Semi-supervised clustering via multi-level random walk

Ping He, Xiaohua Xu *, Kongfa Hu, Ling Chen

Department of Computer Science, Yangzhou University, Yangzhou 225009, China

A R T I C L E   I N F O

Article history:
Received 12 February 2013
Received in revised form 17 June 2013
Accepted 31 July 2013
Available online 12 August 2013

Keywords:
Semi-supervised clustering
Pairwise constraint
Influence expansion
Multi-level random walk
Spectral clustering

A B S T R A C T

A key issue of semi-supervised clustering is how to utilize the limited but informative pairwise constraints. In this paper, we propose a new graph-based constrained clustering algorithm, named SCRAWL. It is composed of two random walks with different granularities. In the lower-level random walk, SCRAWL partitions the vertices (i.e., data points) into constrained and unconstrained ones, according to whether they are in the pairwise constraints. For every constrained vertex, its influence range, or the degrees of influence it exerts on the unconstrained vertices, is encapsulated in an intermediate structure called component. The edge set between each pair of components determines the affecting scope of the pairwise constraints. In the higher-level random walk, SCRAWL enforces the pairwise constraints on the components, so that the constraint influence can be propagated to the unconstrained edges. At last, we combine the cluster membership of all the components to obtain the cluster assignment for each vertex. The promising experimental results on both synthetic and real-world data sets demonstrate the effectiveness of our method.

© 2013 Elsevier Ltd. All rights reserved.

1. Introduction

Semi-supervised clustering, also called constrained clustering, has become a hotspot in the current research of machine learning and data mining communities. Compared with traditional clustering [1], semi-supervised clustering takes advantage of the additional prior knowledge, such as cluster seeds or pairwise constraints, to improve the clustering result and avoid clustering ambiguity.

There are two types of supervision mostly used in semi-supervised clustering. The first one is cluster seed set [2], very similar to the labeled data set in semi-supervised classification. The second type is pairwise constraint set, which specifies the pairs of data belonging to the same cluster (must-link constraints) or different clusters (cannot-link constraints) [3]. If we view every data point as a vertex on graph, then the first category of semi-supervised clustering is a vertex-constrained learning problem, while the second category is as an edge-constrained learning problem. Since the edge constraints can be inferred from the vertex constraints, but not vice versa, it is more challenging to deal with the edge-constrained clustering problems than those vertex-constrained ones.

By far, various methods have been proposed to handle semi-supervised clustering with pairwise constraints. Generally, they can be classified into two lines. The first line, namely metric learning, learns optimized metric(s) to keep the must-linked data close and the cannot-linked data far away [4–6]. Most of the existing metric learning algorithms learn linear Mahalanobis distance metrics [4,5], but Wu et al. [6] develop a novel scheme to learn nonlinear Bregman distance functions. However, they have a generally known disadvantage that metric learning approaches require a large number of pairwise constraints to learn the correct metrics [7]. Moreover, they heavily rely on the prior assumption about the metric scope, which is hard to predict in advance. For instance, Xing et al. [4] assume that all the data points share a single global metric, while Bilento et al. [8] assume that every cluster has an independent local metric.

The second line of semi-supervised clustering algorithms focuses on adapting the existing clustering (generative) or classification (discriminative) models to deal with this problem. The early algorithms adapt the traditional methods like k-means [9], all-pairs shortest path [10], and Gaussian mixtures models [11] to find a clustering result that can satisfy all the pairwise constraints greedily. However, without the mechanism of backtracking, they may fail to find a satisfying partition even when there exists one. To tackle the sub-optimality problem, people adopt bio-inspired metaheuristic methods, such as genetic algorithm [12] and Ant Colony Optimization [13], which can explore the solution space more exhaustively and hence have a larger chance to find the global optimal solution. In recent years, more and more graph-based methods are incorporated in semi-supervised clustering algorithms. Lu [14] generalize the MAP Gaussian process classifiers to express the uncertainty information associated with the pairwise constraints in a probabilistic framework. In addition, semi-supervised clustering based on kernel methods [15], maximum
margin clustering [7], ensembles [16] and fuzzy c-means [17] have all been developed along this line.

Along the second line, there is an emerging trend in developing semi-supervised clustering algorithms based on the spectral method [18]. Kamvar et al. [19] first modify the pairwise similarity matrix by setting the must-link similarities as 1 and the cannot-link similarities as 0, then apply spectral clustering on the modified similarity matrix. However, the 1/0 modification strategy seems extreme because the data in the same cluster may not coincide and the data in different clusters probably share similar attributes. To overcome the shortcoming, Kulis et al. [20] propose a reward/penalty strategy, which adds a reward to the must-link similarities and subtracts a penalty from the cannot-link similarities. The drawback, as Li et al. [21] criticized, is that it may cause non-positive-semidefinite problem for convergence if the penalty is larger than the original similarities.

It is soon realized that by only revising the similarities of the constrained edges, it is hard to utilize the limited but informative pairwise constraints. A straightforward solution is to expand the constraint influence to the unconstrained edges, but the key issue lies in how. Although diverse efforts have been made, including the formulation of the constrained normalized cut [22], the alteration of the Laplacian matrix eigenspace [23], and the incorporation of the Gaussian process [24], they either cannot deal with the multi-class semi-supervised clustering problems or fail to handle the cannot-link constraints. Wang and Davidson [25] develop an objective function that allows real-valued degree-of-belief constraints, but it can hardly produce satisfactory result when the number of pairwise constraints is small. Li et al. [21] combine the spectral method with global metric learning to adapt the spectral embedding of the data as consistent with the pairwise constraints as possible. Nevertheless, a metric is rarely uniform in the whole domain. In another word, the structure of patterns may vary between different local neighborhoods. Thus a more appropriate way is to spread the pairwise constraints locally and exert greater influence on the nearby edges than on the faraway edges.

To confine the influence of the pairwise constraints to local areas, it is a natural choice to replace a global metric with several local metrics. Bilenko et al. [8] integrate local metric learning with constrained k means to learn an individual local metric for every cluster. The disadvantage is that they cannot deal with the data sets containing two or more local metrics in one cluster. Moreover, users need to provide much more pairwise constraints to ensure the correctness of all the local metrics. Besides, Lu and Peng [26] transform the pairwise constraint propagation into solving a continuous-time Lyapunov equation, which requires a high computational cost. Although the authors provide an approximation strategy to obtain a suboptimal solution, it still costs quadratic time complexity.

