Correlation Clustering [*]

Nikhil Bansal (nikhil@cs.cmu.edu), Avrim Blum (avrim@cs.cmu.edu)
and Shuchi Chawla (shuchi@cs.cmu.edu)
Computer Science Department, Carnegie Mellon University, Pittsburgh, PA 15213, USA.

2003/10/27

Abstract. We consider the following clustering problem: we have a complete graph on n vertices (items), where each edge (u,v) is labeled either $+$ or $-$ depending on whether u and v have been deemed to be similar or different. The goal is to produce a partition of the vertices (a clustering) that agrees as much as possible with the edge labels. That is, we want a clustering that maximizes the number of $+$ edges within clusters, plus the number of $-$ edges between clusters (equivalently, minimizes the number of disagreements: the number of $-$ edges inside clusters plus the number of $+$ edges between clusters). This formulation is motivated from a document clustering problem in which one has a pairwise similarity function f learned from past data, and the goal is to partition the current set of documents in a way that correlates with f as much as possible; it can also be viewed as a kind of "agnostic learning" problem.

An interesting feature of this clustering formulation is that one does not need to specify the number of clusters k as a separate parameter, as in measures such as k-median or min-sum or min-max clustering. Instead, in our formulation, the optimal number of clusters could be any value between 1 and n, depending on the edge labels. We look at approximation algorithms for both minimizing disagreements and for maximizing agreements. For minimizing disagreements, we give a constant factor approximation. For maximizing agreements we give a PTAS, building on ideas of Goldreich, Goldwasser and Ron (1998) and de la Vega (1996). We also show how to extend some of these results to graphs with edge labels in $[-1,+1]$, and give some results for the case of random noise.

Keywords: clustering, approximation algorithm, document classification

1. Introduction

Suppose that you are given a set of n documents to cluster into topics. Unfortunately, you have no idea what a "topic" is. However, you have at your disposal a classifier $f(A,B)$ that given two documents A and B, outputs whether or not it believes A and B are similar to each other. For example, perhaps f was learned from some past training data. In this case, a natural approach to clustering is to apply f to every pair of documents in your set, and then to find the clustering that agrees as much as possible with the results.

Specifically, we consider the following problem. Given a fully-connected graph G with edges labeled "$+$" (similar) or "$-$" (different), find a partition of the vertices into clusters that agrees as much as possible with the edge labels. In particular, we can look at this in terms of maximizing *agreements*

[*] This research was supported in part by NSF grants CCR-0085982, CCR-0122581, CCR-0105488, and an IBM Graduate Fellowship.

(the number of $+$ edges inside clusters plus the number of $-$ edges between clusters) or in terms of minimizing *disagreements* (the number of $-$ edges inside clusters plus the number of $+$ edges between clusters). These two are equivalent at optimality but, as usual, differ from the point of view of approximation. In this paper we give a constant factor approximation to the problem of minimizing disagreements, and a PTAS[1] for maximizing agreements. We also extend some of our results to the case of real-valued edge weights.

This problem formulation is motivated in part by a set of clustering problems at Whizbang Labs in which learning algorithms were trained to help with various clustering tasks (Cohen and McCallum, 2001, Cohen and Richman, 2001, Cohen and Richman, 2002). An example of one such problem, studied by Cohen and Richman (2001, 2002) is clustering entity names. In this problem, items are entries taken from multiple databases (e.g., think of names/affiliations of researchers), and the goal is to do a "robust uniq" — collecting together the entries that correspond to the same entity (person). E.g., in the case of researchers, the same person might appear multiple times with different affiliations, or might appear once with a middle name and once without, etc. In practice, the classifier f typically would output a probability, in which case the natural edge label is log(Pr(same)/Pr(different)). This is 0 if the classifier is unsure, positive if the classifier believes the items are more likely in the same cluster, and negative if the classifier believes they are more likely in different clusters. The case of $\{+,-\}$ labels corresponds to the setting in which the classifier has equal confidence about each of its decisions.

What is interesting about the clustering problem defined here is that unlike most clustering formulations, we do not need to specify the number of clusters k as a separate parameter. For example, in k-median (Charikar and Guha, 1999, Jain and Vazirani, 1999) or min-sum clustering (Schulman, 2000) or min-max clustering (Hochbaum and Shmoys, 1986), one can always get a perfect score by putting each node into its own cluster — the question is how well one can do with only k clusters. In our clustering formulation, there is just a single objective, and the optimal clustering might have few or many clusters: it all depends on the edge labels.

To get a feel for this problem, notice that if there exists a perfect clustering, i.e., one that gets all the edges correct, then the optimal clustering is easy to find: just delete all "$-$" edges and output the connected components of the graph remaining. (This is called the "naive algorithm" in (Cohen and Richman, 2002).) Thus, the interesting case is when no clustering is perfect. Also, notice that for any graph G, it is trivial to produce a clustering that agrees with at least *half* of the edge labels: if there are more $+$ edges than $-$ edges, then simply put all vertices into one big cluster; otherwise, put each vertex into its own

[1] A PTAS (polynomial-time approximation scheme) is an algorithm that for any given *fixed* $\varepsilon > 0$ runs in polynomial time and returns an approximation within a $(1+\varepsilon)$ factor. Running time may depend exponentially (or worse) on $1/\varepsilon$, however.

cluster. This observation means that for maximizing agreements, getting a 2-approximation is easy (note: we will show a PTAS). In general, finding the optimal clustering is NP-hard (shown in Section 3).

Another simple fact to notice is that if the graph contains a triangle in which two edges are labeled $+$ and one is labeled $-$, then no clustering can be perfect. More generally, the number of edge-disjoint triangles of this form gives a lower bound on the number of disagreements of the optimal clustering. This fact is used in our constant-factor approximation algorithm.

For maximizing agreements, our PTAS is quite similar to the PTAS developed by de la Vega (1996) for MAXCUT on dense graphs, and related to PTASs of Arora et al. (1995, 1996). Notice that since there must exist a clustering with at least $n(n-1)/4$ agreements, this means it suffices to approximate agreements to within an additive factor of εn^2. This problem is also closely related to work on testing graph properties of (Goldreich et al., 1998, Parnas and Ron, 1999, Alon et al., 1999). In fact, we show how we can use the General Partition Property Tester of Goldreich, Goldwasser, and Ron (1998) as a subroutine to get a PTAS with running time $O(ne^{O((\frac{1}{\varepsilon})^{\frac{1}{\varepsilon}})})$. Unfortunately, this is doubly exponential in $\frac{1}{\varepsilon}$, so we also present an alternative direct algorithm (based more closely on the approach of (de la Vega, 1996)) that takes only $O(n^2 e^{O(\frac{1}{\varepsilon})})$ time.

Relation to agnostic learning: One way to view this clustering problem is that edges are "examples" (labeled as positive or negative) and we are trying to represent the target function f using a hypothesis class of vertex clusters. This hypothesis class has limited representational power: if we want to say (u,v) and (v,w) are positive in this language, then we have to say (u,w) is positive too. So, we might not be able to represent f perfectly. This sort of problem — trying to find the (nearly) best representation of some arbitrary target f in a given limited hypothesis language — is sometimes called *agnostic* learning (Kearns et al., 1994, Ben-David et al., 2001). The observation that one can trivially agree with at least half the edge labels is equivalent to the standard machine learning fact that one can always achieve error at most $1/2$ using either the *all positive* or *all negative* hypothesis.

Our PTAS for approximating the number of agreements means that if the optimal clustering has error rate ν, then we can find one of error rate at most $\nu+\varepsilon$. Our running time is exponential in $1/\varepsilon$, but this means that we can achieve any constant error gap in polynomial time. What makes this interesting from the point of view of agnostic learning is that there are very few problems where agnostic learning can be done in polynomial time.[2] Even

[2] Not counting trivial cases, like finding the best linear separator in a 2-dimensional space, that have only polynomially-many hypotheses to choose from. In these cases, agnostic learning is easy since one can just enumerate them all and choose the best.

for simple classes such as conjunctions and disjunctions, no polynomial-time algorithms are known that give even an error gap of $1/2-\varepsilon$.

Organization of this paper: We begin by describing notation in Section 2. In Section 3 we prove that the clustering problem defined here is NP complete. Then we describe a constant factor approximation algorithm for minimizing disagreements in Section 4. In Section 5, we describe a PTAS for maximizing agreements. In Section 6, we present simple algorithms and motivation for the random noise model. Section 7 extends some of our results to the case of real-valued edge labels. Finally, subsequent work by others is briefly described in Section 8.

2. Notation and Definitions

Let $G=(V,E)$ be a complete graph on n vertices, and let $e(u,v)$ denote the label ($+$ or $-$) of the edge (u,v). Let $N^+(u)=\{u\}\cup\{v:e(u,v)=+\}$ and $N^-(u)=\{v:e(u,v)=-\}$ denote the positive and negative neighbors of u respectively.

We let OPT denote an optimal clustering on this graph. In general, for a clustering $\mathcal{C}$, let $\mathcal{C}(v)$ be the set of vertices in the same cluster as v. We will use A to denote the clustering produced by our algorithms.

In a clustering $\mathcal{C}$, we call an edge (u,v) a mistake if either $e(u,v)=+$ and yet $u\notin\mathcal{C}(v)$, or $e(u,v)=-$ and $u\in\mathcal{C}(v)$. When $e(u,v)=+$, we call the mistake a *positive mistake*, otherwise it is called a *negative mistake*. We denote the total number of mistakes made by a clustering $\mathcal{C}$ by $m_{\mathcal{C}}$, and use m_{OPT} to denote the number of mistakes made by OPT.

For positive real numbers x, y and z, we use $x\in y\pm z$ to denote $x\in[y-z,y+z]$. Finally, let $\overline{X}$ for $X\subseteq V$ denote the complement $(V\setminus X)$.

3. NP-completeness

In this section, we will prove that the problem of minimizing disagreements, or equivalently, maximizing agreements, is NP-complete. It is easy to see that the decision version of this problem (viz. Is there a clustering with at most z disagreements?) is in NP since we can easily check the number of disagreements given a clustering. Also, if we allow arbitrary weights on edges with the goal of minimizing *weighted* disagreements, then a simple reduction from the Multiway Cut problem proves NP-hardness – simply put a $-\infty$-weight edge between every pair of terminals, then the value of the multiway cut is equal to the value of weighted disagreements. We use this reduction to give a hardness of approximation result for the weighted case in Section 7.

We give a proof of NP hardness for the *unweighted* case by reducing the problem of Partition into Triangles (GT11 in (Garey and Johnson, 2000)) to the problem of minimizing disagreements. The reader who is not especially interested in NP-completeness proofs should feel free to skip this section.

The Partition into Triangles problem is described as follows: Given a graph G with $n = 3k$ vertices, does there exist a partition of the vertices into k sets $V_1, \ldots, V_k$, such that for all i, $|V_i| = 3$ and the vertices in V_i form a triangle.

Given a graph $G = (V, E)$, we first transform it into a complete graph G' on the same vertex set V. An edge in G' is weighted +1 if it is an edge in G and -1 otherwise.

