Cluster Validation Measures for Label Noise Filtering

Veselka Boeva, Lars Lundberg
Computer Science and Engineering
Blekinge Institute of Technology
Karlskrona, Sweden
{veselka.boeva, lars.lundberg}@bth.se

Milena Angelova
Computer Systems and Technologies
TU of Sofia, Branch Plovdiv
Plovdiv, Bulgaria
mangelova@tu-plovdiv.bg

Jan Kohstall
Computer Science and Engineering
Blekinge Institute of Technology
Karlskrona, Sweden
jan.kohstall@bth.se

***Abstract*—Cluster validation measures are designed to find the partitioning that best fits the underlying data. In this paper, we show that these well-known and scientifically proven validation measures can also be used in a different context, i.e., for filtering mislabeled instances or class outliers prior to training in supervised learning problems. A technique, entitled CVI-based Outlier Filtering, is proposed in which mislabeled instances are identified and eliminated from the training set, and a classification hypothesis is then built from the set of remaining instances. The proposed approach assigns each instance several cluster validation scores representing its potential of being an outlier with respect to the clustering properties the used validation measures assess. We examine CVI-based Outlier Filtering and compare it against the LOF detection method on ten data sets from the UCI data repository using five well-known learning algorithms and three different cluster validation indices. In addition, we study two approaches for filtering mislabeled instances: local and global. Our results show that for most learning algorithms and data sets, the proposed CVI-based outlier filtering algorithm outperforms the baseline method (LOF). The greatest increase in classification accuracy has been achieved by combining at least two of the used cluster validation indices and global filtering of mislabeled instances.**

***Keywords*—class noise; classification; cluster validation measures; label noise**

I. Introduction

Supervised learning algorithms are used to generate classifiers [16]. For this machine learning task, the main idea is to apply a learning algorithm to detect patterns in a data set (inputs) that are associated with known class labels (outputs) in order to automatically create a generalization; i.e., a classifier. Under the assumption that the known data properly represent the complete problem studied, it is further assumed that the generated classifier will be able to predict the classes of new data instances. However, noise and outliers exist in real world data sets due to different errors. When the data are modeled using machine learning algorithms, the presence of label noise and outliers can affect the model that is generated. Improving how learning algorithms handle noise and outliers can produce better models.

Outlier mining is the process of finding unexpected events and exceptions in the data. There is a lot of work on outlier detection including statistical methods [17], rule creation [15], and clustering techniques [3]. Conventional outlier mining methods find exceptions or rare cases with respect to the whole data set.

In this paper, we introduce a novel outlier filtering technique that is close to class outlier detection approaches which find suspicious instances taking into account the class label [10], [11], [12], [18]. Such filtering approaches are also referred to as label noise cleansing [6]. The proposed approach, called Cluster Validation Index (CVI)-based Outlier Filtering, applies cluster validation measures to identify mislabeled instances or class outliers. We remove these instances prior to training and study how this effects the performance of the machine learning algorithm.

Cluster validation measures are usually used for evaluating and interpreting clustering solutions in unsupervised learning. However, we apply these well-known and scientifically proven measures in a different context; we use them for detecting mislabeled instances or class outliers in training sets in supervised learning scenarios. In supervised learning the clusters (in the form of classes) are known, and if there exists a strong relation among the instances of these clusters the classes of new instances can be accurately predicted. The intuition behind our approach is that instances in the training set that are not strongly connected to their clusters are mislabeled instances or class outliers and should be removed prior to training to improve the classification performance of the classifier. Our idea can also be considered into the context of cluster assumption, a notion originating in semi-supervised learning. The cluster assumption states that two points which are in the same cluster (i.e. which are linked by a high density path) are likely to be of the same label [5]. In this way by applying internal cluster validation measures on the classes of the training set we measure the degree of violation of the cluster assumption without explicitly computing clusters.

Our approach assigns each instance in the training set several cluster validation scores representing its potential of being a class outlier with respect to the clustering properties the used validation measures assess. In this respect, the proposed approach may be referred to a multi-criteria outlier filtering measure. Namely, it uses a combination of different cluster validation indices in order to reflect different aspects of the clustering model determined by the labeled instances of the training set.

978-1-5386-7097-2/18/$31.00 ©2018 IEEE

We evaluate the effects of mining class outliers for five commonly used learning algorithms on ten data sets from the UCI data repository using three different cluster validation measures (Silhouette Index (SI), Connectivity (Co) and Average Intracluster gap (IC-av)). In the current experimental setup the used cluster validation indices are combined by logical operators: $\vee$ (OR) and $\wedge$ (AND). We conduct twelve different experimental scenarios. For example, we have evaluated the instances of each data set with respect to SI, Co, IC-av, SI $\vee$ Co, SI $\wedge$ Co, SI $\vee$ IC-av etc. For further evaluation and validation of the proposed class outlier filtering approach we plan to study some ranked-based ensemble methods that are able to assemble cluster validation indexes in more informed way [14], [25]. In this context, we are also interested in defining an outlier scoring method that assigns a degree of outlierness to each instance in the training set.

In the current work, the CVI-based outlier filtering is compared against Local Outlier Factor (LOF) detection method [3], a well-known baseline algorithm. In addition, we study two approaches for filtering outliers: local and global. In case of local filtering we remove x percent from each class. In global filtering we relax the restriction that we need to filter out the same percentage from each class; it is enough that we filter x percent from the entire training set. Our results reveal that the proposed CVI-based outlier filtering approach outperforms the used baseline algorithm (LOF) for more of the conducted experimental scenarios. In addition, it is shown that for most learning algorithms and data sets the combination of SI and IC-av and global filtering of outliers produces the greatest increase in classification accuracy.

The rest of the paper is organized as follows. Section II reviews related works. Section III discusses the cluster validation measures and describes the proposed class outlier filtering approach. Section IV presents the initial evaluation of the introduced approach. Section V is devoted to conclusions and future work.

