Towards Theoretical Bounds for Resource-bounded Information Gathering for Correlation Clustering

Work done as part of the Synthesis Project with Andrew McCallum and Ramesh Sitaraman

Pallika Kanani

University of Massachusetts, Amherst

Abstract. Resource-bounded Information Gathering for Correlation Clustering deals with designing efficient methods for obtaining and incorporating information from external sources to improve accuracy of clustering tasks. In this paper, we formulate the problem, and some specific goals and lay the foundation for better theoretical understanding of this framework. We address the challenging problem of analytically quantifying the effect of changing a single edge weight on the partitioning of the entire graph, under some simplifying assumptions, hence demonstrating a method to calculate the expected reduction in error. Our analysis of different query selection criteria provides a formal way of comparing different heuristics. We compare the solution of our theoretical analysis with simulation results. We also estimate the probability of recovering the true partition under various query selection strategies for general random graphs and discuss some possible directions for approximation. Next, we prove a related bound under certain assumptions. We also describe some general techniques to efficiently query and select nodes for expanding graphs.

1 Introduction

Learning under resource constraints has been of interest to the Machine Learning community in various contexts. One of the least understood of these is designing efficient methods for obtaining and incorporating information from external sources to improve accuracy of clustering tasks. We assume that there exists some "true" underlying clustering for our data, which we are not able to recover due to noise. In many real life scenarios, we may also have access to an external source of information, which can be queried in order to reduce the noise or in other ways help get closer to the true clustering. Invariably, such external information is available at a cost, and we must try to select a subset of the available information, so as to achieve the best cost-benefit ratio.

What makes this problem really interesting is the non-trivial effect of obtaining a single piece of information on the clustering results of the entire dataset. If we formulate the clustering problem as graph partitioning of a weighted graph, changing a single edge weight can affect the clustering result, depending on the structure of the graph. Similarly, introducing an additional node in the graph can also affect the clustering results significantly. Our strategy for querying external information needs to take this into account to be efficient.

Our theoretical understanding of graph partitioning problems is still evolving, and to the best of our knowledge, this problem has not been studied in this form. Correlation clustering is an important class of algorithms that address similarity - dissimilarity type of data well. There has been very little work on the effect of correlation clustering on changing graphs. Probably the most significant contribution of this project has been to formulate the problem of Resource-bounded Information Gathering for Correlation Clustering, along with specific subgoals.

The most challenging aspect of this problem is to quantify the impact made by a single edge on the partitioning of the entire graph. We address this question under several simplifying assumptions to demonstrate a method to calculate the expected reduction in error as the result of a single query. We propose some interesting directions for approximation of this quantity. Our analysis of different query selection criteria provides a formal way of comparing different heuristics. We also prove a related bound under certain assumptions. Next, we describe some general techniques for efficiently querying an external source to obtain new nodes and incorporating them in the graph to gain most improvement with least investment of resources.

This is a rich area with many interesting possible directions, and this work lays the foundation for better theoretical understanding of algorithms for performing graph partitioning, as well as Resource-bounded Information Gathering.

1.1 Problem Definition

Resource Bounded Information Gathering for Correlation Clustering The standard correlation clustering problem on a graph with real-valued edge weights is as follows: there exists a fully connected graph $G(V, E)$ with n nodes and edge weights, $w_{ij} \in [-1, +1]$. The goal is to partition the vertices in V by minimizing the inconsistencies with the edge weights [Bansal *et al.*2002]. That is, we want to find a partitioning that maximizes the objective function $\mathcal{F} = \sum_{ij} w_{ij} f(i, j)$, where $f(i, j) = 1$ when v_i and v_j are in the same partition and -1 otherwise.

Now consider a case in which there exists some “true” partitioning $\mathcal{P}$, and the edge weights $w_{ij} \in [-\infty, +\infty]$ are drawn from a random distribution (noise model) that is correlated with whether or not edge $e_{ij} \in E$ is cut by a partition boundary in $\mathcal{P}$. The goal is to find an approximate partitioning, $\mathcal{P}_a$, of V into an unknown number of k partitions, such that $\mathcal{P}_a$ is as ‘close’ to $\mathcal{P}$ as possible. There are many different possible measures of closeness to choose from. Let $\mathcal{L}(\mathcal{P}, \mathcal{P}_a)$ be some arbitrary loss function. If no additional information is available, then we could simply find a partitioning that optimizes $\mathcal{F}$ on the given weights.

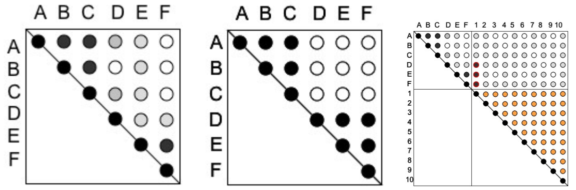
In this paper, we consider settings in which we may issue queries for additional information to help us reduce loss $\mathcal{L}$. Let $G_0(V_0, E_0)$ be the original graph.

Let $\mathcal{F}_0$ be the objective function defined over G_0. Our goal is to perform correlation clustering and optimize $\mathcal{F}_0$ with respect to the true partitioning of G_0. We can augment the graph with additional information using two alternative methods: (1) updating the weight on an existing edge, (2) adding a new vertex and edges connecting it to existing vertices. We can obtain this additional information by querying a (possibly adversarial) oracle using two different types of queries. In the first method, we use a query of type Q1, which takes as input edge e_{ij} and returns a new edge weight w'_{ij}, where w'_{ij} is drawn from a different distribution that has higher correlation with the true partitioning $\mathcal{P}$. Alternatively, we may also assume that the oracle is friendly, and always returns a true value of the edge weight that is queried.

In the second method, we can expand the graph G_0, by adding a new set of vertices, V_1 and the corresponding new set of edges, E_1 to create a larger, fully connected graph, G'. Although we are not interested in partitioning V_1, we hypothesize that partitioning G' would improve the optimization of $\mathcal{F}_{/}$ on G_0 due to transitivity of partition membership. These can be obtained by second type of query, Q2, which takes as input (V_0, E_0) and returns a subset $V'_s \subset V_1$. Note that the additional nodes obtained as a result of the queries of type Q2 help by inducing a new, and presumably more accurate partitioning on the nodes of G_0. In this case, given resource constraints, we must select $V'_s \subset V_1$ to add to the graph. Fig. 1 illustrates the result of these queries.

However, there exist many possible queries of type Q1 and Q2, each with an associated cost. There is also a cost for performing computation on the additional information. Hence, we need an efficient way to select and order queries under the given resource constraints.

Formally, we define the problem of *resource-bounded information gathering for correlation clustering* as follows. Let $c(q)$ be the cost associated with a query $q \in Q1 \cup Q2$. Let b be the total budget on queries and computation. Find distinct queries $q_1, q_2, ...q_m \in Q1 \cup Q2$ and $\mathcal{P}_a$, to minimize $\mathcal{L}(\mathcal{P}, \mathcal{P}_a)$, s.t. $\sum_{q_i} c(q_i) \leq b$.



(a) G_0 (b) Result of Q1 (c) Result of Q2

Fig. 1. Results of the two kinds of queries. (a) The adjacency matrix of G_0 where darker circles represent edges with higher weight. (b) The new edge weights w'_{ij} after issuing the queries from Q1. (c) The graph expanded after issuing queries from Q2. The upper left corner of the matrix corresponds to G_0 and the remaining rows and columns correspond to the nodes in V_1.

Some Specific Goals for Queries of type Q1 In our previous work [Kanani *et al.*2007], we compare random, heuristic based, and expected entropy based approaches for selecting queries. Following are some specific goals to be proved for these approaches. These examples compare the random vs. uncertainty based approaches, but they can be generalized to any proposed approach.

Approach 1: Assuming μ shifts :

Goal 0: Here we assume that the weight matrix observed as a result of the queries comes from a different (and presumable less noisy) distribution. We then want to prove that we get at least ϵ improvement as a result of each query.

Approach 2: Assuming we get the perfect W_{ij} or the perfect label for the queried edge :

