# Constrained Clustering With Imperfect Oracles

Xiatian Zhu, *Student Member, IEEE*, Chen Change Loy, *Member, IEEE*, and Shaogang Gong

***Abstract*— While clustering is usually an unsupervised operation, there are circumstances where we have access to prior belief that pairs of samples should (or should not) be assigned with the same cluster. Constrained clustering aims to exploit this prior belief as constraint (or weak supervision) to influence the cluster formation so as to obtain a data structure more closely resembling human perception. Two important issues remain open: 1) how to exploit sparse constraints effectively and 2) how to handle ill-conditioned/noisy constraints generated by imperfect oracles. In this paper, we present a novel pairwise similarity measure framework to address the above issues. Specifically, in contrast to existing constrained clustering approaches that blindly rely on all features for constraint propagation, our approach searches for neighborhoods driven by discriminative feature selection for more effective constraint diffusion. Crucially, we formulate a novel approach to handling the noisy constraint problem, which has been unrealistically ignored in the constrained clustering literature. Extensive comparative results show that our method is superior to the state-of-the-art constrained clustering approaches and can generally benefit existing pairwise similarity-based data clustering algorithms, such as spectral clustering and affinity propagation.**

***Index Terms*— Affinity propagation, constrained clustering, constraint propagation, feature selection, imperfect oracles, noisy constraints, similarity/distance measure, spectral clustering (SPClust).**

## I. Introduction

PAIRWISE similarity-based clustering algorithms, such as spectral clustering (SPClust) [1]–[4], or affinity propagation [5], search for coherent data clusters based on (dis)similarity relationship between data samples. In this paper, we consider the problem of pairwise similarity-based constrained clustering given constraints derived from human/oracles. The constraint is often available in a small quantity, and expressed in the form of pairwise link, namely, *must-link*—a pair of samples must be in the same cluster, and *cannot-link*—a pair of samples belong to different clusters. The objective is to exploit this small amount of supervision effectively to help revealing the semantic data partitions/groups that capture consistent concepts as perceived by human.

Constrained clustering has been extensively studied in the past and it remains an active research area [6]–[8]. Though great strides have been made in this field, two important and nontrivial questions remain open as detailed below.

Manuscript received October 3, 2013; revised December 12, 2014; accepted December 27, 2014.

X. Zhu and S. Gong are with the School of Electronic Engineering and Computer Science, Queen Mary University of London, London E1 4NS, U.K. (e-mail: xiatian.zhu@qmul.ac.uk; s.gong@qmul.ac.uk).

C. C. Loy is with the Department of Information Engineering, Chinese University of Hong Kong, Hong Kong (e-mail: ccloy@ie.cuhk.edu.hk).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TNNLS.2014.2387425

### A. Sparse Constraint Propagation

While constraints can be readily transformed into pairwise similarity measures, e.g., assign 1 to the similarity between two must-linked samples, and 0 to that between two cannot-linked samples [9], samples labeled with link preference are typically insufficient since exhaustive pairwise labeling is laborious. As a result, a limited number of constraints are usually employed together with data features to positively affect the similarity measures over unconstrained sample pairs so that the yielded similarities are closer to the intrinsic semantic structures. Such a similarity distortion/adaptation process is often known as *constraint propagation* [7], [8].

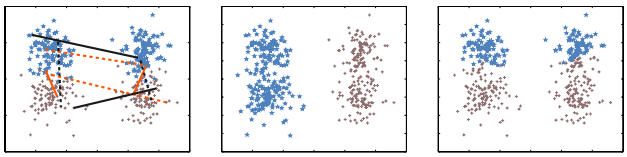
Effective constraint propagation relies on robust identification of unlabelled nearest neighbors (NNs) around the labeled samples in the feature space. Often, the NN search is susceptible to noisy or ambiguous features, especially so on image and video datasets. Trusting all the available features blindly for NN search (as what most existing constrained clustering approaches [6]–[8] did) is likely to result in suboptimal constraint diffusion. It is challenging to determine how to propagate their influence effectively to neighboring unlabelled points. In particular, it is nontrivial to reliably identify the neighboring unlabelled points for propagation.

### B. Noisy Constraints From Imperfect Oracles

Human annotators (oracles) may provide invalid/mistaken constraints. For instance, a portion of the must-links are actually cannot-links and vice versa. For example, annotations or constraints obtained from online crowdsourcing services, e.g., Amazon Mechanical Turk [10], are very likely to contain errors or noises due to data ambiguity, unintentional human mistakes, or even intentional errors by malicious workers [10], [11]. Learning such constraints blindly may result in sub-optimal cluster formation. Most existing methods make an unrealistic assumption that constraints are acquired from perfect oracles and thus they are noise-free. It is nontrivial to quantify and determine which constraints are noisy prior to clustering.

To address the above issues, we formulate a novel COnstraint Propagation Random Forest (COP-RF), capable of not only effectively propagating sparse pairwise constraints, but also dealing with noisy constraints produced by imperfect oracles. The COP-RF is flexible in that it generates an affinity matrix that encodes the constraint information for existing SPClust methods [1]–[4], or other pairwise similarity-based clustering algorithms for constrained clustering.

More precisely, the proposed model allows effective sparse constraint propagation through using the NN samples that are found in discriminative feature subspaces, rather than those



(a) Ground Truth (b) Spectral Clustering (c) Our Approach

Fig. 1. (a) Ground-truth cluster formation, with invalid pairwise constraints highlighted in light red; must- and cannot-links are represented by solid and dashed lines, respectively. (b) Clustering result obtained using unsupervised clustering. (c) Clustering result obtained using our method.

that found considering the whole feature space, which can be suboptimal due to noisy and ambiguous features. This is made possible by introducing a new objective/split function into COP-RF, which searches for discriminative features that induce the best data subspaces while simultaneously considering the model parameters that best satisfy the constraints imposed. To identify and filter noisy constraints generated from imperfect oracles, we introduce a novel constraint inconsistency quantification algorithm based on the outlier detection mechanism of random forest. Fig. 1 shows an example to illustrate how a COP-RF is capable of discovering data partitions close to the ground truth clusters despite that it is provided only with sparse and noisy constraints.

The sparse and noisy constraint issues are inextricably linked but no existing constrained clustering methods, to our knowledge, address them in a unified framework. This is the very first study that addresses them jointly. In particular, our work makes the following contributions.

1) We formulate a novel discriminative-feature driven approach for effective sparse constraint propagation. Existing methods fundamentally ignore the role of feature selection in this problem.
2) We propose a new method to cope with potentially noisy constraints based on constraint inconsistency measures, a problem that is largely unaddressed by existing constrained clustering algorithms.

We evaluate the effectiveness of the proposed approach by combining it with SPClust [1]. We demonstrate that the SPClust + COP-RF is superior when compared with the state-of-the-art constrained SPClust algorithms [8], [9] in exploiting sparse constraints generated by imperfect oracles. In addition to SPClust, we show the possibility of using the proposed approach to benefit affinity propagation [5] for effective constrained clustering.

## II. Related Work

A number of studies suggest that human similarity judgements are nonmetric [12]–[14]. Incorporating nonmetric pairwise similarity judgements into clustering has been an important research problem. There are generally two paradigms to exploit such judgements as constraints. The first paradigm is distance metric learning [15]–[19], which learns a distance metric that respects the constraints, and runs ordinary clustering algorithms, such as $k$-means, with distortion defined using the learned metric. The second paradigm is constrained clustering, which adapts existing clustering methods, such as $k$-means [6], [20] and SPClust methods [21], [22] to satisfy the given pairwise constraints. In this paper, we focus on constrained clustering approach. We now detail related work to this method.

### A. Sparse Constraint Propagation

Studies that perform constrained SPClust in general follow a procedure that first manipulates the data affinity matrix with constraints and then performs SPClust. For instance, Kamvar *et al.* [9] trivially adjust the elements in an affinity matrix with 1 and 0 to respect must-link and cannot-link constraints, respectively. No constraint propagation is considered in this method.

The problem of sparse constraint propagation is considered in [7], [8], [23], and [24]. Lu and Carreira-Perpinán [7] propose to perform propagation with a Gaussian process. This method is limited to the two-class problem, although a heuristic approach for multiclass problems is also discussed. Li *et al.* [24] formulate the propagation problem as a semidefinite programming (SDP) optimization problem. The method is not limited to the two-class problem, but solving the SDP problem involves an extremely large computational cost. In [23], the constraint propagation is also formulated as a constrained optimization problem, but only must-link constraints can be employed. In contrast to the above methods, the proposed approach is capable of performing effective constrained clustering using both available must-links and cannot-links, while it is not limited to two-class problems.

The state-of-the-art results are achieved by Lu and Ip [8]. They address the propagation problem through manifold diffusion [25]. The locality-preserving character in learning a manifold with dominant eigenvectors makes the solution less susceptible to noise to a certain extent, but the manifold construction still considers the full feature space, which may be corrupted by noisy features. We will show in Section IV that the manifold-based method is not as effective as the proposed discriminative-feature-driven constraint propagation. Importantly, the method proposed in [8], as well as those in [7], [23], [24], do not have a mechanism to handle noisy constraints.

### B. Handling Imperfect Oracles

Few constrained clustering studies consider imperfect oracles whereas most assume perfect constraints available. Coleman *et al.* [26] propose a constrained clustering algorithm capable of dealing with inconsistent constraints. This model is restricted only to the two-class problem due to the adoption of 2-correlation clustering idea. On the other hand, some strategies to measure constraint inconsistency and incoherence are discussed in [27] and [28]. Nevertheless, no concrete method is proposed to exploit such metrics for improved constrained clustering. Beyond constrained clustering, the problem of imperfect oracles has been explored in active learning [29]–[32] and online crowdsourcing [10], [33]. Our work differs significantly from these studies as we are interested in identifying noisy or inconsistent pairwise constraints rather than inaccurate class labels.

In comparison with our earlier version of this paper [34], in this paper, we provide more comprehensive explanations and justifications of the proposed approach, a new approach to filtering noisy constraints, along with more extensive comparative experiments.

