IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received February 17, 2019, accepted March 9, 2019, date of publication March 21, 2019, date of current version April 13, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2906644

Knowledge Reused Outlier Detection

WEIREN YU[1], ZHENGMING DING[2], (Member, IEEE), CHUNMING HU[1], AND HONGFU LIU[3], (Member, IEEE)

[1]School of Computer Science, Beihang University, Beijing 100191, China
[2]Department of Computer, Information, and Technology, Indiana University-Purdue University Indianapolis, Indianapolis, IN 46202, USA
[3]Department of Computer Science, Brandeis University, Waltham, MA 02453, USA

Corresponding author: Weiren Yu (yuwr@act.buaa.edu.cn)

This work was supported in part by the China 973 Plan under Grant 2014CB340300, in part by the NSFC program under Grant 61472022 and SKLSDE-2016ZX-11, and in part by the Beijing Advanced Innovation Center for Big Data and Brain Computing.

ABSTRACT Tremendous efforts have been invested in the unsupervised outlier detection research, which is conducted on unlabeled data set with abnormality assumptions. With abundant related labeled data available as auxiliary information, we consider transferring the knowledge from the labeled source data to facilitate the unsupervised outlier detection on target data set. To fully make use of the source knowledge, the source data and target data are put together for joint clustering and outlier detection using the source data cluster structure as a constraint. To achieve this, the categorical utility function is employed to regularize the partitions of target data to be consistent with source data labels. With an augmented matrix, the problem is completely solved by a K-means-- a based method with the rigid mathematical formulation and theoretical convergence guarantee. We have used four real-world data sets and eight outlier detection methods of different kinds for extensive experiments and comparison. The results demonstrate the effectiveness and significant improvements of the proposed methods in terms of outlier detection and cluster validity metrics. Moreover, the parameter analysis is provided as a practical guide, and noisy source label analysis proves that the proposed method can handle real applications where source labels can be noisy.

INDEX TERMS Outlier detection, transfer learning, K-means, joint clustering and outlier detection, knowledge transfer, knowledge reuse.

I. INTRODUCTION

Outlier detection, also known as anomaly detection, aims to identify the minority data points with distinctive characters from the majority, which has wide real-world applications, including credit card fraud, network intrusion, and precision marketing and so on. Tremendous efforts have been invested in this area, especially on unsupervised outlier detection. Various methods are put forward from different mathematical aspects, including density-based LOF [1], COF [2], distance-based LODF [3], angle-based FABOD [4], ensemble-based iForest [5], eigenvector-based OPCA [6], cluster-based TONMF [7], and so on. More details on outlier detection can be found in [8], [9].

Although these methods achieve reasonable performance in some use cases, there is still huge space for further improvement. The extensibility of existing methods are limited for supporting more flexible scenarios. With the rapid development of intelligent techniques, massive data has been created and collected for analysis to provide informed decision making. To better understand these data, tremendous human efforts have been taken for labeling and tagging, where similar documents are grouped together with predefined class labels, and photos or videos with multiple tags. Considering the sheer amount of the data, it is appealing to label and tag for the newly arrived data automatically. Naturally we apply the classification or clustering models to cope with the above task. However, it is quite common that the new arrival data does not belong to any of the predefined classes.

In classification or clustering, few outliers can easily destroy the structure of the class or cluster [10]. Meanwhile, the outliers are defined by the concept of cluster, where the data points don't belong to any of the clusters [1]. Similarly, the class or cluster structure in labeled data sets are also helpful in detecting outliers in the newly arrived data. With tremendous labeled data available, we wonder whether the performance of the unsupervised outlier detection task can be improved with these auxiliary labeling information. In light of this new scenario, we propose a new problem,

The associate editor coordinating the review of this manuscript and approving it for publication was Lu Liu.

knowledge reused outlier detection, where auxiliary labeling information will be exploited to facilitate the outlier detection on the target data. We face two challenges in this scenario, 1) the inherent challenge of outlier detection and clustering, and 2) the challenge on effectively transferring the knowledge from labeled data to outlier detection in target data set.

To achieve this, we put the target data together with labeled data (information source) for joint clustering and outlier detection, where the data points with large distance to their nearest centroids are regarded as outliers. Drawing inspiration from consensus clustering, categorical utility function is used as regularization to preserve the source data cluster structure as a constraint on data partitions, which provides supervision on clustering of the source and target data. Based on this, the objective function is provided for the knowledge reused outlier detection task. To solve it elegantly, a modified K-means-- is designed with a neat mathematical formulation and theoretical convergence guarantee. To demonstrate the competitiveness and effectiveness of our proposed method, extensive experiments have been performed. We use four real world data sets and compared with 8 algorithms over various outlier detection and cluster validity metrics. We highlight our major contributions as follows.

- For the problem, We consider the new scenario of knowledge reused outlier detection, which is an extension of the traditional unsupervised outlier detection. Auxiliary data with labels are employed to faciliate the outlier detection on the target data.
- For the methodology, we focus on joint clustering and outlier detection to preserve cluster structure from information source for knowledge transfer. The categorical utility function plays as a regularizer to make the learnt cluster structure on target data consistent with information source data.
- For the mathematical solution, We formulate this problem as a semi-supervised clustering problem with missing label values, which is completely solved by a modified K-means-- on an augmented matrix in rigid mathematical formulation with convergence guarantee.

II. RELATED WORK

In this section, we provide the related work analysis in terms of unsupervised outlier detection and transfer learning.

Outlier detection, also known as anomaly detection, seeks the points deviated from others and identifies them as outliers. Most of the existing studies focus on unsupervised outlier detection. Some criteria are designed to assign a score to each point, and the points with large scores are regarded as the outlier candidates. There are some representative methods: density-based LOF [1], COF [2], distance-based LODF [3], frequent pattern-based Fp-outlier [11], angle-based ABOD [12] and its fast version FABOD [4], ensemble-based iForest [5], BSOD [13], eigenvector-based OPCA [6], cluster-based TONMF [7].

Cluster analysis and outlier detection are both hot topics consistently in data mining area. However, they are usually considered as two independent tasks. Although robust clustering has resistance against the impact of outliers, and each data point including outliers is assigned a cluster label. Few of the existing works treat the cluster analysis and outlier detection in a unified framework. Unified outlier detection and cluster analysis has been theoretically studied in the context of facility location problem. Charikar *et al.* proposed a bi-criteria approximation algorithm for the facility location problem with outliers problem [14]. Chen proposed a constant factor approximation algorithm for the K-median with outliers problem [15]. K-measn-- [16] detects o outliers and partitions the rest points into K clusters, where the instances with large distance to the nearest centroid are regarded as outliers during the clustering process. Langrangian Relaxation (LP) [17] formulates the clustering with outliers as an integer programming problem, which requires the cluster creation cost to be the input parameter. None of the methods leverage the knowledge from available labeled data as information source.