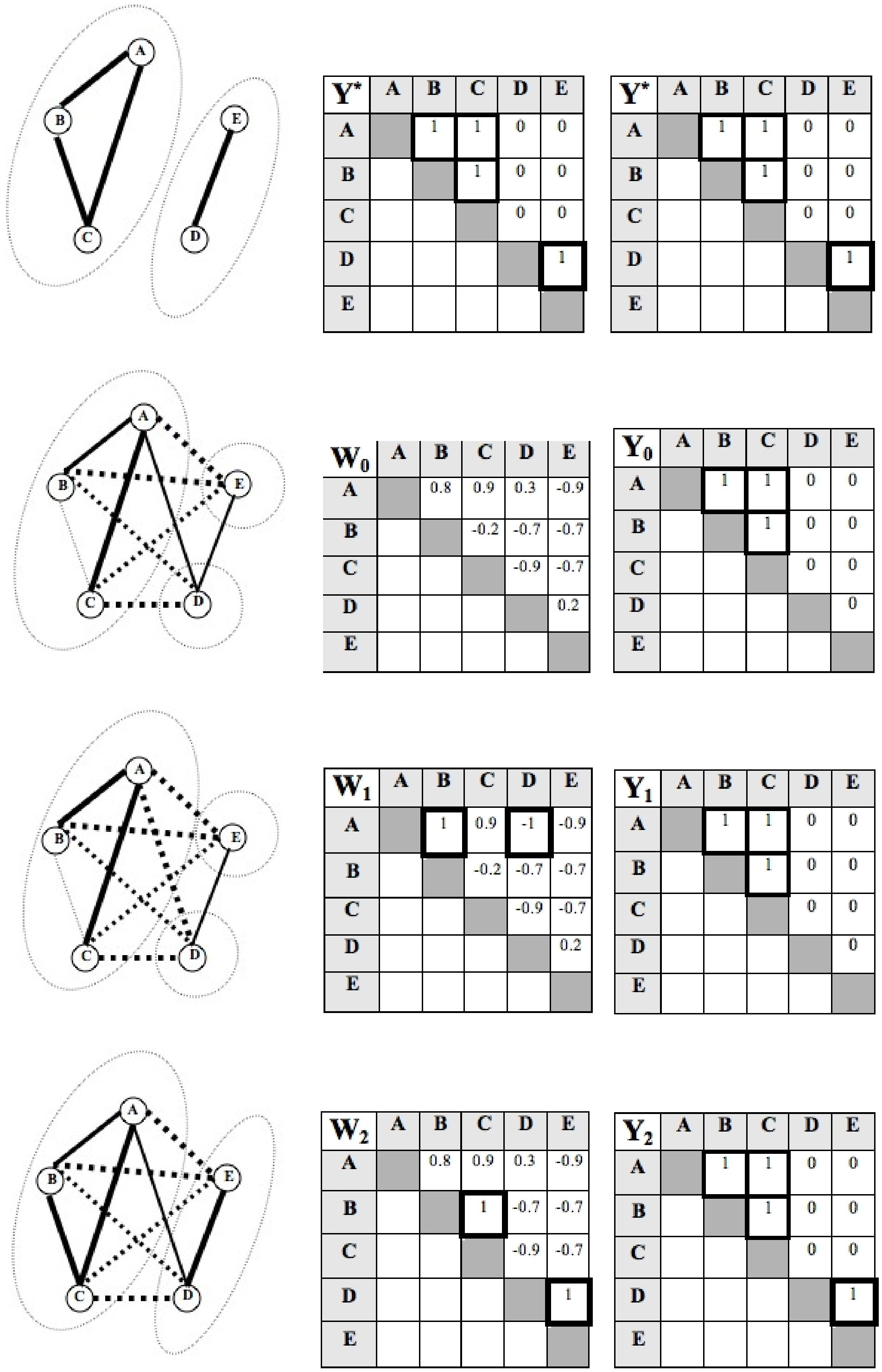
Goal 1: Let γ be the fraction of edges to be selected for labeling. If the method of selection is uniformly random, then the expected reduction in the cost of clustering is $f_1(\gamma.\delta, \mu)$. If the method of selection is based on uncertainty, then the expected reduction in the cost of clustering is $f_2(\gamma.\delta, \mu)$. To Prove: f_2 rises faster than f_1 w.r.t. γ. Or that value of f_2 is great than f_1 for a $\gamma > c$

Goal 2: Let γ be the fraction of edges to be selected for labeling. If the method of selection is uniformly random, then the error bound on correlation clustering is $g_1(\gamma.\delta, \mu, n, \epsilon)$ Similarly, the error bound of graph partially labeled by method based on uncertainty is $g_2(\gamma.\delta, \mu, n, \epsilon)$. To Prove: g_2 is tighter than g_1.

Illustrative Example Fig. 2 illustrates approach 2 for an example graph. The cost reduces when new information is obtained, but the magnitude of reduction depends on the selection criterion used for querying the edge. Also note the effect on the true error of partitioning.

A Note on Optimality In the general context of this problem, we can work with two different optimal criteria. One is utility based, i.e. given a fixed budget, b, what is the expected reduction in error (true or observed) after b queries. The other one is erorr based, i.e. given that we require a specific amount of reduction in error (again, true or observed), what is the expected number of queries needed to achieve this reduction. A related optimality criterion is maximizing the area under the curve of measured accuracy after execution of each query. In general, the optimality criterion most suitable to the problem at hand should be selected.

A Note on True Error vs. Cost Function In this work, we assume a true underlying clustering of our data. Even though different cost functions can be



Y^*	A	B	C	D	E
A		1	1	0	0
B			1	0	0
C				0	0
D					1
E					

Y^*	A	B	C	D	E
A		1	1	0	0
B			1	0	0
C				0	0
D					1
E					

W_0	A	B	C	D	E
A		0.8	0.9	0.3	-0.9
B			-0.2	-0.7	-0.7
C				-0.9	-0.7
D					0.2
E					

Y_0	A	B	C	D	E
A		1	1	0	0
B			1	0	0
C				0	0
D					0
E					

W_1	A	B	C	D	E
A		1	0.9	-1	-0.9
B			-0.2	-0.7	-0.7
C				-0.9	-0.7
D					0.2
E					

Y_1	A	B	C	D	E
A		1	1	0	0
B			1	0	0
C				0	0
D					0
E					

W_2	A	B	C	D	E
A		0.8	0.9	0.3	-0.9
B			1	-0.7	-0.7
C				-0.9	-0.7
D					1
E					

Y_2	A	B	C	D	E
A		1	1	0	0
B			1	0	0
C				0	0
D					1
E					

Fig. 2. From top to bottom: True clustering, The original graph, Two edges picked randomly, Two edges picked based on uncertainty

used to measure "goodness" of a particular partitioning, ultimately, we are interested in reducing the "true" error as a result of information gathering. This should be kept in mind while analyzing this problem.

1.2 Synthesis Contributions

This work is done as part of the synthesis project, combining the areas of Machine Learning and Theory. Section 5 describes the motivating Machine Learning application. Here, we describe the specific contributions of this project towards the theoretical analysis of our problem.

Problem Formulation and Specifying Goals The first and probably the most significant contribution of this project has been to formulate the problem of Resource-bounded Information Gathering for Correlation Clustering, along with specific subgoals. This problem in it's specific form has not been studied or defined in the literature to the best of our knowledge. In [Kanani and McCallum2007], we introduced it as an open problem. There are several other subgoals that can be defined, along with several possible directions for future work. In this work, we discuss some of these rich possibilities.

Calculating Expected Reduction in Error At the heart of Resource-bounded Information Gathering is the following question. How does the change in one edge weight affect the partitioning of the entire graph, resulting in the change in error? In this case, we assume that we have an access to the "friendly" oracle, which unveils the true value of the edge weight being queried. We address this question under several simplifying assumptions to demonstrate a method to calculate the expected reduction in error as the result of a single query. We discuss various possibilities for approximation. Our analysis of different query selection criteria provides a formal way of comparing different heuristics. We compare the solution of our theoretical analysis with simulation results. We also perform simulations to estimate the probability of recovering the true partition under various query strategies for general random graphs.

Proving a Bound on Reduction in Error Alternatively, we can assume that the edge weights of the graph resulting at the end of a single query come from a different, but related distribution. In this case, we prove an upper bound on the probability of obtaining at least ϵ improvement at the end of each query, as a function of the gap between the parameters of the model before and after obtaining new information. We use previously proven error bounds on correlation clustering as a basis for our proof.

Techniques for RBIG for Expanding the Graph This part of our work is a classic example of going back and forth between our problem domain and theory. We started by defining the general problem of selectively filling entries of a potentially infinite matrix. Next, we defined it in terms of our domain and applied some simple algorithmic techniques to solve it. Here, we abstract away again from our specific domain and present the solutions in general form. We describe the techniques for selecting a subset of the queries to obtain additional nodes as well as for selecting a subset of the nodes to add to the graph for improving partitioning. We also describe the assumptions under which these techniques hold.

2 Changing Edge Weights - Expected Reduction in Error

In this section, we assume that we have access to an oracle, which unveils the "true" edge weight of the requested edge, i.e. it returns a value of 0(do not merge) or 1(must merge) for the queried edge. We now demonstrate a method to calculate the expected reduction in error as the result of a single query.

2.1 Notation

Let G be a weighted, undirected, complete graph with n nodes. Let W be the corresponding weight matrix, and Y be the indicator matrix representing partitions S of G. Note that S partitions the nodes of graph G into equivalence sets, and $Y_{ij} \in \{0,1\}$.

We assume there is an arbitrary "true" partition $S^* = \{S^*_1, ..., S^*_{k^*}\}$ of the vertices. i.e. $S^*_1 \cup .... \cup S^*_{k^*} = X$ and $S^*_i \cap S^*_j = \emptyset$. Number of clusters k^* and the size of each cluster are arbitrary and unknown.

To this partition S^* corresponds a probability distribution $P_{S^*}(W)$ over edge weights. We assume that $P_{S^*}(W)$ is the process that generates the data we want to cluster. The goal of the clustering algorithm is to recover the true partition S^* underlying the data generating process $P_{S^*}(W)$ from a single realization of edge weights W

Let us define the error of a partitioning S with respect to true clustering S^* using the following pair-wise loss function.

$$d(S, S^*) = ||Y(S) - Y(S^*)||^2_F \tag{1}$$

Here, $||.||_F$ denotes the Frobenius norm. The function d measures the number of pairs on which the two partitionings disagree.

