with two different distance functions, Euclidean distance and Cosine distance with the RBF kernel and we used the *hard-label* approach with smoothness maximization (Gaussian Random Field Model – GRFM), as strongly recommended in [43].

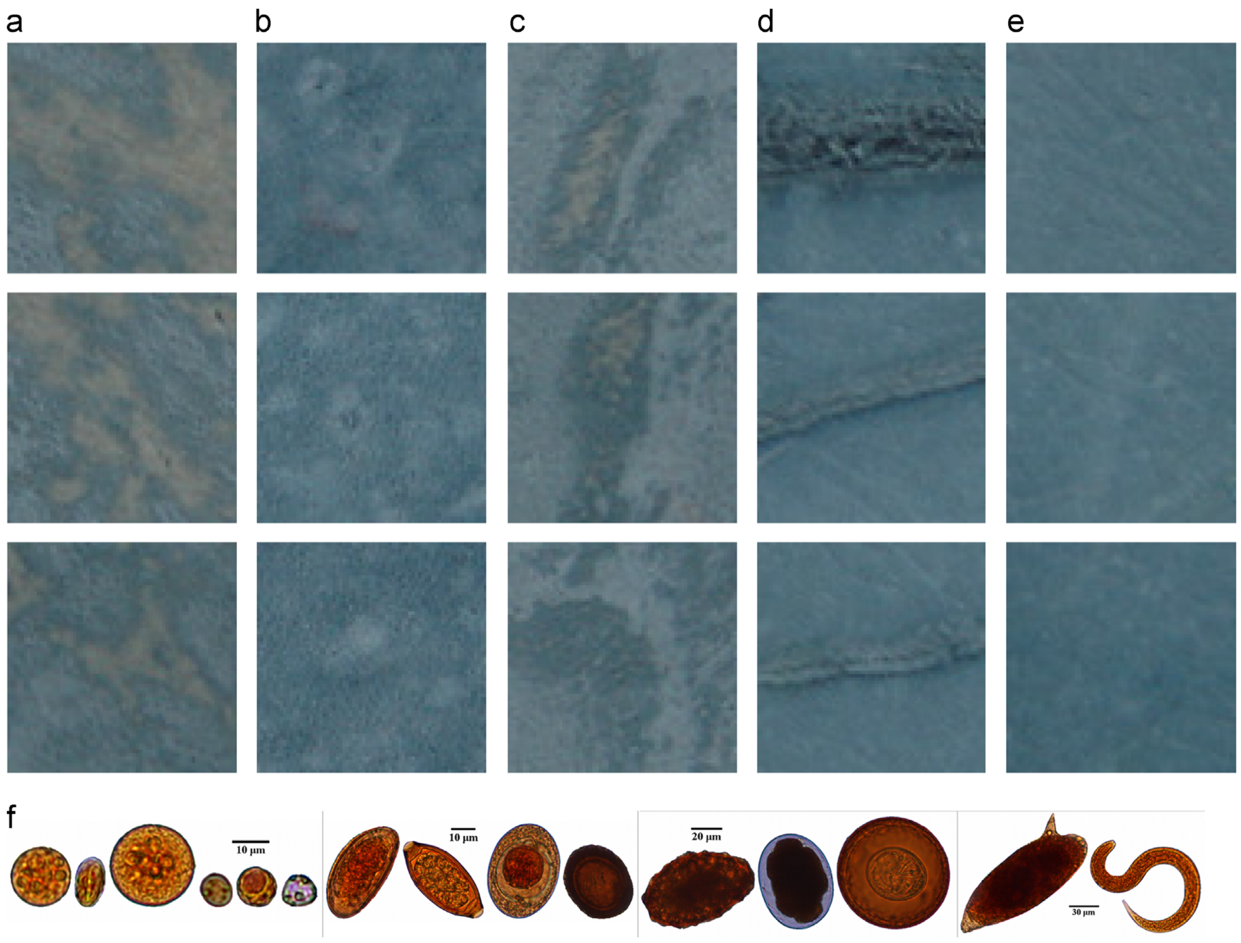

Fig. 6. (a) Scabies, (b) Tick, (c) Hot-iron, (d) Cut, (e) without defect and (f) examples of images from each class of the structures of intestinal parasites in our data set.

Table 2
Mean accuracy (%), standard deviation, training time (*tt.*) in seconds, and percentage of label propagation errors (*le.*) on $\mathcal{Z}_1^u$ – Cowhide data set.

$\mathcal{Z}_1^l$ (%)	$\mathcal{Z}_1^u$ (%)	**OPFSUP**	**SVM**	**TSVM**	**SemiL**	**LapSVM**	**SSELM**
1	99	82.71 ± 0.038	**85.19 ± 0.035**	83.14 ± 0.085	83.81 ± 0.007	83.44 ± 0.007	80.64 ± 0.083
10	90	90.88 ± 0.107	85.55 ± 0.033	84.25 ± 0.041	81.61 ± 0.016	92.40 ± 0.009	84.87 ± 0.046
20	80	93.19 ± 0.038	85.86 ± 0.085	89.48 ± 0.019	80.14 ± 0.012	92.06 ± 0.050	87.09 ± 0.042
30	70	93.59 ± 0.018	86.45 ± 0.050	86.48 ± 0.059	81.58 ± 0.042	94.38 ± 0.043	90.32 ± 0.087
40	60	94.29 ± 0.031	86.09 ± 0.083	86.68 ± 0.097	82.63 ± 0.075	95.43 ± 0.087	93.54 ± 0.016
50	50	95.77 ± 0.069	87.76 ± 0.040	90.04 ± 0.057	91.36 ± 0.096	96.01 ± 0.064	**96.77 ± 0.076**

