Online Correlation Clustering

Claire Mathieu, Ocan Sankur, Warren Schudy

Symposium on Theoretical Aspects of Computer Science 2010 (Nancy, France), pp. 573-584
www.stacs-conf.org

ONLINE CORRELATION CLUSTERING

CLAIRE MATHIEU [1] AND OCAN SANKUR [1,2] AND WARREN SCHUDY [1]

[1] Department of Computer Science, Brown University, 115 Waterman Street, Providence, RI 02912

[2] Ecole Normale Supérieure, 45, rue d'Ulm, 75005 Paris, France

Abstract. We study the online clustering problem where data items arrive in an online fashion. The algorithm maintains a clustering of data items into similarity classes. Upon arrival of v, the relation between v and previously arrived items is revealed, so that for each u we are told whether v is similar to u. The algorithm can create a new cluster for v and merge existing clusters.

When the objective is to minimize disagreements between the clustering and the input, we prove that a natural greedy algorithm is O(n)-competitive, and this is optimal.

When the objective is to maximize agreements between the clustering and the input, we prove that the greedy algorithm is .5-competitive; that no online algorithm can be better than .834-competitive; we prove that it is possible to get better than 1/2, by exhibiting a randomized algorithm with competitive ratio .5+c for a small positive fixed constant c.

1. Introduction

We study online correlation clustering. In correlation clustering [2, 15], the input is a complete graph whose edges are labeled either *positive*, meaning similar, or *negative*, meaning dissimilar. The goal is to produce a clustering that agrees as much as possible with the edge labels. More precisely, the output is a clustering that maximizes the number of agreements, *i.e.*, the sum of positive edges within clusters and the negative edges between clusters. Equivalently, this clustering minimizes the disagreements. This has applications in information retrieval, e.g. [8, 10].

In the online setting, vertices arrive one at a time and the total number of vertices is unknown to the algorithm *a priori*. Upon the arrival of a vertex, the labels of the edges that connect this new vertex to the previously discovered vertices are revealed. The algorithm updates the clustering while preserving the clusters already identified (it is not permitted to split any pre-existing cluster). Motivated by information retrieval applications, this online model was proposed by Charikar, Chekuri, Feder and Motwani [5] (for another clustering problem). As in [5], our algorithms maintain *Hierarchical Agglomerative Clusterings* at all times; this is well suited for the applications of interest.

1998 ACM Subject Classification: F.2.2 Nonnumerical Algorithms and Problems.
Key words and phrases: correlation clustering, online algorithms.
Part of this work was funded by NSF grant CCF 0728816.

The problem of correlation clustering was introduced by Ben-Dor et al. [3] to cluster gene expression patterns. Unfortunately, it was shown that even the offline version of correlation clustering is NP-hard [15, 2]. The following are the two approximation problems that have been studied [2, 7, 1]: Given a complete graph whose edges are labeled positive or negative, find a clustering that minimizes the number of disagreements, or maximizes the number of agreements. We will call these problems MinDisAgree and MaxAgree respectively. Bansal et al. [2] studied approximation algorithms both for minimization and maximization problems, giving a constant factor algorithm for MinDisAgree, and a *Polynomial Time Approximation Scheme (PTAS)* for MaxAgree. Charikar et al. [7] proved that MinDisAgree is APX-hard and gave a factor 4 approximation. Ailon et al. [1] presented a randomized factor 2.5 approximation for MinDisAgree, which is currently the best known factor. The problem has attracted significant attention, with further work on several variants [9, 6, 11, 13, 3, 12, 14].

In this paper, we study online algorithms for MinDisAgree and MaxAgree. We prove that MinDisAgree is essentially hopeless in the online setting: the natural greedy algorithm is $O(n)$-competitive, and this is optimal up to a constant factor, even with randomization (Theorem 3.4). The situation is better for MaxAgree: we prove that the greedy algorithm is a .5-competitive (Theorem 2.1), but that no algorithm can be better than 0.803 competitive (0.834 for randomized algorithms, see Theorem 2.2). What is the optimal competitive ratio? We prove that it is better than .5 by exhibiting an algorithm with competitive ratio $0.5 + \epsilon_0$ where ϵ_0 is a small absolute constant (Theorem 2.6). Thus Greedy is not always the best choice!