In this paper, we propose a novel approach to spreading the constraint influence to the surrounding unconstrained edges with sufficient smoothness. To this end, we decompose the constraint propagation process into three steps. First, we extract the vertices in the pairwise constraints, called constrained vertices (edge → vertex). Second, we determine the influence range of every constrained vertex by computing the degrees of influence it exerts on the unconstrained vertices (vertex → vertex). Third, we derive the affecting scope of each pairwise constraint, and enforce the pairwise constraints on the affected edges. During these three steps, each pairwise constraint is at first treated as a single constrained edge, then transformed into the influence range of two constrained vertices, and at last expanded to a group of affected edges. Therefore, we call this procedure an “edge → vertex → edge” constraint utilization strategy.

More specifically, our algorithm named SCRAWL, short for Semi-supervised Clustering via RAndom Walk, is composed of two random walks with different granularities. In the lower-level random walk, SCRAWL partitions the vertex set into the constrained and unconstrained two vertex subsets. Then it determines the influence range of every constrained vertex by translating the problem to a well-studied issue in semi-supervised classification, that is estimating the probabilities of the unlabeled data belonging to the same class of a labeled data [27]. For this purpose, a semi-supervised classification algorithm, label propagation [28], is incorporated in SCRAWL. We further encapsulate the vertices within the influence range of every constrained vertex in an intermediate structure called component. The component membership degree of each vertex equals to the degree of influence it receives from the constrained vertex. Since the overall component membership degree of each vertex is 1, we divide a whole vertex to multiple fractional vertices [29], e.g., \( v = \left[ \frac{1}{2} v, \frac{1}{2} v \right] \), according to its degrees of different component membership. In this point of view, a component is the union of the fractional vertices affected by a distinct constrained vertex. In the higher-level random walk, SCRAWL derives the affecting scope of each pairwise constraint, which is the edge set connecting the components around the two constrained vertices. We call such an edge between two fractional vertices in different components, e.g., \( E_{1/2,1/2} = (V_1, V_2) \), a fractional edge. Its fraction, determined by the product of the fractions of the connected vertices (e.g., \( 1/2 = \frac{1}{2} \times \frac{1}{2} \)), indicates the degree of influence that the whole edge \( ((V_1, V_2)) \) receives from the constrained edge. To expand the constraint influence, we enforce the pairwise constraints on the fractional edges among components, and group the components into different clusters. Finally, we obtain the cluster assignment for each vertex by combining the cluster membership of the fractional vertices distributed in different components. The promising experimental results on the synthetic data sets, UCI database and image segmentations demonstrate the effectiveness of SCRAWL.

There are several aspects of our proposed approach worthwhile to highlight here:

- SCRAWL can propagate the pairwise constraints to the surrounding unconstrained edges in proportion to the degrees of influence they receive from the constrained edges. The greater influence an unconstrained edge receives from a constrained edge, the more likely it is to satisfy the same pairwise constraint.
- The existing graph-based semi-supervised clustering algorithms confine the utilization of the pairwise constraints on edges. In contrast, SCRAWL develops an “edge → vertex → edge” constraint utilization strategy, which can expand a single constrained edge, through its two connected vertices, to a group of affected edges.
- SCRAWL introduces an intermediate structure between the fine-grained vertex and the coarse-grained cluster, called “component”. It can effectively uncover the underlying sub-structures of the clusters.
- SCRAWL establishes a connection between semi-supervised clustering and semi-supervised classification algorithms. It provides a new way to develop semi-supervised clustering algorithms based on the semi-supervised classification algorithms, which can predict the degrees of different class membership for each unlabeled data.
- SCRAWL can effectively handle the clustering problems with extremely small or large amount of pairwise constraints.
- For large real-world data sets, the time complexity of SCRAWL is approximately linear, if given a k NN sparse similarity matrix.

The remainder of this paper is organized as follows. Section 2 introduces the label propagation algorithm incorporated in SCRAWL. Section 3 describes the algorithm of SCRAWL in detail. Section 4 discusses the parameters of SCRAWL. Section 5 evaluates...
the clustering performance of SCRAWL, and Section 6 concludes the whole paper.

2. Preliminaries

Label propagation [28,30,31] is a class of semi-supervised classification algorithms based on the smoothness assumption, i.e. nearby data points bear similar labels. Given a data set \( D = \{X_i, Y_i\} \subseteq X_u \) where \( X_i = \{x_1, \ldots, x_n\} \) is the labeled data subset, \( Y_i = \{y_1, \ldots, y_n\}, y_i \in \{1, 2, \ldots, C\} \) contains the labels of \( x_i \), and \( X_u = \{x_{n+1}, \ldots, x_m\} \) is the unlabeled data set, the aim of semi-supervised classification is to predict the labels of \( X_u \), i.e. \( Y_u \).

Usually, a \( n \times C \) label indicating matrix \( Y \) is constructed in a label propagation algorithm.

\[
\hat{Y} = \begin{bmatrix} \hat{Y}_1; \ldots; \hat{Y}_m \end{bmatrix} = \hat{Y}_n \times C
\]

Finally, each unlabeled data in \( X_u \) most probably belongs to, i.e., \( \hat{Y}_u \).

3. Algorithm

3.1. Component construction

Definition 1 (Constrained vertex set). The constrained vertex set is composed of all the vertices constrained by \( C \).

\[
\mathcal{V}_c = \{v_i, v_j | \exists c \in C \land (v_i, v_j) \in E_c \lor c \in C_c \land (v_i, v_j) \in C_c \}
\]

\( |\mathcal{V}_c| \) denotes the number of the constrained vertices.

Definition 2 (Unconstrained vertex set). The unconstrained vertex set \( \mathcal{V}_u \) is the complement of \( \mathcal{V}_c \).

\[
\mathcal{V}_u = \mathcal{V} \setminus \mathcal{V}_c
\]

\( |\mathcal{V}_u| \) denotes the number of the unconstrained vertices.