$\mathcal{Z}_1^l$ (%)	$\mathcal{Z}_1^u$ (%)	**OPFSEMI**	*le.*	*tt.*	**OPFSEMI$_{mst}$**	*le.*	*tt.*
1	99	84.54 ± 0.076	24.84	0.282	84.54 ± 0.069	24.84	0.093
10	90	91.69 ± 0.040	12.01	0.286	**92.44 ± 0.015**	10.41	0.094
20	80	93.75 ± 0.008	9.50	0.296	**94.79 ± 0.053**	8.76	0.096
30	70	94.40 ± 0.076	9.52	0.296	**95.82 ± 0.053**	6.75	0.094
40	60	94.76 ± 0.092	8.59	0.311	**96.16 ± 0.016**	5.49	0.096
50	50	95.80 ± 0.107	4.56	0.296	96.68 ± 0.069	3.89	0.090

5. Results

Tables 2–9 show the mean accuracy (%) and its standard deviation for each classifier on $\mathcal{Z}_2$. The training time (*tt.*) in seconds and the percentage of the label propagation errors on $\mathcal{Z}_1^u$ (*le.*), are also shown for OPFSEMI$_{mst}$ and OPFSEMI. The best results among the methods of both tables are displayed in bold.

According to the Friedman test [44], the results presented in Tables 2–9 reject the *null* hypothesis that all classifiers are equivalent. Therefore, Figs. 7–8 present a graphical representation of the Nemenyi test, in which 1 represents the best technique, while 8 stands for the worst one. Groups of classifiers that are considered equivalent (at $p=0.05$) are connected by using a calculated critical distance (CD) equals to 4.2863 (Fig. 7). Only in the case of Fig. 8 (regarding the analysis of all data sets together), we can see a different critical distance (CD) equals to 1.5154 (at $p=0.05$). It is worth noting the importance of the statistical test, since the mean values in some cases are not sufficient to indicate the best classifier.

The tests evidenced that OPFSEMI$_{mst}$ performed best among

Table 3
Mean accuracy (%), standard deviation, training time (*tt.*) in seconds, and percentage of label propagation errors (*le.*) on $\mathcal{Z}_1^u$ – Statlog data set.

$\mathcal{Z}_1^l$ (%)	$\mathcal{Z}_1^u$ (%)	**OPFSUP**	**SVM**	**TSVM**	**SemiL**	**LapSVM**	**SSELM**
1	99	84.75 ± 0.061	81.45 ± 0.022	73.61 ± 0.013	78.62 ± 0.077	82.53 ± 0.065	82.05 ± 0.024
10	90	88.76 ± 0.031	85.41 ± 0.028	88.68 ± 0.081	87.20 ± 0.111	89.70 ± 0.093	85.18 ± 0.032
20	80	87.11 ± 0.084	90.33 ± 0.054	85.05 ± 0.042	83.80 ± 0.056	90.04 ± 0.013	89.07 ± 0.068
30	70	91.6 ± 0.084	92.33 ± 0.088	89.13 ± 0.048	80.22 ± 0.048	92.21 ± 0.054	90.86 ± 0.010
40	60	92.54 ± 0.038	93.20 ± 0.033	89.22 ± 0.015	88.38 ± 0.071	93.56 ± 0.043	91.20 ± 0.012
50	50	93.14 ± 0.051	93.25 ± 0.079	89.01 ± 0.067	90.29 ± 0.080	93.99 ± 0.097	94.84 ± 0.053

$\mathcal{Z}_1^l$ (%)	$\mathcal{Z}_1^u$ (%)	**OPFSEMI**	*le.*	*tt.*	**OPFSEMI$_{mst}$**	*le.*	*tt.*
1	99	85.56 ± 0.013	29.10	0.294	**85.72 ± 0.053**	29.10	0.093
10	90	91.73 ± 0.107	25.01	0.299	**91.76 ± 0.044**	18.78	0.094
20	80	91.94 ± 0.099	24.83	0.293	**93.11 ± 0.099**	16.80	0.091
30	70	91.91 ± 0.046	21.76	0.312	**93.85 ± 0.114**	15.38	0.095
40	60	93.15 ± 0.081	17.81	0.310	**94.47 ± 0.031**	15.09	0.095
50	50	93.96 ± 0.014	16.71	0.316	**95.21 ± 0.015**	14.11	0.091

Table 4
Mean accuracy (%), standard deviation, training time (*tt.*) in seconds, and percentage of label propagation errors (*le.*) on $\mathcal{Z}_1^u$ – Faces data set.