Consider the following true errors in the partitions before and after obtaining information. We use the subscript (or superscript) 0 to denote the variables related to the original graph and the corresponding correlation clustering and the subscript (or superscript) 1 to denote the corresponding variables after obtaining information.

$$d_0 = d(S_0, S^*) \tag{2}$$

$$d_1 = d(S_1, S^*) \tag{3}$$

2.2 The Planted Partition Model

(Karp). In a graph with n vertices, the edge weights are generated by a distribution

$$P_{S^*}(W|M,a,b) = \prod_{i=1}^{n}\prod_{j=1}^{n} P_{S^*}(W_{ij}|M_{ij},a,b) \tag{4}$$

So that each element W_{ij} of W is a bounded independent random variable in the interval $[a,b]$ with mean M_{ij}. Each $P_{S^*}(W_{ij}|M_{ij},a,b)$ is constrained by the true partitioning S^* as follows. If $Y(S^*)_{ij} = 1$ (vertices i and j are in the same cluster), the mean M_{ij} of W_{ij} must fulfill the constraint that $M_{ij} = m_+ > 0$. If $Y(S^*)_{ij} = 0$ (vertices i and j are in different clusters), the mean M_{ij} of W_{ij} must fulfill $M_{ij} = m_- > 0. Here, m_+ > m_-$

2.3 Specific Model Assumptions

We assume that each W_{ij} is drawn from a beta distribution with mean either as m_+ or m_-, depending on the constraint described above. Therefore, $W_{ij} \in [0,1]$. The betas we use are $beta_1(1, \frac{1-m_+}{m_+})$ and $beta_2(1, \frac{1-m_-}{m_-})$. We select these values of the parameters of beta for convenience. We shall describe other simplifying assumptions as we encounter them.

We shall now describe a method to calculate the expected reduction in error as a result of a single query.

2.4 Method to calculate expected reduction in error at the end of one query

Expected error for the original graph Let us first consider the expected error of a partitioning generated by a simple stochastic graph partitioning algorithm on the original graph.

$$E(d_0) = \sum_{S_0} P(S^0|S^*).d(S^0,S^*) \tag{5}$$

In order to compute $P(S^0|S^*)$, we integrate over W. Hence, we need to compute the following integral.

$$P(S^0|S^*) = \int_W \prod_{i,j=1}^{n} P(Y_{ij}|W).P(W_{ij}|S^*)dW \tag{6}$$

Notice that the probability of whether or not two nodes are partitioned depends on the graph partitioning algorithm.

Let's say we use a very simple algorithm. Consider the following stochastic algorithm for graph partitioning:

Step 1. Starting with graph G and the corresponding weight matrix W, we build a new graph H, with adjacency matrix A. For each edge (i,j), flip a coin

with probability proportional to the edge weight that an edge exists in H between nodes i and j.
Step 2. Take a transitive closure of H.

Note that the probability $P(Y_{ij}|W)$ now consists of the following. Let $H^k_{i\rightarrow j}$ be a path in H from node i to node j of length at most k:

$$P(Y_{ij}|W) = P(A_{ij} = 1) * P(\exists H^n_{i\rightarrow j}) \tag{7}$$

We consider the edge weight W_{ij} as the probability that there exists an edge in A between nodes i and j. Also, given a clustering, under the planted partition model, W_{ij}s are independent. For simplicity, we do not consider a full transitive closure but a 2-step transitive closure of H.

$$\begin{aligned}
P(Y_{ij} = 1|W) &= P(\exists(H^1_{i\rightarrow j} \vee H^2_{i\rightarrow j})) \\
&= 1 - P(\neg\exists(H^1_{i\rightarrow j} \vee H^2_{i\rightarrow j})) \\
&= 1 - P(\neg\exists H^1_{i\rightarrow j} \wedge \neg\exists H^2_{i\rightarrow j}) \\
&= 1 - [(1 - P(\exists H^1_{i\rightarrow j})) \times \prod_k P(\neg\exists H^2_{i\rightarrow j} passingthroughk)] \\
&= 1 - [(1 - W_{ij}) \times \prod_k (1 - P(\exists H^2_{i\rightarrow j} passingthroughk))] \\
&= 1 - [(1 - W_{ij}) \times \prod_k (1 - P(\exists H^1_{i\rightarrow k} \wedge \exists H^1_{k\rightarrow j}))] \\
&= 1 - [(1 - W_{ij}) \times \prod_k (1 - W_{ik}.Wkj)]
\end{aligned}$$

This can be generalized to n-step via a dynamic programming solution. Hence, we now get an expression for $P(Y_{ij} = 1|W)$ as follows:

$$P(Y_{ij} = 1|W) = 1 - [(1 - W_{ij}) \prod_k (1 - W_{ik}.Wkj)] \tag{8}$$

Similarly, we get

$$P(Y_{ij} = 0|W) = (1 - W_{ij}) \prod_k (1 - W_{ik}.Wkj) \tag{9}$$

Now, in order to evaluate the integral in eqn. 6, we need to know the configuration of the underlying true clustering, S^*. Let us assume that we are given this information. Let D^+ be the set of edges drawn from $beta_1$ and D_- be the set of edges drawn from $beta_2$. Therefore, eqn. 6 now becomes

$$
\begin{aligned}
P(S^0|S^*) = \int_W \prod_{Y_{ij}=1 \wedge (i,j) \in D^+} & [1 - \{(1 - W_{ij}) \prod_k (1 - W_{ik}.Wkj)\}] beta(1, \frac{1 - m_+}{m_+}) \\
\times \prod_{Y_{ij}=1 \wedge (i,j) \in D^-} & [1 - \{(1 - W_{ij}) \prod_k (1 - W_{ik}.Wkj)\}] beta(1, \frac{1 - m_-}{m_-}) \\
\times \prod_{Y_{ij}=1 \wedge (i,j) \in D^+} & [(1 - W_{ij}) \prod_k (1 - W_{ik}.Wkj)] beta(1, \frac{1 - m_+}{m_+}) \\
\times \prod_{Y_{ij}=1 \wedge (i,j) \in D^-} & [(1 - W_{ij}) \prod_k (1 - W_{ik}.Wkj)] beta(1, \frac{1 - m_-}{m_-}) dW
\end{aligned}
\tag{10}
$$

Notice that the W_{ij} terms affect each other while calculating the product terms. Since the above integral is extremely hard to evaluate (even under our simplifying assumptions), we first consider an example graph to understand the properties of the above integral. We shall use this example to demonstrate the method to evaluate the required probability $P(S^0|S^*)$.

Consider a graph G_3 with 3 nodes. We will also assume that all three nodes are connected with each other in the underlying true clustering. Also notice that since we have an undirected graph, we assume the matrix to be symmetric and focus on the upper triangular matrix only. Also, we use the following simplifying notation. Let $w_1 = W_{12}$, $w_2 = W_{13}$, $w_3 = W_{23}$, and $m = m_+$.

For graph G_3, eqn. 10 becomes:

$$
P(S^0|S^*) = \int_W \prod_{(i,j) \in \{w_1, w_2, w_3\}} [1 - \{(1 - W_{ij}) \prod_k (1 - W_{ik}.Wkj)\}] beta(1, \frac{1 - m}{m})
\tag{11}
$$

First, let us evaluate the product terms over (i,j).

$$
\begin{aligned}
\prod_{(i,j)} &= (w_1 + w_2 w_3 - w_1 w_2 w_3) \\
&\quad \times (w_2 + w_1 w_3 - w_2 w_1 w_3) \\
&\quad \times (w_3 + w_1 w_2 - w_3 w_1 w_2) \\
&= -w_1^3 w_2^3 w_3^3 + w_1^2 w_2^3 w_3^3 + w_1^3 w_2^2 w_3^3 - w_1 w_2^2 w_3^3 \\
&\quad - w_1^2 w_2 w_3^3 + w_1 w_2 w_3^3 + w_1^3 w_2^3 w_3^2 - w_1 w_2^3 w_3^2 \\
&\quad + w_1^2 w_3^2 + w_1^2 w_2^2 w_3^2 - w_1 w_2^2 w_3^2 + w_2^2 w_3^2 + w_1^2 w_2^2 \\
&\quad - w_1^3 w_2 w_3^2 - w_1^2 w_2^3 w_3 + w_1 w_2^3 w_3 - w_1^3 w_2^2 w_3 \\
&\quad - w_1^2 w_2^2 w_3 + w_1^3 w_2 w_3 + w_1 w_2 w_3 - w_1^2 w_2 w_3^2
\end{aligned}
$$

(12)

Observing this, we can see that the integral can be represented as a sum of products and it is bounded, because the positive terms are bounded by 1 and the negative terms are bounded by 0.

Each of the terms in the above product can be integrated separately in the following form, since W_{iij}'s are independent, given S^*.

$$\int f(w)g(w)h(w)dw = \int f(w)dw \int g(w)dw \int h(w)dw \tag{13}$$

Following general method illustrates how we can compute these integrals. Let us for now, take $beta(\alpha, \beta)$ and let $f(w) = w_{ij}^k$.

$$\begin{aligned}
\int W_{ij}^k beta(\alpha,\beta)dW &= \int \frac{W_{ij}^k}{Beta(\alpha,\beta)} W_{ij}^{(\alpha-1)}(1-W_{ij})^{(\beta-1)}dW \\
&= \frac{1}{Beta(\alpha,\beta)} \int W_{ij}^{(\alpha+k-1)}(1-W_{ij})^{(\beta-1)}dW \\
&= \frac{Beta(\alpha+k-1,\beta)}{Beta(\alpha,\beta)} \\
&= \frac{\frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)}}{\frac{\Gamma(\alpha+k+\beta)}{\Gamma(\alpha+k)\Gamma(\beta)}} \\
&= \frac{\alpha(\alpha+1)(\alpha+2)...(\alpha+k-1)}{(\alpha+\beta)(\alpha+\beta+1)(\alpha+\beta+2)..(\alpha+\beta+k-1)}
\end{aligned}$$

Using $\Gamma(\alpha+1) = \alpha\Gamma(\alpha)$. Hence, we get

$$\int W_{ij}^k beta(\alpha,\beta)dW_{ij} = \frac{\alpha(\alpha+1)(\alpha+2)...(\alpha+k-1)}{(\alpha+\beta)(\alpha+\beta+1)(\alpha+\beta+2)..(\alpha+\beta+k-1)} \tag{14}$$

In general, under the substitution of $\alpha = 1$, $\beta = \frac{1-m}{m}$, we get,

$$\int W_{ij}^k beta(\alpha,\beta)dW_{ij} = \frac{k!m^k}{\prod_{l=1}^{k-1}(1+km)} \tag{15}$$

Note that this general form is very useful to evaluate the integral for a large graph analytically. However, this computation is very elaborate as k increases and we would need to resort to approximation methods, as we shall see in the next section.

