Selecting Graph Cut Solutions via Global Graph Similarity

Canh Hao Nguyen, Nicolas Wicker, and Hiroshi Mamitsuka

Abstract— **Graph cut is a common way of clustering nodes on similarity graphs. As a clustering method, it does not give a unique solution under usually used loss functions. We specifically show the problem in similarity graph-based clustering setting that the resulting clusters might be even disconnected. This is counter-intuitive as one wish to have good clustering solutions in the sense that each cluster is well connected and the clusters are balanced. The key property of good clustering solutions is that the resulting graphs (after clustering) share large components with the original ones. We wish to detect this case by deriving a graph similarity measure that shows high similarity values to the original graph for good clustering solutions. The similarity measure considers global connectivities of graphs by treating graphs as distributions in (potentially different) Euclidean spaces. The global graph comparison is then turned into distribution comparison. Simulation shows that the similarity measure could consistently distinguish different qualities of clustering solution beyond what could be done with the usually used loss functions of clustering algorithms.**

Index Terms— **Graph cut, graph embedding, graph similarity, Hilbert–Schmidt information criterion.**

I. Introduction

The problem is that clustering algorithms might produce many solutions with the same optimality. The loss functions of clustering algorithms might not be able to detect the bad solutions due to their limitations. It is particularly the case in graph-based clustering [4], [9], [10], [15], [21]. The loss functions are usually the cut quality involving a small cut value (total weight cut) and the balance of clusters. However, the gap is that the small cut value does not directly lead to the well connectedness within clusters as desired. Detecting a cluster being connected or not is computationally expensive to be used as a loss function in clustering.

To simplify the discussion, we limit to the case of cutting a graph into two clusters. We show the cases in which the optimal solutions, if they were actually returned from the clustering algorithms, are not unique. Hence, even if we suppose that the algorithms work as desired, the solutions might have different qualities. For our problem setting of graph-based clustering, an optimal solution of the problem of finding two clusters might have disconnected clusters. Graph cut methods in computer vision [2], [3], [8], [19], while having the same objective of coherent clusters, require the clusters to follow supervised information rather than only the balance of clusters in graph cut-based clustering setting.

We wish to be able to select the good solutions from many equally optimal solutions of graph cut, two-cluster clustering algorithms using the usual criteria that clustering algorithms use. This is not supposed to be used to replace the clustering loss functions, but to address the issue, which is failed to be accommodated by the graph cut loss functions. Finally, we wish to find a criterion that, even though it might be too expensive to be used as a clustering objective function, should be more expressive to distinguish some bad solutions that are failed to be recognized by the usual loss functions. A problem of graph cut with spectral clustering was illustrated in [20]. However, this brief does not give a single criterion to compare clustering solutions. It is possible to use this criterion to multicluster clustering problem. However, as one has to combine the criterion on more clusters, like other criteria, it is harder to analyze why it should work or fail.

The usual objective of clustering is to find clusters, which are balanced in size and well connected within each cluster. Hence, a good clustering solution, in terms of the objective function, should have two large clusters that each shares many edges with the original graph. On the other hand, a bad one might have clusters of uneven sizes, and there might be disconnected clusters. As the former case is cut with a smaller value in the sense of having a less number of connected components, the resulting graph after cut is closer to the original one than that of the latter case. To distinguish the two cases, we need to design a graph comparison method that considers global connectivities of graphs. Specifically, we need a similarity measure that makes the former case more similar to the original graph than the latter case is. This is a new graph similarity criterion that has not been addressed before.

We formalize this requirement as the problem of comparing labeled graphs with the same node set. The problem is stated as follows. Graph $G_i = (V, E_i)$, $\forall i = 1, 2, \ldots$ with the node set V fixed. We wish to be able to compare the graphs, such as a similarity function between G_i and G_j, considering global connectivities of the graphs.

There are many ways to compare graphs proposed in the literature. Almost all of them are based on the parts of the graphs as features for comparison. The examples include walks, paths [12], [16], and subgraphs of the graphs [5]. The idea behind all these methods is that the properties of a graph are determined by its parts separately. Therefore, these methods do not reflect the topology of a graph as a whole, such as connectivity. Another line of work is based on graph edit distance [6]. However, in our special case of graphs with the same node set, edit distance becomes the number of different edges. The role of each edge is different for the global connectivity of the whole graph (for example, some edges are bridges while others are not). In this case, the difference of a bridge in the graph would be more than other edges, while edit distance treats them all equally. We wish to be able to incorporate graph connectivity into our comparison as well. There are some other works using graph Laplacians [24] for feature extractions.