Let A be an algorithm that given a graph outputs a clustering that minimizes the number of mistakes. First notice that if we impose the additional constraint that all clusters produced by A should be of size at most 3, then given the graph G', the algorithm will produce a partition into triangles if the graph admits one. This is because if the graph admits a partition into triangles, then the clustering corresponding to this triangulation has no negative mistakes, and any other clustering with clusters of size at most 3 has more positive mistakes than this clustering. Thus we could use such an algorithm to solve the Partition into Triangles problem.

We will now design a gadget that forces the optimal clustering to contain at most 3 vertices in each cluster. In particular, we will augment the graph G' to a larger complete graph H, such that in the optimal clustering on H, each cluster contains at most 3 vertices from G'.

The construction of H is as follows: In addition to the vertices and edges of G', for every 3-tuple $\{u, v, w\} \subset G'$, H contains a clique $C_{u,v,w}$ containing n^6 vertices. All edges inside these cliques have weight $+1$. Edges between vertices belonging to two different cliques have weight -1. Furthermore, for all $u, v, w \in G'$ each vertex in $C_{u,v,w}$ has a positive edge to u, v and w, and a negative edge to all other vertices in G'.

Now assume that G admits a triangulation and let us examine the behavior of algorithm A on graph H. Let $N = n^6 \binom{n}{3}$.

LEMMA 1. *Given H as input, in any clustering that A outputs, every cluster contains at most three vertices of G'.*

Proof: First consider a clustering $\mathcal{C}$ of the following form:

1. There are $\binom{n}{3}$ clusters.
2. Each cluster contains exactly one clique $C_{u,v,w}$ and some vertices of G'.
3. Every vertex $u \in G'$ is in the same cluster as $C_{u,v,w}$ for some v and w.

In any such clustering, there are no mistakes among edges between cliques. The only mistakes are between vertices of G' and the cliques, and those between the vertices of G'. The number of mistakes of this clustering is at most $n^7(\binom{n}{2}-1)+\binom{n}{2}$ because each vertex in G' has n^6 positive edges to $\binom{n}{2}$ cliques and is clustered with only one of them.

Now consider a clustering in which some cluster has four vertices in G', say, u, v, w and y. We show that this clustering has at least $n^7(\binom{n}{2}-1)+\frac{n^6}{2}$ mistakes. Call this clustering X. Firstly, without loss of generality we can assume that each cluster in X has size at most n^6+n^4, otherwise there are at least $\Omega(n^{10})$ negative mistakes within a cluster. This implies that each vertex in G' makes at least $\binom{n}{2}n^6-(n^6+n^4)$ positive mistakes. Hence the total number of positive mistakes is at least $n^7(\binom{n}{2}-1)-n^5$. Let X_u be the cluster containing vertex $u,v,w,y \in G'$. Since X_u has at most n^6+n^4 vertices, at least one of u,v,w,y will have at most n^4 positive edges inside X_u and hence will contribute at least an additional n^6-n^4 negative mistakes to the clustering. Thus the total number of mistakes is at least $(\binom{n}{2}-1)n^7-n^5+n^6-n^4 \geq n^7(\binom{n}{2}-1)+n^6/2$. Thus the result follows. □

The above lemma shows that the clustering produced by A will have at most 3 vertices of G in each cluster. Thus we can use the algorithm A to solve the Partition into Triangles problem and the reduction is complete.

4. A Constant Factor Approximation for Minimizing Disagreements

As a warm-up to the general case, we begin by giving a very simple 3-approximation to the best clustering containing two clusters. That is, if the best two-cluster partition of the graph has x mistakes, then the following algorithm will produce one with at most $3x$ mistakes.

Let $\mathrm{OPT}(2)$ be the best clustering containing two clusters, and let the corresponding clusters be $\mathcal{C}_1$ and $\mathcal{C}_2$. Our algorithm simply considers all clusters of the form $\{N^+(v), N^-(v)\}$ for $v \in V$. Of these, it outputs the one that minimizes the number of mistakes.

THEOREM 2. *The number of mistakes of the clustering output by the algorithm stated above is at most $m_A \leq 3m_{\mathrm{OPT}(2)}$.*

Proof: Let's say an edge is "bad" if $\mathrm{OPT}(2)$ disagrees with it, and define the "bad degree" of a vertex to be the number of bad edges incident to it. Clearly, if there is a vertex that has no bad edges incident to it, the clustering produced by that vertex would be the same as $\{\mathcal{C}_1, \mathcal{C}_2\}$, and we are done with as many mistakes as $m_{\mathrm{OPT}(2)}$.

Otherwise, let v be a vertex with minimum bad degree d, and without loss of generality, let $v \in \mathcal{C}_1$. Consider the partition $\{N^+(v), N^-(v)\}$. Let X be the set of bad neighbors of v – the d vertices that are in the wrong set of the partition with respect to $\{\mathcal{C}_1, \mathcal{C}_2\}$. The total number of extra mistakes due to this set X (other than the mistakes already made by OPT) is at most dn. However, since all vertices have bad degree at least d, $m_{\text{OPT}(2)} \geq nd/2$. So, the number of extra mistakes made by taking the partition $\{N^+(v), N^-(v)\}$ is at most $2m_{\text{OPT}(2)}$. This proves the theorem. □

We now describe our main algorithm: a constant-factor approximation for minimizing the number of disagreements.

The high-level idea of the algorithm is as follows. First, we show (Lemma 3 and 4) that if we can cluster a portion of the graph using clusters that each look sufficiently "clean" (Definition 1), then we can charge off the mistakes made within that portion to "erroneous triangles": triangles with two $+$ edges and one $-$ edge. Furthermore, we can do this in such a way that the triangles we charge are nearly edge-disjoint, allowing us to bound the number of these mistakes by a constant factor of OPT. Second, we show (Lemma 6) that there must exist a nearly optimal clustering OPT$'$ in which all non-singleton clusters are "clean". Finally, we show (Theorem 7 and Lemma 11) that we can algorithmically produce a clustering of the entire graph containing only clean clusters and singleton clusters, such that mistakes that have an endpoint in singleton clusters are bounded by OPT$'$, and mistakes with both endpoints in clean clusters are bounded using Lemma 4.

We begin by showing a lower bound for OPT. We call a triangle "erroneous" if it contains two positive edges and one negative edge. A fractional packing of erroneous triangles is a set of erroneous triangles $\{T_1, \cdots, T_m\}$ and positive real numbers r_i associated with each triangle T_i, such that for any edge $e \in E$, $\sum_{e \in T_i} r_i \leq 1$.

LEMMA 3. *Given any fractional packing of erroneous triangles* $\{r_1, \cdots, r_m\}$, *we have* $\sum_i r_i \leq$ OPT.

Proof: Let M be the set of mistakes made by OPT. Then, $m_{\text{OPT}} = \sum_{e \in M} 1 \geq \sum_{e \in M} \sum_{e \in T_i} r_i$, by the definition of a fractional packing. So we have $m_{\text{OPT}} \geq \sum_i |M \cap T_i| r_i$. Now, for each T_i, we must have $|M \cap T_i| \geq 1$, because OPT must make at least one mistake on each erroneous triangle. This gives us the result. □

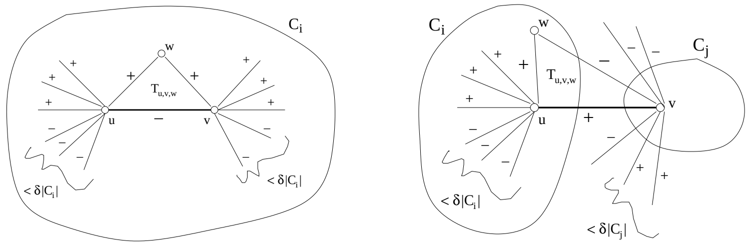
Next we give a definition of a "clean" cluster and a "good" vertex.

DEFINITION 1. *A vertex v is called* δ-**good** *with respect to C, where $C \subseteq V$, if it satisfies the following:*

– $|N^+(v) \cap C| \geq (1-\delta)|C|$

– $|N^+(v) \cap (V \setminus C)| \leq \delta |C|$

If a vertex v is not δ-good with respect to (w.r.t.) C, then it is called δ-**bad** *w.r.t. C. Finally, a set C is* δ-**clean** *if all $v \in C$ are δ-good w.r.t. C.*



(a) Erroneous triangles for negative mistakes (b) Erroneous triangles for positive mistakes

Figure 1. Construction of a triangle packing for Lemma 4

We now present two key lemmas.

LEMMA 4. *Given a clustering of V in which all clusters are δ-clean for some $\delta \leq 1/4$, then there exists a fractional packing $\{r_i, T_i\}_{i=1}^m$ such that the number of mistakes made by this clustering is at most $4\sum_i r_i$.*

Proof: Let the clustering on V be $(\mathcal{C}_1, \cdots, \mathcal{C}_k)$. First consider the case where the number of negative mistakes $(m_{\mathcal{C}}^-)$ is at least half the total number of mistakes $m_{\mathcal{C}}$. We will construct a fractional packing of erroneous triangles with $\sum_i r_i \geq \frac{1}{2} m_{\mathcal{C}}^- \geq \frac{1}{4} m_{\mathcal{C}}$.

Pick a negative edge $(u,v) \in \mathcal{C}_i \times \mathcal{C}_i$ that has not been considered so far. We will pick a vertex $w \in \mathcal{C}_i$ such that both (u,w) and (v,w) are positive, and associate (u,v) with the erroneous triangle (u,v,w) (see Figure 1). We now show that for all (u,v), such a w can always be picked such that no other negative edges (u',v) or (u,v') (i.e. the ones sharing u or v) also pick w.

Since $\mathcal{C}_i$ is δ-clean, neither u nor v has more than $\delta|\mathcal{C}_i|$ negative neighbors inside $\mathcal{C}_i$. Thus (u,v) has at least $(1-2\delta)|\mathcal{C}_i|$ vertices w such that both (u,w) and (v,w) are positive. Moreover, at most $2\delta|\mathcal{C}_i| - 2$ of these could have already been chosen by other negative edges (u,v') or (u',v). Thus (u,v) has at least $(1-4\delta)|\mathcal{C}_i| + 2$ choices of w that satisfy the required condition. Since $\delta \leq 1/4$, (u,v) will always be able to pick such a w. Let T_{uvw} denote the erroneous triangle u,v,w.

Note that any positive edge (v,w) can be chosen at most 2 times by the above scheme, once for negative mistakes on v and possibly again for negative mistakes on w. Thus we can give a value of $r_{uvw} = 1/2$ to each erroneous triangle picked, ensuring that $\sum_{T_i \text{ contains } (u,v)} r_i \leq 1$. Now, since we pick a triangle for each negative mistake, we get that $\sum_{T_i} r_i = \frac{1}{2}\sum_{T_i} 1 \geq \frac{1}{2} m_{\mathcal{C}}^-$.

Next, consider the case when at least half the mistakes are positive mistakes. Just as above, we will associate mistakes with erroneous triangles. We will start afresh, without taking into account the labelings from the previous part.