More formally, let $v_1, \dots, v_n$ denote the sequence of vertices of the input graph, where n is not known in advance. Between any two vertices, v_i and v_j for $i \neq j$, there is an edge labeled positive or negative. In MinDisAgree (resp. MaxAgree), the goal is to find a clustering $\mathcal{C}$, *i.e.* a partition of the nodes, that minimizes the number of disagreements $\text{cost}(\mathcal{C})$: the number of negative edges within clusters plus the number of positive edges between clusters (resp. maximizes the number of agreements $\text{profit}(\mathcal{C})$: the number of positive edges within clusters plus the number of negative edges between clusters). Although these problems are equivalent in terms of optimality, they differ from the point of view of approximation. Let OPT denote the optimum solution of MinDisAgree and of MaxAgree.

In the online setting, upon the arrival of a new vertex, the algorithm updates the current clustering: it may either create a new singleton cluster or add the new vertex to a pre-existing cluster, and may decide to merge some pre-existing clusters. It is not allowed to split pre-existing clusters.

A c-competitive algorithm for MinDisAgree outputs, on any input σ, a clustering $\mathcal{C}(\sigma)$ such that $\text{cost}(\mathcal{C}(\sigma)) \leq c \cdot \text{cost}(\text{OPT}(\sigma))$. For MaxAgree, we must have $\text{profit}(\mathcal{C}(\sigma)) \geq c \cdot \text{profit}(\text{OPT}(\sigma))$. (When the algorithm is randomized, this must hold in expectation).

2. Maximizing Agreements Online

2.1. Competitiveness of Greedy

For subsets of vertices S and T we define $\Gamma(S, T)$ as the set of edges between S and T. We write $\Gamma^+(S, T)$ (resp. $\Gamma^-(S, T)$) for the set of positive (resp. negative) edges of $\Gamma(S, T)$.

We define the *gain* of merging S with T as the change in the profit when clusters S and T are merged:

$$\text{gain}(S,T) = |\Gamma^+(S,T)| - |\Gamma^-(S,T)| = 2|\Gamma^+(S,T)| - |S||T|.$$

We present the following greedy algorithm for online correlation clustering.

Algorithm 1 Algorithm GREEDY

1: **Upon the arrival of** vertex v **do**
2: Put v in a new cluster consisting of $\{v\}$.
3: **while** there are two clusters C, C' such that $\text{gain}(C,C') > 0$ **do**
4: Merge C and C'
5: **end while**
6: **end for**

Theorem 2.1. *Let OPT denote the offline optimum.*

- *For every instance, profit*(GREEDY) ≥ 0.5 *profit*(OPT).
- *There are instances with profit*(GREEDY) $\leq (0.5 + o(1))$*profit*(OPT).

2.2. Bounding the optimal competitive ratio

Theorem 2.2. *The competitive ratio of any randomized online algorithm for* MAXAGREE *is at most* 0.834. *The competitive ratio of any deterministic online algorithm for* MAXAGREE *is at most* 0.803.

The proof uses Yao's Min-Max Theorem [4] (maximization version).

Theorem 2.3 (Yao's Min-Max Theorem)**.** *Fix a distribution D over a set of inputs $(I_\sigma)_\sigma$. The competitive ratio of any randomized online algorithm is at most*

$$\max\{\frac{E_I[\textit{profit}(\mathcal{A}(I))]}{E_I[\textit{profit}(OPT(I))]} : \mathcal{A} \textit{ deterministic online algorithm}\},$$

where the expectations are over a random input I drawn from distribution D.

To prove Theorem 2.2, we first define two generic inputs that we will use to apply Theorem 2.3. The first input is a graph G_1 with $2m$ vertices and all positive edges between them The second input is a graph with $6m$ vertices defined as follows. The first $2m$ vertices have all positive edges between them, the next $2m$ vertices have all positive edges between them, and the last $2m$ vertices also have all positive edges between them. In each of these three sets G_1, G_2, G_3 of $2m$ vertices, half are labelled "left side" vertices and the other half are labelled "right side" vertices. All edges between left vertices are positive, but edges between a vertex u on the left side of G_i and a vertex v on the right side of G_j, $j \neq i$, are all negative.