$\mathcal{Z}_1^l$ (%)	$\mathcal{Z}_1^u$ (%)	**OPFSUP**	**SVM**	**TSVM**	**SemiL**	**LapSVM**	**SSELM**
1	99	80.77 ± 0.089	75.13 ± 0.019	68.45 ± 0.070	79.25 ± 0.064	80.12 ± 0.014	79.62 ± 0.095
10	90	85.47 ± 0.038	71.31 ± 0.051	77.61 ± 0.051	83.81 ± 0.021	91.06 ± 0.005	81.48 ± 0.094
20	80	92.95 ± 0.072	80.81 ± 0.002	84.38 ± 0.088	84.37 ± 0.082	92.31 ± 0.005	85.63 ± 0.096
30	70	95.43 ± 0.015	82.24 ± 0.017	80.34 ± 0.052	87.49 ± 0.041	95.28 ± 0.084	90.74 ± 0.036
40	60	97.15 ± 0.13	90.24 ± 0.013	84.35 ± 0.036	90.38 ± 0.044	96.63 ± 0.017	92.59 ± 0.042
50	50	97.48 ± 0.023	97.13 ± 0.008	90.04 ± 0.059	93.61 ± 0.094	98.14 ± 0.060	96.28 ± 0.071

$\mathcal{Z}_1^l$ (%)	$\mathcal{Z}_1^u$ (%)	**OPFSEMI**	*le.*	*tt.*	**OPFSEMI$_{mst}$**	*le.*	*tt.*
1	99	88.62 ± 0.046	23.21	0.556	**89.15 ± 0.053**	23.21	0.181
10	90	91.75 ± 0.084	15.22	0.552	**93.35 ± 0.031**	13.86	0.180
20	80	94.75 ± 0.042	9.57	0.558	**96.12 ± 0.015**	6.96	0.182
30	70	97.02 ± 0.046	4.26	0.562	**98.14 ± 0.038**	2.40	0.180
40	60	97.79 ± 0.061	3.16	0.581	**98.38 ± 0.137**	2.15	0.182
50	50	97.61 ± 0.015	3.11	0.617	**98.61 ± 0.053**	1.67	0.187

Table 5
Mean accuracy (%), standard deviation, training time (*tt.*) in seconds, and percentage of label propagation errors (*le.*) on $\mathcal{Z}_1^u$ – Parasites data set.

$\mathcal{Z}_1^l$ (%)	$\mathcal{Z}_1^u$ (%)	**OPFSUP**	**SVM**	**TSVM**	**SemiL**	**LapSVM**	**SSELM**
1	99	88.09 ± 0.092	78.15 ± 0.035	71.78 ± 0.016	82.52 ± 0.058	82.56 ± 0.051	82.44 ± 0.059
10	90	96.11 ± 0.015	94.45 ± 0.029	91.84 ± 0.108	88.25 ± 0.012	92.28 ± 0.026	89.36 ± 0.040
20	80	97.26 ± 0.084	98.56 ± 0.047	89.32 ± 0.007	84.33 ± 0.073	94.93 ± 0.041	90.42 ± 0.069
30	70	98.00 ± 0.114	98.41 ± 0.068	94.22 ± 0.035	88.11 ± 0.111	94.85 ± 0.064	94.14 ± 0.020
40	60	97.93 ± 0.039	97.79 ± 0.013	95.85 ± 0.061	86.66 ± 0.019	95.67 ± 0.034	95.21 ± 0.042
50	50	98.42 ± 0.031	**98.88 ± 0.040**	94.56 ± 0.008	93.42 ± 0.064	96.52 ± 0.009	97.87 ± 0.070

$\mathcal{Z}_1^l$ (%)	$\mathcal{Z}_1^u$ (%)	**OPFSEMI**	*le.*	*tt.*	**OPFSEMI$_{mst}$**	*le.*	*tt.*
1	99	91.94 ± 0.019	13.03	0.902	**92.01 ± 0.053**	13.03	0.301
10	90	97.85 ± 0.094	4.56	0.946	**97.94 ± 0.038**	4.56	0.311
20	80	97.69 ± 0.023	4.40	0.909	**97.82 ± 0.046**	3.76	0.298
30	70	98.36 ± 0.023	3.14	0.949	**98.43 ± 0.094**	3.14	0.306
40	60	98.43 ± 0.069	3.31	0.893	**98.45 ± 0.094**	3.31	0.280
50	50	98.79 ± 0.015	2.64	0.916	98.85 ± 0.084	1.90	0.281

the evaluated techniques in most cases, followed by OPFSEMI, LapSVM, OPFSUP, SVM, SSELM and finally TSVM and SemiL. We shall highlight the good performance of LapSVM and SVM in Spambase (Fig. 7e), but overall LapSVM produces better classification results than SVM. SVM outperformed its semi-supervised version, TSVM, in most cases, which did not happen for OPFSEMI$_{mst}$ with respect to OPFSUP in any of the cases. Although the performance of SSELM was below the one of OPFSEMI$_{mst}$, SSELM outperformed TSVM and SemiL in most cases. The performances of TSVM and SemiL considerably improved with the increase from 1% to 50% of samples in $\mathcal{Z}_1^l$, but they were still not enough to outperform OPFSEMI$_{mst}$ (see, e.g., the Cowhide and Parasites data sets).

By performing a deeper analysis, we can compare statistically the pair, OPFSEMI and OPFSEMI$_{mst}$, by using Wilcoxon signed-rank test [44]. The Wilcoxon test is an important analysis that turns out to be more sensitive since it does not assume normal distributions. In this case, for $p = 3.640^{-9} (p < 0.05)$, they can be considered

Table 6
Mean accuracy (%), standard deviation, training time (*tt.*) in seconds, and percentage of label propagation errors (*le.*) on $\mathcal{Z}_1^u$ – Spambase data set.