Imagine such a random walk on graph \( G \). All the vertices are regarded as different states of a Markov chain. Every constrained vertex represents an absorbing state, while each unconstrained vertex represents a transient state. \( n \) particles start from \( n \) different vertices and walk randomly. For each step, a particle moves from \( v_i \) to \( v_j \) with probability \( p_{ij} \). If it reaches one of the absorbing states, it is trapped and never moves on; otherwise, it continues moving. The random walk stops when all the particles are absorbed. We use a \( n \times |\mathcal{V}_c| \) matrix \( F \) to include the probabilities of each vertex being absorbed by the different constrained vertices. Its elements also indicate the degrees of influence that each vertex receives from the constrained vertices. According to the partition of \( \mathcal{V}_c \) and \( \mathcal{V}_u \), \( F \) is composed of two parts

\[
F = \begin{bmatrix} F_c & F_u \end{bmatrix}
\]

where \( F_c \) indicates the degrees of influence within \( \mathcal{V}_c \), while \( F_u \) indicates the degrees of influence from \( \mathcal{V}_c \) to \( \mathcal{V}_u \).

First of all, we propose a \( q/q^{-1} \) modification strategy to adapt the similarity matrix consistent with the pairwise constraints. It assumes the original similarities \( w(i,j) \in [0,1] \), and defines the modified similarity matrix \( \tilde{W} = (\tilde{w}(i,j))_{n \times n} \) as follows:

\[
\tilde{w}(i,j) = \begin{cases} 
0 & \text{if } \exists c \land (v_i, v_j) \in C_c \\
n^{q-1} \frac{w(i,j)}{n} & \text{if } \exists \phi \land (v_i, v_j) \in C_{\phi} \\
n^q w(i,j) & \text{otherwise}
\end{cases}
\]

The parameter \( q \in (0,1] \) in Eq. (8) controls the power of the increase and decrease of the constrained edge similarities. When \( q = 0 \), the \( q/q^{-1} \) strategy approaches to the 1/0 strategy [19]. When \( q = 1 \), \( \tilde{W} = W \). The value of \( q \) indicates how much original similarities can be kept on the constrained edges. The larger \( q \) preserves the more original similarity, while the smaller \( q \) integrates more supervision. Fig. 1 illustrates the similarity modification of the must-linked and cannot-linked edges using our proposed \( q/q^{-1} \) strategy with various values of \( q \). Compared with the existing 1/0 and reward/penalty strategies [20], our \( q/q^{-1} \) strategy not only avoids extreme modifications, but also keeps the modified similarities within \([0,1]\).

Based on the modified similarity matrix, we can compute the transition probability matrix \( P = D^{-1}W \), where \( D = \text{diag}(W_{1,n}) \). Then we apply the label propagation method (in Section 2) to determine the degrees of influence that the unconstrained vertices receive from each constrained vertex.

\[
F_u = (I - P_{uw})^{-1}P_{uw}F_c^0
\]

Here \( P_{uw} \) indicates the transition probabilities within \( \mathcal{V}_u \), and \( P_{uc} \) indicates the transition probabilities from \( \mathcal{V}_u \) to \( \mathcal{V}_c \). Although \( \hat{Y}_u \) (Eq. [4]) and \( F_u \) (Eq. [9]) share the same converged solution, they have different meanings and are used differently for diverse purpose. First, \( \hat{Y}_u \) indicates the probabilities of the unlabeled data belonging to the different classes. It can directly predict the classes
of the unlabeled data for semi-supervised classification. In contrast, \( F_0 \) suggests the degrees of influence that each constrained vertex exerts on the unconstrained vertices. It cannot directly provide the cluster partition for semi-supervised clustering, but according to its degrees of membership to different components, it can still instruct the transition probabilities among different clusters.

\[ T \] indicates the initial state of the labeled data \( Y_0 \), which decides the converged solution of \( Y_n \), is assigned with 1/0 according to the classes of the labeled data. Therefore, the number of the different states in \( Y_n \) equals to the number of the classes. In contrast, the initial state of the constrained vertices \( F_0^\alpha \) depends on the assumption about how the constrained vertices affect each other. In this paper, we simply assume that the constrained vertices are independent from each other, i.e. \( F_0^\alpha = 1 \). Hence the number of the different states in \( F_0^\alpha \) equals to the capacity of \( v_i \).

**Definition 3 (Component).** Let \( T \) denote the component set, \( |T| \) denote the number of components. The \( j \)th component, \( T_j \), is composed of the vertices within the influence range of the \( j \)th constrained vertex. The membership degree of each vertex to \( T_j \) is indicated by the \( j \)th column of \( F \). Therefore, we call \( F \) the component indicating matrix of the vertices. Since \( \forall v_i \sum f_{ij} = 1 \), we can divide every vertex to multiple fractional vertices [29] according to its degrees of membership to different components, e.g., \( \forall v_i \sum f_{ij} = 1 \). From this perspective, \( T_j \) is the union of the fractional vertices affected by the \( j \)th constrained vertex, \( T_j = \{ f_{ij} | f_{ij} > 0 \} \).

**3.2 Component clustering**

Consider a higher-level random walk on the components. Recall that \( T_1, T_2, \ldots, T_{|T|} \) denote the \( |T| \) components, \( T_\alpha \) represents the \( \alpha \)th component around the constrained vertex \( v^\alpha \), \( T_\beta \) represents the \( \beta \)th component around the constrained vertex \( v^\beta \). The edge set between \( T_\alpha \) and \( T_\beta \) is composed of the edges that connect two fractional vertices in different components, e.g., \( e(f_{ij}v_i, f_{jk}v_k) \forall v_i, v_k, f_{ij}, f_{jk} \in T_j \). Since \( \sum f_{ij}f_{jk} = \sum f_{ij}f_{jk} = 1 \), we let \( e(f_{ij}v_i, f_{jk}v_k) = f_{ij}f_{jk} \cdot e(v_i, v_k) \).