The online algorithm cannot distinguish between the two inputs until time $2m + 1$, so it must hedge against two very different possible optimal structures.

2.3. Beating Greedy

2.3.1. *Designing the algorithm.* Our algorithm is based on the observation that Algorithm Greedy always satisfies at least half of the edges. Thus, if profit(OPT) is less than $(1-\alpha/2)|E|$ for some constant α, then the profit of Greedy is better than half of optimal. We design an algorithm called Dense, parameterized by constants α and τ, such that if profit(OPT) is greater than $(1-\alpha/2)|E|$, then the approximation factor is at least $0.5+\eta$ for some positive constant η. We use both algorithms Greedy and Dense to define Algorithm 2.

Theorem 2.4. *Let $\alpha \in (0,1)$, $\tau > 1$ and $\eta \in (0, \frac{1}{2})$ be such that*

$$\eta \leq 1.5 - \tau^2 - ((2\sqrt{3}+9/2)\alpha^{1/4} + \frac{\alpha^{1/4}}{1-\alpha^{1/4}} + \alpha/2)2\frac{2\tau-1}{(\tau-1)}. \tag{2.1}$$

Then, for every instance such that $OPT \geq (1-\alpha/2)E$, Algorithm $\text{Dense}_{\alpha,\tau}$ has profit at least $(1/2+\eta)OPT$.

Using Theorem 2.4 we can bound the competitive ratio of Algorithm 2.

Corollary 2.5. *Let α, τ and η be as above, and let $p = \alpha/(2+2\eta(2-\alpha))$. Then Algorithm 2 has competitive ratio at least $\frac{1}{2} + \frac{\alpha\eta/2}{1+2\eta(1-\alpha/2)}$.*

Corollary 2.6. *For $\alpha = 10^{-12}$, $\tau = 1.0946$, $\eta = 0.0555$ and $p = 4,5 \cdot 10^{-13}$, Algorithm 2 is $\frac{1}{2} + 2 \cdot 10^{-14}$-competitive.*

Algorithm 2 A $\frac{1}{2}+\epsilon_0$-competitive algorithm

Given p, α, τ,
With probability $1-p$, run Greedy,
With probability p, run $\text{Dense}_{\alpha,\tau}$.

Algorithm 3 Algorithm $\text{Dense}_{\alpha,\tau}$

Let $\mathcal{C} = \widehat{\text{OPT}}_1$ and for every cluster $D \in \mathcal{C}$, let $\text{repr}_1(D) := D \in \widehat{\text{OPT}}_1$.
Upon the arrival of a vertex v at time t **do**
  Put v in a new cluster $\{v\}$.
  if $t = t_i$ for some i **then**
    for every cluster D in $\widehat{\text{OPT}}_i$ **do**
      Define a cluster D'' obtained by merging the restriction of D to $\{t_{i-1}, \ldots, t_i\}$ with every cluster $C \in \mathcal{C}$ in $\{1, \ldots, t_{i-1}\}$ such that $\text{repr}_{i-1}(C)$ is defined and is half-contained in D.
      If D'' is not empty, set $\text{repr}_i(D'') := D \in \widehat{\text{OPT}}_i$.
    end for
  end if
end for

How do we define algorithm Dense? Using the PTAS of [2], one can compute offline a factor $(1-\alpha/2)$ approximative solution OPT' of any instance of MaxAgree in polynomial

time. We will design algorithm DENSE so that it guarantees an approximation factor of $0.5+\eta$ whenever $\text{profit}(\text{OPT}') \geq (1-\alpha)|E|$. Since $\text{profit}(\text{OPT}) \geq (1-\alpha/2)|E|$ implies that $\text{profit}(\text{OPT}') \geq (1-\alpha)|E|$, Theorem 2.4 will follow.