II. Related Work

A number of methods that treat individual instances in a data set differently during training to focus on the most informative ones have been developed. For example, an automated method that orders the instances in a data set by complexity based on their likelihood of being misclassified for supervised classification problems is presented in [22]. The underlying assumption of this method is that instances with a high likelihood of being misclassified represent more complex concepts in a data set. Authors have shown that focusing on the simpler instances during training significantly increases generalization accuracy. Identifying and removing noisy instances and outliers from a data set prior to training generally result in an increase in classification accuracy on nonfiltered test data [4], [7], [21].

Conventional outlier detection methods find exceptions or rare cases in a data set irrespective of the class label of these cases, whereas class outlier detection approaches find suspicious instances taking the class label into account [10], [11], [12], [18].

Papadimitriou and Faloutsos [18] proposed a solution of the problem when two classes of objects are given and it is necessary to find those which deviate with respect to the other class. Those points are called cross-outlier, and the problem is identified by cross-outlier detection.

He et al. [10] tried to find meaningful outliers that called Semantic Outlier Factor (SOF). The approach is based on applying a clustering algorithm on a data set with a class label, it is expected that the instances in every output cluster are to be identified with the same class label. However, this is not always true. The semantic outlier definition is a data point which behaves differently compared to other data points in the same class, but looks normal with respect to data points in another class. He et al. [11] further defined a general framework to contributions presented in [10], [18] by proposing a practical solution and extending existing outlier detection algorithms. The generalization does not consider only outliers that deviate with respect to their own class, but also outliers that deviate with respect to other classes.

Hewahi and Saad [12] introduced a novel definition for class outlier and a new method for mining class outliers based on distance-based approach and nearest neighbors. This method is called the CODB algorithm and is based on the concept of Class Outlier Factor (COF) which represents the degree of being a class outlier for a data object. The key factors of computing COF for an instance are the probability of the instance's class among its neighbors, the deviation of the instance from the instances of the same class, and the distance between the instance and its k nearest neighbors.

An interesting local outlier detection approach is proposed in [3]. It assigns to each object a degree of being an outlier. This degree is called the local outlier factor (LOF) of an object. It is local in that the degree depends on how isolated the object is with respect to the surrounding neighborhood.

Closely related to class outlier mining is noise reduction [20], [23] that attempts to identify and remove mislabeled instances. For example, Brodley and Friedl [4] attempt to identify mislabeled instances using an ensemble of classifiers. Rather than determining if an instance is mislabeled, the approach introduced in [21] filters instances that should be misclassified. A different approach by Zeng and Martinez [26] uses multi-layer perceptrons that changes the class label on suspected outliers assuming that the wrong label was assigned to that instance. A comprehensive survey on the different types of label noise, their consequences and the algorithms that consider label noise is presented by Frènay and M. Verleysen in [6]. In addition, a review of some typical problems associated with high-dimensional data and outlier detection specialized for high-dimensional data is published in [27].

III. Methods and Technical Solutions

A. Cluster Validation Techniques

One of the most important issues in cluster analysis is the validation of clustering results. The data mining literature provides a range of different cluster validation measures, which are broadly divided into two major categories: *external* and *internal* [13]. External validation measures have the benefit of providing an independent assessment of clustering quality, since they evaluate the clustering result with respect to a prespecified structure. However, previous knowledge about data is rarely available. Internal validation techniques, on the other hand, avoid the need for using such additional knowledge, but have the problem that they need to base their validation on the same information used to derive the clusters themselves. Internal measures can be split with respect to the specific clustering property they reflect and assess to find an optimal clustering scheme: *compactness*, *separation*, *connectedness*, and *stability* of the cluster partitions. A detailed and comparative overview of different types of validation measures can be found in [8], [24].

According to Bezdek and Pal [2], a possible approach to bypassing the selection of a single cluster validity criterion is to rely on multiple criteria in order to obtain more robust evaluations. In a recent work, Jaskowiak et al. proposed a method for combining internal cluster validation measures into ensembles, which show superior performance when compared to any single ensemble member [14]. In the context of the presented study, using a combination of several internal cluster validation measures to analyze the labeled instances prior to training may be referred to a multi-criteria outlier filtering measure which tries to find a trade-off between the evaluation performance of the combined measures.Thus in this work, we have selected to use three internal validation measures for analyzing the labeled instances prior to training in supervised classification problems in order to identify mislabeled ones. Based on the above mentioned classification, we have selected one validation measure for assessing compactness and separation properties of a partitioning - *Silhouette Index*, one for assessing connectedness - *Connectivity*, and one for assessing tightness and dealing with arbitrary shaped clusters - *IC-av*.

Silhouette Index (SI) [19] is a cluster validity index that is used to judge the quality of any clustering solution $C = C_1, C_2, \ldots, C_k$. Suppose a_i represents the average distance of object i from the other objects of the cluster to which the object is assigned, and b_i represents the minimum of the average distances of object i from objects of the other clusters. Then the *Silhouette Index* of object i can be calculated by $s(i) = (b_i - a_i)/\max\{a_i, b_i\}$. The overall Silhouette Index for clustering solution C of m objects, is defined as:

$$s(C) = \frac{1}{m}\sum_{i=1}^{m}(b_i - a_i)/\max\{a_i, b_i\}. \tag{1}$$

The values of silhouette index vary from -1 to 1 and higher value indicates better clustering results.

Connectivity (Co) captures the degree to which objects are connected within a cluster by keeping track of whether the neighboring objects are put into the same cluster [9]. Define m_{ij} as the jth nearest neighbor of object i, and let $\chi_{im_{ij}}$ be zero if i and m_{ij} are in the same cluster and $1/j$ otherwise. Then for a particular clustering solution $C = C_1, \ldots, C_k$ of m objects and a neighborhood size n_r, the *Connectivity* is defined as

$$Co(C) = \sum_{i=1}^{m}\sum_{j=1}^{n_r}\chi_{im_{ij}}. \tag{2}$$

The connectivity has a value between zero and mH_{n_r} and should be minimized. Evidently, the *Connectivity* of object i can be calculated by $Co(i) = \sum_{j=1}^{n_r}\chi_{im_{ij}}$.