## III. Constrained Clustering With Imperfect Oracles

### A. Problem Formulation

Given a set of samples denoted by $X = \{\mathbf{x}_i\}$, $i = 1, \ldots, N$, with $N$ denoting the total number of samples, and $\mathbf{x}_i = (x_{i,1}, \ldots, x_{i,d}) \in \mathcal{F}$, $d$ the feature dimensionality of the feature space $\mathcal{F} \subset \mathbb{R}^d$, the goal of unsupervised clustering is to assign each sample $\mathbf{x}_i$ with a cluster label $c_i$. In constrained clustering, additional pairwise constraints are available to influence the cluster formation. There are two typical types of pairwise constraints

$$\begin{aligned} \text{Must-link} &: \mathcal{M} = \{(\mathbf{x}_i, \mathbf{x}_j) \mid c_i = c_j\} \\ \text{Cannot-link} &: \mathcal{C} = \{(\mathbf{x}_i, \mathbf{x}_j) \mid c_i \neq c_j\}. \end{aligned} \tag{1}$$

We denote the full constraint set as $\mathcal{P} = \mathcal{M} \cup \mathcal{C}$. The pairwise constraints may arise from pairwise similarity as perceived by a human annotator (oracle), temporal continuity, or prior knowledge of the sample class label. Acquiring pairwise constraints from a human annotator is expensive. In addition, owing to data ambiguity and human unintentional mistakes, the pairwise constraints are likely to be incorrect and inconsistent with the underlying data distribution.

We propose a model that can flexibly generate constraint-aware affinity matrices, which can be directly employed as input by existing pairwise similarity-based clustering algorithms, e.g., SPClust [3] or affinity propagation [5] for constrained clustering (Fig. 4). Before detailing our proposed model, we briefly describe the conventional random forests.

### B. Conventional Random Forests

*1) Classification Forests:* A general form of random forests is the classification forests. A classification forest [35] is an ensemble of $T_{\text{class}}$ binary decision trees $\mathcal{T}(\mathbf{x})$: $\mathcal{F} \rightarrow \mathbb{R}^K$, with $\mathbb{R}^K = [0, 1]^K$ denoting the space of class probability distribution over the label space $\mathcal{L} = \{1, \ldots, K\}$. During testing, each decision tree yields a posterior distribution $p_t(l|\mathbf{x}^*)$ for a given unseen sample $\mathbf{x}^* \in \mathcal{F}$, and the output probability of forest is obtained via averaging

$$p(l|\mathbf{x}^*) = \frac{1}{T_{\text{class}}} \sum_{t}^{T_{\text{class}}} p_t(l|\mathbf{x}^*). \tag{2}$$

The final class label $\hat{l}$ is obtained as $\hat{l} = \text{argmax}_{l \in \mathcal{L}}\, p(l|\mathbf{x}^*)$.

*2) Tree Training:* Decision trees are learned independently of each other, each with a random training set $X^t \subset X$, i.e., bagging [35]. Growing a decision tree involves a recursive node splitting procedure until some stopping criterion is satisfied, e.g., the number of training samples arriving at a node is equal to or smaller than a predefined node-size $\phi$, and leaf nodes are then formed, and their class probability distributions

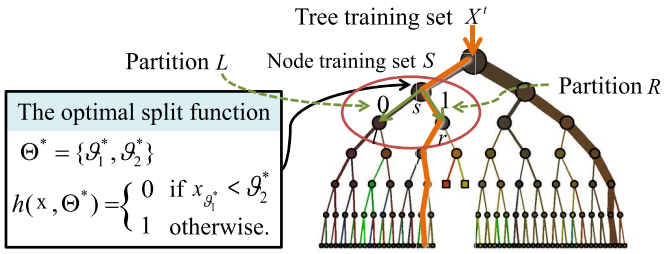



Fig. 2. Illustrative example of the training process of a decision tree.

are estimated with the labels of the arrival samples as well. Obviously, smaller $\phi$ leads to deeper trees.

The training of each internal (or split) node $s$ is a process of optimizing a binary split function defined as

$$\mathrm{h}(\mathbf{x}, \boldsymbol{\vartheta}) = \begin{cases} 0, & \text{if } x_{\vartheta_1} < \vartheta_2 \\ 1, & \text{otherwise} \end{cases} \tag{3}$$

this split function is parameterized by two parameters: 1) a feature dimension $x_{\vartheta_1}$, with $\vartheta_1 \in \{1, \ldots, d\}$ and 2) a feature threshold $\vartheta_2 \in \mathbb{R}$. We denote the parameter set of the split function as $\boldsymbol{\vartheta} = \{\vartheta_1, \vartheta_2\}$. All arrival samples of a split node will be channeled to either the left or right child node according to the output of (3).

The optimal split parameter $\boldsymbol{\vartheta}^*$ is chosen via

$$\boldsymbol{\vartheta}^* = \underset{\Theta}{\text{argmax}}\, \Delta\mathcal{I}_{\text{class}} \tag{4}$$

where $\Theta = \{\boldsymbol{\vartheta}^i\}_{i=1}^{m_{\text{try}}(|S|-1)}$ represents a parameter set over $m_{\text{try}}$ randomly selected features, with $S$ the sample set arriving at the node $s$. The cardinality of a set is given by $|\cdot|$. Particularly, multiple candidate data splittings are attempted on $m_{\text{try}}$ random feature-dimensions during the above node optimization process. Typically, a greedy search strategy is exploited to identify $\boldsymbol{\vartheta}^*$. The information gain $\Delta\mathcal{I}_{\text{class}}$ is formulated as

$$\Delta\mathcal{I}_{\text{class}} = \mathcal{I}_s - \frac{|L|}{|S|}\mathcal{I}_l - \frac{|R|}{|S|}\mathcal{I}_r \tag{5}$$

where $s$, $l$, $r$ refer to a split node and the left and right child nodes, respectively. The sets of data routed into $l$ and $r$ are denoted by $L$ and $R$, and $S = L \cup R$ denotes the sample set residing at $s$. The $\mathcal{I}$ can be computed as either the entropy or Gini impurity [36]. In this paper, we utilize the Gini impurity due to its simplicity and efficiency. The Gini impurity is computed as $\mathcal{G} = \sum_{i \neq j} p_i p_j$, with $p_i$ and $p_j$ being the proportion of samples belonging to the $i$th and $j$th categories, respectively. Fig. 2 provides an illustration of the training procedure of a decision tree.

*3) Clustering Forests:* In contrast to classification forests, clustering forests [37]–[40] require no ground truth label information during the training phase. A clustering forest consists of $T_{\text{clust}}$ binary decision trees. The leaf nodes in each tree define a spatial partitioning of the training data. Interestingly, the training of a clustering forest can be performed using the classification forest optimization approach by adopting the pseudo two-class algorithm [35], [41], [42].

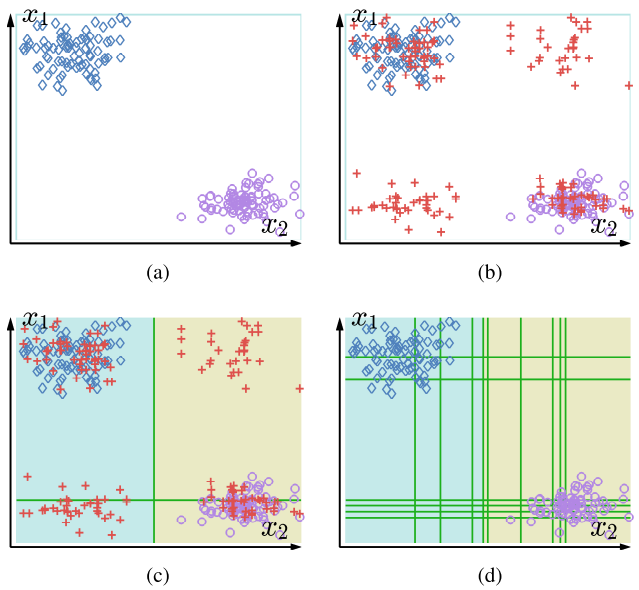



Fig. 3. Illustration of performing clustering with a random forest over a toy dataset. (a) Original toy data samples are labeled as class 1, while (b) red pseudopoints + are labeled as class 2. (c) Forest performs a two-class classification on the augmented space. (d) Resulting data partitions on the original data.

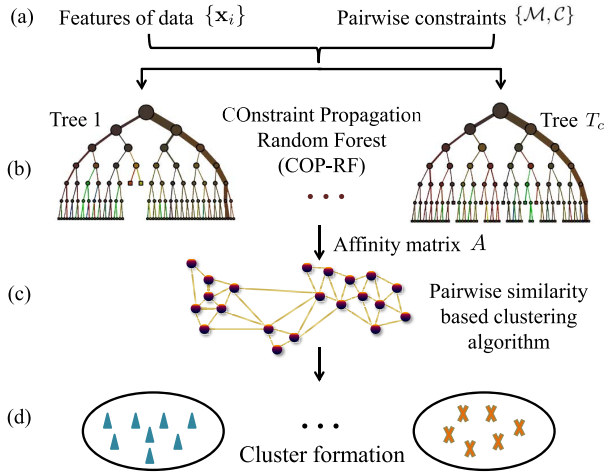



Fig. 4. Overview of the proposed constrained clustering approach. (a) The inputs into a constrained clustering model: features of data and pairwise constraints. (b) The proposed COP-RF model. (c) Performing data clustering on the derived similarity graph. (d) The obtained cluster formation.

Specifically, we add $N$ pseudosamples $\bar{\mathbf{x}} = \{\bar{x}_1, \ldots, \bar{x}_d\}$ [Fig. 3(b)] into the original data space $X$ [Fig. 3(a)], with $\bar{x}_i \sim \text{Dist}(x_i)$ sampled from certain distribution $\text{Dist}(x_i)$. In the proposed model, we adopt the empirical marginal distributions of the feature variables owing to its favorable performance [42]. With this data augmentation strategy, the clustering problem becomes a canonical classification problem that can be solved by the classification forest training method as discussed above. The key idea behind this algorithm is to partition the augmented data space into dense and sparse regions [41, Fig. 3(c) and (d)].

### C. Our Model: Constraint Propagation Random Forest

To address the issues of sparse and noisy constraints, we formulate a COP-RF, a novel variant of clustering forest (Fig. 4). We consider using a random forest, particularly a clustering forest [35], [40], [41], [43] as the basis to derive our new model for two main reasons.