In this brief, we first show the conditions in which the undesirable solutions with disconnected clusters might be obtained in even the optimal clustering solutions in graphs. We propose to use graph Laplacians to compare graphs with the same node set to reflect its global connectivity. As graph Laplacians allow embeddings of graphs into Euclidean spaces, we instead compare the embeddings. Hence, the method is called graph embedding-based graph comparison.

Manuscript received November 18, 2012; revised August 20, 2013; accepted November 21, 2013. Date of publication December 12, 2013; date of current version June 10, 2014. This work was supported in part by KAKENHI, MEXT, Japan, under Grant 24300054.

C. H. Nguyen and H. Mamitsuka are with the Bioinformatics Center, Institute for Chemical Research, Kyoto University, Uji 611-0011, Japan (e-mail: canhhao@kuicr.kyoto-u.ac.jp; mami@kuicr.kyoto-u.ac.jp).

N. Wicker is with the UFR de Mathématiques, Université Lille 1, Villeneuve d'Ascq 59655, France (e-mail: nicolas.wicker@math.univ-lille1.fr).

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.2013.2292975

We also show extensions of the comparison using eigendecompositions of the graph Laplacians. We show the properties of the comparison to observe the importance of edges toward global connectivity of the whole graph, which is different from edit distance where all the edges have an equal weight. We provide some experiments to show that our similarity measure of graphs can detect the good solutions from the bad one in two-cluster clustering problem.

II. Problem With Graph Cut-Based Clustering

It is commonly known that, even though the clustering algorithms return the optimal solutions according to their loss functions, there might be undesirable solutions. Following are some undesirable properties of the solutions.

1) There are many different solutions. This is a result of the nonconvexity of the loss functions.
2) For graph-based clustering, some clusters are not connected.

This section aims to provide the conditions in which the solutions are not desirable. This is the problem we are trying to solve.

One of the common graph-based clustering algorithms is based on graph cut [4] as follows. Given a graph $G = (V, E)$, a graph cut problem is to divide the set of nodes into two subsets that: 1) are even in size or volume and 2) have a small number of edges between the two subsets. Let $\bar{X}$ denote the complement of any set X in V. One looks for a cut $(C, \bar{C})$ $(C \subset V)$ that minimizes the loss function

$$l(C,\bar{C}) = \frac{\mathrm{cut}(C,\bar{C})}{\min(s(C), s(\bar{C}))} \qquad (1)$$

where $s(\cdot)$ could be the total degrees (volume) or total number of nodes (size) of a set. The cut function in the numerator can be instantiated as $\mathrm{cut}(C,\bar{C}) = \sum_{(i,j)\in E, V_i \in C, V_j \in \bar{C}} w_{ij}$. In our setting, we allow s to be any positive modular function $(s(C \cup D) + s(C \cap D) = s(C) + s(D),\ \forall C, D \subseteq V)$. Examples of s include the commonly used normalized Cheeger cut and ratio Cheeger cut. There are other loss functions for graph-based clustering [10], [21]. These loss functions have different drawbacks, such as favoring more unbalanced clusters [4]. Further discussion is out of the scope of this brief.

The problem with graph cut for clustering using the above loss functions is that the cut $(C, \bar{C})$, which minimizes the loss functions l might be disconnected. The following theorem shows the necessary and sufficient conditions for this to happen.

Theorem 1: Suppose that $(C, \bar{C})$ is a cut that minimizes the cost function in (1). Suppose that the C-induced subgraph G' of G is disconnected, then the following conditions hold:

1) $s(C) < s(\bar{C})$;
2) for D as any union of the nodes in any set of maximally connected components of G', then

$$l(C,\bar{C}) = l(D,\bar{D}).$$

Proof: Let D' be the complement of D in G', then $D \cup D' = C$, $D \cap D' = 0$ and $\bar{D} = \bar{C} \cup D'$. Because D is a set of maximally connected components, D is not connected to D' in G. Then

$$\mathrm{cut}(C,\bar{C}) = \mathrm{cut}(D,\bar{D}) + \mathrm{cut}(D',\bar{D}').$$

Proving the first condition by contradiction. Suppose that $\min(s(C), s(\bar{C})) = s(\bar{C}) < s(C)$, meaning that the larger cluster is disconnected.

Without loss of generality, assume that $s(D) \geq s(D')$. Then, there are two cases. In both cases, we use the assumption that C minimizes the loss function in (1).