Transfer learning has been a successful and attractive method in computer vision and pattern recognition, including a lot of real-world applications e.g., collaborative recommendation [18], text categorization [19], image classification [20], and sentiment analysis [21]. More specifically, transfer learning is a technique designed to facilitate the tasks in less labeled and known target domain with the labeled source domain information. The distribution of the target domain is usually different from the distribution of the source domain. For example, object images from Amazon website has rich labels. Photos captured by digital cameras has fewer labels. When trying to recognize photos from the cameras, we can use object images from Amazons to help train the model. In this case, the key challenge here is to mitigate the distribution differences between the source data and the target data, by adapting either or both of them. Among transfer learning techniques, transductive transfer learning is the category that applies to the scenario when the same or similar tasks will be performed on the source and target data, but the two data domains have different feature space or data distribution [22]. Through feature or classifier adaptation, learning task on target data will be facilitated with well-learned source domain knowledge. Over the past decades, a variety of transfer learning algorithms have been proposed and achieved promising performance, e.g. feature adaptation [23], classifier adaptation [24], [25] and dictionary learning [22], [26]. More details can be referred to the excellent survey [27].

III. PROBLEM DEFINITION

In this section, we formally define the problem of knowledge reused outlier detection. Leveraging the knowledge from the labeled data as information source, we try to simultaneously uncover the target data points that belong to unseen class as outliers and label the rest of the target data points with the classes as defined in the information source.

Let X_s denote the features of the information source data matrix with n_s instances and d features, and Y_s is the corresponding label matrix with K as the number of classes. X_t is the newly arrived or target data matrix with n_t instances and d features for labeling. Our task is to identify l data points in X_t as outliers and partition the rest $n_t - l$ data points into K classes.

Since the information source data X_s with K classes cannot model the unseen classes in the target data, the traditional classification methods cannot classify a data point into an unseen class. Hence they cannot work on this scenario. On the other hand, without the information source data X_s, the problem degrades into the classical unsupervised outlier detection problem. In this case, the information from source data is not fully utilized. Therefore, the key point here is how to employ the knowledge in X_s to facilitate the joint cluster analysis and outlier detection task on X_t.

Cluster analysis and outlier detection are tightly coupled together. We notice that few outliers can easily destroy cluster structure while the outliers are defined based on the cluster concept. Moreover, we aim to involve the information from source data to facilitate the cluster analysis on target data, where the labels from information source data are employed to guide the clustering as the partition level constraint. In light of this, we formulate the problem as the joint semi-supervised clustering and outlier detection with the following objective function,

$$\min_{H_s, H_t', C, O} \left|\left| \begin{bmatrix} X_s \\ X_t' \end{bmatrix} - \begin{bmatrix} H_s \\ H_t' \end{bmatrix} C \right|\right|_{\mathrm{F}}^2 - \lambda U_c(H_s, Y_s), \qquad (1)$$

where $X_t' = X_t \backslash O$ and O denote the inlier and outlier data points in the target data set, respectively, H_s and H_t' are the label indicator for X_s and X_t' with the centroid matrix $C \in \mathcal{R}^{K \times d}$, U_c is the widely known categorical utility function [28], which measures the similarity among partitions with high value indicating high partition-level similarity, and λ is the trade-off parameter.

In Eq. (1), the objective function contains two parts. The first term is a standard K-means clustering on the source and target data. It is worthy to note that the detected outliers are not assigned with the pre-defined class labels, neither does it get involved into the clustering process. The second term employs U_c to treat the similarity of two partitions. In the context of constrained clustering and consensus clustering, similarity calculation of two partitions widely adopts the categorical utility function. For example, constrained clustering at partition level applies the categorical utility function to compute the similarity of source and target data partitions using the source partial labels as the constraint. As a result, the learned target data partitions is close to the partial labels in the source data [29], [30]; consensus clustering applies the categorical utility function to calculate the similarity between basic partitions and consensus partition, and maximize the similarity by fusing several basic partitions into an integrated one. [31]–[34]. We introduce the contingency table in Table 1. The categorical utility function U_c can be calculated as follows:

TABLE 1. Contingency matrix.

		Y_s				
		$C_1^{(s)}$	$C_2^{(s)}$	$\cdots$	$C_K^{(s)}$	$\sum$
H_s	C_1	n_{11}	n_{12}	$\cdots$	n_{1K}	n_{1+}
	C_2	n_{21}	n_{22}	$\cdots$	n_{2K_i}	n_{2+}
	$\cdot$	$\cdot$	$\cdot$	$\cdots$	$\cdot$	$\cdot$
	C_K	n_{K1}	n_{K2}	$\cdots$	n_{KK}	n_{K+}
	$\sum$	n_{+1}	n_{+2}	$\cdots$	n_{+K}	n_s

$$U_c(H_s, Y_s) = \sum_{k=1}^{K} p_{k+} \sum_{j=1}^{K} (\frac{p_{kj}}{p_{k+}})^2 - \sum_{j=1}^{K} (p_{+j})^2, \qquad (2)$$

where H_S and Y_S are two partitions, p_{kj} denotes the probability of one instance belonging to the k-th cluster in H_S and the j-th cluster in Y_S at the same time, and p_{k+} and p_{+j} are the k-th cluster's portion in H_S and j-th cluster's portion in Y_S, respectively. The categorical utility function constraint measures the partition level similarity between H_S and Y_S, by calculating the difference between predicting H_S with Y_S and without Y_S.

The benefits of our proposed model lie in the following two aspects: (1) The joint clustering and outlier detection framework is employed, where high quality cluster structure contributes to the outlier detection, and the outlier detection also alleviates the negative impact on the cluster structure. (2) To fully make use of the auxiliary information source data, categorical utility function preserves the source data cluster structure as the partition level information, and further guides the target data clustering. By this objective function, we formulate the problem as the joint semi-supervised clustering and outlier detection. In the next section, we propose an efficient solution to solve this problem with convergence guarantee.

IV. SOLUTION

For the problem in Eq. (1), we face the following challenges. (1) The same variable lies in different granularities, where H_s is a matrix formulation in the first term, and the joint probability p_{kj} is employed in an element-wise formulation to calculate the similarity between H_s and Y_s. (2) Several unknown variables require to be optimized. Thus, a unified procedure with convergence guarantee is highly needed. (3) Some data points should be identified as the outliers, which should not be involved in the clustering process.

