Internet Mathematics Vol. 9, No. 1: 82–112

On Clustering on Graphs with Multiple Edge Types

Matthew Rocklin and Ali Pinar

Abstract. We study clustering on graphs with multiple edge types. Our main motivation is that similarities between objects can be measured by many different metrics. For instance, similarity between two papers can be based on common authors, where they were published, keyword similarity, citations, etc. As such, graphs with multiple edges give a more accurate model to describe similarities between objects than models using single-edge graphs. Each edge/metric provides only partial information about the data; recovering full information requires aggregation of all the similarity metrics. Clustering becomes much more challenging in this context, since in addition to the difficulties of the traditional clustering problem, we have to deal with a space of clusterings. Reducing the multidimensional space into a single dimension poses significant challenges. At the same time, the multidimensional space can contain latent structures, and searching this multidimensional space can reveal important information about the graph. We generalize the concept of clustering in single-edge graphs to multiedged graphs and investigate problems such as the following: Can we find a clustering that remains good, even if we change the relative weights of metrics? How can we describe the space of clusterings efficiently? Can we find unexpected clusterings (a good clustering that is distant from all given clusterings)? If we are given the ground-truth clustering, can we recover how the weights for edge types were aggregated?

ISSN: 1542-7951 print

1. Introduction

A community or a cluster in a graph is a subset of vertices that are tightly coupled among themselves and loosely coupled with the rest of the graph. Finding these communities is one of the fundamental problems of graph analysis and has been the subject of numerous research efforts. Most of these efforts begin with the premise that a simple graph has already been constructed. A relation between two objects is represented by an edge between two nodes, which may be weighted by the strength of the connection or left as a binary variable. This paper studies the community-detection problem on graphs with multiple edge types or multiple similarity metrics, as opposed to traditional graphs with a single edge type. We will discuss the challenges in reducing the multidimensional space into a single dimension (e.g., finding aggregate edge weights), as well as the challenges in searching the multidimensional space (e.g., finding latent structures and clusterings that are different from what one would expect).

In many real-world problems, similarities between objects can be defined by many different relationships. For instance, similarity between two scientific articles can be defined based on authorship, citations to, citations from, keywords, titles, where they were published, text similarity, and many more. Relationships between people can be based on the nature of the relationship (e.g., business, family, friendship) or the means of communication (e.g., email, phone, in person), etc. Electronic files can be grouped by their type (Latex, C, html), names, the time they were created, or the pattern in which they are accessed. In these examples, there are multiple graphs that define relationships between the subjects. We may choose to reduce this multivariate information to construct a single composite graph. This is convenient, for it enables application of many strong results from the literature. However, information being lost during this aggregation may be crucial, and we believe that graphs with multiple edge types provide a more precise representation of the problem, and thus will lead to more accurate analyses. Despite its importance, the literature on clustering graphs with multiple edge types is very sparse. In [Mucha et al. 10], the authors looked at community detection when multiple edge types are sampled in time and are strongly correlated. This problem is described in [Dunlavy et al. 06] as a three-dimensional tensor, and the authors used a PARAFAC decomposition (SVD generalization) to identify dominant factors.

The community-detection problem on graphs with multiple edge types yields many interesting questions:

- If a ground-truth clustering is known, can we recover an aggregation scheme that best resonates with the ground-truth data?

- How can we efficiently represent a space of clusterings spanned by the multiple edge types.
- Are the clusterings within this space clustered themselves?
- How do we find significantly different clusterings for the same data?

These questions add another level of complexity to the already difficult problem of community detection in graphs. As in the single-edge case, the challenges lie not only in algorithms, but also in formulations of these problems. In this paper, we investigate these problems and propose some solutions.

Our techniques rely on using optimization (specifically, methods that rely only on function values), methods for classical community detection (i.e., community detection with single edge types), and metrics for quantifying the distance between two clusterings. We present results of three case studies, on (1) file system data, where files are grouped to projects; (2) papers submitted to arXiv; and (3) countries, where the data are based on the *CIA World Factbook* [CIA 06].

The rest of the paper is organized as follows. In the next section, we will review some of the background information for this paper. In particular, we present the *variation-of-information* metric [Meila 03], which we use to quantify the distance between two clusterings. In Section 3, we study the problem of computing an aggregate similarity measure for a given ground-truth clustering. Given a graph with multiple edge types and the ground-truth clustering, can we find an aggregation scheme to reduce the information from multiple edge types into a single metric such that this metric best resonates with the ground-truth data? We apply our methods to synthetic and file-system data. Section 4 addresses the latent clustering structure on graphs with multiple edge types and introduces the concept of metaclustering. We also discuss how we can represent the metaclustering structure efficiently, and provide results on arXiv data. In Section 5 we discuss how we can look for unexpected clusters and how we can improve the significance of clusters for multiple edge types. We summarize our results in Section 6.

2. Background

A weighted graph is represented as a tuple $G = (V, E)$, where V is a set of vertices and E is a set of edges. Each edge e_i is a tuple $e_i = \{v_a, v_b, w_i \mid v_a, v_b \in V,\ w_i \in \mathbb{R}\}$ representing a connection between vertices v_a and v_b with weight w_i. In this work we replace $w_i \in \mathbb{R}$ with $\vec{w_i} \in \mathbb{R}^k$, with k the number of edge types. We will refer to such graphs as *graphs with multiple edge types* or *multiweighted* *graphs.* We will construct functions that map multiweighted edges $\vec{w_i} \in \mathbb{R}^k$ to *composite edge types* $f(\vec{w_i}) = \omega_i \in \mathbb{R}$. For much of this paper, f will be linear: $\omega_i = f(\vec{w_i}) = \sum \alpha_i w_i$.

2.1. Clustering

Intuitively, the goal of clustering is to break down the graph into smaller groups such that vertices in each group are tightly coupled among themselves and loosely coupled with the remainder of the graph. Both the translation of this intuition into a well-defined mathematical formula and design of associated algorithms pose significant challenges. Despite the high quality and the high volume of the literature, the area continues to draw considerable interest due to the growing importance of the problem and the challenges posed by the size and mathematical variety of the subject graphs. For further information on clustering, see [Lancichinetti and Fortunato 09].

Our goal is to extend the concept of clustering to graphs with multiple edge types without getting into the details of clustering algorithms and formulations, since such a detailed study would be well beyond the scope of this paper. In this paper, we used the Graclus software [Dhillon et al. 07], which uses a top-down approach that recursively splits the graph into smaller pieces, and FastCommunity [Clauset et al. 04], which uses an agglomerative approach that optimizes the modularity metric.

2.2. Variation of Information of Clusterings

At the core of most of our discussions will be the notion of similarity between two clusterings, which calls for a method to quantify the distance between two clusterings. Several metrics and methods have been proposed for comparing clusterings, such as *variation of information* [Meila 03], *scaled coverage measure* [Stichting et al. 00], *classification error* [Lange et al. 04, Luo 05, Meila 03], and *Mirkin's metric* [Mirkin 96]. Out of these, we have used the variation-of-information metric in our experiments.

Let $\mathbf{C}_0 = \left\langle C_0^1, C_0^2, \ldots, C_0^K \right\rangle$ and $\mathbf{C}_1 = \left\langle C_1^1, C_1^2, \ldots, C_1^K \right\rangle$ be two clusterings of the same node set. Here we use boldface $\mathbf{C}_i$ to represent the ith clustering and C_i^j to represent the jth cluster in that clustering. Let n be the total number of nodes, and $P(\mathbf{C}, k) = |C^k|/n$ the probability that a random node is in cluster C^k in a clustering $\mathbf{C}$. Similarly, the probability that a random node is in cluster C^k in clustering $\mathbf{C}_i$ and in cluster C^l in clustering $\mathbf{C}_j$ is $P(\mathbf{C}_i, \mathbf{C}_j, k, l) = |C_i^k \cap C_j^l|/n$. The *entropy of information*, or expectation value of learned information, in $\mathbf{C}_i$

is defined as

$$H(\mathbf{C}_i) = -\sum_{k=1}^{K} P(\mathbf{C}_i, k) \log P(\mathbf{C}_i, k),$$

where K_i is the number of clusters in $\mathbf{C}_i$. The mutual information shared by $\mathbf{C}_i$ and $\mathbf{C}_j$ is

$$I(\mathbf{C}_i, \mathbf{C}_j) = \sum_{k=1}^{K_i} \sum_{l=1}^{K_j} P(\mathbf{C}_i, \mathbf{C}_j, k, l) \log P(\mathbf{C}_i, \mathbf{C}_j, k, l).$$

Given these two quantities, the variation-of-information metric is defined in [Meila 03] by

$$d_{\mathrm{VI}}(\mathbf{C}_i, \mathbf{C}_j) = H(\mathbf{C}_i) + H(\mathbf{C}_j) - 2I(\mathbf{C}_i, \mathbf{C}_j), \tag{2.1}$$

where the intuition behind this metric is explained as follows: $H(\mathbf{C}_i)$ denotes the average uncertainty of the position of a node in clustering $\mathbf{C}_i$. If, however, we are given $\mathbf{C}_j$, then $I(\mathbf{C}_i, \mathbf{C}_j)$ denotes the average reduction in uncertainty of where a node is located in $\mathbf{C}_i$. If we rewrite (2.1) as

$$d_{\mathrm{VI}}(\mathbf{C}_i, \mathbf{C}_j) = (H(\mathbf{C}_i) - I(\mathbf{C}_i, \mathbf{C}_j)) + (H(\mathbf{C}_j) - I(\mathbf{C}_i, \mathbf{C}_j)),$$

then the first term measures the information lost if $\mathbf{C}_j$ is the true clustering and we know instead $\mathbf{C}_i$, and the second term is the opposite. Note that $d_{\mathrm{VI}}(\mathbf{C}_i, \mathbf{C}_j)$ will be zero when the two clusterings are the same, and it will be maximal when the two clusterings are independent. The variation-of-information metric can be computed in $O(n)$ time.