1) It has been shown that random forest has a close connection with adaptive $k$-NN methods, as a forest model adapts neighborhood shape according to the local importance of different input variables [44]. This motivates us to exploit the adaptive neighborhood shape[1] for effective constraint propagation.
2) The forest model also offers an implicit feature selection mechanism that allows more accurate constraint propagation in the provided feature space by exploiting identified discriminative features during model training.

The proposed COP-RF differs significantly from the conventional random forests in that the COP-RF is formulated with a new split function, which considers not only the bottom-up data feature information gain maximization, but also the joint satisfaction of top-down pairwise constraints. In what follows, we first detail the training of COP-RF followed by how COP-RF performs constraint propagation through discriminative feature subspaces.

*1) Training of COP-RF:* The training of a COP-RF involves independently growing an ensemble of $T_c$ constraint-aware COP-trees. To train a COP-tree, we iteratively optimize the split function (3) by finding the optimal $\Theta^*$ including both the best feature dimension and cut-point to partition the node training samples $S$, similar to an ordinary decision tree (Section III-B). The difference is that the term best or optimal is no longer defined only as to maximizing the bottom-up feature information gain, but also simultaneously satisfying the imposed top-down pairwise constraints. More precisely, at the $t$th COP-tree, its training set $X^t$ only encompasses a subset of the full constraint set $\mathcal{P}$

$$\mathcal{P}^t = \left\{\mathcal{M}^t \cup \mathcal{C}^t\right\} \subset \mathcal{P} \tag{6}$$

where $\mathcal{M}$ and $\mathcal{C}$ are defined in (1). Instead of directly using the information gain in (5), we optimize each internal node $s$ in a COP-tree via enforcing additional conditions on the candidate data splittings

$$\begin{aligned} \forall(\mathbf{x}_i, \mathbf{x}_j) \in \mathcal{M}^t \Rightarrow \mathbf{x}_i, \mathbf{x}_j \in L \text{ (or } \mathbf{x}_i, \mathbf{x}_j \in R), \\ \exists(\mathbf{x}_i, \mathbf{x}_j) \in \mathcal{C}^t \Rightarrow \mathbf{x}_i \in L \;\&\; \mathbf{x}_j \in R \text{ (or opposite)}, \\ \text{where } \mathbf{x}_i, \mathbf{x}_j \in S, \text{ and } \mathcal{P}^t = \mathcal{M}^t \cup \mathcal{C}^t \end{aligned} \tag{7}$$

where $L$ and $R$ are data subsets at left child and right child (5). Owing to the conditions in (7), COP-RF differs significantly from the conventional information gain function [35], [41], [43] as the maximization of (5) is now bounded by the constraint set $\mathcal{P}^t$. Specifically, the optimization routine automatically selects discriminative features and their optimal cut-point via feature-information-based gain maximization, while at the same time fulfilling the

[1]The neighbors of a data $\mathbf{x}$ in forest interpretation are the points that fall into the same child node.

guiding conditions imposed by pairwise constraints, leading to semantically adapted data partitions.

More concretely, a data split in COP-tree can be considered as a candidate if and only if it respects all involved must-links, i.e., the constrained two samples by some must-link have to be grouped together. Moreover, candidate data splits that fulfill more cannot-links are preferred. The difference in treating must-links and cannot-links originates from their distinct inherent properties.

1) Once a particular must-link is violated at some split node, i.e., the two linked samples are separated apart, there will be no chance to compensate for agreeing again with this must-link in the subsequent process. That means that all must-links have to be fulfilled anytime.
2) While a cannot-link would be fulfilled forever once it is respected one time. This property allows us to ignore a cannot-link temporarily.

In particular, although the learning process prefers data splits that fulfill more cannot-links, it does not need to forcefully respect all cannot-links at the current split node. Algorithm 1 summarizes the split function optimization procedure in a COP-tree.

*2) Generating Affinity Matrix by COP-RF:* Every individual COP-tree within a COP-RF partitions the training samples at its leaves $\ell(\mathbf{x})$: $\mathbb{R}^d \rightarrow \mathbb{L} \subset \mathbb{N}$, where $\ell$ represents a leaf index and $\mathbb{L}$ refers to the set of all leaves in a given tree. For a given COP-tree, we can compute a tree-level $N \times N$ affinity matrix $A^t$ with elements defined as $A^t_{i,j} = \exp^{-\text{dist}^t(\mathbf{x}_i, \mathbf{x}_j)}$ where

$$\text{dist}^t(\mathbf{x}_i, \mathbf{x}_j) = \begin{cases} 0, & \text{if } \ell(\mathbf{x}_i) = \ell(\mathbf{x}_j) \\ +\infty, & \text{otherwise} \end{cases} \tag{8}$$

hence, we assign the maximum affinity (affinity $= 1$, distance $= 0$) between points $\mathbf{x}_i$ and $\mathbf{x}_j$ if they fall into the same leaf, and the minimum affinity (affinity $= 0$, distance $= +\infty$) otherwise. A smooth affinity matrix can be obtained through averaging all the tree-level affinity matrices

$$A = \frac{1}{T_c} \sum_{t=1}^{T_c} A^t. \tag{9}$$

Equation (9) is adopted as the ensemble model of COP-RF due to its advantage of suppressing the noisy tree predictions, though other alternatives such as the product of tree-level predictions are possible [45].

*3) Discussion:* Recall that the data partitions in COP-RF are required to agree with the imposed pairwise constraints, which are defined by splitting conditions in (7). From (8), it is clear that the pairwise similarity matrix induced by COP-RF is determined by the data partitions formed over its leaves. Hence, the pairwise similarity matrix induced by COP-RF indirectly encodes the pairwise constraints defined by oracles. To summarize, we denote the constraint propagation in COP-RF by the process chain below: *pairwise constraints* $\rightarrow$ *steering data partitions in COP-RF* $\rightarrow$ *distorting pairwise similarity measures*. As the data partitioning operation in COP-RF is driven by the optimal split functions that are defined on discovered discriminative features (3), the corresponding constraint propagation process takes place naturally in discriminative feature subspaces.

**Algorithm 1** Split Function Optimisation in a COP-Tree

**Input**: At a split node $s$ of a COP-tree $t$:
- Training samples $S$ arriving at a splitnode $s$;
- Pairwise constraints: $\mathcal{P}^t = \mathcal{M}^t \cup \mathcal{C}^t$;

**Output**:
- The best feature cut-point $\Theta^*$ and;
- The associated child node partition $\{L^*, R^*\}$;

**Optimisation**:
Initialise $L = R = \emptyset$ and $\Delta\mathcal{I} = 0$;
maxCLs $= 0$; /* the max number of respected cannot-links */
**for** *var* $\leftarrow 1$ to $m_{\text{try}}$ **do**
  Select a feature $x_{\text{var}} \in \{1, \ldots, d\}$ randomly;
  **for** *each possible cut-point of the feature* $x_{var}$ **do**
    Split $S$ into a candidate partition $\{L, R\}$;
    $dec =$ **validate**$(\{L, R\}, \{\mathcal{M}^t, \mathcal{C}^t\}, \text{maxCLs})$;
    **if** *dec is true* **then**
      Compute information gain $\Delta\hat{\mathcal{I}}$ following (7);
      **if** $\Delta\hat{\mathcal{I}} > \Delta\mathcal{I}$ **then**
        Update $\Theta^*$;
        Update $\Delta\mathcal{I} = \Delta\hat{\mathcal{I}}$, $L = \hat{L}$, and $R = \hat{R}$.
      **end**
    **end**
    **else**
      Ignore the current splitting.
    **end**
  **end**
**end**
**if** *No valid splitting found* **then**
  A leaf is formed.
**end**
function **validate**$(\{L, R\}, \{\mathcal{M}, \mathcal{C}\}, \text{maxCLs})$
{
  /* Deal with must-links */
  $\forall(\mathbf{x}_i, \mathbf{x}_j) \in \mathcal{M}$,
  if ($\mathbf{x}_i \in L$ and $\mathbf{x}_j \in R$, or vice versa)  return false.
  /* Deal with cannot-links */
  Count the number $\kappa$ of respected cannot-links;
  if ($\kappa <$ maxCLs)  return false.
  else  maxCLs $= \kappa$.
  Otherwise, return true.
}

### D. Coping With Imperfect Constraints

Most existing models [6], [8], [9] assume that all the available pairwise constraints are correct. It is not always so in reality, e.g., annotations from crowdsourcing are likely to contain invalid constraints due to data ambiguity or mistakes by human. The existence of fault constraints can result in error propagation to neighboring unlabelled points. To overcome this problem, we formulate a novel method

to measure the quality of individual constraints by estimating their inconsistency with the underlying data distribution, so as to facilitate more reliable constraint propagation in COP-RF.

Incorrect pairwise constraints are likely to conflict with the intrinsic data distributions in the feature space. Motivated by this intuition, we propose a novel approach to estimating constraint inconsistency measure, as described below.

Specifically, we adopt the outlier detection mechanism offered by classification random forest [35] to measure the inconsistency of a given constraint. First, we establish a set of samples with $Z = \{\mathbf{z}_i\}_{i=1}^{|\mathcal{P}|}$ with class labels $Y = \{y_i\}_{i=1}^{|\mathcal{P}|}$, where $|\mathcal{P}|$ represents the total of constraints. Here, a sample $\mathbf{z}$ is defined as

$$\mathbf{z} = \begin{bmatrix} |\mathbf{x}_i - \mathbf{x}_j| \\ \frac{1}{2}(\mathbf{x}_i + \mathbf{x}_j) \end{bmatrix} \tag{10}$$

where $(\mathbf{x}_i, \mathbf{x}_j)$ is a sample pair labelled with either must-link or cannot-link. We assign $\mathbf{z}$ with class $y = 0$ if the associated constraint is cannot-link, and $y = 1$ for must-link. Equation (10) considers both the relative position and the absolute locations of $(\mathbf{x}_i, \mathbf{x}_j)$. This characteristic enables the forest learning process to be position-sensitive and thus achieve data-structure-adaptive transformation [46].