To solve the first challenge, we explore the categorical utility function and provide a new insight of the second term in Eq. (1).

Lemma 1: Given one fixed partition Y_s and any partition H_s, we have

$$U_c(H_s, Y_s) \propto -||Y_s - H_s G||_{\mathrm{F}}, \qquad (3)$$

where $G_k = (\frac{p_{k1}}{p_{k+}}, \cdots, \frac{p_{kj}}{p_{k+}}, \cdots, \frac{p_{kK}}{p_{k+}})$ is the k-th row of $G, \forall k, k = 1, \cdots, K$.

Given a fixed $Y_s = \{y_l\}_{1 \leq l \leq n_s}$ and any partition H, which contains the same number of instances with Y_s, we have g_k to denote the k-th centroid $\mathcal{C}_k$ according to H as follows.

$$g_k = (g_{k,1}, g_{k,2}, \cdots, g_{k,K}). \tag{4}$$

Recall the definitions in Table 1, we have

$$g_{k,j} = \frac{\sum_{y_l \in \mathcal{C}_k} y_l}{|\mathcal{C}_k|} = \frac{n_{kj}}{n_{k+}} = \frac{p_{kj}}{p_{k+}}, 1 \leq j \leq K. \tag{5}$$

According to the objective function of K-means, we have

$$\begin{aligned}
&||Y_s - H_s G||_{\mathrm{F}}^2 \\
&\quad = \sum_{k=1}^{K} \sum_{y_l \in \mathcal{C}_k} ||y_l - g_k||_2^2 \\
&\quad = \sum_{k=1}^{K} \sum_{y_l \in \mathcal{C}_k} (\langle y_l, y_l \rangle - 2\langle y_l, g_k \rangle + \langle g_k, g_k \rangle) \\
&\quad = \sum_{l=1}^{n_s} \langle y_l, y_l \rangle - \sum_{k=1}^{K} (2 \sum_{y_l \in \mathcal{C}_k} \langle y_l, g_k \rangle - \sum_{y_l \in \mathcal{C}_k} \langle g_k, g_k \rangle) \\
&\quad = \sum_{l=1}^{n_s} \langle y_l, y_l \rangle - \sum_{k=1}^{K} |\mathcal{C}_k| \langle g_k, g_k \rangle. \qquad (6)
\end{aligned}$$

Here $\langle \cdot \rangle$ means the inner product of two vectors. The above equation holds due to the definition of the centroid. With Eq. (5) and categorical utility function, we have

$$\begin{aligned}
&||Y_s - H_s G||_{\mathrm{F}}^2 \\
&\quad = \sum_{l=1}^{n_s} \langle y_l, y_l \rangle - n_s \sum_{k=1}^{K} p_{k+} \langle g_k, g_k \rangle \\
&\quad = \sum_{l=1}^{n_s} \langle y_l, y_l \rangle - n_s \sum_{k=1}^{K} p_{k+} \sum_{j=1}^{K} (\frac{p_{kj}}{p_{k+}})^2 \\
&\quad = \underbrace{\sum_{l=1}^{n_s} \langle y_l, y_l \rangle}_{(\alpha)} - n_s U_c(H_s, Y_s) - \underbrace{\sum_{j=1}^{K} (p_{+j})^2}_{(\beta)}. \qquad (7)
\end{aligned}$$

Since (α) and (β) are constants, we complete the proof.

We can see that $||Y_s - H_s G||_{\mathrm{F}}$ directly calculates the difference of two partitions with alignment through matrix G. According to Lemma 1, the categorical utility function U_c calculates how similar the Y_s and H_s partitions are. Then the problem in Eq. (1) can be reformulated as follows:

$$\min_{H_s, H_t', C, O, M} \left|\left| \begin{bmatrix} X_s \\ X_t' \end{bmatrix} - \begin{bmatrix} H_s \\ H_t' \end{bmatrix} C \right|\right|_{\mathrm{F}}^2 + \lambda ||Y_s - H_s G||_{\mathrm{F}} \tag{8}$$

In Eq. (8), it is important to get the two partitions aligned on the order of cluster labels, in this case with the help of alignment matrix G. Without alignment, even if the two partitions have the same structure, their similarity could be minimum if the order of the cluster labels are different.

The cluster labels of either data set are un-ordered. The distance between two exactly the same partitions will not be zero under this circumstanec. It is crucial to first align these two partitions before calculating the partition distance. With variable G introduced, we unify H_s in the two terms with matrix formulation.

If we take a close look at the second term in Eq. (8), it is exactly a standard K-means objective function with Y_s, H_s and M as the input data, indicator and centroid matrix, respectively. Moreover, if there is no outliers in the target data, the first term is also a standard K-means objective function as well. There two points motivate us for a K-means-like solution with an augmented matrix.

Before introducing the K-means-like optimization, We give the augmented matrix D as follows:

$$D = \begin{pmatrix} X_s & Y_s \\ X_t & 0 \end{pmatrix}, \tag{9}$$

where $d_i = (d_i^{(1)}, d_i^{(2)})$ is the i-th row of D, which consists of two parts and represents the original features and labels, respectively. Since the target data does not have label information, zeros are used to fill up for the augmented matrix.

Through these formulations, the problem is defined as a semi-supervised clustering problem with target label as missing information. Recall that in the standard K-means, arithmetic means decide the centroids, where the denominator represents the number of instances in its corresponding cluster. Directly applying the K-means method on the augmented matrix D will bring in interference to the final cluster structure. This is because the zero values in augmented matrix D are artificially added as placeholders and should not contribute to computing the centroids. Now we introduce the new update rule for centroids. Let $m_k = (m_k^{(1)}, m_k^{(2)})$ be the k-th centroid M on the augmented matrix, we have the following,

$$m_k^{(1)} = \frac{\sum_{x_i \in \mathbf{C}_k} d_i^{(1)}}{|\mathbf{C}_k|}, \; m_k^{(2)} = \frac{\sum_{x_i \in \mathbf{C}_k \cap Y_s} d_i^{(2)}}{|\mathbf{C}_k \cap Y_s|}. \tag{10}$$

Here the computation only takes "real" instances where there is a label. We now have the following Theorem after the computation of centroides is modified.