IC-av estimates cluster tightness, but instead of assuming spherical shape, it assumes that clusters are connected structures with arbitrary shape [1]. Then for a particular clustering solution $C = C_1, \ldots, C_k$, the *IC-av* is define as

$$IC - av(C) = \sum_{r=1}^{k}\frac{1}{n_r}\sum_{i,j\in C_r} d_{ij}^2, \tag{3}$$

where n_r is the number of objects in cluster C_r ($r = 1, 2, \ldots, k$) and d_{ij} is maximum edge distance which represents the longest edge in the path joining objects i and j in the minimum spanning tree (MST) built on the clustered set of objects. The *IC-av Index* of object i, which is partitioned in cluster C_r, can be calculated by $IC - av(i) = 1/n_r \sum_{j\in C_r} d_{ij}^2$. The IC-av has a value between zero and the longest edge in the MST and should be minimized.

B. A Class Outlier Filtering Technique Using Cluster Validation Measures

In this study, we propose a class outlier filtering technique, entitled Cluster Validation Index (CVI)-based Outlier Filtering, that uses cluster validation measures to identify mislabeled instances.

Let us formally describe the proposed outlier filtering technique. We assume that X is the available set of labeled instances and $C = \{C_1, C_2, \ldots, C_k\}$ is the determined clustering partition of X, i.e., C_i denotes the set of instances belonging to class i. In addition, v[1] is the selected cluster validation index and P is a predicate defined on X and v. The pseudocode of the CVI-based Outlier Filtering algorithm is given in Algorithm 1.

Algorithm 1 CVI-based Outlier Filtering

```
function CVI-BASED OUTLIER FILTERING($X, C, v, P$)
    for all classes $C_i \in C$ do
        for all instances $c \in C_i$ do
            Calculate $v(c)$
            if $P(v, c)$ then
                Remove instance $c$ from $X$
            end if
        end for
    end for
end function
```

In this work, we use three internal validation measures for the evaluation of the labeled instances prior to training: Silhouette Index (SI), Connectivity and IC-av. Figure 1 shows a hypothetical 2-dimensional data set with two classes (circle

[1] v can represent a combination of several cluster validation indices connected by logical operators or assembled by a suitable ranked-based ensemble method.

and square) and three outliers (filled in two circles and one square). If we apply SI for assessing the instances of this data set instance 2 will be recognized as an outlier, while instance 1 will be removed in case of Connectivity is used. However, outlier instance 3 will be not considered as an outlier with respect to neither of SI and Connectivity measures. This instance would be filtered out as an outlier by IC-av measure estimating cluster tightness. The choice of cluster validation measure is therefore crucial for the performance of the proposed outlier mining technique. A rather straightforward solution to the described problem is to use different cluster validation measures in order to find some complementarity among the clustering properties they assess. In this way different aspects of the clustering model determined by the known class labels will be reflected in the filtering phase. For instance, the selected cluster validation measures can be combined by logical operators: $\vee$ (OR) or $\wedge$ (AND). This idea is further studied and validated by experimental scenarios explained in the following section. For our future work we also plan to study some ensemble methods for assembling cluster validation indexes in an guided way [14].

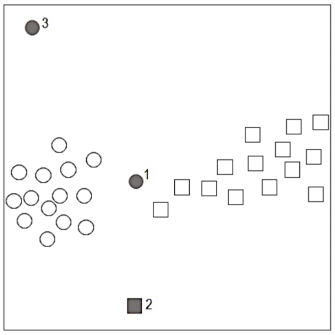

Fig. 1. A hypothetical 2-dimensional data set.

IV. Initial Evaluation and Results

A. Experimental Setup

We study how filtering of mislabeled instances affects the classification accuracy of 10 data sets from the UCI data repository and 5 learning algorithms trained with and without filtering. The algorithms that have been used are shown in Table I. We have chosen a diverse set of learning algorithms commonly used in practice. No parameter optimization has been performed on any of the algorithms. The selected data sets are listed in Table II. As one can see, these data sets vary significantly along important dimensions such as the number of attributes, the types of the attributes, the number of instances and the application domain.

Each method for filtering has been evaluated using 5 by 10-fold cross-validation (running 10-fold cross-validation 5 times, each time with a different seed to partition the data). Once the data sets are filtered, we evaluate each learning algorithm using 10-fold cross-validation using the filtered data set for training and the whole data set for testing. We then compare these results to those obtained by training the learning algorithm using all of the instances.

TABLE I. LIST OF LEARNING ALGORITHMS

Learning Algorithms
1-NN (1 nearest neighbor)
5-NN (5 nearest neighbor)
Support Vector Machine (SVM)
Gaussian Naïve Bayes (GNB)
Decision Tree (CART)

TABLE II. LIST OF DATA SETS

data sets	**#Instances**	**#Attributes**	**#Classes**	**Data Type**
digits	1797	64	10	Numeric
ecoli	336	8	8	
iris	150	4	3	
lung	32	56	3	
wine	178	13	3	
yeast	1484	8	10	
breast	286	9	2	Categorical
lenses	24	4	3	
arrhyth	452	279	16	Mixed
derma	366	34	6	

We also benchmark our class outlier filtering approach against LOF method [3], which is widely used as a baseline algorithm.

We have studied twelve different experimental scenarios (see Table III and Table IV). Initially, SI, Connectivity and IC-av scores are calculated for each instance of the considered data set. Then the instances in each class are ranked based on the assigned cluster validation scores separately for each measure. The latter is illustrated in Figure 2, which depicts the ranked SI, Connectivity and IC-av scores calculated on the instances of Iris data set. Then a local (w.r.t. the classes) or global (w.r.t. the entire data set) percentage of the top ranked instances can be identified and filtered out from the training set as outliers. In case of local filtering we remove x percent from each class; in global filtering it is enough that we filter x percent from the entire training set. In the first case one and the same number of instances with the lowest SI (respectively, the highest Connectivity or IC-av) scores will be removed from each class of the training set. However, the fact that the number of instances identified to be removed as outliers is fixed to be one and the same for each class may lead to a somewhat random choice of outliers. For instance, it may happen that the SI scores of some instances recognized as outliers are rather high, since these have only been included in the list of outliers in order to reach the required fixed number of instances. This can easily be noticed in Figure 2, e.g., see Class 0 Setosa (Silhouette Index). The described negative effect due to the use of local filtering can be mitigated by applying global filtering, instead. Namely, in this case a varying number of instances is removed as outliers from each class, since these are identified by a percent from the entire training set.