Using eqn. 14, we can evaluate the integral in our example, using $\alpha = 1$, $\beta = \frac{1-m}{m}$, $k = 0, 1, 2, 3$. Using eqns. 11, 12 and 14, we get

$$P(S^0|S^*) = \frac{-24m^9 - 156m^8 + 12m^7 + 239m^6 + 123m^5 + 21m^4 + m^3}{(1+2m)^3(1+m)^3} \tag{16}$$

This can be substituted back in eqn. 5 to get the expected error. As we can see, this solution is in closed form, however, given the nature of $P(Y_{ij}|W)$ for our simplistic algorithm, it is difficult to express the solution of the integral in general form. In the future, we would like to explore other versions of the algorithm that lead to a simpler form of the required integral.

Expected error at the end of a single query We now turn our attention to the expected error as the result of a single query. Note that, we can assume that with a probability p, the oracle returns the value 0 for W_{ij}. This case is not very useful for the current version of our partitioning algorithm, since all the terms containing W_{ij} vanish, and hence, it does not influence the decision on the surrounding edges. In the future, we would like to address this problem by using a slightly more sophisticated partitioning algorithm.

Consider the case when the oracle returns the value 1 for W_{ij}. In our example, we set the value of, say $w_1 = 1$.

The product term reduces to

$$\prod_{i,j} = w_2^2 + w_3^2 + 2w_2w_3 + w_2^2w_3^2 - 2w_2w_3^2 - 2w_2^2w_3 \tag{17}$$

Making the substitutions as before, we get

$$P(S^1|S^*) = \frac{6m^2 - 2m^4}{(1+m)^2} \tag{18}$$

Expected reduction in error We can now derive the expression for the expected reduction in error as follows. Using eqns. 5, 16 and 18

$$E(d_0 - d_1) = \sum_{S^0} P(S^0|S^*)d(S^0, S^*) - \sum_{S^1} P(S^1|S^*)d(S^1, S^*) \tag{19}$$

However, since we are summing over all possible S's, we can write this as

$$E(d_0 - d_1) = \sum_{S} [P(S^0|S^*) - P(S^1|S^*)]d(S, S^*) \tag{20}$$

For a given true clustering, we have now shown a method to derive the expected reduction in error after obtaining the true value of one edge weight. As mentioned before, we would like to work with a different partitioning scheme in the future to yield a simpler form of the above expectation.

2.5 Techniques for Approximation

As we saw in the previous section, for general graphs with a reasonable number of nodes, our technique becomes mathematically cumbersome, and even computationally intractable. Here we describe some possible directions for approximating the value of the required integral. We leave the analysis of the quality of these approximation techniques to our future work.

Beta for high powers Recall that in order to evaluate the integral in eqn. 10 we need to evaluate individual integrals as described in eqn. 15. For higher values of k, this calculation becomes more and more cumbersome. We used Matlab Symbolic package for these calculations and the analytical calculations soon became intractable. However, fig. 3 shows the plot of the function $f(k,m) = \frac{k!m^k}{\prod_{l=1}^{k-1}(1+km)}$ for increasing values of k. Notice that as k increases, this function tends to converge to a single curve, especially, for smaller and larger values of m (which can be interpreted as a reasonable probability gap or amount of uncertainty in the graph). This fact can be used to approximate integrals involving high values of k very effectively.

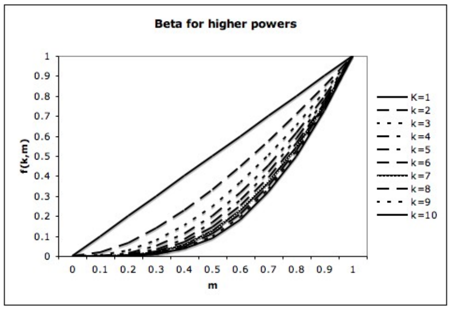


Fig. 3. Approximation by exploiting the nature of the integral function

Exchanging the product and integration Recall that the integral in 10 is an integral of a product. Another strategy for approximation is to exchange the product and integration. Of course, this makes the independence assumption that is not valid, but provides us with a technique that could prove to be very effective for large graphs, without losing precision. Fig. 4 shows the exact and approximate value of the integral for graphs of size 3 and 4 using this method. For larger graphs, we could not calculate the exact values due to computation limitations.

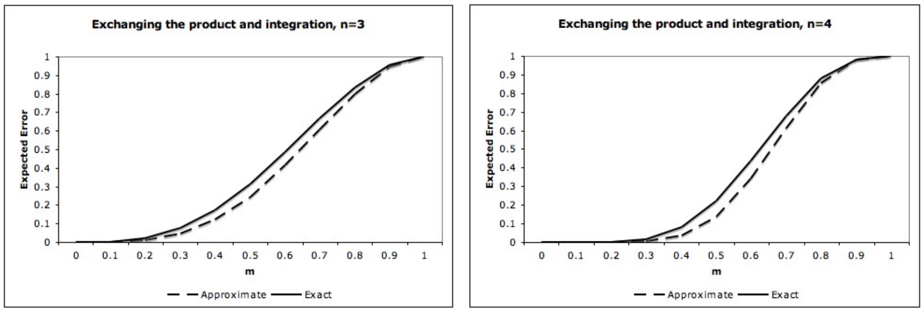


Fig. 4. Approximation by exchanging product and integration

Observing patterns in powers This method is useful for approximating the product terms by observing the pattern of terms under different types of edge weight combinations. We show this technique for our transitivity of length 2. We can either assume that this is adequate, or generalize this method in the future.

We first define the following numbers:

$$N_1 = \sum_{ij} \sum_{k} 1_A : A = \{W_{ik} = 1 \wedge W_{kj} = 1\} \tag{21}$$

$$N_2 = \sum_{ij} \sum_{k} 1_A : A = \{(W_{ik} = 1 \wedge W_{kj} = 0) \vee (W_{ik} = 0 \wedge W_{kj} = 1)\} \tag{22}$$

$$N_3 = \sum_{ij} \sum_{k} 1_A : A = \{W_{ik} = 0 \wedge W_{kj} = 0\} \tag{23}$$

Now, we observe that the multiplication term for $P(S|S^*)$ takes the form $(1-a^2)^{N_1} * (1-ab)^{N_2} * (1-b^2)^{N_3}$ after simplifications. This help us evaluate the required integral. This also gives us ideas for designing new heuristics for query selection, as discussed in the next section.

2.6 Analysis of Different Query Selection Criteria

The analysis so far estimates the expected reduction in error at the end of a single query, for any edge. We now present some ideas for estimating this quantity under different query selection criteria.

Let $P_{s_{ij}}$ be the probability that edge e_{ij} gets selected for querying under the given query selection criteria. Let us define the event Q_r to be the result of a query q. If an edge e_{ij} is selected as part of q, then $Q_r = \{W^1_{ij} = 1, W^1_{ij} = 0\}$, otherwise, some other edge in the graph is set to 0 or 1. Let p be the probability that the oracle gives a positive response to the selected query q, i.e. $p = P(W_{ij} = 1)$

$$
\begin{aligned}
E(P(Yij^1 = 1|W^1)) &= P_{s_{ij}} * \textstyle\sum_{Q_r} P(Y_{ij} = 1|W) + (1 - P_{s_{ij}}) * \textstyle\sum_{Q_r} P(Y_{ij} = 1|W) \\
&= P_{s_{ij}} * [p * 1 + (1-p)[1 - \textstyle\prod_k (1 - W_{ik}.Wkj)]] + (1 - P_{s_{ij}}) * \textstyle\sum_{Q_r} P(Y_{ij} = 1|W) \\
&= P_{s_{ij}} * [1 - (1-p) \textstyle\prod_k (1 - W_{ik}.Wkj)] + (1 - P_{s_{ij}}) * \textstyle\sum_{Q_r} P(Y_{ij} = 1|W)
\end{aligned}
$$

Simillarly,

$$
\begin{aligned}
E(P(Yij^1 = 0|W^1)) &= P_{s_{ij}} * \textstyle\sum_{Q_r} P(Y_{ij} = 0|W) + (1 - P_{s_{ij}}) * \textstyle\sum_{Q_r} P(Y_{ij} = 0|W) \\
&= P_{s_{ij}} * [(1-p) \textstyle\prod_k (1 - W_{ik}.Wkj)] + (1 - P_{s_{ij}}) * \textstyle\sum_{Q_r} P(Y_{ij} = 0|W)
\end{aligned}
$$

The case when the edge e_{ij} is not selected for querying is much more tricky, since there are two possibilities in that case. Either the edge selected for querying is incident on one of the nodes of e_{ij}, in which case, $P(Y_{ij} = 0|W)$ gets affected (under our two step transitivity), or some other edge in the graph gets selected, in which case, this probability is unaffected. We leave this complex analysis for our future work. However, here we show method to compute $P_{s_{ij}}$ for different query selection criteria and discuss some other heuristics to consider.

