Information Sciences 465 (2018) 86–104

Contents lists available at ScienceDirect

Information Sciences

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

Multi-label semi-supervised classification through optimum-path forest

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

[a] Institute of Computing, Federal University of Mato Grosso do Sul, Campo Grande, MS, Brazil
[b] Institute of Computing, University of Campinas, Campinas, SP, Brazil
[c] Department of Computing, São Paulo State University, Bauru, SP, Brazil

ARTICLE INFO

Article history:
Received 23 June 2016
Revised 29 June 2018
Accepted 30 June 2018
Available online 2 July 2018

Keywords:
Semi-supervised learning
Multi-label assignment
Optimum-path forest classifiers

ABSTRACT

Multi-label classification consists of assigning one or multiple classes to each sample in a given dataset. However, the project of a multi-label classifier is usually limited to a small number of supervised samples as compared to the number of all possible label combinations. This scenario favors semi-supervised learning methods, which can cope with the absence of supervised samples by adding unsupervised ones to the training set. Recently, we proposed a semi-supervised learning method based on *optimum connectivity* for single-label classification. In this work, we extend it for multi-label classification with considerable effectiveness gain. After a single-label data transformation, the method propagates labels from supervised to unsupervised samples, as in the original approach, by assuming that samples from the same class are more closely connected through sequences of nearby samples than samples from distinct classes. Given that the procedure is more reliable in high-density regions of the feature space, an additional step repropagates labels from the maxima of a probability density function to correct possible labeling errors from the previous step. Finally, the data transformation is reversed to obtain multiple labels per sample. The new approach is experimentally validated on several datasets in comparison with state-of-the-art methods.

1. Introduction

Multi-label assignment is required in several scenarios, such as text categorization and computer-aided medical diagnosis systems. In the former, a newspaper article may be categorized as belonging to religion and arts, while, in the latter, a patient may be affected by multiple diseases simultaneously. Therefore, a sample may be assigned to one or multiple classes in such scenarios. Existing methods either reduce the problem into several single-label classification problems, or create one meta-class for each possible label combination, being divided into two main categories [34]: *problem transformation* and *algorithm adaptation* strategies, respectively. In both cases, the samples are relabeled according to a different strategy in order to use one or multiple well-known classification models. Algorithm adaptation methods modify supervised learning approaches, such as *k*-nearest neighbors [33], decision trees [10], and artificial neural networks [32] aiming at identifying the meta-classes of samples based on ranking and statistical information. Similarly, there are solutions that exploit label correlation information [20], methods that use this correlation to perform feature selection [8,9], and works that investigate

* Corresponding author.
E-mail addresses: willianamorim@ufgd.edu.br (W.P. Amorim), afalcao@ic.unicamp.br (A.X. Falcão), papa@fc.unesp.br (J.P. Papa).

https://doi.org/10.1016/j.ins.2018.06.067
0020-0255/

label dependencies [31]. In short, each category brings benefits but also has disadvantages that are desired to be known beforehand.

Once single-label datasets are created by problem transformation from multi-label samples, for instance, a supervised classifier can be trained to assign single labels to new samples. These samples may then be reassigned to multiple classes by the reverse data transformation process. However, the design of a multi-label classifier is often limited to a small number of supervised samples (those labeled by experts) as compared to the number of all possible label combinations, thus favoring semi-supervised learning methods. In the case of algorithm adaptation strategies, one shall use a self-training approach [21,23,30].

The most interesting semi-supervised learning methods explore the distribution of supervised and unsupervised training samples in the feature space for label propagation purposes [1,2,4,5,17–19,28]. Others use path-based similarity to capture the structure of the data and maximize the separability among classes [7,35]. This process can be repeated a few times, such that a final classifier is created from the most confident samples of the fully labeled training set.

The method presented in this work is based on the Optimum-Path Forest (OPF) framework, initially proposed to the design of image processing operators [15] and subsequently extended to clustering [29] and supervised classification [25–27]. In OPF, training samples (supervised and/or unsupervised) are the nodes of a graph defined by a given *adjacency relation*, and the classifier is an optimum-path forest computed over the graph for a given *connectivity function* — i.e., a function that assigns a value to any path in the graph, including the trivial ones formed by single nodes. The roots of the forest are derived from the minima (maxima) of a connectivity map, that is minimized (maximized) by computing optimum paths with the terminus at each node of the graph. The design of a classifier can be obtained by executing the OPF algorithm one or multiple times for different input graphs and connectivity functions.

In [1], we proposed a semi-supervised learning method for the single-label assignment problem, which propagates labels from supervised to unsupervised training samples by optimum connectivity. This method was recently improved in accuracy and efficiency through a new algorithm, named OPFSEMI$_{mst}$ [2]. One can simply use OPFSEMI$_{mst}$ to directly assign single labels to new samples and reversely transform those labels into the multiple classes per sample. In OPFSEMI$_{mst}$, the classifier is an optimum-path forest computed over the topology of a minimum-spanning tree. The roots of the forest are the supervised samples and unsupervised samples are labeled by their most closely connected root — i.e., the one that offers a path whose maximum arc-length is minimum. This label propagation process assumes that samples from the same class are more closely connected than samples from distinct classes. In order to classify a new sample, all training examples are connected to the new sample and extended paths are evaluated to assign the label of its most closely connected root. This classifier has shown to be robust to a certain amount of label propagation errors in the training set for the single-label assignment problem [2]. However, in the multi-label assignment problem, classification errors increase when wrong labels are reversely transformed into multiple classes.

Essentially, the multi-label assignment problem requires a more conservative classifier than OPFSEMI$_{mst}$ to better deal with possible overlaps among classes. In this work, we observe that the label propagation errors of OPFSEMI$_{mst}$ are concentrated in feature space regions of lower probability density values — i.e., in the frontier among domes of a probability density function (pdf) computed over the training set. Therefore, we propose to repropagate labels from the maxima of the pdf to the remaining samples in the training set. This process is accomplished by a variant of the OPF clustering algorithm [29], in which the arcs of the graph are formed by connecting each training sample to its k-nearest nodes in the feature space. The connectivity function assigns the minimum density value along a path and the connectivity map must be maximized. In the resulting optimum-path forest (final classifier), each training sample belongs to one of the rooted trees and is relabeled by the class of the corresponding root node.

The new semi-supervised classifier is named OPFSEMI$_{mst+knn}$. For classification, the training samples closer to their roots have higher priority to assign labels to new samples. Since mislabeled samples tend to be in the frontier among classes, where the pdf values are lower, OPFSEMI$_{mst+knn}$ can significantly outperform OPFSEMI$_{mst}$ in multi-label classification.

In [29], the best value of k results from the optimum-path forest that produces the minimum normalized cut in the k-nn graph. In [24], the authors use another variant of the OPF algorithm over a k-nn graph for supervised learning. They select the value of k that minimizes the label propagation errors in the training set. In this case, the criterion leads to low values of k (i.e., too many clusters), which also implies a poor estimation of the pdf. In this work, we estimate k such that the number of labeling disagreements with OPFSEMI$_{mst}$ is less or equal to the number of the label propagation errors on half of the supervised samples. The criterion tends to obtain a reasonable pdf estimation with higher values of k, and it also reduces the number of mislabeled samples in the lower density regions of the pdf.

In order to show the robustness of OPFSEMI$_{mst+knn}$, we assess its performance in multi-label classification problems against three semi-supervised learning approaches adapted to four problem transformation strategies: its counterpart OPFSEMI$_{mst}$ [2], LapSVM [4] (manifold regularization), and Transductive Support Vector Machines (TSVM) [19], being the last two well-established methods. Additionally, we compare OPFSEMI$_{mst+knn}$ against two algorithm adaptation techniques: multi-label kNN (MLkNN) [33] and Back-Propagation Multi-Label Learning (BPMLL) [32]. In short, the main contributions of this work are twofold: (i) to present an extension of OPFSEMI$_{mst}$, namely OPFSEMI$_{mst+knn}$, which aims at reducing label propagation errors to unsupervised samples, and (ii) to propose a semi-supervised multi-label OPF classifier based on OPFSEMI$_{mst+knn}$.

The remainder of this paper is organized as follows. Section 2 explains algorithm adaptation and problem transformation strategies, and Section 3 provides the theoretical background about the OPF framework. Section 4 introduces the proposed

OPFSEMI$_{mst+knn}$, and the experimental results are presented in Section 5. Finally, Section 6 states conclusion and future work.

2. Methods for multi-label classification

In this section, we present a brief review concerning the two categories of multi-label learning, (i) algorithm adaptation and (ii) problem transformation, and discuss how to propagate labels from supervised to unsupervised samples in each strategy.

2.1. Algorithm adaptation strategies

The focus of the algorithm adaptation approaches concerns modifying existing algorithms so that they can deal directly with the multi-label samples. Since the number of works in this topic has increased considerably, we will focus on the techniques used for comparison purposes in the experimental section. The first one is the multi-label *k*NN [33], which evaluates, for each unseen example, its *k*-nearest neighbors in the training set. After that, based on statistical information gained from the label sets of these neighboring samples, i.e. the number of neighboring instances belonging to each possible class, a maximum *a posteriori* (MAP) principle is employed to determine the label set for that unseen sample. Another traditional method in the area is the Back-Propagation Multi-Label Learning [32]. In principle, BPMLL is derived from the popular Back-propagation algorithm by employing a novel error function that encodes the characteristics of multi-label learning — i.e., the labels belonging to an example should be ranked higher than those not belonging to that sample.

A common and simple technique used for semi-supervised learning based on algorithm adaptation strategy is known as "self-training", in which the classifier uses its own predictions to teach itself. In this approach, an apprentice classifier is first trained with a small number of supervised samples (an "initial" training set). Next, the classifier assigns labels to the unsupervised samples and the most confidently labeled ones are used to augment the training set for retraining. This procedure may repeat until all unsupervised examples have been labeled and moved to the training set. In this paper, we use the self-training approach for the evaluation of both ML*k*NN and MPMLL methods.

2.2. Problem transformation strategies

In order to understand the difference among problem transformation strategies, let $\mathcal{Z}$ be a d-dimensional feature space and $\mathcal{Y} = \{y_1, y_2, \ldots, y_\mathcal{L}\}$ be a label space with $\mathcal{L}$ possible class labels. Each sample $\mathbf{s}_i \in \mathcal{Z}$ can be assigned to a label set $\mathcal{Y}_i \subseteq \mathcal{Y}$ by specifying the binary values $y_i^k \in \{0, 1\}$, $1 \leq k \leq \mathcal{L}$, where $y_i^k = 1$ denotes the sample $\mathbf{s}_i$ belongs to class k, and $y_k^i = 0$ stands for the opposite situation[1] A training set $\mathcal{Z}$ with l supervised samples and u unsupervised samples is then defined by $\mathcal{Z} = \mathcal{Z}^l \cup \mathcal{Z}^u$, where $\mathcal{Z}^l = \{(\mathbf{s}_1, \mathcal{Y}_1), \ldots, (\mathbf{s}_l, \mathcal{Y}_l)\}$ and $\mathcal{Z}^u = \{\mathbf{s}_{l+1}, \ldots, \mathbf{s}_{l+u}\}$ stand for the supervised and unsupervised sets of samples, respectively. In this case, the learning problem aims at finding from $\mathcal{Z}$ a family of real-valued functions $f_i : \mathcal{Z} \times \mathcal{Y} \to \Re$, $i = 1, 2, \ldots, \mathcal{L}$ such that $f_i(\mathbf{s}_i, \mathcal{Y}_i)$ is the confidence that $\mathcal{Y}_i$ is the true label set of $\mathbf{s}_i$ (i.e., the label assigned to the sample by the expert supervision).

Binary Relevance (BR) decomposes the original multi-label $\mathcal{Z}^l$ dataset into $\mathcal{L}$ single-label datasets $\mathcal{Z}^l[k] = \{(\mathbf{s}_1, y_1^k), \ldots, (\mathbf{s}_l, y_l^k)\}$, as well as $\mathcal{L}$ training-independent binary classifiers f_k from $\mathcal{Z}^l[k]$ in order to predict $y_i^k \in \{0, 1\}$, $k = 1, 2, \ldots, \mathcal{L}$, where $y_i[k]$ denotes the label of sample i with respect to class k. For instance, in a multi-label problem with $\mathcal{L} = 3$ classes, $\mathcal{Z}^l[1]$ stands for the dataset in which samples that belong to class 1 are marked as positive, and the remaining ones (i.e., those from classes 2 and 3) are marked as negative samples. Label Powerset (LP) considers each possible label combination from $\mathcal{L}$ as a single label, handling the problem with multiple classes. LP transforms $\mathcal{Z}^l$ into a new dataset $\hat{\mathcal{Z}}^l = \{(\mathbf{s}_1, c_1), \ldots, (\mathbf{s}_l, c_l)\}$, where $c_i = g(\mathcal{Y}_i)$ and $g : \{0, 1\}^\mathcal{L} \to \{1, \ldots, 2^\mathcal{L}\}$, $i = 1, 2, \ldots, l$, stands for a function that maps each single label combination to a new label representation. A multi-label classifier is then trained from new dataset to predict the label of each new sample.

Classifier Chain (CC) and Hierarchy of Multi-Label Classifiers (HOMER) are optimized extensions of BR and LP, respectively. Classifier Chain performs the mapping into $\mathcal{L}$ binary datasets as BR, but it also extends the feature space for each $\mathcal{Z}^l[k]$ by adding the 0/1 labels from the previous $\mathcal{Z}^l[k-1]$. That is, $\mathcal{Z}^l[k] = \{(\mathbf{s}'_1, y_1^k), \ldots, (\mathbf{s}'_l, y_l^k)\}$ and $\mathbf{s}'_i = (\mathbf{s}_i \cdot y_i^{k-1})$, where $\cdot$ stands for the concatenation operator[2], $i = 1, 2, \ldots, l$. HOMER transforms a multi-label classifier into a hierarchy of simpler multi-label classifiers, such that the classifier of a child node deals with a smaller set of labels than the classifier of the parent node. The root node deals with $\mathcal{L}$ labels, which are grouped into $k \leq \mathcal{L}$ disjoint children nodes. The grouping process repeats for each node in a depth-first fashion until the nodes become leaves with $\mathcal{L}$ single-label classifiers.