Case 1: $s(D) \leq s(\bar{C}) + s(D')$, then

$$\begin{aligned}\frac{\mathrm{cut}(C,\bar{C})}{\min(s(C),s(\bar{C}))} &\leq \frac{\mathrm{cut}(D,\bar{D})}{\min(s(D),s(\bar{D}))} = \frac{\mathrm{cut}(D,\bar{D})}{s(D)}\\ \frac{\mathrm{cut}(C,\bar{C})}{\min(s(C),s(\bar{C}))} &\leq \frac{\mathrm{cut}(D',\bar{D}')}{\min(s(D'),s(\bar{D}'))} = \frac{\mathrm{cut}(D',\bar{D}')}{s(D')}.\end{aligned}$$

Then, using the weighted average of fractions, we have

$$\begin{aligned}\frac{\mathrm{cut}(C,\bar{C})}{\min(s(C),s(\bar{C}))} &\leq \frac{\mathrm{cut}(D,\bar{D}) + \mathrm{cut}(D',\bar{D}')}{s(D)+s(D')}\\ &= \frac{\mathrm{cut}(C,\bar{C})}{s(C)}\\ &< \frac{\mathrm{cut}(C,\bar{C})}{s(\bar{C})}\\ &= \frac{\mathrm{cut}(C,\bar{C})}{\min(s(C),s(\bar{C}))}.\end{aligned}$$

This is a contradiction.

Case 2: by the definition of C, we have $s(D) > s(\bar{C}) + s(D')$. Then

$$\begin{aligned}\frac{\mathrm{cut}(C,\bar{C})}{\min(s(C),s(\bar{C}))} &\leq \frac{\mathrm{cut}(D,\bar{D})}{\min(s(D),s(\bar{D}))}\\ \Leftrightarrow \frac{\mathrm{cut}(D,\bar{D}) + \mathrm{cut}(D',\bar{D}')}{s(\bar{C})} &\leq \frac{\mathrm{cut}(D,\bar{D})}{s(\bar{C})+s(D')}.\end{aligned}$$

As all components in the inequalities are nonnegative, the second inequality does not hold, leading to a contradiction. Both cases lead to contradictions. Hence, the larger cluster is not disconnected. The first condition is proved.

Second condition:

$$\begin{aligned}\min(s(C),s(\bar{C})) &= s(C)\\ &= s(D)+s(D')\\ &= \min(s(D),s(\bar{D})) + \min(s(D'),s(\bar{D}')).\end{aligned}$$

Then

$$\begin{aligned}\frac{\mathrm{cut}(C,\bar{C})}{\min(s(C),s(\bar{C}))} &\leq \frac{\mathrm{cut}(D,\bar{D})}{\min(s(D),s(\bar{D}))}\\ \frac{\mathrm{cut}(C,\bar{C})}{\min(s(C),s(\bar{C}))} &\leq \frac{\mathrm{cut}(D',\bar{D}')}{\min(s(D'),s(\bar{D}'))}\\ \frac{\mathrm{cut}(C,\bar{C})}{\min(s(C),s(\bar{C}))} &\leq \frac{\mathrm{cut}(D,\bar{D}) + \mathrm{cut}(D',\bar{D}')}{\min(s(D),s(\bar{D})) + \min(s(D'),s(\bar{D}'))}\\ &= \frac{\mathrm{cut}(C,\bar{C})}{\min(s(C),s(\bar{C}))}.\end{aligned}$$

This shows that all the three inequalities above must be equalities, then

$$\frac{\mathrm{cut}(C,\bar{C})}{\min(s(C),s(\bar{C}))} = \frac{\mathrm{cut}(D,\bar{D})}{\min(s(D),s(\bar{D}))}.$$

■

The implication of the theorem is that, given any optimal graph cut that produces a disconnected cluster, equally optimal solutions can be obtained by cutting only some of the maximally connected components of the disconnected cluster.

Theorem 1: Given $(D, \bar{D})$ and $(D', \bar{D}')$ are two optimal solutions of (1). If $D \cap D' = \emptyset$, $\mathrm{cut}(D, D') = 0$, and $s(D) + s(D') \leq 1/2 s(V)$, then $(D \cup D', \bar{D} \cap \bar{D}')$ is also an optimal solution of (1).

Proof:

$$\begin{aligned}\frac{\mathrm{cut}(D,\bar{D})}{s(D)} &= \frac{\mathrm{cut}(D',\bar{D}')}{s(D')}\\ &= \frac{\mathrm{cut}(D,\bar{D})+\mathrm{cut}(D',\bar{D}')}{s(D)+s(D')}\\ &= \frac{\mathrm{cut}(D\cup D',\bar{D}\cap\bar{D}')}{s(D\cup D')}.\end{aligned}$$

Thus, the combined cut $(D\cup D',\bar{D}\cap\bar{D}')$ is also optimal. ■

This theorem gives a way to combine the optimal solutions to make new solutions without losing its optimality. It also shows that it is possible to obtain the optimal solutions with a disconnected cluster.

