Correlation Clustering

Nikhil Bansal* Avrim Blum* Shuchi Chawla*

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. 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.

1 Introduction

Suppose that you are given a set of n documents to cluster into topics. Unfortunately, you have no idea of 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* (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 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 some clustering problems at Whizbang Labs in which learning algorithms have been trained to help with various clustering tasks [8, 9, 10].[1]

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 [7, 15] or min-sum clustering [20] or min-max clustering [14], 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,

*Department of Computer Science, Carnegie Mellon University. {nikhil,avrim,shuchi}@cs.cmu.edu. This research was supported in part by NSF grants CCR-0085982, CCR-0122581, CCR-0105488, and an IBM Graduate Fellowship.

[1] An example of one such problem 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.

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 [10].) 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 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, which can be seen via a tedious reduction from X3C (details can be found in [5]).

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 [12] for MAX-CUT on dense graphs, and related to PTASs of [4, 3]. 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 [13, 19, 1]. In fact, we show how we can use the General Partition Property Tester of [13] as a subroutine to get a PTAS with running time $O(ne^{O((\frac{1}{\epsilon})^{\frac{1}{\epsilon}})})$. Unfortunately, this is doubly exponential in $\frac{1}{\epsilon}$, so we also present an alternative direct algorithm (based more closely on the approach of [12]) that takes only $O(n^2 e^{O(\frac{1}{\epsilon})})$ 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 [17, 6]. 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 $\nu + \epsilon$. Our running time is exponential in $1/\epsilon$, 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 nontrivial problems where agnostic learning can be done in polynomial time. Even for simple classes such as conjunctions and disjunctions, no polynomial-time algorithms are known that give even an error gap of $1/2 - \epsilon$.

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 the 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 A Constant Factor Approximation for Minimizing Disagreements

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 1) 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 2) that there must exist a nearly optimal clustering OPT' in which all non-singleton clusters are "clean". Finally, we show (Theorem 3 and Lemma 7) 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 1.

We begin with a definition of a "clean" cluster and a "good" vertex.

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

- $|N^+(v) \cap \mathcal{C}| \geq (1-\delta)|\mathcal{C}|$
- $|N^+(v) \cap (V \setminus \mathcal{C})| \leq \delta|\mathcal{C}|$

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

We now present two key lemmas.

Lemma 1 *Given a clustering of V in which all clusters are δ-clean for some $\delta \leq 1/4$, then the number of mistakes made by this clustering is at most $8m_{\mathrm{OPT}}$.*

Proof: Let the clustering on V be $(\mathcal{C}_1, \cdots, \mathcal{C}_k)$. We will bound the number of mistakes made by this clustering by 8 times the number of edge-disjoint "erroneous triangles" in the graph, where an erroneous triangle is a triangle having two $+$ edges and one $-$ edge. We then use the fact that OPT must make at least one mistake for each such triangle.

First consider the negative mistakes. 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 $w \in \mathcal{C}_i$ such that both (u, w) and (v, w) are positive and associate (u, v) with the erroneous triangle (u, v, w). 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 $(1-4\delta)s + 2$ choices of w that satisfy the required condition. Since $\delta \leq 1/4$, (u, v) will always be able to pick such a w.

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 account for at least a fourth (because only positive edges are double counted) of the negative mistakes using edge disjoint erroneous triangles.

Now, we consider the 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 $u \in \mathcal{C}_i$ and $v \in \mathcal{C}_j$. Let $|\mathcal{C}_i| \geq |\mathcal{C}_j|$. Pick a $w \in \mathcal{C}_i$ such that (u, w) is positive and (v, w) is negative. There will be at least $|\mathcal{C}_i| - \delta(|\mathcal{C}_i| + |\mathcal{C}_j|)$ such vertices as before and at most $\delta(|\mathcal{C}_i| + |\mathcal{C}_j|)$ of them will be already taken. Moreover only the positive edge (u, w) can be chosen twice (once as (u, w) and once as (w, u)). Repeating the above argument, we again see that we account for at least half (hence at least a quarter) of the positive mistakes using edge disjoint triangles.