Therefore, one can apply a data transformation function $\mathcal{T}(\mathcal{Z}^l)$, propagate labels from $\mathcal{T}(\mathcal{Z}^l)$ to $\mathcal{Z}^u$, and revert the process $\mathcal{T}^{-1}(\mathcal{Z}^u)$ for multi-label assignment. By projecting a semi-supervised classifier from $\mathcal{T}(\mathcal{Z}^l) \cup \mathcal{Z}^u$ and using it to assign labels to new samples, the inverse $\mathcal{T}^{-1}$ discovers their multiple classes. Notice we use the transformation strategies for the evaluation of OPFSEMI$_{mst+knn}$, OPFSEMI$_{mst}$, LapSVM and TSVM methods.

[1] Notice $\mathcal{Y}_i = \{y_i^1, y_i^2, \ldots, y_i^\mathcal{L}\}$, $i = 1, 2, \ldots, l$.

[2] Notice when $k = 1$, the feature vector is not extended.

3. Optimum-path forest framework

In this section, we present different ways to define graphs from training samples and connectivity functions, such that the resulting optimum-path forest is a classifier built upon the training data.

3.1. Training

Let $\mathcal{Z} = \mathcal{Z}_1 \cup \mathcal{Z}_2$ be a dataset such that $\mathcal{Z}_1$ and $\mathcal{Z}_2$ stand for the training and testing sets, respectively. Additionally, $\mathcal{Z}_1 = \mathcal{Z}_1^l \cup \mathcal{Z}_1^u$ consists of a supervised $\mathcal{Z}_1^l$ and unsupervised $\mathcal{Z}_1^u$ subsets of samples. Let also $\lambda(\mathbf{s}) \in \{1, 2, \ldots, \mathcal{L}\}$ be the true label of each sample $\mathbf{s} \in \mathcal{Z}$ and $d(\mathbf{s}, \mathbf{t}) \geq 0$ be a distance function between the feature vectors of samples $\mathbf{s}, \mathbf{t} \in \mathcal{Z}$.

In the optimum-path forest framework, one can create a supervised classifier from $\mathcal{Z}_1^l$ and an unsupervised classifier from $\mathcal{Z}_1^u$, or a semi-supervised classifier from $\mathcal{Z}_1$ by defining an adjacency relation $\mathcal{A}$ and a connectivity function f. The adjacency relation $\mathcal{A}$ tells how training samples are connected in the feature space, thus forming a weighted graph $(\mathcal{N}, \mathcal{A}, d)$ where each pair $(\mathbf{s}, \mathbf{t}) \in \mathcal{A}$ is an arc weighted by the distance $d(\mathbf{s}, \mathbf{t})$ between its corresponding nodes $\mathbf{s}, \mathbf{t} \in \mathcal{N} \subseteq \mathcal{Z}_1$. We also use $\mathbf{t} \in \mathcal{A}(\mathbf{s})$ to indicate an element from the set of nodes adjacent to $\mathbf{s}$. The connectivity function f assigns a value $f(\pi_t)$ to any sequence $\pi_t = \langle \mathbf{s}_1, \mathbf{s}_2, \ldots, \mathbf{s}_n = \mathbf{t} \rangle$ of nodes, $(\mathbf{s}_i, \mathbf{s}_{i+1}) \in \mathcal{A}$, $i = 1, 2, \ldots, n-1$, with terminus at node $\mathbf{t}$ in the graph, including trivial paths $\pi_t = \langle t \rangle$. A trivial path value indicates the cost of starting a path from $\mathbf{t}$, while the value of a path with terminus $\mathbf{s}_n = \mathbf{t}$ indicates the cost of conquering $\mathbf{t}$ by a path that starts in some node $\mathbf{s}_1 \neq t$.

For a non-decreasing connectivity function f, the algorithm starts from trivial paths, such that the minima of the initial cost map $C_0(\mathbf{t}) = f(\langle \mathbf{t} \rangle)$, $\forall \mathbf{t} \in \mathcal{N}$, compete among themselves by offering paths of lower costs to the remaining nodes. This relaxation process creates a connectivity map $C(\mathbf{t}) = \min_{\forall \pi_t \in \Pi_t} \{f(\pi_t)\} \leq C_0(\mathbf{t})$, where Π_t is the set of all possible paths with terminus at $\mathbf{t}$ and an *optimum-path forest* P rooted at the winner minima. The forest is an acyclic predecessor map P that assigns to each node $\mathbf{t}$ its predecessor $P(\mathbf{t})$ in the optimum path, or a distinct marker $P(\mathbf{t}) = nil \notin \mathcal{N}$ when the node is a root of the cost map.

3.2. Classification

The optimum-path forest P generated in the previous phase can be used for classification purposes. Let $\mathbf{v} \in Z_2$ be a sample to be classified. The classification process evaluates the cost of extending a path π_s by a segment $\langle \mathbf{s}, \mathbf{v} \rangle$, as follows:

$$C(\mathbf{v}) = \min_{\forall \mathbf{s} \in \mathcal{N}} \{f(\pi_s \cdot \langle \mathbf{s}, \mathbf{v} \rangle)\}. \tag{1}$$

Let $\mathbf{s}^* \in \mathcal{N}$ be the node that satisfies the above equation. Therefore, the classifier assigns the label of $\mathbf{s}^*$ as the new class of sample $\mathbf{v}$.

The main OPF algorithm is a variant of Dijkstra's algorithm [12] for multiple sources and more general connectivity functions [15]. Distinct operations may require other simple variants. For instance, a similar process can be defined for non-increasing connectivity functions and roots at the maxima of the final connectivity map. The next section exemplifies its application to the design of the proposed semi-supervised classifier — OPFSEMI$_{mst+knn}$.

4. The semi-supervised classifier — OPFSEMI$_{mst+knn}$

In this section, we present the semi-supervised learning method based on optimum-path forest, OPFSEMI$_{mst}$ [2], and its proposed extension OPFSEMI$_{mst+knn}$, which is more suitable for the multi-label assignment problem.

First, in Section 4.1, we explain the rationale about OPFSEMI$_{mst+knn}$. The semi-supervised learning process of OPFSEMI$_{mst+knn}$ consists of three executions of the OPF algorithm for different choices of adjacency relation and connectivity functions. The first execution closely connects supervised and unsupervised training samples in the feature space (Section 4.2) – Algorithm 1, and the second execution propagates labels from the supervised samples to the unsupervised ones (Section 4.3) – Algorithm 2. The third execution estimates a probability density function, propagates labels from its maxima, and creates a final Optimum-Path Forest classifier (Section 4.4) – Algorithm 3. Finally, Section 4.5 describes how such classifier assigns labels to new samples.

4.1. What is the rationale about OPFSEMI$_{mst+knn}$?

In regard to overlapping among classes, we showed in [2] the performance of OPFSEMI$_{mst}$ is better when the expert annotates the samples in $\mathcal{Z}_1^l$ at the overlapped regions. However, by assuming a random choice of samples for $\mathcal{Z}_1$, they are more likely to fall at the higher density regions of the feature space, wherein the center of the classes usually appear. In the multi-label assignment problem, class overlap seems to increase when multiple classes are transformed into single labels. This increases the label propagation errors of OPFSEMI$_{mst}$, and when the single labels are reversely transformed into multiple classes, the final performance of OPFSEMI$_{mst}$ is worse than the one observed for the single-label assignment problem.

On the other hand, the errors in label propagation tend to concentrate in the lower density regions. Therefore, samples at the maxima of the pdf (center of the classes) are more reliable to repropagate labels, which justifies our choice for OPFSEMI$_{mst+knn}$. This scenario is illustrated in Fig. 1. Fig. 1a shows two overlapped classes (2D feature space) and the random

INPUT: A complete and weighted graph $(\mathcal{Z}_1, \mathcal{A}, d)$.
OUTPUT: A minimum-spanning tree $(\mathcal{Z}_1, \mathcal{B}, d)$.
AUXILIARY: Priority queue Q, variable cst, connectivity map C, predecessor map P, and $color(s)$, which is a function that outputs *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 B ← ∅.
2.  For each t ∈ Z1 do
3.      Set C(t) ← +∞ and color(t) ← white.
4.  Select any node a ∈ Z1,
5.  Set C(a) ← 0, P(a) ← nil, color(a) ← gray, and insert a in Q.
6.  While Q is not empty, do
7.      Remove from Q a sample s such that,
8.      C(s) is minimum and color(s) ← black.
9.      If P(s) ≠ nil then B ← B ∪ {(s, P(s)), (P(s), s)}.
10.     For each t ∈ A(s) do
11.         If color(t) ≠ black, then
12.             Set cst ← d(s, t).
13.             If cst < C(t), then
14.                 Set P(t) ← s and C(t) ← cst.
15.                 If color(t) = gray
16.                 then Update position of t in Q
17.                 Else Insert t in Q and
18.                     Set color(t) ← gray.
19. Return a minimum-spanning tree (Z1, B, d).
```

Algorithm 1. The OPF algorithm for f_{mst} on $(\mathcal{Z}_1, \mathcal{A}, d)$.

INPUT: A λ-labeled graph $(\mathcal{Z}_1, \mathcal{B}, d)$.
OUTPUT: Maps of the optimum-path forest with attributes $[P_1, C_1, L_1]$.
AUXILIARY: Priority queue Q, cost variable cst, and function $color(\mathbf{s})$ that outputs *white* when $\mathbf{s}$ has never been inserted in Q; *gray* when $\mathbf{s} \in Q$; and *black* when $\mathbf{s}$ has been removed from Q.