Consider a positive edge between $u \in C_i$ and $v \in C_j$. Let $|C_i| \geq |C_j|$. Pick a $w \in C_i$ such that (u,w) is positive and (v,w) is negative (see Figure 1). There will be at least $|C_i| - \delta(|C_i| + |C_j|)$ such vertices as before and at most $\delta(|C_i| + |C_j|)$ of them will be already taken. Thus, there are at least $|C_i| - 2\delta(|C_i| + |C_j|) \geq |C_i|(1-4\delta) > 0$ choices for w. Moreover only the positive edge (u,w) can be chosen twice (once as (u,w) and once as (w,u)). Thus, as before, to obtain a packing, we can give a fractional value of $r_{uvw} = \frac{1}{2}$ to the triangle T_{uvw}. We get that $\sum_{T_i} r_i = \frac{1}{2}\sum_{T_i} 1 \geq \frac{1}{2} m_C^+$.

Now depending on whether there are more negative mistakes or more positive mistakes, we can choose the triangles appropriately, and hence account for at least a quarter of the total mistakes in the clustering. □

Lemma 4 along with Lemma 3 gives us the following corollary.

COROLLARY 5. *Any clustering in which all clusters are* δ*-clean for some* $\delta \leq \frac{1}{4}$ *has at most* $4m_{\mathrm{OPT}}$ *mistakes.*

LEMMA 6. *There exists a clustering* OPT$'$ *in which each non-singleton cluster is* δ*-clean, and* $m_{\mathrm{OPT}'} \leq (\frac{9}{\delta^2} + 1) m_{\mathrm{OPT}}$.

Proof: Consider the following procedure applied to the clustering of OPT and call the resulting clustering OPT$'$.

Procedure δ*-Clean-Up:* Let $C_1^{\mathrm{OPT}}, C_2^{\mathrm{OPT}}, \ldots, C_k^{\mathrm{OPT}}$ be the clusters in OPT.

1. Let $S = \emptyset$.
2. For $i = 1, \cdots, k$ do:
   a) If the number of $\frac{\delta}{3}$-bad vertices in C_i^{OPT} is more than $\frac{\delta}{3}|C_i^{\mathrm{OPT}}|$, then, $S = S \cup C_i^{\mathrm{OPT}}$, $C_i' = \emptyset$. We call this "dissolving" the cluster.
   b) Else, let B_i denote the $\frac{\delta}{3}$-bad vertices in C_i^{OPT}. Then $S = S \cup B_i$ and $C_i' = C_i^{\mathrm{OPT}} \setminus B_i$.
3. Output the clustering OPT$'$: $C_1', C_2', \ldots, C_k', \{x\}_{x \in S}$.

We will prove that m_{OPT} and $m_{\mathrm{OPT}'}$ are closely related.

We first show that each C_i' is δ clean. Clearly, this holds if $C_i' = \emptyset$. Now if C_i' is non-empty, we know that $|C_i^{\mathrm{OPT}}| \geq |C_i'| \geq |C_i^{\mathrm{OPT}}|(1-\delta/3)$. For each point $v \in C_i'$, we have:

$$\begin{aligned} |N^+(v) \cap C_i'| &\geq (1-\frac{\delta}{3})|C_i^{\mathrm{OPT}}| - (\frac{\delta}{3})|C_i^{\mathrm{OPT}}| \\ &= (1-2\frac{\delta}{3})|C_i^{\mathrm{OPT}}| \\ &> (1-\delta)|C_i'| \end{aligned}$$

Similarly, counting positive neighbors of v in $C_i^{\mathrm{OPT}} \cap \overline{C_i'}$ and outside C_i^{OPT}, we get,

$$\begin{aligned} |N^+(v) \cap \overline{C_i'}| &\leq \frac{\delta}{3}|C_i^{\mathrm{OPT}}| + \frac{\delta}{3}|C_i^{\mathrm{OPT}}| \\ &\leq \frac{2\delta}{3}\frac{|C_i'|}{(1-\delta/3)} \\ &< \delta|C_i'| \quad (\text{ as } \delta < 1) \end{aligned}$$

Thus each C_i' is δ-clean.

We now account for the number of mistakes. If we dissolve some C_i^{OPT}, then clearly the number of mistakes associated with vertices in the original cluster C_i^{OPT} is at least $(\delta/3)^2|C_i^{\mathrm{OPT}}|^2/2$. The mistakes added due to dissolving clusters is at most $|C_i^{\mathrm{OPT}}|^2/2$.

If C_i^{OPT} was not dissolved, then, the original mistakes in C_i^{OPT} were at least $\delta/3|C_i^{\mathrm{OPT}}||B_i|/2$. The mistakes added by the procedure is at most $|B_i||C_i^{\mathrm{OPT}}|$. Noting that $6/\delta < 9/\delta^2$, the lemma follows. □

For the clustering OPT$'$ given by the above lemma, we use C_i' to denote the non-singleton clusters and S to denote the set of singleton clusters. We will now describe Algorithm *Cautious* that tries to find clusters similar to OPT$'$. Throughout the rest of this section, we assume that $\delta = \frac{1}{44}$.

Algorithm Cautious:

1. Pick an arbitrary vertex v and do the following:

   a) Let $A(v) = N^+(v)$.

   b) (**Vertex Removal Step**): While $\exists x \in A(v)$ such that x is 3δ-bad w.r.t. $A(v)$, $A(v) = A(v) \setminus \{x\}$.

   c) (**Vertex Addition Step**): Let $Y = \{y | y \in V, y \text{ is } 7\delta\text{-good w.r.t. } A(v)\}$. Let $A(v) = A(v) \cup Y$.[3]

2. Delete $A(v)$ from the set of vertices and repeat until no vertices are left or until all the produced sets $A(v)$ are empty. In the latter case, output the remaining vertices as singleton nodes.

Call the clusters output by algorithm *Cautious* $A_1, A_2, \cdots$. Let Z be the set of singleton vertices created in the final step. Our main goal will be to show that the clusters output by our algorithm satisfy the property stated below.

THEOREM 7. *$\forall j, \exists i$ such that $C_j' \subseteq A_i$. Moreover, each A_i is 11δ-clean.*

[3] Observe that in the vertex addition step, all vertices are added in one step as opposed to in the vertex removal step

In order to prove this theorem, we need the following two lemmas.

LEMMA 8. *If $v \in \mathcal{C}_i'$, where $\mathcal{C}_i'$ is a δ-clean cluster in* OPT$'$*, then, any vertex $w \in \mathcal{C}_i'$ is 3δ-good w.r.t. $N^+(v)$.*

Proof: As $v \in \mathcal{C}_i$, $|N^+(v) \cap \mathcal{C}_i'| \geq (1-\delta)|\mathcal{C}_i'|$ and $|N^+(v) \cap \overline{\mathcal{C}_i'}| \leq \delta|\mathcal{C}_i'|$. So, $(1-\delta)|\mathcal{C}_i'| \leq |N^+(v)| \leq (1+\delta)|\mathcal{C}_i'|$. The same holds for w. Thus, we get the following two conditions.

$$|N^+(w) \cap N^+(v)| \geq (1-2\delta)|\mathcal{C}_i'| \geq (1-3\delta)|N^+(v)|$$

$$\begin{aligned} |N^+(w) \cap \overline{N^+(v)}| &\leq |N^+(w) \cap \overline{N^+(v)} \cap \mathcal{C}_i'| + |N^+(w) \cap \overline{N^+(v)} \cap \overline{\mathcal{C}_i'}| \\ &\leq 2\delta|\mathcal{C}_i'| \leq \tfrac{2\delta}{1-\delta}|N^+(v)| \leq 3\delta|N^+(v)| \end{aligned}$$

Thus, w is 3δ-good w.r.t. $N^+(v)$. □

LEMMA 9. *Given an arbitrary set X, if $v_1 \in \mathcal{C}_i'$ and $v_2 \in \mathcal{C}_j'$, $i \neq j$, then v_1 and v_2 cannot both be 3δ-good w.r.t. X.*

Proof:

Suppose that v_1 and v_2 are both 3δ-good with respect to X. Then, $|N^+(v_1) \cap X| \geq (1-3\delta)|X|$ and $|N^+(v_2) \cap X| \geq (1-3\delta)|X|$, hence $|N^+(v_1) \cap N^+(v_2) \cap X| \geq (1-6\delta)|X|$, which implies that

$$|N^+(v_1) \cap N^+(v_2)| \geq (1-6\delta)|X| \tag{1}$$

Also, since v_1 and v_2 lie in δ-clean clusters $\mathcal{C}_i'$ and $\mathcal{C}_j'$ in OPT$'$ respectively, $|N^+(v_1) \setminus \mathcal{C}_i'| \leq \delta|\mathcal{C}_i'|$, $|N^+(v_2) \setminus \mathcal{C}_j'| \leq \delta|\mathcal{C}_j'|$ and $\mathcal{C}_i' \cap \mathcal{C}_j' = \emptyset$. It follows that

$$|N^+(v_1) \cap N^+(v_2)| \leq \delta(|\mathcal{C}_i'| + |\mathcal{C}_j'|) \tag{2}$$

Now notice that $|\mathcal{C}_i'| \leq |N^+(v_1) \cap \mathcal{C}_i'| + \delta|\mathcal{C}_i'| \leq |N^+(v_1) \cap X \cap \mathcal{C}_i'| + |N^+(v_1) \cap \overline{X} \cap \mathcal{C}_i'| + \delta|\mathcal{C}_i'| \leq |N^+(v_1) \cap X \cap \mathcal{C}_i'| + 3\delta|X| + \delta|\mathcal{C}_i'| \leq (1+3\delta)|X| + \delta|\mathcal{C}_i'|$. So, $|\mathcal{C}_i'| \leq \frac{1+3\delta}{1-\delta}|X|$. The same holds for $\mathcal{C}_j'$. Using Equation 2, $|N^+(v_1) \cap N^+(v_2)| \leq 2\delta\frac{1+3\delta}{1-\delta}|X|$.

However, since $\delta < 1/9$, we have $2\delta(1+3\delta) < (1-6\delta)(1-\delta)$. Thus the above equation along with Equation 1 gives a contradiction and the result follows. □

This gives us the following important corollary.

COROLLARY 10. *After every application of the removal step 1b of the algorithm, no two vertices from distinct $\mathcal{C}_i'$ and $\mathcal{C}_j'$ can be present in $A(v)$.*

Now we go on to prove Theorem 7.

Proof of Theorem 7: We will first show that each A_i is either a subset of S or contains exactly one of the clusters $\mathcal{C}'_j$. The first part of the theorem will follow.

We proceed by induction on i. Consider the inductive step. For a cluster A_i, let A'_i be the set produced after the vertex removal phase such the cluster A_i is obtained by applying the vertex addition phase to A'_i. We have two cases. First, we consider the case when $A'_i \subseteq S$. Now during the vertex addition step, no vertex $u \in \mathcal{C}'_j$ can enter A'_i for any j. This follows because, since $\mathcal{C}'_j$ is δ-clean and disjoint from A'_i, for u to enter we need that $\delta|\mathcal{C}'_j| \geq (1-7\delta)|A'_i|$ and $(1-\delta)|\mathcal{C}'_j| \leq 7\delta|A'_i|$, and these two conditions cannot be satisfied simultaneously. Thus $A_i \subseteq S$.

