Pattern Recognition ■ (■■■■) ■■■–■■■

Contents lists available at ScienceDirect

Pattern Recognition

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

Robust multiple-instance learning ensembles using random subspace instance selection

Marc-André Carbonneau [a,b,*], Eric Granger [b], Alexandre J. Raymond [a], Ghyslain Gagnon [a]

[a] Laboratoire de communications et d'intégration de la microélectronique (LACIME), École de technologie supérieure, Université du Québec, 1100, rue Notre-Dame Ouest, Montréal (Qc), Canada H3C 1K3

[b] Laboratoire d'imagerie, de vision et d'intelligence artificielle (LIVIA), École de technologie supérieure, Université du Québec, 1100, rue Notre-Dame Ouest, Montréal (Qc), Canada H3C 1K3

ARTICLE INFO

Article history:
Received 10 August 2015
Received in revised form
3 February 2016
Accepted 11 March 2016

Keywords:
Multiple-instance learning
Random subspace methods
Classifier ensembles
Instance selection
Weakly supervised learning
Classification
MIL

ABSTRACT

Many real-world pattern recognition problems can be modeled using multiple-instance learning (MIL), where instances are grouped into bags, and each bag is assigned a label. State-of-the-art MIL methods provide a high level of performance when strong assumptions are made regarding the underlying data distributions, and the proportion of positive to negative instances in positive bags. In this paper, a new method called Random Subspace Instance Selection (RSIS) is proposed for the robust design of MIL ensembles without any prior assumptions on the data structure and the proportion of instances in bags. First, instance selection probabilities are computed based on training data clustered in random subspaces. A pool of classifiers is then generated using the training subsets created with these selection probabilities. By using RSIS, MIL ensembles are more robust to many data distributions and noise, and are not adversely affected by the proportion of positive instances in positive bags because training instances are repeatedly selected in a probabilistic manner. Moreover, RSIS also allows the identification of positive instances on an individual basis, as required in many practical applications. Results obtained with several real-world and synthetic databases show the robustness of MIL ensembles designed with the proposed RSIS method over a range of witness rates, noisy features and data distributions compared to reference methods in the literature.

1. Introduction

Multiple-instance learning (MIL) is a form of weakly-supervised learning [1], where data *instances* are grouped into *bags*. A label is not provided for each instance, but for a whole bag. Typically, a negative bag contains only negative instances, while positive bags contain instances from both classes [2].

Since the first formulations of the MIL problem [2,3] many solutions have been proposed. In many cases, MIL algorithms were developed with a specific application in mind. For instance, Dietrich [2] proposed Axis Parallel Rectangle (APR) to solve a molecule classification problem. Later, many methods were proposed to solve image categorization [4–8], web mining [9,10], object and face detection [11–15] and tracking [16] problems. While they can achieve a high level of performance in their respective application domains, many of these methods are less efficient over a wide variety of data distributions and pattern classification problems.

For instance, many methods rely on the assumption that the proportion of positive instances in positive bags, hereafter called *witness rate*, is high. Sometimes, these methods implicitly assume that all instances in a positive bag are positive. This is the case for methods such as APR [2], Citation-kNN [17] and diverse density-based (DD) methods [5,6,18,19]. This assumption is also made in the initialization of the optimization process in mi-SVM and MI-SVM [4]. Other methods assume a high witness rate by representing bags as the average of the instances it contains, as in MI-Kernel [20] and MIBoosting [21]. The performance of all these methods decreases when the high witness rate assumption is not verified, which limits the applicability of MIL methods to many problems. For instance, until recently, object identification systems were limited to problems where instances represent slight translational and scale uncertainties around localization bounding boxes [15].

* Corresponding author.
E-mail addresses: marcandre.carbonneau@gmail.com (M.-A. Carbonneau), eric.granger@etsmtl.ca (E. Granger), alexandre.raymond@lacime.etsmtl.ca (A.J. Raymond), ghyslain.gagnon@etsmtl.ca (G. Gagnon).

http://dx.doi.org/10.1016/j.patcog.2016.03.035
0031-3203/

To deal with lower witness rates, Gehler and Chapelle [22] applied deterministic annealing to an SVM-based MIL algorithm. Bunescu and Mooney [23] enforced the constraint that positive bags contain at least one positive instance in their SVM formulation. Both obtained good results with lower witness rates, but observed performance degradation with higher witness rates. SVR-SVM [24] and the γ-rule [25] have been proposed to overcome these problems by estimating the witness rate and then using it as a system parameter. These techniques provide a high level of performance over a range of high and low witness rates, yet, the witness rate is assumed to be constant across all bags. This assumption proves to be problematic in some applications, such as image categorization [26], where images are segmented and features are extracted from the different segments [4,5]. The image corresponds to a bag, while each segment is an instance. Depending on the visual complexity of the image, a different proportion of target and non-target segments will be obtained. Therefore, the witness rate of a bag depends on the image content, and is likely to vary from one bag to another.

Another challenge of MIL problems is the fact that the shape of positive and negative distributions affect the performance of some algorithms. For instance, some methods such as APR [2] are not designed to deal with multi-modal distributions where instances are grouped in distinct clusters. Methods based on DD [5,6,18,19] assume that positive instances form a compact cluster [7]. In MILIS [7], the negative distribution is modeled with Gaussian kernels, which can be difficult when the quantity of data available is limited. On the other hand, in Citation-kNN [17] the presence of compact data cluster in the negative distribution increases the probability of misclassification.

Finally, some methods classify bags as a whole instead of trying to label each instance individually. Some of these methods [17,20,27,28] use different types of bag distance measure, while others embed bags using distance to a set of prototypes [6,7,5], vocabulary [29] and sparse coding [30]. Bag-level classification approaches cannot identify instances individually, which is necessary in certain applications such as object detection and tracking in images or videos. Moreover, by considering bags as a whole, the performance of these methods often decreases in problems where the witness rate is low.

To address these limitations, this paper proposes a new ensemble-based method for MIL called Random Subspace Instance Selection (RSIS). Classifier ensembles are generally known to provide accurate and robust classification systems when data is limited [31]. The key feature of RSIS is that it constructs classifier ensembles based on a probabilistic identification of positive instances. The proposed method allows to classify instances individually and does not rely on a specific witness rate or specific type of data distribution. It can therefore be applied in a wide variety of context.

In the proposed method, the training data is projected onto several random subspaces before being clustered. The proportion of instances from positive and negative bags is computed for every cluster. Based on these bag proportions, a *positivity score* is computed for every instance in the data set. These scores are later converted into selection probabilities, and used to select diverse training sets to generate base classifiers in the ensemble. The general intuition for RSIS is that it is easier to identify positive instance clusters while only considering a discriminant subset of features. The optimal feature subset to represent a given concept is unknown, and may vary from one concept to another. However, if a data set is projected into all possible subspaces, instances from the same concept are more likely to be grouped together than with the other instances.

The RSIS method allows to design MIL ensembles that are robust to various witness rate, because each time one of the classifiers in the ensemble is trained, only one instance is used from each bag. The instances are drawn based on their probability of being positive. If the witness rate is low and only one instance is likely to be positive, this instance will be the only one selected. In contrast, if many instances appear to be positive, each instance will have a similar probability of being selected, and thus being used as a training instance in one or another classifier. Since selection probabilities are computed for each bag separately, the witness rate does not have to be constant across all bags. Moreover, by clustering the data in many different subspaces, RSIS can inherently uncovers multiple underlying concepts in the data distributions. This makes the algorithm resistant to multi-modal distributions of various shapes, and robust to noisy or irrelevant features.

In this paper, the performance of MIL ensembles designed using RSIS is compared to several methods in the literature using benchmark data sets. Further experiments are performed on synthetic data sets to study the algorithm's tolerance to various multi-modal distributions, witness rate and irrelevant features. Five well-known baseline methods, APR [2], Citation-kNN [17], mi-SVM [4], AL-SVM [22] and CCE [32] are also used for comparison. Finally, the sensitivity of the proposed approach to internal parameters is also characterized experimentally, and some general guidelines for parameter selection are provided.

The remainder of this paper is organized as follows. The MIL problem is formalized and state-of-the-art techniques are reviewed in Section 2. Then, in Section 3, the proposed RSIS algorithm is described. Section 4 presents the experimental methodology. Section 5 presents robustness experiments on synthetic data, while Sections 6 and 7 present experimental results on benchmark data sets, and experiments on parameter sensitivity respectively. Time complexity is discussed in Section 8.

2. Multiple instance learning

Let $\mathcal{B} = \left\{B^1, B^2, \ldots, B^Z\right\}$ be a set composed of Z bags.[1] Each bag B^i corresponds to a positive or negative label $L^i \in \{-1, +1\}$ in the set $\mathcal{L} = \left\{L^1, L^2, \ldots, L^Z\right\}$, and contains N^i feature vectors: $B^i = \left\{\mathbf{x}^i_1, \mathbf{x}^i_2, \ldots, \mathbf{x}^i_{N^i}\right\}$ where $\mathbf{x}^i_j = (x^i_{j1}, x^i_{j2}, \ldots, x^i_{jd}) \in \mathbb{R}^d$. Each of these feature vector instances corresponds to a positive or negative label in the set $Y^i = \left\{y^i_1, y^i_2, \ldots, y^i_{N^i}\right\}$, where $y^i_j \in \{-1, +1\}$. Instance labels are unknown in positive bags, but are assumed negative in negative bags. A bag is labeled positive if at least one instance contained in the bag is labeled positive [2]:

$$L^i = \begin{cases} +1 & \text{if } \exists y \in Y^i : y^i_j = +1; \\ -1 & \text{if } \forall y \in Y^i : y^i_j = -1. \end{cases} \quad (1)$$

Many methods have been proposed over the years to address MIL problems in a variety of domains. An overview of these methods and a review of the MIL assumptions can be found in recent surveys by Amores [33] and Foulds and Frank [34]. In the taxonomy proposed by Amores, [33] MIL methods are divided in three categories, based on how bags are represented. A first corpus of methods operates at the instance level. Each instance is classified individually, and scores are aggregated to label bags. The two other types of method operate on the bag level. In one case, bags are mapped to a vector representation, which reformulate the MIL problem as a standard supervised classification problem, while in

[1] Throughout this paper, upper indexes are used to denote bags, while lower indexes designate instances. For the sake of clarity, when unnecessary, these indexes are omitted.

Please cite this article as: M.-A. Carbonneau, et al., Robust multiple-instance learning ensembles using random subspace instance selection, Pattern Recognition (2016), http://dx.doi.org/10.1016/j.patcog.2016.03.035

the other case, distance metrics are proposed to compare whole bags.

The proposed method falls in the instance-level category. When operating at this level, it is not only possible to categorize bags but also to identify positive instances in bags individually. This is necessary in some application such as object detection and tracking applications [6,11–16]. There exists many instance-level techniques in the literature, starting with APR, proposed as early as 1997 by Dietterich et al. [2]. In this method, an hyper-rectangle is expanded and shrunk to maximize the number of instances from positive bags, while minimizing the number of instances from negative ones. Instances falling inside the hyper-rectangle are considered positive, while others are labeled negative. APR considers all instances in positive bags to be positive, and, thus, assumes a high witness rate. Also, the use of an hyper-rectangle as a single classification region implies the assumption that positive instances come from a single cluster in space.

Later, Maron and Lorenzo-Pérez proposed to use a measure called diverse density (DD) [18]. The DD of a location in feature space is high if its neighborhood contains many instances from different positive bags and few from negative bags. Later, with EM-DD, Zhang and Goldman [19] proposed to use the Expectation-Maximization algorithm to search for the maxima of the DD function. DD-based methods work under the assumption that the positive data comes from a compact clusters in feature space [7], which limits their applicability in many problems. Also, DD and EM-DD performance decreases with number of relevant features [19].

In some methods bags are represented by averaging the instances they contain. In MI-Kernel [20], a bag is summarized by a normalized sum of the instances it contains. In MILBoost [21] the probability of a bag being positive is obtained by averaging the probabilities of each instance it contains. By pooling all instances together, these methods assume a high witness rate.

Many max-margin classifiers were proposed for MIL problems. These methods were recently surveyed and analyzed by Doran and Ray [35]. Andrews et al. [4] were among the firsts to extend SVMs to solve MIL problems. Two algorithms were proposed: mi-SVM and MI-SVM. In mi-SVM, the margin is maximized jointly over instance label assignations and a discriminant function. Every instance found in a positive bag is initialized as positive. The SVM is first trained based on these assignments. The resulting classifier is then used on the same training data to update the instance labels. Next, the SVM is trained based on the new label assignments, and so forth. The second algorithm, MI-SVM, focuses on maximizing the margin over the bags instead of instances by choosing a single instance to represent bags. MICA works similarly but selects a convex combination of witnesses to represent bags [36]. By initializing all instance labels in positive bags as positive, these methods rely on the assumption that the witness rate is high.

To deal with lower witness rates, Gehler and Chapelle [22] applied deterministic annealing to the aforementioned SVM-based MIL algorithms. With Sparse-MIL, Bunescu and Mooney [23] proposed to enforce the constraint that there is at least one positive instance in each positive bag in a transductive SVM formulation. Both methods obtain a high level of performance at low witness rates, but observe performance degradation at higher witness rates.