Subsequently, we train a classification random forest $\mathbb{F}$ using $Z$ and $Y$. The learned $\mathbb{F}$ can then used to measure the inconsistency of each sample $\mathbf{z}_i$. A sample is deemed inconsistent if it is unique against other samples with the same class label. Formally, based on the affinity $\mathcal{A}$ on $Z$ that can be computed with (8) and (9) using $\mathbb{F}$, the inconsistency measure $\xi$ of $\mathbf{z}_i$ is defined as

$$\xi(\mathbf{z}_i) = \frac{\rho_i - \bar{\rho}}{\bar{\rho}}$$

where

$$\begin{aligned} \bar{\rho} &= \text{median}([\rho_1, \ldots, \rho_{|Z^i|}]) \\ \rho_i &= \frac{1}{\sum_{\mathbf{z}_j \in Z^i} (\mathcal{A}(\mathbf{z}_i, \mathbf{z}_j))^2} \end{aligned} \tag{11}$$

where $Z^i$ comprises all samples with the same class label as $\mathbf{z}_i$ in $Z$. By (11), we assign a high inconsistency score to $\mathbf{z}_i$ if it has low similarity to samples with the same class label, and a low inconsistency score otherwise. Finally, the inconsistency measure of each constraint $(\mathbf{x}_i, \mathbf{x}_j) \in \mathcal{P}$ is obtained by simply taking the $\xi$ of the corresponding $\mathbf{z}$. An overview of the proposed constraint inconsistency quantification is depicted in Algorithm 2.

To remove potentially noisy constraints, we rank all the pairwise constraints based on their inconsistency score in an ascending order. Given the rank list, we keep the top $\beta\%$ of the constraints for COP-RF training. In our study, we set $\beta = 50$ obtained by cross-validation.

**Algorithm 2** Quantifying Constraint Inconsistency

**Input**: Pairwise constraints: $(\mathbf{x}_i, \mathbf{x}_j) \in \mathcal{P} = \{\mathcal{M} \cup \mathcal{C}\}$;
**Output**: Inconsistency scores of individual constraints $(\mathbf{x}_i, \mathbf{x}_j) \in \mathcal{P}$;
**Quantifying process**:
Generate a new sample set $Z = \{\mathbf{z}_i\}_{i=1}^{|\mathcal{P}|}$ with class labels $Y = \{y_i\}_{i=1}^{|\mathcal{P}|}$ from constraints $\mathcal{P}$ (10);
Train a classification forest $\mathbb{F}$ with $Z$ and $Y$;
Compute an inconsistency score $\xi$ for each $\mathbf{z}$ or constraint (11).

### E. Constrained Clustering

After computing the affinity matrix by COP-RF (9), it can be fed into any pairwise similarity-based clustering methods, such as SPClust [1]–[4], and affinity propagation [5]. Since the affinity matrix $A$ is constraint-aware, these conventional clustering models are automatically transformed to conduct constrained clustering on data. For SPClust, we generate as model input a $k$-NN graph from $A$, a typical local neighborhood graph in the SPClust literature [3]. Following [5], we perform affinity propagation directly on $A$. In Section IV, we will show extensive experiments to demonstrate the effectiveness of the proposed COP-RF in constrained clustering.

### F. Model Complexity Analysis

COP-trees in a COP-RF model can be trained independently in parallel, as in most of the random forest models. For the worst case complexity analysis, here we consider a sequential training mode, i.e., each tree is trained one after another with a one-core CPU.

The learning complexity of a whole COP-RF can be examined from its constituent parts. Specifically, it can be decomposed into tree- and node-levels as: 1) the complexity of learning a COP-RF is directly determined by individual COP-tree training costs and 2) similarly, the training time of a single COP-tree relies on the costs of learning individual split nodes. Formally, given a COP-tree $t$, we denote the set of all the internal nodes by $\Pi_t$ and the sample subset used for training an internal node $s \in \Pi_t$ by $S$, and the training complexity of $s$ is then $m_{\text{try}}(|S| - 1)u$ when a greedy search algorithm is adopted, with $m_{\text{try}}$ the number of features attempted to partition $S$ during training $s$, and $u$ the complexity of conducting one data splitting operation. As shown in Algorithm 1, the cost of a single data partition in a COP-tree includes two components: 1) the validation of constraint satisfaction and 2) the computation of information gain. Therefore, the overall computational cost of learning a COP-RF can be estimated as

$$\Omega = \sum_{t}^{T_c} \sum_{s \in \Pi_t} m_{\text{try}} |S| u = m_{\text{try}} \sum_{t}^{T_c} \sum_{s \in \Pi_t} |S| u \tag{12}$$

where $T_c$ is the number of trees in a COP-RF. Note that the value of $\sum_{s \in \Pi_t} |S|$ depends on both the training sample size $N$ and the tree topological structure, so it is difficult to express in an explicit form if possible. In Section IV-E, we will examine the actual run time needed for training a COP-RF.

TABLE I
DETAILS OF DATASETS

| Dataset | # Clusters | # Features | # Instances |
|---|---|---|---|
| Ionosphere (Iono.) | 2 | 34 | 351 |
| Iris | 3 | 4 | 150 |
| Segmentation (Seg. ) | 7 | 19 | 210 |
| Parkinsons (Park.) | 2 | 22 | 195 |
| Glass | 6 | 10 | 214 |
| ERCe | 6 | 2672 | 600 |

## IV. EVALUATIONS

### A. Experimental Settings

*1) Evaluation Metrics:* We use the widely adopted adjusted rand index (ARI) [47] as the evaluation metric. ARI measures the agreement between the cluster results and the ground truth in a pairwise fashion, with higher values indicating better clustering quality in the range of $[-1, 1]$. Throughout all the experiments, we report the ARI values averaged over 10 trials. In each trial, we generate a random pairwise constraint set from the ground truth cluster labels.

*2) Implementation Details:* The number of trees, $T_c$, in a COP-RF is set to 1000. In general, we found that better results can be achieved by adding more trees, in line with the observation in [45]. Each $X^t$ is obtained by performing $N$ times of random selection with replacement from the augmented data space of $2 \times N$ samples (Section III-B). The depth of each COP-tree is governed by either constraint satisfaction, i.e., a node will stop growing if during any attempted data partitioning constraint validation fails (see Algorithm 1), or the size of a node equals to 1 (i.e., $\phi = 1$). We set $m_{\text{try}}$ (4) to $\sqrt{d}$ with $d$ the feature dimensionality of the input data and employ a linear data separation [45] as the split function (3). More complex split functions, e.g., quadratic functions or support vector machine, can be adopted at a higher computational cost. We set $k \approx N/10$ for the $k$-NN graph construction in the constrained SPClust experiments.

### B. Evaluation on Spectral Clustering

*Datasets:* To evaluate the effectiveness of our method in coping with data of varying numbers of dimensions and clusters, we select five diverse UC Irvine machine learning repository (UCI) benchmark datasets [48], which have been widely employed to evaluate clustering and classification techniques. We also collect an intrinsically noisy video dataset from a publicly available web-camera deployed in a university's educational resource center (ERCe). The video dataset is challenging as it contains a wide range of physical events characterized by large changes in the environmental setup, participants, and crowdedness, as well as intricate activity patterns. It also potentially contains a large amount of noise in its high-dimensional feature space. The dataset consists of 600 video clips with six possible clusters of events, namely, student orientation, cleaning, career fair, gun forum, group studying, and scholarship competition (see Fig. 5 for example images). The details of all datasets are summarized in Table I.


Fig. 5. Example images from the ERCe video dataset. It contains six events including (a) student orientation, (b) cleaning, (c) career fair, (d) group study, (e) gun forum, and (f) scholarship competition.

*Features:* For the UCI datasets, we use the original features provided. As for the ERCe video data, we segment a long video into nonoverlapping clips (each consisting of 100 frames), from which a number of visual features are then extracted, including color features (red–green–blue and hue–saturation–value), local texture features [49], optical flow, image features GISTification (GIST) [50], and person detections [51]. The resulting 2672-D feature vectors of video clips may contain a large number of less informative dimensions; we perform PCA on them and the first 30 PCA components are used as the final representation. All raw features are scaled to the range of $[-1, 1]$.

*Baselines:* For comparison, we present the results of the baselines[2] as follows.

1) *SPClust [1]:* The conventional SPClust algorithm without exploiting pairwise constraints.
2) *Constraint Propagation k-Means (COP-Kmeans) [6]:* A popular constrained clustering method based on $k$-means. The algorithm attempts to satisfy all pairwise constraints during the iterative refinement of clusters.
3) *Spectral Learning [9]:* A constrained SPClust method without constraint propagation. It extends SPClust by trivially adjusting the elements in a data affinity matrix with 1 and 0 to satisfy must-link and cannot-link constraints, respectively.
4) *$E^2CP$ [8]:* A state-of-the-art constrained SPClust approach, in which constraint propagation is achieved by manifold diffusion [25]. We use the original code released by [8], with parameter setting as suggested by the paper, i.e., we set the propagation trade-off parameter as 0.8.
5) *RF + $E^2CP$:* We modify exhaustive and efficient constraint propagation ($E^2$CP) [8], i.e., instead of generating the data affinity matrix with Euclidean-based measure, we use a conventional clustering forest (equivalent to a COP-RF without constraints imposed and noisy constraint filtering mechanism) to generate the affinity matrix. The constraint propagation is then performed using the original $E^2$CP-based manifold dif-

[2]We experimented the constrained clustering method in [26] which turns out to produce the worst performance across all datasets, and thus ignored in our comparison.