In the second case, some $u \in \mathcal{C}'_j$ is present in A'_i. However, in this case observe that from Corollary 10, no vertices from $\mathcal{C}'_k$ can be present in A'_i for any $k \neq j$. Also, by the same reasoning as for the case $A'_i \subseteq S$, no vertex from $\mathcal{C}'_k$ will enter A'_i in the vertex addition phase. Now it only remains to show that $\mathcal{C}'_j \subseteq A_i$. Note that all vertices of $\mathcal{C}'_j$ are still present in the remaining graph $G \setminus (\bigcup_{\ell<i} A_\ell)$.

Since u was not removed from A'_i it follows that many vertices from $\mathcal{C}'_j$ are present in A'_i. In particular, $|N^+(u) \cap A'_i| \geq (1-3\delta)|A'_i|$ and $|N^+(u) \cap \overline{A'_i}| \leq 3\delta|A'_i|$. Now $(1-\delta)|\mathcal{C}'_j| \leq |N^+(u)|$ implies that $|\mathcal{C}'_j| \leq \frac{1+3\delta}{1-\delta}|A'_i| < 2|A'_i|$. Also, $|A'_i \cap \mathcal{C}'_j| \geq |A'_i \cap N^+(u)| - |N^+(u) \cap \overline{\mathcal{C}'_j}| \geq |A'_i \cap N^+(u)| - \delta|\mathcal{C}'_j|$. So we have $|A'_i \cap \mathcal{C}'_j| \geq (1-5\delta)|A'_i|$.

We now show that all remaining vertices from $\mathcal{C}'_j$ will enter A_i during the vertex addition phase. For $w \in \mathcal{C}'_j$ such that $w \notin A'_i$, $|A'_i \cap \overline{\mathcal{C}'_j}| \leq 5\delta|A'_i|$ and $|\overline{N^+(w)} \cap \mathcal{C}'_j| \leq \delta|\mathcal{C}'_j|$ together imply that $|A'_i \cap \overline{N^+(w)}| \leq 5\delta|A'_i| + \delta|\mathcal{C}'_j| \leq 7\delta|A'_i|$. The same holds for $|\overline{A'_i} \cap N^+(w)|$. So w is 7δ-good w.r.t. A'_i and will be added in the Vertex Addition step. Thus we have shown that $A(v)$ can contain $\mathcal{C}'_j$ for at most one j and in fact will contain this set entirely.

Next, we will show that for every j, $\exists i$ s.t. $\mathcal{C}'_j \subseteq A_i$. Let v chosen in Step 1 of the algorithm be such that $v \in \mathcal{C}'_j$. We show that during the vertex removal step, no vertex from $N^+(v) \cap \mathcal{C}'_j$ is removed. The proof follows by an easy induction on the number of vertices removed so far (r) in the vertex removal step. The base case $(r=0)$ follows from Lemma 8 since every vertex in $\mathcal{C}'_j$ is 3δ-good with respect to $N^+(v)$. For the induction step observe that since no vertex from $N^+(v) \cap \mathcal{C}'_j$ is removed thus far, every vertex in $\mathcal{C}'_j$ is still 3δ-good w.r.t. to the intermediate $A(v)$ (by mimicking the proof of Lemma 8 with $N^+(v)$ replaced by $A(v)$). Thus A'_i contains at least $(1-\delta)|\mathcal{C}'_j|$ vertices of $\mathcal{C}'_j$ at the end of the vertex removal phase, and hence by the second case above, $\mathcal{C}'_j \subseteq A_i$ after the vertex addition phase.

Finally we show that every non-singleton cluster A_i is 11δ-clean. We know that at the end of the vertex removal phase, $\forall x \in A'_i$, x is 3δ-good w.r.t. A'_i.

Thus, $|N^+(x)\cap\overline{A'_i}| \leq 3\delta|A'_i|$. So the total number of positive edges leaving A'_i is at most $3\delta|A'_i|^2$. Since, in the vertex addition step, we add vertices that are 7δ-good w.r.t. A'_i, the number of these vertices can be at most $3\delta|A'_i|^2/(1-7\delta)|A'_i| < 4\delta|A'_i|$. Thus $|A_i| < (1+4\delta)|A'_i|$.

Since all vertices v in A_i are at least 7δ-good w.r.t. A'_i, $N^+(v)\cap A_i \geq (1-7\delta)|A'_i| \geq \frac{1-7\delta}{1+4\delta}|A_i| \geq (1-11\delta)|A_i|$. Similarly, $N^+(v)\cap\overline{A_i} \leq 7\delta|A'_i| \leq 11\delta|A_i|$. This gives us the result. □

Now we are ready to bound the mistakes of A in terms of OPT and OPT$'$. Call mistakes that have both end points in some clusters A_i and A_j as internal mistakes and those that have an end point in Z as external mistakes. Similarly in OPT$'$, we call mistakes among the sets $\mathcal{C}'_i$ as internal mistakes and mistakes having one end point in S as external mistakes. We bound mistakes of *Cautious* in two steps: the following lemma bounds external mistakes.

LEMMA 11. *The total number of external mistakes made by* Cautious *are less than the external mistakes made by* OPT$'$.

Proof: From Theorem 7, it follows that Z cannot contain any vertex v in some $\mathcal{C}'_i$. Thus, $Z \subseteq S$. Now, any external mistakes made by *Cautious* are positive edges adjacent to vertices in Z. These edges are also mistakes in OPT$'$ since they are incident on singleton vertices in S. Hence the lemma follows. □

Now consider the internal mistakes of A. Notice that these could be many more than the internal mistakes of OPT$'$. However, we can at this point apply Lemma 5 on the graph induced by $V' = \bigcup_i A_i$. In particular, the bound on internal mistakes follows easily by observing that $11\delta \leq 1/4$, and that the mistakes of the optimal clustering on the graph induced by V' is no more than m_{OPT}. Thus,

LEMMA 12. *The total number of internal mistakes of* Cautious *is* $\leq 4m_{\mathrm{OPT}}$.

Summing up results from the Lemmas 11 and 12, and using Lemma 6, we get the following theorem:

THEOREM 13. $m_{Cautious} \leq (\frac{9}{\delta^2}+5)m_{\mathrm{OPT}}$, *with* $\delta = \frac{1}{44}$.

5. A PTAS for maximizing agreements

In this section, we give a PTAS for maximizing agreements: the total number of positive edges inside clusters and negative edges between clusters.

As before, let OPT denote an optimal clustering and A denote our clustering. We will abuse notation and also use OPT to denote the number of agreements in the optimal solution. As noticed in the introduction, OPT $\geq$

$n(n-1)/4$. So it suffices to produce a clustering that has at least $\mathrm{OPT}-\varepsilon n^2$ agreements, which will be the goal of our algorithm. Let $\delta^+(V_1,V_2)$ denote the number of positive edges between sets $V_1,V_2 \subseteq V$. Similarly, let $\delta^-(V_1,V_2)$ denote the number of negative edges between the two. Let $\mathrm{OPT}(\varepsilon)$ denote the optimal clustering that has all non-singleton clusters of size greater than εn.

LEMMA 14. $\mathrm{OPT}(\varepsilon) \geq \mathrm{OPT}-\varepsilon n^2/2$.

Proof: Consider the clusters of OPT of size less than or equal to εn and break them apart into clusters of size 1. Breaking up a cluster of size s reduces our objective function by at most $\binom{s}{2}$, which can be viewed as $s/2$ per node in the cluster. Since there are at most n nodes in these clusters, and these clusters have size at most εn, the total loss is at most $\varepsilon \frac{n^2}{2}$. □

The above lemma means that it suffices to produce a good approximation to $\mathrm{OPT}(\varepsilon)$. Note that the number of non-singleton clusters in $\mathrm{OPT}(\varepsilon)$ is less than $\frac{1}{\varepsilon}$. Let $\mathcal{C}_1^{\mathrm{OPT}},\ldots,\mathcal{C}_k^{\mathrm{OPT}}$ denote the non-singleton clusters of $\mathrm{OPT}(\varepsilon)$ and let $\mathcal{C}_{k+1}^{\mathrm{OPT}}$ denote the set of points which correspond to singleton clusters.

5.1. A PTAS DOUBLY-EXPONENTIAL IN $1/\varepsilon$

If we are willing to have a run time that is doubly-exponential in $1/\varepsilon$, we can do this by reducing our problem to the General Partitioning problem of (Goldreich et al., 1998). The idea is as follows.

Let G^+ denote the graph of only the $+$ edges in G. Then, notice that we can express the quality of $\mathrm{OPT}(\varepsilon)$ in terms of just the sizes of the clusters, and the number of edges in G^+ between and inside each of $\mathcal{C}_1^{\mathrm{OPT}},\ldots,\mathcal{C}_{k+1}^{\mathrm{OPT}}$. In particular, if $s_i = |\mathcal{C}_i^{\mathrm{OPT}}|$ and $e_{i,j} = \delta^+(\mathcal{C}_i^{\mathrm{OPT}},\mathcal{C}_j^{\mathrm{OPT}})$, then the number of agreements in $\mathrm{OPT}(\varepsilon)$ is:

$$\left[\sum_{i=1}^{k} e_{i,i}\right] + \left[\binom{s_{k+1}}{2} - e_{k+1,k+1}\right] + \left[\sum_{i\neq j}(s_i s_j - e_{i,j})\right].$$

The General Partitioning property tester of (Goldreich et al., 1998) allows us to specify values for the s_i and e_{ij}, and if a partition of G^+ exists satisfying these constraints, will produce a partition that satisfies these constraints approximately. We obtain a partition that has at least $\mathrm{OPT}(\varepsilon)-\varepsilon n^2$ agreements. The property tester runs in time exponential in $(\frac{1}{\varepsilon})^{k+1}$ and polynomial in n.

Thus if we can guess the values of these sizes and number of edges accurately, we would be done. It suffices, in fact, to only guess the values up to an additive $\pm\varepsilon^2 n$ for the s_i, and up to an additive $\pm\varepsilon^3 n^2$ for the $e_{i,j}$, because this introduces an additional error of at most $O(\varepsilon)$. So, at most $O((1/\varepsilon^3)^{1/\varepsilon^2})$ calls to the property tester need to be made. Our algorithm proceeds by finding a partition for each possible value of s_i and $e_{i,j}$ and returns the partition with the maximum number of agreements. We get the following result:

THEOREM 15. *The General Partitioning algorithm returns a clustering of graph G which has more than* $\mathrm{OPT} - \varepsilon n^2$ *agreements with probability at least* $1 - \delta$. *It runs in time* $e^{O((\frac{1}{\varepsilon})^{1/\varepsilon})} \times poly(n, \frac{1}{\delta})$.

5.2. A SINGLY-EXPONENTIAL PTAS

We will now describe an algorithm that is based on the same basic idea of random sampling used by the General Partitioning algorithm. The idea behind our algorithm is as follows: Notice that if we knew the density of positive edges between a vertex and all the clusters, we could put v in the cluster that has the most positive edges to it. However, trying all possible values of the densities requires too much time. Instead we adopt the following approach: Let $\{O_i\}$ be the clusters in OPT. We select a small random subset W of vertices and cluster them correctly into $\{W_i\}$ with $W_i \subset O_i \; \forall i$, by enumerating all possible clusterings of W . Since this subset is picked randomly, with a high probability, for all vertices v, the density of positive edges between v and W_i will be approximately equal to the density of positive edges between v and O_i. So we can decide which cluster to put v into, based on this information. However this is not sufficient to account for edges between two vertices v_1 and v_2, both of which do not belong to W. So, we consider a partition of the rest of the graph into subsets U_i of size m and try out all possible clusterings $\{U_{ij}\}$ of each subset, picking the one that maximizes agreements with respect to $\{W_i\}$. This gives us the PTAS.