$\mathcal{Z}_1^l$ (%)	$\mathcal{Z}_1^u$ (%)	**OPFSUP**	**SVM**	**TSVM**	**SemiL**	**LapSVM**	**SSELM**
1	99	58.76 ± 0.042	60.75 ± 0.026	60.37 ± 0.082	58.59 ± 0.085	65.12 ± 0.028	61.43 ± 0.045
10	90	65.89 ± 0.017	64.15 ± 0.066	**67.98 ± 0.069**	62.20 ± 0.102	66.29 ± 0.037	62.75 ± 0.088
20	80	66.13 ± 0.038	**74.75 ± 0.031**	70.13 ± 0.082	66.31 ± 0.078	72.87 ± 0.074	66.89 ± 0.096
30	70	67.54 ± 0.096	**76.39 ± 0.054**	73.41 ± 0.014	70.23 ± 0.066	76.01 ± 0.028	69.07 ± 0.022
40	60	68.99 ± 0.044	**76.95 ± 0.075**	69.84 ± 0.007	71.34 ± 0.014	75.35 ± 0.081	72.11 ± 0.031
50	50	70.16 ± 0.099	77.48 ± 0.056	71.03 ± 0.013	71.85 ± 0.099	**78.06 ± 0.016**	73.91 ± 0.030

$\mathcal{Z}_1^l$ (%)	$\mathcal{Z}_1^u$ (%)	**OPFSEMI**	*le.*	*tt.*	**OPFSEMI$_{mst}$**	*le.*	*tt.*
1	99	64.78 ± 0.053	35.07	1.890	**65.22 ± 0.061**	33.49	0.607
10	90	65.90 ± 0.027	32.22	1.899	66.12 ± 0.056	30.11	0.605
20	80	67.25 ± 0.092	30.73	1.950	68.69 ± 0.023	27.32	0.619
30	70	68.46 ± 0.062	30.42	1.998	71.53 ± 0.021	25.43	0.621
40	60	69.55 ± 0.058	29.02	2.071	73.42 ± 0.061	24.13	0.690
50	50	70.81 ± 0.067	27.74	2.133	74.18 ± 0.037	19.25	0.693

Table 7
Mean accuracy (%), standard deviation, training time (*tt.*) in seconds, and percentage of label propagation errors (*le.*) on $\mathcal{Z}_1^u$ – Pendigits data set.

$\mathcal{Z}_1^l$ (%)	$\mathcal{Z}_1^u$ (%)	**OPFSUP**	**SVM**	**TSVM**	**SemiL**	**LapSVM**	**SSELM**
1	99	94.31 ± 0.076	75.05 ± 0.025	74.23 ± 0.088	75.20 ± 0.015	93.05 ± 0.077	94.16 ± 0.042
10	90	96.54 ± 0.015	70.27 ± 0.064	70.43 ± 0.083	74.50 ± 0.035	97.20 ± 0.089	96.72 ± 0.083
20	80	98.88 ± 0.092	79.07 ± 0.028	76.80 ± 0.094	86.60 ± 0.076	97.49 ± 0.055	97.53 ± 0.094
30	70	99.19 ± 0.099	87.68 ± 0.078	88.57 ± 0.078	97.61 ± 0.010	97.93 ± 0.001	98.90 ± 0.020
40	60	99.14 ± 0.053	91.22 ± 0.035	82.88 ± 0.023	97.13 ± 0.098	98.05 ± 0.055	99.01 ± 0.024
50	50	99.17 ± 0.088	97.87 ± 0.020	85.65 ± 0.047	98.06 ± 0.026	98.29 ± 0.063	99.17 ± 0.089

$\mathcal{Z}_1^l$ (%)	$\mathcal{Z}_1^u$ (%)	**OPFSEMI**	*le.*	*tt.*	**OPFSEMI$_{mst}$**	*le.*	*tt.*
1	99	95.66 ± 0.015	7.58	6.957	**95.78 ± 0.084**	7.54	2.141
10	90	98.27 ± 0.053	5.89	7.168	**99.22 ± 0.031**	1.05	2.212
20	80	99.10 ± 0.076	1.33	7.084	**99.30 ± 0.069**	0.95	2.162
30	70	99.28 ± 0.046	1.07	7.214	**99.44 ± 0.015**	0.85	2.155
40	60	98.56 ± 0.033	2.62	7.328	**99.44 ± 0.015**	0.73	2.151
50	50	98.61 ± 0.053	2.67	7.469	**99.51 ± 0.071**	0.57	2.142

Table 8
Mean accuracy (%), standard deviation, training time (*tt.*) in seconds, and percentage of label propagation errors (*le.*) on $\mathcal{Z}_1^u$ – KddCup data set.

$\mathcal{Z}_1^l$ (%)	$\mathcal{Z}_1^u$ (%)	**OPFSUP**	**SVM**	**TSVM**	**SemiL**	**LapSVM**	**SSELM**
1	99	89.23 ± 0.033	80.44 ± 0.029	88.15 ± 0.063	73.92 ± 0.040	85.48 ± 0.052	79.15 ± 0.030
10	90	89.67 ± 0.088	87.01 ± 0.038	90.12 ± 0.011	83.57 ± 0.021	89.8 ± 0.065	83.36 ± 0.053
20	80	92.99 ± 0.072	86.91 ± 0.071	91.54 ± 0.081	84.35 ± 0.039	**93.16 ± 0.098**	88.14 ± 0.043
30	70	93.57 ± 0.092	87.30 ± 0.048	92.69 ± 0.093	87.66 ± 0.042	93.37 ± 0.081	90.74 ± 0.077
40	60	93.33 ± 0.048	87.14 ± 0.034	92.99 ± 0.076	90.54 ± 0.028	**94.55 ± 0.059**	92.32 ± 0.041
50	50	94.84 ± 0.086	91.76 ± 0.092	94.76 ± 0.084	92.28 ± 0.060	95.98 ± 0.052	94.47 ± 0.086