We say that OPT'_t is *large* if $\text{profit}(\text{OPT}'_t) \geq (1-\alpha)|E|$. We define a sequence $(t_i)_i$ of *update times* inductively as follows: By convention $t_0 = 0$. Time t_1 is the earliest time $t \geq 100$ such that OPT'_t is large. Assume t_i is already defined, and let j be such that $\tau^{j-1} \leq t_i < \tau^j$. If OPT'_{τ^j} is large, then $t_{i+1} = \tau^j$, else t_{i+1} is the earliest time $t \geq \tau^j$ such that OPT'_t is large. Let $t_1, t_2, \ldots, t_K$ be the resulting sequence. We will note, with an abuse of notation, OPT'_i instead of OPT'_{t_i} for $1 \leq i \leq K$.

We say that a cluster A is *half-contained* in B if $|A \cap B| > |A|/2$. Let $\epsilon = \alpha^{1/4}$. For each t_i, we inductively define a near optimal clustering of the nodes $[1, t_i]$. For the base case, let $\widehat{\text{OPT}}_1$ be the clustering obtained from OPT'_1 by keeping the $1/\epsilon^2$ largest clusters and splitting the other clusters into singletons. For the general case, to define $\widehat{\text{OPT}}_i$ given $\widehat{\text{OPT}}_{i-1}$, mark the clusters of OPT'_i as follows. For any D in OPT'_i, mark D if either one of the $1/\epsilon^2 - 1/\epsilon$ largest clusters of $\widehat{\text{OPT}}_{i-1}$ is half-contained in D, or D is one of the $1/\epsilon$ largest clusters OPT'_i. Then $\widehat{\text{OPT}}_i$ contains all the marked clusters of OPT'_i and the rest of the vertices in $[1, t_i]$ as singleton clusters. (Note that, by definition, any $\widehat{\text{OPT}}_i$ contains at most $1/\epsilon^2$ non-singleton clusters; this will be useful in the analysis.)

Note that DENSE only depends on parameters α and τ indirectly via the definition of update times and of $\widehat{\text{OPT}}$.

2.3.2. *Analysis: Proof of Theorem 2.4.* The analysis is by induction on i, assuming that we start from clustering $\widehat{\text{OPT}}_i$ at time t_i, then apply the above algorithm from time t_i to the final time t. If $i = 1$ this is exactly our algorithm, and if $i = K$ then this is simply $\widehat{\text{OPT}}_K$; in general it is a mixture of the two constructions.

More formally, define a forest $\mathcal{F}$ (at time t) with one node for each $t_i \leq t$ and cluster of $\widehat{\text{OPT}}_i$. The node associated to a cluster A of $\widehat{\text{OPT}}_{i-1}$ is a child of the node associated to a cluster B of $\widehat{\text{OPT}}_i$ if and only if A is half-contained in B. With a slight abuse of notation, we define the following clustering $\mathcal{F}$ associated to the forest. There is one cluster T for each tree of the forest: for each node A of the tree, if i is such that $A \in \widehat{\text{OPT}}_i$, then cluster T contains $A \cap (t_{i-1}, t_i]$. This defines T.

One interpretation of DENSE is that at all times t, there is an associated forest and clustering $\mathcal{F}$; and our algorithm DENSE simply maintains it. See Figure 1 for an example.

Lemma 2.7. *Algorithm 3 is an online algorithm that outputs clustering $\mathcal{F}$ at time t.*

Let $\mathcal{F}_i$ be the forest obtained from $\mathcal{F}$ by erasing every node associated to clusters of $\widehat{\text{OPT}}_j$ for every $j < i$. With a slight abuse of notation, we define the following clustering $\mathcal{F}_i$ associated to that forest: there is one cluster C for each tree of the forest defined as follows. For each node A of the tree, let $k \geq i$ be such that $A \in \widehat{\text{OPT}}_k$: then C contains $A \cap (t_{k-1}, t_k]$ if $k > i$, and C contains A if $k = i$. This defines a sequence of clusterings such that $\mathcal{F}_1 = \mathcal{F}$ is the output of the algorithm, and $\mathcal{F}_K = \widehat{\text{OPT}}_K$.