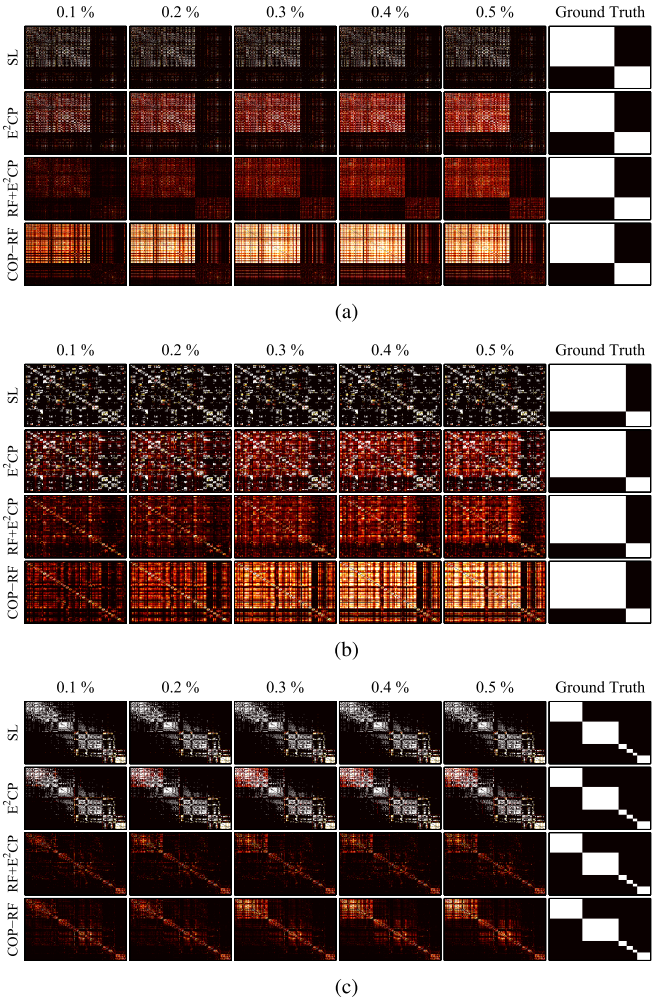



Fig. 6. Comparison of affinity matrices by different methods, given a varying number (0.1∼0.5%) of perfect pairwise constraints. (a) Ionosphere. (b) Parkinson's. (c) Glass.

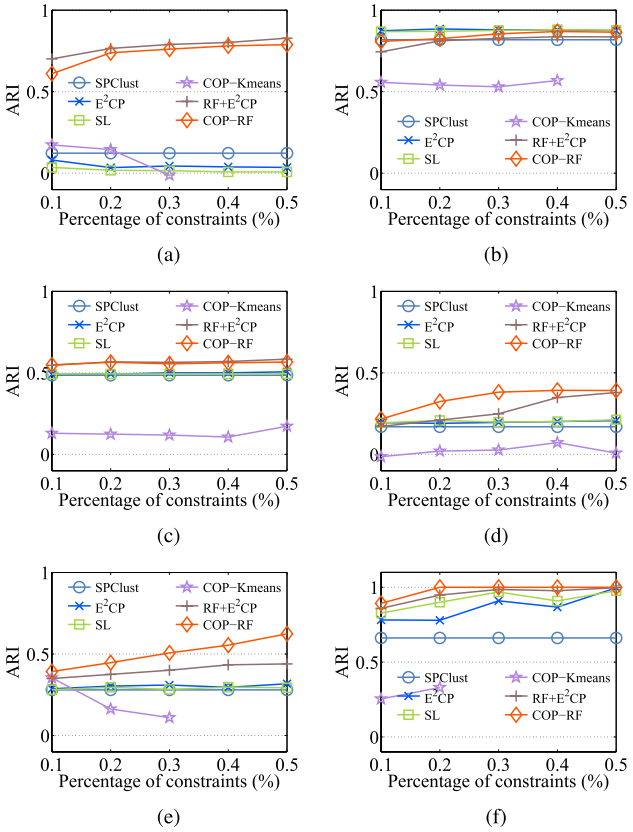



Fig. 7. ARI comparison of clustering performance between different methods given a varying number of perfect pairwise constraints. (a) Ionosphere. (b) Iris. (c) Segmentation. (d) Parkinson's. (e) Glass. (f) ERCe.

fusion. This allows $E^2$CP to enjoy a limited capability of feature selection using a random forest model.

We carried out comparative experiments to: 1) evaluate the effectiveness of different clustering methods in exploiting sparse but perfect pairwise constraints (Section IV-B1) and 2) compare their clustering performances in the case of having imperfect oracles to provide ill-conditioned pairwise constraints (Section IV-B2).

*1) Evaluation of Sparse Constraint Propagation:* In this experiment, we assume perfect oracles and thus all the pairwise constraints agree with the ground truth cluster labels. First, we examined the data affinity matrix after employing the available constraints, which may reflect how effective a constrained clustering method is. Fig. 6 shows some examples of affinity matrices produced by SL, E$^2$CP, RF + E$^2$CP, and COP-RF, respectively. COP-Kmeans is excluded since it is not a spectral method. It can be observed that COP-RF produces affinity matrices with a more distinct block structure in comparison with its competitors in the most cases. Moreover, the block structure becomes clearer when more pairwise constraints are considered. The results demonstrate the superiority of the proposed approach in propagating sparse pairwise constraints, leading to more compact and separable clusters.

Fig. 7 reports the ARI curves of different methods along with varying numbers of pairwise constraints (in the range 0.1∼0.5% of total constraints $N(N-1)/2$, where $N$ is the number of data samples). The overall performance of various methods can be quantified by the area under the ARI curve and the results are reported in Table II. It is evident from the results (Fig. 7 and Table II) that on most datasets, the proposed COP-RF outperforms other baselines, by as much as >400% against COP-Kmeans and >40% against the state-of-the-art E$^2$CP in averaged area under the ARI curve. This is in line with our previous observations on the affinity matrices (Fig. 6). Unlike E$^2$CP that relies on the conventional Euclidean-based affinity matrix that considers all features for constraint propagation, COP-RF propagate constraints via discriminative subspaces (Section III-C), leading to its superior clustering results.

We now examine and discuss the performance of other baselines. The poorest results are given by COP-Kmeans on majority datasets, beyond which some incomplete curves are observed in Fig. 7 as the model fails to converge (early termination without a solution) as more constraints are introduced into the model. On the contrary, COP-RF is empirically more stable than COP-Kmeans, as COP-RF casts the difficult constraint optimization task into smaller sub-problems to be addressed by individual trees. This characteristic is reflected

TABLE II
COMPARING DIFFERENT METHODS BY THE AREA UNDER THE ARI CURVE.
PERFECT ORACLES ARE ASSUMED. HIGHER IS BETTER

| Dataset | SPClust [1] | COP-Kmeans [6] | SL [9] | E$^2$CP [8] | RF+E$^2$CP | COP-RF |
|---|---|---|---|---|---|---|
| Ionosphere | 0.490 | 0.225 | 0.063 | 0.176 | **3.120** | 2.979 |
| Iris | 3.273 | 1.632 | 3.499 | **3.516** | 3.265 | 3.385 |
| Segmentation | 1.943 | 0.499 | 1.973 | 1.989 | **2.266** | 2.239 |
| Parkinsons | 0.677 | 0.114 | 0.811 | 0.787 | 1.082 | **1.403** |
| Glass | 1.121 | 0.394 | 1.162 | 1.210 | 1.602 | **2.015** |
| ERCe | 2.647 | 0.292 | 3.681 | 3.447 | 3.840 | **3.947** |
| Average | 1.692 | 0.526 | 1.865 | 1.854 | 2.529 | **2.661** |

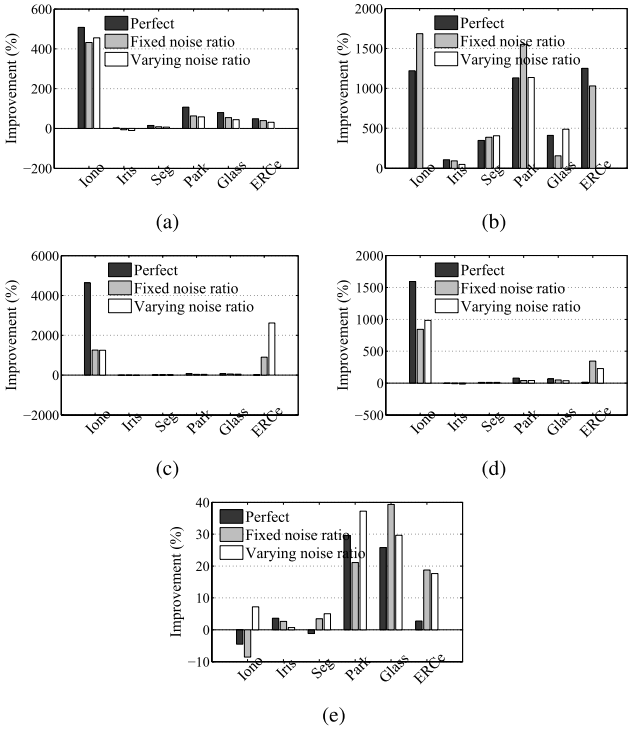



Fig. 8. Improvement of the area under the ARI curve achieved by COP-RF relative to other methods. Dark bars: when perfect constraints are provided. Gray bars: when 15% of the total constraints are noisy. White bars: when varying ratios (5∼30%) of noisy constraints are provided. (a) COP-RF over SPClust [1]. (b) COP-RF over COP-Kmeans [6]. (c) COP-RF over SL [9]. (d) COP-RF over E$^2$CP [8]. (e) COP-RF over RF + E$^2$CP.

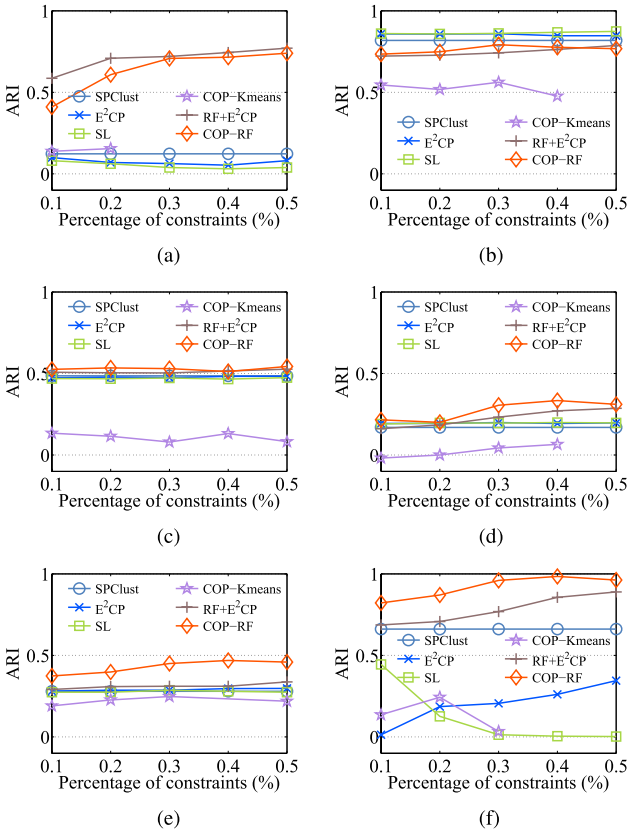



Fig. 9. ARI comparison of clustering performance between different methods, given a fixed (15%) ratio of invalid constraints. (a) Ionosphere. (b) Iris. (c) Segmentation. (d) Parkinson's. (e) Glass. (f) ERCe.

in (6), where each tree in a COP-RF only needs to consider a subset of constraints $\mathcal{P}^t \subset \mathcal{P}$.