We conclude that the type of loss functions as in (1) cannot guarantee the well connectedness of clusters. We aim to fix this case.

III. Graph Embedding-Based Graph Comparison

To consider graph connectivity, we use the embeddings of graphs that reflect their global connectivities. We propose to embed the graphs into Euclidean spaces and compare them. We use graph Laplacians [7] for embedding and Hilbert–Schmidt information criterion (HSIC) [13] for graph comparison. We first review the HSIC to be used later in our similarity measure of graphs. We then propose our graph embedding-based similarity measure (ges), its equivalent formulation and its extensions.

A. HSIC

HSIC is a measure of statistical independence between the two random variables. Let x be a random variable in the domain $\mathcal{X}$ and z is another random variable in $\mathcal{Z}$. Let $\mathcal{F}$ and $\mathcal{G}$ be the feature spaces on $\mathcal{X}$ and $\mathcal{Z}$ with kernels $k:\mathcal{X}\times\mathcal{X}\to\mathbb{R}$ and $l:\mathcal{Z}\times\mathcal{Z}\to\mathbb{R}$, respectively. Drawing samples (x,z) and (x',z') from a joint probability distribution $p_{(x,z)}$, then HSIC is computed via kernel functions as

$$\begin{aligned}\mathrm{HSIC}(p_{(x,z)},\mathcal{F},\mathcal{G}) = {} & E_{x,x',z,z'}[k(x,x')l(z,z')]\\ & +E_{x,x'}[k(x,x')]E_{z,z'}[k_{z,z'}]\\ & -2E_{x,z}\big[E_{x'}[k(x,x')]E_{z'}[k(z,z')]\big].\end{aligned} \tag{2}$$

Empirical estimation of HSIC is as follows:

$$\mathrm{HSIC}(p_{(x,z)},\mathcal{F},\mathcal{G}) = \frac{1}{(n-1)^2}\,\mathrm{trace(KHLH)} \tag{3}$$

where H is the centering matrix.

Considering the nodes of a graph in its embedded space as a set of samples of a random variable. Specifically, K_1 is a kernel for a sample of a random variable, and K_2 is a kernel for a sample for another variable. HSIC between the two variables, intuitively the correlation information between the two embeddings of the same node set, can be computed as

$$\begin{aligned}\mathrm{HSIC}(G_1,G_2) &\overset{\mathrm{def}}{=} \mathrm{HSIC}(p_{(K_1,K_2)})\\ &= \frac{1}{(n-1)^2}\mathrm{trace}(K_1K_2).\end{aligned}$$

(K_i are already centered as a property of eigenvectors of graph Laplacians).

B. Graph Embedding-Based Graph Comparison

To reflect the global connectivities of graphs, we use graph Laplacians. The procedure has two steps as follows.

Step 1 (Graph Embedding): We use graph Laplacians to embed nodes of a graph into an Euclidean space. There are many ways to embed nodes for different purposes of learning methods on the nodes [22]. For the moment, we just use the inverse of graph Laplacian as the kernel for the node set

$$K_i = L_i^{-1}. \tag{4}$$

Step 2 (Graph Comparison): The kernel encodes a sampling distribution; hence the problem of comparing graphs becomes the problem of comparing distributions. We specifically use HSIC as a similarity measure (to be specific, a kernel) between the distributions. We define our graph embedding-based graph comparison with the following similarity measure.

Definition 1: The similarity of two graphs ges is defined to be

$$\begin{aligned}\mathrm{ges}(G_1,G_2) &\overset{\mathrm{def}}{=} (n-1)^2\cdot\mathrm{HSIC}(K_1,K_2)\\ &= \mathrm{trace}(K_1K_2).\end{aligned} \tag{5}$$

For HSIC being a kernel between the distributions, our similarity measure is also a kernel (a similarity measure) between the graphs. Other similarity measures or dissimilarity measures such as Euclidean distance between the graphs can be constructed on the top of this kernel as usual.

Similarly, the distance between the two graphs can be defined based on $\mathrm{ges}(G_1,G_2)$, called graph embedding-based distance (ged), as

$$\mathrm{ged}(G,G') = \sqrt{\mathrm{ges}(G,G)+\mathrm{ges}(G',G')-2\mathrm{ges}(G,G')}. \tag{6}$$

C. Spectral Computation and Extensions

It is known that there are many ways of defining kernels from graph Laplacians by spectral transforms. We show that for all the kernels obtained by spectral transforms, there is a formulation to compute the graph similarity measure using eigenvalues and eigenvectors of the graph Laplacians, as in [23]. Moreover, the formulation allows natural extensions of the similarity measure that are not available from the distribution comparisons.