Lemma 2.8 (Main lemma)**.** *For any $2 \leq i \leq K$,*

$$\textit{cost}(\mathcal{F}_{i-1}) - \textit{cost}(\mathcal{F}_i) \leq \left((4+2\sqrt{3})\epsilon + \frac{\epsilon}{1-\epsilon}\right) t_i t_K.$$

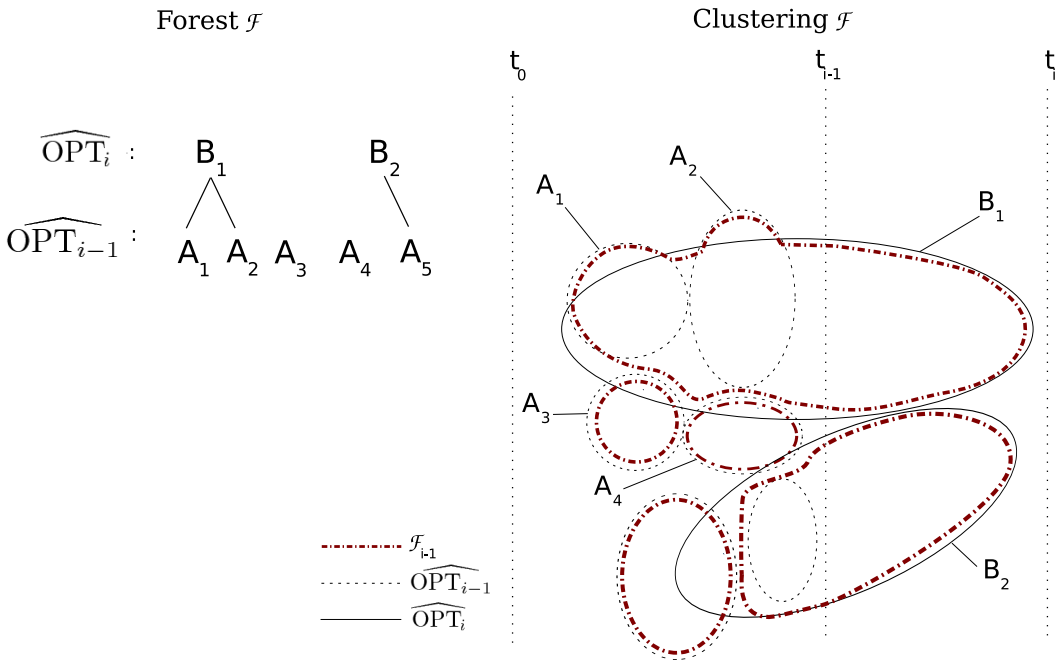

Figure 1: An example of a forest $\mathcal{F}$ given in left, and the corresponding clustering given in right. Here, we have $\widehat{\mathrm{OPT}_i} = \{B_1, B_2\}$ and $\widehat{\mathrm{OPT}_{i-1}} = \{A_1, \ldots, A_5\}$.

We defer the proof of Lemma 2.8 to next section. Assuming Lemma 2.8, we upper-bound the cost of clustering $\mathcal{F}$.

Lemma 2.9 (Lemma 14, [2])**.** *For any $0 < c < 1$ and clustering $\mathcal{C}$, let $\mathcal{C}'$ be the clustering obtained from $\mathcal{C}$ by splitting all clusters of $\mathcal{C}$ of size less than cn, where n is the number of vertices. Then $cost(\mathcal{C}') \leq cost(\mathcal{C}) + cn^2/2$.*

Lemma 2.10. $cost(\mathcal{F}) \leq ((2\sqrt{3} + 9/2)\epsilon + \frac{\epsilon}{1-\epsilon} + \epsilon^4/2)\frac{2\tau-1}{\tau-1}t_K^2$.