3. Recovering a Graph Given a Ground-Truth Clustering

Suppose we are given a ground-truth clustering for a graph with multiple edge types/similarity metrics. Can we recover an aggregation scheme that best resonates with the ground-truth data? Such an aggregation scheme that reduces multiple similarity measurements into a single similarity measurement can be a crucial enabler that reduces the problem of finding communities with multiple similarity metrics to a well-known fundamental problem in data analysis. Additionally, if we can obtain this aggregation scheme from data sets for which the ground-truth is available, we may then apply the same aggregation to other data instances in the same domain.

Formally, we work on the following problem. Given a graph $G = (V, E)$ with multiple similarity measurements for each edge $\langle w_i^1, w_i^2, \ldots, w_i^K \rangle \in \mathbb{R}^K$ and a ground-truth clustering for this graph $\mathbf{C}^*$, our goal is to find a weighting vector

$\alpha \in \mathbb{R}^K$ such that $\mathbf{C}^*$ is an optimal clustering for the graph G whose edges are weighted as $w_i = \sum_{j=1}^{K} \alpha_j w_i^j$. Note that this is only a semiformal definition, since we have not formally defined what we mean by an *optimal clustering.* In addition to the well-known difficulty of defining what a good clustering means, matching a graph to a ground-truth clustering has specific challenges, which we discuss in the following section.

Below, we describe two approaches. The first approach is based on inverse problems, and we try to find weighting parameters for which the clustering on the graph yields the ground-truth clustering. The second approach computes weighting parameters that maximize the quality of the ground-truth clustering.

3.1. Solving an Inverse Problem

Inverse problems arise in many scientific computing applications in which the goal is to infer unobservable parameters from finite observations. Solutions typically involve iterations of predictions and solution of "forward" problems to compute the accuracy of the prediction.

Our problem can be considered an inverse problem, since we are trying to compute an aggregation function from a given clustering. The forward problem in this case is the clustering operation. We can start with a random guess for the edge weights, cluster the new graph, and use the distance between two clusterings as a measure of the quality of the guess. We can further put this process within an optimization loop to find the parameters that yield the closest clustering to the ground-truth.

The disadvantage of this method is that it relies on the accuracy of the forward solution, i.e., the clustering algorithm. If we are given the true solution to the problem, can we construct the same clustering? This will not be easy for two reasons. First, there is no perfect clustering algorithm, and second, even if we were able to solve the clustering problem optimally, we would not have the exact objective function for clustering. Also, solving many clustering problems will be time-consuming, especially for large graphs.

3.2. Maximizing the Quality of Ground-Truth Clustering

An alternative approach is to find an aggregation function that maximizes the quality of the ground-truth clustering. For this purpose, we have to take into account not only the overall quality of the clustering, but also the placement of individual vertices, since the individual vertices represent local optimality. For instance, if the quality of the clustering will improve by moving a vertex to a cluster other than its ground-truth, then the current solution cannot be ideal.

While it is fair to assume that some vertices might have been misclassified in the ground-truth data, there should be a penalty for such vertices. Thus we have two objectives while computing α: (i) justifying the location of each vertex and (ii) maximizing the overall quality of the clustering.

3.2.1. Justifying Locations of Individual Vertices.

For each vertex $v \in V$, we define the *pull* to each cluster C^k in $\mathbf{C} = \left\langle C^1, C^2, \ldots C^K \right\rangle$ to be the cumulative weights of edges between v and its neighbors in C^k:

$$P_\alpha(v, C^k) = \sum_{\substack{w_i = (u,v) \in E \\ u \in C^k}} w_i(\alpha). \tag{3.1}$$

We further define the *holding power* $H_\alpha(v)$ for each vertex to be the pull of the cluster to which the vertex belongs in C^* minus the next-largest pull among the remaining clusters. If this number is positive, then v is held more strongly to the proper cluster than to any other. We can then maximize the number of vertices with positive holding power by maximizing $|\{v : H_\alpha(v) > 0\}|$. What is important here is the concept of pull and hold, since the specific definitions may be changed without altering the core idea.

While this method is local and easy to compute, its discrete nature limits the tools that can be used to solve the associated optimization problem. Because gradient information is not available, it hinders our ability to navigate in the search space. In our experiments, we smoothed the steplike nature of the function $H(v) > 0$ by replacing it with $\arctan(\beta \cdot H_\alpha(v))$. This functional form still encodes that we want the holding power to be positive for each node, but it allows the optimization routine to benefit from small improvements. It emphasizes nodes that are close to the $H(v) = 0$ crossing point (large gradients) over nodes that are well entrenched (low gradients near extremes). The parameter β determines how small a holding value must be to be considered of high priority.

This objective function sacrifices holding scores for nodes that are safely entrenched in their cluster (high holding power) or are lost causes (very low holding power) for those that are near the crossover point. The extent to which it does this can be tuned by the steepness parameter of the arctangent. For very steep parameters, this function resembles classification (step function), while for very shallow parameters, it resembles a simple linear sum, as seen in Figure 1. We can solve the following optimization problem to maximize the number of vertices whose positions in the ground-truth clustering are justified by the weighting vector α:

$$\arg\max_{\alpha \in \mathbb{R}^K} \sum_{v \in V} \arctan(H_\alpha(v)). \tag{3.2}$$

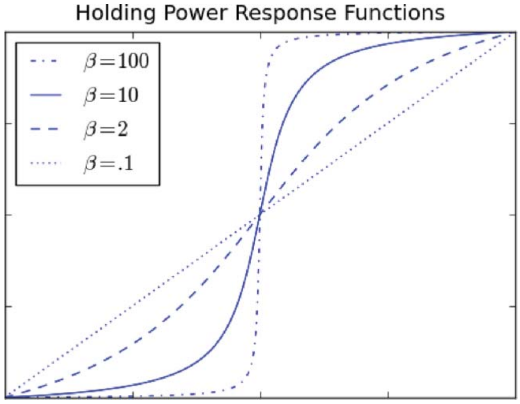


Figure 1. Arctangent provides a smooth blend between step and linear functions (color figure available online).

3.2.2. Overall Clustering Quality. In addition to individual vertices being justified, overall quality of the clustering should be maximized as well. Any good metric can potentially be used for this purpose. However, we find that some strictly linear functions have a trivial solution. Consider an objective function that measures the quality of a clustering as the sum of the intercluster edges. To minimize the cumulative weights of cut edges, or equivalently to maximize the cumulative weights of internal edges, we solve

$$\min_{|\alpha|=1} \sum_{e_j \in \text{cut}} \sum_{i=1}^{K} \alpha_i w_j^i,$$

where cut denotes the set of edges whose endpoints are in different clusters.

Let S^k denote the sum of the cut edges with respect to the kth metric. That is, $S^k = \sum_{e_j \in \text{cut}} w_j^k$. Then the objective function can be rewritten as $\min_\alpha \sum_1^K \alpha_k S^k$. Because this is linear, it has a trivial solution that assigns 1 to the weight of the maximum S^k, which means that only one similarity metric is taken into account. While additional constraints may exclude this specific solution, a linear formulation of the quality will always yield only a trivial solution within the new feasible region.

In our experiments we used the *modularity metric* [Newman 06]. The modularity metric uses a random graph generated with respect to the degree distribution as the null hypothesis, setting the modularity score of a random clustering to 0. Formally, the modularity score for an unweighted graph is

$$\frac{1}{4m} \sum_{ij} \left[e_{ij} - \frac{d_i d_j}{2m} \right] \delta_{ij}, \tag{3.3}$$

where e_{ij} is a binary variable that is 1 if and only if vertices v_i and v_j are connected; d_i denotes the degree of vertex i, m is the number of edges, and $\delta_{i,j}$ is a binary variable that is 1 if and only if vertices v_i and v_j are on the same cluster. In this formulation, $d_i d_j/2m$ corresponds to the expected number of edges between vertices v_i and v_j in a random graph with the given expected degree distribution, and its subtraction corresponds to the null hypothesis. The graph model used in [Chung and Lu 02a, Chung and Lu 02b, Aiello et al. 01] as well as the edge-configuration model of [Newman et al. 02] are based on the same core idea.