Suppose that the graph Laplacians are eigendecomposed as $L_1 = UD(\lambda)U^T$ and $L_2 = VD(\mu)V^T$, in which $D(x)$ denotes a diagonal matrix with diagonal entries are the vector x. It is known that there are many kernels built on graph Laplacian with spectral transform $g(\lambda)$, then $K_1 = UD(g(\lambda))U^T$ [22]. For example, the kernels K_1, K_2 as in (4), $g(\lambda)=1/\lambda$ (for $\lambda>0$, and $g(0)=0$).

Theorem 2: Suppose that the two graphs G_1 and G_2 define a joint distribution of nodes, denoted as $p(G_1,G_2)$, then the similarity defined in (5), as a HSIC between the two distributions defined by K_1 and K_2, is computed with their eigenvalues (λ_i,μ_j) and eigenvectors (u_i,v_j) as follows:

$$\mathrm{ges}(G_1,G_2) = \sum_{i,j=1}^{n} g(\lambda_i)\,\cdot <u_i,v_j>^2\cdot g(\mu_j). \tag{7}$$

Proof:

$$\begin{aligned}\mathrm{ges}(G_1,G_2) &= \mathrm{trace}(K_1K_2)\\ &= \mathrm{trace}\big\{UD(g(\lambda))U^T\cdot VD(g(\mu))V^T\big\}\\ &= \mathrm{trace}\big\{D(\sqrt{g(\lambda)})U^T\cdot VD(g(\mu))V^T\cdot\\ &\quad UD(\sqrt{g(\lambda)})\big\}\\ &= \mathrm{trace}\big\{D(\sqrt{g(\lambda)})U^TVD(\sqrt{g(\mu)})\cdot\\ &\quad D(\sqrt{g(\mu)})V^TUD(\sqrt{g(\lambda)})\big\}\\ &= \mathrm{trace}\big\{D(\sqrt{g(\lambda)})U^TVD(\sqrt{g(\mu)})\cdot\\ &\quad [D(\sqrt{g(\lambda)})U^TVD(\sqrt{g(\mu)})]^T\big\}.\end{aligned} \tag{8}$$

Let $M = D(\sqrt{g(\lambda)})U^TVD(\sqrt{g(\mu)})$, then $M_{ij} = \sqrt{g(\lambda_i)}u_i^Tv_j\sqrt{g(\mu_j)}$. Note that $trace(MM^T) = \sum_{i,j}M_{ij}^2$, then

$$\begin{aligned}\mathrm{ges}(G_1, G_2) &= \mathrm{trace}(K_1 K_2)\\ &= \mathrm{trace}(MM^T)\\ &= \sum_{ij}^{n} g(\lambda_i)\cdot < u_i, v_j >^2 \cdot g(\mu_j).\end{aligned} \tag{9}$$

This completes the proof. ■

Note that the graph Laplacian-based comparison has the form

$$\mathrm{ges}(G_1, G_2) = \sum_{i,j=1}^{n} f(\lambda_i, \mu_j) < u_i, v_j >^2 \tag{10}$$

with $f(\lambda_i, \mu_j) = g(\lambda_i)g(\mu_j)$. This allows extensions of the comparison by any other function f instead of $g(\lambda_i)g(\mu_j)$. Some examples, as in [23] are:

1) $f(\lambda_i, \mu_j) = \mathrm{Breg}(g(\lambda), g(\mu))$, where Breg means Bregman divergence [1] between the two vectors. In this case, the comparisons are also Bregman divergences;
2) $f(\lambda_i, \mu_j) = s(g(\lambda), g(\mu))$, where s denotes any similarity measure such as correlation.

IV. Edge Weighting Property

We show how the similarity measure gives different weights to different edges in a graph to reflect its importance to connectivity of the graph, as opposed to edit distance.

Proposition 1: Edit distance between the two graphs G and G_1 is the number of different edges between the two graphs.

Given a graph $G = (V, E)$, we remove one edge, namely E_{kl} from G to become $G_1 = (V, E_1)$. The similarity measure $\mathrm{ges}(G, G_1)$ is different from edit distance that edges give different weights. In other words, some edges are more important than others, according to their roles in the graph's connectivity.

Theorem 3: The similarity function $\mathrm{ges}(G, G_1)$, as well as the kernel for the new graph, are all dependent on (functions of) the edge.