Proof. We write: $\mathrm{cost}(\mathcal{F}) = \mathrm{cost}(\widehat{\mathrm{OPT}_K}) + \sum_{i=2}^{K}(\mathrm{cost}(\mathcal{F}_{i-1}) - \mathrm{cost}(\mathcal{F}_i))$.By definition, $\widehat{\mathrm{OPT}_K}$ contains the $1/\epsilon$ largest clusters of OPT'_K. Then the remaining clusters of OPT'_K are of size at most ϵt_K. By Lemma 2.9, the cost of $\widehat{\mathrm{OPT}_K}$ is at most $\mathrm{cost}(\mathrm{OPT}'_K) + \epsilon t_K^2/2 \leq (\alpha + \epsilon)t_K^2/2$. Applying Lemma 2.8, and summing over $2 \leq i \leq K$, we get

$$\mathrm{cost}(\mathcal{F}) \leq (\alpha + \epsilon)t_K^2/2 + \left((4 + 2\sqrt{3})\epsilon + \frac{\epsilon}{1-\epsilon}\right)\sum_i t_i t_K.$$

By definition of the update times $(t_i)_i$, for any $j > 0$ there exists at most one t_i such that $\tau^j \leq t_i < \tau^{j+1}$. Let L be such that $\tau^L \leq t_K < \tau^{L+1}$. Then

$$\sum_{1 \leq i \leq K} t_i \leq \sum_{1 \leq i \leq K-1} t_i + t_K \leq \sum_{1 \leq j \leq L} \tau^j + t_K \leq \frac{\tau^{L+1}}{\tau - 1} + t_K \leq \frac{2\tau - 1}{\tau - 1} t_K.$$

Hence the desired bound on $\mathrm{cost}(\mathcal{F})$. ∎

Proof of Theorem 2.4. Fix an input graph of size n, such that $\text{profit}(\text{OPT}) \geq (1-\alpha/2)\binom{n}{2}$. By Lemma 2.10, at time t_K, Algorithm 3 has clustering $\mathcal{F}$ with $\text{cost}(\mathcal{F}) \leq O(\epsilon)\frac{2\tau-1}{\tau-1}t_K^2$.

By definition of the update times, $n < \tau t_K$. To guarantee a competitive ratio of $0.5+\eta$, for some η, the cost must not exceed $(0.5-\eta)\binom{n}{2}$ at time n, when all vertices $t_K+1,\dots,n$ are added as singleton clusters. The number of new edges added to the graph between times t_K and n is $\binom{n-t_K}{2} + t_K(n-t_K)$. We must have

$$\frac{2\tau-1}{\tau-1}O(\epsilon)t_K^2 + \binom{n-t_K}{2} + t_K(n-t_K) \leq (0.5-\eta)\binom{n}{2}, \tag{2.2}$$

for some $0 < \eta < 0.5$. Using the fact that $n - t_K \leq (\tau-1)t_K$ and $t_K \leq n-1$, to satisfy (2.2), it suffices to have

$$\frac{2\tau-1}{\tau-1}O(\epsilon)t_K^2 + t_K^2(\tau-1)^2/2 + (\tau-1)t_K^2 \leq (0.5-\eta)t_K^2/2,$$

which is equivalent to (2.1). Moreover we have the following natural constraints on constants η, ϵ and τ: $0 < \eta < 0.5$, $0 < \epsilon < 1$, and $\tau > 1$. Then, for any set of values of constants η, ϵ, τ verifying those constraints, Algorithm DENSE is $0.5+\eta$-competitive. ∎

2.3.3. *The core of the analysis: proof of Lemma 2.8.*

Lemma 2.11. *Let $\mathcal{S}^i$ be the set of vertices of the non-singleton clusters that are not among the $1/\epsilon^2 - 1/\epsilon$ largest clusters of $\widehat{OPT_{i-1}}$. Then $|\mathcal{S}^i| \leq \frac{\epsilon}{1-\epsilon}t_{i-1}$.*

Proof. Let C be a cluster of $\widehat{\text{OPT}_{i-1}}$, such that $C \subseteq \mathcal{S}^i$. Then $|C| \leq (1/\epsilon^2 - 1/\epsilon)^{-1}t_{i-1}$. Since there are at most $1/\epsilon$ such clusters, the number of vertices of these are at most $1/\epsilon(1/\epsilon^2 - 1/\epsilon)^{-1}t_{i-1}$. ∎

Notation 2.12. For any $i \neq j$, and a cluster B of OPT'_i, we denote by $\gamma_B^{i,j}$ the square root of the number of edges of $[1, t_{\min(i,j)}] \times [1, t_{\min(i,j)}]$, adjacent to at least one node of B, and which are classified differently in OPT'_i and in OPT'_j.