This formulation can be generalized for weighted graphs by redefining e_{ij} as the weight of this edge (0 if no such edge exists), d_i as the cumulative weight of edges incident to v_i, and m as the cumulative weight of all edges in the graph [Newman 04].

3.2.3. Solving Optimization Problems. We have presented several nonlinear optimization problems for which the derivative information is not available. To solve these problems, we used HOPSPACK (Hybrid Optimization Parallel Search PACKage) [Plantenga 09], which was developed at Sandia National Laboratories to solve linear and nonlinear optimization problems when the derivatives are not available.

3.3. Experimental Results

3.3.1. Recovering Edge Weights. The goal of this set of experiments is to see whether we can find aggregation functions that justify a given clustering. We have performed our experiments on three data sets.

Synthetic data. Our generation method is based on [Lancichinetti et al. 08], in which the authors propose a method to generate graphs as benchmarks for clustering algorithms. We generated graphs of sizes 500, 1000, 2000, and 4000 nodes, 30 edges per node on average, mixing parameters $\mu_t = 0.7$, $\mu_w = 0.75$, and known communities. We then perturbed edge weights w_i with additive and multiplicative noise so that $w_i \leftarrow \nu(w_i + \sigma) : \sigma \in (-2w_a, 2w_a), \nu \in (0,1)$ uniformly, independently, and identically distributed, where w_a is the average edge weight.

After the noise, none of the original metrics preserved the original clustering structure. We display this in Figure 2, which presents histogram information for the holding power for vertices. For this figure, holding powers of vertices were grouped into 20 bins. The figure displays the centers of the bins (horizontal axis) vs. the number of vertices in the bin (vertical axis). The red curve (circles)

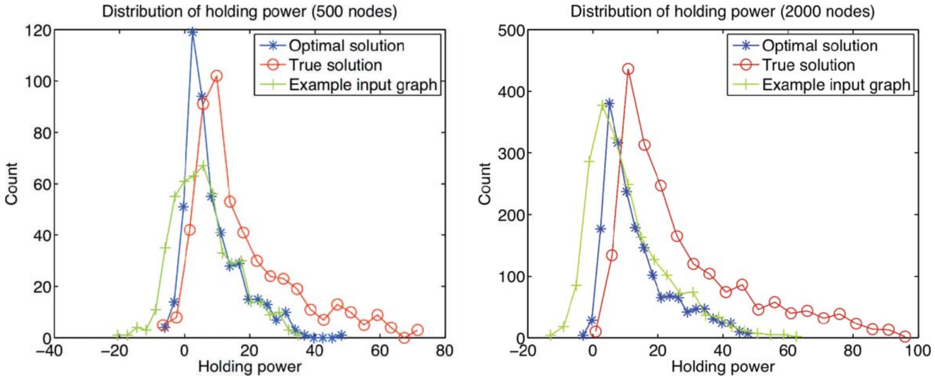


Figure 2. Three histograms of holding powers for (blue stars) an example perturbed (poor) edge type; (green pluses) the original data (very good); (red circles) the optimal blend of ten of the perturbed edge types (color figure available online).

corresponds to vertices of the original graph, which all have positive holding power. The green curve (pluses) corresponds to holding powers after noise is added. We present only one edge type for clarity of presentation. As can be seen, a significant portion (30%) of the vertices have negative holding power, which means that these vertices have a stronger bound to another cluster than to their own. The blue curve (stars) shows the holding powers after we compute an optimal linear aggregation. As seen in the figure, almost all vertices move to the positive side, justifying the ground-truth clustering. A few vertices with negative holding power are expected, even after an optimal solution due to the noise. These results show that a composite similarity that resonates with a given clustering can be computed out of many metrics, none of which gives a good solution by itself.

In Table 1, we present these results on graphs with different numbers of vertices. While the percentages change for different numbers of vertices, our main

Number of nodes	Number of clusters	Ground-truth	Optimized	Perturbed (average)
500	14	0.965	0.922	0.703
1000	27	0.999	0.977	0.799
2000	58	0.999	0.996	0.846
4000	118	1.00	0.997	0.860

Table 1. Fraction of nodes with positive holding power for ground-truth, perturbed, and optimized graphs.

conclusion that a good clustering can be achieved via a better aggregation function remains valid.

File system data. An owner of a workstation classified 300 of his files as belonging to one of his three ongoing projects, which we took as the ground-truth clustering. We used filename similarity, time-of-modification/time-of-creation similarity, ancestry (distance in the directory tree), and parenthood (edges between a directory node with file nodes in this directory) as the similarity metrics among these files.

Our results showed that only three metrics (time-of-modification, ancestry, and parenthood) affected the clustering. However, the solutions were sensitive to the choice of the arctangent parameter. In Figure 3, each column corresponds to an optimal solution for the corresponding arctangent parameter. Recall that higher values of the arctangent parameter correspond to sharper step functions. Hence, the right side of the figure corresponds to maximizing the total number of vertices with positive holding power, while the left side corresponds to maximizing the sum of holding powers. The difference between the two is that the solutions on the right side may have many nodes with barely positive values, while those on the left may have nodes farther away from zero at the cost of more nodes with negative holding power. This is expected in general, but drastic change in the optimal solutions as we go from one extreme to another was surprising to us, and should be taken into account in further studies.

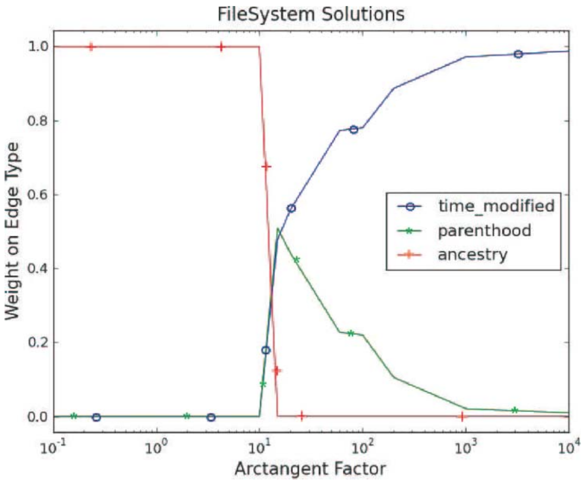


Figure 3. Optimal solutions for the file system data for different arctangent parameters (color figure available online).

arXiv data. We took 30 000 high-energy physics articles published on the website arXiv.org and considered abstract text similarity, title similarity, citation links, and shared authors as edge types for these articles. We used the top-level domain of the submitter's e-mail (.edu, .uk, .jp, and so on) as a proxy for the region where the work was done. We used these regions as the ground-truth clustering.

The best parameters that explained the ground-truth clustering were 0.0 for abstracts, 1.0 for authors, 0.059 for citations, and 0.0016 for titles. This means that the shared-authors edge type is almost entirely favored, with cross-citations coming in a distant second. This is intuitively clear, because a graph of articles linked by common authors will be linked both by topic (we work with people in our field) but also by geography (we often work with people in nearby institutions), whereas edge types such as abstract text similarity tend to encode only the topic of a paper, which is less geographically correlated. Different groups can work on the same topic, and it was good to see that citations factored in, and such a clear dominance of the authors information was noteworthy. In future work, we plan to investigate nonlinear aggregation functions on this graph.

3.3.2. Clustering Quality vs. Holding Vertices.

We have stated two goals in computing an aggregation function: justifying the position of each vertex and the overall quality of clustering. In Figure 4, we present the Pareto frontier for the two objectives. The vertical axis represents the quality of the clustering with respect to the modularity metric [Newman 06], while the horizontal axis represents the

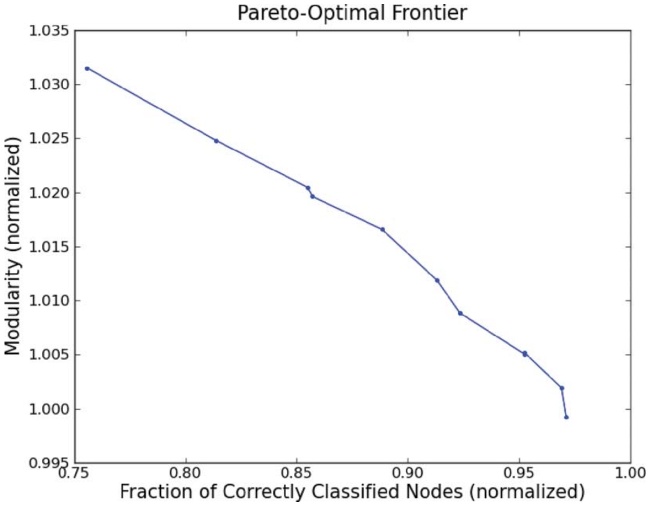


Figure 4. Pareto frontier for two objectives: normalized modularity and percentage of nodes with positive holding power (color figure available online).

percentage of nodes with positive holding power. The modularity numbers are normalized with respect to the modularity of the ground-truth clustering, and normalized numbers can be greater than 1, since the ground-truth clustering does not specifically aim at maximizing modularity.