Theorem 1: Given the data matrix X_s and X_t and the label information from the source domain Y_s, we build the augmented matrix D in Eq. (9) and have the following equivalence

$$\begin{aligned}
&\min_{H_s, H_t, C, G, O} \left|\left| \begin{bmatrix} X_s \\ X_t' \end{bmatrix} - \begin{bmatrix} H_s \\ H_t \end{bmatrix} C \right|\right|_{\mathrm{F}}^2 + \lambda ||Y_s - H_s G||_{\mathrm{F}} \\
&\Leftrightarrow \min_{\mathcal{C}_k} \sum_{k=1}^{K} \sum_{d_i \in \mathcal{C}_k} f(d_i, m_k) \qquad (11)
\end{aligned}$$

where the distance function f can be computed by

$$f(d_i, m_k) = ||d_i^{(1)} - m_k^{(1)}||_2^2 + \mathbf{1}(d_i \in X_s) \lambda ||d_i^{(2)} - m_k^{(2)}||_2^2 \tag{12}$$

where $\mathbf{1}(\cdot)$ returns 1 when the condition is true, and returns 0 otherwise.

Given the objective function of K-means and the modified distance function, we have,

$$
\begin{aligned}
&\sum_{k=1}^{K}\sum_{d_i\in\mathcal{C}_k} f(d_i, m_k)\\
&= \sum_{k=1}^{K}\sum_{d_i\in\mathcal{C}_k} ||d_i^{(1)} - m_k^{(1)}||^2 + (\mathbf{1}(d_i \in X_s)||d_i^{(2)} - m_k^{(2)}||^2)\\
&= \underbrace{\sum_{k=1}^{K}\sum_{d_i\in\mathcal{C}_k} ||d_i^{(1)} - m_k^{(1)}||^2}_{(\alpha)} + \underbrace{\sum_{k=1}^{K}\sum_{d_i\in\mathcal{C}_k\cap X_s} ||d_i^{(2)} - m_k^{(2)}||^2}_{(\beta)}\\
&= \left|\left|\begin{bmatrix} X_s \\ X_t' \end{bmatrix} - \begin{bmatrix} H_s \\ H_t \end{bmatrix} C\right|\right|_{\mathrm{F}}^2 + \lambda||Y_s - H_sG||_{\mathrm{F}}. \qquad (13)
\end{aligned}
$$

We complete the proof.

Theorem 1 provides a solution to the problem in Eq. (11) through K-means-like optimization. Such optimization problems are well formulated and can be solved efficiently. More specifically, the label columns can be thought of features where λ is the weights of the features. Two-phase iterative optimization can still be used after computation for distance function and the update rules for centroids have changed

By Theorem 1, the problem in Eq. (11) is transformed into a K-means-like clustering problem. H_s and H_t are put together as a concatenated variable for optimization; similarity, C and G build up the new centroid M. By this means, only two variables, the indicator matrix and corresponding centroid matrix are iteratively updated within a neat mathematical and efficient way.

However, Theorem 1 conducts on all the target data points, where the outliers should be detected during the clustering process. In this paper, we consider the joint clustering and outlier detection, which simultaneously partitions the data and discovers outliers. That means the outlier detection and clustering are conducted in a unified framework. Since the centroids in K-means clustering are vulnerable to outliers, these outliers should not contribute to the centroids. Inspired by K-means-- [16], the outliers are identified as the points with large distance to the nearest centroid.

Thanks to Theorem 1, we formulate the problem in Eq. (11) with inliers into K-means framework so that the second challenge can fortunately solved by K-means--, where we calculate the distance between each point and its corresponding nearest centroid, and label l points in the target data as outliers with the largest distance. In light of this, we summarize the complete process of our proposed method in Algorithm 1. The complex outlier detection with knowledge reuse problem in Eq. (11) can be exactly solved by a modified K-means-- algorithm on the augmented matrix D. It is worthy to note that in the traditional K-means--, any data point might be an outlier; however, only the target data points might be ouliters. After delicate transformation and derivation, K-means-- is used as a tool to solve the problem in Eq. (8), which returns K clusters $C_1, \cdots, C_K$ and outlier set O.

Since we solve the problem in Eq. (8) via a modified K-means--, our solution also inherits the similar time complexity with K-means--. The augmented matrix D has n rows and $d + K$ dimensions, where $n = n_s + n_t$. During the data assignment in Line 4, we calculate the distance between each data point and K centroids, which requires $\mathcal{O}(nK(m + K))$; in Line 5, a quick sorting is employed to find the outliers with the largest nearest distances and takes $\mathcal{O}(n)$; for centroid updating, $n - l$ inliers contribute the centroids, and it takes $\mathcal{O}(n - l)$. Let t denotes the number of iterations, the total time complexity of our solution takes $\mathcal{O}(tnK(m+K))$, which is roughly linear to the number of data points and suitable to handle large scale clustering and outlier detection.

Finally, we provide the convergence property of our solution by the following Theorem 2.

Theorem 2: The objective function value would continuously decrease by Algorithm 1, which converges to a local optimum.

Algorithm 1 Knowledge Reused Outlier Detection.

```
Input: $X_s, Xt$: source and target data matrix; $Y_s$: labels of
    source data; $K$: number of clusters;
        $l$: number of outliers; $\lambda$: trade-off parameter.
Output: $K$ clusters $C_1, \cdots, C_K$ and $l$ outliers.
1: Build the augmented matrix $D$ by Eq. (9);
2: Initialize K centroids from $D$;
3: repeat
4:     Calculate the distance between each point in $D$ and its
       nearest centroid with the distance function in Eq. (12);
5:     For the target data, identify $l$ points with largest distances as outliers, and assign the label -1;
6:     Assign the rest $n_s + n_t - l$ points to their nearest
       centroids with label $1, 2, \cdots, K$;
7:     Update the centroids by Eq. (10);
8: until The objective value in Eq. (8) remains unchanged.
```

Proof: It has been widely recognized that the classical K-means consists of two iterative steps, assigning data points to their nearest centroids and updating the centroids, which is guaranteed to converge to a local optimum. Our solution by Algorithm 1 has two major differences with the classical K-means, incomplete label assignment in Line 5 and non-exhausted centroid updating by Eq. (10). In the following, we proof that Algorithm 1 still converges with these two modifications.