```
1.  For each node t ∈ Z1, do
2.      set C1(t) ← +∞, color(t) ← white and P1(t) ← nil.
3.      If t ∈ Z1^l, then
4.          set C1(t) ← 0, color(t) ← gray.
5.          L1(t) ← λ(t).
6.          insert t in Q.
7.  While Q is not empty, do
8.      Remove from Q a sample s such that,
9.      C1(s) is minimum and set color(s) ← black.
10.     For each t ∈ B(s) do
11.         If color(t) ≠ black, then
12.             Set cst ← max{C1(s), d(s, t)}.
13.             If cst < C1(t), then
14.                 Set P1(t) ← s, L1(t) ← L1(s), and
15.                 C1(t) ← cst.
16.                 If color(t) = gray
17.                 then update position of t in Q
18.                 Else insert t in Q and
19.                     set color(t) ← gray.
20. Return [P1, C1, L1].
```

Algorithm 2. The OPF algorithm for $f_{\max}$ on $(\mathcal{Z}_1, \mathcal{B}, d)$.

INPUT: Graph $(\mathcal{Z}_1, \mathcal{A}_k, d)$ and the previous label map L_1.
OUTPUT: Optimum-path forest and its attributes $[P_2, C_2, L_2]$ and the sorted list $\mathcal{Z}'_1$.
AUXILIARY: Priority queue Q and connectivity variable val.

1. **For each** *node* $\boldsymbol{t} \in \mathcal{Z}_1$, **do**
2. $\quad$ *set* $C_2(\boldsymbol{t}) \leftarrow \rho(\boldsymbol{t}) - \delta$, $P_2(\boldsymbol{t}) \leftarrow nil$, *and insert* $\boldsymbol{t}$ *in* Q.
3. **While** Q *is not empty,* **do**
4. $\quad$ *Remove from* Q *a sample* $\boldsymbol{s}$ *such that,*
5. $\quad$ $C_2(\boldsymbol{s})$ *is maximum.*
6. $\quad$ *Insert* $\boldsymbol{s}$ *in* $\mathcal{Z}'_1$.
7. $\quad$ **If** $P_2(\boldsymbol{s}) = nil$, **then** $C_2(\boldsymbol{s}) \leftarrow \rho(\boldsymbol{s})$ *and* $L_2(\boldsymbol{s}) \leftarrow L_1(\boldsymbol{s})$
8. $\quad$ **For each** $t \in \mathcal{A}_k(\boldsymbol{s})$ **do**
9. $\quad\quad$ **If** $C_2(\boldsymbol{t}) < C_2(\boldsymbol{s})$, **then**
10. $\quad\quad\quad$ *Set* $val \leftarrow \min\{C_2(s), \rho(t)\}$.
11. $\quad\quad\quad$ **If** $val > C_2(\boldsymbol{t})$, **then**
12. $\quad\quad\quad\quad$ *Remove* t *from* Q.
13. $\quad\quad\quad\quad$ *Set* $P_2(\boldsymbol{t}) \leftarrow \boldsymbol{s}$, $L_2(\boldsymbol{t}) \leftarrow L_2(\boldsymbol{s})$, *and*
14. $\quad\quad\quad\quad$ $C_2(\boldsymbol{t}) \leftarrow val$, *and insert* $\boldsymbol{t}$ *in* Q.
15. **Return** $[P_2, C_2, L_2]$ *and* $\mathcal{Z}'_1$.

Algorithm 3. The OPF algorithm for $f_{\min}$ on $(\mathcal{Z}_1, \mathcal{A}_k, d)$.

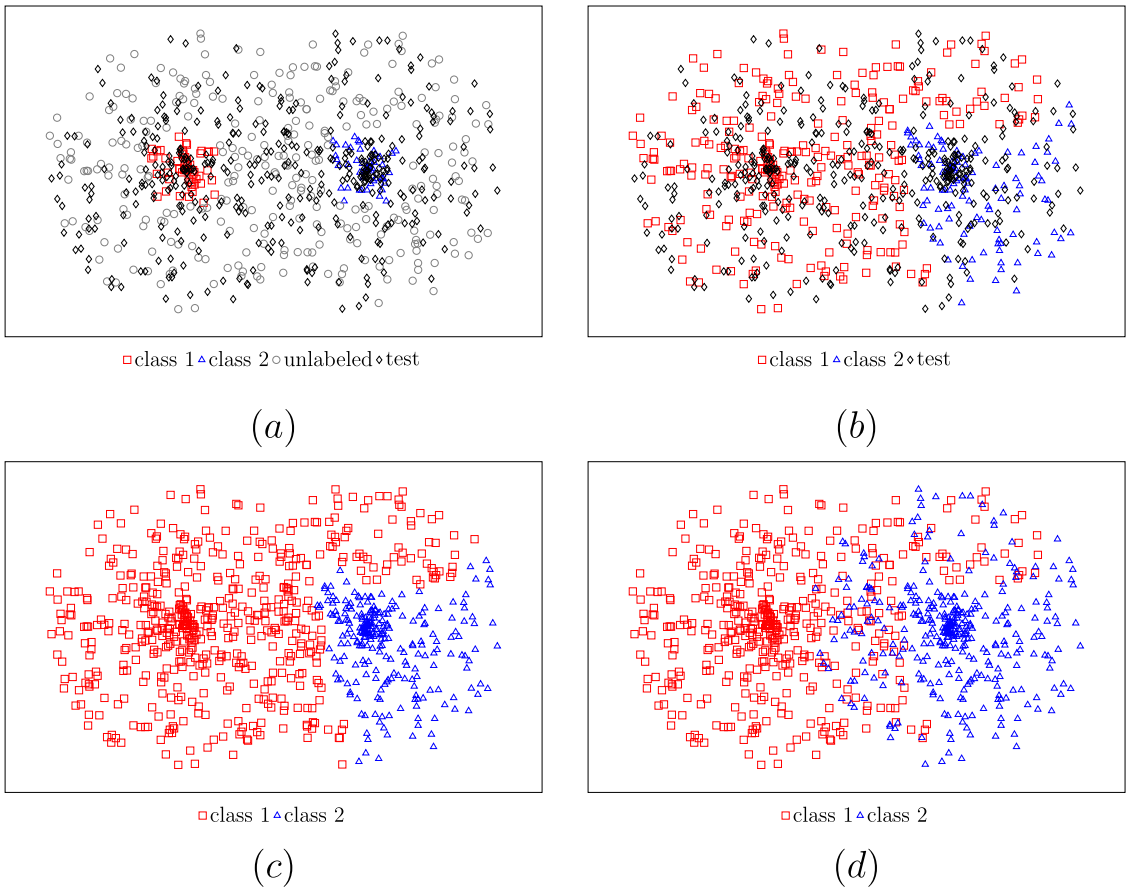

Fig. 1. (a) A dataset with two overlapped classes (2D feature space) with supervised, unsupervised, and test samples. (b) Result of label propagation by OPFSEMI$_{mst}$, and classification results considering (c) OPFSEMI$_{mst}$ and (d) OPFSEMI$_{mst+knn}$.

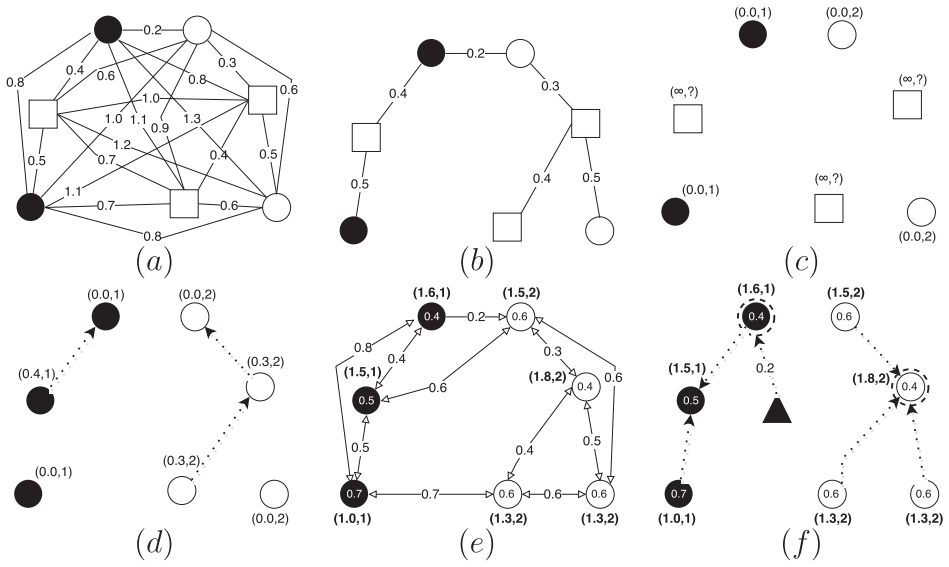

Fig. 2. (a) Complete and weighted graph for a simple training set (• supervised samples of class 1, ◦ supervised samples of class 2 and □ unsupervised samples). (b) A minimum-spanning tree from (a). (c) A trivial connectivity map for optimum-path forest computation using $f_{\max}$ on (b) and $\mathcal{S} = \mathcal{Z}_1^l$. (d) The resulting optimum-path forest. (e) Graph connected by means of a k-nearest neighbors adjacency relation (e.g. $k = 3$). The arrows point ($-\triangleright$) to the nearest neighbors. The entries (x,y) over the nodes are, respectively, their density value and label. The value within the node represents the radius (the median value). (f) Optimum-path forest computation using $f_{\min}$, where a test sample (triangle) is added in the graph, and classification result such that $L_2(\mathbf{v}) = L_2(\mathbf{s}^*)$. Nodes with dashed lines are the maxima that represent the classes.

choice of supervised, unsupervised, and test samples. The results of label propagation and classification by OPFSEMI$_{mst}$ are shown in Fig. 1b and c, respectively. When labels are repropagated from the maxima of the pdf, the result of classification by OPFSEMI$_{mst+knn}$ reduces the errors (Fig. 1d), thus resulting in higher accuracies when single labels are reversely transformed into multiple classes.

4.2. Closely connecting supervised and unsupervised samples

For a given training set $\mathcal{Z}_1 = \mathcal{Z}_1^l \cup \mathcal{Z}_1^u$ with supervised and unsupervised samples, OPFSEMI$_{mst+knn}$ first defines a complete and weighted graph $(\mathcal{Z}_1, \mathcal{A}, d)$ whose nodes are the training samples and arcs connect all pairs of samples (i.e., $\mathcal{A} = \mathcal{Z}_1 \times \mathcal{Z}_1$). The idea is to closely connect the supervised and unsupervised samples into a simpler representation — a Minimum Spanning Tree (MST): an acyclic, weighted, and connected graph $(\mathcal{Z}_1, \mathcal{B}, d)$, $\mathcal{B} \subset \mathcal{A}$, where $\sum_{\forall(\mathbf{s},\mathbf{t})\in\mathcal{B}} d(\mathbf{s},\mathbf{t})$ is minimum. This representation assumes that samples from a same class are more closely connected through sequences of nearby samples than samples from distinct classes, which makes it more efficient and suitable for label propagation.

The MST of $(\mathcal{Z}_1, \mathcal{A}, d)$ can be computed by the OPF algorithm for a connectivity function f_{mst}, as follows:

$$f_{mst}(\langle \mathbf{t} \rangle) = \begin{cases} 0 & \text{for an arbitrary node } \mathbf{t} \in \mathcal{Z}_1, \\ +\infty & \text{otherwise,} \end{cases} \tag{2}$$

$$f_{mst}(\pi_s \cdot \langle \mathbf{s}, \mathbf{t} \rangle) = d(\mathbf{s}, \mathbf{t}). \tag{3}$$

Roughly speaking, Eq. (2) makes the OPF algorithm to start the path propagation process from some arbitrary node $\mathbf{t} \in \mathcal{Z}_1$. During the process, Eq. (3) computes the cost of extending a path π_s by an arc (**s, t**) as the distance between the feature vectors of its nodes. Function f_{mst} cannot guarantee an optimum connectivity map. Instead, it degenerates the optimum-path tree rooted at **t** into an MST, making the OPF algorithm equivalent to the Prim's algorithm [12] (Fig. 2a and b). Notice the symbol $+\infty$ (infinity) represents a real number strictly higher than any other generated in the process.

Algorithm 1 describes the OPF algorithm modified for f_{mst} on $(\mathcal{Z}_1, \mathcal{A}, d)$. Lines 1–5 initialize the connectivity map and select any node to start the MST computation. The main loop (Lines 6–19) computes the MST in a non-decreasing order of minimum total cost. For each neighbor in Line 10, a path of minimum cost $C(\mathbf{s})$ is obtained in P, as well as the current arc that is being evaluated is added to the minimum spanning tree. The algorithm then outputs an MST $(\mathcal{Z}_1, \mathcal{B}, d)$ with supervised and unsupervised samples connected into a single graph component.

4.3. Propagating labels to unsupervised samples

In order to propagate labels from $\mathcal{Z}_1^l$ to $\mathcal{Z}_1^u$, we use the MST generated in the previous section $(\mathcal{Z}_1, \mathcal{B}, d)$ together with the connectivity function $f_{\max}$, which is defined as follows:

$$f_{\max}(\langle \mathbf{t} \rangle) = \begin{cases} 0 & \text{if } \mathbf{t} \in \mathcal{Z}_1^l, \\ +\infty & \text{otherwise,} \end{cases}$$
$$f_{\max}(\pi_s \cdot \langle \mathbf{s}, \mathbf{t} \rangle) = \max\{f_{\max}(\pi_s), d(\mathbf{s}, \mathbf{t})\}. \tag{4}$$

Function $f_{\max}$ essentially forces the roots of the forest to be the supervised samples, and assigns to extended paths the maximum arc-weight along them as the connectivity value. The minimization of the connectivity map, as computed by Algorithm 2, outputs an optimum-path forest with cost and label attributes $[P_1, C_1, L_1]$ (Fig. 2c and d).

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 $|\mathcal{B}| \ll |\mathcal{Z}_1| \log |\mathcal{Z}_1|$. A label propagation error occurs when $L_1(\mathbf{t}) \neq \lambda(\mathbf{t})$ for $\mathbf{t} \in \mathcal{Z}_1^u$. At this point, $[P_1, C_1, L_1]$ concerns the OPFSEMI$_{mst}$ classifier. It assigns labels to new samples $\mathbf{v} \in \mathcal{Z}_2$ by evaluating the following equation:

$$C_1(\mathbf{v}) = \min_{\forall \mathbf{s} \in \mathcal{Z}_1} \{\max\{C_1(\mathbf{s}), d(\mathbf{s}, \mathbf{v})\}\}. \tag{5}$$

Notice the above formula is quite similar to Eq. (1), but now considering a different nomenclature.

Let $\mathbf{s}^* \in \mathcal{Z}_1$ be the node that satisfies Eq. (5), then the classifier assigns $L_1(\mathbf{v}) \leftarrow L_1(s^*)$ and a classification error occurs when $\lambda(\mathbf{v}) \neq L_1(\mathbf{s}^*)$. Eq. (5) connects all training samples with each new sample for classification purposes, without performing any analysis about the probability of the training samples be correctly labeled. When inversely transforming multiple single-label datasets into a multi-label dataset, this becomes an issue that asks for a more conservative approach. We circumvent the problem by adding a last step to this training process, as follows.

4.4. Generating the final classifier