As expected, Figure 4 shows a tradeoff between two objectives. However, the scale difference between the two axes should be noted. The full range in modularity change is limited to only 3% for modularity, while the range is more than 20% for percentage of vertices with positive holding power. More importantly, by looking only at the holding powers, we can preserve the original modularity quality. The reason for this is that we have relatively small clusters, and almost all vertices have a connection with a cluster besides their own. If we had clusters in which many vertices had all their connections within their clusters (e.g., much larger clusters), then this would not have been the case, and having a separate quality-of-clustering metric would have made sense. However, we know that most complex networks have small communities no matter how big the graphs are [Leskovec et al. 09]. Therefore, we expect that looking only at the holding powers of vertices will be sufficient to recover aggregation functions.

3.3.3. Inverse Problems vs. Maximizing Clustering Quality. We used the file system data set to investigate the relationship between the two proposed approaches, and we present our results in Figure 5. For this figure, we computed the objective function for the ground-truth clustering for various aggregation weights and used the same weights to compute clusterings with Graclus. From these clusterings we computed the variation of information (VI) distance to the ground-truth.

Figure 5 presents the correlation between the measures: VI distance for Graclus clusterings for the first approach, and the objective function values for the second approach. This tries to determine whether solutions with higher objective function values yield clusterings closer to the ground-truth using Graclus. In this figure, a horizontal line fixed at 1 would have been ideal, indicating perfect correlation. Our results show, nevertheless, a very strong correlation between the two. These results are not conclusive, since we need more experiments and other clustering tools. However, this experiment produced promising results and shows how such a study may be performed.

3.3.4. Runtime Scalability. In our final set of experiments, we show the scalability of the proposed method. First, we want to note that the number of unknowns for the optimization problem is a function of the aggregation function only and is independent of the graph size. The required number of operations for one function evaluation, on the other hand, depends linearly on the size of the graph, as illustrated in Figure 6. In this experiment, we used synthetic graphs with 30 as the average degree, and the presented numbers correspond to averages over

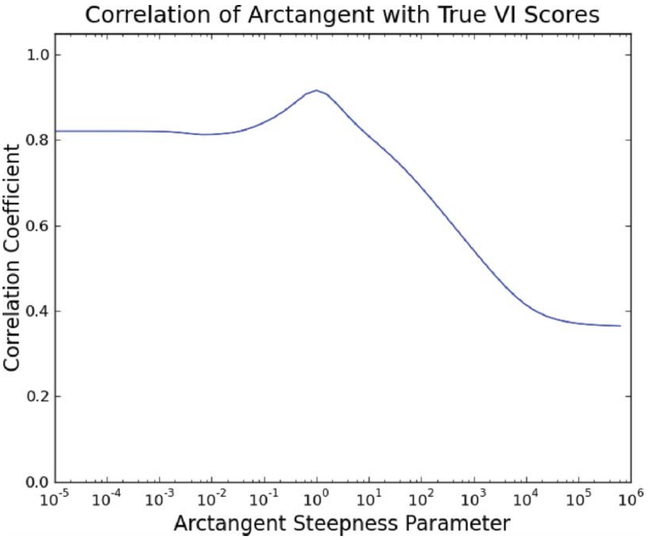


Figure 5. The correlation of the arctan-smoothed objective function with variation of information distance using clusterings generated by Graclus as we vary the steepness parameter (color figure available online).

10 different runs. As expected, the runtimes scale linearly with the number of edges.

The runtime of the optimization algorithm depends on the number of function evaluations. Since the algorithm we used is nondeterministic, the number of function evaluations, hence runtime, varies even for different runs on the same problem. We are not presenting these results in detail due to space constraints. However, we want to reemphasize that the size of the optimization problem does not grow with the graph size, and we do not expect the number of function evaluations to cause any scalability problems.

We also observed that the number of function evaluations increases linearly with the number of similarity metrics. These results are also omitted due to space constraints.

4. Finding Latent Clusters

Consider the situation in which several edge types share redundant information yet as an ensemble combine to form some broader structure. For example, scientific journal articles can be connected by text similarity, abstract similarity, keywords, shared authors, cross-citations, etc. Many of these edge types reflect

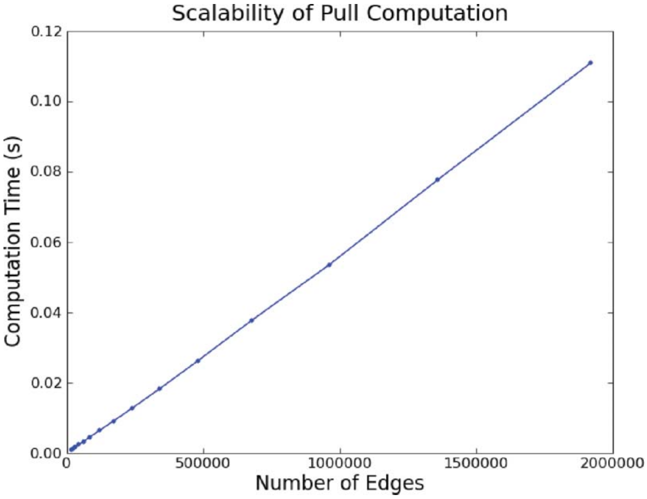


Figure 6. Scalability of the proposed method (color figure available online).

the *topic* of the document, while others are also influenced by the *location* of the work.

Text, abstract, and keyword similarity are likely to be redundant in conveying topic information (physics, math, biology), while shared authorship (two articles sharing a common author) is likely to convey both topic and location information because joint authors tend to work on similar topics and to work with those at the same or nearby institutions. We say that the topic and location attributes are *latent* because they do not exist explicitly in the data.

We can represent much of the variation in the data by two relatively independent clusterings based on the topic of documents and their location. This compression of information from five edge types to two meaningful clusterings is the goal of this paper.

4.1. An Illustrative Problem

We construct a graph with multiple edge types to demonstrate latent classes. For illustration, we assume that our graph is perfectly embedded in $\mathbb{R}^2$, as seen in Figure 7(a). In this example, each point on the plane represents a vertex, and two vertices are connected by an edge if they are close in distance. The similarity/weight for each edge is inversely proportional to the Euclidean distance. We see visually that there are nine natural clusters. More interestingly, we see that these clusters are arranged symmetrically along two axes. These clusters have more structure than the set $\{1, 2, 3, \ldots, 9\}$. Instead, they have the

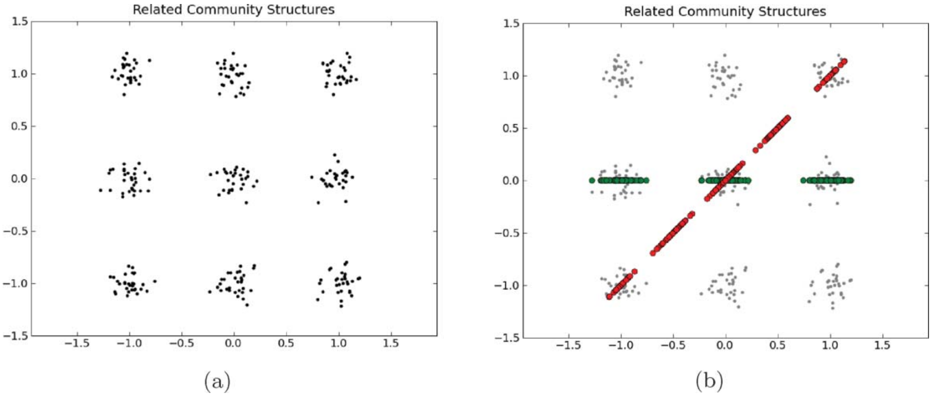


Figure 7. Illustrating clusters (a) underlying structure and (b) low-dimensional/partial views. Figure (a) shows 270 vertices arranged in nine clusters in the plane. Edges exist between vertices so that close points are well connected and distant points are poorly connected. Figure (b) shows two 1D graphs arranged to suggest their relationship to the underlying 3×3 community structure. Both have clear community structures that are related but not entirely descriptive of the underlying 3×3 communities (color figure available online).

structure $\{1, 2, 3\} \times \{1, 2, 3\}$. An example of such a structure is the separation of academic papers along two factors, {physics, mathematics, biology} and {West Coast, Midwest, East Coast}. The nine clusters (with examples such as physics articles from the West or biology articles from the Midwest) have underlying structure.

Our datasets do not directly provide this information. For instance, with journal articles we can collect information about authors, where the articles were published, and their citations. Each of these aspects provides only a partial view of the underlying structure. Analogous to our geometric example above, we could consider features of the data as projections of the points to one-dimensional subspaces. Distances/similarities between the points in a projection contain only partial information. This is depicted in Figure 7(b). For instance, the horizontal projection represents a metric that clearly distinguishes between rows but cannot differentiate between different communities on the same row. The diagonal projection, on the other hand, captures partial information about columns and partial information about rows.