We refer to non singleton clusters as *large* clusters.

Lemma 2.13. *Let $\mathcal{T}^i$ be the set of vertices of those $1/\epsilon^2 - 1/\epsilon$ largest clusters of $\widehat{OPT_{i-1}}$ that are not half-contained in any cluster of OPT'_i. Then $|\mathcal{T}^i| \leq \sqrt{6}\sum_{\text{large } C \in \widehat{OPT_{i-1}}} \gamma_C^{i,i-1}$.*

Let B be a cluster of $\widehat{\text{OPT}_i}$. For any $j \leq i$, we define $\mathcal{C}_j(B)$ as the cluster associated with the tree of $\mathcal{F}_j$ that contains B. For any $B \in \widehat{\text{OPT}_i}$, we call $\mathcal{C}_{i-1}(B)$ the *extension* of $\mathcal{C}_i(B)$ to $\mathcal{F}_{i-1}$. By definition of $\mathcal{F}_i$, the following lemma is easy.

Lemma 2.14. *For any $B \in \widehat{OPT_i}$, the restriction of $\mathcal{C}_{i-1}(B)$ to $(t_{i-1}, t_K]$ is equal to the restriction of $\mathcal{C}_i(B)$ to $(t_{i-1}, t_K]$.*

Let $(A_j)_j$ denote the clusters of $\widehat{\text{OPT}_{i-1}}$ that are half-contained in B. We define $\delta^i(B)$ as the symmetric difference of the restriction of B to $[1, t_{i-1}]$ and $\cup_j A_j$:

$$\delta^i(B) = (B \cap [1, t_{i-1}])\Delta \cup_j A_j.$$

Lemma 2.15. *For any cluster C_i of $\mathcal{F}_i$, let C_i' denote the extension of C_i to $\mathcal{F}_{i-1}$. Then*

$$\bigcup_{C_i \in \mathcal{F}_i} C_i \setminus C_i' \subseteq \mathcal{S}^i \cup \mathcal{T}^i \cup \bigcup_{\text{large } B \in \widehat{OPT_i}} \delta^i(B)$$

Proof. By Lemma 2.14, the partition of the vertices $(t_{i-1}, t_K]$ is the same for C_i as for C_i'. So C_i and C_i' only differ in the vertices of $[1, t_{i-1})$:

$$\bigcup_{C_i \in \mathcal{F}_i} C_i \setminus C_i' \subseteq \bigcup_{B \in \widehat{\mathrm{OPT}_i}} \delta^i(B).$$

We will show that for a singleton cluster B of $\widehat{\mathrm{OPT}_i}$, $\delta^i(B)$ is included in $\mathcal{S}^i \cup \mathcal{T}^i \bigcup_{\text{large } B \in \widehat{\mathrm{OPT}_i}} \delta^i(B)$, which yields the lemma.

Let $B = \{v\}$ be a singleton cluster of $\widehat{\mathrm{OPT}_i}$ such that $\delta^i(B) \neq \{\}$. A non-singleton cluster cannot be half-contained in a singleton cluster so we conclude no clusters are half-contained in B and hence $\delta^i(B) = \{v\}$. By definition of $\delta^i(B)$, $v \in [1, t_{i-1}]$. So there exists a cluster A of $\widehat{\mathrm{OPT}_{i-1}}$ that contains v. Clearly A is not a singleton since otherwise $\delta^i(B)$ would be $\{\}$. There are two cases.

First, if A is half-contained in a cluster $B' \neq B$ of $\widehat{\mathrm{OPT}_i}$ then cluster B' is necessarily large since it contains more than one vertex of A. Then we have $v \in \delta^i(B')$.