We may think of the training samples in $\mathcal{Z}_1$ as points in the feature space, which can be observed from different perspectives. From an infinity distance, all points are sought as a single cluster. As we approach, the scale in which the points are observed changes, and multiple clusters may appear. The determination of the best scale that solves a clustering problem is an application-dependent task.

The OPF clustering algorithm [29] follows the above principle by defining the scale as an integer $1 \leq k < |\mathcal{Z}_1|$. For a given scale k, the training samples in $\mathcal{Z}_1$ are the nodes of a graph $(\mathcal{Z}_1, \mathcal{A}_k, d)$ that connects the k-nearest neighbors in the feature space to form directed arcs in $\mathcal{A}_k$. A probability density function (pdf) ρ is estimated to weight nodes as well:

$$\rho(\mathbf{s}) = \frac{1}{\sqrt{2\pi\sigma^2 k}} \sum_{\forall t \in \mathcal{A}_k(\mathbf{s})} exp\left(\frac{-d^2(\mathbf{s}, \mathbf{t})}{2\sigma^2}\right), \tag{6}$$

where $\sigma = \frac{d_f}{3}$ and $d_f = \max_{\forall (\mathbf{s},\mathbf{t}) \in \mathcal{A}_k}\{d(\mathbf{s}, \mathbf{t})\}$. The OPF algorithm can be executed multiple times to search for the best value of $k \in [k_{\min}, k_{\max}]$, $k_{\min} \geq 1$ and $k_{\max} < |\mathcal{Z}_1|$. Each time, labels from the maxima of the pdf (roots) are propagated to the nodes in the optimum-path tree (dome or cluster) of each maximum, thus producing a cut in the graph. In [29], the best value of k is estimated as the one that produces the minimum normalized cut, and in [13] such value is computed by means of meta-heuristic-based optimization.

In our case, we propose a different approach, as explained later. Assuming, for the time being, a given best value of k, the proposed variant of the OPF algorithm (Algorithm 3) requires a connectivity function $f_{\min}$ [15,29], where $\delta = min_{\forall (\mathbf{s},\mathbf{t}) \in \mathcal{A}_k | \rho(\mathbf{t}) \neq \rho(\mathbf{s})} |\rho(\mathbf{t}) - \rho(\mathbf{s})|$, as follows:

$$f_{min}(\langle \mathbf{t} \rangle) = \begin{cases} \rho(\mathbf{t}) & \text{if } \mathbf{t} \in \mathcal{S} \subset \mathcal{Z}_1, \\ \rho(\mathbf{t}) - \delta & \text{otherwise,} \end{cases}$$
$$f_{min}(\pi_s \cdot \langle \mathbf{s}, \mathbf{t} \rangle) = \min\{f(\pi_s), \rho(\mathbf{t})\}. \tag{7}$$

The algorithm maximizes a connectivity map $C_2(\mathbf{t}) = \max_{\forall \pi_t \in \Pi_t}\{f_{\min}(\pi_t)\}$, such that the graph is partitioned into an optimum-path forest P_2 (classifier) rooted at the maxima of the pdf. Indeed, it detects clusters on-the-fly, as well as only one sample per maximum to compose the set $\mathcal{S}$ (i.e, prototype set), and guarantees a single optimum-path tree per dome of the pdf. In order to guarantee that, we insert in $\mathcal{A}_k(\mathbf{s})$ the nodes $\mathbf{t}$, such that $\mathbf{s} \in \mathcal{A}_k(\mathbf{t})$ and $\rho(\mathbf{s}) = \rho(\mathbf{t})$ to make the graph symmetric on plateaus of the pdf. The label $L_1(\mathbf{s})$ of each root $\mathbf{s} \in \mathcal{S}$ is then propagated to each sample $\mathbf{t}$ of its tree. The result is an optimum-path forest with attributes $[P_2, C_2, L_2]$ (Fig. 2e and f), and a sorted list $\mathcal{Z}_1'$ of nodes in $\mathcal{Z}_1$ for the purpose of speeding up classification, as described in the next section.

Notice Algorithm 3 does not require color coding to control the status of nodes in Q. In the beginning, all nodes in $\mathcal{Z}_1$ are root candidates, and one node per maximum of the pdf ρ is selected (set $\mathcal{S}$) when $P_2(\mathbf{s}) = nil$ as root of the map in Line 7. This root node will then propagate the label $L_1(\mathbf{s})$ to all nodes on the same plateau (since $\mathcal{A}_k$ is symmetric on plateaus),

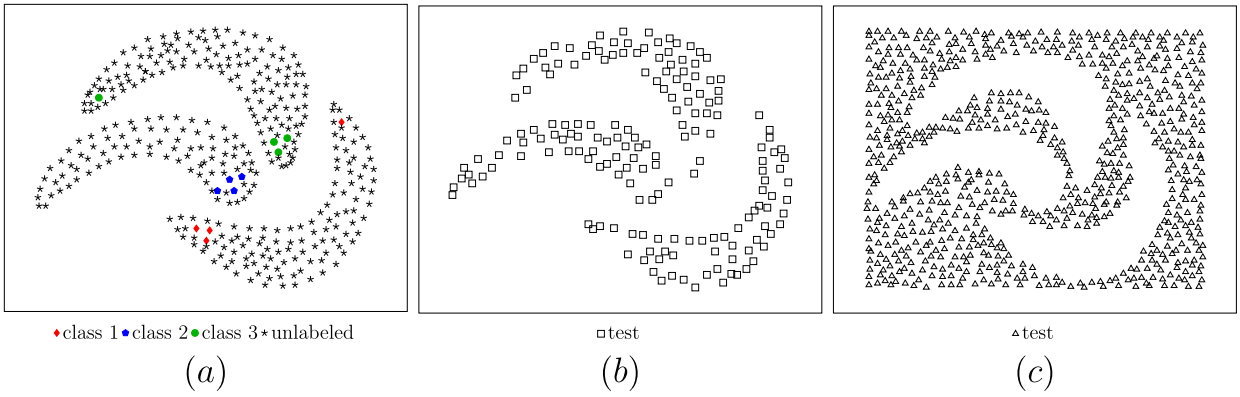

Fig. 3. A training set (2D feature space) with (a) unsupervised and a few supervised (colored) samples, (b) test samples inside the classes, and (c) test samples outside the classes.

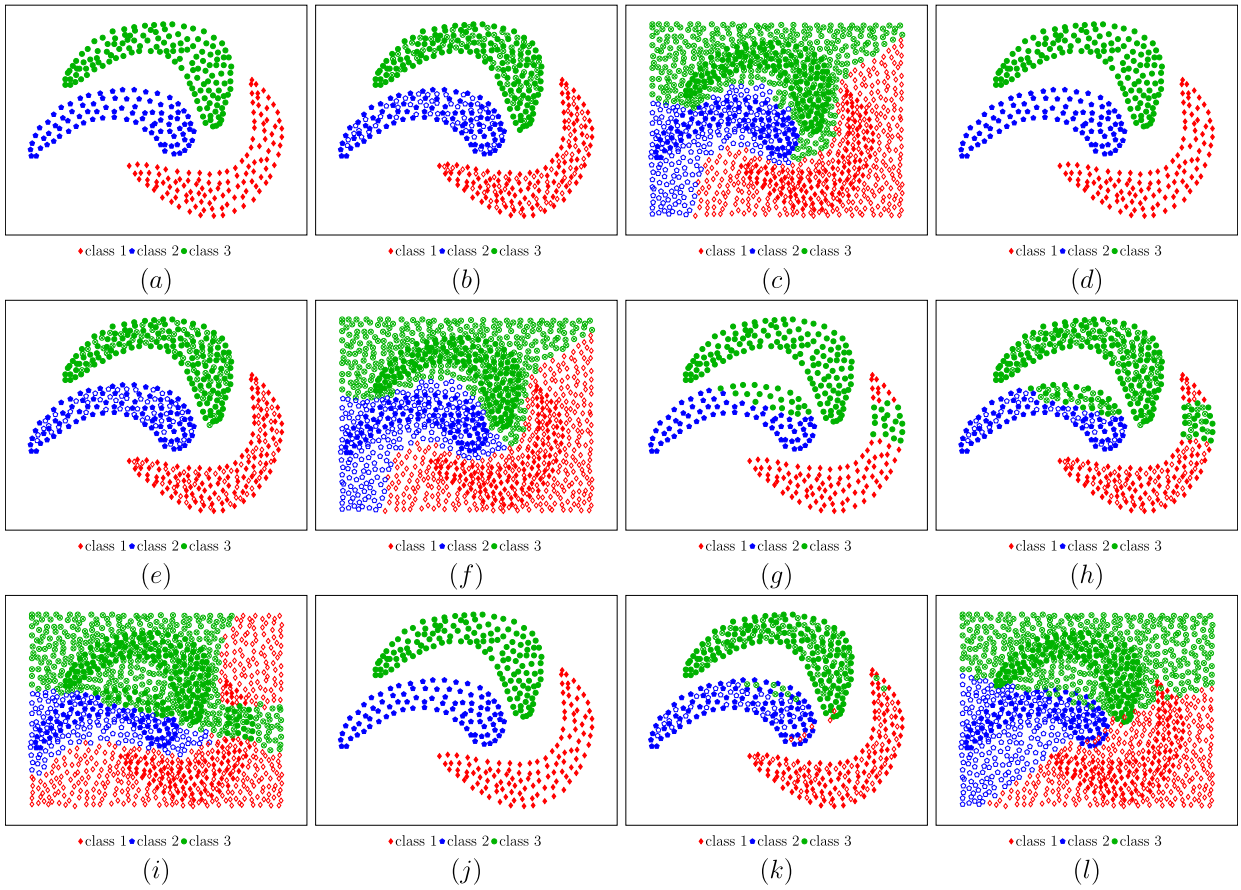

Fig. 4. Single-label assignment: label propagation to unsupervised samples, classification of test samples inside the classes, and classification of all test samples in Fig. 3 for (a–c) OPFSEMI$_{mst+knn}$ with 0.0% of propagation error on $\mathcal{Z}_1^u$ and 100.0% accuracy inside the classes on $\mathcal{Z}_2$, (d–f) OPFSEMI$_{mst}$ with 0.0% of propagation error on $\mathcal{Z}_1^u$ and 100.0% accuracy on test samples inside the classes of $\mathcal{Z}_2$, (g–i) TSVM with 11.2% of propagation error on $\mathcal{Z}_1^u$ and 94.7% accuracy inside the classes of $\mathcal{Z}_2$, and (j–l) LapSVM with 0.0% propagation error on $\mathcal{Z}_1^u$ and 97.1% accuracy inside the classes of $\mathcal{Z}_2$, respectively.

as well as on the same dome of the pdf. Note that, whenever a node **s** finds an adjacent node **t**, satisfying the condition in Line 11, such node **t** will be in Q, and it will be conquered by the root of **s**.

In order to choose the value of k, we first estimate the percentage of label propagation errors $\mathcal{E}$ that OPFSEMI$_{mst}$ might have committed in the given training set. We then execute Algorithms 1 and 2 on $\mathcal{Z}_1^l$ by defining the roots of the forest on half of the supervised samples, for further measuring $\mathcal{E}$ on the other half. This process can also be repeated a few times to better estimate the percentage of label propagation errors. Finally, we execute Algorithm 3 on the k-nn graph $(\mathcal{Z}_1, \mathcal{A}_k, d)$ for

Table 1
Experiment (ID - Identifier). Description of the benchmark problems in terms of application domain (domain), number of instances (#ins), number of attributes (#att), total number of labels (l_n), label cardinality (l_c) and label density (l_d).

ID–Dataset	Domain	#ins	#att	l_n	l_c	l_d
d_1–Scene	Multimedia	2047	294	6	1074	0.179
d_2–Yeast	Biology	2417	103	14	4237	0.303
d_3–Emotions	Multimedia	593	72	6	1869	0.311
d_4–Mediamill	Multimedia	43,907	120	101	4376	0.043
d_5–Birds	Audio	645	260	19	1014	0.053
d_6–Cal500	Music	502	68	174	26,044	0.150
d_7–Enron	Text	1702	1001	53	3378	0.064
d_8–Medical	Text	978	1449	45	1245	0.028

Table 2
Mean F-measure considering OPFSEMI$_{mst+knn}$ and OPFSEMI$_{mst}$.

	$\mathcal{Z}_1^l$	$\mathcal{Z}_1^u$	OPFSEMI$_{mst+knn}$			
			LP	BR	CC	HOMER
d_1	10%	90%	0.6100 ± 0.069	0.5838 ± 0.080	0.5928 ± 0.037	0.5765 ± 0.027
	50%	50%	0.6659 ± 0.042	0.6479 ± 0.074	0.6479 ± 0.083	**0.7086 ± 0.049**
d_2	10%	90%	**0.6474 ± 0.051**	0.5850 ± 0.055	0.6168 ± 0.094	0.5837 ± 0.061
	50%	50%	0.6529 ± 0.098	0.5999 ± 0.090	0.6357 ± 0.092	**0.6695 ± 0.057**
d_3	10%	90%	**0.6016 ± 0.047**	0.5977 ± 0.072	0.5714 ± 0.052	0.5331 ± 0.014
	50%	50%	0.6299 ± 0.097	0.5966 ± 0.043	0.6375 ± 0.055	**0.6516 ± 0.043**
d_4	10%	90%	0.4560 ± 0.028	0.4434 ± 0.043	0.4430 ± 0.065	**0.4752 ± 0.043**
	50%	50%	**0.5326 ± 0.059**	0.4859 ± 0.037	0.4870 ± 0.026	0.5158 ± 0.075
d_5	10%	90%	**0.5205 ± 0.032**	0.5131 ± 0.041	0.5198 ± 0.044	0.4900 ± 0.048
	50%	50%	**0.6144 ± 0.027**	0.6088 ± 0.042	0.6070 ± 0.021	0.5622 ± 0.036
d_6	10%	90%	**0.4330 ± 0.026**	0.4193 ± 0.082	0.3926 ± 0.051	0.4125 ± 0.011
	50%	50%	0.6750 ± 0.072	0.6308 ± 0.045	**0.6831 ± 0.032**	0.5936 ± 0.053
d_7	10%	90%	0.4678 ± 0.075	0.4653 ± 0.053	**0.4864 ± 0.031**	0.4787 ± 0.067
	50%	50%	0.5118 ± 0.036	0.5076 ± 0.019	0.5100 ± 0.086	**0.5172 ± 0.032**
d_8	10%	90%	0.3801 ± 0.025	0.3684 ± 0.036	0.4006 ± 0.056	0.4081 ± 0.036
	50%	50%	0.6384 ± 0.094	0.6134 ± 0.057	0.6368 ± 0.050	**0.6475 ± 0.052**
	$\mathcal{Z}_1^l$	$\mathcal{Z}_1^u$	**OPFSEMI$_{mst}$**			
			LP	BR	CC	HOMER
d_1	10%	90%	0.5512 ± 0.028	0.5371 ± 0.091	0.5649 ± 0.010	0.5534 ± 0.028
	50%	50%	0.6212 ± 0.089	0.6163 ± 0.026	0.6290 ± 0.094	0.6158 ± 0.020
d_2	10%	90%	0.6225 ± 0.041	0.5537 ± 0.094	0.5308 ± 0.051	0.5612 ± 0.031
	50%	50%	0.6400 ± 0.018	0.5942 ± 0.096	0.5706 ± 0.045	0.6495 ± 0.044
d_3	10%	90%	0.5604 ± 0.013	0.4393 ± 0.040	0.5603 ± 0.062	0.5194 ± 0.087
	50%	50%	0.6227 ± 0.079	0.6016 ± 0.033	0.6050 ± 0.090	0.6432 ± 0.028
d_4	10%	90%	0.4408 ± 0.085	0.4281 ± 0.041	0.4282 ± 0.099	0.4560 ± 0.023
	50%	50%	0.5215 ± 0.069	0.4757 ± 0.095	0.4770 ± 0.043	0.5103 ± 0.023
d_5	10%	90%	0.3852 ± 0.080	0.3775 ± 0.011	0.3934 ± 0.015	0.3938 ± 0.020
	50%	50%	0.5931 ± 0.098	0.5895 ± 0.066	0.5917 ± 0.045	0.5617 ± 0.092
d_6	10%	90%	0.3979 ± 0.021	0.3820 ± 0.091	0.3898 ± 0.024	0.3782 ± 0.042
	50%	50%	0.6613 ± 0.013	0.6131 ± 0.099	0.6752 ± 0.059	0.5833 ± 0.017
d_7	10%	90%	0.3272 ± 0.054	0.3062 ± 0.091	0.3100 ± 0.012	0.3514 ± 0.055
	50%	50%	0.3825 ± 0.068	0.3626 ± 0.032	0.3898 ± 0.063	0.3784 ± 0.047
d_8	10%	90%	0.3250 ± 0.011	0.3135 ± 0.047	0.3114 ± 0.070	0.3443 ± 0.044
	50%	50%	0.4104 ± 0.094	0.3932 ± 0.072	0.4234 ± 0.033	0.4410 ± 0.026