Neither of the two metrics can provide the full information for the underlying data. However, considered as an ensemble, they do provide a complete picture. Our goal is to be able to tease out the latent factors of data from a given set of partial views.

In this paper, we will use this 3 × 3 example for conceptual purposes and for illustrations. Our approach is to construct many multiweighted graphs using combinations of the partial views of the data. We will cluster these graphs and analyze these clusters to recover the latent structure.

We expect that different regions of this space will have different clusterings. How drastic these differences are will depend on the particular multiweighted graph. How can we characterize this space of clusterings? Are there homogeneous regions, easily identifiable boundaries, groups of similar clusterings, etc.? We investigate the existence of a metaclustering structure. That is, we investigate whether several clusterings in this space exhibit community structure themselves. In this section, we present our methods for these questions on the 3 × 3 data. We will later provide results on a larger dataset.

4.2. Sampling the Clustering Space

To inspect the space of clusterings, we sample in a Monte Carlo fashion. We take points $\vec{\alpha_i} \in \mathbb{R}^k$ such that $|\vec{\alpha_i}| = 1$ and compute the appropriate graph and clustering at each point using the Graclus algorithm. We may then compare these clusterings using the variation of information metric.

As our first experiment, we take 16 random one-dimensional projections of the points laid out in the plane shown in Figure 7 and consider the projected pointwise distances in aggregate as a multiweighted graph. From this multiweighted graph we take 800 samples of the linear space of clusterings. These 800 clusterings approximate the clustering structure of the multiweighted graph.

The results of these experiments are presented in Figure 8(a). In this figure, each row and each column corresponds to a clustering of the graph. Entries in the matrix represent the variation-of-information distance between two clusterings. Therefore, dark regions in this matrix are sets of clusterings that are highly similar. White bands show informational independence between regions. The rows/columns of this matrix have been ordered to have more similar clusterings closer to each other so as to highlight the clusters of clusterings detected. This ordering was taken from the ordering of the leaves in a hierarchical clustering of these clusterings.

4.3. Metaclusters: Clusters of Clusterings

While it is interesting to know that significantly different clusterings can be found, the lack of stable clustering structure is not helpful for applications of clustering such as for unsupervised learning. We need to reduce this set of clusterings

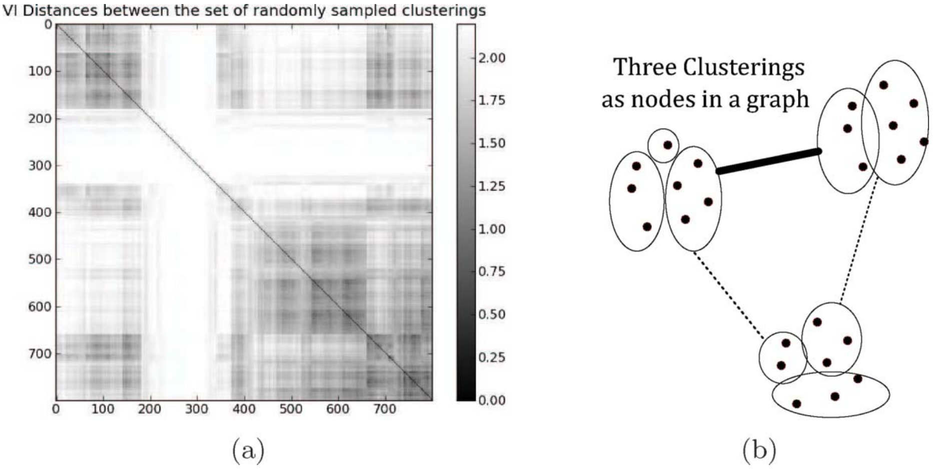


Figure 8. The metaclustering information. (a) VI distances between 800 sampled clusterings. Vertices are ordered to show optimal clustering of this graph. Dark blocks on the diagonal represent clusters. The white band is a group of completely independent clusterings. (b) Three clusterings treated as nodes in a graph. Similar clusterings (top two) are connected with high-weighted edges. Distant clusterings are connected with low-weighted edges (color figure available online).

further. We approach this problem by applying the idea of clustering to this set of clusterings. We call this problem the *metaclustering problem.*

We represent the clusterings as nodes in a graph and connect them with edge weights determined by the inverse of the variation-of-information metric [Meila 03]. We inspect this graph to see whether it contains clusters. That is, we *cluster the graph of clusterings* to see whether there exist some tightly coupled clusters of clusterings within the larger space. For instance, in Figure 8(b), the top two clusterings differ only in the position of a single vertex and thus are highly similar. In contrast, the bottom clustering is different from both and is weakly connected.

Figure 8(a) reveals the metaclustering structure in our experiments. The dark blocks around the diagonal correspond to metaclusters. We can see two big blocks in the upper left and lower right corners. Furthermore, there is a hierarchical clustering structure within these blocks, since there are smaller blocks within the larger blocks.

In this experiment, we were able to observe metaclusters. As usual, results depend on the particular problem instance. While we do not claim that one can always find such metaclusters, we expect that they will exist in many multiweighted graphs, and exploiting the metaclustering structure can enable efficient handling of this space, which is the topic of the next section.

4.4. Efficient Representation of the Clusterings

In this section, we study how to represent the metaclustering structure efficiently. First we will study how to reduce a cluster of clusterings into a single averaged or representative clustering. Then we will study how to select and order a small number of metaclusters to cover the clustering space efficiently.

4.4.1. Averaging Clusterings within a Cluster. To increase the human accessibility of this information, we reduce each cluster of clusterings into a single representative clustering. We use the Cluster-Based Similarity Partitioning Algorithm (CSPA) proposed in [Strehl and Ghosh 03] to combine several clusterings into a single average. In this algorithm, each pair of vertices is connected with an edge with weight equal to the number of clusters in which they co-occur. If v_a and v_b are in the same cluster in k of the clusterings, then in this new graph they are connected with weight k. If they are never in the same cluster, then they are not connected. We then cluster this graph and use the resultant clustering as the representative. In Figure 9, we depict the addition of three clusterings to form an average graph, which can then be clustered.

We perform this process on the clusters of clusterings found in Section 4.3 and presented in Figure 8(a) to obtain the representative-clusterings in Figure 10. We see that the product of the first two representative-clusterings identifies the original nine clusterings with little error. We see also that the two factors are identified perfectly by each of these clusterings individually.

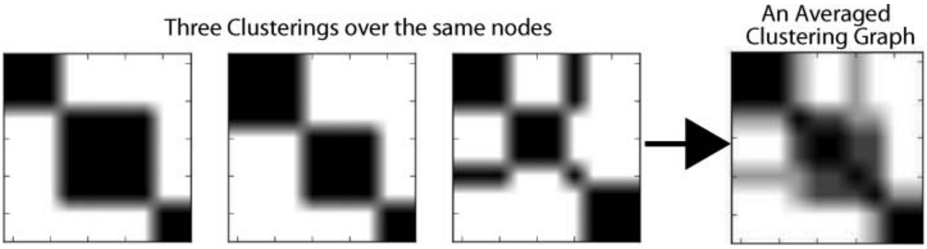

Three Clusterings over the same nodes
An Averaged Clustering Graph

Figure 9. The CSPA [Strehl and Ghosh 03] averaging procedure for clusterings. Each clustering is displayed as a block-diagonal graph (or permutation) with two nodes connected if and only if they are in the same cluster. Then an aggregate graph (right) is formed by the addition of these graphs. This graph on the right is then clustered using a traditional algorithm. This clustering is returned as the representative clustering.

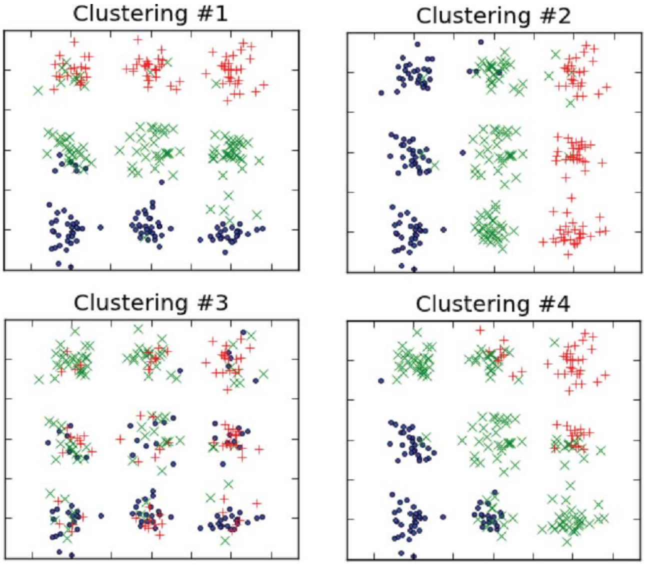


Figure 10. Representative-clusterings of the four dominant clusters of clusterings from Figure 8(a). Clusterings are displayed as colorings/markings of the original points in the 2D plane. These are ordered to maximize cumulative setwise information. Notice how the first two representative-clusterings recover the original nine clusterings exactly (color figure available online).