Firstly note that if $|\mathcal{C}_{k+1}^{\mathrm{OPT}}| < \varepsilon n$, then if we only consider the agreements in the graph $G \setminus \mathcal{C}_{k+1}^{\mathrm{OPT}}$, it affects the solution by at most εn^2. For now, we will assume that $|\mathcal{C}_{k+1}^{\mathrm{OPT}}| < \varepsilon n$ and will present the algorithm and analysis based on this assumption. Later we will discuss the changes required to deal with the other case.

In the following algorithm ε is a performance parameter to be specified later. Let $m = \frac{88^3 \times 40}{\varepsilon^{10}}(\log \frac{1}{\varepsilon} + 2)$, $k = \frac{1}{\varepsilon}$ and $\varepsilon' = \frac{\varepsilon^3}{88}$. Let p_i denote the density of positive edges inside the cluster $\mathcal{C}_i^{\mathrm{OPT}}$ and n_{ij} the density of negative edges between clusters $\mathcal{C}_i^{\mathrm{OPT}}$ and $\mathcal{C}_j^{\mathrm{OPT}}$. That is, $p_i = \delta^+(\mathcal{C}_i^{\mathrm{OPT}}, \mathcal{C}_i^{\mathrm{OPT}}) / \binom{|\mathcal{C}_i^{\mathrm{OPT}}|}{2}$ and $n_{ij} = \delta^-(\mathcal{C}_i^{\mathrm{OPT}}, \mathcal{C}_j^{\mathrm{OPT}}) / (|\mathcal{C}_i^{\mathrm{OPT}}||\mathcal{C}_j^{\mathrm{OPT}}|)$. Let $W \subset V$ be a random subset of size m.

We begin by defining a measure of goodness of a clustering $\{U_{ij}\}$ of some set U_i with respect to a fixed partition $\{W_i\}$, that will enable us to pick the right clustering of the set U_i. Let $\hat{p}_i$ and $\hat{n_{ij}}$ be estimates of p_i and n_{ij} respectively, based on $\{W_i\}$, to be defined later in the algorithm.

DEFINITION 2. $U_{i1}, \ldots, U_{i(k+1)}$ *is* ε'-**good** *w.r.t.* $W_1, \ldots, W_{k+1}$ *if it satisfies the following for all* $1 \leq j, \ell \leq k$

(1) $\delta^+(U_{ij}, W_j) \geq \hat{p}_j \binom{W_j}{2} - 18\varepsilon' m^2$

(2) $\delta^-(U_{ij}, W_\ell) \geq \hat{n_{j\ell}} |W_j||W_\ell| - 6\varepsilon' m^2$

and, for at least $(1-\varepsilon')n$ *of the vertices* x *and* $\forall\ j$*,*

(3) $\delta^+(U_{ij}, x) \in \delta^+(W_j, x) \pm 2\varepsilon' m$*.*

Our algorithm is as follows:

Algorithm Divide&Choose*:*

```
1. Pick a random subset $W \subset V$ of size $m$.

2. For all partitions $W_1, \ldots, W_{k+1}$ of $W$ do

   a) Let $\hat{p}_i = \delta^+(W_i, W_i)/\binom{|W_i|}{2}$, and $\hat{n_{ij}} = \delta^-(W_i, W_j)/|W_i||W_j|$.
   b) Let $q = \frac{n}{m} - 1$. Consider a random partition of $V \setminus W$ into $U_1, \ldots, U_q$,
      such that $\forall i$, $|U_i| = m$.
   c) For all $i$ do:
      Consider all $(k+1)$-partitions of $U_i$ and let $U_{i1}, \ldots, U_{i(k+1)}$ be a partition
      that is $\varepsilon'$-good w.r.t. $W_1, \ldots, W_{k+1}$ (by Definition 2 above). If
      there is no such partition, choose $U_{i1}, \ldots, U_{i(k+1)}$ arbitrarily.
   d) Let $A_j = \bigcup_i U_{ij}$ for all $i$. Let $a(\{W_i\})$ be the number of agreements of
      this clustering.

3. Let $\{W_i\}$ be the partition of $W$ that maximizes $a(\{W_i\})$. Return the clusters
   $\{A_i\}, \{x\}_{x \in A_{k+1}}$ corresponding to this partition of $W$.
```

We will concentrate on the "right" partition of W given by $W_i = W \cap C_i^{\mathrm{OPT}}$, $\forall i$. We will show that the number of agreements of the clustering $A_1, \ldots, A_{k+1}$ corresponding to this partition $\{W_i\}$ is at least $\mathrm{OPT}(\varepsilon) - 2\varepsilon n^2$ with a high probability. Since we pick the best clustering, this gives us a PTAS.

We will begin by showing that with a high probability, for most values of i, the partition of U_is corresponding to the optimal partition is good with respect to $\{W_i\}$. Thus the algorithm will find at least one such partition. Next we will show that if the algorithm finds good partitions for most U_i, then it achieves at least $\mathrm{OPT} - O(\varepsilon)n^2$ agreements.

We will need the following results from probability theory. Please refer to (Alon and Spencer, 1992) for a proof.

FACT 1: Let $H(n,m,l)$ be the hypergeometric distribution with parameters n, m and l (choosing l samples from n points without replacement with the random variable taking a value of 1 on exactly m out of the n points). Let $0 \leq \varepsilon \leq 1$. Then

$$Pr\left[|H(n,m,l) - \frac{lm}{n}| \geq \frac{\varepsilon lm}{n}\right] \leq 2e^{-\frac{\varepsilon^2 lm}{2n}}$$

FACT 2: Let $X_1, X_2, ..., X_n$ be mutually independent random variables such that $|X_i - E[X_i]| < m$ for all i. Let $S = \sum_{i=1}^{n} X_i$, then

$$Pr[|S - E[S]| \geq a] \leq 2e^{-\frac{a^2}{2nm^2}}$$

We will also need the following lemma:

LEMMA 16. *Let Y and S be arbitrary disjoint sets and Z be a set picked from S at random. Then we have the following:*

$$Pr\left[|\delta^+(Y,Z) - \frac{|Z|}{|S|}\delta^+(Y,S)| > \varepsilon'|Y||Z|\right] \leq 2e^{\frac{-\varepsilon'^2|Z|}{2}}$$

Proof: $\delta^+(Y,Z)$ is a sum of $|Z|$ random variables $\delta^+(Y,v)$ $(v \in Z)$, each bounded above by $|Y|$ and having expected value $\frac{\delta^+(Y,S)}{|S|}$.

Thus applying Fact 2, we get

$$Pr[|\delta^+(Y,Z) - |Z|\delta^+(Y,S)/|S|| > \varepsilon'|Z||Y|] \leq 2e^{-\varepsilon'^2|Z|^2|Y|^2/2|Z||Y|^2} \leq 2e^{-\varepsilon'^2|Z|/2}$$

□

Now notice that since we picked W uniformly at random from V, with a high probability the sizes of W_is are in proportion to $|\mathcal{C}_i^{\mathrm{OPT}}|$. The following lemma formalizes this.

LEMMA 17. *With probability at least $1 - 2ke^{-\varepsilon'^2\varepsilon m/2}$ over the choice of W, $\forall i$, $|W_i| \in (1 \pm \varepsilon')\frac{m}{n}|\mathcal{C}_i^{\mathrm{OPT}}|$.*

Proof: For a given i, using Fact 1 and since $|\mathcal{C}_i^{\mathrm{OPT}}| \geq \varepsilon n$,

$$Pr\left[||W_i| - \frac{m}{n}|\mathcal{C}_i^{\mathrm{OPT}}|| > \varepsilon'\frac{m}{n}|\mathcal{C}_i^{\mathrm{OPT}}|\right] \leq 2e^{-\varepsilon'^2 m|\mathcal{C}_i^{\mathrm{OPT}}|/2n} \leq 2e^{-\varepsilon'^2\varepsilon m/2}$$

Taking a union bound over the k values of i we get the result. □

Using Lemma 17, we show that the computed values of $\hat{p}_i$ and $\hat{n_{ij}}$ are close to the true values p_i and n_{ij} respectively. This gives us the following two lemmas.

LEMMA 18. *If $W_i \subset \mathcal{C}_i^{\mathrm{OPT}}$ and $W_j \subset \mathcal{C}_j^{\mathrm{OPT}}$, $i \neq j$, then with probability at least $1 - 4e^{-\varepsilon'^2\varepsilon m/4}$ over the choice of W, $\delta^+(W_i, W_j) \in \frac{m^2}{n^2}\delta^+(\mathcal{C}_i^{\mathrm{OPT}}, \mathcal{C}_j^{\mathrm{OPT}}) \pm 3\varepsilon' m^2$.*

Proof: We will apply Lemma 16 in two steps. First we will bound $\delta^+(W_i,W_j)$ in terms of $\delta^+(W_i,C_j^{\mathrm{OPT}})$ by fixing W_i and considering the process of picking W_j from C_j^{OPT}.

Using W_i for Y, W_j for Z and C_j^{OPT} for S in Lemma 16, we get the following[4].

$$Pr\left[|\delta^+(W_i,W_j) - \frac{m}{n}\delta^+(W_i,C_j^{\mathrm{OPT}})| > \varepsilon' m^2\right] \le 2e^{-\varepsilon'^2\varepsilon m/4}$$

We used the fact that $m \ge |W_j| \ge \varepsilon m/2$ with high probability. Finally, we again apply Lemma 16 to bound $\delta^+(W_i,C_j^{\mathrm{OPT}})$ in terms of $\delta^+(C_i^{\mathrm{OPT}},C_j^{\mathrm{OPT}})$. Taking Y to be C_j^{OPT}, Z to be W_i and S to be C_i^{OPT}, we get

$$Pr\left[|\delta^+(W_i,C_j^{\mathrm{OPT}}) - \frac{m}{n}\delta^+(C_i^{\mathrm{OPT}},C_j^{\mathrm{OPT}})| > 2\varepsilon'\frac{m}{n}|C_i^{\mathrm{OPT}}||C_j^{\mathrm{OPT}}|\right] \le 2e^{-\varepsilon'^2\varepsilon m/4}$$

Again we used the fact that $|W_i| < \frac{2m}{n}|C_i^{\mathrm{OPT}}|$ with high probability. So, with probability at least $1-4e^{-\varepsilon'^2\varepsilon m/4}$, we have, $|\frac{m}{n}\delta^+(W_i,C_j^{\mathrm{OPT}}) - \frac{m^2}{n^2}\delta^+(C_i^{\mathrm{OPT}},C_j^{\mathrm{OPT}})| < 2\varepsilon'\frac{m^2}{n^2}|C_i^{\mathrm{OPT}}||C_j^{\mathrm{OPT}}| < 2\varepsilon' m^2$ and $|\delta^+(W_i,W_j) - \frac{m}{n}\delta^+(W_i,C_j^{\mathrm{OPT}})| < \varepsilon' m^2$. This gives us

$$Pr\left[|\delta^+(W_i,W_j) - \frac{m^2}{n^2}\delta^+(C_i^{\mathrm{OPT}},C_j^{\mathrm{OPT}})| > 3\varepsilon' m^2\right] \le 4e^{-\varepsilon'^2\varepsilon m/4}$$

□

LEMMA 19. *With probability at least* $1-\frac{8}{\varepsilon'^2}e^{-\varepsilon'^3\varepsilon m/4}$ *over the choice of* W, $\hat{p}_i \ge p_i - 9\varepsilon'$

Proof: Note that we cannot use an argument similar to the previous lemma directly here since we are dealing with edges inside the same set. Instead we use the following trick.