SPClust's performance is surprisingly better than COP-Kmeans although it does not utilize any pairwise constraint. This may be because of: 1) the fact that in comparison with the conventional $k$-means, SPClust is less sensitive to noise as it partitions data in a low-dimensional spectral domain [3] and 2) the limited ability of COP-Kmeans in exploiting pairwise constraints. SL performs slightly better than SPClust through switching the pairwise affinity value in accordance with must-link and cannot-link constraints. Due to the lack of constraint propagation, SL is less effective in exploiting limited supervision information when compared with propagation-based models.

Better results are obtained by constraint propagation-based E$^2$CP. Nevertheless, the state-of-the-art E$^2$CP is inferior to the proposed COP-RF, since its manifold construction still considers the full feature space, which may be corrupted by noisy features. We observe in some cases, such as the challenging ERCe dataset, the performance of E$^2$CP is worse than that of the naive SL method that comes without constraint propagation. This result suggests that propagation could be *harmful* when the feature space is noisy. The variant modified by us, i.e., RF + E$^2$CP, employs a conventional clustering forest [41], [43] to generate the data affinity matrix. This allows E$^2$CP to take advantage of a limited capability of forest-based feature selection, and better results are obtained compared with the pure E$^2$CP. Nevertheless, RF + E$^2$CPs performance is generally poorer than COP-RFs (Table II). This is because the feature selection of the ordinary forest model is less effective than that of COP-RF, which jointly considers

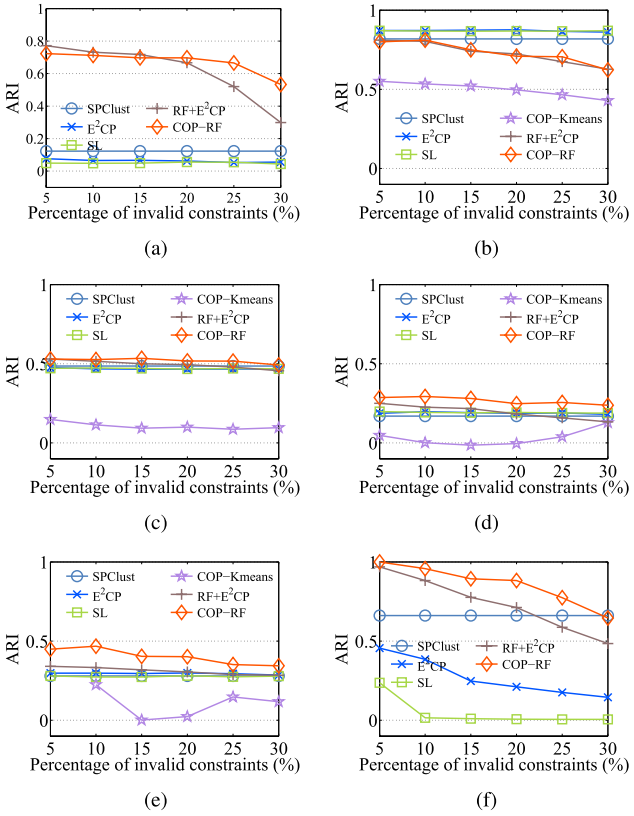



Fig. 10. ARI comparison of clustering performance between different constraint propagation methods given varying ratios of invalid constraints. (a) Ionosphere. (b) Iris. (c) Segmentation. (d) Parkinson's. (e) Glass. (f) ERCe.

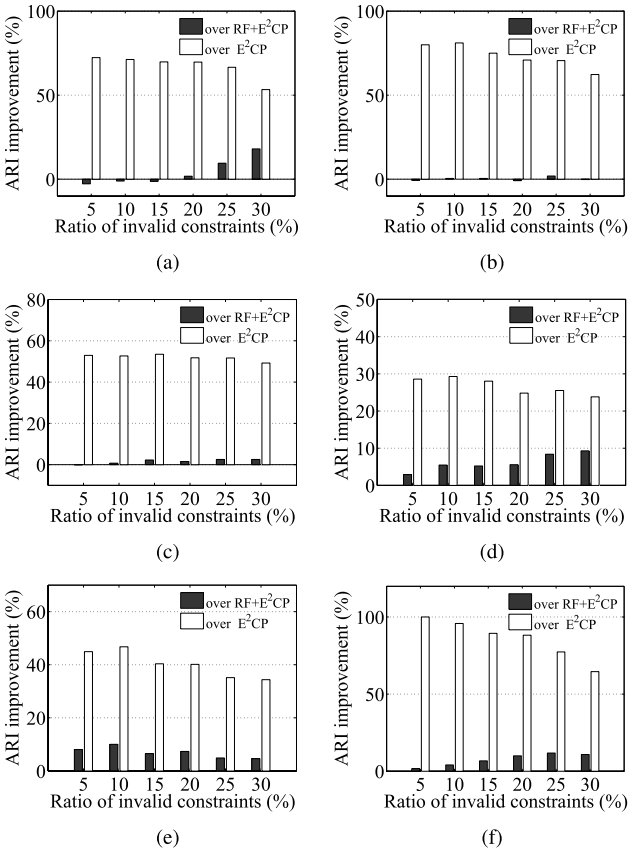



Fig. 11. ARI relative improvement of COP-RF over baseline constraint propagation models given varying ratios of noisy constraints in 0.3% out of the full constraints. Higher is better. (a) Ionosphere. (b) Iris. (c) Segmentation. (d) Parkinson's. (e) Glass. (f) ERCe.

feature-based information gain maximization and constraint satisfaction.

To further highlight the superiority of COP-RF, we show in Fig. 8 the improvement of area under the ARI curve achieved by COP-RF relative to other methods (dark bars). Clearly, while COP-RF rarely performs noticeably worse than the others, the potential improvement is large.

*2) Evaluation of Propagating Noisy Constraints:* In this experiment, we assume imperfect oracles and thus pairwise constraints are noisy. We conduct two sets of comparative experiments.

1) We deliberately introduced a fixed ratio (15%) of random invalid constraints into the perfect constraint sets as used in the previous experiment (Section IV-B1). This is to simulate the annotation behavior of imperfect oracles for the comparison of our approach with existing models.
2) Given a set of random constraints sized 0.3% of the total constraints, we varied the quantity of random noisy constraints, e.g., from 5% to 30%. This allows us to further compare the robustness of different models against mistaken pairwise constraints.

In both experiments, we repeat the same experimental protocol, as discussed in Section IV-B1.

*a) Fixed ratio of noisy constraints:* In this evaluation, we examined the performance of different models when 15% of noisy constraints are included in the given constraint sets. The performance comparison is reported in Fig. 9 and Table III and the relative improvement in Fig. 8. It is observed from Table III that in spite of the imperfect oracle assumption, COP-RF again achieves better results than other constrained clustering models on most datasets as well as the best average clustering performance across datasets, e.g., a >300% increase against COP-Kmeans and a >70% increase against E$^2$CP. Furthermore, Fig. 8 also shows that COP-RF maintains encouraging performance given noisy constraints, in some cases such as the challenging ERCe video dataset even larger improvements are obtained over E$^2$CP and other models, compared with the perfect constraint case.

*b) Varying ratios of noisy constraints:* Noisy constraints bring a negative impact on the clustering results, as shown in the above experiment. We wish to investigate how constrained clustering models would perform under different ratios of noisy constraints. To this end, we evaluated the robustness of compared models against different amounts of noisy constraints involved in sets of 0.3% out of the full pairwise constraints. Fig. 10 and Table IV show that COP-RF once again outperforms the competitor models on most datasets. As shown in Fig. 11, the performance improvement of COP-RF over constraint propagation baselines maintains over varying degrees of noisy constraints in most cases. Specifically, COP-RFs average relative improvements over E$^2$CP and RF + E$^2$CP across all datasets are 63% and 2% given 5% noisy constraints, while 48% and 8% given 30% noise, respectively.

TABLE III
COMPARING DIFFERENT METHODS BY THE AREA UNDER THE ARI CURVE. A FIXED RATIO (15%) OF INVALID PAIRWISE CONSTRAINTS IS INVOLVED. HIGHER IS BETTER

| Dataset | SPClust [1] | COP-Kmeans [6] | SL [9] | $E^2CP$ [8] | RF+$E^2CP$ | COP-RF |
|---|---|---|---|---|---|---|
| Ionosphere | 0.490 | 0.146 | 0.192 | 0.276 | **2.851** | 2.606 |
| Iris | 3.273 | 1.590 | **3.454** | 3.416 | 2.988 | 3.067 |
| Segmentation | 1.943 | 0.433 | 1.877 | 1.913 | 2.039 | **2.109** |
| Parkinsons | 0.677 | 0.067 | 0.786 | 0.780 | 0.910 | **1.102** |
| Glass | 1.121 | 0.679 | 1.114 | 1.159 | 1.244 | **1.734** |
| ERCe | 2.647 | 0.328 | 0.368 | 0.832 | 3.119 | **3.705** |
| Average | 1.692 | 0.540 | 1.299 | 1.396 | 2.192 | **2.387** |

TABLE IV
COMPARING DIFFERENT METHODS BY THE AREA UNDER THE ARI CURVE. VARYING RATIOS (5∼30%) OF INVALID PAIRWISE CONSTRAINTS ARE INVOLVED. HIGHER IS BETTER

| Dataset | SPClust [1] | COP-Kmeans [6] | SL [9] | $E^2CP$ [8] | RF+$E^2CP$ | COP-RF |
|---|---|---|---|---|---|---|
| Ionosphere | 0.536 | 0.000 | 0.253 | 0.314 | 3.172 | **3.399** |
| Iris | 4.341 | 2.507 | 4.339 | **4.352** | 3.659 | 3.684 |
| Segmentation | 2.462 | 0.514 | 2.348 | 2.336 | 2.481 | **2.605** |
| Parkinsons | 0.979 | 0.108 | 0.957 | 0.948 | 0.975 | **1.338** |
| Glass | 1.421 | 0.343 | 1.380 | 1.477 | 1.558 | **2.020** |
| ERCe | 3.160 | 0.000 | 0.159 | 1.320 | 3.682 | **4.331** |
| Average | 2.150 | 0.579 | 1.573 | 1.791 | 2.588 | **2.896** |

Fig. 12. Example face images from 10 different identities. Two distinct individuals are included in each row, each with 10 face images.