$\mathcal{Z}_1^l$ (%)	$\mathcal{Z}_1^u$ (%)	**OPFSEMI**	*le.*	*tt.*	**OPFSEMI$_{mst}$**	*le.*	*tt.*
1	99	85.57 ± 0.021	0.77	353.842	**90.06 ± 0.021**	0.53	142.629
10	90	90.53 ± 0.025	0.24	346.997	**90.64 ± 0.057**	0.24	141.964
20	80	92.78 ± 0.068	0.16	350.471	93.02 ± 0.029	0.14	140.428
30	70	93.69 ± 0.014	0.13	357.392	**93.73 ± 0.026**	0.12	140.857
40	60	93.71 ± 0.024	0.15	367.103	93.71 ± 0.083	0.12	141.467
50	50	95.60 ± 0.036	0.15	381.697	**96.21 ± 0.092**	0.09	141.231

statistically different. This confirms the improvement of OPFSEMI$_{mst}$ over its previous conference version in accuracy, since it is already better in efficiency. OPFSEMI$_{mst}$ was on average three times faster than OPFSEMI for training and it has also demonstrated to be robust to label propagation errors, which ranged from 0.11% to 33.49% of the samples in $\mathcal{Z}_1^u$.

6. Conclusion

We introduced a semi-supervised approach, named OPFSEMI$_{mst}$, based on the Optimum-Path Forest methodology. The method connects labeled and unlabeled samples into a minimum-spanning tree and computes an optimum-path forest rooted at the labeled nodes.

Table 9
Mean accuracy (%), standard deviation, training time (*tt.*) in seconds, and percentage of label propagation errors (*le.*) on $\mathcal{Z}_1^u$ – Letter data set.

$\mathcal{Z}_1^l$ (%)	$\mathcal{Z}_1^u$ (%)	**OPFSUP**	**SVM**	**TSVM**	**SemiL**	**LapSVM**	**SSELM**
1	99	75.05 ± 0.052	78.68 ± 0.070	74.56 ± 0.017	73.03 ± 0.083	**80.97** ± **0.037**	76.62 ± 0.051
10	90	88.76 ± 0.069	80.82 ± 0.016	79.23 ± 0.084	76.41 ± 0.082	82.69 ± 0.031	77.53 ± 0.018
20	80	92.94 ± 0.089	84.56 ± 0.021	84.06 ± 0.059	78.68 ± 0.098	84.73 ± 0.010	82.08 ± 0.063
30	70	94.17 ± 0.076	89.25 ± 0.035	90.11 ± 0.049	91.95 ± 0.046	92.31 ± 0.060	90.71 ± 0.011
40	60	94.33 ± 0.058	92.98 ± 0.055	93.42 ± 0.076	90.82 ± 0.092	94.16 ± 0.055	93.55 ± 0.078
50	50	95.43 ± 0.077	93.58 ± 0.024	94.28 ± 0.073	91.56 ± 0.032	**96.50** ± **0.088**	95.11 ± 0.060

$\mathcal{Z}_1^l$ (%)	$\mathcal{Z}_1^u$ (%)	**OPFSEMI**	*le.*	*tt.*	**OPFSEMI**$_{mst}$	*le.*	*tt.*
1	99	76.94 ± 0.059	32.79	45.612	77.50 ± 0.084	30.58	15.279
10	90	88.11 ± 0.083	20.51	47.494	**91.38** ± **0.011**	15.67	15.419
20	80	89.44 ± 0.091	17.99	47.539	**94.28** ± **0.088**	10.31	15.351
30	70	91.45 ± 0.046	14.19	48.453	**95.21** ± **0.034**	8.77	15.296
40	60	90.94 ± 0.012	15.12	51.003	**95.87** ± **0.031**	7.54	15.743
50	50	93.01 ± 0.053	11.10	51.757	96.17 ± 0.012	7.12	15.658