For random query selection criteria,

$$P_{s_{ij}} = \frac{1}{N} \tag{24}$$

For uncertainty based criteria,

$$P_{s_{ij}} = argmax_k \frac{-w_k log w_k}{\sum_k -w_k log w_k} \tag{25}$$

We now derive this for the expected entropy criteria, as proposed in our previous work [Kanani *et al.*2007],

We first define entropy of the graph G_0 as

$$H_0 = \sum_{ij} P(Y_{ij} = 1|W) \log P(Y_{ij} = 1|W) \tag{26}$$

We next define entropy of the graph after obtaining the value of W_{ij}. Depending on the two possible outcomes, we have,

$$H_p = \sum_{ij} P(Y_{ij} = 1|W, W_{ij} = 1) \log P(Y_{ij} = 1|W, W_{ij} = 1) \tag{27}$$

$$H_n = \sum_{ij} P(Y_{ij} = 1|W, W_{ij} = 0) \log P(Y_{ij} = 1|W, W_{ij} = 0) \tag{28}$$

We now define expected entropy of an edge e_{ij} as,

$$EE_{ij} = \frac{H_p + H_n}{2} \tag{29}$$

Finally, as per this query selection criterion, the probability of the edge getting selected is

$$P_{s_{ij}} = \frac{EE_{ij}}{\sum_{ij} EE_{ij}} \tag{30}$$

Other Query Selection Criteria We now propose some of the other heuristics to consider for selecting queries. Assuming that the path of length 2 is enough to establish transitivity in a fully connected graph, we can count the number of positive and negative edges incident on the two nodes to assign a score to each edge, which can then be used for selection (this is similar to the force of attraction in an electric field). The derivative heuristic works by taking the partial derivative of the weight matrix with respect to each edge weight. This is similar to doing sensitivity analysis. We leave the analysis of these criteria for our future work.

2.7 Simulation Experiments

Evaluating the expected reduction in error analytically, even for our example is hard, since we need to sum over all possible clusterings of the graph. Furthermore, we would like to compare the expected reduction in error for different strategies for selecting the edges. We would also like to extend these results to a general random graph. We turn to simulation technique to achieve these goals.

Experimental Setup We sample each graph using the Planted Partition Model described above, where each edge weight is drawn from a beta distribution, whose mean is determined by whether or not the corresponding vertices are merged in the true partitioning of the graph. We use 500 samples of graphs for each of these experiments.

We compare the uncertainty-based strategy for selecting the edge to query versus the random approach. For our experiments, we store the samples generated initially for baseline, select an edge using one of the strategies and modify the edge weight (to either 0 or 1) based on the true clustering.

We also extend some of the results to general graphs. The random graphs are generated as follows. We first randomly select the number of nodes between two and the specified maximum size. Next, we randomly select the number of clusters between one and the number of nodes divided by two. We assume the clusters are of uniform size, with the exception of singletons.

Comparison with Analytical Results We first compare the results of the theoretical analysis presented in the previous section to the estimated probability of recovering the true partition of the simple example graph against different values of m_+. We use arbitrary m_- in this experiment, since none of the edges come from $beta_2$. We see that after querying for one edge, we achieve most improvement when there is most uncertainty in the original graph.

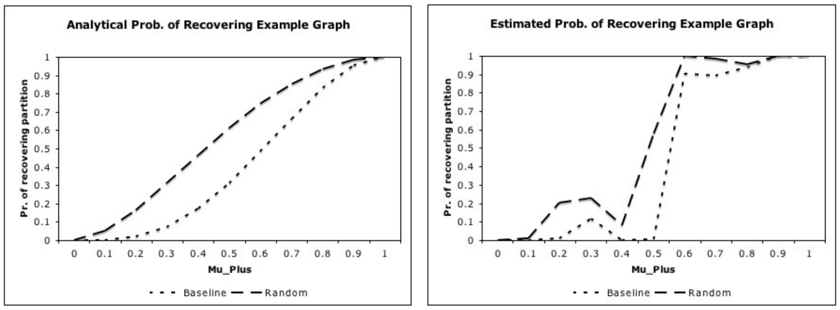


Fig. 5. Comparison of Analytical and Estimated Prob. of Recovering Partitioning of Example Graph

Estimating Expected Reduction in Error for Example Graph We now estimate the expected error before and after obtaining the results of a single query, as described in equation 20 for our example graph. Fig. 5 shows that when the underlying graph is very noisy, the uncertainty based approach is not very useful, in the case of our example graph. However, at higher values of m_+, it is profitable to query the edge with most uncertainty.

Note that we sum over all possible clusterings of the given graph to calculate the expected error. The number of possible partitionings for a graph with n nodes is given by the Bell number of n (See Note), which can be approximated to $n!$. Hence prohibitively expensive to generate all possible partitionings for large graphs. In future, we would like to estimate the expected reduction in error for general graphs.

Estimating Probability of Recovering True Partition in General Graph Since, we are not calculating the expected error, we estimate the probability of recovering the true partitioning of a random graph. In our experiments, we use the process described earlier to generate 10 random graph structures, with a max. size of 10 (which are then sampled 500 times).

Fig.7(a) shows this probability for different values of m_+. We set the value of m_- to 0 for this experiment. For values of m_+ greater than 0.5, the uncertainty based approach proves to be an effective strategy for improving performance. In fig.7(b), we average over several curves corresponding to different values of m_- . Each point on this curve is an average over different probability gaps in

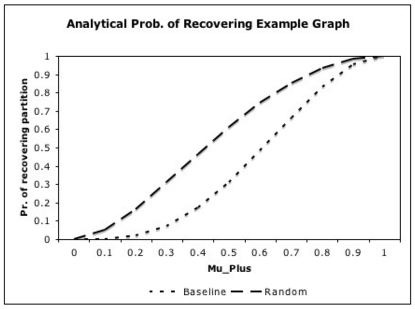
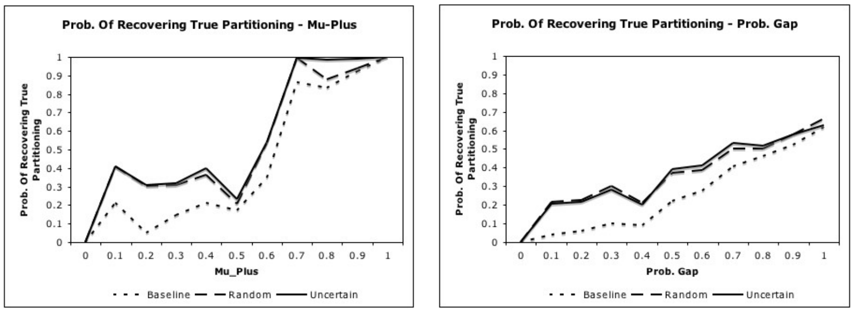


Fig. 6. Expected Error for Different Query Selection Strategies for Example Graph

the graph. As we can see, uncertainty based approach performs better than the random approach. In future, we would like to extend these results to multiple queries.



(a) Setting the value of m_- to 0 (b) Averaging over several values of m_-

Fig. 7. Prob. of Recovering Partitioning of General Graphs

2.8 A Note on True Clustering

In the future, we would like to integrate over all possible true clusterings of the graph. Currently, we do not do this because it is hard to assume the distribution of true clusterings. The obvious case of uniform distribution is easier to analyze, but very unlikely. One interesting question we can ask about this case is as follows :

Given an undirected, complete graph of n nodes, the total number of ways in which we can partition it is given by the Bell number, $B(n)$. (We assume disjoint, complete partitions). What is the total number of times that two nodes are merged in the same partition. For. e.g. for $n = 3$, there are 5 possible partitions.

The total number of times two nodes are merged together is (0+1+1+1+3 =)6. Hence, the ratio is $\frac{6}{15}$. Similarly, for $n = 4$, it is $\frac{30}{90}$.

In general, this quantity is related to the following integer sequence: Number of partitions of {1...n} containing 2 detached pairs of consecutive integers, i.e., partitions in which only 1- or 2-strings of consecutive integers can appear in a block and there are exactly two 2-strings [Munagi2005]. The required formula is $binomial(n, 2) * B(n-1)$ and the required ratio is $\frac{B(n-1)}{B(n)}$. This provides an interesting direction for future analysis.

3 Changing Edge Weights - Proving a Bound on Reduction in Error

In this section, we assume that the edge weights of the graph resulting at the end of a single query come from a different, but related distribution. In this case, we will prove an upper bound on the probability of obtaining at least ϵ improvement at the end of each query, as a function of the gap between the parameters of the planted partition model before and after obtaining new information.

3.1 Assumptions

This section uses a slightly different set of assumptions, as compared to the previous section. We assume that the two distributions from which the weight matrices are drawn before and after obtaining information are related to each other by a linear relation on their means. We also use the cost function of correlation clustering to estimate the true error. This is done on the basis of the error bound presented in [Joachims and Hopcroft2005], which provides the basis for our proof.