4.4.2. Ordering by Setwise Information Content. In Figure 10, the original 3 × 3 community structure can be reconstructed using only the first two representative-clusterings. Why are these two chosen first? Selecting the third and fourth representative-clusterings would not have had this pleasant result. How should we order the set of representative-clusterings?

We may judge a set of representative-clusterings by a number of factors: (i) How many of our samples pattern with the associated metaclusters; what fraction of the space of clusterings do they cover? (ii) How much information do the clusterings cover as a set? (iii) How redundant are the clusterings? How much informational overlap is present?

We would like to maximize information while minimizing redundancy. In Figure 10, we ordered the representative-clusterings to maximize setwise information. Minimizing redundancy came as a fortunate side effect. Notice how each of the clusterings in order is independent of the preceding ones. Knowing that a vertex is in one cluster in the first image tells you nothing about the cluster to which it belongs in the second. The second therefore brings only novel information and no redundancy.

To compute the information content of a set of clusterings, we extend the variation-of-information metric in a natural way. In Section 2.2, we introduced the mutual information of two clusterings $\mathbf{C}_1$ and $\mathbf{C}_2$ as follows:

$$I(\mathbf{C}_1, \mathbf{C}_2) = \sum_{k=1}^{K_1} \sum_{l=1}^{K_2} P(\mathbf{C}_1, \mathbf{C}_2, k, l) \log P(\mathbf{C}_1, \mathbf{C}_2, k, l),$$

where $P(\,)$ is the probability that a randomly selected node was in the specified clusters. This is equivalent to the self-information of the Cartesian product of the two clusterings. Its extension to a set of n clusterings $I(\mathbf{C}_1, \mathbf{C}_2, \dots, \mathbf{C}_n)$ is

$$\sum_{a=1}^{K_1} \sum_{b=1}^{K_2} \cdots \sum_{z=1}^{K_n} P(\mathbf{C}_1, \mathbf{C}_2, \dots, \mathbf{C}_n, a, b, \dots, z) \log P(\mathbf{C}_1, \mathbf{C}_2, \dots, \mathbf{C}_n, a, b, \dots, z).$$

For a large number of clusterings or large K, this quickly becomes inconvenient. In these cases we order the clusterings by adding new clusterings to the set based on maximizing the minimum pairwise distance to every other clustering currently in the set. This process is seeded with the informationally maximal pair within the set. This does not avoid triple information overlap, but it works well in practice.

4.5. Physics Articles from arXiv.org

The e-print archive arXiv.org releases convenient metadata (title, authors, etc.) for all articles in their database. Additionally, a special set of 30 000 high-energy physics articles is released with abstracts and citation networks. We apply our process to this graph of papers with edge types *title, author, abstract*, and *citation*.

Articles are connected by *title* or *abstract* based on the cosine similarity of the text (using the bag of words model [Blei et al. 03]). Two articles are connected by *author* by the number of authors that the two articles have in common. Two articles are connected by *citation* if either article cites the other (undirected). We inspect this system with the following process, discussed in greater detail above.

These graphs are normalized by the L_2 norm, and then the space of composite edge types is sampled uniformly. That is, $\omega_j = \sum_{i=1}^{4} \alpha_i w_i$, where $\alpha_i \in (-1, 1)$, $w_i \in \{$*titles, abstract, authors, citation*$\}$. The resulting graphs are then clustered using the FastModularity algorithm [Clauset et al. 04]. The resulting clusterings are compared in a graph that is then clustered to produce clusters of clusterings. The clusters of clusterings are averaged [Strehl and Ghosh 03], and we inspect the resultant representative-clusterings.

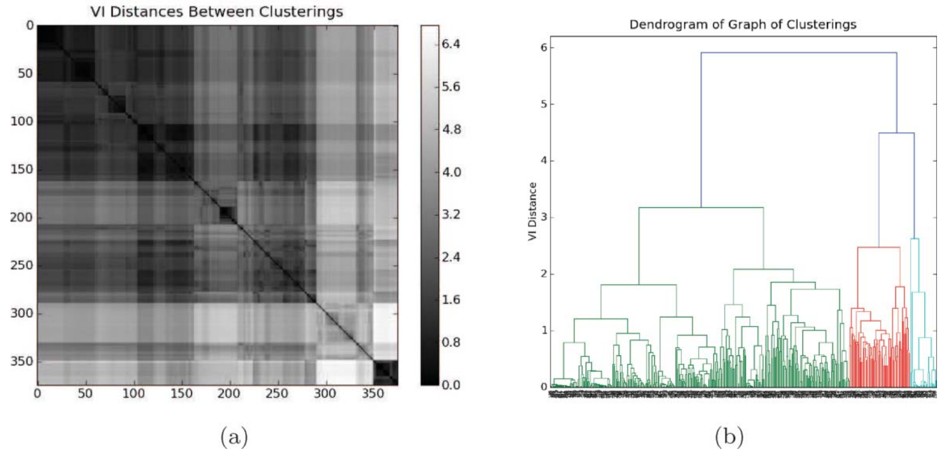


Figure 11. (a) The pairwise distances between the sampled clusterings form a graph. Note the dark blocks along the diagonal. These are indicative of tightly knit clusters. (b) A dendrogram of this graph. We use the ordering of the vertices picked out by the dendrogram to optimally highlight the blocks in the left-hand image (color figure available online).

The similarity matrix of the graph of clusterings is shown in Figure 11(a). The presence of blocks on the diagonal implies clusters of clusterings. From this process we obtain representative-clusterings. The various partitionings of the original set of papers vary considerably (large VI distance) yet exhibit high modularity scores, implying a variety of high-quality clusterings within the dataset.

Cluster	Statistically Significant Words in Clustering 1
1	quantum, algebra, integr, equat, model, chern-simon, lattic, particl, affin
2	potenti, casimir, self-dual, dilaton, induc, cosmolog, brane, anomali, scalar
3	black, hole, brane, supergrav, cosmolog, ads/cft, sitter, world, entropi
4	cosmolog, black, hole, dilaton, graviti, entropi, dirac, 2d, univers
5	d-brane, tachyon, string, matrix, theori, noncommut, dualiti, supersymmetr, $n = 2$

Cluster	Statistically Significant Words in Clustering 2
1	potenti, casimir, self-dual, dilaton, induc, energi, scalar, cosmolog, gravit
2	integr, model, toda, equat, function, fermion, casimir, affin, dirac
3	tachyon, d-brane, string, orbifold, $n = 2$, $n = 1$, dualiti, type, supersymmetr
4	black, hole, noncommut, supergrav, brane, sitter, entropi, cosmolog, graviti

Table 2. Commonly appearing words (stemmed) in two distinct representative-clusterings. Clusters within each clustering correspond to well-known subfields in high-energy physics (traditional field theory/lattice QCD, cosmology/GR, supersymmetry/string theory). These data, however, do not show a strong distinction between the clusterings. Further investigation is warranted.

Analysis of this dataset is challenging and still in progress. We can look at articles in a clustering and inspect attributes such as country (by submitting an e-mail's country code), or words that occur more often than statistically expected given the corpus. Most clusterings found show a separation into various topics identifiable by domain experts (example in Table 2). However, a distinction between clusterings has not yet been found. While the VI distance between metaclusterings presented in Figure 11(a) is large, it has so far proven difficult to identify the qualitative distinction for the quantitative difference. More in-depth inspection by a domain expert may be necessary.

5. Finding Unexpected Clusters

In many applications, a domain expert can predict what the clusters will look like before computing a clustering. For instance, if we have a graph whose vertices are countries with edges representing "strong ties" between the countries, we expect the geographic structure to have a strong influence on the clusters for many quantifications of a "strong tie." While this domain-specific information can be helpful in many ways, such a bias may obscure critical information about the data. For instance, we would like to know whether our graph has any good clusterings that are significantly different from those in a given list of clusterings, which brings us to the topic of this section: *Given a graph with multiple edge types and a set of clusterings, how do we compute an aggregate weighting function such that clusters of the aggregate graph will be maximally different from the given set of clusterings?*

This problem can be most naturally modeled as a multicriterion optimization problem, with the first criterion being the quality of the clustering, and the second criterion the information-independence from the given clusterings. We need high-quality clusterings, since we want the clustering to show significant value. One can impose a clustering structure even on an Erdős–Rényi-style graph, but such clusters will not have any value. We want clusters that are statistically significant. We also want to learn something that we do not already know, whence the second criterion, information-independence. If there are multiple clusters from which we want to deviate, then the second criterion will hold multiple criteria within it.

The methods we have proposed so far can be employed to solve this problem as well. Again, we need to quantify the information-independence between two clusterings for which the variation-of-information metric, described in Section 2.2, can be used. And we need a method to search the space for the aggregate function efficiently, which can be done using the optimization methods adopted in Section 3.2.3.