Proof: We demonstrate by graphs with one edge difference. We show that the similarity measure $\mathrm{ges}(G, G_1)$ and K_1 are dependent on the edge, as opposed to edit distance, which is independent of edge, which was removed.

Let $e \in \mathbb{R}^n$ denote a vector with $e_k = 1$ and $e_l = -1$, where the remaining coordinates are 0. From the way graph Laplacians are constructed, we have

$$L = L_1 + e \cdot e^T.$$

We prove here the results for connected graphs. Results for nonconnected graphs can be considered by decomposing the graph into the connected subgraphs and merging the results. Once the graph G is connected, its Laplacian L is of rank $n-1$. Note that the null space of L is spanned by the vector $o = \{1, \cdots 1\}^T \in \mathbb{R}^n$, meaning that any vector $x \in \mathbb{R}^n$ that $x^T o = 0$ is in the range of L (for any graph L only).

Denote: $J \in \mathbb{R}^{n\times n}$ is a matrix of all 1 as in [14], then $J = oo^T$. Therefore

$$(I - LL^{-1})(-e) = \frac{1}{n}J(-e) = -\frac{1}{n}oo^T e = 0.$$

This is the vector denoted as u in [18].

From [18] (theorem 5), we want to find the update of pseudoinverse[1] of L_1.

Denote $\beta = 1 - e^T L^{-1} e$. Then, ignoring the component with u in this brief.

Case 1: $\beta \neq 0$

$$\begin{aligned}L_1^{-1} &= (L - e \times e^T)^{-1}\\ &= L^{-1} + \frac{L^{-1}ee^T L^{-1}}{1 - e^T L^{-1} e}.\end{aligned}$$

Hence, the amount of update from the inverse of the original graph Laplacian is

$$\frac{L^{-1}ee^T L^{-1}}{1 - e^T L^{-1} e}.$$

The denominator β means the ratio of the number of spanning trees in the graph that pass through e to the total number of spanning trees. $\beta \neq 0$ means that removing the edge e does not disconnect the graph. In other words, e is not a bridge. $\beta = 0$ means e is a bridge. $1 - \beta$ is also known as the resistance distance in electronic circuit or chemistry [17], or Euclidean commute time distance in social networks [11].

Case 2: $\beta = 0$, using theorem 6 in [18]. Let $q = |L^{-1}e|$

$$\begin{aligned}L_1^{-1} &= (L - e \times e^T)^{-1}\\ &= L^{-1} - \frac{1}{q}L^{-1}e(L^{-1}e)^T L^{-1} - \frac{1}{q}L^{-1}(e^T L^{-1})^T e^T L^{-1}\\ &\quad + \frac{1}{q^2}((L^{-1}e)^T L^{-1}(e^T L^{-1})^T)L^{-1}ee^T L^{-1}\\ &= L^{-1} - L^{-1}\left(\frac{1}{q}ee^T L^{-1} + \frac{1}{q}L^{-1}ee^T - \frac{e^T L^{-3} e}{q^2}ee^T\right)L^{-1}.\end{aligned}$$

Hence, the amount of update from the inverse of the original graph Laplacian is

$$-\frac{1}{q}L^{-1}\left(ee^T L^{-1} + L^{-1}ee^T - \frac{e^T L^{-3} e}{q}ee^T\right)L^{-1}.$$

■

Both cases prove that, the similarity between the original and a new graph (by removing one edge) is a function depending on the edge itself. This is totally different from the edit distance in which the importance of an edge in graph connectivity is not detected.

V. Experiment

As in Section II, the problem with graph cuts is that the loss functions only concern two criteria: the total amount of weight cut and the balance of the clusters. One of the purposes of graph cut-based clustering is to have highly connected clusters; it makes sense to require all nodes in the same cluster to connect to each other. However, this condition is not reflected in the loss functions in (1).

We show by simulation that our similarity measure or distance can reflect whether the clusters are connected or not among clustering solutions of the same loss function. It is the case that graph cut algorithms may find many possible optimal solutions with respect to the loss function but, intuitively, some solutions make more sense than others. The simulation graph is shown in Fig. 1 (the first graph). The graph is designed based on the result of the previous sections.

The simulation shows different graph cuts, which are all optimal [for ratio cut loss function, an instantiation of (1)]. The graph is cut by red slides, producing disconnected clusters, resulting in graph G' (the second graph). Graph is cut by blue dotted slides, producing connected clusters, resulting in graph G^* (the third graph).

According to the previous section, they are equally optimal cuts, so as the cuts by any single red slide. This makes it impossible for the loss function to determine whether the cut results in disconnected clusters or not.

[1] x^{-1} also denotes the pseudoinverse of a matrix x as well, where appropriate.