values of $k \in [k_{\max}, k_{\min}]$ in order to select the highest value of k that maintains the percentage of labeling disagreement between L_1 and L_2 less than or equal to $\mathcal{E}$. This criterion tends to obtain a good pdf estimation with higher values of k, and it also helps reducing mislabeled samples from Algorithm 2, since samples in higher density regions will conquer the ones in lower density regions (see Section 4.1).

4.5. Classifying new samples

In order to classify a new sample $\mathbf{v} \in \mathcal{Z}_2$, the algorithm evaluates optimum paths in an incremental way as follows:

$$C_2(\mathbf{v}) = \max_{\forall \mathbf{s} \in \{\mathcal{Z}_1 \cap A_k(\mathbf{v})\}} \{\min\{C_2(\mathbf{s}), \rho(\mathbf{v})\}\}. \tag{8}$$

Let node $\mathbf{s}^* \in \mathcal{Z}_1$ be the one that satisfies the above equation. Classification simply assigns $L_2(\mathbf{v}) \leftarrow L_2(\mathbf{s}^*)$.

Table 3
Mean *F*-measure considering LapSVM and TSVM.

	$\mathcal{Z}_1^l$	$\mathcal{Z}_1^u$	LapSVM			
			LP	BR	CC	HOMER
d_1	10%	90%	**0.6274 ± 0.045**	0.5884 ± 0.077	0.6249 ± 0.090	0.5946 ± 0.031
	50%	50%	0.6944 ± 0.024	0.6255 ± 0.052	0.6847 ± 0.016	0.6868 ± 0.095
d_2	10%	90%	0.5701 ± 0.019	0.5728 ± 0.012	0.5902 ± 0.073	0.5639 ± 0.063
	50%	50%	0.5933 ± 0.014	0.5819 ± 0.040	0.6101 ± 0.095	0.5792 ± 0.053
d_3	10%	90%	0.5873 ± 0.056	0.5731 ± 0.083	0.5769 ± 0.091	0.5770 ± 0.031
	50%	50%	0.6213 ± 0.028	0.5868 ± 0.043	0.6212 ± 0.054	0.6187 ± 0.099
d_4	10%	90%	0.4454 ± 0.071	0.4349 ± 0.065	0.4391 ± 0.082	0.4583 ± 0.086
	50%	50%	0.4841 ± 0.046	0.4516 ± 0.043	0.4862 ± 0.044	0.4932 ± 0.089
d_5	10%	90%	0.4783 ± 0.069	0.4717 ± 0.026	0.4703 ± 0.089	0.4957 ± 0.033
	50%	50%	0.5700 ± 0.045	0.5788 ± 0.039	0.5445 ± 0.019	0.5581 ± 0.095
d_6	10%	90%	0.4250 ± 0.098	0.3502 ± 0.068	0.3474 ± 0.042	0.3641 ± 0.064
	50%	50%	0.5394 ± 0.058	0.4781 ± 0.025	0.4714 ± 0.011	0.4847 ± 0.019
d_7	10%	90%	0.4454 ± 0.065	0.4214 ± 0.073	0.4398 ± 0.069	0.4350 ± 0.053
	50%	50%	0.4710 ± 0.073	0.4606 ± 0.068	0.4893 ± 0.098	0.4816 ± 0.064
d_8	10%	90%	0.4061 ± 0.095	0.3837 ± 0.049	0.3936 ± 0.078	**0.4174 ± 0.016**
	50%	50%	0.6213 ± 0.054	0.5316 ± 0.018	0.5850 ± 0.084	0.6313 ± 0.020
	$\mathcal{Z}_1^l$	$\mathcal{Z}_1^u$	**TSVM**			
			LP	BR	CC	HOMER
d_1	10%	90%	0.6034 ± 0.023	0.5182 ± 0.010	0.5338 ± 0.024	0.5845 ± 0.085
	50%	50%	0.6842 ± 0.069	0.6145 ± 0.079	0.6291 ± 0.069	0.6551 ± 0.026
d_2	10%	90%	0.5437 ± 0.042	0.5236 ± 0.028	0.5220 ± 0.069	0.5934 ± 0.095
	50%	50%	0.6048 ± 0.090	0.6001 ± 0.008	0.6012 ± 0.026	0.6088 ± 0.043
d_3	10%	90%	0.5116 ± 0.032	0.4382 ± 0.040	0.4567 ± 0.041	0.4892 ± 0.007
	50%	50%	0.5855 ± 0.027	0.5232 ± 0.026	0.5330 ± 0.091	0.5719 ± 0.092
d_4	10%	90%	0.4091 ± 0.045	0.4420 ± 0.014	0.4443 ± 0.002	**0.4752 ± 0.065**
	50%	50%	0.4386 ± 0.060	0.4538 ± 0.071	0.4481 ± 0.024	0.5069 ± 0.004
d_5	10%	90%	0.4778 ± 0.060	0.4867 ± 0.012	0.4893 ± 0.010	0.4771 ± 0.012
	50%	50%	0.5991 ± 0.019	0.5991 ± 0.061	0.5977 ± 0.084	0.5337 ± 0.065
d_6	10%	90%	0.3934 ± 0.066	0.3649 ± 0.073	0.3784 ± 0.029	0.3999 ± 0.030
	50%	50%	0.6543 ± 0.028	0.3791 ± 0.095	0.3581 ± 0.065	0.5421 ± 0.023
d_7	10%	90%	0.3366 ± 0.075	0.3260 ± 0.071	0.3285 ± 0.062	0.3444 ± 0.035
	50%	50%	0.4002 ± 0.028	0.3924 ± 0.054	0.3968 ± 0.010	0.4169 ± 0.038
d_8	10%	90%	0.3298 ± 0.055	0.3140 ± 0.028	0.3487 ± 0.037	0.3612 ± 0.048
	50%	50%	0.4840 ± 0.077	0.4721 ± 0.069	0.4668 ± 0.038	0.4854 ± 0.045

In [6], the authors improved speed in label propagation to new samples based on the pdf by avoiding the computation of $\mathcal{A}_k(\mathbf{v})$ for all $\mathbf{v} \in \mathcal{Z}_2$. A similar idea applies to Eq. (8). During training, a radius $\Omega(s)$ can be estimated as the maximum distance $d(\mathbf{s}, \mathbf{t})$ (the median value makes it more robust to outliers) between samples $\mathbf{s}, \mathbf{t} \in \mathcal{Z}_1$, such that $\mathbf{t} \in \mathcal{A}_k(\mathbf{s})$. If the distance $d(\mathbf{s}, \mathbf{v}) \leq \Omega(\mathbf{s})$ for a given $\mathbf{s} \in \mathcal{Z}_1$ and $\mathbf{v} \in \mathcal{Z}_2$, then $\mathbf{v}$ is within the region defined by the k-adjacency of $\mathbf{s}$.

The list $\mathcal{Z}_1'$ of training samples sorted in the non-increasing order of path values makes unnecessary to compute $\rho(\mathbf{v})$, and by following the order of nodes in $\mathcal{Z}_1'$, we set $L_2(\mathbf{v}) \leftarrow L_2(\mathbf{s}^*)$ for the first node $\mathbf{s}^* \in \mathcal{Z}_1'$ that satisfies $d(\mathbf{s}^*, \mathbf{v}) \leq \Omega(\mathbf{s}^*)$. Note that nodes more strongly connected to their roots will have higher priority in label assignment to new samples. Since mislabeled samples in $\mathcal{Z}_1$ (due to the previous step described in Section 4.3) are more likely to have lower pdf values, they will be the more weakly connected ones to their roots. This justifies the improvement of OPFSEMI$_{mst+knn}$ over OPFSEMI$_{mst}$ for the multi-label assignment problem.

Fig. 3a presents a simple training set (2D feature space) with supervised (colored) and unsupervised samples. Consider test samples inside the classes, as shown in Fig. 3b, and outside the classes, as shown in Fig. 3c. Fig. 4 illustrates the label propagation from the supervised to the unsupervised training samples, the classification of the test samples inside the classes, and the classification of all test samples for a single-label assignment problem using OPFSEMI$_{mst+knn}$ (Fig. 4a–c), OPFSEMI$_{mst}$ (Fig. 4d–f), TSVM [11] (Fig. 4g–i), and LapSVM [4] (Fig. 4j–l), respectively.

The connectivity between labeled and unlabeled (training and test) samples in OPFSEMI$_{mst+knn}$, OPFSEMI$_{mst}$ and manifold regularization in LapSVM can considerably reduce label propagation and classification errors in $\mathcal{Z}_1^u$ and $\mathcal{Z}_2$, respectively, as compared to TSVM. Note also that, by assuming a problem transformation strategy in multi-label assignment, those classification errors in single-label assignment tend to be amplified when the data transformation is reversed to obtain multiple labels per sample.

5. Experimental setup

In this section, we present the experimental analysis employed to compare the proposed OPFSEMI$_{mst+knn}$ against OPFSEMI$_{mst}$ [2], TSVM [19] and the manifold regularization approach [4] implemented in LapSVM[3] using four problem

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

Table 4
Mean Hamming Loss considering OPFSEMI$_{mst+knn}$ and OPFSEMI$_{mst}$.

	$\mathcal{Z}_1^l$	$\mathcal{Z}_1^u$	**OPFSEMI$_{mst+knn}$**			
			LP	BR	CC	HOMER
d_1	10%	90%	0.1154 ± 0.059	0.1110 ± 0.018	0.1162 ± 0.087	0.1203 ± 0.088
	50%	50%	0.1187 ± 0.076	0.1089 ± 0.040	0.1116 ± 0.072	0.1076 ± 0.059
d_2	10%	90%	$\mathbf{0.2161 \pm 0.045}$	0.2452 ± 0.054	0.2231 ± 0.044	0.2338 ± 0.046
	50%	50%	0.2091 ± 0.025	0.2385 ± 0.041	0.2143 ± 0.027	$\mathbf{0.2018 \pm 0.021}$
d_3	10%	90%	$\mathbf{0.2207 \pm 0.041}$	0.3263 ± 0.077	0.2359 ± 0.028	0.2667 ± 0.063
	50%	50%	0.2066 ± 0.054	0.2630 ± 0.056	0.2266 ± 0.084	0.2125 ± 0.034
d_4	10%	90%	0.0445 ± 0.059	0.0485 ± 0.015	0.0459 ± 0.040	0.0445 ± 0.084
	50%	50%	0.0413 ± 0.078	0.0470 ± 0.060	0.0457 ± 0.072	0.0413 ± 0.081
d_5	10%	90%	0.1388 ± 0.043	0.1441 ± 0.055	0.1320 ± 0.070	0.1218 ± 0.041
	50%	50%	0.0993 ± 0.017	0.1402 ± 0.084	0.1295 ± 0.080	0.0991 ± 0.030
d_6	10%	90%	0.2269 ± 0.083	0.2558 ± 0.049	0.2467 ± 0.031	$\mathbf{0.2123 \pm 0.078}$
	50%	50%	0.2168 ± 0.059	0.2401 ± 0.065	0.2318 ± 0.031	$\mathbf{0.2070 \pm 0.045}$
d_7	10%	90%	0.2950 ± 0.055	0.3110 ± 0.070	0.3009 ± 0.061	0.2596 ± 0.058
	50%	50%	0.2168 ± 0.071	0.3043 ± 0.022	0.2816 ± 0.057	0.2140 ± 0.083
d_8	10%	90%	0.0347 ± 0.022	0.0557 ± 0.028	0.0435 ± 0.013	0.0347 ± 0.070
	50%	50%	0.0339 ± 0.017	0.0401 ± 0.027	0.0366 ± 0.084	$\mathbf{0.0311 \pm 0.077}$
	$\mathcal{Z}_1^l$	$\mathcal{Z}_1^u$	**OPFSEMI$_{mst}$**			
			LP	BR	CC	HOMER
d_1	10%	90%	0.1103 ± 0.054	0.1172 ± 0.011	0.1151 ± 0.090	0.1417 ± 0.072
	50%	50%	$\mathbf{0.0817 \pm 0.069}$	0.0892 ± 0.016	0.0889 ± 0.034	0.1167 ± 0.065
d_2	10%	90%	0.2411 ± 0.072	0.2472 ± 0.052	0.2488 ± 0.053	0.2833 ± 0.084
	50%	50%	0.2057 ± 0.043	0.2053 ± 0.060	0.2027 ± 0.039	0.2359 ± 0.056
d_3	10%	90%	0.2877 ± 0.079	0.2950 ± 0.084	0.3041 ± 0.022	0.3277 ± 0.023
	50%	50%	0.2241 ± 0.079	0.2275 ± 0.058	0.2382 ± 0.041	0.2410 ± 0.036
d_4	10%	90%	0.0360 ± 0.016	0.0343 ± 0.083	$\mathbf{0.0341 \pm 0.063}$	0.0380 ± 0.028
	50%	50%	0.0348 ± 0.055	0.0339 ± 0.069	$\mathbf{0.0338 \pm 0.021}$	0.0358 ± 0.066
d_5	10%	90%	0.1418 ± 0.067	0.1456 ± 0.085	0.1356 ± 0.022	0.1253 ± 0.030
	50%	50%	0.1009 ± 0.031	0.1439 ± 0.059	0.1310 ± 0.025	0.1012 ± 0.070
d_6	10%	90%	0.2320 ± 0.049	0.2594 ± 0.010	0.2525 ± 0.029	0.2170 ± 0.025
	50%	50%	0.2191 ± 0.080	0.2461 ± 0.049	0.2360 ± 0.040	0.2123 ± 0.044
d_7	10%	90%	0.2991 ± 0.063	0.3159 ± 0.038	0.3064 ± 0.038	0.2664 ± 0.068
	50%	50%	0.2201 ± 0.030	0.3111 ± 0.071	0.2866 ± 0.012	0.2180 ± 0.059
d_8	10%	90%	0.0356 ± 0.033	0.0569 ± 0.087	0.0439 ± 0.060	0.0350 ± 0.041
	50%	50%	0.0347 ± 0.084	0.0406 ± 0.070	0.0373 ± 0.037	0.0318 ± 0.078

transformation methodologies: Label Powerset, Binary Relevance, Classifier Chains and Hierarchy of Multi-label Classifiers. We also evaluated two adaptation algorithms using self-training (BPMLL and ML*k*NN), as implemented in the Mulan Java Library.[4] This package includes implementations of a number of multi-label classification methods.