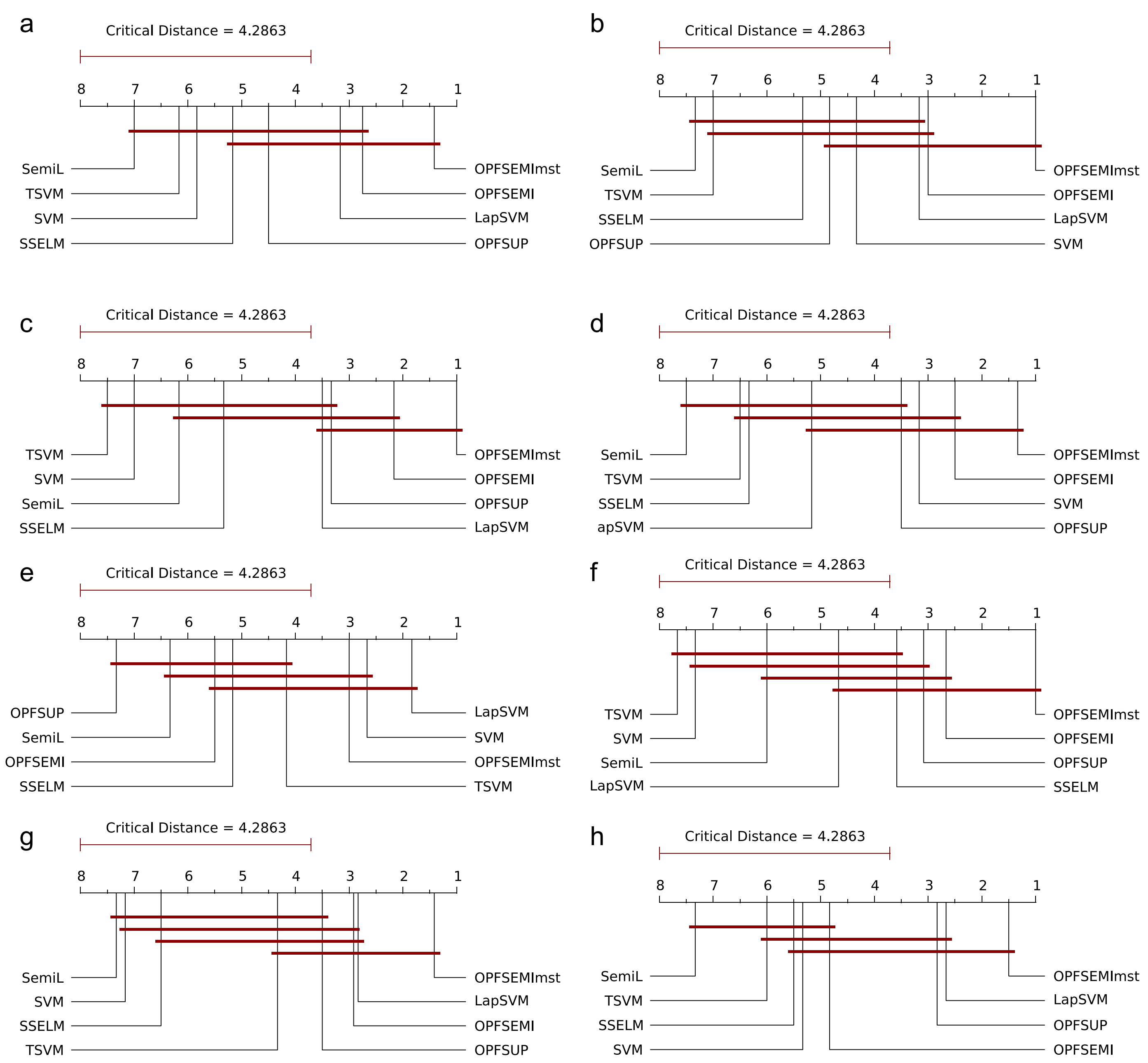

Fig. 7. Results of the Nemenyi test for all classifiers. Groups of equivalent classifiers are connected at $p = 0.05$. (a) Cowhide, (b) Statlog, (c) Faces, (d) Parasites, (e) Spambase, (f) Pendigits, (g) KddCup and (h) Letter.

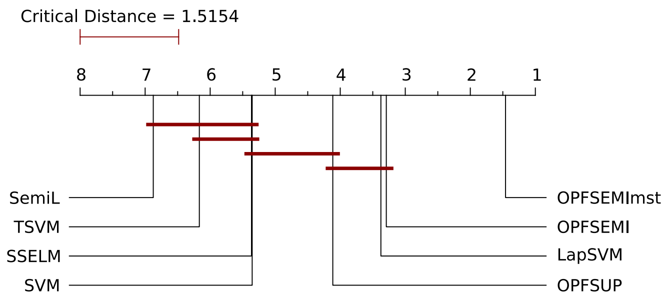

Fig. 8. Results of the Nemenyi test for all classifiers and all data sets. Groups of equivalent classifiers are connected at $p = 0.05$.

The methods exploit optimum connectivity between labeled and unlabeled samples to correctly classify informative unlabeled samples, increasing the training set size, and so improving classification performance on unseen samples. We discussed its pros and cons, and assessed OPFSEMI$_{mst}$ in comparison to two supervised approaches and five semi-supervised methods on eight data sets with a variety of feature space dimensions. OPFSEMI$_{mst}$ outperformed all approaches in most cases as statistically verified by Friedman with Nemenyi test.

We may conclude that OPFSEMI$_{mst}$ is a significant contribution for the literature of semi-supervised learning. Its potential to address classification problems with a small number of labeled samples in comparison to the number of unlabeled samples indicates that OPFSEMI$_{mst}$ should be further investigated in the development of active learning approaches. We then intend to use it in order to pursue the work recently reported in [45].

Conflict of interest

None declared.

Acknowledgment

The authors are grateful to Fundect-MS, CNP$_q$ grants: #303673/2010-9, #479070/2013-0, #302970/2014-2, #303182/2011-3, #470571/2013-6 and #306166/2014-3 and FAPESP grants: #2013/20387-7 and #2014/16250-9.