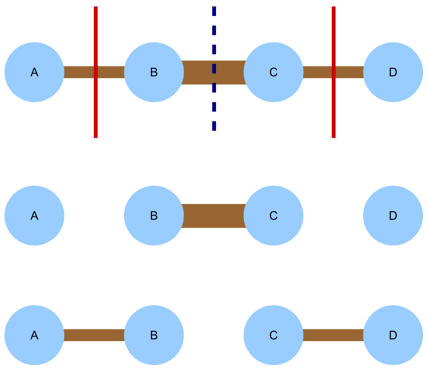

Fig. 1. Simulation graph: each round-shaped area A, B, C, or D is a clique with all edges of weight 1. All the edges between cliques: A-B, C-B with weight w, B-C with weight $2w$. The second graph is the resulting graph cut by red slides (G'). The last graph is the one cut by the blue dotted slide (G^*).

TABLE I
COMPARING DISTANCES d_1 VERSUS d_2 AND SIMILARITY VALUES s_1 VERSUS s_2. IT SHOWS THAT GRAPH G^* IS USUALLY MORE SIMILAR TO G THAN G' IS

w	d_1	d_2	s_1	s_2
0.0500	9.2797	8.1652	3.5524	22.9934
0.1000	4.6404	4.0835	1.3834	6.2337
0.1500	3.0949	2.7243	0.9309	3.0717
0.2000	2.3233	2.0460	0.7424	1.9290
0.2500	1.8618	1.6401	0.6360	1.3765
0.3000	1.5556	1.3704	0.5652	1.0602

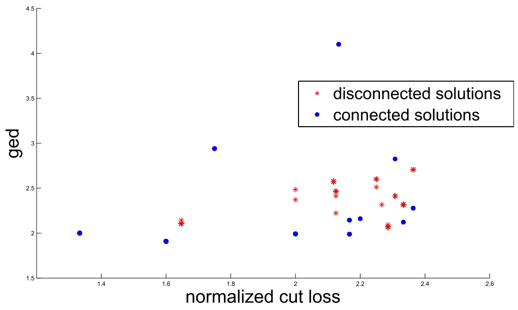

Fig. 2. Normalized cut losses versus the ged distances of many clustering solutions with small losses.

We compare distances with the original graph: $d_1 = \text{ged}(G, G')$ and $d_2 = \text{ged}(G, G^*)$. We also show the similarities between the new graphs to the original one: $s_1 = \text{ges}(G, G')$ and $s_2 = \text{ges}(G, G^*)$. The similarities and distances are shown in Table I.

We observed that the distance of G' to G is always greater than that of G^* to G, i.e., $d_1 > d_2$. Likewise, the similarity of G' to G is always smaller than that of G^* to G, i.e., $s_1 < s_2$. Hence, graphs G^* are consistently more similar to G than G' to G. It means that the graphs with disconnected clusters are farther than the graphs with connected clusters. This shows that as the graphs G^* have connected clusters, they tend to be more similar to the original graph G. We conclude that our similarity measure and its induced distance can detect the good clustering solutions while the loss functions in (1) cannot.

We also show the effectiveness of the ged distance on a real network of Zachary's karate club[2] in Fig. 2. We show the distance against the normalized cut loss function for different clustering solutions having the loss functions close to the optimal one. We found that for all the solutions having a similar graph cut loss's values, the ones with the smallest ged values are usually the ones with connected clusters. This means that, for solutions with loss functions close to optimal ones, we can also rely on the distance to select connected solutions. This result extends the optimal solutions selection in the above simulation to a real world case.

VI. CONCLUSION

We studied the problem of clustering on graphs by graph cuts. We showed the conditions that make the clusters not well connected. We proposed to detect these cases by comparing the resulting graphs after cut with the original graph using a graph similarity measure considering global graph connectivity. We proposed the graph embedding-based graph comparison that first embeds graphs into Euclidean spaces and then compares the embeddings in the spaces. We showed that the comparison does consider the global connectivities of graphs, as opposed to edit distance. It was confirmed by our experiments that the similarity and distance measure can consistently distinguish between the clustering solutions with disconnected clusters and the ones without disconnected clusters. Studying more on the relationship between the types of global connectivity and the ways of embedding would be interesting future work. Our methods also can be applied to other scenarios such as comparing biological networks for phylogenetic tree construction.

[2] www-personal.umich.edu/∼mejn/netdata/

REFERENCES

[1] A. Banerjee, S. Merugu, I. S. Dhillon, and J. Ghosh, "Clustering with Bregman divergences," *J. Mach. Learn. Res.*, vol. 6, pp. 1705–1749, Dec. 2005.