In the classical K-means, the objective function value with n data points would decrease. For the incomplete label assignment, $n - l$ inliers contribute the objective function value, which lead to the decrease of the objective function value as well. For the non-exhausted centroid updating, we focus on the second part $m_k^{(2)}$ in Eq. (10). Let $y_k^{(2)}$ be any second part centroid with $y_k^{(2)} \neq m_k^{(2)}$, we have the difference of the partial

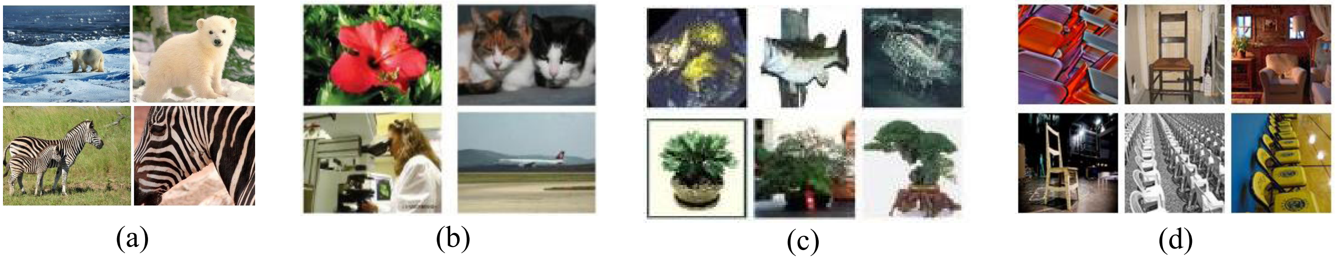


FIGURE 1. Some image samples from (a) *AWA*, (b) *Bing*, (c) *Caltech101* and (d) *ImageNet*.

objective function value with $m_k^{(2)}$ and $y_k^{(2)}$.

$$\begin{aligned}\Delta &= \sum_{d_i\in\mathcal{C}_k\cap Y_s} ||d_i^{(2)} - y_k^{(2)}||_2^2 - \sum_{d_i\in\mathcal{C}_k\cap Y_s} ||d_i^{(2)} - m_k^{(2)}||_2^2 \\ &= -2\sum_{d_i\in\mathcal{C}_k\cap Y_s} d_i^{(2)}y_k^{(2)} + |\mathcal{C}_k\cap Y_s|(y_k^{(2)})^2 \\ &\quad + 2\sum_{d_i\in\mathcal{C}_k\cap Y_s} d_i^{(2)}m_k^{(2)} - |\mathcal{C}_k\cap Y_s|(m_k^{(2)})^2 \\ &= |\mathcal{C}_k\cap Y_s|||y_k^{(2)} - m_k^{(2)}||_2^2 > 0\end{aligned}$$

The above equation holds due to the centroid updating $\sum_{d_i\in\mathcal{C}_k\cap Y_s} d_i^{(2)} = |\mathcal{C}_k \cap Y_s|m_k^{(2)}$ in Eq. (10). Therefore, the non-exhausted centroid updating is optimal to decrease the objective function value.

In summary, the incomplete label assignment and non-exhausted centroid updating also decrease the objective function value during the iterations. Since the clustering and outlier detection solution space is finite, Algorithm 1 converges to a local optimum. We finish the proof.

V. EXPERIMENTAL RESULTS

In this section, we introduce the experiments performed to showcase the effectiveness of our proposed method. We first introduce the experimental settings and the four real world data sets we used. We then compare our method with eight state-of-the-art outlier detection methods. The cluster validity is also explored for evaluating the quality of the clustering results. In the end, parameter analysis is provided as practical usage guidance.

A. DATA AND SETTINGS

1) DATA SETS

Four large-scale data sets are employed to evaluate our proposed algorithm. We choose these data sets because the data has various number of categories ranging from 50 to 20,000, and reasonably large number of records to demonstrate the performance and scalability of the proposed method. Figure 1 shows some samples from these four data sets.

- *Animals with Attributes (AwA)* [35] is a database of 50 animals categories, containing a total of 30475 images. Each class has 85 general attributes shared among different classes. The animals could appear in different scales and poses in the images.
- *Caltech101*[1] is a widely used database for object recognition which contains a total of 9,144 images from 100 object classes plus one background class. In total there are 101 classes. The object classes are animals, vehicles, trees, etc. Each category contains 40 to 800 images.
- *Bing* [36] is a collection of Internet images. It has the same set of 256 object categories as the dataset of Caltech256 [37]. Queries to Bing with class names at the time return these images. For each query, several noisy images that don't belong to this category are acquired but not removed. This produces a weakly labeled collection. The Bing sets are both semantically and visually less coherent than the Caltech256 [37]. The number of samples per class ranges from 197 to 593.
- *Imagenet* [2] contains around 21000 object classes and over five hundred images per class on average. The classes are organized based on the WordNet hierarchy, in which each node of the hierarchy is depicted by up to thousands of images.

To simulate the knowledge reuse scenario we address in this paper, 50% instances from the first 40, 200, 80 and 80 classes from these four data sets, respectively are treated as the information source data, while the rest 50% data from these classes and some random selected data beyond these classes build up the target data. We use deep neural network to extract the visual features (DeCaf [38] and VGG [39]) from these images for joint clustering and outlier detection. Table 2 shows key metrics of these data sets. CV denotes the coefficient of variation.

2) COMPETITIVE METHODS

Several classical or state-of-the-art outlier detection methods including density-based LOF [1], COF [2], distance-based LODF [3], angle-based FABOD [4], ensemble-based iForest [5], eigenvector-based OPCA [6], cluster-based TONMF [7] are involved as the competitive methods to evaluate the outlier detection performance.[3]

l points with the largest scores by these methods are

[1] http://www.vision.caltech.edu/Image_Datasets/Caltech101/

[2] http://www.image-net.org/

[3] The codes of outlier detection methods can be found at https://github.com/dsmi-lab-ntust/AnomalyDetectionToolbox and https://github.com/ramkikannan/outliernmf.

TABLE 2. Some key characteristics of these four data sets.

Data set	Feature type	#feature	#class	#instance of source	#instance of target	CV	#outlier	outlier raio
AWA	vgg	4096	40	12678	13659	0.4264	900	0.0659
Bing	decaf	1000	200	47109	49860	0.0895	2850	0.0572
Caltech101	decaf	4096	80	3690	4065	1.4346	420	0.1033
ImageNet	decaf	1000	80	56208	59969	0.2725	3800	0.0634

TABLE 3. Performance of outlier detection and cluster validity by different algorithms (%).