Consider an arbitrary partition of C_i^{OPT} into $\frac{1}{\varepsilon'}$ sets of size $\varepsilon' n'$ each where $n' = |C_i^{\mathrm{OPT}}|$. Let this partition be $C_{i,1}^{\mathrm{OPT}},\cdots,C_{i,1/\varepsilon'}^{\mathrm{OPT}}$ and let $W_{i,j} = W_i \cap C_{i,j}^{\mathrm{OPT}}$. Let $m' = |W_i|$. Now consider $\delta^+(W_{i,j_1},W_{i,j_2})$. Using an argument similar to the previous lemma, we get that with probability at least $1-4e^{-\varepsilon'^3\varepsilon m/4}$,

$$\delta^+(W_{i,j_1},W_{i,j_2}) \in \frac{|W_{i,j_1}||W_{i,j_2}|}{|C_{i,j_1}^{\mathrm{OPT}}||C_{i,j_2}^{\mathrm{OPT}}|}\delta^+(C_{i,j_1}^{\mathrm{OPT}},C_{i,j_2}^{\mathrm{OPT}}) \pm 2\varepsilon'|W_{i,j_1}||W_{i,j_2}|$$

Noting that $\frac{|W_{i,j_1}||W_{i,j_2}|}{|C_{i,j_1}^{\mathrm{OPT}}||C_{i,j_2}^{\mathrm{OPT}}|} < (1+3\varepsilon')\frac{m'^2}{n'^2}$, with probability at least $1-4e^{-\varepsilon'^3\varepsilon m/4}$, we get,

$$Pr\left[|\delta^+(W_{i,j_1},W_{i,j_2}) - \frac{m'^2}{n'^2}\delta^+(C_{i,j_1}^{\mathrm{OPT}},C_{i,j_2}^{\mathrm{OPT}})| < 8\varepsilon'|W_{i,j_1}||W_{i,j_2}|\right] \ge 1-8e^{-\varepsilon'^3\varepsilon m/4}$$

[4] We are assuming that W is a set of size m chosen randomly from n with replacement, since m is a constant, we will have no ties with probability $1-O(n^{-1})$.

This holds for every value of j_1 and j_2 with probability at least $1-\frac{8}{\varepsilon'^2}e^{-\varepsilon'^3\varepsilon m/4}$

Now, $\delta^+(W_i,W_i) \geq \sum_{j_1<j_2}\delta^+(W_{i,j_1},W_{i,j_2}) \geq \frac{1}{1+8\varepsilon'}\frac{m'^2}{n'^2}\sum_{j_1<j_2}\delta^+(\mathcal{C}^{\mathrm{OPT}}_{i,j_1},\mathcal{C}^{\mathrm{OPT}}_{i,j_2}) \geq \frac{1}{1+8\varepsilon'}\frac{m'^2}{n'^2}(p_i\frac{n'^2}{2}-\frac{1}{\varepsilon'}\frac{\varepsilon'^2n'^2}{2}) \geq (p_i-9\varepsilon')\frac{|W_i|^2}{2}$ □

Now let $U_{ij}=U_i\cap\mathcal{C}^{\mathrm{OPT}}_j$. The following lemma shows that for all i, with a high probability all U_{ij}s are ε'-good w.r.t. $\{W_i\}$. So we will be able to find ε'-good partitions for most U_is.

LEMMA 20. *For a given i, let $U_{ij}=U_i\cap\mathcal{C}^{\mathrm{OPT}}_j$, then with probability at least $1-32k\frac{1}{\varepsilon'^2}e^{-\varepsilon'^3\varepsilon m/4}$ over the choice of U_i, $\forall j\leq k$, $\{U_{ij}\}$ are ε'-good w.r.t. $\{W_j\}$.*

Proof: Consider the partition $\{U_{ij}\}$ of U_i.

Using an argument similar to Lemma 18, we get $|\delta^+(U_{ij},W_l)-\frac{m^2}{n^2}\delta^+(\mathcal{C}^{\mathrm{OPT}}_j,\mathcal{C}^{\mathrm{OPT}}_l)|\leq 3\varepsilon' m^2$ with probability at least $1-4e^{-\varepsilon'^2\varepsilon m/4}$. Also, again from Lemma 18, $|\delta^+(W_j,W_l)-\frac{m^2}{n^2}\delta^+(\mathcal{C}^{\mathrm{OPT}}_j,\mathcal{C}^{\mathrm{OPT}}_l)|\leq 3\varepsilon' m^2$. So, $|\delta^+(U_{ij},W_l)-\delta^+(W_j,W_l)|\leq 6\varepsilon' m^2$ with probability at least $1-8e^{-\varepsilon'^2\varepsilon m/4}$. This gives us the second condition of Definition 2.

Similarly, using Lemma 19, we obtain the first condition. The failure probability in this step is at most $16\frac{1}{\varepsilon'^2}e^{-\varepsilon'^3\varepsilon m/4}$.

Now, consider $\delta^+(x,U_{ij})$. This is a sum of m $\{0,1\}$ random variables (corresponding to picking U_i from V), each of which is 1 iff the picked vertex lies in $\mathcal{C}^{\mathrm{OPT}}_j$ and is adjacent to x. Applying Chernoff bound, we get,

$Pr\left[|\delta^+(x,U_{ij})-\frac{m}{n}\delta^+(x,\mathcal{C}^{\mathrm{OPT}}_j)|>\varepsilon' m\right]\leq 2e^{-\varepsilon'^2m/2}$

Similarly we have, $Pr\left[|\delta^+(x,W_j)-\frac{m}{n}\delta^+(x,\mathcal{C}^{\mathrm{OPT}}_j)|>\varepsilon' m\right]\leq 2e^{-\varepsilon'^2m/2}$.

So we get, $Pr\left[|\delta^+(x,U_{ij})-\delta^+(x,W_j)|>2\varepsilon' m\right]\leq 4e^{-\varepsilon'^2m/2}$.

Note that, here we are assuming that W and U_i are picked independently from V. However, picking U_i from $V\setminus W$ is similar to picking it from V since the collision probability is extremely small.

Now, the expected number of points that do not satisfy condition 3 for some U_{ij} is $4ne^{-\varepsilon'^2m/2}$. The probability that more than $\varepsilon' n$ of the points fail to satisfy condition 3 for one of the U_{ij}s in U_i is at most $k\frac{1}{\varepsilon' n}4ne^{-\varepsilon'^2m/2}\leq\frac{4k}{\varepsilon'}e^{-\varepsilon'^2m/2}$. This gives us the third condition.

The total probability that some U_i does not satisfy the above conditions is at most

$$\begin{aligned}&8e^{-\varepsilon'^2\varepsilon m/4}+16\frac{1}{\varepsilon'^2}e^{-\varepsilon'^3\varepsilon m/4}+\frac{4k}{\varepsilon'}e^{-\varepsilon'^2m/2}\\&\leq 32\frac{1}{\varepsilon'^2}e^{-\varepsilon'^3\varepsilon m/4}\end{aligned}$$

□

Now we can bound the total number of agreements of $A_1,\ldots,A_k,\{x\}_{x\in A_{k+1}}$ in terms of OPT:

THEOREM 21. *If $|C_{k+1}^{\mathrm{OPT}}| < \varepsilon n$, then $A \geq \mathrm{OPT} - 3\varepsilon n^2$ with probability at least $1-\varepsilon$.*

Proof: From Lemma 20, the probability that we were not able to find a ε'-good partition of U_i w.r.t. $W_1,\cdots,W_k$ is at most $32\frac{1}{\varepsilon'^2}e^{-\varepsilon'^3\varepsilon m/4}$. By our choice of m, this is at most $\varepsilon^2/4$. So, with probability at least $1-\varepsilon/2$, at most $\varepsilon/2$ of the U_is do not have an ε'-good partition.

In the following calculation of the number of agreements, we assume that we are able to find good partitions of all U_is. We will only need to subtract at most $\varepsilon n^2/2$ from this value to obtain the actual number of agreements, since each U_i can affect the number of agreements by at most mn.

We start by calculating the number of positive edges inside a cluster A_j. These are given by $\sum_a \sum_{x\in A_j}\delta^+(U_{aj},x)$. Using the fact that U_{aj} is good w.r.t. $\{W_i\}$ (condition (3)),

$$\begin{aligned}\textstyle\sum_{x\in A_j} \; & \delta^+(U_{aj},x)\\ & \geq \textstyle\sum_{x\in A_j}(\delta^+(W_j,x)-2\varepsilon' m)-\varepsilon' n|U_{aj}|\\ & = \textstyle\sum_b \delta^+(W_j,U_{bj})-2\varepsilon' m|A_j|-\varepsilon' n|U_{aj}|\\ & \geq \textstyle\sum_b\{\hat{p}_j\frac{|W_j|^2}{2}-18\varepsilon' m^2\}-2\varepsilon' m|A_j|-\varepsilon' n|U_{aj}|\end{aligned}$$

The last inequality follows from the fact that U_{bj} is good w.r.t. $\{W_i\}$ (condition (1)). From Lemma 17,

$$\begin{aligned}\textstyle\sum_{x\in A_j}\delta^+(U_{aj},x) & \geq \textstyle\sum_b\{\frac{m^2}{n^2}\hat{p}_j(1-\varepsilon')^2\frac{|C_j^{\mathrm{OPT}}|^2}{2}-18\varepsilon' m^2\}-2\varepsilon' m|A_j|-\varepsilon' n|U_{aj}|\\ & \geq \textstyle\frac{m}{n}\hat{p}_j(1-\varepsilon')^2\frac{|C_j^{\mathrm{OPT}}|^2}{2}-18\varepsilon' mn-2\varepsilon' m|A_j|-\varepsilon' n|U_{aj}|\end{aligned}$$

Thus we bound $\sum_a \delta^+(A_j,U_{aj})$ as $\sum_a \delta^+(A_j,U_{aj}) \geq \hat{p}_j(1-\varepsilon')^2\frac{|C_j^{\mathrm{OPT}}|^2}{2}-18\varepsilon' n^2-3\varepsilon' n|A_j|$.
Now using Lemma 19, the total number of agreements is at least

$$\begin{aligned}\textstyle\sum_j & \left[\hat{p}_j(1-\varepsilon')^2\frac{|C_j^{\mathrm{OPT}}|^2}{2}\right]-18\varepsilon' n^2 k-3\varepsilon' n^2\\ & \geq \textstyle\sum_j\left[(p_j-9\varepsilon')(1-\varepsilon')^2\frac{|C_j^{\mathrm{OPT}}|^2}{2}\right]-18\varepsilon' n^2 k-3\varepsilon' n^2\end{aligned}$$

Hence, $A^+ \geq \mathrm{OPT}^+ - 11\varepsilon' kn^2 - 21\varepsilon' n^2 k \geq \mathrm{OPT}^+ - 32\varepsilon' n^2 k$.
Similarly, consider the negative edges in A. Using Lemma 18 to estimate $\delta^-(U_{ai},U_{bj})$, we get,

$$\sum_{ab}\delta^-(U_{ai},U_{bj}) \geq \delta^-(C_i^{\mathrm{OPT}},C_j^{\mathrm{OPT}})-9\varepsilon' n^2-2\varepsilon' n|A_i|-\varepsilon' n|A_j|$$

Summing over all $i<j$, we get the total number of negative agreements is at least $\mathrm{OPT}^- - 12\varepsilon' k^2 n^2$.