**Definition 4 (Component similarity matrix).** The component similarity matrix is

\[ W_c = F^T \tilde{W} F \]

where the element of \( W_c = (w_c(a, b))_{T_\alpha \cap T_\beta} \) is

\[ w_c(a, b) = \frac{\sum_{i,j} f_{ij}^a f_{ij}^b \tilde{w}_{ij}}{\sum_{i,j} f_{ij}^a f_{ij}^b} \]

In **Definition 4**, \( f_{ij}^a f_{ij}^b \) is the fraction of the edge \( e(v_i, v_j) \) within the influence range of \( e(v^\alpha, v^\beta) \) (Eq. (11)). Therefore, given a pairwise constraint on \( e(v^\alpha, v^\beta) \), the pairwise component similarity \( w_c(a, b) \) computes the accumulated similarity of the fractional edges under the influence of \( \forall c = (v^\alpha, v^\beta) \) or \( \forall c = (v^\alpha, v^\beta) \). This allows us to expand the constraint influence by directly enforcing the pairwise constraints on the component similarity matrix.

To ensure the component similarities within the range of \([0, 1]\), we first compute the normalized component similarity matrix \( \tilde{W}_c \).

\[ \tilde{W}_c = D_c^{-1/2}W_c D_c^{-1/2} \]

where \( D_c = \text{diag}(W_c 1_{|T|}) \). Then we can apply our proposed \( q/q^1 \) modification strategy on \( \tilde{W}_c \), leading to the modified component similarity matrix \( W_c = (w_c(a, b))_{T_\alpha \cap T_\beta} \).

\[ \tilde{W}_c(a, b) = \begin{cases} \tilde{W}_c(a, b) & \text{if } \forall c = (v^\alpha, v^\beta) \in C_\alpha, \\ \tilde{W}_c(a, b) & \text{if } \exists c = (v^\alpha, v^\beta) \in C_\beta \\ \tilde{W}_c(a, b) & \text{otherwise} \end{cases} \]

By this way, each pairwise constraint is propagated to all the affected edges according to the degrees of influence they receive from the constrained edge.

Next, we compute the transition probability matrix of the component-level random walk, \( P_c = D_c^{-1} \tilde{W}_c \) with \( D_c = \text{diag}(W_c 1_{|T|}) \). Meila and Shi [32] have proved that the sum of the transition probabilities among different clusters equals to the normalized cut among them. Then we can group the components into different clusters based on \( P_c \). An approximately optimal solution to the normalized cut is \( U = [u_1, u_2, \ldots, u_p] \), where \( u_1, u_2, \ldots, u_p \) satisfy \( P_c u_i = \lambda_1 u_i, \lambda_1 \geq \cdots \geq \lambda_p \). Since the \( i \)th row of \( U \) indicates the cluster membership of \( T_i \), we call \( U \) the cluster indicating matrix of the components.

**3.3 Component combination**

To combine the cluster membership of the fractional vertices distributed in the different components, we multiply the component indicating matrix of the vertices \( \{ F_{n \times |T|} \} \) with the cluster indicating matrix of the components \( \{ U_{|T| 	imes p} \} \), to obtain the cluster indicating matrix of the vertices.

\[ G = FU \]

The \( i \)th row vector of \( G \) indicates the cluster membership of \( v_i \).

Fig. 2 shows an intuitive illustration of the computation of \( G \). It is similar to a three-layer feedforward artificial neural network. The vertices constitute the input layer, the components form the hidden layer, and the clusters are in the output layer. \( F \) is the input to the hidden layer. \( U \) is the weight of the connections from the hidden layer to the output layer. \( G \) records the output of the neural network. The \( j \)th column of \( G \), provided by the \( j \)th output neuron, is an approximation to the semi-supervised clustering function for the \( j \)th cluster. The components in the hidden layer play a crucial
role in the representation of semi-supervised clustering functions by this neural network.

To explain the relationship of \( G \) with other unsupervised and semi-supervised spectral clustering algorithms, we rewrite the optimization of \( U \) as

\[
U = \arg \max_{U \in \mathbb{R}^{n \times p}} \text{tr}(U^T \phi_C(W,F)U)
\]

where \( \phi_C(W,F) \) is a function that computes and modifies the component similarity matrix based on \( C \).

1. If \( \phi_C(W,F) = F^TWF \), which means all the pairwise constraints are discarded or \( C = \emptyset \), then \( G = FU \) is the optimal solution to the unsupervised spectral clustering algorithms, like Normalized Cut [33].

2. If \( \phi_C(W,F) = F^T\psi(W)F \), where \( \psi(W) \) is a function that adapts the vertex similarity matrix to be consistent with the pairwise constraints, e.g., similarity modification strategies or metric learning approaches, then \( G = FU \) is the optimal solution to the semi-supervised spectral clustering algorithms based on the adaptation of \( W \), including Spectral Learning [19], SS-Kernel K-means [20] and Constrained Clustering with Spectral Regularization [21].

3. If \( \phi_C(W,F) = \phi_C \circ \phi_C(W,F) \), where \( \phi_C(W,F) = F^T\psi(W)F = W_{C,F} \), \( \phi_C(W) = \phi_C(W_{1/2}) \), which modifies the component similarity matrix, then \( G = FU \) is the optimal solution to our algorithm SCRAWL. Compared with the semi-supervised spectral clustering algorithms in 2, SCRAWL takes one more step to expand the constraint influence through components.

Finally, we project the row vectors of \( G \) onto a unit hypersphere [34], \( \tilde{G} = D_{G}^{-1/2} G \), where \( D_{G} = \text{diag}(\text{diag}(GG^T)^{-1}) \), and then use \( k \)-means to group the \( n \) row vectors of \( \tilde{G} \) into \( p \) clusters.

### 3.4. Time complexity

The time complexity of the component construction stage is dominated by the computation of the component indicating submatrix \( F_{u} \). If we directly compute the converged solution of \( F_{u} \) with Eq. (9), it will cost \( O(n(|T|)^3) \), which is too expensive for large real-world data sets. However, it can be reduced if we use a \( k \) NN sparse similarity matrix and compute \( F_{u} \) with the state transition equation:

\[
F_{u}^{t+1} = P_{uv} + P_{au}F_{u}^{t}
\]

which is derived from \( F^{t+1} = PF^{t} \) and \( P_{u}^2 = I \). Since each vertex in the \( k \) NN sparse similarity matrix is only connected with its \( k \) nearest neighbors, each row of \( P_{uv} \) contains \( k \) nonzero elements at most. As a result, the time complexity of the iterative computation of \( F_{u} \) is \( O(n(|T|)kT_{\text{max}}) \), where \( T_{\text{max}} \) denotes the maximal number of iterations.