To address the performance dependency to specific witness rates, Li and Sminchisescu proposed SVR-SVM [24]. In SVR-SVM, the MIL problem is formulated as a convex joint estimation of the likelihood ratio function and the likelihood ratio values on training instances. They obtained high level of performance at high and low witness rate, but assumed the witness is constant across all bags.

Chen and Wang [5] used DD and SVM to embed and classify bags. DD-SVM selects multiple instance prototype corresponding to local maxima of the DD response function. Bags are represented by distance from these prototypes. This idea was later used in MILES [6], except that instances from the training set are used, instead of prototype, to embed bags. While yielding high level of performance, the method does not scale well to large problems, since the dimension of bag feature vectors depends on the number of training instances in the data set [7]. Fu et al. [7] proposed MILIS to minimize this problem, with an initial selection of the prototype instances via several runs of EM-DD.

Zhou and Zhang proposed CCE [32], an algorithm based on clustering and classifier ensembles. Training data is clustered, and the bags are represented as binary vectors in which each bit corresponds to a cluster. A bit is set to 1 if at least one instance of the bag is attributed to its corresponding cluster. To design the ensemble, several clusterings are performed and a classifier is trained using each different data representation. This method represents whole bags based on clustering results, while with ensembles created with RSIS classify instances individually in the original feature space.

Other ensemble methods have been proposed to solve MIL problems. For instance, many authors proposed variations of boosting for object detection [11,15,21], while others proposed to combine different classifiers [37]. Li et al. proposed the γ-rule for classifier combination in MIL contexts [25]. They assume that instances in data sets can be modeled as a mixture of concept and non-concept distributions. Once estimated, the mixture is used to re-weigh the posteriors of classifiers. In this method, the witness rate is estimated, and is assumed to be constant across all bags.

Finally, some methods, like Citation-kNN (CKNN) proposed by Wang and Zucker [17], operate at the bag level. This method is inspired by the notion of citations in research. For a given bag b, the r nearest *references* correspond to the r nearest bags, using the Hausdorff distance. The nearest *citers* are the bags that count b in their c nearest bags. The label of bag b is obtained by a majority vote on the *reference* bags and *citers* bags pooled together. Many other methods use bag distance measures such as the dissimilarity measure [27], or the graph kernels [28].

For most of these methods, strong assumptions have been made implicitly or explicitly regarding the witness rate and the data distribution. When very little is known about the nature of the data and the content of the bags, selecting a robust MIL method can be difficult. The proposed RSIS method presented in Section 3 is a general method that allows to design discriminant MIL ensembles without prior assumptions regarding witness rate and data distributions. Classifier ensembles are known to handle complex data structures and to provide better generalization and accuracy than single classifier systems [31]. Moreover, because the proposed method classifies instances individually, it can be used in MIL problems like object tracking and detection for which bag-based methods cannot be used.

3. Random subspace instance selection for MIL ensembles

The basic steps of the proposed approach for MIL ensemble design using RSIS are represented in Fig. 1. At first, each instance receives a *positivity score* based on clustering of data in random subspaces, which indicates the likelihood that an instance is positive. The computation of these scores is described in Section 3.1. Given these scores, an instance selection probability distribution is obtained for each bag. To generate a diverse pool of base classifiers, each one is trained on a different subset of the training data, where each subset contains one instance from each positive bags and instances from the negative bags. These instances are

Please cite this article as: M.-A. Carbonneau, et al., Robust multiple-instance learning ensembles using random subspace instance selection, Pattern Recognition (2016), http://dx.doi.org/10.1016/j.patcog.2016.03.035

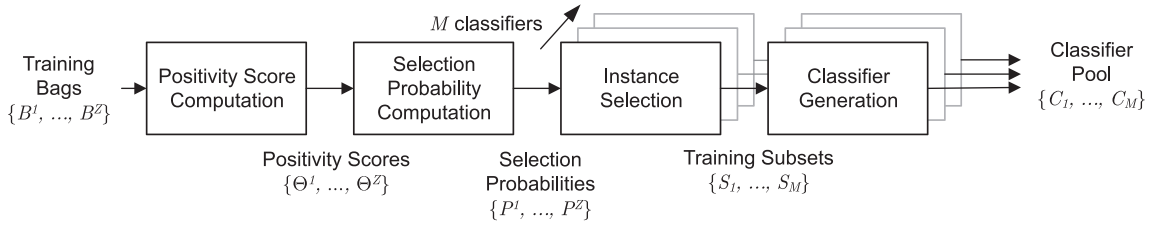


Fig. 1. MIL ensemble design using the proposed RSIS technique.

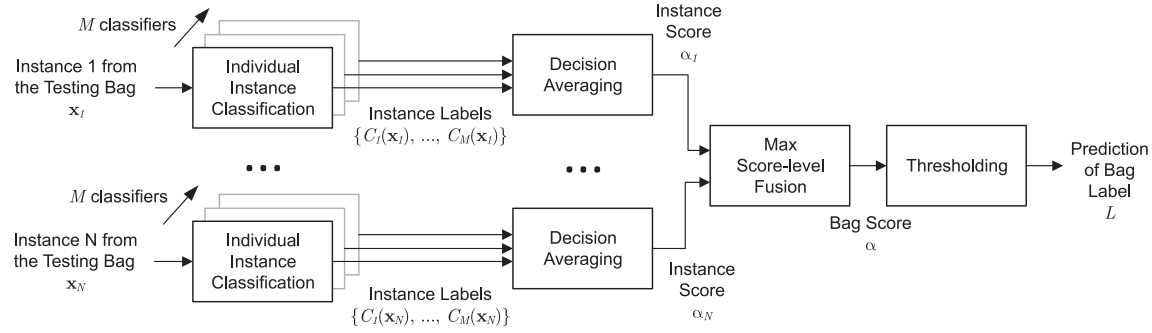


Fig. 2. Bag label prediction using MIL ensemble.

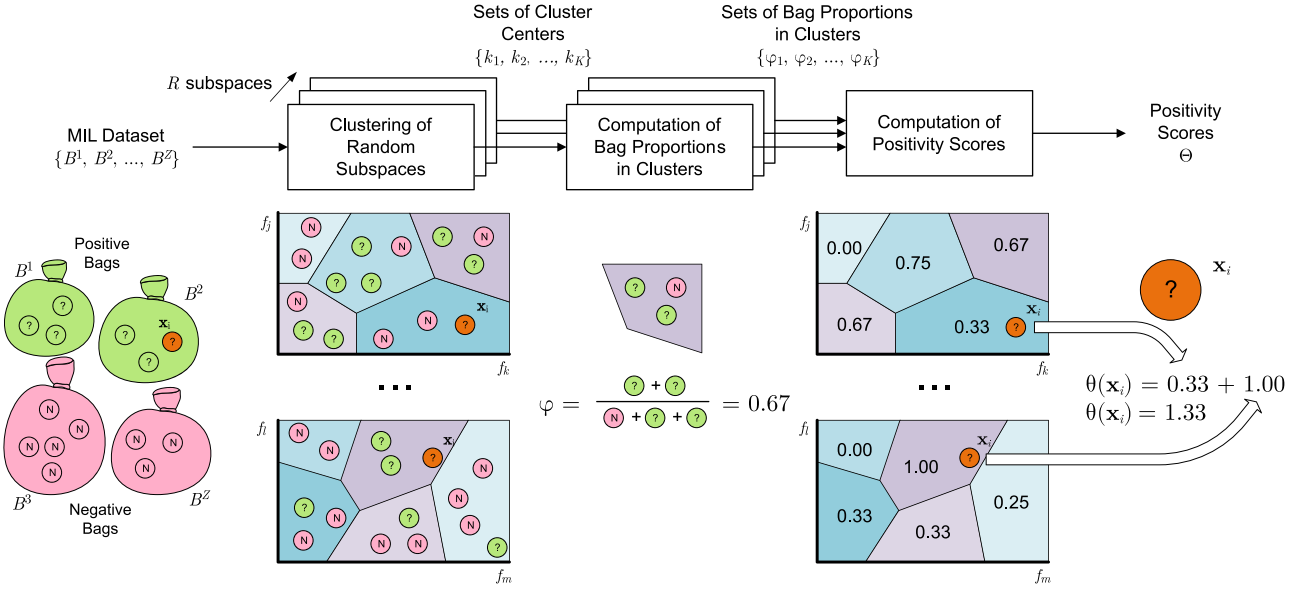


Fig. 3. Illustration of the pipeline to compute positivity scores with RSIS.

randomly selected based on the previously computed instance selection probability distribution. This process may be viewed as a variation on bagging [38], with the novelty that subset sampling is guided by the positivity scores. Ensemble design is detailed in Section 3.2.

As depicted in Fig. 2, when an unknown test bag is presented to the system during operation, each classifier predicts a label for each instance. The decisions of the classifiers are averaged to produce a score for each instance. The highest instance score is attributed to the bag, and this bag score is compared to a threshold for final prediction of class label. Bag classification is described in Section 3.3.

3.1. Positivity score computation

The computation of positivity scores is illustrated in Fig. 3 and summarized in Algorithm 1. The first step consists in randomly selecting p features from the complete set of d features to create a subspace $\mathcal{P}$. If $\mathcal{F}$ is the complete space, then $\mathcal{P} \subseteq \mathcal{F}$.

Every instance $\mathbf{x}$ from each bag B^i is projected onto the subspace $\mathcal{P}$. A clustering of this space is then performed. Next, the proportion φ_n of instances belonging to positive bags is computed for each cluster k_n, where $n = 1, 2, \ldots, K$:

$$\varphi_n = \frac{\sum_{\forall \mathbf{x}} c(\mathbf{x}^i, n)}{|\mathcal{K}_n|} \in [0..1], \tag{2}$$

where:

$$c(\mathbf{x}^i, n) = \begin{cases} 1, & \text{if } \mathbf{x}^i \in \mathcal{K}_n \quad \text{and} \quad L^i = +1; \\ 0, & \text{otherwise.} \end{cases} \tag{3}$$

In these equations, $\mathcal{K}_n$ is the set of instances belonging to cluster k_n, and $|\mathcal{K}_n|$ is the size of this set.

The complete process of selecting a random subspace, projecting the data into the subspace and clustering the projected

data is repeated R times. At the end of repetition $r = 1, 2, \ldots, R$, each instance $\mathbf{x}$ receives the positive bag proportion $\varphi_n(r)$ of its cluster assignment. The values from all repetitions are summed in order to get a positivity score set $\Theta^i = \left\{\theta_1^i, \theta_2^i, \ldots, \theta_{N^i}^i\right\}$ in which each value corresponds to an instance in the data set:

$$\theta(\mathbf{x}) = \frac{1}{R}\sum_{r=1}^{R}\sum_{n=1}^{K}\varphi_n(r)\cdot d(\mathbf{x}, n, r), \tag{4}$$

where

$$d(\mathbf{x}, n, r) = \begin{cases} 1, & \text{if } \mathbf{x} \in \mathcal{K}_n \text{ at repetition } r; \\ 0, & \text{otherwise.} \end{cases} \tag{5}$$

Positivity scores indicate the likelihood that the instances belong to the positive class. In positive bags, these scores indicate the most likely positive instances, while in negative bags, they allow to rank instances according to classification difficulty.

Algorithm 1. Computation of positivity $\theta(\mathbf{x})$ score for each instance $\mathbf{x}$.

Data: Training set $\mathcal{B}$
Result: Positivity score set Θ
for $r = 1$ *to* R *subspaces* **do**
  randomly select a $p-$feature subspace $\mathcal{P}$;
  project all instances in $\mathcal{B}$ onto subspace $\mathcal{P}$;
  perform clustering of projected data using K cluster centers;
  for $n = 1$ *to* K *clusters* **do**
    compute $\varphi_n(r)$ using Eq. (2);
  end
end
for $\forall \mathbf{x} \in \mathcal{B}$ **do**
  compute score $\theta(\mathbf{x})$ for using Eq. (4);
end
return positivity score set Θ;

3.2. Ensemble design

Each classifier in the pool $\mathcal{C} = \{C_1, C_2, \ldots, C_M\}$ maps instances to binary hard labels: $C : \mathbb{R}^d \rightarrow \{0, 1\}$. Each classifier is trained on a different data subset $\mathcal{S}_p$ composed of instances selected based on the positivity scores Θ^i (see Eq. (4) in Section 3.1). At this point, the domain of the instances is the entire feature space. In each bag, these scores are converted to selection probabilities by applying a soft-max function on all instances it contains, and one instance $\mathbf{x}_*$ is selected per bag:

$$P(\mathbf{x}_* = \mathbf{x}_k \mid \Theta) = \frac{e^{\theta_k/T}}{\sum_{j=1}^{N_i} e^{\theta_j/T}}, \tag{6}$$

where $T \in \mathbb{R}^+$ is the temperature parameter. The training subset is created by choosing one instance from each bag based on the selection probabilities. The label of the selected instances corresponds to the label of their bags ($y_j^i = L^i$).

Finally, classification performance can be enhanced by adding randomly selected instances from negative bags to the training subsets.

Algorithm 2. Generation of classifier pools with the RSIS method.