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 1/8 of the total mistakes in the clustering. ■

Lemma 2 *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 $\mathcal{C}_1^{\mathrm{OPT}}, \mathcal{C}_2^{\mathrm{OPT}}, ..., \mathcal{C}_k^{\mathrm{OPT}}$ 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 $\mathcal{C}_i^{\mathrm{OPT}}$ is more than $\frac{\delta}{3}|\mathcal{C}_i^{\mathrm{OPT}}|$, then, $S = S \cup \mathcal{C}_i^{\mathrm{OPT}}$, $\mathcal{C}'_i = \emptyset$. We call this "dissolving" the cluster.
   (b) Else, let B_i denote the $\frac{\delta}{3}$-bad vertices in $\mathcal{C}_i^{\mathrm{OPT}}$. Then $S = S \cup B_i$ and $\mathcal{C}'_i = \mathcal{C}_i^{\mathrm{OPT}} \setminus B_i$.
3. Output the clustering OPT': $\mathcal{C}'_1, \mathcal{C}'_2, ..., \mathcal{C}'_k, \{x\}_{x \in S}$.

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

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

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

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

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

Thus each $\mathcal{C}'_i$ is δ-clean.

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

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

For the clustering OPT' given by the above lemma, we use $\mathcal{C}_i'$ 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 wrt $A(v)$, $A(v) = A(v) \setminus \{x\}$.
   (c) (**Vertex Addition Step**): Let $Y = \{y | y \in V, y$ is 7δ-good wrt $A(v)\}$. Let $A(v) = A(v) \cup Y$.[2]
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 3 *$\forall j$, $\exists i$ such that $\mathcal{C}_j' \subseteq A_i$. Moreover, each A_i is 11δ-clean.*

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

Lemma 4 *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 wrt $N^+(v)$.*

Proof: As $v, w \in \mathcal{C}_i$, $|N^+(v) \cap \mathcal{C}_i'| \geq (1-\delta)|\mathcal{C}_i'|$, $|N^+(w) \cap \mathcal{C}_i'| \geq (1-\delta)|\mathcal{C}_i'|$ and $|N^+(w) \cap \overline{\mathcal{C}_i'}| \leq \delta|\mathcal{C}_i'|$.

Also, $(1-\delta)|\mathcal{C}_i'| \leq |N^+(v)| \leq (1+\delta)|\mathcal{C}_i'|$. Thus, we get the following two conditions.

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

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

Thus, w is 3δ-good wrt $N^+(v)$. ■

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

Lemma 5 *Given an arbitrary set X, if $v_1 \in \mathcal{C}_i'$ and $v_2 \in \mathcal{C}_j'$, then v_1 and v_2 cannot both be 3δ-good wrt X.*

Proof: Firstly if v is 3δ-good wrt some arbitrary set X, then $(1-3\delta)|X| < N^+(v) < (1+3\delta)|X|$.

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

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

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 $\mathcal{C}_j'$. So, $|N^+(v_1) \cap N^+(v_2)| \leq 2\delta\frac{1+3\delta}{1-\delta}|X|$.

However, since $\delta < 1/9$, $2\delta(1+3\delta) < (1-6\delta)(1-\delta)$ and we have a contradiction. Thus the result follows. ■

This gives us the following important corollary.

Corollary 6 *After the remove phase 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 3.

Proof of Theorem 3: 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.

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 6, 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 $\mathcal{C}_j' \subseteq A_i$.

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 wrt 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 4 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 wrt to the intermediate $A(v)$ (by mimicking the proof of lemma 4 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 vertex removal phase, $\forall x \in A'_i$, x is 3δ-good wrt 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 wrt A'_i, these 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 wrt 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 7 *The total number of external mistakes made by Cautious are less than the external mistakes made by* OPT'.

Proof: From theorem 3, 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 1 on the graph induced by $V' = \cup_i A_i$. 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 8 *The total number of internal mistakes of Cautious is* $\leq 8m_{\mathrm{OPT}}$.

Summing up results from the lemmas 7 and 8, and using lemma 2, we get the following theorem:

Theorem 9 $m_{\mathit{Cautious}} \leq 9(\frac{1}{\delta^2}+1)m_{\mathrm{OPT}}$.

4 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.

Let OPT denote the 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, $\mathrm{OPT} \geq n(n-1)/4$. So it suffices to produce a clustering that has at least $\mathrm{OPT} - \epsilon n^2$ 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 $\mathrm{OPT}(\epsilon)$ denote the optimal clustering that has all non-singleton clusters of size greater than ϵn.

Lemma 10 $\mathrm{OPT}(\epsilon) \geq \mathrm{OPT} - \epsilon n^2/2$.