In the component clustering stage, the time complexity is dominated by the computation of \( W_{i} \) and the eigenvalue decomposition of \( P_{i} \). The computation of \( W_{i} \) costs \( O(n^2|T|) \) if the vertex similarity matrix \( W \) is dense, but it only costs \( O(nk|T|) \) if \( W \) is \( k \) NN sparse. Because \( W_{i} \) is always dense, the component transition probability matrix \( P_{i} \) is also dense. The time complexity of the eigenvalue decomposition of \( P_{i} \) is \( O(|T|^3) \). In the end, the component combination stage costs \( O(n|T|p) \) for the computation of \( G \). To summarize, if we use a dense vertex similarity matrix, the time complexity of SCRAWL is \( O(n(|T|)^3) + O(n^2|T|) + O(n|T|^3)(\text{for computation of } G) + O(n|T|p) \). If we use a \( k \) NN sparse vertex similarity matrix, the time complexity is \( O(n(|T|)k|T|_{\text{max}}) + O(nk|T|) + O(|T|^3) + O(n|T|p) \), where both \( k \) and \( T_{\text{max}} \) are removed as user-specified constants. Since \( |T| \) and \( p \) compared with \( n \) on large data sets are so small that can be ignored, we can further reduce the time complexity of SCRAWL to approximately linear \( O(n) \).

### 4. Parameters

In this section, we discuss two important parameters, including the similarity modification strength \( q \) and the component number \( |T| \), by studying their impact on the clustering performance of SCRAWL. We choose a modified version of \( F \)-measure, named Constrained \( F \)-measure, as the performance evaluation criterion.

\[
F-\text{measure} = \frac{2\text{PR}}{P + R}
\]

where

\[
P = \frac{TP - IC_{-\text{true}}}{TP + FP - IC_{-\text{true}}}, \quad R = \frac{TP - IC_{-\text{false}}}{TP + FN - IC_{-\text{false}}}
\]

\( TP \) (True Positive) is the number of the pairs of data that belong to the same class and are assigned to the same cluster, \( FP \) (False Positive) is the number of the pairs of data that belong to different classes but are assigned to the same cluster, \( FN \) (False Negative) is the number of the pairs of data that belong to the same class but are assigned to different clusters, \( |C_{-\text{true}}| \) is the number of the must-link constraints. Notice that Constrained \( F \)-measure excludes the influence of the pairwise constraints (Eq. (20)) to adapt for the evaluation of semi-supervised clustering algorithms.

In the following section, we take iris data set for example, and illustrate the way to determine the appropriate parameters for SCRAWL. The number of clusters is set the number of classes, which is 3 for iris data set.

#### 4.1. Similarity modification strength

In the component construction stage, we have proposed a \( q/q^{-1} \) similarity modification strategy. It is first applied on the vertex similarity matrix \( W \) (Eq. (8)) and later on the normalized component similarity matrix \( W_{C} \) (Eq. (15)). In order to differentiate the two similarity modification in different granularities, we use \( q_{t} \) to denote the modification strength on the vertex similarity matrix, and use \( q_{\nu} \) to denote the modification strength on the component similarity matrix.

#### 4.1.1. \( q_{t} \)

The \( q_{t}/q_{\nu}^{-1} \) modification of \( W \) refines the influence range of each pairwise constraint. We fix the other parameters \( q_{t} = 1 \) (which ignores the modification of \( W_{C} \)), \( |T| = |V_{C}| \), and analyze the influence of \( q_{t} \) in Fig. 3.

In Fig. 3(b), as \( q_{t} \) decreases from 1 to 0.01, more and more supervision is integrated into the constrained edge similarities, and the clustering performance of SCRAWL exhibits significant improvement. However, as \( q_{t} \) further declines from 0.01 to 0.001,
It indicates that as long as $q_v$ is smaller than a threshold (e.g., $q_v \leq 0.01$), the influence of $q_v$ on the clustering performance of SCRAWL essentially keeps unchanged.

### 4.1.2. $q_c$

The $q_c/q_v^{-1}$ modification of $W_e$ expands the influence of the pairwise constraints to the affected fractional edges. To reveal the relationship of $q_v$ and the clustering performance of SCRAWL, we fix $q_v = 0.01, |T| = |V_c|$, and depict the learning curves of $q_v$ in Fig. 4.

In Fig. 4, the learning curves of $q_v$ at first climb up rapidly with the decreases of $q_v$ from 1 to 0.1, but then slow down as $q_v$ declines from 0.1 to 0.01 (Fig. 4(b)), and finally keep about flat when $q_v \in (0.001, 0.01)$ (Fig. 4(a)). It indicates that as long as $q_v$ is smaller than a threshold (e.g., $q_v \leq 0.01$), the influence of $q_v$, similar to that of $q_v$, on the clustering performance of SCRAWL is almost the same. Although in Figs. 3 and 4 the performance of the $q_v/q_v^{-1}$ strategy is similar to that of the 0/1 strategy on iris data set, it does not indicate that we can replace the $q_v/q_v^{-1}$ strategy with the 0/1 strategy for all the data sets. Assume such a case, in which the similarities within clusters are very low ($10^{-5}$) and the similarities among clusters are even lower ($10^{-7}$). Given a must-link constraint, if we use the 0/1 strategy that sets the must-linked similarities with 1, then the difference between the intra-cluster and inter-cluster similarities will be replaced by the larger difference between the constrained and unconstrained similarities (1 and near 0), leading to a wrong clustering result. In contrast, if we use the $q_v/q_v^{-1}$ strategy for similarity modification, then the optimal $q_v$ will be close to 1 instead of 0. Therefore, a significant difference between the 0/1 strategy and the $q_v/q_v^{-1}$ strategy is that the 0/1 strategy sets the constrained edge similarities regardless of their original values, while the $q_v/q_v^{-1}$ strategy integrates the original similarities and the supervisory information with $q_v$ controlling their tradeoff, being more general and flexible.