Data: Training set $\mathcal{B} = \left\{B^1, B^2, \ldots, B^Z\right\}$
Result: Classifier pool $\mathcal{C}$
initialize $\mathcal{C} = \varnothing$;
compute positivity scores (see Algorithm 1);
compute selection probabilities $P(\cdot \mid \Theta)$ using Eq. (6);
for $i = 1$ *to* M **do**
  $\mathcal{S} = \varnothing$;
  for $j = 1$ *to* Z **do**
    select one instance $\mathbf{x}_*^j$ using $P(\cdot \mid \Theta^j)$;
    add it to the training subset $\mathcal{S}$;
  end
  add randomly selected instances from negative bags to $\mathcal{S}$;
  train classifier C_i using $\mathcal{S}$;
  add C_i to pool $\mathcal{C}$;
end
return classifier pool $\mathcal{C}$;

3.3. Prediction of bag labels

During operation, each unknown test instance is classified individually, and a bag is deemed positive when it contains a positive instance. Formally, the label L of a bag B is given by:

$$L = \begin{cases} +1, & \text{if } \alpha > \beta; \\ -1, & \text{otherwise,} \end{cases} \tag{7}$$

where β is a threshold set empirically on validation data, and $\alpha \in [0, 1]$ is the averaged outputs of the classifiers for the *most positive* instance in the bag:

$$\alpha = \max_{\mathbf{x} \in B}\left\{\frac{1}{M}\sum_{j=1}^{M} C_j(\mathbf{x})\right\}. \tag{8}$$

In applications such as tracking and object recognition, labeling bags is not sufficient. The algorithm must identify which instances in the bag are the most likely to be positive. Using the proposed algorithm, this translates to simply ranking and selecting the instance $\hat{\mathbf{x}}$ with the highest score in a positive bag if only one instance is needed:

$$\hat{\mathbf{x}} = \arg\max_{\mathbf{x} \in B}\left\{\frac{1}{M}\sum_{j=1}^{M} C_j(\mathbf{x})\right\}. \tag{9}$$

In applications where more than one instance needs to be selected, the threshold β is applied to each instance, as performed for bags in Eq. (7).

3.4. Why it works

In essence, ensemble design with RSIS is akin to Bagging [38]. There are theoretical and experimental evidences that Bagging pushes unstable classification procedures, such as classification trees and neural nets, towards optimality in prediction [38]. Ensembles created through some Bagging procedure consist of classifiers trained with different subsets of instances. In supervised learning problems, any instances can be randomly selected. However, in MIL problems, blind selection of the instances may result in poor classifier and ensemble performance. This is because negative instances may be used as positive instances, which would introduce noise into the training data. For example, if the witness rate is as high as 50%, half of the training instances of a class are incorrectly labeled. Integrating a positive instance identification and selection mechanism into the ensemble design procedure, is a key idea of the proposed RSIS algorithm. The remainder of the section presents an analysis of the positive instance identification process in RSIS.

Let us consider data with two underlying concepts (one positive and one negative) that do not overlap in the input feature space. In an ideal case, the clustering process would result in two distinct clusters, each corresponding to a concept. In MIL problems, the cluster corresponding to the negative concept will

Please cite this article as: M.-A. Carbonneau, et al., Robust multiple-instance learning ensembles using random subspace instance selection, Pattern Recognition (2016), http://dx.doi.org/10.1016/j.patcog.2016.03.035

contain instances from the positive and negative bags, where the proportion of instances from positive bags (see Eq. (2)) depends on the witness rate and the proportion of positive bags in the data set:

$$\varphi- = (1-WR)\frac{\|\mathcal{B}^+\|}{\|\mathcal{B}\|} \tag{10}$$

Since positive instances cannot be found in negatives bags, the proportion of instances from positive bags in the positive cluster will be 1 ($\varphi_+ = 1$). In this simple example with ideal clustering, the positivity score of a negative instance (see Eq. (4)) is given by $\theta(\mathbf{x}_-) = \varphi_- < 1$, while the positivity score of a positive instance is $\theta(\mathbf{x}_+) = 1$, therefore $\theta(\mathbf{x}_+) > \theta(\mathbf{x}_-)$. Data subsets used to train the classifiers are constructed based on these scores. By using a very low temperature parameter ($T \to 0$) in Eq. (6), all of the instances selected as positive example will necessarily belong to the positive concept. Furthermore, the negative instances belonging to a positive bag will not be selected. In this ideal case, the value of the witness rate and the proportion of positive bags are of no consequence for positive instance identification.

The assumptions made in the previous example rarely hold in practice. First of all, the result of clustering algorithm is rarely perfect, and the data is not always grouped in distinct clusters. Assuming negative instances of the negative and positive bags come from the same distribution, the worst case clustering would equally distribute the real positive instances between all clusters. In that case, if the data set size tends to infinity, in all clusters, the proportion of instances from positive bags is given by:

$$\varphi\big|_{Z\to\infty} = \frac{\|\mathcal{B}^+\|}{\|\mathcal{B}\|} \tag{11}$$

In this worst case clustering result, the contribution to positivity scores is the same for all instances. Thus, this has no impact on the instance selection probabilities, except for the optimal temperature setting. However, if a clustering happens to group positive instances together, the proportion of instances from positive bags (φ) of each cluster may improve discrimination between positive and negative instances. In RSIS, the data is projected in a number of subspace, and then clustered. Thus, different clustering results are obtained, which are either informative or at worst, do not provide useful information. Thus, as the number of clustered random subspace increases, the positive instances tend to be identified more accurately. This is observed in results of Section 7 concerning parameter sensitivity. In Fig. 8(c), one can see performances increase (or remain stable) as the number of generated subspaces increases.

4. Experimental setup

Three different experiments were conducted to assess RSIS performance. In the first experiment, MIL ensembles designed with RSIS are compared to five well-known reference MIL classification methods on synthetic data sets. The experiment is designed to measure the algorithms robustness to various witness rates, data distributions and noisy features. In the second experiment, an ensemble based on RSIS is compared to 29 other state-of-the-art MIL methods on real-world benchmark data sets: the two Musk data sets [2] and the Tiger, Elephant and Fox data sets [4]. Finally, the third experiment studies the impact of RSIS parameters on the MIL ensemble performance.

4.1. Data sets

4.1.1. Benchmark data sets

Drug activity prediction: The Musk data sets are the most widely used benchmarks for MIL classifier performance evaluation. These data sets were introduced by Dietterich et al. [2] and are both publicly available from the UCI Machine Learning repository.[2] In this data set, each bag corresponds to a type of molecule, and each instance corresponds to a low-energy conformation of this molecule. The task consists in determining if a molecule is musky or not. For the same molecule, not all conformations are musky, hence comes the MIL problem formulation. Each molecule conformation is described by a 166-dimensional vector. The second data set contains many more instances, mostly negative. Table 1 summarizes the two data sets.

Tiger, Elephant and Fox: These three data sets come from the COREL data set [4]. The bags in these data sets correspond to animal images. In each data set, there are 100 images of a target animal and 100 images of other random animals. An image corresponds to a bag and the segments in the image are instances. Each instance is described by a 230-dimensional feature vector containing shape, color and texture information. The data set is also publicly available[3] and summarized in Table 1.

Some papers [24,22,25] include an estimation of the witness rate for the most popular benchmark data sets. These estimations are reported in Table 1, and suggest that, in most of these data sets, a large portion of instances in positive bags are positive. This biases results towards methods that classify bags as a whole instead of individual instances [25]. Also, some methods need a high witness rate to perform well. In order to assess the performance of the proposed RSIS technique with a low witness rate, the Newsgroups benchmark data set [28] is also used as a benchmark. Finally, we created a new synthetic data set allowing control over witness rate, shape of the data distribution and the proportion of noisy features.

Newsgroups: This set was derived by Zhou et al. [28] from the 20 Newsgroups [39] data set corpus. The set contains posts from newsgroups on various subjects. Each bag contains 50 posts from the 20 news categories. In positive bags, 3% of posts belongs to the target class while the other posts are uniformly drawn from all other classes. Each post is represented by 200 TFIDF features. Because of its low witness rate, the data set has been used to highlight the insensitivity to witness rate of the SVR-SVM [24] method. The data set is publicly available from the same site as the Tiger, Elephant and Fox data sets. The characteristics of the Newsgroups data set are summarized in Table 1. The numbers reported are the average value of all 20 data sets.

4.1.2. Synthetic data

In this data set, different configurations are proposed to assess the performance of the algorithms under different situations. Several parameter configurations are produced with various data distributions, witness rates, number of concepts and number of irrelevant features. The data set is made available publicly.[4]

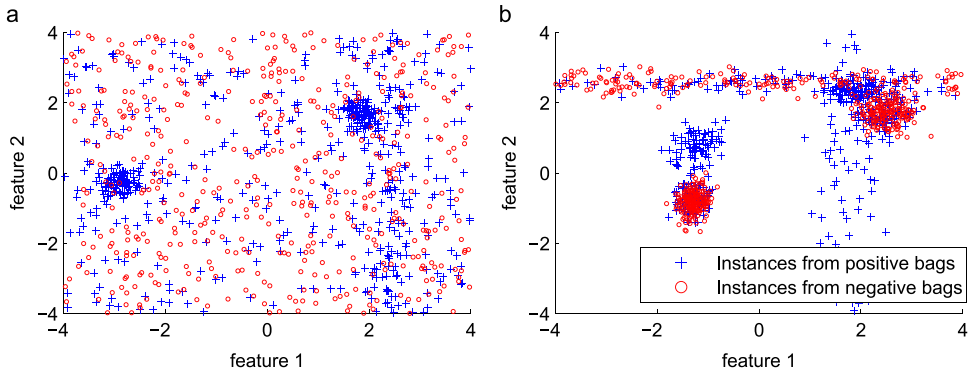
The positive instances are drawn from the concept distribution, while negative instances are drawn either from the uniform distribution $\mathcal{U}(-4,4)$ or from a negative concept distribution. Concept distributions are multivariate Gaussians distributions $\mathcal{G}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$. The values of $\boldsymbol{\mu}$ are drawn from $\mathcal{U}(-3,3)$. The covariance matrix ($\boldsymbol{\Sigma}$) is a randomly generated semi-definite positive matrix in which the diagonal values are scaled to]0, 0.1].

[2] http://archive.ics.uci.edu/ml/
[3] http://www.miproblems.org/mi-learning/
[4] http://www.etsmtl.ca/Professeurs/ggagnon/Projects/ai-MIL

Please cite this article as: M.-A. Carbonneau, et al., Robust multiple-instance learning ensembles using random subspace instance selection, Pattern Recognition (2016), http://dx.doi.org/10.1016/j.patcog.2016.03.035

Table 1
Properties of the benchmark data sets.

Data set	+ Bags	− Bags	Instances	Features	Instances per bag			Witness rate			
					Min.	Max.	Avg.	[25]	[24]	[22]	[28]
Musk1	47	45	476	166	2	40	5	0.82	1.00	1.00	–
Musk2	39	63	6598	166	1	1044	65	0.77	0.90	0.28	–
Tiger	100	100	1220	230	1	13	6	0.51	0.43	0.60	–
Fox	100	100	1302	230	2	13	8	0.88	1.00	0.71	–
Elephant	100	100	1391	230	2	13	7	0.80	0.38	0.58	–
Newsgroups	50	50	4006	200	18	65	40	–	–	–	0.03



Negative instances from uniform distribution.

Negative instances from concept distribution.

Fig. 4. Example distribution from the synthetic data set. In both (a) and (b), 2D samples were randomly generated. In (a), negative instances are sampled from an uniform distribution, while in (b), positive and negative instances are sampled from clustered distributions. Each cluster represents a concept. Marker correspond to bag labels.

In order to model irrelevant features in the data, in each concept, some features are drawn from the uniform distribution instead of the multivariate Gaussian distribution. The number of irrelevant features is controlled by the irrelevant feature proportion (IFP) parameter. For each parameter configuration, the data set is generated 5 times to get results that are more significant. The 10-fold CV procedure is repeated 10 times on each of the 5 generated data sets.

Examples of 2D data distributions are given in Fig. 4. In each distribution, one of the features of a concept is irrelevant, which yields the line-shaped cluster. The negative instance distribution is uniform in (a), while negative instances are grouped in Gaussian clusters in (b).

4.2. Protocol and performance metrics

Experiments were conducted using nested cross-validation (CV) [40] where an inner CV loop is used to select the model parameters, while the outer CV loop is used to estimate the algorithm performance. Both the inner and outer CV loop use 10 folds. At each iteration of the outer loop, a fold is reserved for testing, and model selection is performed via CV grid search on the remaining parts. The best performing configuration is selected by averaging the results obtained on each fold of inner loop CV process. The algorithm is then retrained with the best configuration using all training data, and performance is obtained on the held-out test fold. Results reported in this paper are the average of 10 repetitions of this 10-fold nested CV process.[5] At each repetition, the data is shuffled, and a new fold partitioning is performed.

Five parameters are optimized in the inner loop of the nested CV procedure. In the random subspace selection procedure, there is the number of dimensions of each subspace ($|\mathcal{P}|$), the number of clusters (K) and the number of subspaces (R) generated. When creating the ensemble, the temperature (T) and the number of classifiers (M) in the ensemble also have to be selected. The robustness of the proposed system to these 5 parameters is studied in Section 7. The recommended parameter values of Section 7 are applied in experiments on the Newsgroups data sets. Thus, only two parameters were optimized.