For each experimental scenario we have tested the following (local and global) percentages: 0%, 2%, 4%, 6%, 8%, 10%.

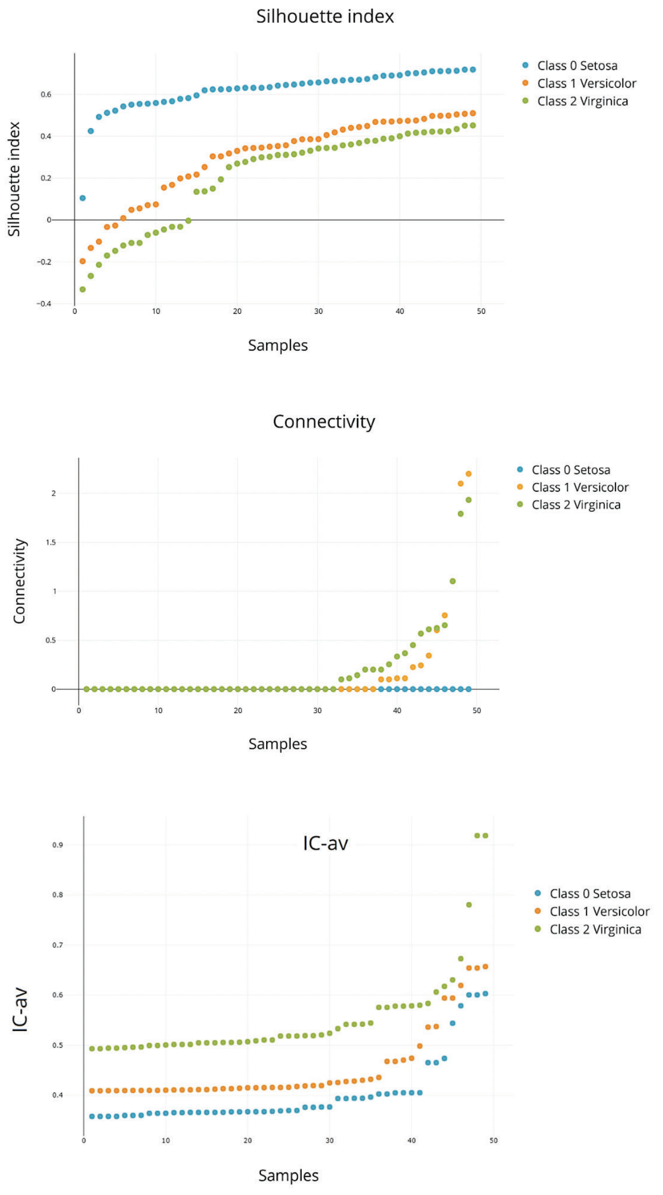

Fig. 2. SI, Connectivity and IC-av scores generated on the instances of Iris data set.

B. Implementation and Availability

The CVI-based Outlier Filtering algorithm used in our experiments has been implemented in Python. The experiments described in Section IV-A have been conducted on data downloaded from UCI Machine Learning Repository. For our experiments, we have downloaded the 10 data sets given in Table II. In our work we have used three different cluster validation measures: Silhouette Index, Connectivity and IC-av. Silhouette Index is used from the Python library scikit-learn. IC-av index has been implemented in Python according to the description given in Section III while Connectivity has been coded following its R script definition. We compare the CVI-based outlier filtering algorithm against LOF, a well-known baseline method [3]. We have used the implementation of LOF provided by scikit-learn with a neighborhood size of 20, an optimal number suggested by the authors. In addition, the used neighborhood size of Connectivity is 10, a default value of its R implementation. We have used the *F-measure* to evaluate the accuracy of the learning algorithms used in our experiments. We have used the scikit-learn implementation of the F-measure (*micro-average* F_1) in our experiments.

The executable of the CVI-based Outlier Filtering algorithm, the used data sets and the experimental results are available at GitLab[2].

C. Results and Discussion

The results are given in Table III and Table IV. Table III shows the average classification accuracy values produced by the twelve studied scenarios on the data sets listed in Table II, while Table IV presents the values for the corresponding twelve scenarios for global filtering, i.e., in that way there are 24 studied cases in total in the two tables. As one can notice the best results (0.835, 0.837 and 0.840) are obtained for SVM, 8% global filtering and SI $\vee$ IC-av, SI $\vee$ Co $\vee$ IC-av and IC-av $\vee$ Co, respectively. In general, SVM generates the highest classification accuracy values on the non-filtered data sets (see column 0%), i.e., there is a modest improvement of SVM due to outlier filtering. In addition, it is interesting to notice that IC-av index is involved in all the three combinations above. It also outperforms the other two indices (SI and Co) in case of SVM, GNB and CART, but it has not generated any improvement for 1-NN and 5-NN algorithms. The three cluster validation measures perform better than LOF for almost all the studied learning algorithms except for SVM (local filtering) and GNB (global filtering). However, some combinations of the indices (e.g., see IC-av $\vee$ Co (local filtering) and SI $\vee$ IC-av (global filtering)) outperform LOF in these scenarios.

Figure 3 depicts the average improvement for all eleven experimental scenarios involving the cluster validation indices given in Table III and Table IV, i.e., the line "Local" shows the average improvement in Table III (excluding LOF) and the line "Global" represents the average improvement in Table IV (excluding LOF). It is clear that global filtering gives the best values (see the discussion in Section IV-A). In addition, 6-8% filtering gives the best results for the used data sets and learning algorithms.

Figure 4 presents, for each learning algorithm, the average improvement for all 22 studied cases involving the cluster validation indices (i.e., excluding LOF). As it can be seen, the GNB algorithm has the largest improvement in classification accuracy. This means that removing outliers is particularly beneficial for GNB. We believe that the reason for this is that the Gaussian curve that is used in the GNB algorithm is very sensitive to outliers. In addition, the gained improvement produced by the 1-NN learning algorithm is higher than the one generated by the 5-NN algorithm. The 5-NN algorithm

[2] https://gitlab.com/machine_learning_vm/outliers

[3] Cluster Validation Index

TABLE III. THE AVERAGE ACCURACY FOR THE FIVE CONSIDERED LEARNING ALGORITHM USING LOCAL FILTERING FOR MINING OUTLIERS FROM THE DATA SETS.