### 4.2. Component number

In the component construction stage, we build the same number of components as the constrained vertices ($|T| = |V_c|$). However, this strategy has problems in dealing with extremely small or large number of pairwise constraints. When $|V_c| < p$, such as traditional unsupervised clustering, SCRAWL cannot partition $|V_c|$ components into $p$ clusters. When $|V_c| \rightarrow n$, the affecting scope of each constrained vertex decreases to itself ($F = I$), then SCRAWL is reduced to the semi-supervised clustering algorithm based on vertex similarity modification.

In order to appropriately deal with the above mentioned extreme cases, we put a cap on the number of components, denoted as $|T_{lu}|$. When $|V_c| \leq |T_{lu}|$, $|T| = |V_c|$. When $|V_c| > |T_{lu}|$, $|T| = |T_{lu}|$, we randomly select $|T_{lu}|$ constrained vertices for component construction. It ensures that each component contains at least $n/|T_{lu}|$ vertices on average, and at most $(|T_{lu}|^2/|V_c|^2)\cdot C_0$ pairwise constraints can be propagated through the components on
average. Fig. 5(a) illustrates the learning curves of $n/|I_u|$ on iris data set, where the influence of the other parameters is excluded with $q_r = q_c = 0.01$.

In Fig. 5(a), the minimal component size $n/|I_u| = 1$ corresponds to the maximal component number $|T| = |V_c|$, because $|V_c| \leq |I_u| = n$. The maximal component size $n/|I_u| = 50$ corresponds to the minimal component number $|T| = p$, where $p = 3$ is the number of classes for iris data set. With the increase of $n/|I_u|$, the learning curves of SCRAWL first rise rapidly because of the expanded influence range of each pairwise constraint, and then decline at a slower pace due to the reduced number of the propagated pairwise constraints. The performance gap between the optimal $n/|I_u|$ and the benchmark $n/|I_u| = 1$ demonstrates that an appropriate component size improves the clustering performance of SCRAWL significantly.

On the other hand, we also set a floor under the number of components, denoted as $|I_u|$. When $|V_c| \geq |I_u|$, $|T| = \min(|V_c|, |I_u|)$. When $|V_c| < |I_u|$, $|T| = |I_u|$, we randomly select $(|I_u| - |V_c|)$ unconstrained vertices together with $V_c$ to construct components. It ensures that each cluster contains at least $|T|/p$ components on average. Fig. 5(b) illustrates the learning curves of $|T|/p$ when $|V_c| < |I_u|$ on iris data set, where $q_r = q_c = 0.01$ and $|T| = n$.

In Fig. 5(b), when $|C| = 0$ (or $|C| = 1$), $|V_c| = 0$ (or $|V_c| = 2$), we use different numbers of unconstrained vertices for component construction. Although none pairwise constraint is propagated through the components around the unconstrained vertices, we find that SCRAWL still produces much better clustering result than the unsupervised spectral clustering algorithm NCut [34]. This may because the components reveal the underlying structure of the data set, which makes the clustering easier. Besides, the rise and decline of the learning curves about $|T|/p$ suggest the importance of component diversity within clusters.

Algorithm 1 summarizes the complete algorithm of SCRAWL. Step 1 infers the transitive closure of $C$ to obtain more pairwise constraints [9]. Steps 2–11 are the component construction stage, steps 12–16 form the component clustering stage, and steps 17–19 constitute the component combination stage. At last, step 20 returns the cluster partition predicted by SCRAWL.

5. Experiments

In this section, we evaluate the clustering performance of SCRAWL in comparison with another four semi-supervised clustering algorithms including Spectral Learning (SL) [19], SS-Kernel-Kmeans (SSKK) [20], Constrained Clustering with Spectral Regularization (CSSR) [21] and Metric Pairwise Constrained Kmeans (MPCK) [8]. Among them, SL, SSKK, CSSR and SCRAWL are all semi-supervised spectral clustering algorithms implemented in Matlab, while MPCK is an integrated approach of constrained k-means and local metric learning implemented in Java.\footnote{\url{http://www.cs.utexas.edu/users/ml/risc/code/}}

Algorithm 1. SCRAWL($W, C, p, q, |T|$).

\begin{algorithm}
\hspace{1cm}1 $C^{*} \leftarrow \text{ConstrainTransitiveClosure}(C)$
\hspace{1cm}2 $\tilde{W} \leftarrow q^{1/2} W$ modification of $W$ using Eq. (8)
\hspace{1cm}3 $P \leftarrow D^{-1/2} \tilde{W}$
\hspace{1cm}4 $F_t \leftarrow I_{|T| 	imes |T|}$
\hspace{1cm}5 if $W$ is dense
\hspace{1cm}6 $F_u \leftarrow (1 - P_{uu})^{-1} P_{uu}$
\hspace{1cm}7 else
\hspace{1cm}8 $\tilde{F}_u^{0} \leftarrow 0_{|T| 	imes |T|}$
\hspace{1cm}9 $F_u \leftarrow \text{iterative computation based on Eq. (18)}$
\hspace{1cm}10 end
\hspace{1cm}11 $F_w \leftarrow F_t^{T} F_u^{T}$
\hspace{1cm}12 $W_c \leftarrow F_t^{T} W F$
\hspace{1cm}13 $\tilde{W}_c \leftarrow D_{c}^{-1/2} \tilde{W} D_{c}^{-1/2}$, where $D_c \leftarrow \text{diag}(W_c 1_{|T|})$
\hspace{1cm}14 $W_c \leftarrow q^{1/2}$ modification of $\tilde{W}_c$ using Eq. (15)
\hspace{1cm}15 $P_{c} \leftarrow D_{c}^{-1} \tilde{W}_c$, where $D_c \leftarrow \text{diag}(\tilde{W}_c 1_{|T|})$
\hspace{1cm}16 $U \leftarrow [u_1, u_2, \ldots, u_p]$, where $P_c u_i = \lambda_i u_i$, $\lambda_1 \geq \cdots \geq \lambda_p$
\hspace{1cm}17 $G \leftarrow \text{FU}$
\hspace{1cm}18 $\tilde{G} \leftarrow D_{c}^{-1/2} G$, where $D_c = \text{diag}(\text{diag}(G G^T))$
\hspace{1cm}19 $f \leftarrow \text{KMEANS} (\tilde{G}, p)$
\hspace{1cm}20 return $f$
\end{algorithm}