Integer Linear Programming Formulation of Correlation Clustering The correlation clustering $\hat{S}$ of a graph with edge weights W is given by solution $\hat{Y}$ of the following integer program given by Demaine, et. al.:

Let W be the adjacency matrix of a weighted, undirected graph, G, with n vertices. Let W^+ be the same as W, after replacing all negative edges by zero and let W^- be the same as W, after replacing all positive edges by zero. The optimization is over the nXn matrix Y with elements $Y_{ij} \in \{0, 1\}$. A value of 1 for Y_{ij} indicates that nodes indexed by i and j are in the same cluster. A value of 0 indicates they are in different clusters. That is, Y is the cluster indicator matrix.

$$\min_Y \sum_{i=1}^{n} \sum_{j=1}^{n} [(1 - Y_{ij})W_{ij}^{+} - Y_{ij} W_{ij}^{-}] \tag{31}$$

subject to

$$\forall i : Y_{ii} = 1 \tag{32}$$

$$\forall i, j : Y_{ij} = Y_{ji} \tag{33}$$

$$\forall i, j, k : Y_{ij} + Y_{jk} \leq Y_{ik} + 1 \tag{34}$$

$$\forall i, j : Y_{ij} \in \{0, 1\} \tag{35}$$

We assume there is an arbitrary "true" partition $S^* = \{S_1^*, ..., S_{k^*}^*\}$ of the vertices. i.e. $S_1^* \cup .... \cup S_{k^*}^* \;= X$ and $S_i^* \cap S_j^* = \emptyset$. Number of clusters k^* and the size of each cluster are arbitrary and unknown.

To this partition S^* corresponds a probability distribution $P_{S^*}(W)$ over edge weights. We assume that $P_{S^*}(W)$ is the process that generates the data we want to cluster. The goal of the clustering algorithm is to recover the true partition S^* underlying the data generating process $P_{S^*}(W)$ from a single realization of edge weights W

The Generalized Planted Partition Model *In a graph with n vertices, the edge weights are generated by a distribution*

$$P_{S^*}(W|M, a, b) = \prod_{i=1}^{n} \prod_{j=1}^{n} P_{S^*}(W_{ij}|M_{ij}, a, b) \tag{36}$$

So that each element W_{ij} of W is a bounded independent random variable in the interval $[a, b]$ with mean M_{ij}. Each $P_{S^*}(W_{ij}|M_{ij}, a, b)$ is constrained by the true partitioning S^* as follows. If $Y(S^*)_{ij} = 1$ (vertices i and j are in the same cluster), the mean M_{ij} of W_{ij} must fulfill the constraint that $M_{ij} \geq \mu_+ > 0$. If $Y(S^*)_{ij} = 0$ (vertices i and j are in different clusters), the mean M_{ij} of W_{ij} must fulfill $M_{ij} \leq \mu_- < 0$

Error Bounds on Correlation Clustering [Joachims and Hopcroft2005] proves the following error bound for correlation clustering.

Given the true partition S^* of n points, the probability that correlation clustering returns a partition () with $Err((), S^*) \geq \epsilon$ in the planted partition model with $\mu = \min\{\mu_+, -\mu_-\}$ and $a \leq W_{ij} \leq b$ is bounded by

$$P(Err((), S^*) \geq \epsilon) \leq e^{nln(n) - 2\epsilon n(n-1)\frac{\mu^2}{(b-a)^2}} \tag{37}$$

This bounds the probability of drawing a W so that correlation clustering returns a partition () which has an error $d((), S^*)$ greater than δ. This provides us with a basis to use the cost function of correlation clustering for the following proof.

3.2 Bound Proof

Let us first define the error of a partitioning () with respect to true clustering S^* using the following pair-wise loss function.

$$d((), S^*) = ||Y(()) - Y(S^*)||_F^2 \tag{38}$$

Here, $||.||_F$ denotes the Frobenius norm. The function d measures the number of pairs on which the two partitionings disagree. The following proof is on the similar lines of the upper bound on $d((), S^*)$ as presented in (Joachims).

Let $d_+(S, S^*) = \delta_+$ be the number of edges that are not cut in S^* but cut in S. Let $d_-(S, S^*) = \delta_-$ be the number of edges that are cut in S^* but not cut in S . Let D_+ and D_- be the corresponding sets of edges.

Consider the following true errors in the partitions before and after obtaining information. We use the subscript (or superscript) 0 to denote the variables related to the original graph and the corresponding correlation clustering. We use the subscript (or superscript) 1 to denote the corresponding variables after obtaining information.

$$d_0 = d(S_0, S^*) \tag{39}$$

$$d_1 = d(S_1, S^*) \tag{40}$$

We want to bound the probability $P(d_0 - d_1 \leq \epsilon)$. We will use the following Hoeffding's inequality[1], which bounds the deviation of a sum of independent and bounded random variables $X_k \in [a_i, b_i]$ from its mean:

$$P(\sum X_k - E(\sum X_k) \leq c) \leq e^{\frac{-2c^2}{\Sigma(b_i - a_i)^2}} \tag{41}$$

In our case, we set

$$\begin{aligned}
\sum X_k &= d_0 - d_1 \\
&= (\sum_{(i,j)\in D_+^0} W_{ij}^0 - \sum_{(i,j)\in D_-^0} W_{ij}^0) - (\sum_{(i,j)\in D_+^1} W_{ij}^1 - \sum_{(i,j)\in D_-^1} W_{ij}^1) \\
&= (\sum_{(i,j)\in D_+^0} W_{ij}^0 - \sum_{(i,j)\in D_+^1} W_{ij}^1) - (\sum_{(i,j)\in D_-^0} W_{ij}^0) - \sum_{(i,j)\in D_-^1} W_{ij}^1) \\
E(\sum X_k) &= (\sum_{(i,j)\in D_+^0} M_{ij}^0 - \sum_{(i,j)\in D_+^1} M_{ij}^1) - (\sum_{(i,j)\in D_-^0} M_{ij}^0) - \sum_{(i,j)\in D_-^1} M_{ij}^1) \\
&= (\delta_+^0 \mu_+^0 - \delta_+^1 \mu_+^1) - (\delta_-^0 \mu_-^0 - \delta_-^1 \mu_-^1)
\end{aligned}$$

Let m be the gap introduced by new information. Here, we are assuming a two sided model. We can even make it one sided at this point. I.e., we can say that only positive information is obtained from the external source and the negative edges remain unaltered. We leave this analysis to our future work. Now, we assume that the parameters of the two gen. planted partition models are related by the following relation:

[1] Note that Hoeffding's inequality applies to a sum of independent random variables.The W0's and W1's do not seem to be independent, but actually they are. The Mij_0 and Mij_1 are independent of each other. They are only related by their maximums, that is the μ's.

$$\mu_+^0 = \mu_+^1 - m_+ \tag{42}$$

$$\mu_-^0 = \mu_-^1 + m_- \tag{43}$$

Also, we use the following: $\mu_0 = \min\{\mu_+^0, -\mu_+^0\}$, $m = \min\{m_+, -m_-\}$, $\delta_0 = \delta_+^0 + \delta_-^0$, and $\delta_1 = \delta_+^1 + \delta_-^1$. Therefore,

$$\begin{aligned}
E(\sum X_k) &= (\delta_+^0\mu_+^0 - \delta_+^1\mu_+^1) - (\delta_-^0\mu_-^0 - \delta_-^1\mu_-^1) \\
&= (\delta_+^0\mu_+^0 - \delta_+^1(\mu_+^0 + m_+)) - (\delta_-^0\mu_-^0 - \delta_-^1(\mu_-^0 - m_-)) \\
&= (\delta_+^0\mu_+^0 - \delta_-^0\mu_-^0) - (\delta_+^1\mu_+^0 - \delta_-^1\mu_-^0) - (\delta_+^0 m_+ - \delta_-^0 m_-) \\
&= \mu_0(\delta_+^0 + \delta_-^0) - \mu_0(\delta_+^1 + \delta_-^1) - m(\delta_+^1 + \delta_-^1) \\
&= \mu_0\delta_0 - \mu_0\delta_1 - m\delta_1 \\
&= \mu_0(\delta_0 - \delta_1) - m\delta_1
\end{aligned}$$

Now, we use Hoeffding's inequality with the following values:

$$\begin{aligned}
c &= \epsilon - E(X_k) = \epsilon - \mu_0(\delta_0 - \delta_1) - m\delta_1 \\
\sum(b_i - a_i)^2 &= ((\delta_+^0 - \delta_+^1) + (\delta_-^0 - \delta_-^1))(b-a)^2 = (\delta_0 - \delta_1)(b-a)^2 \\
P(\sum X_k - E(\sum X_k) \le c) &\le e^{\frac{-2c^2}{\sum(b_i - a_i)^2}} \\
P(\sum X_k - E(\sum X_k) \le \epsilon - E(X_k)) &\le e^{\frac{-2[\epsilon - \mu_0(\delta_0 - \delta_1) - m\delta_1]^2}{(\delta_0 - \delta_1)(b-a)^2}} \\
P(\sum X_k \le \epsilon) &\le e^{\frac{-2[\epsilon - \mu_0(\delta_0 - \delta_1) - m\delta_1]^2}{(\delta_0 - \delta_1)(b-a)^2}} \\
P(d_0 - d_1 \le \epsilon) &\le e^{\frac{-2[\epsilon - \mu_0(\delta_0 - \delta_1) - m\delta_1]^2}{(\delta_0 - \delta_1)(b-a)^2}}
\end{aligned}$$

This gives us the required bound.

Fraction of the Queries Here is a general idea for analyzing the effect of a fraction of the queries. Under the planted partition model, M_{ij}'s are bounded by μ_+ and μ_-. When a subset of edges are queries, the corresponding M_{ij}'s turn into +1 or -1 and we can adjust $\sum M_{ij}$ accordingly. Hence our final error bound can shift based on the amount of queries made. We would like to explore this direction in our future work.