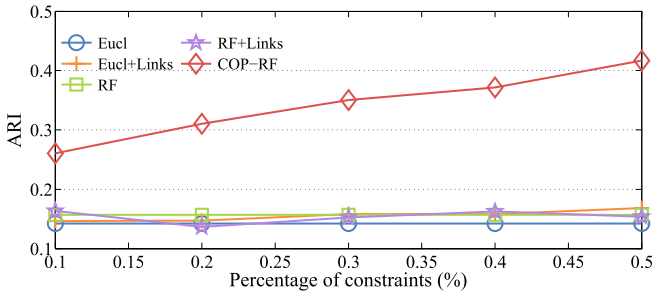



Fig. 13. Comparison of different methods on clustering face images with affinity propagation.

### C. Evaluation of Affinity Propagation

To demonstrate the generalization of our COP-RF model, we show its effectiveness on affinity propagation, an exemplar-location-based clustering algorithm [5]. Similarly, ARI is used as performance evaluation metrics.[3]

*Dataset:* We select the same face image set as [5], which is extracted from the Olivetti database. Particularly, this dataset includes a total of 900 gray images with a resolution of $50 \times 50$ from 10 different persons, each with 90 images obtained by the Gaussian smoothing and rotation/scaling transformation. It is challenging to distinguish these faces (Fig. 12) due to large variations in lighting, pose, expression, and facial details (glasses/no glasses). The features of each image are normalized pixel values with mean 0 and variance 0.1.

*Baselines:* Typically, negative squared Euclidean distance is used to measure the data similarity. Here, we compare COP-RF against the following.

1) *Eucl:* The Euclidean metric.
2) *Eucl + Links:* We encode the information of pairwise constraints into the Euclidean-metric-based affinity matrix by making the similarity between cannot-linked pairs minimal and the similarity between must-linked pairs maximal, similar to [9].
3) *Random Forest (RF):* The conventional clustering random forest [35] so that the pairwise similarity measures can benefit from feature selection.
4) *RF + Links:* Analogous to Eucl+Links, but with the affinity matrix generated by the clustering forest.

In this experiment, we use the perfect pairwise links (0.1∼0.5%) as constraints, similar to Section IV-B1. The results are reported in Fig. 13. It is evident that the feature

[3]Average squared error (ASE) is adopted in [5] as evaluation metric. This metric requires all comparative methods to produce affinity matrices based on a particular type of similarity/distance function. In our experiments ASE is not applicable since distinct affinity matrices are generated by different comparative methods.

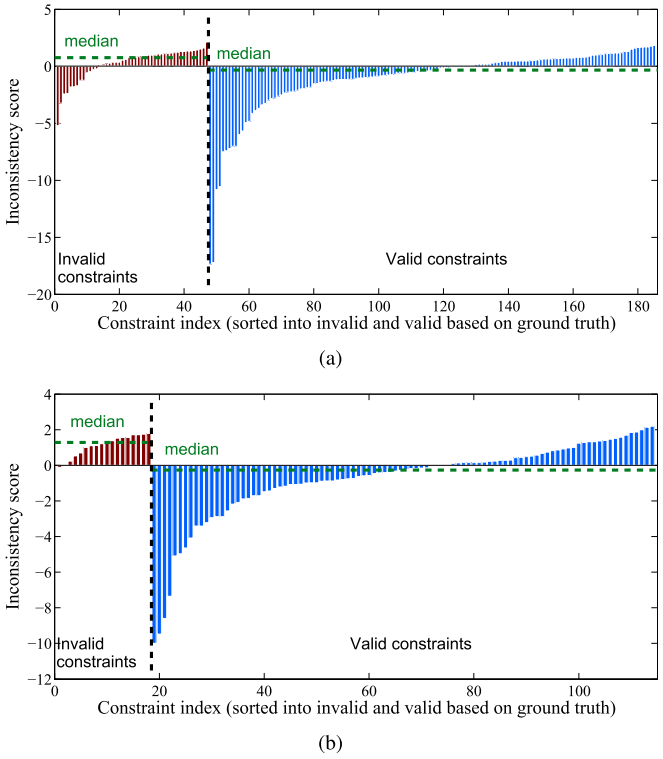



Fig. 14. Quantifying constraint inconsistency using the proposed algorithm (Section III-D). High values suggest large probabilities of being invalid constraints. (a) Ionosphere. (b) Glass.

selection-based similarity (i.e., RF) is favorable over the Euclidean metric that considers the whole feature spaces. This observation is consistent with the earlier findings in Section IV-B. Manipulating affinity matrix naively using sparse constraints helps little in performance, primarily due to the lack of constraint propagation. The superiority of COP-RF over all the baselines justifies the effectiveness of the proposed constraint propagation model in exploiting constraints for facilitating cluster formation. In addition, obviously larger performance margins are acquired when one increases the amount of pairwise constraints, further suggesting the effectiveness of constraint propagation by the proposed COP-RF model.

### D. Evaluation of Constraint Inconsistency Measure

The superior performance of COP-RF in handling imperfect oracles can be better explained by examining more closely the capability of our constraint inconsistency quantification algorithm (11). Fig. 14 shows the inconsistency measures of individual pairwise constraints on Ionosphere and Glass datasets. It is evident that the median inconsistency scores induced by invalid/noisy constraints are much higher than those by valid ones.

### E. Computational Cost

In this section, we report the computational complexity of our COP-RF model. Time is measured on a Linux machine of Intel quad-core CPU at 3.30 GHz and 8.0 GB with C++ implementation of COP-RF. Note that only one core is utilized during the model training procedure. Time analysis is conducted on the ERCe dataset using the same experimental setting as stated in Section IV-B. A total of 60 repetitions were performed, each utilizing 0.3% out of the full constraints with varying (5%∼30%) amounts of invalid ones. On average, training a COP-RF takes 213 s. Note that the above process can be conducted in parallel in a cluster of machines to speed up the model training.

## V. Conclusion

We have presented a novel constrained clustering framework to: 1) propagate sparse pairwise constraints effectively and 2) handle noisy constraints generated by imperfect oracles. There has been little work that considers these two closely related problems jointly. The proposed COP-RF model is novel in that it propagates constraints more effectively via discriminative feature subspaces. This is in contrast to existing methods that perform propagation considering the whole feature space, which may be corrupted by noisy features. Effective propagation regardless of the constraint quality could lead to poor clustering results. Our work addresses this crucial issue by formulating a new algorithm to quantify the inconsistency of constraints and effectively perform selective constraint propagation. The model is flexible in that it generates a constraint-aware affinity matrix that can be used by the existing pairwise similarity-measure-based clustering methods for readily performing constrained data clustering, e.g., SPClust and affinity propagation. Experimental results demonstrated the effectiveness and advantages of the proposed approach over the state-of-the-art methods. Future work includes the investigation of active constraint selection with the proposed model.

## References

[1] A. Y. Ng, M. I. Jordan, and Y. Weiss, "On spectral clustering: Analysis and an algorithm," in *Proc. 15th Adv. Neural Inf. Process. Syst.*, Vancouver, BC, Canada, Dec. 2002, pp. 849–856.
[2] P. Perona and L. Zelnik-Manor, "Self-tuning spectral clustering," in *Proc. 17th Adv. Neural Inf. Process. Syst.*, Vancouver, BC, Canada, Dec. 2004, pp. 1601–1608.
[3] U. von Luxburg, "A tutorial on spectral clustering," *Statist. Comput.*, vol. 17, no. 4, pp. 395–416, Aug. 2007.
[4] T. Xiang and S. Gong, "Spectral clustering with eigenvector selection," *Pattern Recognit.*, vol. 41, no. 3, pp. 1012–1029, Mar. 2008.
[5] B. J. Frey and D. Dueck, "Clustering by passing messages between data points," *Science*, vol. 315, no. 5814, pp. 972–976, Feb. 2007.
[6] K. Wagstaff, C. Cardie, S. Rogers, and S. Schrödl, "Constrained K-means clustering with background knowledge," in *Proc. 18th Int. Conf. Mach. Learn.*, Williamstown, MA, USA, Sep. 2001, pp. 577–584.
[7] Z. Lu and M. A. Carreira-Perpinán, "Constrained spectral clustering through affinity propagation," in *Proc. 21st IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2008, pp. 1–8.
[8] Z. Lu and H. H. S. Ip, "Constrained spectral clustering via exhaustive and efficient constraint propagation," in *Proc. 11th Eur. Conf. Comput. Vis.*, Sep. 2010, pp. 1–14.
[9] K. Kamvar, S. Sepandar, K. Klein, D. Dan, M. Manning, and C. Christopher, "Spectral learning," in *Proc. 18th Int. Joint Conf. Artif. Intell.*, Acapulco, Mexico, Aug. 2003, pp. 561–566.
[10] A. Kittur, E. H. Chi, and B. Suh, "Crowdsourcing user studies with mechanical turk," in *Proc. 26th Annu. SIGCHI Conf. Human Factors Comput. Syst.*, Florence, Italy, Apr. 2008, pp. 453–456.
[11] G. Patterson and J. Hays, "SUN attribute database: Discovering, annotating, and recognizing scene attributes," in *Proc. 25th IEEE Conf. Comput. Vis. Pattern Recognit.*, Providence, RI, USA, Jun. 2012, pp. 2751–2758.
[12] B. E. Rogowitz, T. Frese, J. R. Smith, C. A. Bouman, and E. B. Kalin, "Perceptual image similarity experiments," *Proc. Photon. West Electron. Imag.*, San Jose, CA, United States, Jan. 1998, pp. 576–590.