Each experiment was repeated 100 times with random generated sets partitioned into two parts: 70% for the training set $\mathcal{Z}_1$ and 30% for the test set $\mathcal{Z}_2$. We also evaluated different proportions between the sizes of the supervised $\mathcal{Z}_1^l$ and unsupervised $\mathcal{Z}_1^u$ training sets[5] The results were evaluated by means of the *F*-measure and Hamming Loss [22], that are two standard evaluation metrics for multi-label classification. Eight multi-labels[6] datasets were used in the experiments, as presented in Table 1. These datasets come from five domains: multimedia ('Emotions', 'Scene' and 'Mediamill'), biology ('Yeast'), audio ('Birds'), music ('Cal500') and text ('Enron' and 'Medical').

5.1. Parameter tuning

In regard to TSVM, we used SVMLight [19]. In both implementations (TSVM and LapSVM), we considered radial basis kernels, being their parameters optimized by a 5-fold cross validation in $\mathcal{Z}_1^l$. In our experiments, we considered $C \in \{10^{-5}, 10^{-3}, 10^{-1}, 10, 10^3, 10^5\}$ and $\gamma \in \{10^{-5}, 10^{-3}, 10^{-1}, 1, 10\}$. With respect to OPFSEMI$_{mst+knn}$ and OPFSEMI$_{mst}$, we used the LibOPF library[7]. In regard to HOMER, we evaluated five different numbers of clusters (i. e, $2-6$), and selected the best one to build the hierarchy of multi-label classifiers. The remaining parameters used their default values.

5.2. Evaluation of multi-label assignment

Tables 2 and 3 present the classification performance according to the following format $a \pm b$, where a and b denote, respectively, the mean *F*-measure and its standard deviation concerning OPFSEMI$_{mst+knn}$, OPFSEMI$_{mst}$, LapSVM and TSVM over

[4] http://mulan.sourceforge.net/.
[5] The percentages were empirically chosen.
[6] http://mulan.sourceforge.net/datasets-mlc.html.
[7] http://www.ic.unicamp.br/~afalcao/libopf.

Table 5
Mean Hamming Loss considering LapSVM and TSVM.

	$\mathcal{Z}_1^l$	$\mathcal{Z}_1^u$	LapSVM			
			LP	BR	CC	HOMER
d_1	10%	90%	**0.0930 ± 0.065**	0.1252 ± 0.084	0.0981 ± 0.062	0.1236 ± 0.023
	50%	50%	0.0955 ± 0.046	0.1187 ± 0.049	0.0942 ± 0.022	0.1208 ± 0.025
d_2	10%	90%	0.2495 ± 0.077	0.2538 ± 0.059	0.2511 ± 0.085	0.2636 ± 0.064
	50%	50%	0.2530 ± 0.012	0.2560 ± 0.041	0.2412 ± 0.040	0.2563 ± 0.061
d_3	10%	90%	0.2523 ± 0.084	0.2663 ± 0.066	0.2698 ± 0.044	0.2511 ± 0.061
	50%	50%	0.2421 ± 0.033	0.2432 ± 0.077	0.2592 ± 0.060	0.2466 ± 0.058
d_4	10%	90%	0.0459 ± 0.028	0.0493 ± 0.040	0.0462 ± 0.048	0.0451 ± 0.039
	50%	50%	0.0457 ± 0.018	0.0473 ± 0.082	0.0458 ± 0.044	0.0352 ± 0.039
d_5	10%	90%	0.1404 ± 0.019	0.1456 ± 0.058	0.1335 ± 0.041	0.1237 ± 0.088
	50%	50%	0.1011 ± 0.057	0.1427 ± 0.072	0.1316 ± 0.073	0.1008 ± 0.063
d_6	10%	90%	0.2299 ± 0.038	0.2605 ± 0.067	0.2503 ± 0.026	0.2161 ± 0.042
	50%	50%	0.2191 ± 0.031	0.2438 ± 0.022	0.2360 ± 0.044	0.2111 ± 0.023
d_7	10%	90%	0.2983 ± 0.060	0.3172 ± 0.073	0.3050 ± 0.057	0.2631 ± 0.037
	50%	50%	0.2206 ± 0.024	0.3084 ± 0.056	0.2870 ± 0.034	0.2179 ± 0.017
d_8	10%	90%	0.0352 ± 0.029	0.0567 ± 0.071	0.0441 ± 0.072	**0.0342 ± 0.030**
	50%	50%	0.0345 ± 0.063	0.0406 ± 0.076	0.0370 ± 0.029	0.0315 ± 0.056
	$\mathcal{Z}_1^l$	$\mathcal{Z}_1^u$	**TSVM**			
			LP	BR	CC	HOMER
d_1	10%	90%	0.1534 ± 0.016	0.1546 ± 0.021	0.1543 ± 0.056	0.1541 ± 0.033
	50%	50%	0.1340 ± 0.011	0.1355 ± 0.081	0.1356 ± 0.080	0.1348 ± 0.014
d_2	10%	90%	0.2667 ± 0.062	0.2695 ± 0.062	0.2698 ± 0.074	0.2731 ± 0.061
	50%	50%	0.2502 ± 0.038	0.2550 ± 0.029	0.2548 ± 0.010	0.2584 ± 0.087
d_3	10%	90%	0.3001 ± 0.074	0.3007 ± 0.070	0.3012 ± 0.063	0.3170 ± 0.020
	50%	50%	0.2461 ± 0.079	0.2483 ± 0.031	0.2466 ± 0.057	0.2506 ± 0.022
d_4	10%	90%	0.0509 ± 0.083	0.0511 ± 0.031	0.0511 ± 0.055	0.0489 ± 0.041
	50%	50%	0.0478 ± 0.034	0.0480 ± 0.029	0.0411 ± 0.076	0.0462 ± 0.012
d_5	10%	90%	0.1406 ± 0.026	0.1466 ± 0.087	0.1358 ± 0.014	0.1259 ± 0.061
	50%	50%	0.1028 ± 0.068	0.1424 ± 0.033	0.1308 ± 0.032	0.1017 ± 0.076
d_6	10%	90%	0.2293 ± 0.078	0.2643 ± 0.045	0.2513 ± 0.012	0.2175 ± 0.020
	50%	50%	0.2239 ± 0.077	0.2476 ± 0.049	0.2398 ± 0.035	0.2137 ± 0.038
d_7	10%	90%	0.3028 ± 0.010	0.3169 ± 0.013	0.3078 ± 0.029	0.2666 ± 0.049
	50%	50%	0.2198 ± 0.024	0.3127 ± 0.073	0.2875 ± 0.023	0.2205 ± 0.069
d_8	10%	90%	0.0355 ± 0.073	0.0577 ± 0.022	0.0440 ± 0.027	0.0353 ± 0.081
	50%	50%	0.0344 ± 0.080	0.0414 ± 0.047	0.0374 ± 0.077	0.0321 ± 0.073

Table 6
Mean *F*-measure and Hamming Loss considering ML*k*NN and BPMLL.

	$\mathcal{Z}_1^l$	$\mathcal{Z}_1^u$	ML*k*NN		BPMLL	
			F-measure	Hamming Loss	*F*-measure	Hamming Loss
d_1	10%	90%	0.5614 ± 0.087	0.1607 ± 0.026	0.5581 ± 0.027	0.2572 ± 0.019
	50%	50%	0.6166 ± 0.016	0.1434 ± 0.033	0.6205 ± 0.057	0.1542 ± 0.012
d_2	10%	90%	0.5942 ± 0.065	0.2918 ± 0.037	0.6083 ± 0.029	0.2321 ± 0.032
	50%	50%	0.5968 ± 0.084	0.3021 ± 0.036	0.6193 ± 0.042	0.2300 ± 0.045
d_3	10%	90%	0.5279 ± 0.023	0.3287 ± 0.071	0.5801 ± 0.018	0.2444 ± 0.035
	50%	50%	0.5948 ± 0.067	0.2879 ± 0.060	0.6440 ± 0.051	**0.2032 ± 0.016**
d_4	10%	90%	0.4747 ± 0.089	0.0489 ± 0.055	0.4637 ± 0.074	0.0696 ± 0.053
	50%	50%	0.5103 ± 0.063	0.0479 ± 0.010	0.4707 ± 0.072	0.0651 ± 0.085
d_5	10%	90%	0.4639 ± 0.054	**0.0830 ± 0.016**	0.4605 ± 0.060	0.1579 ± 0.081
	50%	50%	0.4821 ± 0.062	**0.0822 ± 0.044**	0.5129 ± 0.024	0.0772 ± 0.085
d_6	10%	90%	0.3519 ± 0.012	0.2172 ± 0.046	0.4017 ± 0.041	0.2882 ± 0.070
	50%	50%	0.3801 ± 0.034	0.2075 ± 0.016	0.4417 ± 0.063	0.2694 ± 0.011
d_7	10%	90%	0.3383 ± 0.057	**0.0858 ± 0.013**	0.3717 ± 0.040	0.1367 ± 0.045
	50%	50%	0.4521 ± 0.019	**0.0809 ± 0.069**	0.4106 ± 0.026	0.1425 ± 0.075
d_8	10%	90%	0.3140 ± 0.010	0.0373 ± 0.075	0.3840 ± 0.072	0.0351 ± 0.062
	50%	50%	0.5053 ± 0.011	0.0317 ± 0.018	0.5536 ± 0.011	0.0313 ± 0.049

all transformation methods. Similarly, Tables 4 and 5 present the classification performance with respect to the mean Hamming Loss measure and its standard deviation. Finally, Table 6 shows the mean *F*-measure and Hamming Loss as well as their standard deviation for ML*k*NN and BPMLL techniques. The values in bold indicate the best results considering the triplet (dataset, percentage of $\mathcal{Z}_1^l$ and $\mathcal{Z}_1^u$, problem transformation method or algorithm adaptation). For instance, OPFSEMI$_{mst+knn}$ (using *F*-measure) was the best technique in the Scene dataset (identifier dataset - d_1) with HOMER as problem transforma-

Table 7
Percentage of label propagation errors ($\mathcal{E}$) on $\mathcal{Z}_1^u$ for OPFSEMI$_{mst+knn}$, OPFSEMI$_{mst}$, and the best value k^* obtained for each dataset concerning OPFSEMI$_{mst+knn}$.

	$\mathcal{Z}_1^l$	$\mathcal{Z}_1^u$	**OPFSEMI$_{mst+knn}$**				**OPFSEMI$_{mst}$**			
			LP	BR	CC	HOMER	LP	BR	CC	HOMER
d_1	10%	90%	35.37 (k^* = 17)	37.50 (k^* = 20)	39.12 (k^* = 20)	38.30 (k^* = 16)	36.63	40.79	43.16	40.71
	50%	50%	18.40 (k^* = 13)	18.86 (k^* = 18)	17.86 (k^* = 17)	17.40 (k^* = 15)	20.16	23.66	22.56	20.46
d_2	10%	90%	39.50 (k^* = 9)	37.60 (k^* = 15)	37.47 (k^* = 13)	37.75 (k^* = 11)	43.87	40.05	40.15	40.74
	50%	50%	16.63 (k^* = 9)	16.09 (k^* = 13)	14.77 (k^* = 12)	15.94 (k^* = 10)	17.79	20.51	16.81	17.17
d_3	10%	90%	39.06 (k^* = 17)	37.54 (k^* = 19)	36.87 (k^* = 20)	36.51 (k^* = 21)	39.21	37.83	37.35	40.30
	50%	50%	17.51 (k^* = 13)	16.33 (k^* = 16)	15.83 (k^* = 18)	14.29 (k^* = 16)	21.16	16.72	19.40	17.62
d_4	10%	90%	39.32 (k^* = 12)	39.91 (k^* = 17)	39.98 (k^* = 15)	37.11 (k^* = 14)	39.59	41.34	41.15	38.48
	50%	50%	16.30 (k^* = 11)	19.48 (k^* = 13)	18.12 (k^* = 13)	16.21 (k^* = 10)	19.71	21.31	18.96	17.67
d_5	10%	90%	25.93 (k^* = 5)	27.74 (k^* = 8)	26.06 (k^* = 9)	20.34 (k^* = 12)	29.97	31.62	30.65	27.18
	50%	50%	10.25 (k^* = 4)	12.91 (k^* = 7)	11.80 (k^* = 5)	7.46 (k^* = 8)	14.13	16.32	15.20	9.63
d_6	10%	90%	40.34 (k^* = 12)	41.64 (k^* = 14)	41.84 (k^* = 12)	42.02 (k^* = 15)	45.02	46.11	46.58	44.27
	50%	50%	16.44 (k^* = 5)	18.40 (k^* = 7)	19.11 (k^* = 12)	18.49 (k^* = 10)	25.02	29.34	30.72	29.56
d_7	10%	90%	38.29 (k^* = 30)	39.05 (k^* = 34)	38.71 (k^* = 32)	37.11 (k^* = 27)	43.32	42.01	40.81	42.64
	50%	50%	18.39 (k^* = 21)	19.62 (k^* = 24)	18.50 (k^* = 18)	16.47 (k^* = 20)	23.42	23.59	21.77	22.19
d_8	10%	90%	32.89 (k^* = 32)	34.24 (k^* = 40)	33.02 (k^* = 36)	30.14 (k^* = 25)	35.29	36.55	35.57	32.54
	50%	50%	14.87 (k^* = 22)	18.56 (k^* = 25)	17.43 (k^* = 20)	12.08 (k^* = 18)	16.82	20.34	18.61	17.71

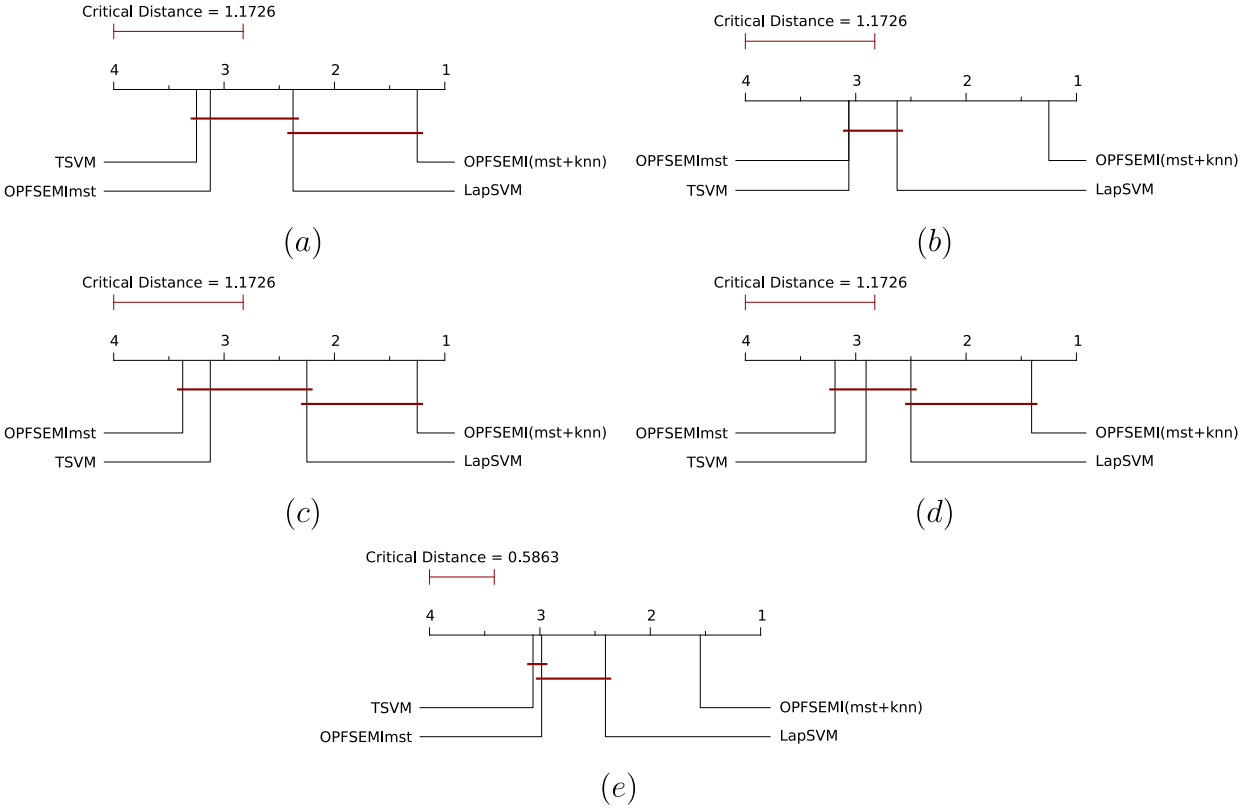

Fig. 5. Comparison of all classifiers using transformation methods against to each other with the Nemenyi test (using *F*-measure values). Groups of classifiers that are not significantly different (at p = .05) are connected: (a) Label Powerset, (b) Binary Relevance, (c) Classifier Chain, and (d) Hierarchy of Multi-Label Classifiers, and (e) all transformation methods.