So we have, $A \geq \mathrm{OPT} - 44\varepsilon' k^2 n^2 = \mathrm{OPT} - \varepsilon n^2/2$. However, since we lose $\varepsilon n^2/2$ for not finding ε'-good partitions of every U_i (as argued before), εn^2 due to $\mathcal{C}_{k+1}^{\mathrm{OPT}}$, and $\varepsilon n^2/2$ for using $k = \frac{1}{\varepsilon}$ we obtain $A \geq \mathrm{OPT} - 3\varepsilon n^2$.

The algorithm can fail in four situations:

1. More than $\varepsilon/2$ U_is do not have an ε'-good partition. However, this happens with probability at most $\varepsilon/2$.

2. Lemma 17 does not hold for some W_i. This happens with probability at most $2ke^{-\varepsilon'^2 \varepsilon m/2}$.

3. Lemma 19 does not hold for some i. This happens with probability at most $\frac{8k}{\varepsilon'^2} e^{-\varepsilon'^3 \varepsilon m/4}$

4. Lemma 18 does not hold for some pair i, j. This happens with probability at most $4k^2 e^{-\varepsilon'^2 \varepsilon m/4}$.

Observing that the latter three probabilities sum up to at most $\varepsilon/2$ by our choice of m. So, the algorithm succeeds with probability greater than $1 - \varepsilon$. □

Now we need to argue for the case when $|\mathcal{C}_{k+1}^{\mathrm{OPT}}| \geq \varepsilon n$. Notice that in this case, using an argument similar to Lemma 17, we can show that $|W_{k+1}| \geq \frac{\varepsilon m}{2}$ with a very high probability. This is good because, now with a high probability, $U_{i(k+1)}$ will also be ε'-good w.r.t. W_{k+1} for most values of i. We can now count the number of negative edges from these vertices and incorporate them in the proof of Theorem 21 just as we did for the other k clusters. So in this case, we can modify algorithm *Divide&Choose* to consider ε'-goodness of the $(k+1)$th partitions as well. This gives us the same guarantee as in Theorem 21. Thus our strategy will be to run Algorithm *Divide&Choose* once assuming that $|\mathcal{C}_{k+1}^{\mathrm{OPT}}| \geq \varepsilon n$ and then again assuming that $|\mathcal{C}_{k+1}^{\mathrm{OPT}}| \leq \varepsilon n$, and picking the better of the two outputs. One of the two cases will correspond to reality and will give us the desired approximation to OPT.

Now each U_i has $O(k^m)$ different partitions. Each iteration takes $O(nm)$ time. There are n/m U_is, so for each partition of W, the algorithm takes time $O(n^2 k^m)$. Since there are k^m different partitions of W, the total running time of the algorithm is $O(n^2 k^{2m}) = O(n^2 e^{O(\frac{1}{\varepsilon^{10}} \log(\frac{1}{\varepsilon}))})$. This gives us the following theorem:

THEOREM 22. *For any $\delta \in [0,1]$, using $\varepsilon = \frac{\delta}{3}$, Algorithm Divide&Choose runs in time $O(n^2 e^{O(\frac{1}{\delta^{10}} \log(\frac{1}{\delta}))})$ and with probability at least $1 - \frac{\delta}{3}$ produces a clustering with number of agreements at least $\mathrm{OPT} - \delta n^2$.*

6. Random noise

Going back to our original motivation, if we imagine there is some true correct clustering OPT of our n items, and that the only reason this clustering does not appear perfect is that our function $f(A,B)$ used to label the edges has some error, then it is natural to consider the case that the errors are random. That is, there is some constant noise rate $\nu < 1/2$ and each edge, independently, is mislabeled with respect to OPT with probability ν. In the machine learning context, this is called the problem of learning with random noise. As can be expected, this is much easier to handle than the worst-case problem. In fact, with very simple algorithms one can (w.h.p.) produce a clustering that is quite close to OPT, much closer than the number of disagreements between OPT and f. The analysis is fairly standard (much like the generic transformation of Kearns (1993) in the machine learning context, and even closer to the analysis of Condon and Karp for graph partitioning (Condon and Karp, 1999)). In fact, this problem nearly matches a special case of the planted-partition problem of McSherry (2001). We present our analysis anyway since the algorithms are so simple.

One-sided noise: As an easier special case, let us consider only one-sided noise in which each true "$+$" edge is flipped to "$-$" with probability ν. In that case, if u and v are in different clusters of OPT, then $|N^+(u) \cap N^+(v)| = 0$ for certain. But, if u and v are in the same cluster, then every other node in the cluster independently has probability $(1-\nu)^2$ of being a neighbor to both. So, if the cluster is large, then $N^+(u)$ and $N^+(v)$ will have a non-empty intersection with high probability. So, consider clustering greedily: pick an arbitrary node v, produce a cluster $C_v = \{u : |N^+(u) \cap N^+(v)| > 0\}$, and then repeat on $V - C_v$. With high probability we will correctly cluster *all* nodes whose clusters in OPT are of size $\omega(\log n)$. The remaining nodes might be placed in clusters that are too small, but overall the number of edge-mistakes is only $\tilde{O}(n)$.

Two-sided noise: For the two-sided case, it is technically easier to consider the symmetric difference of $N^+(u)$ and $N^+(v)$. If u and v are in the same cluster of OPT, then every node $w \notin \{u,v\}$ has probability exactly $2\nu(1-\nu)$ of belonging to this symmetric difference. But, if u and v are in different clusters, then all nodes w in $\mathrm{OPT}(u) \cup \mathrm{OPT}(v)$ have probability $(1-\nu)^2 + \nu^2 = 1 - 2\nu(1-\nu)$ of belonging to the symmetric difference. (For $w \notin \mathrm{OPT}(u) \cup \mathrm{OPT}(v)$, the probability remains $2\nu(1-\nu)$.) Since $2\nu(1-\nu)$ is a constant less than $1/2$, this means we can confidently detect that u and v belong to different clusters so long as $|\mathrm{OPT}(u) \cup \mathrm{OPT}(v)| = \omega(\sqrt{n \log n})$. Furthermore, using just $|N^+(v)|$, we can approximately sort the vertices by cluster sizes. Combining these two facts, we can w.h.p. correctly cluster all vertices in large clusters, and then just place each of the others into a cluster by itself, making a total of $\tilde{O}(n^{3/2})$ edge mistakes.

7. Extensions

So far in the paper, we have only considered the case of edge weights in $\{+,-\}$. Now we consider real valued edge weights. To address this setting, we need to define a cost model – the penalty for placing an edge inside or between clusters.

One natural model is a linear cost function. Specifically, let us assume that all edge weights lie in $[-1,+1]$. Then, given a clustering, we assign a cost of $\frac{1-x}{2}$ if an edge of weight x is within a cluster and a cost of $\frac{1+x}{2}$ if it is placed between two clusters. For example, an edge weighing 0.5 incurs a cost of 0.25 if it lies inside a cluster and 0.75 otherwise. A $0-$weight edge, on the other hand, incurs a cost of $1/2$ no matter what.

Another natural model is to consider weighted disagreements. That is, a positive edge incurs a penalty equal to its weight if it lies between clusters, and zero penalty otherwise, and vice versa for negative edges. The objective in this case is to minimize the sum of weights of positive edges between clusters and negative edges inside clusters. A special case of this problem is edge weights lying in $\{-1,0,+1\}$. Zero-weight edges incur no penalty, irrespective of the clustering, and thus can be thought of as missing edges.

In this section we show that our earlier results generalize to the case of linear cost functions for the problem of minimizing disagreements. However, we do not have similar results for the case of weighted disagreements or agreements. We give evidence that this latter case is hard to approximate.

Linear cost functions

First we consider the linear cost function on $[-1,+1]$ edges. It turns out, as we show in the following theorem, that any algorithm that finds a good clustering in a graph with $+1$ or -1 edges also works well in this case.

THEOREM 23. *Let A be an algorithm that produces a clustering on a graph with $+1$ and -1 edges with approximation ratio ρ. Then, we can construct an algorithm A' that achieves an approximation ratio of $(2\rho+1)$ on a $[-1,1]-$graph, under a linear cost function.*

Proof: Let G be a $[-1,1]-$graph, and let G' be the graph with $+1$ and -1 edges obtained when we assign a weight of 1 to all positive edges in G and -1 to all the negative edges (0 cost edges are weighted arbitrarily). Let OPT be the optimal clustering on G and OPT$'$ the optimal clustering on G'. Also, let m' be the measure of cost (on G') in the $\{+,-\}$ penalty model and m in the new $[-1,1]$ penalty model.

Then, $m'_{\mathrm{OPT}'} \leq m'_{\mathrm{OPT}} \leq 2m_{\mathrm{OPT}}$. The first inequality follows by design. The latter inequality holds because the edges on which OPT occurs a greater penalty in m' than in m are either the positive edges between clusters or negative edges inside a cluster. In both these situations, OPT incurs a cost of at least $1/2$ in m and at most 1 in m'.

Our algorithm A' simply runs A on the graph G' and outputs the resulting clustering A. So, we have, $m'_A \leq \rho m'_{\mathrm{OPT}'} \leq 2\rho m_{\mathrm{OPT}}$.

Now we need to bound m_A in terms of m'_A. Notice that, if a positive edge lies between two clusters in A, or a negative edge lies inside a cluster, then the cost incurred by A for these edges in m' is 1 while it is at most 1 in m. So, the total cost due to such mistakes is at most m'_A. On the other hand, if we consider cost due to positive edges inside clusters, and negative edges between clusters, then OPT also incurs at least this cost on those edges (because cost due to these edges can only increase if they are clustered differently). So cost due to these mistakes is at most m_{OPT}.

So we have,

$$\begin{aligned} m_A &\leq m'_A + m_{\mathrm{OPT}} \leq 2\rho m_{\mathrm{OPT}} + m_{\mathrm{OPT}} \\ &= (2\rho+1)m_{\mathrm{OPT}} \end{aligned}$$

□

Interestingly, the above theorem holds generally for a class of cost functions that we call *unbiased*. An unbiased cost function assigns a cost of at least $\frac{1}{2}$ to positive edges lying between clusters and negative edges inside clusters, and a cost of at most $\frac{1}{2}$ otherwise. A $0-$weight edge always incurs a cost of $\frac{1}{2}$ as before. For example, one such function is $\frac{1+x^3}{2}$ if an edge of weight x lies between clusters and $\frac{1-x^3}{2}$ otherwise.