Data set	Outlier detection									Cluster validity	
	Jaccard									Accuracy	
	LOF	COF	LDOF	FABOD	iForest	OPCA	TONMF	K-means--	Ours	K-means--	Ours
AWA	2.51	4.93	4.11	5.15	4.60	1.63	4.38	8.12	**16.33**	63.11	**71.42**
Bing	2.67	1.73	1.77	0.00	2.78	2.08	0.00	3.73	**5.43**	12.26	**17.22**
Caltech101	11.26	8.95	5.53	7.55	11.70	0.72	6.46	12.44	**17.56**	36.92	**40.58**
ImageNet	4.50	4.78	5.03	3.97	4.11	0.21	0.00	9.58	**15.11**	29.91	**43.59**
	F-measure									NMI	
AWA	4.90	9.40	7.90	9.80	8.80	3.20	8.40	14.96	**28.09**	71.69	**74.72**
Bing	5.19	3.40	3.47	0.00	5.40	4.07	0.00	7.21	**10.31**	22.44	**26.03**
Caltech101	20.24	16.43	10.48	14.05	20.95	1.43	12.14	22.09	**29.87**	64.14	**65.49**
ImageNet	8.61	9.13	9.58	7.63	7.89	0.42	0.00	17.45	**26.26**	45.73	**55.64**

regarded as outliers. For the outlier detection methods, some default settings in the original papers are used for stable results. The number of nearest neighbors in LOF, COF, LODF and FABOD is set to 50 as the authors recommend; the subsampling size and the number of trees in iForest are 200 and 100; the forgetting number is set to 0.1 in OPCA; the rank and two parameters in TONMF are 10, 10 and 0.1, respectively. Moreover, K-means-- [16] is also used for comparisons in terms of jointly clustering and outlier detection. For each algorithm, we feed the true cluster and outlier numbers for fair comparisons. For K-means-- and our method, we run 20 times and report the average performance. Since their standard deviations across results are less than 0.02, we omit the details here. In our method, we set λ to be 10 as the default setting.

3) VALIDATION METRICS

Since the ground truth labels are available for these four data sets, four external metrics are employed for evaluation in terms of outlier detection and cluster validity, where the outlier set is regarded as a special cluster in the ground truth.

Jaccard index and F-measure are designed for the binary classification, which are employed to evaluate the outlier detection. They can be computed as follows.

$$Jaccard = \frac{|O \cap O^*|}{|O \cup O^*|},\ \ F\!-\!measure = 2 * \frac{\text{precition} \cdot \text{recall}}{\text{precition}+\text{recall}},$$

where O and O^* are the outlier sets by the algorithm and ground truth, respectively, and F-measure is the harmonic average of the precision and recall for outlier class.

Accuracy and Normalized Mutual Information (NMI) are two widely used external measurements for cluster validity [40]. NMI measures normalized the mutual information between ground truth labels and resulted cluster labels, followed by a normalization operation, while accuracy comes from classification with the best mapping. They can be computed as follows.

$$accuracy = \sum_{i=1}^{n} \delta(s_i.map(r_i))/n,$$

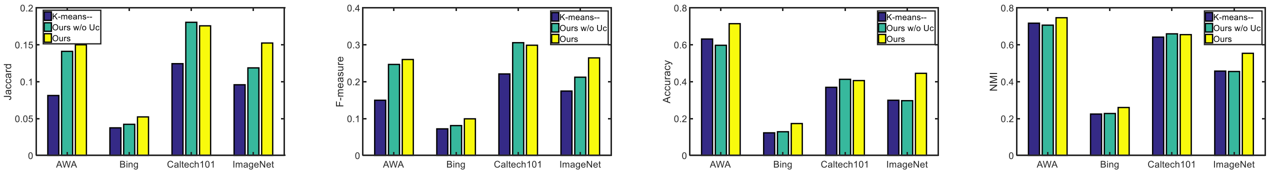
$$NMI = \frac{\sum_{i,j} n_{ij} \log \frac{n \cdot n_{ij}}{n_{i+} \cdot n_{+j}}}{\sqrt{(\sum_i n_{i+} \log \frac{n_{i+}}{n})(\sum_j n_{j+} \log \frac{n_{+j}}{n})}},$$

where $map(r_i)$ is the permutation mapping function that maps each ground truth s_i to cluster label r_i, and $\delta(x, y)$ is the Kronecker delta function that returns one when $x = y$ and zeros if otherwise. The best mapping is applied by the KuhnMunkres algorithms. And the variables in NMI can be found in Table 1. Note that these four metrics are positive measurements, i.e, a larger value means better performance.

B. PERFORMANCE ANALYSIS ON VALIDATION METRICS

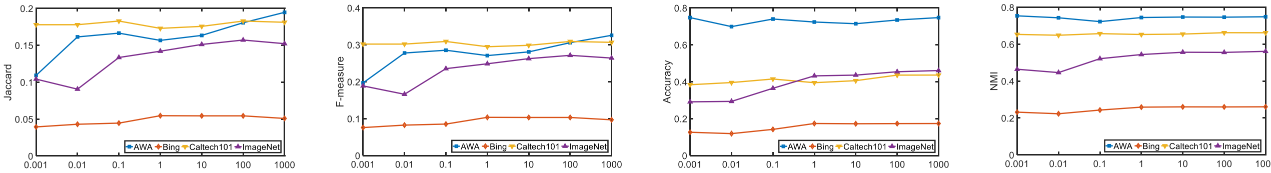
Table 3 shows the performance of outlier detection and cluster validity in terms of the four validation metrics. These competitive outlier detection algorithms are based on different assumptions including density, distance, angle, ensemble, eigenvector and clusters, and effective only on certain data sets. LOF and iForest get the reasonable results on *Caltech101*. However in other cases, the two methods show the obvious disadvantages in terms of performance. The reasons are complicated, but single outlier detection and unsupervised parameter setting might the major reasons. For TONMF, there are three settings as the inputs, which are difficult to set without any knowledge from domain experts. Different from single outlier detection, K-means-- aims to jointly cluster the data with outlier removal, where these detected outliers do not contribute the centroids, and instead high quality cluster structure is conductive to the outlier detection as well. Therefore, K-means-- outperforms the single outlier detection methods.

Compared with K-means--, our method makes full use of the information source data, which employs the knowledge reuse to improve the outlier detection performance.



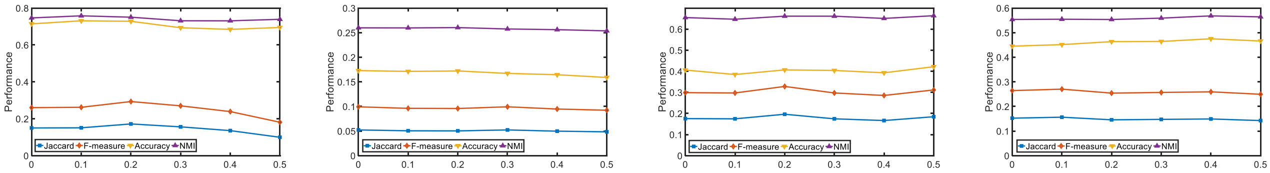
(a) (b) (c) (d)

FIGURE 2. Performance comparison by K-means--, our model without the U_c term and our model in terms of (a) Jaccard, (b) F-measure, (c) accuracy and (d) NMI.