[13] D. W. Jacobs, D. Weinshall, and Y. Gdalyahu, "Class representation and image retrieval with non-metric distances," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 6, pp. 583–600, Jun. 2000.

[14] J. Laub and K.-R. Müller, "Feature discovery in non-metric pairwise data," *J. Mach. Learn. Res.*, vol. 5, pp. 801–818, Dec. 2004.

[15] E. P. Xing, A. Y. Ng, M. I. Jordan, and S. Russell, "Distance metric learning, with application to clustering with side-information," in *Proc. 15th Adv. Neural Inf. Process. Syst.*, Vancouver, BC, Canada, Dec. 2002, pp. 505–512.

[16] L. Yang and R. Jin, "Distance metric learning: A comprehensive survey," Dept. Comput. Sci. Eng., Michigan State Univ., East Lansing, MI, USA, Tech. Rep., May 2006, vol. 2.

[17] K. Q. Weinberger and L. K. Saul, "Distance metric learning for large margin nearest neighbor classification," *J. Mach. Learn. Res.*, vol. 10, pp. 207–244, Feb. 2009.

[18] M. Der and L. K. Saul, "Latent coincidence analysis: A hidden variable model for distance metric learning," in *Proc. 25th Adv. Neural Inf. Process. Syst.*, Lake Tahoe, NV, USA, Dec. 2012, pp. 3239–3247.

[19] Y. Ying and P. Li, "Distance metric learning with eigenvalue optimization," *J. Mach. Learn. Res.*, vol. 13, no. 1, pp. 1–26, Jan. 2012.

[20] S. Basu, A. Banerjee, and R. J. Mooney, "Active semi-supervision for pairwise constrained clustering," in *Proc. 4th SIAM Int. Conf. Data Mining*, Lake Buena Vista, FL, USA, Apr. 2004, pp. 333–344.

[21] X. Wang, B. Qian, and I. Davidson, "Labels vs. pairwise constraints: A unified view of label propagation and constrained spectral clustering," in *Proc. 12th IEEE Int. Conf. Data Mining*, Brussels, Belgium, Apr. 2012, pp. 1146–1151.

[22] X. Wang, B. Qian, and I. Davidson, "On constrained spectral clustering and its applications," *Data Mining Knowl. Discovery*, vol. 28, no. 1, pp. 1–30, Jan. 2012.

[23] S. X. Yu and J. Shi, "Segmentation given partial grouping constraints," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 2, pp. 173–183, Feb. 2004.

[24] Z. Li, J. Liu, and X. Tang, "Constrained clustering via spectral regularization," in *Proc. 22nd IEEE Conf. Comput. Vis. Pattern Recognit.*, Miami, FL, USA, Jun. 2009, pp. 421–428.

[25] D. Zhou, J. Weston, A. Gretton, O. Bousquet, and B. Schölkopf, "Ranking on data manifolds," in *Proc. 17th Adv. Neural Inf. Process. Syst.*. Vancouver, BC, Canada, Dec. 2004, pp. 169–176.

[26] T. Coleman, J. Saunderson, and A. Wirth, "Spectral clustering with inconsistent advice," in *Proc. 25th Int. Conf. Mach. Learn.*, Helsinki, Finland, Jul. 2008, pp. 152–159.

[27] K. L. Wagstaff, S. Basu, and I. Davidson, "When is constrained clustering beneficial, and why?" in *Proc. 21st AAAI Conf. Artif. Intell.*, Boston, MA, USA, Jul. 2006, pp. 62–63.

[28] I. Davidson, K. L. Wagstaff, and S. Basu, "Measuring constraint-set utility for partitional clustering algorithms," in *Proc. 10th Eur. Conf. Principle Pract. Knowl. Discovery Databases*, Berlin, Germany, Sep. 2006, pp. 115–126.

[29] P. Donmez and J. G. Carbonell, "Proactive learning: Cost-sensitive active learning with multiple imperfect oracles," in *Proc. 17th ACM Conf. Inf. Knowl. Manage.*, Napa, CA, USA, Oct. 2008, pp. 619–628.

[30] J. Du and C. X. Ling, "Active learning with human-like noisy oracle," in *Proc. 10th IEEE Int. Conf. Data Mining*, Sydney, NSW, Australia, Dec. 2010, pp. 797–802.

[31] Y. Yan, R. Rosales, G. Fung, and J. G. Dy, "Active learning from crowds," in *Proc. 28th Int. Conf. Mach. Learn.*, Bellevue, WA, USA, Jul. 2011, pp. 1161–1168.

[32] Y. Sogawa, T. Ueno, Y. Kawahara, and T. Washio, "Active learning for noisy oracle via density power divergence," *Neural Netw.*, vol. 46, pp. 133–143, Oct. 2013.

[33] P. Welinder and P. Perona, "Online crowdsourcing: Rating annotators and obtaining cost-effective labels," in *Proc. 23rd IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. Workshops*, San Francisco, CA, USA, Jun. 2010, pp. 25–32.

[34] X. Zhu, C. C. Loy, and S. Gong, "Constrained clustering: Effective constraint propagation with imperfect oracles," in *Proc. 13th IEEE Int. Conf. Data Mining*, Dallas, TX, USA, Dec. 2013, pp. 1307–1312.

[35] L. Breiman, "Random forests," *Mach. Learn.*, vol. 45, no. 1, pp. 5–32, Oct. 2001.

[36] L. Breiman, J. Friedman, C. J. Stone, and R. A. Olshen, *Classification and Regression Trees*. London, U.K.: Chapman & Hall, 1984.

[37] C. Liu, S. Gong, C. C. Loy, and X. Lin, "Person re-identification: What features are important?" in *Proc. 12th Eur. Conf. Comput. Vis., Int. Workshop Re-Identificat.*, Oct. 2012, pp. 391–401.

[38] C. Liu, S. Gong, and C. C. Loy, "On-the-fly feature importance mining for person re-identification," *Pattern Recognit.*, vol. 47, no. 4, pp. 1602–1615, Apr. 2014.

[39] X. Zhu, C. C. Loy, and S. Gong, "Video synopsis by heterogeneous multi-source correlation," in *Proc. 14th IEEE Int. Conf. Comput. Vis.*, Dec. 2013, pp. 81–88.

[40] X. Zhu, C. C. Loy, and S. Gong, "Constructing robust affinity graphs for spectral clustering," in *Proc. 27th IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 1450–1457.

[41] B. Liu, Y. Xia, and P. S. Yu, "Clustering through decision tree construction," in *Proc. 9th ACM Conf. Inf. Knowl. Manage.*, McLean, VA, USA, Nov. 2000, pp. 20–29.

[42] T. Shi and S. Horvath, "Unsupervised learning with random forest predictors," *J. Comput. Graph. Statist.*, vol. 15, no. 1, pp. 118–138, Jun. 2006.

[43] H. Blockeel, L. De Raedt, and J. Ramon, "Top-down induction of clustering trees," in *Proc. 15th Int. Conf. Mach. Learn.*, Madison, WI, USA, Jul. 1998, pp. 55–63.

[44] Y. Lin and Y. Jeon, "Random forests and adaptive nearest neighbors," *J. Amer. Statist. Assoc.*, vol. 101, no. 474, pp. 578–590, Jun. 2002.

[45] A. Criminisi, J. Shotton, and E. Konukoglu, "Decision forests: A unified framework for classification, regression, density estimation, manifold learning and semi-supervised learning," *Found. Trends Comput. Graph. Vis.*, vol. 7, nos. 2–3, pp. 81–227, Feb. 2012.

[46] C. Xiong, D. Johnson, R. Xu, and J. J. Corso, "Random forests for metric learning with implicit pairwise position dependence," in *Proc. 18th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, San Jose, CA, USA, Aug. 2012, pp. 958–966.

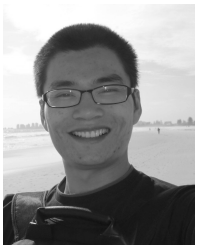
[47] L. Hubert and P. Arabie, "Comparing partitions," *J. Classification*, vol. 2, no. 1, pp. 193–218, Dec. 1985.

[48] A. Asuncion and D. J. Newman, "UCI machine learning repository," School Inf. Comput. Sci., Univ. California, Irvine, Irvine, CA, USA, Tech. Rep., 2007.

[49] T. Ojala, M. Pietikainen, and T. Maenpaa, "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 7, pp. 971–987, Jul. 2002.

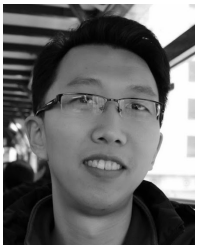
[50] A. Oliva and A. Torralba, "Modeling the shape of the scene: A holistic representation of the spatial envelope," *Int. J. Comput. Vis.*, vol. 42, no. 3, pp. 145–175, May 2001.

[51] P. F. Felzenszwalb, R. B. Girshick, D. A. McAllester, and D. Ramanan, "Object detection with discriminatively trained part-based models," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 9, pp. 1627–1645, Sep. 2010.

**Xiatian Zhu** (S'15) received the B.Eng. and M.Eng. degrees from the University of Electronic Science and Technology of China, Chengdu, China. He is currently pursuing the Ph.D. degree with the Queen Mary University of London, London, U.K.

His current research interests include computer vision, pattern recognition, and machine learning.



**Chen Change Loy** (M'15) received the Ph.D. degree in computer science from the Queen Mary University of London, London, U.K., in 2010.

He was a Post-Doctoral Researcher with Vision Semantics Ltd., London. He is currently a Research Assistant Professor with the Department of Information Engineering, Chinese University of Hong Kong, Hong Kong. His current research interests include computer vision and pattern recognition, with a focus on face analysis, deep learning, and visual surveillance.

**Shaogang Gong** received the D.Phil. degree in computer vision from Keble College, Oxford University, Oxford, U.K., in 1989.

He is currently a Professor of Visual Computation with the Queen Mary University of London, London, U.K. His current research interests include computer vision, machine learning, and video analysis.

Prof. Gong is a fellow of the Institution of Electrical Engineers and the British Computer Society.