5.1. Case Study: Countries

We applied our methods to a graph of the world in which each node is a country, and we modeled the relations between these countries with six types of edges:

1. common land border,
2. major trade partners,
3. common language,
4. common religion,
5. common ethnicity,
6. similar labor force distribution (industry, agriculture, services, etc.).

The data were taken from the *CIA World Factbook 2006* [CIA 06], which is conveniently available in structured text format. The edges of type borders and languages are unweighted, while the others are weighted based on the probability that two people from these countries share the same trait. For example, the probability that a Mexican citizen and American citizen share a common religion is 0.24, which becomes the weight of the edge. The edge types of this graph are various in nature. Some have small diameter; some exhibit hubs and authorities; some have skewed degree distributions, while others do not (see Figure 12). Many of the edge types are correlated geographically. Ethnic, linguistic, religious, and trade relationships are often local. Can we find clusterings that are substantially different from what this rule suggests?

5.1.1. Searching with a Linear Aggregate Function. As in Section 3, we optimized over the space of linear combinations of the edge types, but this time, we allowed the weight of each edge type to be in the interval $\alpha_i \in [-1, 1]$, but we did not allow the weight of any edge to be negative, by truncating at 0. We used the HOPSPACK derivative-free optimization package and a coefficient to balance between the two criteria. Clustering quality is given by the modularity metric, and the FastCommunity software [Clauset et al. 04] is used to find the clusterings.

In Table 3, we present one of the clusterings we found. We note that the modularity score shows that the clusters are statistically significant, and the distance from the continents-based clustering is high. Thus this clustering provides new information. This solution strongly favors aspects like language, which is less geographically linked than others, and strongly disfavors aspects such as the

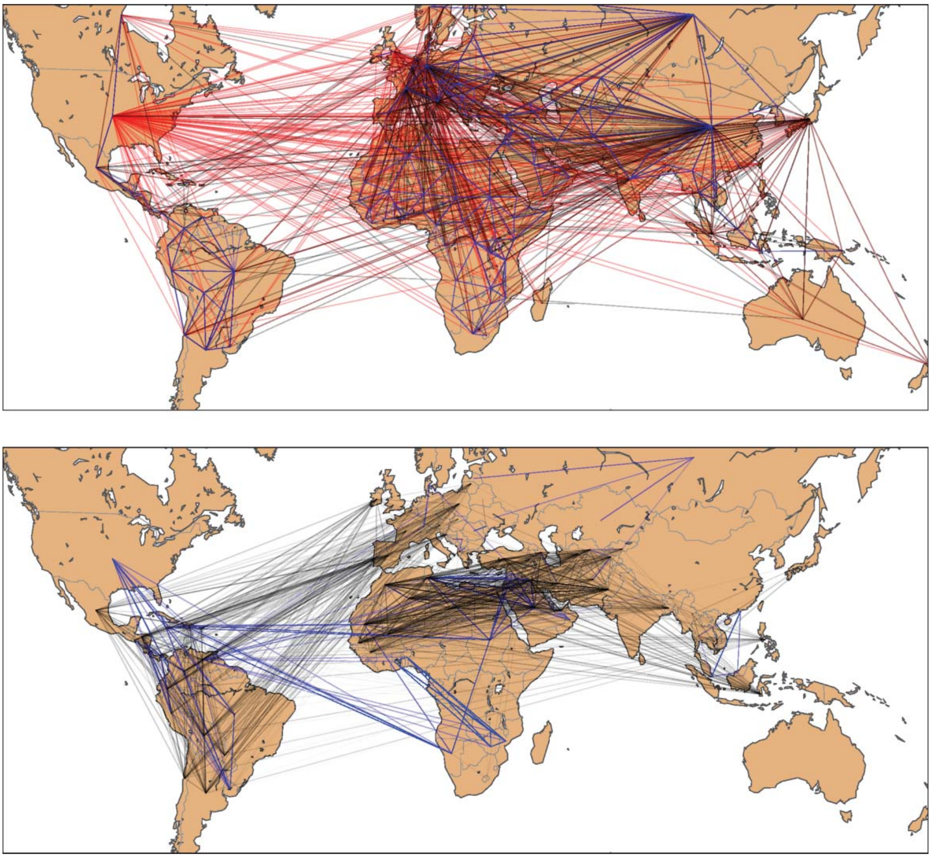


Figure 12. The world map with some edge types plotted. Top: export/import graphs (black and red) and physical land connections (blue). Bottom: religion (black) and ethnicity (blue) (color figure available online).

sharing of a land border. The clusters obtained are at best vaguely related to the continental clustering. One can see traces of pre-World War I empires. Some of the clusters do highlight interesting linguistic groups that are conveniently spread across continental borders. Cluster 2 contains several countries with Turkic roots. Cluster 3 contains a Slavic family. However, it is noteworthy that some countries with Turkic roots, such as Kyrgyzstan, Kazakhstan, and Uzbekistan, were grouped with the Slavic family. This shows that this high-quality clustering cannot be explained by language and/or ethnicity data alone. It is the combination that leads to this particular clustering.

Cluster 1	Angola, Belgium, Cambodia, Republic of the Congo, China, Egypt, Greece, Hong Kong, Italy, Laos, Mali, Malaysia, Nigeria, Rwanda, Singapore, Thailand, Taiwan, Papua New Guinea
Cluster 2	Denmark, Mongolia, Macedonia, Turkey, Turkmenistan
Cluster 3	Belarus, Georgia, Kyrgyzstan, Kazakhstan, Latvia, Lithuania, Serbia, Russia, Ukraine, Uzbekistan
Cluster 4	Bangladesh, Benin, Canada, Cameroon, Ireland, France, Ghana, Germany, Haiti, Indonesia, India, Israel, Cote d'Ivoire, Jamaica

Table 3. Groups of a clustering with maximum information-independence from continent-based clustering. The modularity of the clustering is 0.397, and the VI distance is 2.6. The solution used linear weights of labor (−0.99), exports (−0.5), language (0.5), imports (−0.61), religion(−0.06), borders (−0.95), and ethnicity (0.95).

5.1.2. Drawback of Linear Aggregation. In addition to some interesting results, our experiments also pointed to some potential drawbacks of using a linear aggregate function. We have observed that it is possible to create clusterings with significant information difference, but these clusterings are not necessarily high in quality and/or statistically significant. And the reason for this is that linear aggregation may yield dense or sparse but Erdős–Rényi-like random graphs. In this section, we will explain why this is the case.

The adjacency matrix of an undirected nonnegatively weighted graph is symmetric positive semidefinite, and thus can be pictured as ellipsoids with the major axes along the eigenvectors with the length of any axis corresponding to the eigenvalue. If the matrix has few significant eigenvalues, then the ellipsoid is flat in many directions, and the matrix/graph can be simplified to a lower-dimensional representation without much loss (see Figure 13), which is the underlying idea

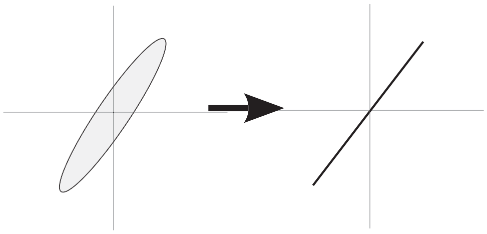
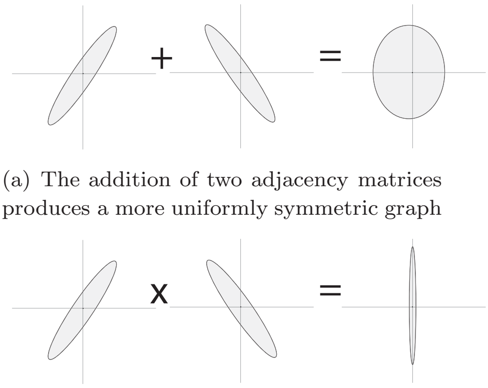


Figure 13. A clustering of a graph is just a low-rank approximation.



(a) The addition of two adjacency matrices produces a more uniformly symmetric graph

(b) The multiplication of two adjacency matrices produces a more degenerate, modular graph, amenable to low-rank approximations/clustering.

Figure 14. Illustration of effects of additive (a) and multiplicative (b) aggregate function on the space of clusterings.

of spectral clustering. A random or complete graph, on the other hand, is closer to a symmetric sphere and cannot be reduced without substantial loss.

This raises another interesting question: Can we use multiple edge types to improve the statistical significance of clusters? For instance, a multiplicative, as opposed to additive, aggregate function may restrict the search space (as illustrated in Figure 14). In this case, we take only connections endorsed by more than one edge type into account. We demonstrate this with an example. Clustering the countries graph with respect to only the religion and language edge types produces modularity scores of 0.44 and 0.3, respectively. If we use a graph in which we have an edge if either a language or a religion edge exists, the modularity score decreases to 0.23, which means that the clusters are statistically less significant. On the other hand, if we use a graph in which we have an edge only if we have both a language and a religion edge, the modularity increases to 0.47, which means that the clusters have much more significance.