(a) (b) (c) (d)

FIGURE 3. Parameter analysis on λ of our method in terms of (a) Jaccard, (b) F-measure, (c) Accuracy and (d) NMI. The x-axis denotes λ values from 0.001 to 1000.



(a) (b) (c) (d)

FIGURE 4. Performance of our method with noisy source labels. The x-axis denotes the noisy label percentage. (a) AWA. (b) Bing. (c) Caltech101. (d) ImageNet.

Moreover, our method shows significant improvements over K-means-- in terms of cluster validity. For example, our method exceeds K-means-- by 13%, 10% on *AWA* and *ImageNet* by F-measure and accuracy, respectively. The source and target data are put together for clustering with the source data cluster structure preserved. Categorical utility function, widely used in consensus clustering, is employed as a regularizer in our model to make the learnt source label consistent with the ground truth as much as possible, and further guide the clustering process. To verify this point, we set *lambda* to be zero, and implement the model with only the first term in Eq. (1). Figure 2 shows the performance by K-means--, our model with and without the U_c regularizer in terms of Jaccard, F-measure, accuracy and NMI. Generally speaking, with extra source data involved, our method without U_c gains consistent improvements over K-means-- in terms of all four metrics. The information source data make the centroids more stable, which benefits the outlier detection and cluster analysis. Furthermore, the second term in Eq. (1) plays a role in preserving the source structure and enhancing the whole structure as a regularizer.

C. PARAMETER ANALYSIS WITH λ

We provide the parameter analysis of our model for practical use. There is only one parameter λ to balance the cluster-outlier task and the structure-preserved regularizer. Figure 3 shows the performance of our model with different λ varying from 0.001 to 1000. We can see that with the increase of λ, the performance goes up in terms of four metrics. The phenomenon is significant on *ImageNet*. This indicates the large λ makes the centroids of the whole data set fixed to the ones of the source data, hence more prominent performance gains have been acquired on the data set with the largest number of labels.

D. RESISTANCE TO NOISE

Since the source data have the label information, the centroids of source data play a role as the supervision to guide the clustering process. This makes sense if the source labels are correct. However in the real-world applications, the labels might be noisy. We continue to explore our model to handle noisy source label. Figure 4 shows the performance of our method with different ratios of noisy labels. To simulate the noisy labels, we randomly select a certain number of data points and randomly assign the labels. We can see that our model is robust to noisy source labels. Although the performance decreases a little with the increase of the noise ratio, our model still delivers satisfactory results with the noise level up to 50%.

VI. CONCLUSION

In this paper, we considered the outlier detection via knowledge reuse problem. Generally speaking, an information source data was employed to facilitate the outlier detection task on the target data. To fully make use of the source knowledge, the source data and target data are put together for joint clustering and outlier detection using the source data cluster structure as constrain By this means, categorical utility function was used so that the target data partitions are consistent with the source data partition structure. With an augmented matrix, the problem was solved by a modified K-means-- with rigid mathematical formulation and theoretical convergence guarantee. We have used four real world data sets and eight outlier detection methods of different kinds for extensive experiments and comparison. The results show the effectiveness of our method in terms of outlier detection and cluster validity metrics over eight outlier detection methods of different kinds. The parameter analysis on the algorithm was provided for practical use. Exploration on noisy source label also proves that the method can handle practical problems where the labeling might be low quality.

REFERENCES

[1] M. M. Breunig, H.-P. Kriegel, R. T. Ng, and J. Sander, "LOF: Identifying density-based local outliers," in *Proc. SIGMOD*, 2000, pp. 93–104.

[2] J. Tang, Z. Chen, A. W.-C. Fu, and D. W. Cheung, "Enhancing effectiveness of outlier detections for low density patterns," in *Proc. PAKDD*, 2002, pp. 535–548.

[3] K. Zhang, M. Hutter, and H. Jin, "A new local distance-based outlier detection approach for scattered real-world data," in *Proc. PAKDD*, 2009, pp. 813–822.

[4] N. Pham and R. Pagh, "A near-linear time approximation algorithm for angle-based outlier detection in high-dimensional data," in *Proc. KDD*, 2012, pp. 877–885.

[5] F. T. Liu, K. M. Ting, and Z.-H. Zhou, "Isolation forest," in *Proc. ICDM*, 2008, pp. 413–422.

[6] Y. J. Lee, Y. R. Yeh, and Y. C. F. Wang, "Anomaly detection via online oversampling principal component analysis," *IEEE Trans. Knowl. Data Eng.*, vol. 25, no. 7, pp. 1460–1470, Jul. 2013.

[7] R. Kannan, H. Woo, C. Aggarwal, and H. Park, "Outlier detection for text data," in *Proc. SDM*, 2017, pp. 489–497.

[8] M. Gupta, J. Gao, C. C. Aggarwal, and J. Han, "Outlier detection for temporal data: A survey," *IEEE Trans. Knowl. Data Eng.*, vol. 26, no. 9, pp. 2250–2267, Sep. 2014.

[9] S. Agrawal and J. Agrawal, "Survey on anomaly detection using data mining techniques," *Procedia Comput. Sci.*, vol. 60, pp. 708–713, 2015.

[10] A. Georgogiannis, "Robust k-means: A theoretical revisit," in *Proc. NIPS*, 2016, pp. 2891–2899.

[11] Z. He, X. Xu, J. Z. Huang, and S. Deng, "FP-outlier: Frequent pattern based outlier detection," *Comput. Sci. Inf. Syst.*, vol. 2, no. 1, pp. 103–118, 2005.

[12] H.-P. Kriegel, M. Schubert, and A. Zimek, "Angle-based outlier detection in high-dimensional data," in *Proc. KDD*, 2008, pp. 444–452.

[13] H. Liu, Y. Zhang, B. Deng, and Y. Fu, "Outlier detection via sampling ensemble," in *Proc. BigData*, 2016, pp. 726–735.

[14] M. Charikar, S. Khuller, D. M. Mount, and G. Narasimhan, "Algorithms for facility location problems with outliers," in *Proc. SODA*, 2001, pp. 642–651.

[15] K. Chen, "A constant factor approximation algorithm for k-median clustering with outliers," in *Proc. SODA*, 2008, pp. 826–835.

[16] S. Chawla and A. Gionis, "*k*-means: A unified approach to clustering and outlier detection," in *Proc. SDM*, 2013, pp. 189–197.

[17] L. Ott, L. Pang, F. T. Ramos, and S. Chawla, "On integrated clustering and outlier detection," in *Proc. NIPS*, 2014, pp. 1359–1367.