CVI[3]	**LA**	**0%**	**2%**	**4%**	**6%**	**8%**	**10%**
LOF	1-NN	0.795	0.796	0.796	**0.799**	**0.799**	0.798
	5-NN	**0.797**	0.797	0.796	0.797	0.796	0.795
	SVM	0.820	0.820	0.821	**0.829**	0.828	0.827
	GNB	0.739	0.799	0.798	**0.806**	0.796	0.796
	CART	0.793	**0.794**	0.792	0.792	0.791	0.789
SI	1-NN	0.795	0.797	**0.800**	0.797	0.799	0.796
	5-NN	0.797	0.796	**0.800**	0.799	0.799	0.799
	SVM	0.820	0.820	**0.821**	0.819	0.815	0.819
	GNB	0.739	0.776	0.787	**0.789**	0.781	0.781
	CART	0.793	0.798	0.792	0.792	0.791	**0.799**
Co	1-NN	0.795	0.798	0.797	0.796	0.802	**0.804**
	5-NN	**0.797**	0.797	0.792	0.792	0.790	0.786
	SVM	0.820	0.820	0.820	0.820	0.825	**0.827**
	GNB	0.739	0.740	0.747	0.755	0.758	**0.762**
	CART	0.793	0.796	0.797	0.803	0.802	**0.804**
IC-av	1-NN	0.795	**0.796**	0.794	0.795	0.794	0.792
	5-NN	**0.797**	0.796	0.796	0.796	0.795	0.793
	SVM	0.820	0.820	0.827	**0.828**	0.827	**0.828**
	GNB	0.739	0.798	0.802	**0.810**	0.798	0.798
	CART	0.793	0.792	0.793	**0.807**	0.799	0.801
SI $\wedge$ Co	1-NN	0.795	0.797	0.798	0.797	**0.799**	0.796
	5-NN	**0.797**	0.796	0.796	0.795	0.797	0.796
	SVM	**0.820**	**0.820**	0.820	0.816	0.816	0.816
	GNB	0.739	0.740	0.743	0.754	**0.758**	**0.758**
	CART	0.793	**0.797**	0.796	0.793	0.796	0.796
SI $\vee$ Co	1-NN	0.795	0.798	0.798	0.797	0.802	**0.805**
	5-NN	**0.797**	0.797	0.795	0.796	0.793	0.788
	SVM	0.820	0.820	0.819	0.819	0.822	**0.828**
	GNB	0.739	0.776	**0.792**	0.791	0.781	0.784
	CART	0.793	0.798	0.796	0.797	0.802	**0.811**
SI $\wedge$ IC-av	1-NN	**0.795**	0.795	0.795	0.795	0.794	0.794
	5-NN	**0.797**	0.796	0.796	0.796	0.795	0.793
	SVM	0.820	0.820	0.827	**0.828**	0.827	**0.828**
	GNB	0.739	0.776	**0.787**	0.785	0.786	0.785
	CART	0.793	0.795	0.793	0.792	0.793	**0.801**
SI $\vee$ IC-av	1-NN	0.795	0.797	**0.799**	0.797	0.798	0.794
	5-NN	0.797	0.796	**0.800**	0.796	0.795	0.795
	SVM	0.820	0.821	**0.830**	0.826	0.825	0.821
	GNB	0.739	0.797	0.801	**0.812**	0.801	0.801
	CART	0.793	0.797	0.801	0.807	**0.809**	0.800
IC-av $\wedge$ Co	1-NN	0.795	0.795	0.795	0.795	**0.796**	**0.796**
	5-NN	0.797	0.797	0.797	0.797	0.797	0.797
	SVM	0.820	0.820	0.820	0.820	**0.822**	**0.822**
	GNB	0.739	0.740	0.746	0.755	0.759	**0.761**
	CART	0.793	0.793	**0.794**	**0.794**	**0.794**	**0.794**
IC-av $\vee$ Co	1-NN	0.795	0.799	0.796	0.796	**0.801**	**0.801**
	5-NN	**0.797**	0.796	0.791	0.788	0.785	0.779
	SVM	0.820	0.820	0.830	0.831	**0.832**	0.829
	GNB	0.739	0.797	0.803	**0.810**	0.795	0.796
	CART	0.793	0.796	0.799	0.804	0.804	**0.811**
SI $\wedge$ Co $\wedge$ IC-av	1-NN	0.795	0.795	0.795	0.795	0.795	0.795
	5-NN	0.797	0.797	0.797	0.797	0.797	0.797
	SVM	0.820	0.820	0.820	0.820	0.820	0.820
	GNB	0.739	0.739	0.743	0.753	**0.755**	**0.755**
	CART	0.793	0.794	0.794	0.794	**0.795**	**0.795**
SI $\vee$ Co $\vee$ IC-av	1-NN	0.795	0.800	0.798	0.797	**0.802**	**0.802**
	5-NN	**0.797**	0.796	0.794	0.792	0.788	0.783
	SVM	0.820	0.820	**0.829**	0.825	0.825	0.819
	GNB	0.739	0.797	0.802	**0.809**	0.800	0.801
	CART	0.793	0.797	0.803	0.806	**0.812**	0.810

TABLE IV. THE AVERAGE ACCURACY FOR THE FIVE CONSIDERED LEARNING ALGORITHM USING GLOBAL FILTERING FOR MINING OUTLIERS FROM THE DATA SETS.