tion method using 50% of $\mathcal{Z}_1$ for $\mathcal{Z}_1^l$. Table 7 presents the percentage of the label propagation error on $\mathcal{Z}_1^u$ ($\mathcal{E}$) concerning OPFSEMI$_{mst+knn}$ and OPFSEMI$_{mst}$. Additionally, Table 7 shows the best value k^* obtained for each dataset in OPFSEMI$_{mst+knn}$.

In general, the best results were obtained by the transformation methods LP/HOMER with the OPFSEMI$_{mst+knn}$ classifier. Optimum connectivity between supervised and unsupervised samples allows a considerable performance for OPFSEMI$_{mst+knn}$, which makes it to generalize better than LapSVM and TSVM when capturing the shapes of the classes in the feature space.

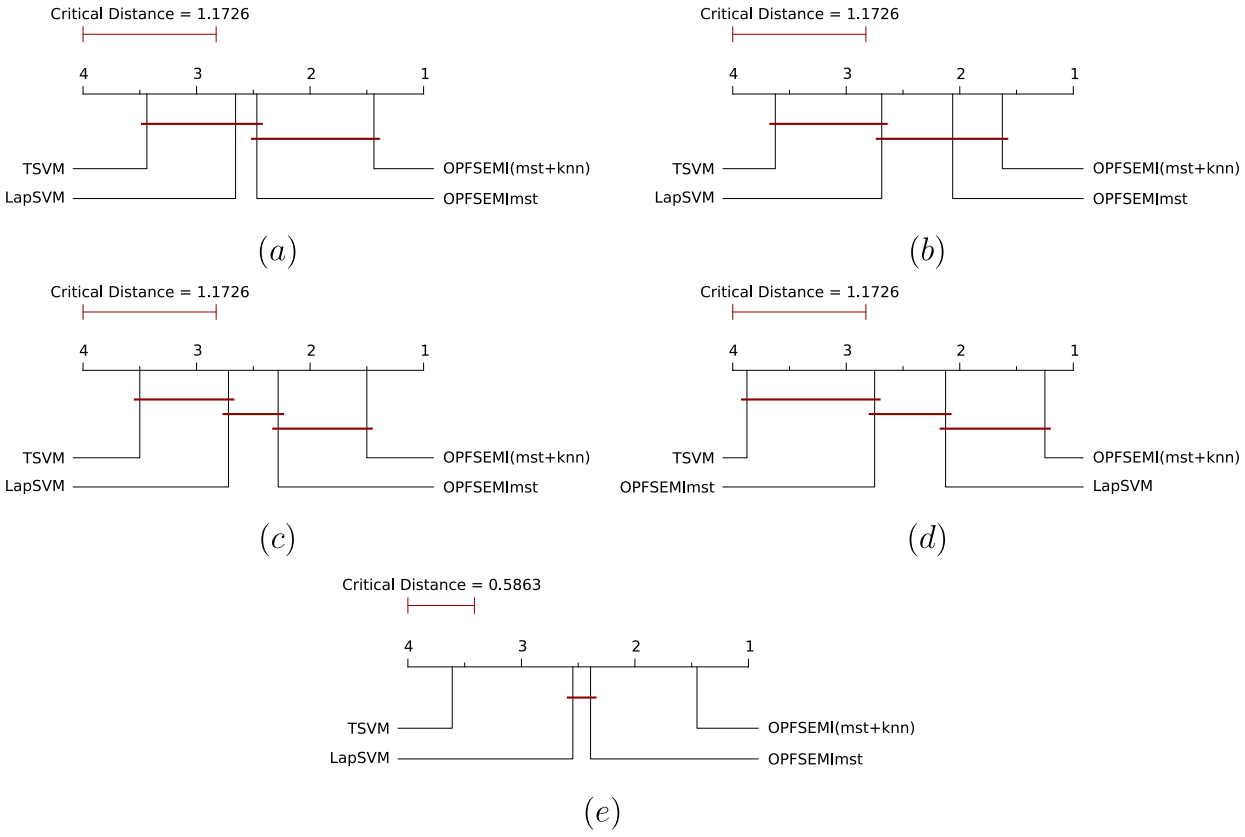

Fig. 6. Comparison of all classifiers using transformation methods against to each other with the Nemenyi test (using Hamming Loss values). Groups of classifiers that are not significantly different (at p = .05) are connected: (a) Label Powerset, (b) Binary Relevance, (c) Classifier Chain, and (d) Hierarchy of Multi-Label Classifiers, and (e) all transformation methods.

Conceptually, a good condition for both OPFSEMI$_{mst+knn}$ and OPFSEMI$_{mst}$ [2] concerns the graph, which should reveal the true intrinsic complexity or dimensionality of the data points (say through local linear relationships), as well as it should capture certain global structures of the data as a whole (i.e. clusters or subspaces), even after the transformation of the data/ to single-label problems. A possible shortcoming refers to the situations that do not ensure the criterion of smoothness among classes, or when we have irrelevant supervised data (e.g., mislabeled samples due to human errors, which might be a problem in large-scale studies). This can impair the label propagation to the unsupervised samples, and such information might not represent the actual relationship among classes, making even worse the classification results than using only supervised data.

The results usually show improvements as the size of $\mathcal{Z}_1^l$ increases. In most cases, when using a smaller set of supervised data (i.e, 10% of $\mathcal{Z}_1$), OPFSEMI$_{mst+knn}$ with LP exceeds the best results under the same conditions for the majority of the cases analyzed, while BPMLL obtained the best performance among the algorithm adaptation strategies. On the other hand, for larger supervised sets (i.e 50% of $\mathcal{Z}_1$), OPFSEMI$_{mst+knn}$ with HOMER seems to be a better choice. We believe that LP preserves better the relation among the actual labels than HOMER after data transformation when using smaller sets of supervised samples, which is usually the case in multi-label classification. The Classifier Chain results are due to an improved relation between the supervised and unsupervised samples, due to the fact that there is a sequence of binary classifiers, making each label be classified considering the prediction of labels previously analyzed. Unfortunately, the BR technique without any treatment is not a good solution to this problem, as it treats each class individually, thus ignoring the possible relations among them. A possible improvement would be an individual treatment of clusters to help maintaining the relationships among samples. Roughly speaking, LapSVM and OPFSEMI$_{mst}$ have very similar behavior, highlighting LapSVM by the number of top results in different domains of datasets. Another interesting observation concerns the robustness of the semi-supervised techniques with respect to the number of supervised samples. This may be explained by the margin correction of the classifier due to the presence of unsupervised samples in $\mathcal{Z}_1^u$, which are correctly supervised from $\mathcal{Z}_1^l$.

5.3. Results of the statistical analysis

In order to provide a statistical analysis of the results, we performed a Friedman test [16] to evaluate all methods. The Friedman test provides reliable conclusions when the assumptions (normal distributions and sphericity) of the traditional

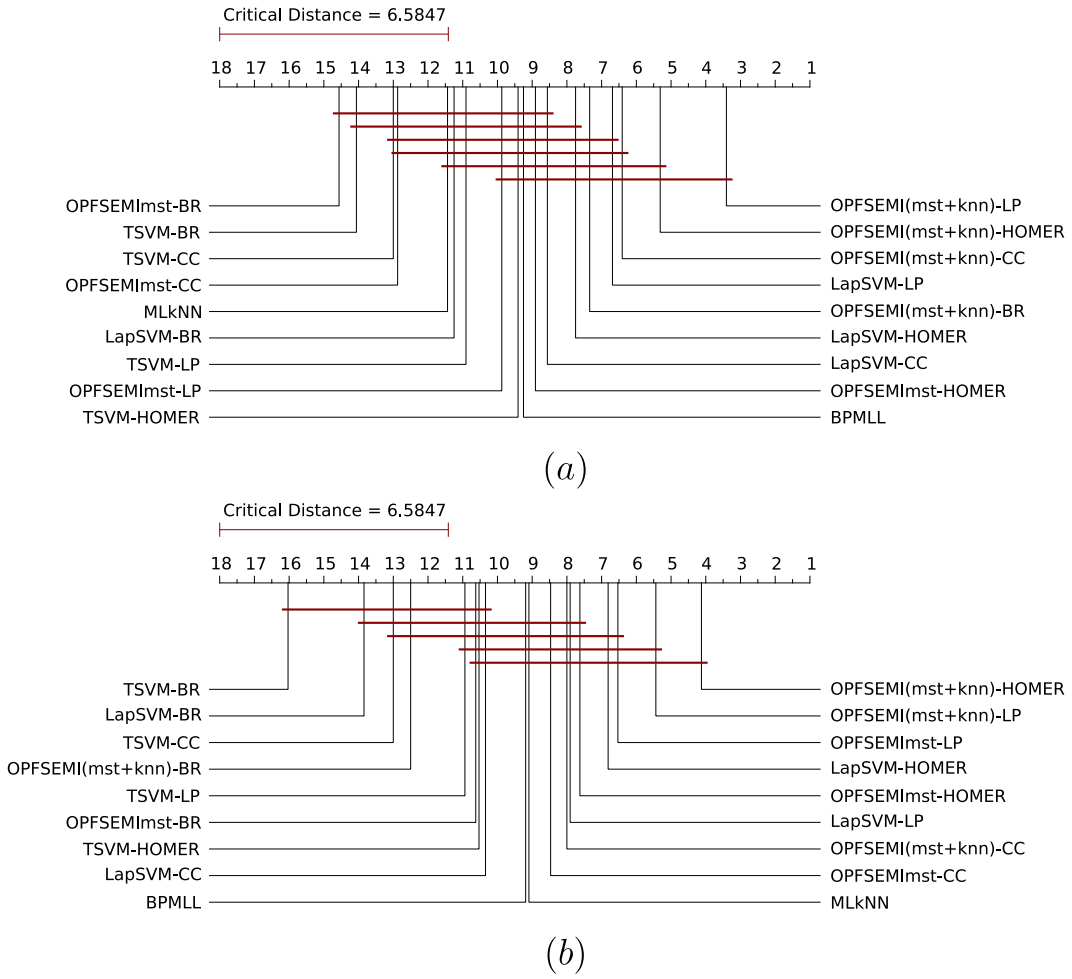

Fig. 7. Comparison of all methods together with algorithm adaptation strategies against to each other with the Nemenyi test. Groups of classifiers that are not significantly different (at p = .05) are connected: (a) using *F*-measure values and (b) using Hamming Loss values.

multiple hypotheses testing ANOVA are violated. The purpose of this statistical evaluation is to validate the results, as well as to show the behavior of different data transformation models and algorithm adaptation strategies for semi-supervised approaches. In the first scenario, we evaluate all semi-supervised algorithms using transformation methods, and in the second round we considered all techniques together with the ones that make use of algorithm adaptation strategies. Therefore, we can make clearer the robustness of OPFSEMI$_{mst+knn}$ against all techniques addressed in this paper concerning different evaluation measures and multi-label assignment strategies. Fig. 5–6 (first scenario) and Fig. 7 (second scenario) illustrate the post-hoc Nemenyi test, since we rejected the null hypotheses that all classifiers are equivalent to each other according to the Friedman test. In regard to Nemenyi test, groups of similar classifiers (with significance of 0.05) are connected using a critical distance (CD), where the far right classifier (i.e. numbered as 1) is the best one. On the other hand, the far left technique stands for the worst one.

Roughly speaking, the results of both tests (Nemenyi test, F-measure and Hamming Loss) in the two assessment scenarios are equivalent. The experimental results highlighted the best results were obtained by OPFSEMI$_{mst+knn}$ followed by LapSVM, OPFSEMI$_{mst}$ and BPMLL. Also, we can stress OPFSEMI$_{mst+knn}$ as being the best approach using BR (Fig. 5b), and in general the best classifier using transformation methods (Figs. 5e–6 e) with results equivalent to the ones obtained by LapSVM and OPFSEMI$_{mst}$.

In order to perform a deeper analysis, we compared the pair of classifiers OPFSEMI$_{mst+knn}$ and OPFSEMI$_{mst}$ by means of the Wilcoxon signed-rank test [14]. Such test is an important analysis that turns out to be more sensitive since it does not assume normal distributions. In case of *F*-measure values using a significance of 8.017^{-8} and Hamming Loss values using a significance of $p = .045$, the techniques can be considered statistically different. This assumption confirms the improvement of OPFSEMI$_{mst+knn}$ over its previous version OPFSEMI$_{mst}$. The OPFSEMI$_{mst+knn}$ shows the gain in using the proposed structure based on the MST with OPF and a final classifier with a *k*-nn graph, thus ensuring stronger relationship among classes even after the binary transformation of multi-label problems as compared against other semi-supervised approaches.

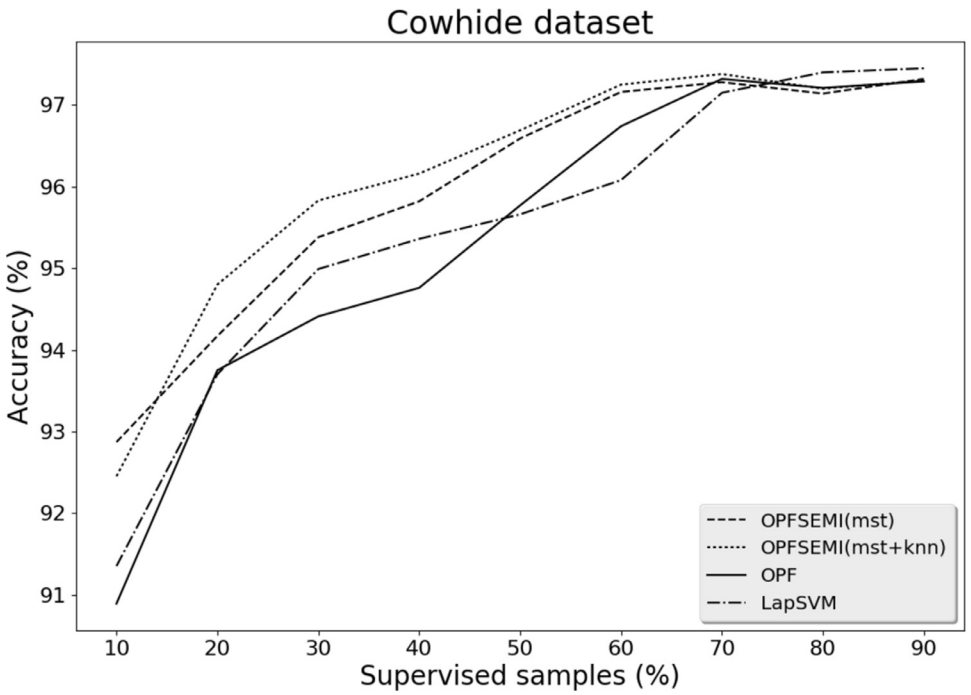

Fig. 8. Accuracy curves of OPFSEMI$_{mst}$, OPFSEMI$_{mst+knn}$, OPF, and LapSVM on the Cowhide dataset.

5.4. Evaluation of single-label assignment on a real application

In this section, we verify if the comparative results for multi-label assignment are confirmed for single-label assignment on a real application — the classification of leather defects. For that, we introduce the Cowhide [3] dataset, with 1690 samples, 160 attributes, and five types of regions of interest (categories) for classification in the Wet-Blue[8] processing stages: scabies, ticks, hot-iron, cut, and regions without defect. This is a challenging problem, especially in areas close to the vicinity of different defects.

The dataset was randomly divided into 70% of the samples for the training set $\mathcal{Z}_1$ and 30% for the test set $\mathcal{Z}_2$. We then evaluated the proposed methods, OPF, and the most competitive one for the multi-label assignment problem (LapSVM) with different proportions of randomly selected samples for the supervised set $\mathcal{Z}_1^l$ and unsupervised set $\mathcal{Z}_1^u$, using $\mathcal{Z}_1^l \cup \mathcal{Z}_1^u = \mathcal{Z}_1$. The sizes of $\mathcal{Z}_1^l$ and $\mathcal{Z}_1^u$ ranged from 10%–90% to 90%–10% with respect to the size of $\mathcal{Z}_1$. As a supervised method, OPF is trained on $\mathcal{Z}_1^l$ only, while the semi-supervised methods, OPFSEMI$_{mst}$, OPFSEMI$_{mst+knn}$, and LapSVM, first propagate labels from $\mathcal{Z}_1^l$ to $\mathcal{Z}_1^u$ before training on $\mathcal{Z}_1$. They are all tested on the same set $\mathcal{Z}_2$. Accuracy is measured as proposed in [25] and the resulting effectiveness curves are presented in Fig. 8.

The results confirm that OPFSEMI$_{mst+knn}$ is the best approach among the evaluated techniques, being OPFSEMI$_{mst}$ the most competitive. OPF and LapSVM, are only competitive when the number of supervised samples is above 70%.

6. Conclusion