The RSIS procedure does not depend on a particular clustering algorithm or base classifier. In this paper, SVM classifiers are used because of their good performance and versatility when used with kernels. The k-means algorithm is used for clustering because of its low computational complexity. The LIBSVM [41] library was used for the SVM implementation. A set of optimal parameters for the SVM classifiers was determined in a prior experiment by coarse grid-search via cross-validation on each data set. The exponential kernel was used in all experiments. For the synthetic data set, $C=10$ and $\gamma=10^{-1}$. For the Musk data sets, $C=10$ and $\gamma=10^{-6}$. The same settings were used for the Elephant and Tiger data sets, except with $\gamma=10^{-3}$. For the Fox data set, $C=100$ and $\gamma=10^{-2}$ were used.

Classification performance was compared using two metrics: the prediction accuracy, used in most papers in the literature, and the area under the ROC curve (AUC). Some authors advocate the use of the AUC over accuracy as a comparison metric for classifiers [42–44]. When available, both are reported. To measure accuracy, a threshold β has to be optimized to maximize bag prediction accuracy once the pool of classifiers is created. Ideally, when enough data is available this is done on a held-out validation set. However, since the number of bags is limited in the benchmark data sets and our experiments showed held-out validation degrades performance. Therefore, the value of the decision threshold β was optimized on the training data. AUC is a global measure over all β values.

[5] Ten repetitions of a 10-fold CV is the protocol used in the vast majority of MIL publications.

Please cite this article as: M.-A. Carbonneau, et al., Robust multiple-instance learning ensembles using random subspace instance selection, Pattern Recognition (2016), http://dx.doi.org/10.1016/j.patcog.2016.03.035

Table 2
Default parameters of synthetic data sets.

+ Bags	− Bags	Features	IFP	Concepts	Instances per bag		
					Witness rate	Min.	Max.
100	100	25	0.1	3	0.5	1	50

4.3. Reference methods

Five well-known reference methods were implemented and tested for experiments with the synthetic data (see Section 5). These methods were selected because they yield good performances and represent a spectrum of different approaches that may perform differently depending on data set characteristics.

APR: This method was selected based on its popularity and its good performance on the Musk data sets. Zhou's MATLAB implementation [37] was used in the experiments. However, a modification was applied to obtain a classification score and compute the AUC. For each instance, the proportion of relevant dimensions in which the instance falls inside the hyper-rectangle is used as score. The score of a bag is given by the maximum instance score it contains. Preliminary experiments were conducted on data sets generated using the parameters listed in Table 2 with non-uniform negative distribution. The overall best results were obtained using $\tau=0.99$ and $\epsilon=0.01$. These settings were used for all subsequent experiments on the synthetic data set. The recommended settings were used in the experiments on benchmark data sets.

Citation-kNN : This method was selected due to its popularity and good performance. Zhou's MATLAB implementation [37] was used, but the distance function was compiled to native code to decrease computation time. Also, to obtain a ROC curve, a score output, corresponding to the proportion of positive *citers* and *references*, was added to the function. Preliminary experiments were conducted on data sets generated using the parameters listed in Table 2 with non-uniform negative distribution. The overall best results were obtained using 5 citers and 5 references. These settings were used for all subsequent experiments on the synthetic data set. The recommended settings were used in the experiments on benchmark data sets.

mi-SVM : This method was selected because it is instance-based, uses SVM and is well-known. The LIBSVM [41] library was used for the SVM implementation. The decision values were used for AUC computation. The score of a bag is the highest decision value in the bag. An exponential kernel was used with parameters $\gamma=0.1$ and $C=10$. These settings were optimized via grid search in a preliminary experiment on data sets generated using the parameters listed in Table 2 with non-uniform negative distribution.

AL-SVM : This method was selected for comparison because it was showed to perform well on low witness rate problems. It is very similar to the mi-SVM algorithm because it minimizes the same objective function under the same constraints [22]. It is different in the way the algorithm is initialized and how labels are attributed by a deterministic annealing procedure, which is hoped to find a better solution. The authors provide an implementation of the algorithm which was used in the experiments. As suggested in the paper, the Gaussian kernel was used, and its width was set to the median pairwise distance between instances. The initial temperature was set to $10C$ and $C=10$, as for mi-SVM.

CCE: The constructive clustering ensemble method (CCE) [32] was selected for comparison with the proposed method because both methods perform a clustering of the feature space and use an ensemble of SVM. At first, the feature space is clustered using a fixed number of clusters. Every bag is then represented by a binary vector, with each bit corresponding to a cluster. When at least one instance from a bag is attributed to a cluster, its corresponding bit is set to 1. The binary codes of the bags are used as feature vectors to train a classifier. Diversity is created in the ensemble by using a different number of clusters each time. The authors implementation is used in the experiment. This implementation uses k-means clustering and SVM classifiers. As recommended in the paper, the ensemble contains 5 classifiers and uses 10, 20, 30, 40 and 50 clusters.

Reference methods for benchmark data sets: For experiments on benchmarking data (see Section 6), many reference MIL techniques are compared. In order to assess the benefits of the random subspace instance selection procedure, tests were also conducted using SVM ensembles in which the training subsets were composed of randomly selected instances. The algorithm is the same as the one proposed in Section 3, except that samples were drawn from bags with uniform probabilities. The results for MILES on the Newsgroups data sets were obtained using the MIL toolbox implementation [45]. The optimal hyper-parameters for MILES and mi-SVM were obtained via grid search using an inner loop cross-validation as described in Section 4.2.

5. Results on synthetic data

Experiments in this section show the robustness of the proposed RSIS method to various data set characteristics.

5.1. Number of concepts

Fig. 5 presents the performance of the proposed and reference methods with the synthetic data set when the number of concepts increases in the data set. As explained in Section 4.1.2, here, a concept refers to a data cluster or a distribution mode that may or may not be defined over the complete feature space.

The figure shows that the performance of APR is affected by the number of concepts in the data set. When there are many concepts, the algorithm either leaves some concepts outside of the hyper-rectangle, or encompasses all of them at the price of a greater false alarm rate. Moreover, this algorithm's performance depends on the geometry of distributions. While APR performs well with uniform negative distribution, it is not the case when the data is clustered. This can also be observed in Figs. 6 and 7. When positive and negative distributions are multi-modal, the algorithm pursues two, sometimes, conflicting objectives. It must maximize the number of positive clusters contained in the hyper-rectangle, while minimizing the inclusion of negative ones. The spatial arrangement of these clusters varies with each generation of the data set, resulting in an higher deviation than with other algorithms in all experiments.

The mi-SVM algorithm is not vulnerable to multi-modal distributions. The use of a kernel enables the SVM to create disjoint data partitions without problems. mi-SVM performs better on multi-modal negative distributions, as opposed to APR and Citation-kNN because the structure of the negative data is informative in the instance label assignation process. This structure is however nonexistent when the negative distribution is uniform, which makes it more difficult to identify negative instances from other known negative instances. By comparing accuracy and AUC results in Figs. 5–7, one can see that the accuracy of mi-SVM can improve in many situations by optimizing the offset of the decision hyper-plane on bags instead of instances. For instance, with the uniform negative distribution, the accuracy is often about 50%, while the AUC results are competitive.

The AL-SVM algorithm is closely related to mi-SVM, as explained earlier. The AL-SVM has inherited some robustness to multi-modal distributions, however the deterministic annealing

Please cite this article as: M.-A. Carbonneau, et al., Robust multiple-instance learning ensembles using random subspace instance selection, Pattern Recognition (2016), http://dx.doi.org/10.1016/j.patcog.2016.03.035

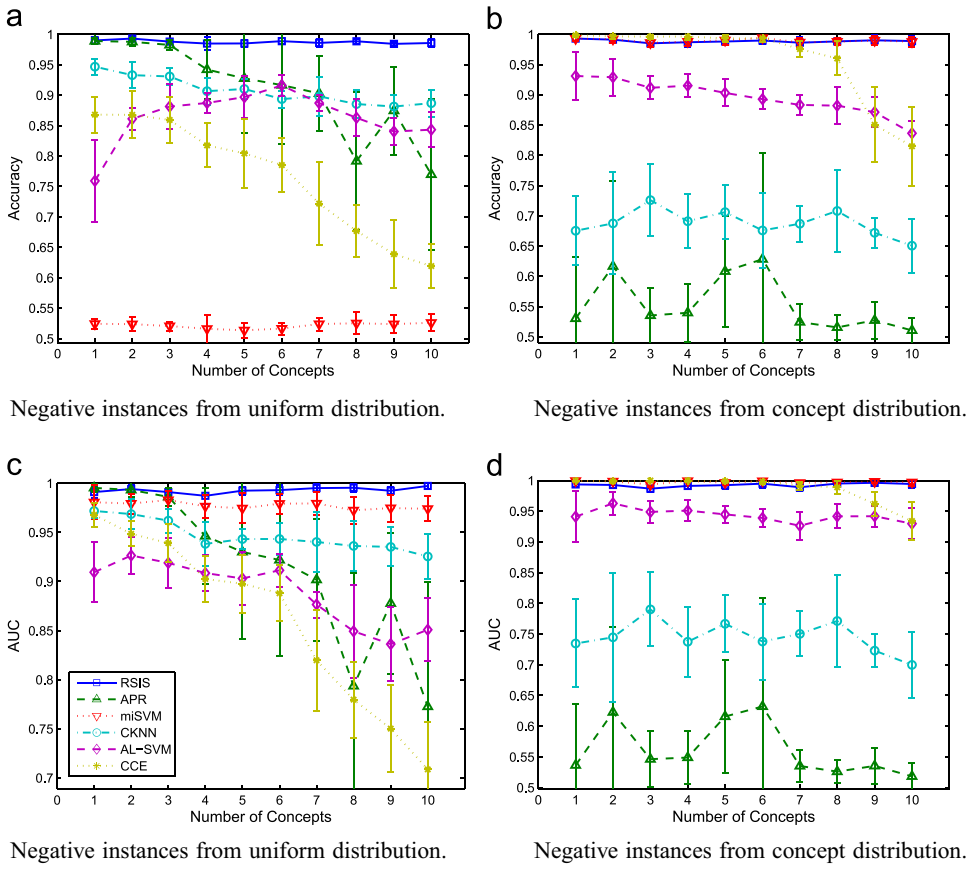


Fig. 5. Average performance of EoSVM with RSIS and the reference methods for a growing number of concepts in the data set. The error bars correspond to the standard deviation. Results obtained in data sets where the negative distribution is uniform are in (a) and (c), while in (b) and (d), the negative distributions were composed of Gaussian clusters.

procedure seems to make the algorithm overlook some concepts in the data when their number increases. This could be because the two algorithms are not initialized in the same way. In mi-SVM, all instances in positive bags are initialized as positive, while it is not the case for AL-SVM. If a majority of positive instances from the same concept are wrongly initialized as negative instances, the concept is never learned as positive. However, the deterministic annealing procedure has proved beneficial to find an the SVM hyper-plane offset in the case of the uniform negative instance distribution, where mi-SVM failed completely (see Fig. 5(a)). The performances of AL-SVM seem to always be inferior when considering the AUC. Inferior performances have also been observed by the authors when comparing the two algorithms on real-life benchmark data sets [22].

In Citation-kNN, instances are assigned the same label as their bags, thus only negative instances may be mislabeled. When the negative distribution is uniform, the mislabeled negative instances are sparsely distributed across the feature space. Therefore, it is more unlikely that a majority of instances in a neighborhood will be mislabeled. However, when the negative instances are grouped in clusters, this particular situation becomes more probable. This explains the difference in the algorithm performance on the two versions of the data set. Citation-kNN appears to be somewhat resistant to the number of clusters in the non-uniform data set. However, a decrease in performance is observed in the uniform distribution case, but this may be due to the limited number of bags in the data set.