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 $\epsilon \frac{n^2}{2}$. ■

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

4.1 A PTAS doubly-exponential in $1/\epsilon$

If we are willing to have a run time that is doubly-exponential in $1/\epsilon$, we can do this by reducing our problem to the General Partitioning problem of [13]. 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 $\mathrm{OPT}(\epsilon)$ 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 $\mathrm{OPT}(\epsilon)$ 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 [13] 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 approximately. We obtain a partition that has at least $\mathrm{OPT}(\epsilon) - \epsilon n^2$ agreements. The property tester runs in time exponential in $(\frac{1}{\epsilon})^{k+1}$ 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 $\pm\epsilon^2 n$ for the s_i, and up to an additive $\pm\epsilon^3 n^2$ for the $e_{i,j}$, because this introduces an additional error of at most $O(\epsilon)$. So, at most $O((1/\epsilon^3)^{1/\epsilon^2})$ 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 11 *The General Partitioning algorithm returns a clustering of graph G which has more than $\mathrm{OPT} - \epsilon n^2$ agreements with probability at least $1 - \delta$. It runs in time exponential in $(\frac{1}{\epsilon})^{1/\epsilon}$ and polynomial in n and $\frac{1}{\delta}$.*

4.2 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: 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 subsets U_i of size m at a time and try out all possible clusterings $\{U_{ij}\}$ of them, picking the one that maximizes agreements with respect to $\{W_i\}$. This gives us the PTAS.

Firstly note that if $|\mathcal{C}_{k+1}^{\mathrm{OPT}}| < \epsilon n$, 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 $|\mathcal{C}_{k+1}^{\mathrm{OPT}}| < \epsilon n$ 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 $m = \frac{88^3 \times 40}{\epsilon^{10}}(\log \frac{1}{\epsilon} + 2)$, $k = \frac{1}{\epsilon}$ and $\epsilon' = \frac{\epsilon^3}{88}$. 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 $n_{ij} = \delta^-(\mathcal{C}_i^{\mathrm{OPT}}, \mathcal{C}_j^{\mathrm{OPT}})/(|\mathcal{C}_i^{\mathrm{OPT}}||\mathcal{C}_j^{\mathrm{OPT}}|)$.

We begin by defining a measure of goodness of a clustering $\{U_{ij}\}$ of some set U_i with respect to $\{W_i\}$, that will enable us to pick the right clustering of the set U_i.

Definition 2 $U_{i1}, \ldots, U_{i(k+1)}$ *is* **ϵ'-good** *wrt* $W_1, \ldots, W_{k+1}$ *if it satisfies the following for all* $1 \leq j, l \leq k$

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

(2) $\delta^-(U_{ij}, W_l) \geq \hat{n_{jl}}|W_j||W_l| - 6\epsilon' m^2$

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

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

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 the partition that is ϵ'-good wrt $W_1, \ldots, W_{k+1}$ (by definition 2 above). If there is no such partition, choose $U_{i1}, \ldots, U_{i(k+1)}$ arbitrarily.
   (d) Let $A_j = \cup_i U_{ij}$ 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 of W given by $W_i = W \cap \mathcal{C}_i^{\mathrm{OPT}}, \forall i$. 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 $\mathrm{OPT}(\epsilon) - 2\epsilon n^2$. 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 $\mathrm{OPT} - O(\epsilon)n^2$ agreements.

We will need the following results from probability theory. Please refer to [2] 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 $0 \le \epsilon \le 1$. Then

$$Pr[|H(n,m,l) - \frac{lm}{n}| \ge \frac{\epsilon lm}{n}] \le 2e^{-\frac{\epsilon^2 lm}{2n}}$$

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

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

We will also need the following lemma:

Lemma 12 *Let Y and S be arbitrary disjoint sets and Z be a set picked from S at random. Then we have the following:* $Pr[|\delta^+(Y,Z) - \frac{|Z|}{|S|}\delta^+(Y,S)| > \epsilon'|Y||Z|] \le 2e^{\frac{-\epsilon'^2|Z|}{2}}$.