[2] Y. Boykov and O. Veksler, *Graph Cuts in Vision and Graphics: Theories and Applications*. New York, NY, USA: Springer-Verlag, 2006.

[3] Y. Boykov, O. Veksler, and R. Zabih, "Fast approximate energy minimization via graph cuts," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 23, no. 11, pp. 1222–1239, Nov. 2001.

[4] T. Bühler and M. Hein, "Spectral clustering based on the graph p-Laplacian," in *Proc. 26th Annu. ICML*, 2009, pp. 81–88.

[5] H. Bunke and K. Shearer, "A graph distance metric based on the maximal common subgraph," *Pattern Recognit. Lett.*, vol. 19, nos. 3–4, pp. 255–259, Mar. 1998.

[6] H. Bunke and G. Allerman, "A metric on graphs for structural pattern recognition," in *Proc. 2nd EUSIPCO*, 1983, pp. 257–260.

[7] F. R. K. Chung, *Spectral Graph Theory*. Providence, RI, USA: AMS, 1994.

[8] P. A. V. de Miranda, A. X. Falcão, and J. K. Udupa, "Synergistic arc-weight estimation for interactive image segmentation using graphs," *J. Comput. Vis. Image Understand.*, vol. 114, no. 1, pp. 85–99, Jan. 2010.

[9] I. S. Dhillon, Y. Guan, and B. Kulis, "Kernel k-means: Spectral clustering and normalized cuts," in *Proc. 10th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2004, pp. 551–556.

[10] C. Ding, X. He, H. Zha, M. Gu, and H. D. Simon, "A minmaxcut spectral method for data clustering and graph partitioning," Lawrence Berkeley Nat. Lab., Berkeley, CA, USA, Tech. Rep. 54111, 2003.

[11] F. Fouss, A. Pirotte, J.-M. Renders, and M. Saerens, "Random-walk computation of similarities between nodes of a graph with application to collaborative recommendation," *IEEE Trans. Knowl. Data Eng.*, vol. 19, no. 3, pp. 355–369, Mar. 2007.

[12] T. Gärtner, P. A. Flach, and S. Wrobel, "On graph kernels: Hardness results and efficient alternatives," in *Proc. 16th Annu. Conf. Comput. Learn. Theory 7th Kernel Workshop*, 2003, pp. 129–143.

[13] A. Gretton, O. Bousquet, A. J. Smola, and B. Schölkopf, "Measuring statistical dependence with Hilbert-Schmidt norms," in *Proc. 16th Int. Conf. Algorithmic Learn. Theory*, 2005, pp. 63–77.

[14] I. Gutman and W. Xiao, "Generalized inverse of the Laplacian matrix and some applications," *Bull., Classe des Sci. Math. et Naturalles*, vol. 129, no. 29, pp. 15–23, 2004.

[15] L. W. Hagen and A. B. Kahng, "Fast spectral methods for ratio cut partitioning and clustering," in *Proc. IEEE/ACM Int. Conf. Comput. Aided Design*, Nov. 1991, pp. 10–13.

[16] H. Kashima, K. Tsuda, and A. Inokuchi, "Marginalized kernels between labeled graphs," in *Proc. 20th ICML*, 2003, pp. 321–328.

[17] D. J. Klein and M. Randic, "Resistance distance," *J. Math. Chem.*, vol. 12, no. 1, pp. 81–95, 1993.

[18] C. D. Meyer, "Generalized inversion of modified matrices," *SIAM J. Appl. Math.*, vol. 24, pp. 315–323, May 1973.

[19] P. A. Miranda and A. X. Falcão, "Links between image segmentation based on optimum-path forest and minimum cut in graph," *J. Math. Imag. Vis.*, vol. 35, no. 2, pp. 128–142, Oct. 2009.

[20] B. Nadler and M. Galun, "Fundamental limitations of spectral clustering," in *Proc. Adv. NIPS*, 2007, pp. 1017–1024.

[21] J. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 8, pp. 888–905, Aug. 2000.

[22] A. J. Smola and R. I. Kondor, "Kernels and regularization on graphs," in *Proc. 18th Annu. Conf. Learn. Theory*, 2003, pp. 144–158.

[23] N. Wicker, C. H. Nguyen, and H. Mamitsuka, "A new dissimilarity measure for labeled graphs," *Linear Algebra Appl.*, vol. 438, no. 5, pp. 2331—2338, Mar. 2013.

[24] R. C. Wilson, E. R. Hancock, and B. Luo, "Pattern vectors from algebraic graph theory," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 7, pp. 1112–1124, Jul. 2005.