3.3 A Note about the Cost Functions

The cost function used in the above formulation and most of the correlation clustering literature [Joachims and Hopcroft2005], [Demaine and Immorlica2003], [Charikar *et al.*2005] is

$$cost'_W(Y) = \sum_{i=1}^{n}\sum_{j=1}^{n}[(1 - Y_{ij})W^+_{ij} - Y_{ij}W^-_{ij}] \tag{44}$$

However, following is a related, but slightly different cost function, which is smoother:

$$cost_W(Y) = \sum_{i=1}^{n}\sum_{j=1}^{n}[(1 - Y_{ij})W_{ij} - Y_{ij}W_{ij}] = \sum_{i=1}^{n}\sum_{j=1}^{n}[(1 - 2Y_{ij})W_{ij}] \tag{45}$$

The two functions are related to each other as shown below:

$$\begin{aligned}cost_W(Y) &= \sum_{i=1}^{n}\sum_{j=1}^{n}[(1 - Y_{ij})W_{ij} - Y_{ij}W_{ij}]\\ &= \sum_{i=1}^{n}\sum_{j=1}^{n}[(1 - Y_{ij})(W^+_{ij} + W^-_{ij}) - Y_{ij}(W^+_{ij} + W^-_{ij})]\\ &= \sum_{i=1}^{n}\sum_{j=1}^{n}[\{(1 - Y_{ij})W^+_{ij} - Y_{ij}W^-_{ij}\} + \{(1 - Y_{ij})W^-_{ij} - Y_{ij}W^+_{ij}\}]\end{aligned}$$

Hence, we note that $cost'_W(Y)$ = no. of positive edges that are cut - no. of negative edges that are not cut = total disagreements. On the other hand, $cost_W(Y) =$ [no. of positive edges that are cut − no. of negative edges that are not cut] + [no. of negative edges that are cut - no. of positive edges that are not cut] = total disagreements - total agreements. As noted in [Bansal *et al.*2002], at optimality, the solution that minimizes disagreements also maximizes agreements, and hence the solutions to the two different objective functions mentioned above would have the same solution. So, we may chose to work with $cost_W(Y)$ for analyzing RBIG as it is smoother and for the sake of simplicity. It is important to remember that their approximate solutions differ, and it would be interesting to study this as part of future research.

4 Expanding the Graph

In certain scenarios, the external information maybe available in the form of additional nodes, which can be incorporated in the graph, to reduce the error in partitioning the original graph. These additional nodes may represent the same kind of objects as the nodes in the original graph or they may be different. The new edges therefore, may also represent a different kind of relation. In either of the cases, the additional nodes help improve the accuracy by inducing a new partitioning over the nodes in the original graph due to the transitive nature of graph partitioning. In this section, we will see some efficient methods of incorporating additional information in the form of nodes under the given cost-benefit assumptions.

4.1 Assumptions

We first assume that the additional nodes are obtained by forming queries using some information available in the existing graph. Since there is a cost associated with each query, we would like to reduce the number of queries. Let P_k be one of the partitions in the underlying true clustering of the original graph. Let P'_k be one of the partitions in the true partitioning of the expanded graph, such that $P_k \subset P'_k$. We assume that a query formed by using information from nodes i and j, such that $i, j \in P_k$ will return nodes that belong to P'_k. Otherwise, we get an empty set.

Next, we assume that there exists an inexpensive way to evaluate a relation function between the additional node and the nodes in the original graph. We further assume that under this function, each additional node is related to only those nodes in the original graph that should be clustered together. However, it may be the case that it is related to only a subset of the nodes that belong to a true partition.

4.2 Selecting Queries

In many scenarios, issuing queries and obtaining the results is itself an expensive task. In this section, we will use the information available in the test data (upper left section of the matrix) to selectively issue queries, such that the results of those queries would have most impact on the accuracy of clustering. We use the assumptions described in the previous section.

In this approach, we start by partitioning the original graph using the graph partitioning algorithm of our choice. This gives us an initial guess of the partitions. We may then use one of the following methods to select queries. The nodes obtained as a result of these queries are then added to the graph, and we rerun the graph partitioning algorithm on the expanded graph.

Inter-cluster queries The first method for reducing the number of queries is to query only a subset of the edges between current partitions. For each cluster of vertices that have been assigned to the same partition under a given partitioning, we define the centroid as the vertex with the largest sum of weights to other members in its cluster. We connect all the centroids with each other and get a collection of queries, which are then used for querying. Let n be the number of nodes in the original graph and m be the number of currently predicted partitions. Under this scheme, we have reduced the number of queries to be executed from $O(n^2)$ to $O(m^2)$. A variation of this method picks multiple centroids, proportional to the size of each initial partition, where the proportion can be dictated by the amount of resources available.

Intra-cluster queries The second method for reducing the number of queries is to query only a subset of the edges within current partitions. For each initial partition, we select two most tightly connected citations to form a query. Under the same assumptions stated above, we have now reduced the number of queries to be executed from $O(n^2)$ to $O(m)$. A variation of this method picks more than two citations in each partition, including some random picks.

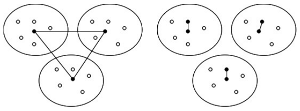


Fig. 8. Inter-cluster and Intra-cluster queries

Inter-cluster vs Intra-cluster queries Both these approaches are useful in different ways. Inter-cluster queries help find evidence that two clusters should be merged, whereas intra-cluster queries help find additional information about a hypothesized partition. The efficiency of these two methods depend on the number of underlying real partitions as well as the quality of initial partitioning. We are currently investigating the correlation between the performance of these query selection criteria with data characteristics, such as number of clusters, distribution of clusters and so on.

4.3 Selecting Nodes

Incorporating additional nodes in the graph can be expensive. Calculating new edge weights can incur high computational cost. Furthermore, the running time of most graph partitioning algorithms depend on the number of nodes in the graph. Hence, instead of adding all possible nodes, computing the corresponding edge weights and partitioning the resulting graph, it is desirable to find a minimal subset of the nodes that would help bring most of the nodes in the same clusters together. This is equivalent to selectively filling the entries of the upper right section of the matrix. We observe that this problem is similar to the classic Set-cover problem with some differences as noted below.

RBIG as Set-cover The standard Set-cover problem is defined as follows. Given a finite set U and a collection $C = \{S_1, S_2, ..., S_m\}$ of subsets of U. Find a minimum sized cover $C' \subseteq C$ such that every element of U is contained in at least one element of C'. It is known that greedy approach provides an $\Omega(\ln \text{n})$ approximation to this NP-Complete problem.

We now cast the problem of *Resource-bounded information gathering* using additional nodes as a variant of Set-cover. We use the assumptions mentioned before. The goal is to “cover” all the nodes in the original graph using the least possible number of new nodes, where ”covers” is defined by the inexpensive relation function. Under our assumptions, we can think of each new node as a

Algorithm 1 RBIG-Set-cover Algorithm

```
Input:
Set of nodes in the original graph $U$
Collection of new nodes $C : \{S_1, S_2, ..., S_n\}$
$O \Leftarrow \emptyset$
while $U$ is "coverable" by $C$ do
    $S_k \Leftarrow arg\max_{S_i \in C} |S_i|$
    $O \Leftarrow O \cup \{S_k\}$
    $U \Leftarrow U \cap S_k$
    $C \Leftarrow \{S_i | S_i = S_i \cap S_k\}$
end while
return $O$
$U$ is "coverable" by $C \equiv \exists_{(e \in U \wedge S_i \in C)}(e \in S_i)$
```

set of nodes in the original graph and the set of nodes in the original graph that should be clustered together as the set of elements to be covered. We now need to choose a minimal set of new nodes such that they can provide information about most of the nodes in the original graph.

There are some differences between Set-cover and our problem that reflect our assumptions as follows. There can be some elements in U which are not covered by any elements in C. That is, $\bigcup S_i \neq U$. Keeping this condition in mind, we modify the greedy solution to Set-cover as shown in Algorithm 1.

4.4 Hybrid Approach

We can also combine the two approaches, i.e. *Selecting Nodes* and *Selecting Queries* to form a hybrid approach. For example, we can first select queries using, say intra-cluster queries to obtain additional nodes. This would help reduce querying cost. We can then reduce the computation cost by selecting a subset of the these nodes using the Set-cover method.

4.5 Cost-Benefit Analysis

It should be noted that the choice of strategy for *Resource-bounded information gathering* in the case of expanded graph should be governed by a careful Cost-Benefit analysis of various parameters of the system. For example, if the cost of computing correct weights for the edges corresponding to the additional nodes is high, or if we are employing a graph partitioning technique that is heavily dependent on the number of nodes in the graph, then the Set-cover method described above would be effective in reducing the cost. On the other hand, if the cost of making a query and obtaining additional nodes is high, then using inter-cluster or intra-cluster methods is more desirable. For a large scale system, a hybrid of these methods could be more suitable.

5 Motivating Application

The problem described above is inspired by our work in author coreference. Here we are given a set of citations that mention similar author names, and must partition them by the true identity of the author. As in our previous work [Kanani *et al.*2007], we build a graph in which nodes represent author mentions. The edge weights indicate the strength of our belief that two mentions refer to the same real author, and are estimated by a binary logistic regression classifier that uses features such as title, co-author overlap, etc. Note that, each partition should represent the set of mentions that correspond to the same real author.

Experimentally, we have shown significant accuracy improvement by making queries of type Q1 and Q2. In our case, we issue the queries to the web. We incorporate the results of the queries either as additional features or as additional nodes in the graph. For example, we can form a query by joining the titles of two citations and issuing it to a search engine API. A hit indicates the presence of a document on the web that contains both of these citations and hence provides some evidence that they are authored by the same person. The result of the query is translated into a binary input feature to our classifier and is used to update the weight on the corresponding edge. The problem is resource bounded because for a fully connected graph, obtaining additional feature value for every pair of mentions is prohibitively expensive.