[18] L. Zhao, S. J. Pan, E. W. Xiang, E. Zhong, Z. Lu, and Q. Yang, "Active transfer learning for cross-system recommendation," in *Proc. AAAI*, 2013, pp. 1205–1211.

[19] F. Zhuang, P. Luo, H. Xiong, Q. He, Y. Xiong, and Z. Shi, "Exploiting associations between word clusters and document classes for cross-domain text categorization," *Stat. Anal. Data Mining, ASA Data Sci. J.*, vol. 4, no. 1, pp. 100–114, 2011.

[20] Z. Ding, M. Shao, and Y. Fu, "Deep low-rank coding for transfer learning," in *Proc. IJCAI*, 2015, pp. 3453–3459.

[21] F. Li, S. J. Pan, O. Jin, Q. Yang, and X. Zhu, "Cross-domain co-extraction of sentiment and topic lexicons," in *Proc. ACL*, 2012, pp. 410–419.

[22] S. Shekhar, V. M. Patel, H. V. Nguyen, and R. Chellappa, "Generalized domain-adaptive dictionaries," in *Proc. CVPR*, 2013, pp. 361–368.

[23] J. Hu, J. Lu, and Y. Tan, "Deep transfer metric learning," in *Proc. CVPR*, 2015, pp. 325–333.

[24] L. Bruzzone and M. Marconcini, "Domain adaptation problems: A DASVM classification technique and a circular validation strategy," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 5, pp. 770–787, May 2010.

[25] L. Duan, I. W. Tsang, and D. Xu, "Domain transfer multiple kernel learning," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 3, pp. 465–479, Mar. 2012.

[26] J. Ni, Q. Qiu, and R. Chellappa, "Subspace interpolation via dictionary learning for unsupervised domain adaptation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2013, pp. 692–699.

[27] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Trans. Knowl. Data Eng.*, vol. 22, no. 10, pp. 1345–1359, Oct. 2010.

[28] B. Mirkin, "Reinterpreting the category utility function," *Mach. Learn.*, vol. 45, no. 2, pp. 219–228, Nov. 2001.

[29] H. Liu and Y. Fu, "Clustering with partition level side information," in *Proc. ICDM*, 2015, pp. 877–882.

[30] H. Liu, Z. Tao, and Y. Fu, "Partition level constrained clustering," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 10, pp. 2469–2483, Oct. 2017.

[31] J. Wu, H. Liu, H. Xiong, and J. Cao, "A theoretic framework of k-means-based consensus clustering," in *Proc. Int. Joint Conf. Artif. Intell.*, 2013, pp. 1799–1805.

[32] J. Wu, H. Liu, H. Xiong, J. Cao, and J. Chen, "K-means-based consensus clustering: A unified view," *IEEE Trans. Knowl. Data Eng.*, vol. 27, no. 1, pp. 155–169, Jan. 2015.

[33] H. Liu, T. Liu, J. Wu, D. Tao, and Y. Fu, "Spectral ensemble clustering," in *Proc. 21st ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2015, pp. 715–724.

[34] H. Liu, J. Wu, T. Liu, D. Tao, and Y. Fu, "Spectral ensemble clustering via weighted k-means: Theoretical and practical evidence," *IEEE Trans. Knowl. Data Eng.*, vol. 29, no. 5, pp. 1129–1143, May 2017.

[35] C. H. Lampert, H. Nickisch, and S. Harmeling, "Learning to detect unseen object classes by between-class attribute transfer," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2009, pp. 951–958.

[36] A. Bergamo and L. Torresani, "Exploiting weakly-labeled Web images to improve object classification: A domain adaptation approach," in *Proc. Adv. Neural Inf. Process. Syst.*, 2010, pp. 181–189.

[37] G. Griffin, A. Holub, and P. Perona, "Caltech-256 object category dataset," California Inst. Technol., Pasadena, CA, USA, Tech. Rep. CNS-TR-2007-001, 2007. [Online]. Available: https://authors.library.caltech.edu/7694/

[38] J. Donahue *et al.*, "DeCAF: A deep convolutional activation feature for generic visual recognition," in *Proc. Int. Conf. Mach. Learn.*, 2014, pp. 647–655.

[39] K. Simonyan and A. Zisserman. (2014). "Very deep convolutional networks for large-scale image recognition." [Online]. Available: https://arxiv.org/abs/1409.1556

[40] J. Wu, H. Xiong, and J. Chen, "Adapting the right measures for k-means clustering," in *Proc. KDD*, 2009, pp. 877–886.

WEIREN YU received the B.E. degree from the School of Advanced Engineering, Beihang University, China, in 2011, where he is currently pursuing the Ph.D. degree with the Department of Computer Science. He was a Visiting Researcher with the Machine Learning Department, CMU, from 2015 to 2016. His research interests include distributed machine learning systems, scalable graphical models, and graph mining models for emerging event detection on social media.

ZHENGMING DING (S'14–M'18) received the B.Eng. degree in information security and the M.Eng. degree in computer software and theory from the University of Electronic Science and Technology of China (UESTC), China, in 2010 and 2013, respectively, and the Ph.D. degree from the Department of Electrical and Computer Engineering, Northeastern University, USA, in 2018. He has been a Faculty Member with the Department of Computer, Information, and Technology, Indiana University-Purdue University Indianapolis, since 2018. His research interests include transfer learning, multi-view learning, and deep learning. He is a member of the IEEE. He received the National Institute of Justice Fellowship 2016–2018. He was a recipient of the Best Paper Award (SPIE 2016) and Best Paper Candidate (ACM MM 2017). He is currently an Associate Editor of the *Journal of Electronic Imaging* (JEI).

CHUNMING HU received the Ph.D. degree from Beihang University, in 2006. He was a Postdoctoral Research Fellow in the distributed system group with The Hong Kong University of Science and Technology, China, from 2006 to 2007. He is currently an Associate Professor with the School of Computer Science and Engineering, Beihang University, China. His current research interests include the distributed systems, system virtualization, large scale data management and processing systems, and data mining applications.

HONGFU LIU received the bachelor's and master's degree in management information systems from the School of Economics and Management, Beihang University, in 2011 and 2014, respectively, and the Ph.D. degree in computer engineering from Northeastern University, Boston, MA, USA, in 2018. He is currently a tenure-track Assistant Professor affiliated with the Michtom School of Computer Science, Brandeis University. His research interests include data mining and machine learning, with special interests in ensemble learning. He has served as a reviewer for many IEEE Transactions journals, including TKDE, TNNLS, TIP, and TBD. He has also served on the program committee for the conferences, including AAAI, IJCAI, and NIPS. He is an Associate Editor of the IEEE *Computational Intelligence Magazine*.