Weighted Agreements/Disagreements

Next we consider minimizing weighted disagreements or maximizing weighted agreements. Consider first, the special case of edge weights lying in $\{-1,0,+1\}$. Notice that, as before, if a perfect clustering exists, then it is easy to find it, by simply removing all the $-$ edges and producing each connected component of the resulting graph as a cluster. The random case is also easy if defined appropriately. However, our approximation techniques do not appear to go through. We do not know how to achieve a constant-factor, or even logarithmic factor, approximation for minimizing disagreements. Note that we can still use our *Divide & Choose* algorithm to achieve an additive approximation of εn^2 for agreements. However, this does not imply a PTAS in this variant, because OPT might be $o(n^2)$.

Now, suppose we allow arbitrary real-valued edge weights, lying in $[-\infty,+\infty]$. For example, the edge weights might correspond to the log odds of two documents belonging to the same cluster. It is easy to see that the problem of minimizing disagreements for this variant is APX-hard, by reducing the problem of minimum multiway cut to it. Specifically, let G be a weighted graph with special nodes $v_1,\cdots,v_k$. The problem of minimum multiway cut is that of finding the smallest cut that separates these special nodes. This problem is known to be APX-hard (Garey and Johnson, 2000). We convert this problem into a disagreement minimization problem as follows: among each pair of special nodes v_i and v_j, we put an edge of weight $-\infty$. Then, notice that any clustering algorithm will definitely put each of $v_1,\cdots,v_k$ into separate clusters. The number (or total weight) of disagreements is equal to the value of the cut separating the special nodes. Thus, any algorithm that achieves an approximation ratio of ρ for minimizing disagreements, would achieve an approximation ratio of ρ for minimum multiway cut problem. We get the following:

THEOREM 24. *The problem of minimizing disagreements on weighted graphs with unbounded weights is APX-hard.*

Note that the above result is pretty weak. It does not preclude the possibility of achieving a constant approximation, similar to the one for $\{+,-\}$-weighted graphs. However we have reason to believe that unlike before, we cannot obtain a PTAS for maximizing agreements in this case. We show that a PTAS for maximizing agreements gives a polynomial time procedure for $O(n^{\varepsilon})$ coloring a $3-colorable$ graph. While it is unknown whether this problem is NP-Hard, the problem is well-studied and the best known result is due to Blum and Karger (1997), who give a polynomial time algorithm to $\tilde{O}(n^{3/14})$ color a 3-colorable graph.

THEOREM 25. *Given a PTAS for the problem of maximizing agreements, we can use the algorithm to obtain an algorithm for $O(n^{\varepsilon})$ coloring a $3-colorable$ graph, for any $\varepsilon > 0$.*

Proof: Let $G=(V,E)$ be a 3 colorable graph, and let $m=|E|$ and $n=|V|$. Let K be an n vertex complete graph obtained from G as follows: an edge e of K has weight -1 if e is an edge in G, and has a positive weight of $\delta m/\binom{n}{2}$ otherwise. Here δ is a parameter to be specified later.

If we choose each color class as a cluster, it is easy to see that the resulting clustering agrees on the m negative weight edges and on at least $3\binom{n/3}{2}$ positive weight edges. Thus the total weight of agreements in the optimal clustering is at least $m(1+\delta/3)$. Let us invoke the PTAS for maximizing agreements with $\varepsilon' = \delta/30$, then we obtain a clustering which has cost of agreements at least $m(1+\delta/3)/(1+\delta/30) \geq m(1+\delta/5)$.

We now claim that the size of largest cluster is at least $n/5$. Suppose not. Then the weight of positive agreements can be at most $\delta\frac{m}{\binom{n}{2}}\cdot 5\cdot\binom{n/5}{2}$ which is about $\delta m/5$. Since the total weight of negative edges is m, the total weight of agreements for the clustering cannot be more than $m(1+\delta/5)$, violating the guarantee given by the PTAS. Hence, there exists a cluster of size at least $n/5$ in this clustering. Call this cluster C.

Now observe that since the PTAS returns a clustering with at least $(1+\delta/5)m$ agreements, and the total weight of all positive edges is at most δm, the total weight of negative agreements is at least $(1-\frac{4\delta}{5})m$. This implies that C contains at most $\frac{4\delta}{5}m$ negative weight edges. Thus the density of negative weight edges in C is at most $\frac{4\delta m}{5}/\binom{n/5}{2}\approx 20\delta\cdot\frac{m}{\binom{n}{2}}$. That is, the cluster C has an edge density of at most about 20δ times that of G and size at least $n/5$.

We can now apply this procedure recursively to C (since C is also 3-colorable). After $2\log_b n$ such recursive steps, where $b=\frac{1}{20\delta}$, we obtain a set of density at most $1/n^2$ times that of C (and hence independent). Call this independent set I. Note that the size of I is at least $n/(5^{2\log_b n})$. Choosing δ such that $b=5^{2/\varepsilon}$, it is easy to verify that I has size at least $n^{1-\varepsilon}$.

Now we can remove I from G and iterate on $G-I$ (since $G-I$ is also 3-colorable). It is easy to see that this procedure gives an $O(n^{\varepsilon})$ coloring of G. □

8. Conclusions

In this paper, we have presented a constant-factor approximation for minimizing disagreements, and a PTAS for maximizing agreements, for the problem of clustering vertices in a fully-connected graph G with $\{+,-\}$ edge labels. In Section 7 we extended some of our results to the case of real-valued labels, under a linear cost metric. As mentioned before, an interesting open question is to construct good approximations for minimizing agreements and maximizing agreements for the case of edge weights lying in $\{-1,0,+1\}$, or to prove hardness of approximation for this case. Another interesting question is to determine whether the lower bound given by erroneous triangles is tight to within a small constant factor [5]. Such a fact might lead to a better approximation for minimizing disagreements.

[5] Interestingly, we were unable to come up with an example for which this factor is larger than 2. The latter is achieved in a star-like topology where all edges incident to a “root” vertex are positive and all other edges are negative.

8.1. Subsequent Work

Following the initial publication of this work, several better approximations and lower bounds have been developed for minimizing disagreements and maximizing agreements for general weighted graphs. Immorlica et. al. (2003) and Emanuel et. al. (2003) independently developed log-factor approximations for the problem of minimizing disagreements. The latter show that this problem is equivalent to the minimum multiway cut problem. The approximation for minimizing disagreements in the unweighted case was improved to a factor of 4 by Charikar, Guruswami and Wirth (2003). They also give a 0.7664-approximation for maximizing agreements in a general weighted graph, which was recently improved to 0.7666 by Swamy (2004). Charikar et. al. also improve our hardness of approximation result for minimizing disagreements to 29/28, and give a hardness of approximation of 115/116 for maximizing agreements.

Acknowledgements

We are grateful to William Cohen and Andrew McCallum for introducing us to the problem and for several useful discussions.

References

Alon, N., E. Fischer, M. Krivelevich, and M. Szegedy: 1999, 'Efficient testing of large graphs'. In: *Proceedings of the 40th Annual Symposium on Foundations of Computer Science*. pp. 645–655.

Alon, N. and J. H. Spencer: 1992, *The Probabilistic Method*. John Wiley and Sons.

Arora, S., A. Frieze, and H. Kaplan: 1996, 'A New Rounding Procedure for the Assignment Problem with Applications to Dense Graph Arrangements'. In: *Proceedings of the 37th Annual Symposium on Foundations of Computer Science*. pp. 21–30.

Arora, S., D. Karger, and M. Karpinski: 1995, 'Polynomial time approximation schemes for dense instances of NP-Hard problems'. In: *Proceedings of the 27nd Annual ACM Symposium on Theory of Computing*.

Ben-David, S., P. M. Long, and Y. Mansour: 2001, 'Agnostic boosting'. In: *Proceedings of the 2001 Conference on Computational Learning Theory*. pp. 507–516.

Blum, A. and D. Karger: 1997, 'An $\tilde{O}(n^{3/14})$-Coloring Algorithm for 3-Colorable Graphs'. *IPL: Information Processing Letters* **61**.

Charikar, M. and S. Guha: 1999, 'Improved combinatorial algorithms for the facility location and k-median problems'. In: *Proceedings of the 40th Annual Symposium on Foundations of Computer Science*.

Charikar, M., V. Guruswami, and A. Wirth: 2003, 'Clustering with Qualitative Information'. In: *Proceedings of the 44th Annual Symposium on Foundations of Computer Science*. pp. 524–533.

Cohen, W. and A. McCallum: 2001, 'Personal communication'.

Cohen, W. and J. Richman: 2001, 'Learning to match and cluster entity names'. In: *ACM SIGIR'01 workshop on Mathematical/Formal Methods in IR*.

Cohen, W. and J. Richman: 2002, 'Learning to match and cluster large high-dimensional data sets for data integration'. In: *Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*.

Condon, A. and R. Karp: 1999, 'Algorithms for graph partitioning on the planted partition model'. *Random Structures and Algorithms* **18**(2), 116–140.

de la Vega, F.: 1996, 'MAX-CUT has a Randomized Approximation Scheme in Dense Graphs'. *Random Structures and Algorithms* **8**(3), 187–198.

Demaine, E. and N. Immorlica: 2003, 'Correlation Clustering with Partial Information'. In: *Proceedings of APPROX*.

Emanuel, D. and A. Fiat: 2003, 'Correlation Clustering—Minimizing Disagreements on Arbitrary Weighted Graphs'. In: *Proceedings of ESA*.

Garey, M. and D. Johnson: 2000, *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman and Company.

Goldreich, O., S. Goldwasser, and D. Ron: 1998, 'Property testing and its connection to learning and approximation'. *JACM* **45**(4), 653–750.

Hochbaum, D. and D. Shmoys: 1986, 'A unified approach to approximation algorithms for bottleneck problems'. *JACM* **33**, 533–550.

Jain, K. and V. Vazirani: 1999, 'Primal-dual approximation algorithms for metric facility location and k-median problem'. In: *Proceedings of the 40th Annual Symposium on Foundations of Computer Science*.

Kearns, M.: 1993, 'Efficient Noise-Tolerant Learning from Statistical Queries'. In: *Proceedings of the 25th Annual ACM Symposium on Theory of Computing*. pp. 392–401.

Kearns, M. J., R. E. Schapire, and L. M. Sellie: 1994, 'Toward efficient agnostic learning'. *Machine Learning* **17**(2/3), 115–142.

McSherry, F.: 2001, 'Spectral Partitioning of Random Graphs'. In: *Proceedings of the 42th Annual Symposium on Foundations of Computer Science*. pp. 529–537.

Parnas, M. and D. Ron: 1999, 'Testing the diameter of graphs'. In: *Proceedings of RANDOM*. pp. 85–96.

Schulman, L.: 2000, 'Clustering for edge-cost minimization'. In: *Proceedings of the 32nd Annual ACM Symposium on Theory of Computing*. pp. 547–555.

Swamy, C.: 2004, 'Correlation Clustering: Maximizing Agreements via Semidefinite Programming'. In: *Proceedings of the Symposium on Discrete Algorithms*.