5.2. Finding Clusters with Higher Significance Using Product Pairs

In the previous section, we discussed how significance of clusterings can be improved by looking at products of edge types (i.e., we add an edge only if both edge types support it). In this section, we present results based on this idea. We computed the modularity scores and VI distances from the continental clustering

Single Metric			Metric Pair		
Name	Modularity	VI distance	Name	Modularity	VI distance
language	0.25	1.79	language × ethnicity	0.43	2.40
religion	0.37	2.23	religion × ethnicity	0.45	2.16
border	0.42	0.69	imports × religion	0.41	2.17
exports	0.29	1.94	imports × labor	0.29	1.85
			exports × imports	0.37	2.35
			ethnicity × exports	0.41	1.93
			labor × language	0.26	2.04
			labor × exports	0.28	1.95

Table 4. A few of the product pairs and singleton edge types listed along with their modularity score and variation-of-information distance from the continental clustering.

for all single-edge types and the product of each pair of edge types. We display those with interesting results in Table 4.

Using the same approach as in Section 4.4.2, we select pairs that maximize setwise information away from the continental clustering. We found that the clustering that is most different from the continental clustering is given by the product of the language and ethnicity graphs. Given those two clusterings, the next most informationally distant comes from looking at import–export relationships. Finally, after the countries have been classified along these edge types, the religion graph is the most distinguishing. Note that the products yield very high quality clusterings that are at the same time distant from each other.

The three clusterings resulting from these three graphs are sufficiently distinct to distinguish almost all of the different countries from one another. The clusterings are sufficiently distant so that almost no two countries lie in the same cluster in all three of the clusterings. Equivalently, most countries can be identified by their membership in the three clusterings. This achieves the goal set forth

Continents	Language × Ethnicity	Imports × Exports	Religion	...

Table 5. A set of edge types ordered to yield maximum setwise information/ minimum setwise redundancy. Each, in turn, is as different as possible from each of the preceding clusterings. Note that these represent relatively orthogonal ideas (geography, culture, economy, ...) in turn.

Group 1	Lebanon, Syria, Yemen
Group 2	Norway, Sweden
Group 3	Belgium, Denmark, Netherlands
Group 4	Colombia, Dominican Republic, Ecuador, El Salvador, Guatemala, Honduras, Mexico, Nicaragua, Peru, Venezuela

Table 6. Some groups of countries that were consistently clustered together.

in Section 4 to represent a graph efficiently through well-chosen representative clusterings, drastically reducing information with minimal content loss.

Another interesting result in our experiments was the groups of countries that were always grouped together, which are listed in Table 6. This list will not be surprising to many, but it still underlines the concept of metric-invariant clusters in graphs with multiple edge types.

6. Conclusions

We have addressed clustering in the context of graphs with multiple edge types. We have investigated several problems within this context: recovering an aggregation scheme from ground-truth clustering, finding a metaclustering structure and efficiently representing that structure, and finding unexpected clusters. We have also presented case studies on real datasets.

The main result of our work is that working on graphs with multiple edge types prevents loss of crucial information. Instead of a single clustering, a rich clustering structure can exist with clusters of clusterings, and latent clusters can be discovered that compactly explain the underlying graph. We have shown that high-quality clusters that are significantly different from what is expected can be found, and significance of clusters can be improved by looking at intersections of multiple edge types. Another important conclusion of this work is that despite the increased complexity due to multiple edge types, we have the algorithmic tools to make the associated problems tractable.

We hope that our work will draw the attention of the research community to this important problem, which bears many intriguing research challenges, and we can see much room for growth in this topic. We have mostly focused on linear aggregation, and nonlinear combinations (as we have seen in Section 5.2) should be investigated. More-intelligent sampling methods should be possible. And most importantly, working on more datasets will give us a chance to better evaluate our methods.

Acknowledgments The authors would like to thank Philip Kegelmeyer for providing the file system dataset, Vince DeSapio and Tamara Kolda for many helpful discussions. We are also grateful to two anonymous reviewers for their helpful comments on an earlier version of this paper.

This work was funded by the applied mathematics program at the United States Department of Energy and performed at Sandia National Laboratories, a multiprogram laboratory managed and operated by Sandia Corporation, a wholly owned subsidiary of Lockheed Martin Corporation, for the United States Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000.

References

[Aiello et al. 01] W. Aiello, F. Chung, and L. Lu. "A Random Graph Model for Power Law Graphs." *Experimental Mathematics* 10 (2001), 53–66.

[Blei et al. 03] David M. Blei, Andrew Y. Ng, and Michael I. Jordan. "Latent Dirichlet Allocation." *Journal of Machine Learning Research* 3:4-5 (2003), 993–1022.

[Chung and Lu 02a] F. Chung and L. Lu. The average distances in random graphs with given expected degrees. *Proceedings of the National Academy of Sciences* 99:25 (2002), 15879–15882.

[Chung and Lu 02b] F. Chung and L. Lu. "Connected Components in Random Graphs with Given Degree Sequences." *Annals of Combinatorics* 6 (2002), 125–145.

[CIA 06] Central Intelligence Agency. *The World Factbook 2006.* Central Intelligence Agency, 2006.

[Clauset et al. 04] Aaron Clauset, M. E. J. Newman, and Cristopher Moore. "Finding Community Structure in Very Large Networks." *Physical review. E* 70 (2004), 066111.

[Dhillon et al. 07] Inderjit S. Dhillon, Yuqiang Guan, and Brian Kulis. "Weighted Graph Cuts without Eigenvectors: A Multilevel Approach." *IEEE Transactions on Pattern Analysis and Machine Intelligence* 29:11 (2007), 1944–1957.

[Dunlavy et al. 06] Daniel M. Dunlavy, Tamara G. Kolda, and W. Philip Kegelmeyer. "Multilinear Algebra for Analyzing Data with Multiple Linkages." Technical Report SAND2006-2079, Sandia National Laboratories, 2006.

[Lancichinetti and Fortunato 09] Andrea Lancichinetti and Santo Fortunato. "Community Detection Algorithms: A Comparative Analysis." *Physical Review E* 80:5 (2009), 056117.

[Lancichinetti et al. 08] Andrea Lancichinetti, Santo Fortunato, and Filippo Radicchi. "Benchmark Graphs for Testing Community Detection Algorithms." *Physical Review E* 78:4 (2008), 1–5.

[Lange et al. 04] T. Lange, V. Roth, M. L. Braun, and J. M. Buhmann. "Stability-Based Validation of Clustering Solutions, Neural Computation." *Neural Computation* 16 (2004), 1299–1323.

[Leskovec et al. 09] J. Leskovec, K. Lang, A. Dasgupta, and M. Mahoney. "Community Structure in Large Networks: Natural Cluster Sizes and the Absence of Large Well-Defined Clusters." *Internet Mathematics* 6 (2009), 29–123.

[Luo 05] X. Luo. "On Coreference Resolution Performance Metrics." In *Proc. Human Language Technology Conf. and Conf. Empirical Methods in Natural Language Processing, Vancouver, British Columbia, Canada, 2005*, pp. 25–32. Association for Computational Linguistics, 2005.

[Meila 03] Marina Meila. "Comparing Clusterings by the Variation of Information." Technical Report, pp. 173–187, 2003.

[Mirkin 96] B. Mirkin. *Mathematical Classification and Clustering.* Kluwer Academic Press, 1996.

[Mucha et al. 10] P. J. Mucha, T. Richardson, K. Macon, M. A. Porter, and J. P. Onnela. "Community Structure in Time-Dependent, Multiscale, and Multiplex Networks." *Science* 328:5980 (2010), 876–878.

[Newman 04] M. Newman. "Analysis of Weighted Networks." *Phys. Rev. E* 70:5 (2004), 056131.

[Newman 06] M. Newman. "Modularity and Community Structure in Networks." *PNAS* 103 (2006), 8577–8582.

[Newman et al. 02] M. Newman, D. Watts, and S. Strogatz. "Random Graph Models of Social Networks." *Proceedings of the National Academy of Sciences* 99 (2002), 2566–2572.

[Plantenga 09] T. Plantenga. "Hopspack 2.0 User Manual." Technical Report SAND2009-6265, Sandia National Laboratories, 2009.

[Stichting et al. 00] C. Stichting, M. Centrum, and S. V. Dongen. "Performance Criteria for Graph Clustering and Markov Cluster Experiments." Technical Report INS-R0012, Centre for Mathematics and Computer Science, 2000.

[Strehl and Ghosh 03] Alexander Strehl and Joydeep Ghosh. "Cluster Ensembles: A Knowledge Reuse Framework for Combining Multiple Partitions." *Journal of Machine Learning Research* 3:3 (2003), 583–617.

Matthew Rocklin, Dept. Computer Science, University of Chicago, 1100E 58th Street, Chicago, IL 60637 (mrocklin@cs.uchicago.edu)

Ali Pinar, Sandia National Laboratories, 7011 East Avenue, Livermore, CA 94551 (pinar@sandia.gov)