CVI	**LA**	**0%**	**2%**	**4%**	**6%**	**8%**	**10%**
LOF	1-NN	0.795	0.795	**0.798**	0.797	0.797	0.797
	5-NN	**0.797**	0.795	0.790	0.786	0.786	0.783
	SVM	0.820	0.827	**0.829**	0.824	0.824	0.822
	GNB	0.739	0.796	0.806	0.806	**0.807**	0.802
	CART	0.793	0.791	**0.800**	0.795	0.795	0.797
SI	1-NN	0.795	0.796	0.796	**0.800**	0.798	0.798
	5-NN	0.797	0.794	0.798	**0.799**	0.798	0.790
	SVM	0.820	0.816	0.816	0.822	**0.825**	0.820
	GNB	0.739	0.783	0.787	0.787	0.784	**0.799**
	CART	0.793	0.793	0.793	0.797	**0.802**	0.801
Co	1-NN	0.795	0.802	0.805	**0.807**	0.803	0.804
	5-NN	0.797	**0.801**	0.794	0.795	0.793	0.789
	SVM	0.820	0.827	0.827	**0.833**	0.828	0.822
	GNB	0.739	0.738	0.742	**0.754**	**0.754**	0.753
	CART	0.793	0.791	0.795	0.799	0.801	**0.804**
IC-av	1-NN	**0.795**	0.794	0.794	0.793	0.793	0.791
	5-NN	**0.797**	0.795	0.795	0.794	0.794	0.788
	SVM	0.820	0.826	0.831	0.830	**0.834**	0.830
	GNB	0.739	0.796	**0.805**	0.804	0.804	0.786
	CART	0.793	0.792	**0.798**	0.792	0.787	0.784
SI $\wedge$ Co	1-NN	0.795	0.797	0.798	0.800	0.802	**0.805**
	5-NN	0.797	0.797	0.796	**0.798**	**0.798**	0.795
	SVM	0.820	0.820	0.820	**0.823**	0.822	0.818
	GNB	0.739	0.738	0.741	0.754	0.756	**0.757**
	CART	0.793	0.795	0.797	0.794	0.794	**0.799**
SI $\vee$ Co	1-NN	0.795	0.799	0.801	**0.806**	0.799	0.799
	5-NN	**0.797**	0.797	0.793	0.791	0.787	0.779
	SVM	0.820	0.820	0.819	0.824	**0.829**	0.827
	GNB	0.739	0.782	0.784	0.786	0.780	**0.794**
	CART	0.793	0.799	0.803	0.806	**0.812**	0.810
SI $\wedge$ IC-av	1-NN	0.795	0.795	0.795	0.795	0.795	0.794
	5-NN	**0.797**	0.796	0.796	0.796	0.795	0.795
	SVM	0.820	0.820	0.820	0.820	**0.825**	0.823
	GNB	0.739	0.780	**0.781**	**0.781**	0.779	0.794
	CART	0.793	0.794	0.795	0.795	0.796	**0.797**
SI $\vee$ IC-av	1-NN	0.795	0.795	0.795	**0.797**	0.795	0.794
	5-NN	**0.797**	0.793	0.796	0.797	0.796	0.786
	SVM	0.820	0.822	0.828	**0.835**	**0.835**	0.824
	GNB	0.739	0.809	0.808	0.806	**0.810**	0.795
	CART	0.793	0.791	0.800	0.797	0.798	**0.811**
IC-av $\wedge$ Co	1-NN	0.795	0.795	0.796	0.796	0.796	**0.797**
	5-NN	**0.797**	0.796	0.796	0.796	0.796	0.796
	SVM	**0.820**	**0.820**	0.819	0.819	0.819	0.819
	GNB	0.739	0.738	0.741	0.752	0.754	**0.756**
	CART	0.793	0.793	0.794	**0.795**	**0.795**	**0.795**
IC-av $\vee$ Co	1-NN	0.795	0.801	0.803	**0.805**	0.799	0.797
	5-NN	0.797	**0.800**	0.792	0.793	0.790	0.785
	SVM	0.820	0.829	0.835	0.839	**0.840**	0.836
	GNB	0.739	0.800	**0.807**	0.802	0.797	0.781
	CART	0.793	0.794	0.797	0.797	0.795	**0.799**
SI $\wedge$ Co $\wedge$ IC-av	1-NN	0.795	0.795	0.795	0.795	0.796	**0.797**
	5-NN	**0.797**	0.797	0.796	0.796	0.796	0.796
	SVM	**0.820**	0.820	0.820	0.820	0.820	0.818
	GNB	0.739	0.738	0.740	0.752	0.754	**0.755**
	CART	0.793	0.793	**0.794**	**0.794**	**0.794**	**0.794**
SI $\vee$ Co $\vee$ IC-av	1-NN	0.795	0.798	0.800	**0.803**	0.796	0.795
	5-NN	**0.797**	0.796	0.790	0.790	0.786	0.775
	SVM	0.820	0.824	0.831	**0.837**	**0.837**	0.826
	GNB	0.739	**0.810**	0.809	0.804	0.809	0.795
	CART	0.793	0.799	0.804	0.809	0.811	**0.812**

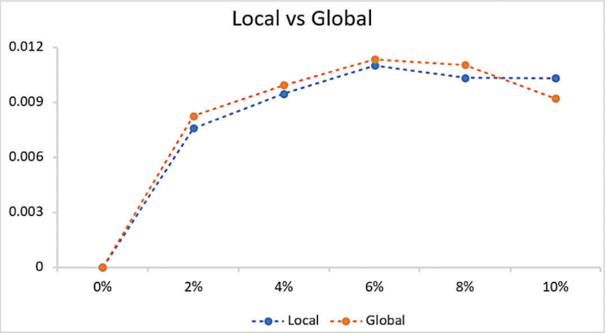

Fig. 3. Average gain of classification accuracy for all learning algorithms.

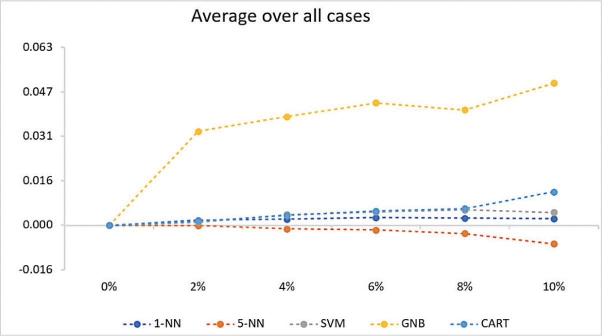

Fig. 4. Average gain of classification accuracy for each learning algorithm.

is naturally robust against label noise and therefore it will not benefit much from label noise filtering. On the other hand, the 1-NN algorithm is more sensitive to noise. Therefore we can see a bigger improvement in classification accuracy due to filtering for 1-NN than for 5-NN.