Similarly, we can add nodes corresponding to documents obtained by web queries. Note that these web documents represent author mentions and help improve accuracy by transitivity. For example, the additional node could be the list of publications or CV of one of the authors and would show strong affinity towards several nodes in the original graph. Hence, by transitivity, applying graph partitioning on this expanded graph leads to improvement in accuracy. However, since the web is too large to incorporate all its data, we need an efficient procedure for selecting a subset of web queries and resulting documents. We have applied the various approaches described in the previous section in [Kanani and McCallum]. In this work we have shown significant reduction in costs experimentally.

6 Related Work

In [Kanani *et al.*2007], we propose an approach to resource bounded information gathering based on expected entropy, in which we use web information as an additional feature. We also propose centroid-based methods in which we add nodes to the graph.

Learning and inference under resource limitations has been studied in various forms, including resource-bounded reasoning and the value of information [Grass and Zilberstein2000], [Lesser *et al.*2000], [Provost *et al.*2007], [Krause and Guestrin], active feature value acquisition [Bilgic and Getoor2007], [Zhu and Wu2004], budgeted learning, [Kapoor and Greiner2005], [Crammer *et al.*2004], semi-supervised learning [Zhu2005], and active learning, [Roy and McCallum2001], [Balcan *et al.*2007].

There exists a vast amount of literature in machine learning as well as graph theory on graph partitioning. There has been a lot of interest in correlation clustering [Bansal *et al.*2002] [Demaine and Immorlica2003], [Charikar *et al.*2005] in the theory community. Spectral methods [Shi and Malik2000], [Ng *et al.*2001], [Bach and Jordan2003] are popular with interesting probabilistic interpretations [Meila and Shi2001]. There has also been some amount of work in learning graph partitioning under resource constraints [Hofmann and Buhmann1998],[Engelberg *et al.*2007].

Other alternative frameworks to approach this problem are budgeted multicut, online graph partitioning, clustering with soft group constraints[Law *et al.*], query based learning [Angluin *et al.*2007], property testing of graphs [Goldreich and Ron1997], random and evolving graphs [Gaertler *et al.*2006] and sensitivity analysis.

7 Conclusion

We show that measuring the impact of a small change in the graph on the overall graph partitioning is an extremely challenging problem. Even under our simplifying assumptions, it is very difficult to get a general expression for the expected reduction in error. However, we show that it is possible to derive a closed form solution and demonstrate a method to do so. We also discuss some interesting directions for approximation. Our analysis of different query selection criteria provides a formal way of comparing different heuristics. We compare the solution of our theoretical analysis with simulation results. We also perform simulations to estimate the probability of recovering the true partition under various query strategies for general random graphs and find that uncertainty based approach works better than a random approach for selecting queries for a reasonable amount of uncertainty in the original graph. We also show a related bound under a different set of assumptions. Finally, we describe some general techniques for the case of expanded graphs. This work is an initial step in the direction of better theoretical understanding of the problem and opens up many possible interesting directions for future work.

8 Acknowledgments

We thank Avrim Blum, Chris Pal, Sridhar Mahadevan, Arnold Rosenberg, Paul Gunnels, Gideon Mann, Siddharth Srivastava, Kedar Bellare, Xuerui Wang, Katrina Ligett, Brian Milch and Lev Reyzin for useful discussions. A special thank you to all the other members of IESL for their ideas, suggestions and feedback. Supported in part by the CIIR, CIA, NSA and NSF under grant #IIS-0326249 and in part by DoD contract #HM1582-06-1-2013.

References

[Angluin *et al.*2007] Dana Angluin, James Aspnes, Jiang Chen, and Lev Reyzin. Learning large-alphabet and analog circuits with value injection queries. In Nader H.

Bshouty and Claudio Gentile, editors, *COLT*, volume 4539 of *Lecture Notes in Computer Science*, pages 51–65. Springer, 2007.

[Bach and Jordan2003] F. Bach and M. Jordan. Learning spectral clustering, 2003.

[Balcan *et al.*2007] Maria-Florina Balcan, Andrei Z. Broder, and Tong Zhang. Margin based active learning. In *COLT*, pages 35–50, 2007.

[Bansal *et al.*2002] N. Bansal, S. Chawla, and A. Blum. Correlation clustering. *In The 43rd Annual Symposium on Foundations of Computer Science (FOCS)*, pages 238–247, 2002.

[Bilgic and Getoor2007] Mustafa Bilgic and Lise Getoor. Voila: Efficient feature-value acquisition for classification. In *AAAI '07: Proceedings of the 22nd National Conference on Artificial Intelligence*, July 2007.

[Charikar *et al.*2005] Moses Charikar, Venkatesan Guruswami, and Anthony Wirth. Clustering with qualitative information. *J. Comput. Syst. Sci.*, 71(3):360–383, 2005.

[Crammer *et al.*2004] K. Crammer, J. Kandola, and Y. Singer. Online classification on a budget, 2004.

[Demaine and Immorlica2003] Erik D. Demaine and Nicole Immorlica. Correlation clustering with partial information. In *RANDOM-APPROX*, page 1, 2003.

[Engelberg *et al.*2007] Roee Engelberg, Jochen Könemann, Stefano Leonardi, and Joseph (Seffi) Naor. Cut problems in graphs with a budget constraint. *J. of Discrete Algorithms*, 5(2):262–279, 2007.

[Gaertler *et al.*2006] Marco Gaertler, Robert Görke, Dorothea Wagner, and Silke Wagner. How to Cluster Evolving Graphs. In *Proceedings of the European Conference of Complex Systems ECCS*, September 2006.

[Goldreich and Ron1997] Oded Goldreich and Dana Ron. Property testing in bounded degree graphs. pages 406–415, 1997.

[Grass and Zilberstein2000] J. Grass and S. Zilberstein. A value-driven system for autonomous information gathering. *Journal of Intelligent Information Systems*, 14:5–27, 2000.

[Hofmann and Buhmann1998] Thomas Hofmann and Joachim M. Buhmann. Active data clustering. In *NIPS '97: Proceedings of the 1997 conference on Advances in neural information processing systems 10*, pages 528–534, Cambridge, MA, USA, 1998. MIT Press.

[Joachims and Hopcroft2005] Thorsten Joachims and John Hopcroft. Error bounds for correlation clustering. In *ICML '05: Proceedings of the 22nd international conference on Machine learning*, pages 385–392, New York, NY, USA, 2005. ACM.

[Kanani and McCallum] Pallika Kanani and Andrew McCallum. Efficient strategies for improving partitioning-based author coreference by incorporating web pages as graph nodes. In *Workshop on Information Integration on the Web (IIWEB 07)*.

[Kanani and McCallum2007] Pallika Kanani and Andrew McCallum. Resource-bounded information gathering for correlation clustering. In *Computational Learning Theory 07, Open Problems Track, COLT 2007*, pages 625–627, 2007.

[Kanani *et al.*2007] Pallika Kanani, Andrew McCallum, and Chris Pal. Improving author coreference by resource-bounded information gathering from the web. In *Proceedings of IJCAI*, 2007.

[Kapoor and Greiner2005] Aloak Kapoor and Russell Greiner. Learning and classifying under hard budgets. In *ECML*, pages 170–181, 2005.

[Krause and Guestrin] Andreas Krause and Carlos Guestrin. Near-optimal nonmyopic value of information in graphical models.

[Law *et al.*] Martin Law, Alexander Topchy, and Anil K. Jain. Clustering with soft and group constraints.

[Lesser *et al.*2000] Victor R. Lesser, Bryan Horling, Frank Klassner, Anita Raja, Thomas Wagner, and Shelley XQ Zhang. Big: An agent for resource-bounded information gathering and decision making. *Artif. Intell*, 118:197, 2000.

[Meila and Shi2001] M. Meila and J. Shi. A random walks view of spectral segmentation, 2001.

[Munagi2005] A. O. Munagi. Set partitions with successions and separations. *Int. J. Math and Math. Sc. 2005, no. 3*, pages 451–463, 2005.

[Ng *et al.*2001] A. Ng, M. Jordan, and Y. Weiss. On spectral clustering: Analysis and an algorithm, 2001.

[Provost *et al.*2007] Foster Provost, Prem Melville, and Maytal Saar-Tsechansky. Data acquisition and cost-effective predictive modeling: targeting offers for electronic commerce. In *ICEC '07: Proceedings of the ninth international conference on Electronic commerce*, pages 389–398, New York, NY, USA, 2007. ACM.

[Roy and McCallum2001] Nicholas Roy and Andrew McCallum. Toward optimal active learning through sampling estimation of error reduction. In *Proc. 18th International Conf. on Machine Learning*, pages 441–448. Morgan Kaufmann, 2001.

[Shi and Malik2000] Jianbo Shi and Jitendra Malik. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):888–905, 2000.

[Zhu and Wu2004] Xingquan Zhu and Xindong Wu. Data acquisition with active and impact-sensitive instance selection. In *ICTAI '04: Proceedings of the 16th IEEE International Conference on Tools with Artificial Intelligence (ICTAI'04)*, pages 721–726, Washington, DC, USA, 2004. IEEE Computer Society.

[Zhu2005] Xiaojin Zhu. Semi-supervised learning literature survey. Technical Report 1530, Computer Sciences, University of Wisconsin-Madison, 2005.