References

[1] S. Basu, A. Banerjee, R.J. Mooney, Semi-supervised clustering by seeding, in: Proceedings of the Nineteenth International Conference on Machine Learning, ICML'02, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2002, pp. 27–34.
[2] K. Li, X. Luo, M. Jin, Semi-supervised Learning for SVM-KNN, J. Comput. 5 (5) (2010) 671–678.
[3] C. Rosenberg, M. Hebert, H. Schneiderman, Semi-supervised self-training of object detection models, in: 2005 Seventh IEEE Workshops on Application of Computer Vision, WACV/MOTIONS '05, vol. 1, 2005, pp. 29–36.
[4] A. Blum, T. Mitchell, Combining labeled and unlabeled data with co-training, in: Proceedings of the Eleventh Annual Conference on Computational Learning Theory (COLT '98), ACM, New York, 1998, pp. 92–100.
[5] W.P. Amorim, A.X. Falcão, M.H.d. Carvalho, Semi-supervised pattern classification using optimum-path forest, in: 27th SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI), 2014, pp. 111–118.
[6] T. Joachims, Transductive inference for text classification using support vector machines, in: Proceedings of the Sixteenth International Conference on Machine Learning, ICML '99, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1999, pp. 200–209.
[7] A.X. Falcão, J. Stolfi, R. Lotufo, The image foresting transform: theory, algorithms, and applications, IEEE Trans. Patterns Anal. Mach. Intell. 26 (1) (2004) 19–29.
[8] L.M. Rocha, F.A.M. Cappabianco, A.X. Falcão, Data clustering as an optimum-path forest problem with applications in image analysis, Int. J. Imaging Syst. Technol. 19 (2) (2009) 50–68, http://dx.doi.org/10.1002/ima.20191.
[9] J.P. Papa, A.X. Falcão, C.T.N. Suzuki, Supervised pattern classification based on optimum-path forest, Int. J. Imaging Syst. Technol. 19 (2) (2009) 120–131.
[10] J.P. Papa, A.X. Falcão, V.H.C. de Albuquerque, J.M.R.S. Tavares, Efficient supervised optimum-path forest classification for large datasets, Pattern Recognit. 45 (1) (2012) 512–520.
[11] B. Boser, I. Guyon, V. Vapnik, A training algorithm for optimal margin classifiers, in: D. Haussler (Ed.), Proceedings of the 5th Annual Workshop on Computational Learning Theory (COLT'92), ACM Press, Pittsburgh, PA, USA, 1992, pp. 144–152.
[12] R. Duda, P. Hart, D. Stork, Pattern Classification, 2nd edition, Wiley, New York, 2001.
[13] X. Zhu, Z. Ghahramani, J.D. Lafferty, Semi-supervised learning using gaussian fields and harmonic functions, in: Proceedings of the Twentieth International Conference on Machine Learning (ICML 2003), August 21–24, 2003, Washington, DC, USA, 2003, pp. 912–919.
[14] L. Zhang, W. Wu, T. Chen, N. Strobel, D. Comaniciu, L. Zhang, W. Wu, T. Chen, N. Strobel, D. Comaniciu, Robust object tracking using semi-supervised appearance dictionary learning, Pattern Recognit. Lett. 62 (Complete) (2015) 17–23.
[15] M. Zhao, T.W. Chow, Z. Zhang, B. Li, Automatic image annotation via compact graph based semi-supervised learning, Knowl.-Based Syst. 76 (2015) 148–165.
[16] C. Gong, T. Liu, D. Tao, K. Fu, E. Tu, J. Yang, Deformed graph Laplacian for semisupervised learning, IEEE Trans. Neural Netw. Learn. Syst. 99 (2015) 1.
[17] C. Wang, W. Chen, P. Yin, J. Wang, Semi-supervised clustering using incomplete prior knowledge, in: Proceedings of the 7th International Conference on Computational Science, Part I: ICCS 2007, ICCS '07, Springer-Verlag, Berlin, Heidelberg, 2007, pp. 192–195.
[18] X. Zhu, Semi-supervised learning literature survey, Technical Report 1530, Computer Sciences, University of Wisconsin-Madison, 2008.
[19] R. Ghani, Combining labeled and unlabeled data for multiclass text categorization, in: ICML '02: Proceedings of the Nineteenth International Conference on Machine Learning, Morgan Kaufmann, San Francisco, CA, 2002, pp. 187–194.
[20] R. Collobert, F. Sinz, J. Weston, L. Bottou, Large scale transductive svms, J. Mach. Learn. Res. 7 (2006) 1687–1712.
[21] H. Yang, K. Huang, I. King, M.R. Lyu, Maximum margin semi-supervised learning with irrelevant data, Neural Netw. 70 (2015) 90–102.
[22] Y.-F. Li, Z.-H. Zhou, Towards making unlabeled data never hurt, IEEE Trans. Pattern Anal. Mach. Intell. 37 (1) (2015) 175–188.
[23] X. Liu, T. Guo, L. He, X. Yang, A low-rank approximation-based transductive support tensor machine for semisupervised classification, IEEE Trans. Image Process. 24 (6) (2015) 1825–1838.
[24] C.-L. Liu, W.-H. Hsaio, C.-H. Lee, T.-H. Chang, T.-H. Kuo, Semi-supervised text classification with universum learning, IEEE Trans. Cybern. 99 (2015) 1.
[25] N. Bridle, X. Zhu, p-voltages: Laplacian regularization for semi-supervised learning on high-dimensional data, in: Workshop on Mining and Learning with Graphs (MLG2013).
[26] A. Iosifidis, A. Tefas, I. Pitas, Regularized extreme learning machine for multiview semi-supervised action recognition, Neurocomputing 145 (2014) 250–262.
[27] G.-B. Huang, Q.-Y. Zhu, C.-K. Siew, Extreme learning machine: a new learning scheme of feedforward neural networks, in: 2004 Proceedings of IEEE International Joint Conference on Neural Networks, vol. 2, 2004, pp. 985–990.
[28] M. Belkin, P. Niyogi, V. Sindhwani, Manifold regularization: a geometric framework for learning from labeled and unlabeled examples, J. Mach. Learn. Res. 7 (2006) 2399–2434.
[29] P. Niyogi, Manifold regularization and semi-supervised learning: some theoretical analyses, J. Mach. Learn. Res. 14 (2013) 1229–1250.
[30] J.D. Lafferty, L.A. Wasserman, Statistical analysis of semi-supervised regression, in: J.C. Platt, D. Koller, Y. Singer, S.T. Roweis (Eds.), NIPS, Curran Associates, Inc., 2007, pp. 801–808.
[31] P.J. Bickel, B. Li, Local polynomial regression on unknown manifolds, in: Complex Datasets and Inverse Problems, Lecture Notes–Monograph Series, vol. 54, Institute of Mathematical Statistics, Beachwood, Ohio, USA, 2007, pp. 177–186.
[32] A.B. Goldberg, X. Zhu, A. Singh, Z. Xu, R.D. Nowak, Multi-manifold semi-supervised learning, in: Proceedings of the Twelfth International Conference on Artificial Intelligence and Statistics, Clearwater Beach, Florida, 2009, pp. 169–176.
[33] G. Huang, S. Song, J. Gupta, C. Wu, Semi-supervised and unsupervised extreme learning machines, IEEE Trans. Cybern. 44 (12) (2014) 2405–2417.
[34] C. Feng, A. Sutherland, R. King, S. Muggleton, R. Henery, Comparison of machine learning classifiers to statistics and neural networks, in: Proceedings of the Third International Workshop in Artificial Intelligence and Statistics, 1993, pp. 41–52.
[35] L. Cranor, B. Lamacchia, Spam! Commun. ACM 41 (8) (1998) 74–83.
[36] Face dataset, ⟨http://www3.nd.edu/cvrl/CVRL/Data_Sets.html⟩, 2014.
[37] F. Alimoglu, E. Alpaydin, Methods of combining multiple classifiers based on different representations for pen-based handwriting recognition, in: Proceedings of the Fifth Turkish Artificial Intelligence and Artificial Neural Networks Symposium (TAINN 96), 1996.
[38] M. Tavallaee, E. Bagheri, L. Wei, A.A. Ghorbani, A detailed analysis of the kdd cup 99 data set, in: 2009 IEEE Symposium on Computational Intelligence for Security and Defense Applications, CISDA 2009, 2009, pp. 1–6.
[39] P.W. Frey, D.J. Slate, Letter recognition using Holland-style adaptive classifiers, Mach. Learn. 6 (2) (1991) 161–182.
[40] W.P. Amorim, H. Pistori, M. Pereira, M. Jacinto, Attributes reduction applied to