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

$$\begin{aligned} &Pr[|\delta^+(Y,Z) - |Z|\delta^+(Y,S)/|S|| > \epsilon'|Z||Y|] \\ &\le 2e^{-\epsilon'^2|Z|^2|Y|^2/2|Z||Y|^2} \le 2e^{-\epsilon'^2|Z|/2} \end{aligned}$$

■

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 13 *With probability at least* $1 - 2ke^{-\epsilon'^2\epsilon m/2}$, $\forall i$, $|W_i| \in (1 \pm \epsilon')\frac{m}{n}|\mathcal{C}_i^{\mathrm{OPT}}|$

Proof: For a given i, using Fact 1 and since $|\mathcal{C}_i^{\mathrm{OPT}}| \le \epsilon n$, $Pr[||W_i| - \frac{m}{n}|\mathcal{C}_i^{\mathrm{OPT}}|| > \epsilon'\frac{m}{n}|\mathcal{C}_i^{\mathrm{OPT}}|] \le 2e^{-\epsilon'^2 m|\mathcal{C}_i^{\mathrm{OPT}}|/2n} \le 2e^{-\epsilon'^2\epsilon m/2}$. Taking union bound over the k values of i we get the result. ■

Using Lemma 13, 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[3].

Lemma 14 *If $W_i \subset \mathcal{C}_i^{\mathrm{OPT}}$ and $W_j \subset \mathcal{C}_j^{\mathrm{OPT}}$, then with probability at least* $1 - 4e^{-\epsilon'^2\epsilon m/4}$, $\delta^+(W_i, W_j) \in \frac{m^2}{n^2}\delta^+(\mathcal{C}_i^{\mathrm{OPT}}, \mathcal{C}_j^{\mathrm{OPT}}) \pm 3\epsilon' m^2$.

Proof Sketch: We can apply lemma 12 in two steps - first to bound $\delta^+(W_i, \mathcal{C}_j^{\mathrm{OPT}})$ in terms of $\delta^+(\mathcal{C}_i^{\mathrm{OPT}}, \mathcal{C}_j^{\mathrm{OPT}})$ by considering the process of picking W_i from $\mathcal{C}_i^{\mathrm{OPT}}$, and second to bound $\delta^+(W_i, W_j)$ in terms of $\delta^+(W_i, \mathcal{C}_j^{\mathrm{OPT}})$ by fixing W_i and considering the process of picking W_j from $\mathcal{C}_j^{\mathrm{OPT}}$. Then using lemma 13, we combine the two and get the lemma. ■

Lemma 15 *With probability at least* $1 - \frac{8}{\epsilon'^2}e^{-\epsilon'^3\epsilon m/4}$, $\hat{p_i} \ge p_i - 9\epsilon'$

Proof Sketch: Note that we cannot use an argument similar to the previous lemma directly here since we are dealing with edges inside the same set. We use the following trick: consider the partition of $\mathcal{C}_i^{\mathrm{OPT}}$ into $\frac{1}{\epsilon'}$ subsets of size $\epsilon' n'$ each, where $n' = |\mathcal{C}_i^{\mathrm{OPT}}|$. The idea is to bound the number of positive edges between every pair of subsets of $\mathcal{C}_i^{\mathrm{OPT}}$ using argument in the previous lemma and adding these up to get the result. ■

Now let $U_{ij} = U_i \cap \mathcal{C}_j^{\mathrm{OPT}}$. The following lemma shows that for all i, with a high probability all U_{ij}s are ϵ'-good wrt $\{W_i\}$. So we will be able to find ϵ'-good partitions for most U_is.

Lemma 16 *For a given i, let $U_{ij} = U_i \cap \mathcal{C}_j^{\mathrm{OPT}}$, then with probability at least* $1 - 32k\frac{1}{\epsilon'^2}e^{-\epsilon'^3\epsilon m/4}$, $\forall j \le k$, *$\{U_{ij}\}$ are ϵ'-good wrt $\{W_j\}$.*

Proof Sketch: The first and second conditions of Definition 2 can be obtained by applying an argument similar to lemmas 15 and 14 respectively.

In order to obtain the third condition, we consider $\delta^+(x, U_{ij})$ as 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 $\mathcal{C}_j^{\mathrm{OPT}}$ and is adjacent to x. Then an application of Chernoff bound followed by union bound gives us the condition. ■

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:

[3] Please refer to [5] for full proofs of the lemmas.

Theorem 17 *If* $|\mathcal{C}_{k+1}^{\mathrm{OPT}}| < \epsilon n$*, then* $A \geq \mathrm{OPT} - 3\epsilon n^2$ *with probability at least* $1-\epsilon$.