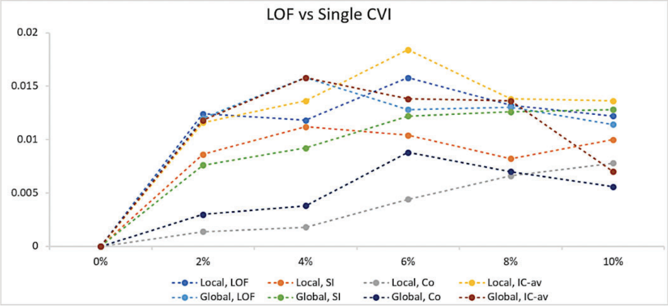

Fig. 5. Comparison of average gain of classification accuracy of SI, Co and IC-av against LOF.

Figure 5 benchmarks the single cluster validation indices against LOF. As one can notice IC-av outperforms LOF and the other two cluster validation indices (SI and Co) for 6-8% filtering. However, LOF generates higher results than SI and Co in case of 2-8% local and global filtering. Therefore in Figure 6 and Figure 7 we further study how combining IC-av with SI and Co will affect the average gain of classification accuracy of SI ∨ IC-av, SI ∧ IC-av, Co ∨ IC-av and Co ∧ IC-av, respectively. Interestingly, SI ∨ IC-av shows higher improvement than LOF in case of 2-10% local and global filtering. The latter is also true for Co ∨ IC-av in case of 6-10% local and 2-8% global filtering, respectively (see Figure 7). Consequently, the performance of our CVI-based outlier filtering approach can easily be improved by adding a suitable cluster validation measure. In other words, it can be customized to the data set under consideration by initially studying the performance of the involved individual cluster validation measures on this set.

Figure 8 presents a comparison of the average improvement for intersection and union scenarios for local and global filtering. Logically the latter scenario outperforms the former one for all the studied cases. This is also supported by the following discussion.

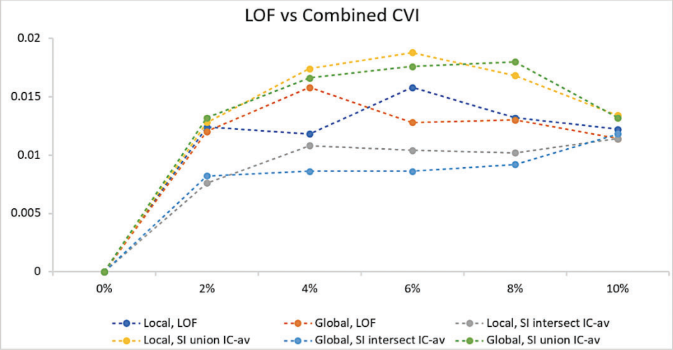

Fig. 6. Comparison of average gain of classification accuracy of SI ∨ IC-av and SI ∧ IC-av against LOF.

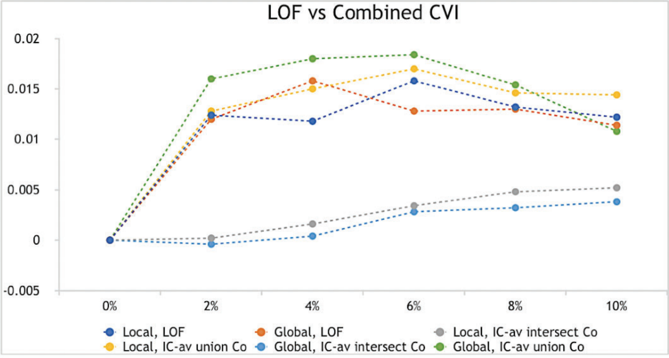

Fig. 7. Comparison of average gain of classification accuracy of Co ∨ IC-av and Co ∧ IC-av against LOF.

Comparing the amount of outliers that have been filtered out, it turns out that the union of two CVIs is on average 1.82 times the size of the sets removed by each of the single indices. The biggest absolute amount is achieved by combining the three cluster validation measures. It is on average 2.52 times the size of the single measures. The highest discrepancy is found between IC-av and Connectivity. For example, in case of union between them the filtered-out instances are 1.90 times the size of the single CVIs, while in case of intersection the filtered-out instances are only 0.10 times the size of the single measures. This also explains the results in Figure 7. SI and IC-av find higher amount of joint outliers. However with the increasing amount of filtered-out data the outliers are getting more disjoint which results in a bigger set of removed outliers, especially for 8% and 10% filtering. The latter can be noticed in Figure 6, where the gained improvement degrades when the number of out-filtered data increases. The main conclusion from the above discussion is that by combining several cluster validation measures we can improve the performance of the proposed outlier filtering approach.

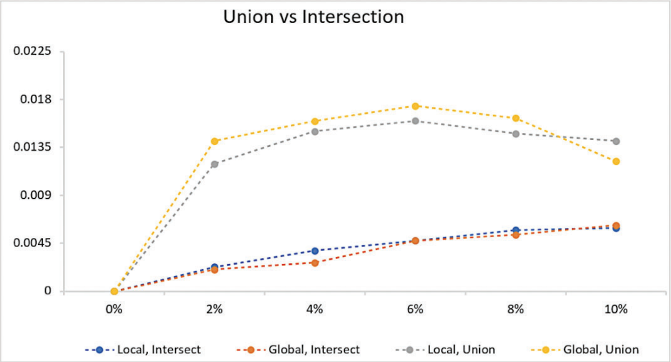

Fig. 8. Average gain of classification accuracy for intersection and union scenarios.

V. Conclusion and Future Work

In this work, we have proposed an outlier filtering approach, entitled CVI-based Outlier Filtering, that applies cluster validation measures to identify mislabeled instances. The implemented version of the CVI-based algorithm uses three internal cluster validation measures. In addition, two different filtering approaches have been studied: local and global. The proposed approach has been evaluated and compared against the LOF detection method for five commonly used learning algorithms on ten data sets from the UCI data repository. The obtained results demonstrate that the proposed approach is a robust outlier filtering technique that is able to improve classification accuracy of the learning algorithm. Our approach can easily be adapted to different data sets and learning algorithms by using a proper combination of cluster validation measures reflecting different clustering properties. It turns out that the CVI-based algorithm outperforms LOF in most cases, particularly when we combine several cluster validation measures.