leather defects classification, in: 23rd SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI), 2010, pp. 353–359.
[41] C. Suzuki, J. Gomes, A. Falcão, J. Papa, S. Shimizu, Automatic segmentation and classification of human intestinal parasites from microscopy images, IEEE Trans. Biomed. Eng. 60 (3) (2013) 803–812.
[42] M. Friedman, A comparison of alternative tests of significance for the problem of m rankings, Ann. Math. Stat. 11 (1) (1940) 86–92.
[43] T.M. Huang, V. Kecman, I. Kopriva, Kernel based algorithms for mining huge data sets: supervised, semi-supervised, and unsupervised learning, in: Studies in Computational Intelligence, vol. 17, Springer, 2006.
[44] J. Demšar, Statistical comparisons of classifiers over multiple data sets, J. Mach. Learn. Res. 7 (2006) 1–30.
[45] P.T.M. Saito, C.T.N. Suzuki, J.F. Gomes, P.J. de Rezende, A.X. Falcão, Robust active learning for the diagnosis of parasites, Pattern Recognit. 48 (11) (2015) 3572–3583.

Willian P. Amorim is professor at the Federal University of Dourados Region. He holds a B.Sc. in Computer Engineering (CE) from the Dom Bosco Catholic University (2006); a M.Sc. in Computer Science (2009) from the Federal University of Mato Grosso do Sul - UFMS; and PhD student in Computer Science from the UFMS. His research areas are graphs and machine learning.

Alexandre Xavier Falcao is full professor at the Institute of Computing, University of Campinas, Campinas, SP, Brazil. He received a B.Sc. in Electrical Engineering from the Federal University of Pernambuco, Recife, PE, Brazil, in 1988. He has worked in biomedical image processing, visualization and analysis since 1991. In 1993, he received a M. Sc. in Electrical Engineering from the University of Campinas, Campinas, SP, Brazil. During 1994–1996, he worked with the Medical Image Processing Group at the Department of Radiology, University of Pennsylvania, PA, USA, on interactive image segmentation for his doctorate. He got his doctorate in Electrical Engineering from the University of Campinas in 1996. In 1997, he worked in a project for Globo TV at a research center, CPqD-TELEBRAS in Campinas, developing methods for video quality assessment. His experience as professor of Computer Science and Engineering started in 1998 at the University of Campinas. His main research interests include image/video processing, visualization, and analysis; graph algorithms and dynamic programming; image annotation, organization, and retrieval; machine learning and pattern recognition; and image analysis applications in Biology, Medicine, Biometrics, Geology, and Agriculture.

J.P. Papa obtained his B.Sc. in Information Systems from the São Paulo State University (UNESP - Univ Estadual Paulista), SP, Brazil. In 2005, he received his M.Sc. in Computer Science from the Federal University of So Carlos, SP, Brazil. In 2008, he received his Ph.D. in Computer Science from the University of Campinas, SP, Brazil. During 2008–2009, he had worked as post-doctorate researcher at the same institute, and during 2014–2015 he had worked as a visiting scholar at Harvard University. He has been Professor at the Computer Science Department, So Paulo State University, since 2009, and his research interests include machine learning, pattern recognition and image processing.

Marcelo H. Carvalho is a Ph.D. in Computer Science from the Universidade Estadual de Campinas (1996), in Brazil. His research areas are graphs and algorithms.