Proof: From lemma 16, the probability that we were not able to find a ϵ'-good partition of U_i wrt $W_1, \cdots, W_k$ is at most $32\frac{1}{\epsilon'^2}e^{-\epsilon'^3\epsilon m/4}$. By our choice of m, this is at most $\epsilon^2/4$. So, with probability at least $1-\epsilon/2$, at most $\epsilon/2$ 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 $\epsilon n^2/2$ from this value to obtain the actual number of agreements, since each U_i can effect 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 wrt $\{W_i\}$ (condition (3)),

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

The last follows from the fact that U_{bj} is good wrt $\{W_i\}$ (condition (1)). From Lemma 13,

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

Thus we bound $\sum_a \delta^+(A_j, U_{aj})$ as $\sum_a \delta^+(A_j, U_{aj}) \geq \hat{p_j}(1-\epsilon')^2\frac{|\mathcal{C}_j^{\mathrm{OPT}}|^2}{2} - 18\epsilon' n^2 - 3\epsilon' n|A_j|$.
Now using Lemma 15, the total number of agreements is at least

$$\begin{aligned}&\sum_j\{\hat{p_j}(1-\epsilon')^2\frac{|\mathcal{C}_j^{\mathrm{OPT}}|^2}{2}\} - 18\epsilon' n^2 k - 3\epsilon' n^2 \\ &\geq \sum_j\{(p_j - 9\epsilon')(1-\epsilon')^2\frac{|\mathcal{C}_j^{\mathrm{OPT}}|^2}{2}\} - 18\epsilon' n^2 k - 3\epsilon' n^2\end{aligned}$$

Hence, $A^+ \geq \mathrm{OPT}^+ - 11\epsilon' k n^2 - 21\epsilon' n^2 k \geq \mathrm{OPT}^+ - 32\epsilon' n^2 k$.
Similarly, consider the negative edges in A. Using lemma 14 to estimate $\delta^-(U_{ai}, U_{bj})$, we get,

$$\begin{aligned}\sum_{ab}\delta^-(U_{ai}, U_{bj}) \quad \geq \quad & \delta^-(\mathcal{C}_i^{\mathrm{OPT}}, \mathcal{C}_j^{\mathrm{OPT}}) \\ & -9\epsilon' n^2 - 2\epsilon' n|A_i| - \epsilon' n|A_j|\end{aligned}$$

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

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

The algorithm can fail in four situations: more than $\epsilon/2$ U_is do not have an ϵ'-good partition with probability at most $\epsilon/2$, lemma 13 does not hold for some W_i with probability at most $2ke^{-\epsilon'^2\epsilon m/2}$, lemma 15 does not hold for some i with probability at most $\frac{8k}{\epsilon'^2}e^{-\epsilon'^3\epsilon m/4}$ or lemma 14 does not hold for some pair i, j with probability at most $4k^2e^{-\epsilon'^2\epsilon m/4}$. The latter three quantities are at most $\epsilon/2$ by our choice of m. So, the algorithm succeeds with probability greater than $1-\epsilon$. ■

Now we need to argue for the case when $|\mathcal{C}_{k+1}^{\mathrm{OPT}}| \geq \epsilon n$. Notice that in this case, using an argument similar to lemma 13, we can show that $|W_{k+1}| \geq \frac{\epsilon m}{2}$ with a very high probability. This is good because, now with a high probability, $U_{i(k+1)}$ will also be ϵ'-good wrt 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 17 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 17. Thus our strategy will be to run Algorithm *Divide&Choose* once assuming that $|\mathcal{C}_{k+1}^{\mathrm{OPT}}| \geq \epsilon n$ and then again assuming that $|\mathcal{C}_{k+1}^{\mathrm{OPT}}| \leq \epsilon n$, 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 $O(n^2k^m)$. Since there are k^m different partitions of W, the total running time of the algorithm is $O(n^2k^{2m}) = O(n^2e^{O(\frac{1}{\epsilon^{10}}\log(\frac{1}{\epsilon}))})$. This gives us the following theorem:

Theorem 18 *For any* $\delta \in [0,1]$*, using* $\epsilon = \frac{\delta}{3}$*, Algorithm Divide&Choose runs in time* $O(n^2e^{O(\frac{1}{\delta^{10}}\log(\frac{1}{\delta}))})$ *and with probability at least* $1-\frac{\delta}{3}$ *produces a clustering with number of agreements at least* $\mathrm{OPT} - \delta n^2$.

5 Minimizing disagreements in $[-1,1]$-weighted graphs

In section 3, we developed an algorithm for minimizing disagreements in a graph with $+1$ and -1 weighted edges. Now we consider the situation in which edge weights lie in the interval $[-1,1]$.

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, 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.