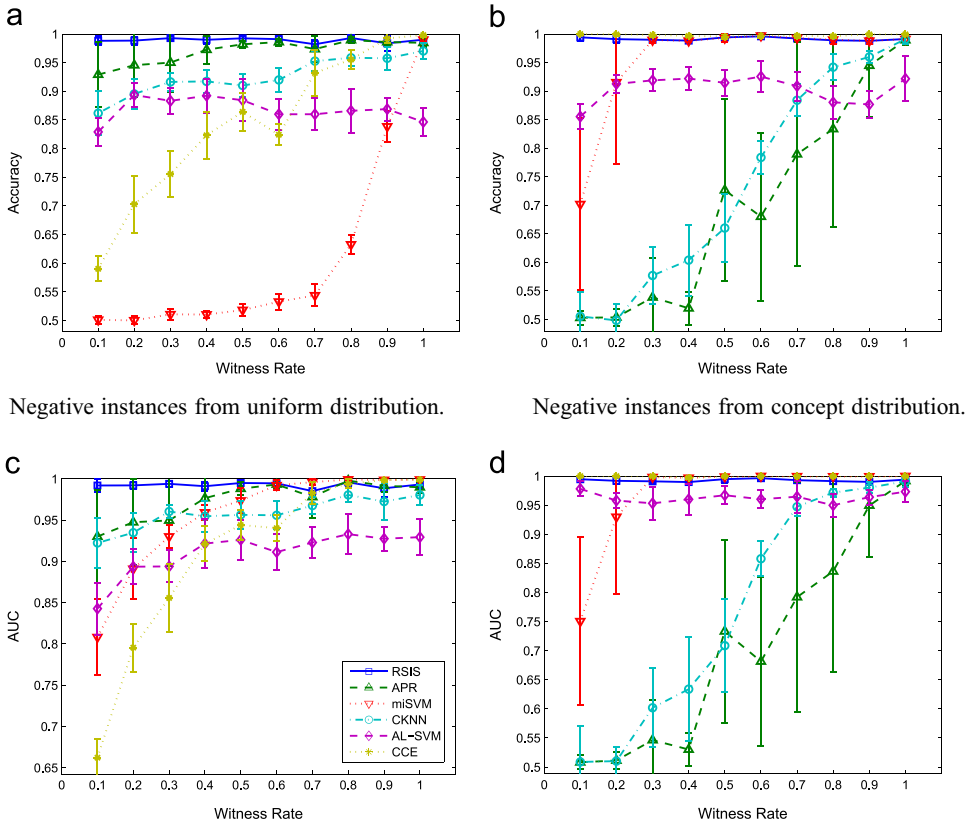
The ensembles created with the CCE procedure are affected by the number of concepts, but only in certain cases. If the positive and negative distributions are composed of clear clusters, the algorithm performs better than all others and obtains consistently near perfect results. A degradation is observed after 7 concepts, but an optimization of the number of clusters used in the clustering phase and the number of classifiers in the ensemble would probably perform better in these cases. However, CCE does not perform as well in situations where the negative distribution is not organized in clusters. This makes sense since the clustering, which is used to create the bag representation, has no clusters to find, and thus fails to create meaningful feature vectors. Fig. 5(a) and (c) shows that the problem worsen as the number of positive concepts increases.

Ensembles with RSIS are resistant to the number of concepts and outperforms reference methods. This is because, in the ensemble, the SVMs are not trained using the same positive data. All positive instances receive similar positivity scores, and thus have similar probabilities of being selected as training instances. The fact that 5 clusters were used in the clustering process does not limit performance event if there are more cluster in the data set. Also, the shape of the negative distribution does not decrease performance as with the other methods. Comparable results were obtained using ensembles with randomly selected instances on this section. Since these ensembles already obtained near perfect results on this synthetic data, there was no room for significant improvement using RSIS. The efficiency of RSIS over random instance selection will be demonstrated on more difficult data sets in Section 6.

5.2. Witness rate

Fig. 6 presents results of obtained on the synthetic data when the witness rate is gradually increased. Some methods rely on the assumption that there is a majority of positive instances in positive

Please cite this article as: M.-A. Carbonneau, et al., Robust multiple-instance learning ensembles using random subspace instance selection, Pattern Recognition (2016), http://dx.doi.org/10.1016/j.patcog.2016.03.035



Negative instances from uniform distribution. Negative instances from concept distribution.

Negative instances from uniform distribution. Negative instances from concept distribution.

Fig. 6. Average performance of ensembles with RSIS and the reference methods when varying the witness rate in the data set. The error bars correspond to the standard deviation. Results obtained in data sets where the negative distribution is uniform are in (a) and (c), while in (b) and (d), the negative distributions were composed of 3 clusters.

bags. These methods thus perform well on certain types of data sets, such as the Musk data sets. This is the case for mi-SVM, because in the initialization, all instances in positive bags are assumed to be positive. However, as the proportion of positives declines, the more challenges arise to correctly identify the proper instance labels during the optimization process.

APR is also affected by the witness rate. The accuracy and AUC both increase as the witness rate rises. As with mi-SVM, instances from positive bags are considered positive. When the witness rate is low, there are more mislabeled instances, which leads to performance degradation. In the case of non-uniform distributions, the learning process does not converge with a very low witness rate. Deterministic annealing in AL-SVM provides a solution to these problems and thus, at lower witness rates, the AL-SVM performs better than mi-SVM.

Citation-kNN is also sensitive to the witness rate because, as stated earlier, instance labels correspond to bag labels. Hence, when the witness rate is low, there is a greater chance that a negative instance from a positive bag will cause a classification error. As for APR, performance rises almost linearly with the witness rate on the non-uniform negative distribution.

As observed in the previous experiment, CCE has difficulties dealing with uniform distributions, however, when both distribution are composed of clear clusters, the algorithm works perfectly regardless of the witness rate.

Ensembles with RSIS performs consistently well under a wide range of witness rates. It is only outperformed by CCE with negative concept distributions and by mi-SVM when the witness rate is very high. This is because RSIS selects the most probably positive instances for training, and thus, when all instances of positive bags are positive, the most difficult instances do not get picked as training instances. On the other hand, mi-SVM includes them in its model, and thus can achieve better performance in these particular cases. Also mi-SVM has lower computational complexity because only one classifier is used instead of an ensemble.

5.3. *Proportion of irrelevant features*

In Fig. 7, the proportion of irrelevant features, was gradually increased to assess robustness to noise. An irrelevant feature is a feature which does not contain any information for a given concept. In other words, it is a feature in which instances, generated by given concept, are uniformly distributed. Irrelevant features are not the same for each concept, as illustrated in Fig. 4, so feature extraction and selection techniques would not alleviate this challenge.

The performance of all of the tested methods decreased as the number of irrelevant features increased. In the non-uniform case, the accuracy of mi-SVM is rapidly affected by the inclusion of irrelevant features. However, the AUC results are as stable as the best performing algorithm, ensembles with RSIS. AL-SVM is affected in the same way as mi-SVM when considering AUC, but, as observed in previous experiment, the algorithm is better at determining the SVM hyper-plan offset. This can be observed through the higher accuracy of AL-SVM vs. mi-SVM in Fig. 7(a). It also explains why the accuracy of AL-SVM degrades progressively in Fig. 7(b), as opposed to the accuracy of mi-SVM.

Performance of Citation-kNN declines when a majority of features are uniformly generated. This algorithm depends on the Hausdroff distance, and when many irrelevant dimensions are

Please cite this article as: M.-A. Carbonneau, et al., Robust multiple-instance learning ensembles using random subspace instance selection, Pattern Recognition (2016), http://dx.doi.org/10.1016/j.patcog.2016.03.035

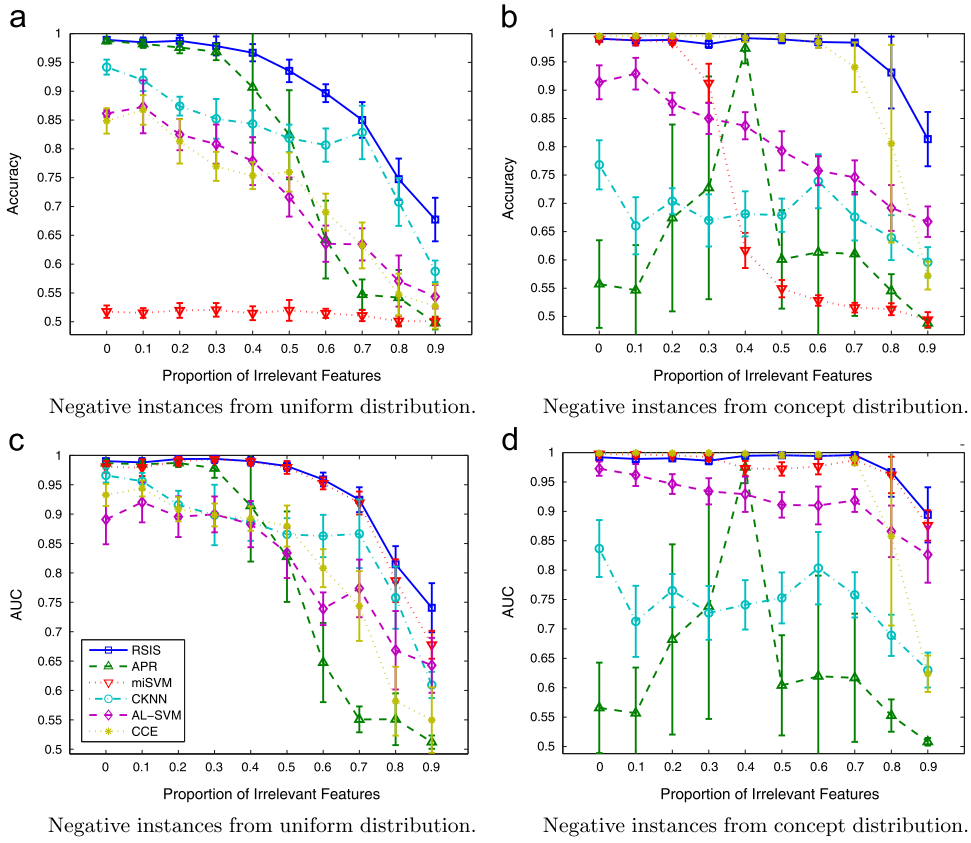


Fig. 7. Average performance of ensembles with RSIS and the reference methods when varying the proportion of irrelevant features used to describe each concept. The error bars correspond to the standard deviation. Results obtained in data sets where the negative distribution is uniform are in (a) and (c), while in (b) and (d), the negative distributions were composed of 3 clusters.

considered in the distance calculation, the measure loses discrimination.

APR performance is affected by irrelevant features. In particular cases, the inclusion of irrelevant dimensions is beneficial (see Fig. 7(b) and (d)). When there are fewer relevant features, the probability that positive concepts share these features decreases. It is therefore easier to define an hyper-rectangle that closely fits the positive instance distribution even if it is separated in distinct concept. This beneficial effect is however offset by ambiguity introduced by a high number of irrelevant features.

In this experiment, an irrelevant feature means a uniform distribution. This was showed to be the weakness of CCE, and this is why, even with the clustered negative distribution, performance drops drastically after the proportion of irrelevant features reaches 0.6.

As with the other algorithms, ensembles with RSIS are affected by irrelevant features. In this case, clustering is performed using the Euclidean distance. When a large proportion of feature in the data set is meaningless, the distance measure also becomes meaningless. However, by isolating features in subsets, the creation of random subspaces provides a certain resilience against this corrupting effect.

6. Results on benchmark data sets

Experiments in the last section demonstrated RSIS robustness to different data set parameters. In this section, RSIS is compared to other methods on widely-used standard benchmark data sets. Results obtained with an ensemble of SVM (EoSVM) without the RSIS procedure are also presented to further assess the benefits of using RSIS.

6.1. Musk data sets

Results for RSIS on Musk data sets are reported alongside results from other alternatives in Table 3. Most papers do not provide the AUC, however results for a number of methods have been published in [23,27,46,47] and are reported here. Standard deviation is provided when available.

The results for Citation-kNN are obtained from Zhou's implementation [37], using the parameter settings suggested in the original paper ($C=4$ and $R=2$). New experiments had to be performed, because the original paper used leave-one-out cross-validation. This is also the case for CCE [32]. The implementation provided by the authors was used, as well as the suggested parameters in the paper. The AUC for APR was also computed with Zhou's implementation, using the original paper's optimal parameters ($\tau=0.999$ and $\epsilon=0.01$). The results for EM-DD come from [4], because the original paper optimized its results on the test data.

In light of the results reported in Table 3, one can see that RSIS delivers a similar or better accuracy on Musk1 than most other methods. Only APR and PC-SVM possess a statistically-significant advantage over RSIS. On Musk2, RSIS also delivers state-of-the-art results. Without standard deviations, it is difficult to assess the significance of the advantage of some methods. Nonetheless, MILIS outperforms RSIS with reasonable certainty (95%) on this data set.

When comparing based on the AUC, RSIS results are significantly superior to methods reported on Musk1. On Musk2,

RSIS, MlnD and MILES provide similar results, while outperforming the other techniques.

Finally, the results obtained with the SVM ensemble without the instance selection procedure are compared against ensembles designed with RSIS. The selection procedure significantly improves the accuracy performance of the ensembles. However, when comparing AUC, the results only differ on Musk1, suggesting that, without the selection procedure, the optimal classification threshold (β) is harder to determine. This is because, without instance selection, many classifiers in the ensemble are unreliable. While an optimal threshold works well with a certain data subset, varying performances will be obtained on different data.

Table 3
Experimental results on the Musk data sets.