We presented a novel semi-supervised approach for multi-label classification tasks named OPFSEMI$_{mst+knn}$, using the Optimum-Path Forest framework. The method can reduce the label propagation errors of OPFSEMI$_{mst}$ in the training set by repropagating labels from the maxima of a probability density function, since misclassified samples usually appear at the boundaries among clusters, i.e., far from the maxima. Additionally, during classification, the training samples closer to their maxima have higher priority to assign labels to new samples. We then demonstrated that this more conservative approach can outperform OPFSEMI$_{mst}$ in the multi-label assignment problem using several datasets, as well as it can be more accurate than some state-of-the-art techniques. We also showed a similar result on a single-label assignment problem from a real application using the effectiveness curves of the methods.

The advances in data acquisition create large datasets to support research and technological development. However, the supervision (sample labeling) of large training sets by experts is infeasible in applications from several areas of the Sciences and Engineering. In this context, the choice of a minimum number of relevant samples for expert supervision becomes

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

crucial. At the same time, the choice of a considerably larger set of unsupervised training samples is important to the design of a more effective semi-supervised classifier.

Active learning approaches have addressed the above problem with the aim of producing an effective classifier with minimum user effort in sample supervision, through learning iterations until user satisfaction. We strongly believe that the semi-supervised OPF methods can be used to accomplish that aim. As relevant samples are selected for expert supervision, the label propagation errors in the unsupervised set are expected to reduce, speeding up the expert satisfaction with the semi-supervised classifier under design. We then intend to explore OPFSEMI$_{mst}$ and OPFSEMI$_{mst+knn}$ for active learning in single-label and multi-label assignment problems, and a more comprehensive evaluation of the impact of self-paced learning to select unsupervised training samples.

Acknowledgments

The authors are grateful to Fundect-MS, CNPq 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, #2014/12236-1 and #2014/16250-9.

References

[1] W.P. Amorim, A.X. Falcão, M.H. Carvalho, Semi-supervised pattern classification using optimum-path forest, in: Proceedings of the 27th SIBGRAPI Conference on Graphics, Patterns and Images, 2014, pp. 111–118, doi:10.1109/SIBGRAPI.2014.45.
[2] W.P. Amorim, A.X. Falcão, J.P. Papa, M.H. Carvalho, Improving semi-supervised learning through optimum connectivity, Pattern Recognit. 60 (C) (2016) 72–85, doi:10.1016/j.patcog.2016.04.020.
[3] W.P. Amorim, H. Pistori, M.C. Pereira, M.A.C. Jacinto, Attributes reduction applied to leather defects classification, in: Proceedings of the 23rd SIBGRAPI Conference on Graphics, Patterns and Images, 2010, pp. 353–359, doi:10.1109/SIBGRAPI.2010.54.
[4] 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. http://dl.acm.org/citation.cfm?id=1248547.1248632.
[5] N. Bridle, X. Zhu, p-Voltages: Laplacian regularization for semi-supervised learning on high-dimensional data, 2013, http://pages.cs.wisc.edu/~jerryzhu/pub/bridle_zhu_mlg.pdf.
[6] F.A.M. Cappabianco, A.X. Falcão, C.L. Yasuda, J.K. Udupa, Brain tissue MR-image segmentation via optimum-path forest clustering, Comput. Vis. Image Underst. 116 (10) (2012) 1047–1059, doi:10.1016/j.cviu.2012.06.002.
[7] H. Chang, D.Y. Yeung, Robust path-based spectral clustering, Pattern Recognit. 41 (1) (2008) 191–203, doi:10.1016/j.patcog.2007.04.010.
[8] X. Chang, F. Nie, Y. Yang, H. Huang, A convex formulation for semi-supervised multi-label feature selection, in: Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence, AAAI'14, AAAI Press, 2014, pp. 1171–1177.
[9] X. Chang, Y. Yang, Semi-supervised feature analysis by mining correlations among multiple tasks, CoRR (2015). arXiv:1411.6232.
[10] A. Clare, R.D. King, Knowledge discovery in multi-label phenotype data, in: L.D. Raedt, A. Siebes (Eds.), Principles of Data Mining and Knowledge Discovery (PKDD), Lecture Notes in Computer Science, vol. 2168, Springer, 2001, pp. 42–53, doi:10.1007/3-540-44794-6_4.
[11] R. Collobert, F. Sinz, J. Weston, L. Bottou, Large scale transductive SVMs, J. Mach. Learn. Res. 7 (2006) 1687–1712. http://dl.acm.org/citation.cfm?id=1248547.1248609.
[12] T.H. Cormen, C. Stein, R.L. Rivest, C.E. Leiserson, Introduction to Algorithms, 2nd ed., McGraw-Hill Higher Education, 2001.
[13] K.A.P. Costa, L.A. Pereira, R.Y.M. Nakamura, C.R. Pereira, J.P. Papa, A.X. Falcão, A nature-inspired approach to speed up optimum-path forest clustering and its application to intrusion detection in computer networks, Inf. Sci. 294 (10) (2015) 95–108, doi:10.1016/j.ins.2014.09.025.
[14] J. Demšar, Statistical comparisons of classifiers over multiple data sets, J. Mach. Learn. Res. 7 (2006) 1–30. http://www.jmlr.org/papers/volume7/demsar06a/demsar06a.pdf.
[15] A.X. Falcão, J. Stolfi, R. de Alencar Lotufo, The image foresting transform: theory, algorithms, and applications, IEEE Trans. Pattern Anal. Mach. Intell. 26 (1) (2004) 19–29, doi:10.1109/TPAMI.2004.1261076.
[16] M. Friedman, A comparison of alternative tests of significance for the problem of m rankings, Ann. Math. Stat. 11 (1) (1940) 86–92, doi:10.1214/aoms/1177731944.
[17] 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. http://proceedings.mlr.press/v5/goldberg09a/goldberg09a.pdf.
[18] A. Iosifidis, A. Tefas, I. Pitas, Regularized extreme learning machine for multi-view semi-supervised action recognition, Neurocomputing 145 (2014) 250–262, doi:10.1016/j.neucom.2014.05.036.
[19] 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. http://dl.acm.org/citation.cfm?id=645528.657646.
[20] F. Kang, R. Jin, R. Sukthankar, Correlated label propagation with application to multi-label learning, in: 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06), vol. 2, 2006, pp. 1719–1726, doi:10.1109/CVPR.2006.90.
[21] A. Lomsadze, V. Ter-Hovhannisyan, Y.O. Chernoff, M. Borodovsky, Gene identification in novel eukaryotic genomes by self-training algorithm, Nucleic Acids Res. 33 (20) (2005) 6494–6506, doi:10.1093/nar/gki937.
[22] G. Madjarov, D. Kocev, D. Gjorgjevikj, S. Deroski, An extensive experimental comparison of methods for multi-label learning, Pattern Recognit. 45 (9) (2012) 3084–3104, doi:10.1016/j.patcog.2012.03.004.
[23] D. McClosky, E. Charniak, M. Johnson, Effective self-training for parsing, in: Proceedings of the Main Conference on Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics, HLT-NAACL '06, Association for Computational Linguistics, Stroudsburg, PA, USA, 2006, pp. 152–159, doi:10.3115/1220835.1220855.
[24] J.P. Papa, A.X. Falcão, A new variant of the optimum-path forest classifier, in: Advances in Visual Computing, vol. LNCS 5358, Springer, 2008, pp. 935–944, doi:10.1007/978-3-540-89639-5_89.
[25] J.P. Papa, A.X. Falcão, V.H.C. Albuquerque, J.M.R.S. Tavares, Efficient supervised optimum-path forest classification for large datasets, Pattern Recognit. 45 (1) (2012) 512–520, doi:10.1016/j.patcog.2011.07.013.
[26] 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, doi:10.1002/ima.v19:2.
[27] J.P. Papa, S.E.N. Fernandes, A.X. Falcão, Optimum-path forest based on k-connectivity: theory and applications, Pattern Recognit. Lett. 87 (2017) 117–126, doi:10.1016/j.patrec.2016.07.026.
[28] L.A.M. Pereira, J.P. Papa, J. Almeida, R.d.S. Torres, W.P. Amorim, A multiple labeling-based optimum-path forest for video content classification, in: Proceedings of the 26th SIBGRAPI Conference on Graphics, Patterns and Images, 2013, pp. 334–340, doi:10.1109/SIBGRAPI.2013.53.
[29] 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, doi:10.1002/ima.v19:2.

[30] C. Rosenberg, M. Hebert, H. Schneiderman, Semi-supervised self-training of object detection models, in: Proceedings of Application of Computer Vision, 2005. WACV/MOTIONS '05. Seventh IEEE Workshops on, vol. 1, 2005, pp. 29–36, doi:10.1109/ACVMOT.2005.107.
[31] Q. Wu, M. Tan, H. Song, J. Chen, M.K. Ng, ML-FOREST: a multi-label tree ensemble method for multi-label classification, IEEE Trans. Knowl. Data Eng. 28 (10) (2016) 2665–2680, doi:10.1109/TKDE.2016.2581161.
[32] M.L. Zhang, Z.H. Zhou, Multilabel neural networks with applications to functional genomics and text categorization, IEEE Trans. Knowl. Data Eng. 18 (10) (2006) 1338–1351, doi:10.1109/TKDE.2006.162.
[33] M.L. Zhang, Z.H. Zhou, ML-KNN: a lazy learning approach to multi-label learning, Pattern Recognit. 40 (7) (2007) 2038–2048, doi:10.1016/j.patcog.2006.12.019.
[34] M.L. Zhang, Z.H. Zhou, A review on multi-label learning algorithms, IEEE Trans. Knowl. Data Eng. 26 (8) (2014) 1819–1837, doi:10.1109/TKDE.2013.39.
[35] Y. Zhang, D.Y. Yeung, Semi-supervised discriminant analysis using robust path-based similarity, in: 2008 IEEE Conference on Computer Vision and Pattern Recognition, 2008, pp. 1–8, doi:10.1109/CVPR.2008.4587357.