We first illustrate the clustering results of SCRAWL on two artificial data sets, and then compare SCRAWL with another four semi-supervised clustering algorithms on nine UCI data sets and
seven image segmentations given both must-link and cannot-link constraints. For each UCI data set, at least 10 different numbers of pairwise constraints are provided based on the ground-truth class labels. For each number of pairwise constraints, 50 different realizations are randomly generated to compute the average clustering performance. We use two criteria, including Constrained $F$-measure and NMI, to evaluate the clustering results visually. The Constrained $F$-measure has been defined in Eqs. (19) and (20). The definition of NMI, short for Normalized Mutual Information [35], is as follows:

$$\text{NMI}(\Omega, \Psi) = \frac{I(\Omega; \Psi)}{[H(\Omega) + H(\Psi)]/2}$$

where $\Omega$ is the set of predicted clusters, $\Psi$ is the set of true classes, $I$ is the mutual information:

$$I(\Omega; \Psi) = \sum_k \sum_j p(o_k \cap \psi_j) \log_2 \frac{p(o_k \cap \psi_j)}{p(o_k)p(\psi_j)}$$

and $H$ is the entropy function:

$$H(\Omega) = -\sum_k p(o_k) \log_2 p(o_k)$$

At last, we list the running time spent on the image segmentations for the discussion of their runtime performance.

In respect of graph construction, we construct 20NN (20-Nearest-Neighbors) sparse graphs for the artificial data sets, and construct fully connected graphs for the UCI data sets. Their pairwise similarity matrices $W$ are computed by the Gaussian radial basis function:

$$w_{ij} = e^{-\frac{(x_i - x_j)^2}{2\sigma^2}}$$

where the scale parameter $\sigma$ is optimized over the interval $(2^{2^{-5/5}}, 2^{2^{-4/5}}, ..., 2^{2^{-4/5}}, 2^{2^{-5/5}})$ in each run for each number of constraints. We select the optimal $\sigma$ that minimizes the sum of the within-cluster point-to-centroid distances, leading to the tightest clusters of the row vectors of $G$ [34]. The rest parameters are set according to the parameter analysis on iris data set in Section 4, including the similarity modification strength $q_v = q_e = 0.01$, the component number $|T| = \max(|T_p|, |T_n|)$, $|T_p| = 2 \cdot \min(|V_c|, |T_n|)$, $|T_n| = 0.05 \cdot |V_c|$. Note that the optimal parameters for iris data set may not be optimum for other applications, but we use the same parameters throughout the experiments for simplicity and consistency.

The images used in the semi-supervised image segmentation are selected from the Berkeley Segmentation Database [36]. Each image is scaled to $170 \times 113$ pixels. The similarities between each pair of pixels are computed by measuring the magnitude of their intervening contours [33].

$$w_{ij} = e^{-\frac{\text{line}(ij) + |\text{edge}(\mathbf{x})|}{2\sigma^2}}$$

$\text{line}(ij)$ is a straight line that connects the $i$th and $j$th pixels, $\text{edge}(\mathbf{x})$ is the magnitude of the intervening contour at location $\mathbf{x}$, and $\sigma$ is set $1/10$ of the maximal edge magnitude on graph. To keep sparsity of the similarity matrices for image segmentation, we
only allow each pixel to connect with its neighboring pixels within a circle of radius $r = 10$.

All the following experiments are executed in MATLAB R2010b on a Mac with 1.7 GHz Intel Core i5 and 4 GB RAM.

5.1. Toy examples

Fig. 6 shows the clustering results of SCRAWL on two synthetic data sets. One is Yin-yang data set, where the two classes of data

Fig. 7. The components constructed by SCRAWL on Yin-yang data set. The solid symbols indicate the constrained data, the hollow symbols indicate the unconstrained data. The darker a symbol, the higher its probability of belonging to the component.

Fig. 8. The Constrained $F$-measure curves of the five semi-supervised clustering algorithms on UCI data sets.
are mutually embedded. The other is Tri-circle data set, where the three classes of data are not only intersecting but also overlapping. On both of the two synthetic data sets, SCRAWL produces satisfactory clustering results. We take Yin-yang data set for example, and plot the components constructed around the constrained vertices in Fig. 7. From Fig. 7, we can see that SCRAWL successfully recognizes the underlying structure of the data set, which allows the pairwise constraints to be propagated to unconstrained data correctly.

5.2. UCI database

Figs. 8 and 9 respectively compare the Constrained F-measure and NMI curves of the five semi-supervised clustering algorithms on nine UCI data sets. The attribute values of tissue, parkinsons, statlog and liver data sets are scaled into the range of $[-1, 1]$, and the missing data and the first attribute (ID number) of breast data set are removed.

In Figs. 8–9, the comparison results of the Constrained F-measure curves and NMI curves demonstrate the same fact that, among the five clustering algorithms, SCRAWL exhibits the best clustering performance with the increase of the number of pairwise constraints. Although the learning curves of SCRAWL at first fall behind those of CCSR on small numbers of pairwise constraints, they climb up quickly and soon exceed the learning curves of the other algorithms with growing advantage. On the contrary, despite the superior performance on the initial pairwise constraints, CCSR hardly presents great improvement even when the number of pairwise constraints grows very large. This is probably due to the increasing difficulty in adapting the limited dimension ($D=15$) of the spectral space to satisfy the growing number of pairwise constraints. Different from SCRAWL and CCSR, SSKK produces good results on some of the data sets, but also shows poor performance on the others. It may because the reward and penalty setting is not universally appropriate for all the data sets. The learning curves of SL have the most similar ascending trend to those of SCRAWL. However, because of its lack of constraint propagation, SL falls behind of SCRAWL with large gaps. Finally, MPCK first shows a “dip” before rising on its learning curves. The reason might be that the metrics learned from few pairwise constraints are unreliable and more pairwise constraints can improve this situation [8].