Algorithm	Accuracy (%)		AUC [23,27,46,47]	
	Musk 1	Musk 2	Musk 1	Musk 2
MILES [6]	86.3 (1.4)	87.7 (1.4)	93.2 (2.9)	97.1 (1.6)
MILIS [7]	88.6 (2.9)	91.1 (1.7)	–	–
APR [2]	92.4 (2.7)	89.2 (3.0)	91.8 (1.0)	88.4 (2.6)
Citation-kNN [17]	90.3 (1.3)	83.7 (2.3)	93.5 (2.0)	88.0 (1.9)
DD [18]	88.9	82.5	89.5	90.3
DD-SVM [5]	85.8	91.3	–	–
EM-DD [19,4]	84.8	84.9	87.4 (2.1)	86.9 (2.1)
mi-SVM [4]	87.4	83.6	93.9 (1.6)	81.5 (2.1)
MI-SVM [4]	77.9	84.3	91.5 (3.7)	93.9 (2.8)
MI-NN [48]	88.0	82.0	–	–
Multinst [49]	76.7 (3.1)	84.0 (2.6)	–	–
RELIC [50]	83.7	87.3	–	–
MICA [36]	84.4	90.5	–	–
AW-SVM [22]	85.7	83.8	–	–
ALP-SVM [22]	86.3	86.2	–	–
SVR-SVM [24]	87.9 (1.7)	85.4 (1.8)	–	–
γ-rule [25]	88.4 (1.1)	84.9 (2.2)	–	–
MILBoost [11]	69.8 (5.4)	76.4 (3.5)	74.8 (6.7)	76.4 (3.5)
MlnD [27]	–	–	93.4 (1.2)	95.4 (1.4)
TLC [51]	88.7 (1.6)	83.1 (3.2)	–	–
MIBoosting [21]	87.9 (2.0)	84.0 (1.3)	–	–
PC-SVM [52]	90.6 (2.7)	91.3 (3.2)	–	–
MI-Graph [28]	90.0 (3.8)	90.0 (2.7)	–	–
mi-Graph [28]	88.9 (3.3)	90.3 (2.6)	–	–
MI-Kernel (NSK) [20]	88.0 (3.1)	89.3 (1.5)	85.6	90.8
sbMIL [23]	–	–	91.8	87.7
stMIL [23]	–	–	79.5	68.4
CCE [32]	81.3 (2.0)	71.7 (3.4)	88.6 (1.4)	79.4 (3.4)
Dissimilarity ensembles [47]	89.3 (3.4)	85.5 (4.7)	95.4 (2.4)	93.2 (3.2)
EoSVM (random selection)	82.8 (1.9)	83.6 (2.0)	94.4 (1.3)	94.4 (1.1)
EoSVM (RSIS)	**88.8 (1.3)**	**89.5 (1.6)**	**96.5 (0.9)**	**95.2 (1.0)**

6.2. Elephant, Fox and Tiger data sets

As for the Musk data sets, the accuracy of the original papers is reported along with the AUC, when available, in Table 4. The results for APR, Citation-kNN and CCE were obtained with Zhou's implementations [37,32]. RSIS performs better or as well as most methods reported on the Elephant data set. Only PC-SVM and mi-Graph have a statistically-significant advantage over RSIS. On the Tiger and Fox data sets, the results obtained with RSIS are surpassed by 4 and 5 methods, respectively. When comparing based on AUC, RSIS's results are superior or equivalent to all other reported methods.

As was the case with the musk databases, there is a clear advantage of using RSIS over SVM without selection when comparing accuracy. In light of the AUC, however, a significant advantage is observed only on the Tiger data set. As for the results on the Musk data sets, these results suggest that RSIS produces a more reliable ensemble, which eases the selection of the final classification threshold.

6.3. Newsgroups

The results reported in Table 5 are taken from [24,28]. Tests were also conducted on the data sets using CCE, mi-SVM and MILES. As mentioned earlier, methods pooling all instances together, like MI-Kernel [20], do not perform well when the witness rate is low. This is also the case for embedding methods, like MILES [6]. By considering the bags as a whole, these methods fail when a majority of instances do not belong to the target class. Results

Table 4
Experimental results on the Tiger, Fox and Elephant data sets.

Algorithm	Accuracy (%)			AUC [23,27,46,47]		
	Elephant	Tiger	Fox	Elephant	Tiger	Fox
MILES [6]	79.0 (2.3)	81.0 (3.4)	62.5 (4.2)	88.3 (1.1)	87.2 (1.7)	69.8 (1.7)
APR [2]	75.1 (1.3)	55.8 (1.1)	53.2 (1.2)	77.8 (0.7)	55.0 (1.0)	54.1 (0.9)
Citation-kNN [17]	82.6 (0.9)	78.8 (1.3)	58.2 (1.1)	89.6 (0.9)	85.5 (0.9)	63.5 (1.5)
DD [18]	–	–	–	90.7	84.1	63.1
EM-DD [19]	78.3	72.1	56.1	88.5	72.3	67.6
mi-SVM [4]	82.2	78.4	58.2	84.3 (13.2)	83.3 (2.1)	56.1 (7.5)
MI-SVM [4]	81.4	84.0	57.8	90.7 (2.1)	87.2 (3.5)	68.7 (2.6)
MICA [36]	80.5 (8.5)	82.6 (7.9)	58.7 (11.3)	–	–	–
AW-SVM [22]	82.0	83.0	63.5	–	–	–
ALP-SVM [22]	83.5	86.0	66.0	–	–	–
SVR-SVM [24]	85.3 (2.8)	79.8 (3.4)	63.0 (3.5)	–	–	–
γ-rule [25]	84.4 (0.9)	80.8 (1.2)	62.8 (0.9)	–	–	–
MILBoost [11]	79.5 (2.8)	78.5 (2.8)	63.0 (2.6)	89.0 (5.2)	84.1 (5.1)	61.1 (7.6)
MlnD [27]	–	–	–	93.1 (0.8)	85.1 (1.7)	60.5 (1.9)
PC-SVM [52]	89.8 (1.2)	83.8 (1.3)	65.7 (1.4)	–	–	–
MI-Graph [28]	85.1 (2.8)	81.9 (1.5)	61.2 (1.7)	–	–	–
mi-Graph [28]	86.8 (0.7)	86.0 (1.0)	61.6 (2.8)	–	–	–
MI-Kernel (NSK) [20]	84.3 (1.6)	84.2 (1.0)	60.3 (1.9)	82.9	79.1	64.0
sbMIL [23]	–	–	–	88.6	83.0	69.8
stMIL [23]	–	–	–	81.6	74.5	60.7
CCE [32]	79.6 (2.3)	75.6 (1.7)	61.5 (2.4)	87.8 (1.1)	81.6 (1.8)	64.9 (2.6)
Dissimilarity Ensembles [47]	84.5 (2.8)	81.0 (4.6)	64.5 (2.2)	92.3 (2.7)	87.8 (4.2)	70.2 (1.8)
EoSVM (random selection)	82.5 (1.2)	73.7 (1.5)	57.9 (2.0)	92.4 (0.7)	84.5 (1.3)	67.3 (1.4)
EoSVM (RSIS)	**84.6 (0.8)**	**82.5 (1.3)**	**61.1 (1.8)**	**90.8 (0.8)**	**88.8 (0.9)**	**68.2 (1.8)**

Please cite this article as: M.-A. Carbonneau, et al., Robust multiple-instance learning ensembles using random subspace instance selection, Pattern Recognition (2016), http://dx.doi.org/10.1016/j.patcog.2016.03.035

Table 5
Experimental results on the Newsgroups data sets.

Data set	Algorithm accuracy (%)							
	MILES [6]	MI-Kernel [20]	mi-SVM [4]	mi-Graph [28]	CCE [32]	SVR-SVM [24]	EoSVM (RSIS)	WR (%)
alt.atheism	55.9 (2.6)	60.2 (3.9)	79.2 (4.0)	65.5 (4.0)	77.8 (2.3)	83.5 (1.7)	**86.0** (**1.8**)	1.83
comp.graphics	52.1 (2.9)	47.0 (3.3)	74.0 (3.2)	77.8 (1.6)	66.6 (1.8)	**85.2** (**1.5**)	80.4 (1.4)	5.19
comp.windows.misc	50.5 (3.8)	51.0 (5.2)	62.3 (2.1)	63.1 (1.5)	59.9 (3.5)	66.9 (2.6)	**70.3** (**2.7**)	2.23
comp.pc.hardware	49.9 (2.4)	46.9 (3.6)	59.3 (3.5)	59.5 (2.7)	66.2 (5.6)	70.3 (2.8)	**74.9** (**2.2**)	2.42
comp.mac.hardware	52.2 (2.2)	44.5 (3.2)	75.4 (2.4)	61.7 (4.8)	61.4 (3.0)	**78.0** (**1.7**)	**79.4** (**2.4**)	4.58
comp.window.x	56.1 (2.0)	50.8 (4.3)	58.7 (4.0)	69.8 (2.1)	72.8 (3.5)	**83.7** (**2.0**)	81.8 (1.6)	5.36
misc.forsale	53.3 (3.5)	51.8 (2.5)	68.9 (2.8)	55.2 (2.7)	63.2 (3.0)	**72.3** (**1.2**)	**73.0** (**2.3**)	4.29
rec.autos	50.5 (2.5)	52.9 (3.3)	61.0 (3.2)	72.0 (3.7)	65.9 (2.6)	**78.1** (**1.9**)	75.0 (2.3)	2.75
rec.motorcycles	60.0 (3.2)	50.6 (3.5)	53.9 (1.7)	64.0 (2.8)	78.6 (2.0)	75.6 (0.9)	**80.0** (**1.8**)	2.86
rec.sport.baseball	52.8 (2.8)	51.7 (2.8)	53.8 (2.5)	64.7 (3.1)	74.2 (1.2)	76.7 (1.4)	**87.1** (**2.2**)	4.31
rec.sport.hockey	51.8 (1.6)	51.3 (3.4)	59.8 (3.8)	85.0 (2.5)	75.8 (2.1)	**89.3** (**1.6**)	**90.5** (**1.5**)	6.52
sci.crypt	56.4 (2.5)	56.3 (3.6)	67.3 (2.2)	69.6 (2.1)	72.9 (1.8)	69.7 (2.5)	**76.7** (**1.6**)	3.22
sci.electronics	50.3 (1.6)	50.6 (2.0)	82.8 (3.2)	87.1 (1.7)	62.4 (2.3)	91.5 (1.0)	**93.7** (**0.5**)	4.29
sci.med	54.4 (3.2)	50.6 (1.9)	69.9 (3.5)	62.1 (3.9)	72.2 (1.9)	74.9 (1.9)	**82.8** (**2.5**)	5.23
sci.space	54.0 (4.0)	54.7 (2.5)	52.3 (1.7)	75.7 (3.4)	75.0 (2.3)	**83.2** (**2.0**)	**81.0** (**2.7**)	3.64
soc.religion.christian	56.7 (3.0)	49.2 (3.4)	50.0 (0.0)	59.0 (4.7)	76.6 (2.1)	**83.2** (**2.7**)	80.6 (2.0)	3.30
talk.politics.guns	53.0 (4.3)	47.7 (3.8)	67.1 (2.8)	58.5 (6.0)	**73.4** (**2.9**)	**73.7** (**2.6**)	**74.5** (**2.5**)	3.23
talk.politics.mideast	55.5 (4.5)	55.9 (2.8)	78.1 (1.9)	73.6 (2.6)	79.2 (2.4)	80.5 (3.2)	**85.0** (**1.1**)	3.88
talk.politics.misc	59.2 (2.5)	51.5 (3.7)	67.6 (2.6)	70.4 (3.6)	**74.0** (**2.2**)	72.6 (1.4)	**74.3** (**1.9**)	2.82
talk.religion.misc	53.2 (1.9)	55.4 (4.3)	41.0 (1.6)	63.3 (3.5)	70.9 (3.1)	71.9 (1.9)	**75.5** (**1.6**)	2.87

obtained in this section support this conclusion. The accuracy obtained with MILES and MI-Kernel does not exceed 60%, and often revolves around 50% for all data sets, which is the proportion of negative bags in the data set. Results obtained with mi-SVM are better. This method considers instances individually which seems to pay off in these low witness rate problems. The mi-Graph method derives an instance affinity matrix for each bag. This matrix is used to re-weigh the influence of instances belonging to the same concept. Thus instances belonging to an under-represented concept in the bags gain more influence during classification. Using this scheme, the results obtained are slightly better than the results obtained with mi-SVM. CCE represents bags as a whole, but the representation is not directly based on the instance feature vectors. The feature vectors representing the bags encode only the presence, and not the quantity, of instances in different clusters. This provides a robustness to low witness rate because the representation remains the same, independently of the number of similar negative instances in the bag. This is why despite using a bag-level representation CCE obtains competitive results. SVR-SVM is a method designed specially to withstand various witness rates. Therefore, the method yields far better results than MILES, MI-Kernel, mi-SVM and mi-Graph. The proposed method (RSIS) gets the best results on 16 of the 20 data sets. On 11 of the 20 data sets, it has a statistically significant advantage over all other methods. These results further illustrate the robustness to low witness rate of the proposed method.

In several additional papers, only the alt.atheism in the Newsgroups data set is used as a benchmark. Results are reported in Table 6. Most of the methods reported in this table yielded state-of-the-art results on at least one of the other benchmark data set, however, in this case, because the witness rate is below 2%, the performances of these methods decrease significantly. It shows that special care needs to be taken when designing a MIL algorithm used in low witness rate contexts. This is why SVR-SVM and RSIS yield the best performances.

Table 6
Experimental results on the alt.atheism data set.

Algorithm	Accuracy (%) [47,28,24]
APR [2]	49.0 (0.0)
Citation-kNN [17]	50.0 (0.0)
Dissimilarity Ensembles [47]	44.0 (4.5)
MI-SVM [4]	48.0 (2.0)
EM-DD [19]	49.0 (5.7)
MILES [6]	55.9 (2.6)
MI-Kernel [20]	60.2 (3.9)
mi-Graph [28]	65.5 (4.0)
Minimax-Kernel [20]	76.0 (4.0)
CCE [32]	77.8 (2.3)
mi-SVM [4]	79.2 (4.0)
SVR-SVM [24]	83.5 (1.7)
EoSVM (RSIS)	**86.0 (1.8)**

Table 7
Initial values in parameter sensitivity experiments.