For future work, the aim is to pursue further enhancement and validation of the proposed outlier filtering approach by applying alternative cluster validation measures on a higher variety of data sets and learning algorithms. We also plan to study ranked-based ensemble methods for assembling the selected cluster validation indexes in a guided way. In addition, our future intention is to extend the proposed approach by a data correction (fixing) phase. Presently, the outliers are identified and eliminated by the training set; an alternative approach would be to relabel the outliers instead of eliminating them.

Acknowledgment

This work is part of the research project *"Scalable resource-efficient systems for big data analytics"* funded by the Knowledge Foundation (grant: 20140032) in Sweden.

References

[1] A. E. Bay*á*, and P. M. Granitto, “How many clusters: A validation index for arbitrary-shaped clusters,“ IEEE/ACM Transactions on Computational Biology and Bioinformatics, vol. 10(2), pp. 401-414, 2013.

[2] J.C. Bezdek, N.R. Pal, “Some new indexes of cluster validity,“ IEEE Trans. on Syst., Man and Cyb., Part B, vol. 28(3), pp. 301-315, 1998.

[3] M. M. Breunig et al., “LOF: identifying density-based local outliers,“ SIGMOD Record, vol. 29(2), pp. 93-104, 2000.

[4] C. E. Brodley, and M. A. Friedl, “Identifying mislabeled training data,“ Journal of Artificial Intelligence Research, vol. 11, pp. 131-167, 1999.

[5] O. Chapelle, B. Scholkopf, A. Zien, “Semi-Supervised learning“, MIT Press, 2006.

[6] B. Frènay, and M. Verleysen, “Classification in the presence of Label Noise: a survey,“ IEEE Trans. on Neural Networks and Learning Syst., vol. 25(5), pp. 845-869, 2014.

[7] D. Gamberger, N. Lavrac, and S. D*ž*eroski, “Noise detection and elimination in data pre-processing: Experiments in medical domains,“ Applied Artificial Intelligence, vol. 14(2), pp. 205-223, 2000.

[8] M. Halkidi, Y. Batistakis, M.Vazirgiannis, “On clustering validation techniques,“ J. of Int. Inf. Syst., vol. 17 2(3), pp. 107-145, 2001.

[9] J. Handl et al., “Computational cluster validation in post-genomic data analysis,“ Bioinformatics, vol. 21, pp. 3201-3212, 2005.

[10] Z. He, S. Deng, X. Xu, “Outlier detection integrating semantic knowledge,“ WAIM’02, pp. 126-131, 2002.

[11] Z. He, X. Xu, J. Huang, S. Deng,“Mining Class Outliers: Concepts, algorithms and applications in CRM,“ Expert Systems with Applications, vol. 27(4), pp. 681-697, 2004.

[12] N. Hewahi, M. Saad, “Class Outliers Mining: Distance based-approach,“ Int. J. of Intelligent Syst. and Technologies, vol. 2(1), pp. 55-68, 2007.

[13] A. K. Jain, R. C. Dubes, “Algorithms for clustering data,“ Prentice Hall, Englewood Cliffs, NJ, 1988.

[14] Jaskowiak, “On strategies for building effective ensembles of relative clustering validity criteria,“ K. and Inf. Sys., vol. 47, pp. 329-354, 2016.

[15] T. M. Khoshgoftaar, N. Seliya, and K. Gao, “Rule-based noise detection for software measurement data,“ The IEEE int. conf. on inf. Reuse and Integration, IEEE Syst., Man, and Cybern. Society, pp. 302-307, 2004.

[16] R. Kohavi, “A study of cross-validation and bootstrap for accuracy estimation and model selection,“ 14th Int. Joint Conference on AI, vol. 2, pp. 1137-1143, 1995.

[17] J. M. Kubica and A. Moore, “Probabilistic noise identification and data cleaning,“ The 3rd IEEE Int. Conf. on Data Mining, IEEE Comput. Society, pp. 131-138, 2003.

[18] S. Papadimitriou, C. Faloutsos, “Cross-outlier detection,“ SSTD’03, pp. 199-213, 2003.

[19] P. Rousseeuw, “Silhouettes: a graphical aid to the interpretation and validation of cluster analysis,“ Journal of Computational Applied Mathematics, vol. 20, pp. 53-65, 1987.

[20] N. Segata, and E. Blanzieri, “Fast and scalable local kernel machines,“ Journal of Machine Learning Research, vol. 11, pp. 1883-1926, 2010.

[21] M. R. Smith, and T. Martinez, “Improving classification accuracy by identifying and removing instances that should be misclassified,“ The 2011 Int. Joint Conference on Neural Networks, pp. 2690-2697, 2011.

[22] M. R. Smith, and T. Martinez, “A Comparative evaluation of curriculum learning with filtering and boosting in supervised classification problems,“ Computational Intelligence, vol. 32(2), pp. 167-195, 2016.

[23] I. Tomek, “An experiment with the edited nearestneighbor rule,“ IEEE Trans. on Systems, Man, and Cybernetics, vol. 6, pp. 448-452, 1976.

[24] L. Vendramin, R.J.G.B. Campello, and ĐŢ. R. Hruschka, “Relative clustering validity criteria: a comparative overview,“ 2010 Wiley Periodicals, Inc. Statistical Analysis and Data Mining, vol. 3, pp. 209-235, 2010.

[25] L. Vendramin et al., “On the combination of relative clustering validity criteria,“ The 25th Int. Conf. on SSDBM, pp. 4:1-4:12, 2013.

[26] X. Zeng, and T. R. Martinez, “An algorithm for correcting mislabeled data,“ Intelligent Data Analysis, vol. 5, pp. 491-502, 2001.

[27] A. Zimek, E. Schubert, and H-P. Kriegel, “A survey on unsupervised outlier detection in high-dimensional numerical data,“ 2012 Wiley Periodicals, Inc. Stat. Anal. and Data Mining, vol. 5, pp. 363-387, 2012.