It turns out that any algorithm that finds a good clustering in a $\{-1,1\}-$graph also works well in the $[-1,1]$ case under a linear cost function.

Theorem 19 *Let A be an algorithm that produces a clustering on a $\{-1,1\}-$graph 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, with the linear cost function.*

Proof: Let G be a $[-1,1]-$graph, and let G' be the $\{-1,1\}-$graph 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 $\{-1,1\}$ penalty model and m in the new $[-1,1]$ penalty model.

Then, $m'_{\mathrm{OPT}'} \le m'_{\mathrm{OPT}} \le 2m_{\mathrm{OPT}}$. The latter is because OPT incurs a greater penalty of 1 in m' as compared to m only when a positive edge is between clusters or a negative edge inside a cluster. In both these situations, OPT incurs a cost of at least $1/2$ in m and at most 1 in m'. This gives us the above equation.

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

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,

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

■

Another natural cost model is one in which an edge of weight x incurs a cost of $|x|$ when it is clustered improperly (inside a cluster if $x<0$ or between clusters of $x>0$) and a cost of 0 when it is correct. We do not know of any good approximation in this case (see Section 7).

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 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 (whp) 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 [16] in the machine learning context, and even closer to the analysis of Condon and Karp for graph partitioning [11]). In fact, this problem nearly matches a special case of the planted-partition problem of McSherry [18]. 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 whp 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 Open Problems and Concluding Remarks

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 5 we extended some of our results to the case of real-valued labels, under a linear cost metric.

One interesting open question is to find good approximations for the case when edge weights are in $\{-1, 0, +1\}$ (equivalently, edges are labeled $+$ or $-$ but G is not necessarily fully-connected) without considering the 0-edges as "half a mistake". In that context it is still easy to cluster if a perfect clustering exists: the same simple strategy works of removing 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 to the number of agreements. However, this does not imply a PTAS for maximizing agreements because OPT might be $o(n^2)$ in this variant.

A further generalization of the problem is to consider unbounded 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. Here the number of disagreements could be defined as the total weight of positive edges between clusters and negative edges inside clusters, and agreements defined analogously. Again, we do not know of any good algorithm for approximating the number of disagreements in this case. We believe the problem of maximizing agreements should be APX-hard for this generalization, but have not been able to prove it. We can show, however, that a PTAS would give an n^ϵ approximation algorithm for k-coloring, for any constant k.[4] The incomplete $\{-1, 0, +1\}$ graph model seems to be as hard as this problem.

For the original problem on a fully connected $\{+, -\}$ graph, another question is whether one can approximate the *correlation*: the number of agreements minus the number of disagreements. It is easy to show that OPT must be $\Omega(n)$ for this measure, but we do not know of any good approximation. It would also be good to improve our (currently fairly large) constant for approximating disagreements.

[4] For details refer to [5].

References

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

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

[3] S. Arora, A. Frieze, and H. Kaplan. A new rounding procedure for the assignment problem with applications to dense graph arrangements. In *Proc. IEEE FOCS* , pages 21–30, 1996.

[4] S. Arora, D. Karger, and M. Karpinski. Polynomial time approximation schemes for dense instances of np-hard problems. In *ACM Symposium on Theory of Computing*, 1995.

[5] N. Bansal, A. Blum, and S. Chawla. Correlation clustering (http://www.cs.cmu.edu/˜shuchi/papers/clusteringfull.ps). *Manuscript*, 2002.

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

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

[8] W. Cohen and A. McCallum. Personal communication, 2001.

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

[10] W. Cohen and J. Richman. 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)*, 2002.

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

[12] F. de la Vega. Max-cut has a randomized approximation scheme in dense graphs. *Random Structures and Algorithms*, 8(3):187–198, 1996.

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

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

[15] K. Jain and V. Vazirani. Primal-dual approximation algorithms for metric facility location and k-median problem. In *Proc. 40th IEEE FOCS*, 1999.

[16] M. Kearns. Efficient noise-tolerant learning from statistical queries. In *Proceedings of the Twenty-Fifth Annual ACM Symposium on Theory of Computing*, pages 392–401, 1993.

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

[18] F. McSherry. Spectral partitioning of random graphs. In *FOCS*, pages 529–537, 2001.

[19] M. Parnas and D. Ron. Testing the diameter of graphs. In *Proceedings of RANDOM*, pages 85–96, 1999.

[20] L. Schulman. Clustering for edge-cost minimization. In *ACM STOC*, pages 547–555, 2000.