Parameter	Symbol	Value
Number of clusters	K	5
Temperature	T	0.01
Number of classifiers	M	100
Number of subspaces	R	500
Proportion of features used to create subspaces	$\lvert\mathcal{P}\rvert/\lvert\mathcal{F}\rvert$	5%

7. Results on parameter sensitivity

In this section, experiments are conducted on the benchmark data sets to evaluate the parameter sensitivity of RSIS. The objective is to identify which parameters need careful tuning, and which parameters have a negligible effect on performance. The basic settings listed in Table 7 are varied one by one to observe their effect on performance. These settings were optimized on the Musk1 data set and then tested, as is, on the other databases to evaluate the specificity of the optimization procedure.

Fig. 8(a) shows that beyond 10, the number of classifiers M used in the ensemble does not significantly affect performance. For all data sets, the accuracy is contained in a maximum range of $\pm$ 1.0%. All of these values have a standard deviation between 0.7% and 2.5%. Moreover, except for some isolated cases (which do not represents a tendency), the accuracy falls in the standard deviation range of all other points on the curve. These small variations are mostly due to the randomness introduced by some parts of the algorithm and the cross-validation procedure.

Please cite this article as: M.-A. Carbonneau, et al., Robust multiple-instance learning ensembles using random subspace instance selection, Pattern Recognition (2016), http://dx.doi.org/10.1016/j.patcog.2016.03.035

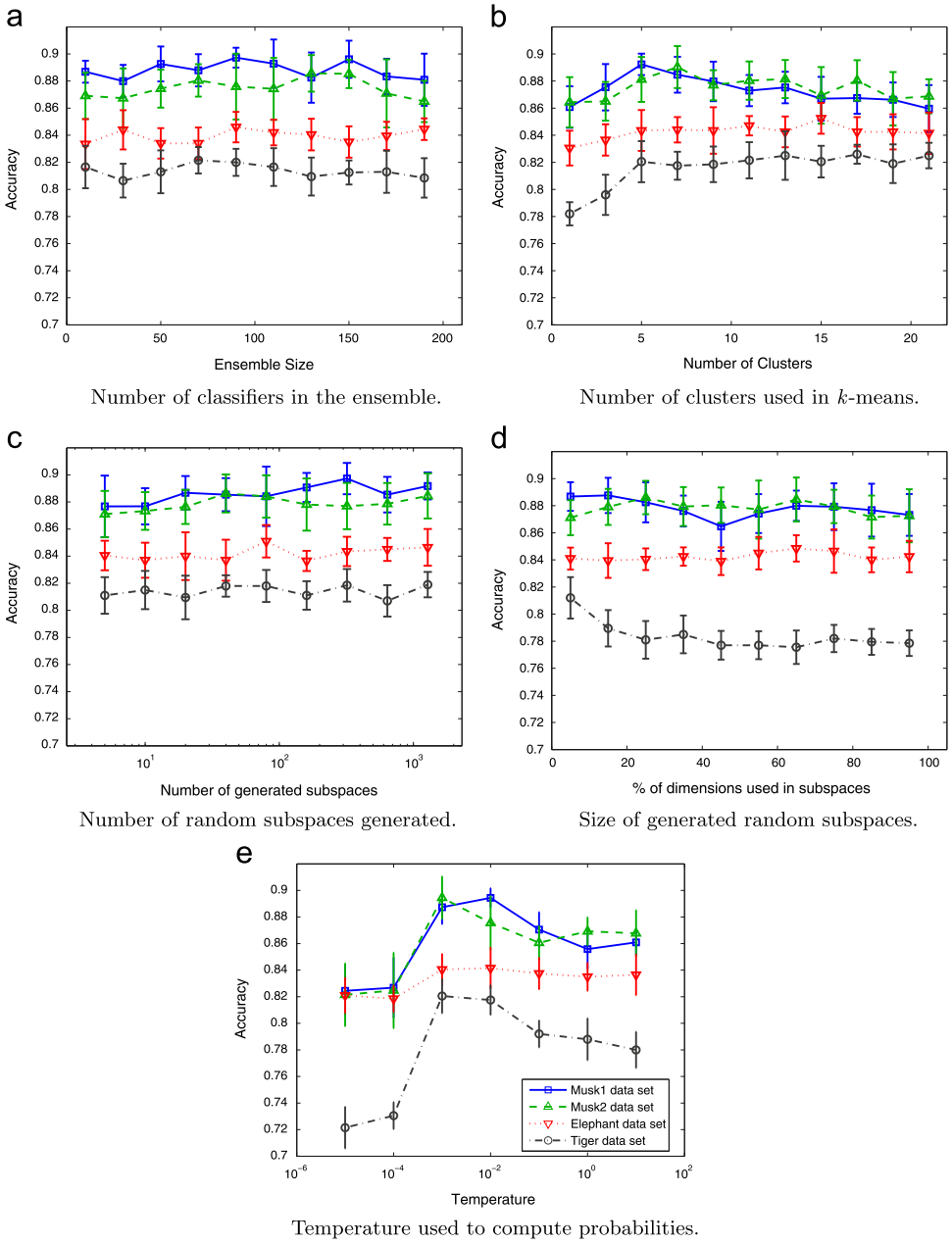


Fig. 8. This figure presents a parameter sensitivity analysis of the proposed method on 4 benchmark data sets. In each graph, a parameter is varied and the average accuracy is reported. The error bars represent the standard deviation.

Fig. 8(b) shows that the number of clusters K should be optimized based on the data. All curves indicate that a minimum of clusters should be used, but the optimal setting seems to vary depending on the data set contents. Indeed, the quality of a clustering process using k-mean depends on the number of expected clusters (k) given the real number of clusters [53].

It can be observed in Fig. 8(c) that the number (R) of subspaces generated does not offset performance significantly. Clearly a minimum number subspaces must be created otherwise performance degrades. However, this number is surprising low, as can be seen in Fig. 8(c). This suggests that the number of generated subspaces is not of paramount importance as long as a minimum number of 100 is met.

The number of dimensions per subspace $|\mathcal{P}|$ is defined in terms of proportion of the complete feature space. From Fig. 8(d), better results are obtained with less than 5% of the complete feature space, on the Tiger and Musk1 data set, while no noticeable difference can be seen on the other two data sets. Results indicates that smaller subspaces are generally preferred.

The temperature (T) is the most critical parameter, as seen in Fig. 8(e). When lower, the same instances are picked for each classifier of the ensemble, and the diversity is lowered, which degrades performance. On the other hand, if the temperature is higher, the selection process becomes more random, and incorrect instances are selected more often, which also degrades performance. This parameter should ideally be optimized for every problem.

Finally, it can be seen from Fig. 8 that the results obtained on the other data sets using the Musk 1 configuration are comparable to those obtained in Sections 6.1 and 6.2 with full parameter optimization. This supports the claim that the algorithm is insensitive to most parameter settings. As for T and K, the optimal settings for the Musk1 data set are a reasonable choice for the other data sets too. However, as shown in Fig. 8(b) and (e), marginally better accuracy

Please cite this article as: M.-A. Carbonneau, et al., Robust multiple-instance learning ensembles using random subspace instance selection, Pattern Recognition (2016), http://dx.doi.org/10.1016/j.patcog.2016.03.035

may be achieved if these parameters are optimized. The recommendations of this section were successfully applied to the experiments on the Newsgroups data set (see Section 6.3).

8. Time complexity

All experiments were conducted on a Intel i7/2.4 GHz processor with 8 GB of RAM. All algorithms have been implemented in MATLAB. However, the compiled implementation of LIBSVM was used for every SVM used in the experiments. Also, in CKNN, the computation of the Euclidean distance was compiled to native code.

The execution time were obtained on the Musk1 and Tiger data sets. Results reported in Table 8 are the average and the standard deviation of 10 repetitions of a 10-fold cross-validation. The training time does not include the time used for parameter selection since it is dependent on user-defined search grids. The number of parameters to be tuned is also reported for each method.

Ensembles with RSIS algorithms have more user defined parameters than all of the other methods because there are parameters to be set for the base learners (kernel type, γ and C) and for the ensemble. The user must set 5 parameters for this particular MIL implementation of ensembles with RSIS. Among the 5 RSIS parameters (see Table 7), only 2 are directly related to the new ensemble learning approach, and require careful tuning, while the others can be set as recommended (see Section 7).

In the computation of positivity scores, the time complexity for clustering random subspaces, when using the k-means algorithm, is given by $\mathcal{O}(ndKR)$ where K is the number of clusters, R is the number of random subspaces, and n and d are the number of instance and the data dimensionality, respectively. The training complexity of SVM is difficult to assess since it depends on the implementation and kernel. Using LIBSVM, it is empirically known that the computational complexity is higher than linear to the n [41]. Here, it will be assumed to be $\mathcal{O}(n^2d)$. Since we train M classifiers, the complexity of the ensemble training phase is given by $\mathcal{O}(n^2dM)$. Along with the number of classifiers and the data dimensionality, the execution time depends on the regularization parameter C and the size of the data set [54]. Testing time depends on the number of classifiers in the ensemble (M), the data dimensionality and on the number of support vectors used in each SVM.

The training complexity of the mi-SVM is given by $\mathcal{O}(n^2dl)$, where l is the number of iterations needed by algorithm to converge. At each of these iterations, the SVM is retrained. The number of iterations required to obtain convergence is dependent on the nature of the data, and this is why the timing results exhibit high standard deviations. Compared to ensembles with RSIS, mi-SVM is faster to train with small data sets, but is slower as n increases. This is because the number of instances used to train each SVM of the ensemble is much smaller than the number used to train the one in mi-SVM. With RSIS, only one instance is selected in each bag to train the SVM, while every instances are used in mi-SVM. At some point, the complexity of mi-SVM ($\mathcal{O}(n^2dl)$) outgrows the complexity of ensemble with RSIS ($\mathcal{O}(B^2dM)$) where B is the number of bags. This can also be observed when comparing ensembles with RSIS and CKNN, which have a complexity of $\mathcal{O}(n^2dl)$. This suggests that ensembles with RSIS would scale better to big data sets. Independently of the data set size, during operation, ensembles with RSIS is the slowest of the four methods because every classifier in the ensemble needs to evaluate the instances in the bag. Finally, APR is the fastest method by far for training and testing regardless of the data set.

9. Conclusion

In this paper, a new instance selection mechanism using random subspaces is proposed to train MIL ensembles. The method can be used with any classifier and clustering algorithms. It is intended to be a versatile solution which can be applied to many types of MIL problems without extensive knowledge on the data structure. This is because its performance is not affected by low and high witness rates, the shape of data distributions is of little impact on its performance, and it increases noise robustness. Moreover, the method is able to identify positive instances in bags which is sometimes required in MIL applications.

The proposed method was compared to stat-of-the-art MIL methods on standard benchmark data sets, and yielded competitive results. A new synthetic data set was created to measure the adaptability of the proposed method to different data structures. The proposed method consistently yielded higher level of performance over the baseline methods for diverse conditions, namely witness rate, number of concepts and irrelevant feature rate. However, experiments suggest that other methods may perform better when the witness rate approaches 100%.

A drawback of the proposed method is the number of user-defined parameters to optimize. However, an analysis showed low sensitivity to most parameters. For instance, the number of generated subspaces is not critical, nor is the ensemble size. The number of dimensions used in subspaces should represent 5–10% of the complete feature space. This leaves only the temperature and the number of clusters in each subspace to be optimized. These recommendations were applied in the Newsgroups data set experiment and achieved state-of-the-art results. The recommendations were also applied to a synthetic data set, and consistently provided near-optimal results. As most ensemble methods, when compared with their single learner counterparts, the proposed method necessitates more processing time during operation. However, the proposed method has better training time scalability properties than mi-SVM and CKNN methods.

In future research, experiments should be conducted with different types of classifiers and clustering algorithms to measure the impact on performance. Also, in future versions of the

Table 8
Timing results on the Musk1 and the Tiger data sets.

Data set	Algorithm	Training time (ms)	Testing time (ms)	Number of parameters
Musk 1	APR	660 (65.0)	0.309 (0.672)	4
	CKNN	–	1290 (122)	2
	mi-SVM	62.9 (22.2)	3.82 (2.06)	3
	RSIS	963 (129)	935 (324)	2+3
Tiger	APR	137 (7.77)	0.541 (0.671)	4
	CKNN	–	22,100 (381)	2
	mi-SVM	21,200 (25,700)	9.45 (2.16)	3
	RSIS	2040 (106)	4480 (373)	2+3

Please cite this article as: M.-A. Carbonneau, et al., Robust multiple-instance learning ensembles using random subspace instance selection, Pattern Recognition (2016), http://dx.doi.org/10.1016/j.patcog.2016.03.035

algorithm, the number of instances selected in a positive bag could be adapted to the problem characteristics. If an estimation of the witness rate can be obtained, selecting more than one instance per bag could increase performance of base-learners, and thus, increase ensemble performance. Also, a diversity measure applicable to MIL problems could enable the use of an ensemble selection mechanism used to prune redundant classifiers. Finally, experiments should be conducted to assess the suitability of RSIS as a preliminary instance labeling stage to increase robustness of existing algorithms. As stated before, many methods, such as mi-SVM, MIBoosting and MI-Kernel, initialize their optimization process assuming that all instances in positive bags are positive. Initializing these methods with RSIS could prove beneficial.