![Diagram showing NMI curves for UCI data sets](image-url)
5.3. Image segmentation

The existing semi-supervised image segmentation algorithms either use cluster seeds or pure must-link constraints to define the desired content for extraction [37,38]. In this experiment, we evaluate the four graph-based semi-supervised clustering algorithms, including SL, SSKK, CCSR and SCRAWL, on seven natural color image segmentations given both must-link and cannot-link constraints. MPCK is replaced by the state-of-the-art unsupervised spectral clustering algorithm NCut [33] in comparison, because NCut is based on the same graphs (Eq. (25)) as the other four semi-supervised clustering algorithms, while MPCK is not.

Fig. 10 illustrates the segmentation results of the four semi-supervised spectral clustering algorithms compared with the benchmark NCut [33]. In the first two rows, different pairwise constraints are provided to localize the worm and the plant respectively on the same input image. The third to the fifth rows bias the extraction content toward the left bird, the right bird and both the two birds on another input image. The last two rows aim to differentiate the trees, the house and the swan from their reflections in the lake and the background. The comparison results demonstrate the superiority of SCRAWL over the other semi-supervised clustering algorithms.

The reason for the success of SCRAWL on image segmentation is because each image is composed of several local image elements, which correspond to different contents or subjects, while the components constructed by SCRAWL are the ideal approximations of those image elements if not their fragments. The automatic recognition of the local image elements, combined with the enforcement of the pairwise constraints, reduces the semi-supervised image segmentation to the local image element clustering based on the adapted similarities. Fig. 11 illustrates the components constructed for different image segmentations by SCRAWL. In each subfigure, a light is placed at a constrained vertex, so that a brighter pixel indicates a higher probability of its belonging to the component. Table 1 summarizes the running time of the five clustering algorithms on the semi-supervised image segmentations in Fig. 10. Although theoretically the time complexities of NCut, SL, SSKK and CCSR are all $O(n^3)$, dominated by the eigenvalue decomposition of $W$, SSKK spends the least computational time, while SL spends the most. The reason lies in the adoption of sparse similarity matrices, whose eigenvalue decomposition has to be executed by the eigs function (in Matlab) that computes the largest $p$ eigenvectors with the Lanczos method. Its time complexity, upper-bounded by $O(n^2)$, depends not only on the matrix sparsity, but also on the relative gap between the
largest eigenvalue and the next one. Since the four algorithms NCut, SL, SSKK and CCSR use diverse sparse similarity matrices for eigenvalue decomposition, they spend different running time on it. In contrast, no matter whether $W$ is sparse or dense, SCRAWL always gets a dense component similarity matrix. As a result, it will never have the instability problem caused by the eigenvalue decomposition of sparse matrices. To summarize, according to the approximately linear time complexity of SCRAWL and its runtime performance in Table 1, we can say that SCRAWL is not only an effective but also efficient semi-supervised image segmentation algorithm.

### 6. Conclusions

Random walk is a popular bottom-up technique used in supervised and unsupervised learning tasks. It can locally explore the neighborhood of each data by computing the transition probabilities from it and to it. In this paper, we propose a semi-supervised clustering algorithm named SCRAWL under a framework of multi-level random walk. It first determines the propagation range of each pairwise constraint in a vertex-level random walk, and then expands the influence of pairwise constraint in a component-level random walk. Compared with other semi-supervised clustering algorithms, SCRAWL exhibits superior performance on artificial data sets, UCI database and semi-supervised image segmentations.

However, our proposed approach also has limitations, which suggest the following directions of our future study.

- In the real-world applications, the pairwise constraints provided by different domain experts may conflict with each other. For instance, $(A,B)$ and $(B,C)$ both satisfy must-link constraints, but $(A,C)$ satisfies a cannot-link constraint. When SCRAWL is applied to such problems, it will infer unexpected constraint transitive closure, because the previously inferred constraints could get easily violated by the subsequently inferred ones.
Hence we intend to investigate how to preprocess the inconsistent pairwise constraints for SCRAWL.

- Even when there is no conflicting constraints, SCRAWL is sensitive to noisy constraints for two reasons. First, it will infer more mislabeled pairwise constraints from the noisy ones. Second, it will expand the influence of the mislabeled pairwise constraints, leading to a wrong clustering result. To alleviate the problem, except for disabling the inference of the transitive closure for noisy constraints, we also intend to adjust the modification strength $q$, whose complement $(1-q)$ indicates the constraint confidence, for every distinct pairwise constraint.

Conflict of Interest

None declared.

Acknowledgments

The authors would like to thank Dr. Li for their code of CCSR and the anonymous reviewers for their valuable comments and suggestions to significantly improve the quality of this paper. This research was supported in part by the Chinese National Natural Science Foundation under Grant nos. 61030180, 61070047 and 61103018, Natural Science Foundation of Education Department of Jiangsu Province under contracts 13KJB520026 and 09KJB00013, Natural Science Foundation of Jiangsu Province under contracts BK2010318 and BK2011442, and the New Century Talent Project of Yangzhou University.

References


Ping He received her Ph.D. degree in computer science from Nanjing University of Aeronautics and Astronautics of China in 2012, and M.S. degree from Yangzhou University of China in 2008. Her research interests include machine learning, data mining and bioinformatics.

Xiao-hua Xu received his Ph.D. degree in computer science from Nanjing University of Aeronautics and Astronautics of China in 2008, and M.S. degree from Yangzhou University of China in 2005. His research interests include machine learning, evolutionary computation, and parallel algorithms.

Kong-fa Hu is a professor in the Computer Science Department at Yangzhou University, Yangzhou, P.R. China. He received his Ph.D. degree in computer science from Southeast University of China in 2004. His research interests include data mining, database, and data warehouse.

Ling Chen is a professor in the Computer Science Department at Yangzhou University, Yangzhou, P.R. China. He did research for two years on parallel algorithms and architectures at the University of Pittsburgh, PA, first as a visiting scholar in 1986 and, later, as a visiting associate professor in 1992. His research interests include parallel algorithm design, artificial intelligence, and bioinformatics. Professor Chen is a member of IEEE Computer Society and Chinese Computer Society.