Conflict of interest

None declared.

Acknowledgment

This work was supported by the Natural Sciences and Engineering Research Council of Canada (NSERC) and Quattriuum Inc.

References

[1] K. Ikeuchi, Computer Vision: A Reference Guide, Springer, New York, 2014, p. 898 Print ISBN: 978-0-387-30771-8.
[2] T.G. Dietterich, R.H. Lathrop, T. Lozano-Pérez, Solving the multiple instance problem with axis-parallel rectangles, Artif. Intell. 89 (1–2) (1997) 31–71.
[3] J.D. Keeler, D.E. Rumelhart, W.-K. Leow, Integrated segmentation and recognition of hand-printed numerals, in: Advances in Neural Information Processing Systems, MIT Press, San Francisco, USA, 1990, pp. 557–563.
[4] S. Andrews, I. Tsochantaridis, T. Hofmann, Support vector machines for multiple-instance learning, in: Advances in Neural Information Processing Systems, vol. 15, MIT Press, Vancouver, Canada, 2003, pp. 561–568.
[5] Y. Chen, J.Z. Wang, Image categorization by learning and reasoning with regions, J. Mach. Learn. Res. 5 (2004) 913–939.
[6] Y. Chen, J. Bi, J.Z. Wang, MILES: multiple-instance learning via embedded instance selection, IEEE Trans. Pattern Anal. Mach. Intell. 28 (12) (2006) 1931–1947.
[7] Z. Fu, A. Robles-Kelly, J. Zhou, MILIS: multiple instance learning with instance selection, IEEE Trans. Pattern Anal. Mach. Intell. 33 (5) (2011) 958–977.
[8] R. Rahmani, S. A. Goldman, MISSL: Multiple-instance Semi-supervised Learning, in: International Conference on Machine Learning, ACM, New York, USA, 2006, pp. 705–712.
[9] Z.-H. Zhou, K. Jiang, M. Li, Multi-instance learning based web mining, Appl. Intell. 22 (2) (2005) 135–147.
[10] A. Zafra, S. Ventura, E. Herrera-Viedma, C. Romero, Multiple instance learning with genetic programming for web mining, Comput. Ambient Intell. 4507 (2007) 919–927.
[11] P. Viola, J. C. Platt, C. Zhang, Multiple instance boosting for object detection, in: Advances in Neural Information Processing Systems, MIT Press, Vancouver, Canada, 2006, pp. 1419–1426.
[12] B. Babenko, N. Verma, P. Dollár, S. J. Belongie, Multiple instance learning with manifold bags, in: International Conference on Machine Learning, ACM, Bellevue, USA, 2011, pp. 81–88.
[13] M. Guillaumin, J. Verbeek, C. Schmid, Multiple instance metric learning from automatically labeled bags of faces, in: European Conference on Computer Vision, Lecture Notes in Computer Science, vol. 6311, Springer, Crete, Greece, 2010, pp. 634–647.
[14] S. Vijayanarasimhan, K. Grauman, Keywords to visual categories: multiple-instance learning for weakly supervised object categorization, in: Conference on Computer Vision and Pattern Recognition, 2008, pp. 1–8.
[15] K. Ali, K. Saenko, Confidence-rated multiple instance boosting for object detection, in: Conference on Computer Vision and Pattern Recognition, 2014, pp. 2433–2440.
[16] B. Babenko, M.-H. Yang, S. Belongie, Robust object tracking with online multiple instance learning, IEEE Trans. Pattern Anal. Mach. Intell. 33 (8) (2011) 1619–1632.
[17] J. Wang, J.-D. Zucker, Solving the multiple-instance problem: a lazy learning approach, in: International Conference on Machine Learning, ACM, San Francisco, USA, 2000, pp. 1119–1126.
[18] O. Maron, T. Lozano-Pérez, A framework for multiple-instance learning, in: Advances in Neural Information Processing Systems, MIT Press, Cambridge, USA, 1998, pp. 570–576.
[19] Q. Zhang, S.A. Goldman, EM-DD: an improved multiple-instance learning technique, in: Advances in Neural Information Processing Systems, MIT Press, Vancouver, Canada, 2001, pp. 1073–1080.
[20] T. Gärtner, P.A. Flach, A. Kowalczyk, A.J. Smola, Multi-Instance Kernels, in: International Conference on Machine Learning, ACM, Sydney, Australia, 2002, pp. 179–186.
[21] X. Xu, E. Frank, Logistic regression and boosting for labeled bags of instances, in: Advances in Knowledge Discovery and Data Mining, Lecture Notes in Computer Science, vol. 3056, Springer, Sydney, Australia, 2004, pp. 272–281.
[22] P. Gehler, O. Chapelle, Deterministic annealing for multiple-instance learning, in: International Conference on Artificial intelligence and Statistics, San Juan, Puerto Rico, 2007, pp. 123–130.
[23] R.C. Bunescu, R.J. Mooney, Multiple instance learning for sparse positive bags, in: International Conference on Machine Learning, ACM, New York, USA, 2007, pp. 105–112.
[24] F. Li, C. Sminchisescu, Convex multiple-instance learning by estimating likelihood ratio, Adv. Neural Inf. Process. Syst. 23 (2010) 1360–1368.
[25] Y. Li, D.M. Tax, R.P. Duin, M. Loog, Multiple-instance learning as a classifier combining problem, Pattern Recognit. 46 (3) (2013) 865–874.
[26] Q. Zhang, S.A. Goldman, W. Yu, J. Fritts, Content-based image retrieval using multiple-instance learning, in: International Conference on Machine Learning, ACM, San Francisco, USA, 2002, pp. 682–689.
[27] V. Cheplygina, D.M. Tax, M. Loog, Multiple instance learning with bag dissimilarities, Pattern Recognit. 48 (1) (2015) 264–275.
[28] Z.-H. Zhou, Y.-Y. Sun, Y.-F. Li, Multi-instance learning by treating instances as non-I.I.D. samples, in: International Conference on Machine Learning, ACM, New York, USA, 2009, pp. 1249–1256.
[29] J. Amores, Vocabulary-based approaches for multiple-instance data: a comparative study, in: International Conference on Pattern Recognition, ACM, Istanbul, Turkey, 2010, pp. 4246–4250.
[30] X. Song, L. Jiao, S. Yang, X. Zhang, F. Shang, Sparse coding and classifier ensemble based multi-instance learning for image categorization, Signal Process. 93 (1) (2013) 1–11.
[31] L.I. Kuncheva, Combining pattern classifiers: methods and algorithms, Wiley, Hoboken, NJ, 2004, p. 376, ISBN: 978-0-471-21078-8.
[32] Z.-H. Zhou, M.-L. Zhang, Solving multi-instance problems with classifier ensemble based on constructive clustering, Knowl. Inf. Syst. 11 (2) (2007) 155–170.
[33] J. Amores, Multiple instance classification: review, taxonomy and comparative study, Artif. Intell. 201 (2013) 81–105.
[34] J. Foulds, E. Frank, A review of multi-instance learning assumptions, Knowl. Eng. Rev. 25 (1) (2010) 1–25.
[35] G. Doran, S. Ray, A theoretical and empirical analysis of support vector machine methods for multiple-instance classification, Mach. Learn. 97 (1–2) (2014) 79–102.
[36] O.L. Mangasarian, E.W. Wild, Multiple instance classification via successive linear programming, J. Optim. Theory Appl. 137 (3) (2008) 555–568.
[37] Z.-H. Zhou, M.-L. Zhang, Ensembles of multi-instance learners, in: European Conference on Machine Learning, Lecture Notes in Computer Science, vol. 2837, Springer, Cavtat-Dubrovnik, Croatia, 2003, pp. 492–502.
[38] L. Breiman, Bagging predictors, Mach. Learn. 24 (2) (1996) 123–140.
[39] K. Lang, NewsWeeder: learning to filter netnews, in: International Conference on Machine Learning, 1995.
[40] M. Stone, Cross-validatory choice and assessment of statistical predictions, Journal of the Royal Statistical Society, Ser. B (Methodol.) 36 (2) (1974) 111–147.
[41] C.-C. Chang, C.-J. Lin, LIBSVM: a library for support vector machines, ACM Trans. Intell. Syst. Technol. 2 (3) (2011) 27:1–27:27.
[42] F.J. Provost, T. Fawcett, R. Kohavi, The case against accuracy estimation for comparing induction algorithms, in: International Conference on Machine Learning, ACM, San Francisco, USA, 1998, pp. 445–453.
[43] C. Ling, J. Huang, H. Zhang, AUC: a better measure than accuracy in comparing learning algorithms, in: Advances in Artificial Intelligence, Lecture Notes in Computer Science, vol. 2671, Springer, Vancouver, Canada, 2003, pp. 329–341.
[44] D. Tax, R. Duin, Learning curves for the analysis of multiple instance classifiers, in: Structural, Syntactic, and Statistical Pattern Recognition, Lecture Notes in Computer Science, vol. 5342, Springer, Orlando, USA, 2008, pp. 724–733.
[45] D. Tax, MIL, A Matlab Toolbox for Multiple Instance Learning, URL ⟨ http://prlab.tudelft.nl/david-tax/mil.html⟩, Version 1.1.0, 2015.
[46] S. Ray, M. Craven, Supervised versus multiple instance learning: an empirical comparison, in: International Conference on Machine Learning, ACM, New York, USA, 2005, pp. 697–704.
[47] V. Cheplygina, D.M.J. Tax, M. Loog, Dissimilarity-based ensembles for multiple instance learning, IEEE Trans. Neural Netw. Learn. Syst. (2015) 1–13.
[48] L. Ramon, Jan and De Raedt, Multi instance neural networks, in: International Conference on Machine Learning, ACM, Stanford, USA, 2000, pp. 53–60.
[49] P. Auer, On learning from multi-instance examples: empirical evaluation of a theoretical approach, in: International Conference on Machine Learning, ACM, San Francisco, USA, 1997, pp. 21–29.
[50] G. Ruffo, Learning single and multiple instance decision trees for computer security applications (Ph.D. thesis), Department of Computer Science, University of Turin, 2000.

Please cite this article as: M.-A. Carbonneau, et al., Robust multiple-instance learning ensembles using random subspace instance selection, Pattern Recognition (2016), http://dx.doi.org/10.1016/j.patcog.2016.03.035

[51] N. Weidmann, E. Frank, B. Pfahringer, A two-level learning method for generalized multi-instance problems, in: European Conference on Machine Learning, Lecture Notes in Computer Science, vol. 2837, Springer, Cavtat-Dubrovnik, Croatia, 2003, pp. 468–479.
[52] Y. Han, Q. Tao, J. Wang, Avoiding false positive in multi-instance learning, in: Advances in Neural Information Processing Systems, MIT Press, Vancouver, Canada, 2010, pp. 811–819.
[53] G. Hamerly, C. Elkan, Learning the k in k-means, in: Advances in Neural Information Processing Systems, vol. 16, MIT Press, Vancouver, Canada, 2004, pp. 281–288.
[54] S. Shalev-Shwartz, N. Srebro, SVM optimization: inverse dependence on training set size, in: International Conference on Machine Learning, ACM, Helsinki, Finland, 2008, pp. 928–935.

Marc-André Carbonneau received his B.Eng. degree in electrical engineering from École de technologie supérieure (Université du Québec), Montréal, Canada, in 2010. He is currently pursuing the Ph.D. in Machine Learning. His research interests include multiple instance learning, reinforcement learning, computer vision, action recognition and signal processing.

Eric Granger earned Ph.D. in EE from École Polytechnique de Montréal in 2001, and worked as a Defence Scientist at DRDC-Ottawa (1999–2001), and in R&D with Mitel Networks (2001–04). Until then, his research focused on neural networks signal processing and microelectronics for electronic surveillance. He joined the École de technologie supérieure (Université du Québec) in 2004, where he is Full Professor. Over the past decade, his research has focused on adaptive pattern recognition, computer vision and computational intelligence, with applications in biometrics, video surveillance, and computer/network security. Among his most significant innovations are adaptive multi-classifier systems for face recognition in video surveillance.

Alexandre J. Raymond received the B.Eng. and M.Eng. degrees from McGill University, Montréal, Québec, Canada, in 2007 and 2014, respectively. His research interests are in the design and implementation of low-level software and custom computing architectures.

Ghyslain Gagnon received the B.Eng. and M.Eng. degrees in electrical engineering from École de technologie supérieure, Montréal, Canada in 2002 and 2003 respectively. He also received the Ph.D. degree in electrical engineering from Carleton University, Canada in 2008. From 2003 to 2004, he worked for ISR Technologies where he designed and implemented several critical synchronization modules for a software defined radio which later obtained the editors' choice award in 2007 by the portable design magazine. He is now an associate professor with the Department of Electrical Engineering, École de technologie supérieure. He is inclined towards industrial research partnerships. His research aims at mixed signal circuits and systems, as well as digital signal processing.

Please cite this article as: M.-A. Carbonneau, et al., Robust multiple-instance learning ensembles using random subspace instance selection, Pattern Recognition (2016), http://dx.doi.org/10.1016/j.patcog.2016.03.035