Second, if A is not half-contained in any cluster of $\widehat{\mathrm{OPT}_i}$ then $A \subseteq \mathcal{S}^i \cup \mathcal{T}^i$. In fact, if A is half-contained in a cluster of OPT_i' which is split into singletons in $\widehat{\mathrm{OPT}_i}$, then A is not one of the $1/\epsilon^2 - 1/\epsilon$ largest clusters of $\widehat{\mathrm{OPT}_{i-1}}$, and $A \subseteq \mathcal{S}^i$. If A is not half-contained in any cluster of OPT_i', then $A \subseteq \mathcal{T}^i$ if A is one of the $1/\epsilon^2 - 1/\epsilon$ largest clusters of $\widehat{\mathrm{OPT}_{i-1}}$ and $A \subseteq \mathcal{S}^i$ otherwise. ∎

Lemma 2.16. *For any large cluster B of $\widehat{OPT_i}$, $|\delta^i(B)| \leq 2\sqrt{2}\gamma_B^{i,i-1}$.*

Proof. Let B' denote the restriction of B to $[1, t_{i-1}]$. We first show that

$$1/2(|\cup_j A_j \setminus B'|)^2 \leq (\gamma_B^{i,i-1})^2.$$

Observe that $(\gamma_B^{i,i-1})^2$ includes all edges uv such that one of the following two cases occurs.

First, if $u \in A_j \setminus B$ and $v \in A_j \cap B$: such edges are internal in the clustering OPT_{i-1}' but external in the clustering OPT_i'. The number of edges of this type is $\sum_j |A_j \setminus B| \cdot |A_j \cap B|$. Since A_j is half-contained in B, this is at least $\sum_j |A_j \setminus B|^2$.

Second, if $u \in A_j \cap B$ and $v \in A_k \cap B$ with $j \neq k$: such edges are external in the clustering OPT_{i-1}' but internal in the clustering OPT_i'. The number of edges of this type is $\sum_{j<k} |A_j \cap B| \cdot |A_k \cap B| \geq \sum_{j<k} |A_j \setminus B| \cdot |A_k \setminus B|$.

Summing, it is easy to infer that $(\gamma_B^{i,i-1})^2 \geq (1/2)\left(\sum_j |A_j \setminus B|\right)^2 = (1/2)|\cup_j A_j \setminus B'|^2$.

Let $(A_j')_j$ denote the clusters of $\widehat{\mathrm{OPT}_{i-1}}$ that are not half-contained in B, but have non-empty intersections with B. We now show that

$$1/2(|B' \setminus \cup_j A_j'|)^2 \leq (\gamma_B^{i,i-1})^2.$$

We have $B' \setminus \cup_j A_j = \cup_j (A_j' \cap B)$. Observe that any A_j' is a large cluster of $\widehat{\mathrm{OPT}_{i-1}}$, thus a cluster of OPT_{i-1}'. Then $(\gamma_B^{i,i-1})^2$ includes all edges uv such that one of the following two cases occurs

First, if $u \in A'_j \setminus B$ and $v \in A'_j \cap B$: such edges are internal in the clustering OPT'_{i-1} but external in the clustering OPT'_i. The number of edges of this type is $\sum_j |A'_j \setminus B| \cdot |A'_j \cap B|$. Since A'_j is not half-contained in B, this is at least $\sum_j |A'_j \cap B|^2$.

Second, if $u \in A'_j \cap B$ and $v \in A'_k \cap B$ with $j \neq k$: such edges are external in the clustering OPT'_{i-1} but internal in the clustering OPT'_i. The number of edges of this type is $\sum_{j<k} |A'_j \cap B| \cdot |A'_k \cap B|$.

Summing, we get

$$(\gamma_B^{i,i-1})^2 \geq (1/2)\left(\sum_j |A'_j \cap B|\right)^2 = (1/2)|B' \setminus \cup_j A'_j|^2.$$

■

Lemma 2.17. *For any $i \geq 1$, $\widehat{OPT_i}$ has at most $1/\epsilon^2$ non singleton clusters, all of which are clusters of OPT'_i*

Proof. By definition, $\widehat{\mathrm{OPT}_1}$ has at most $1/\epsilon^2$ non singleton clusters. For any $i > 1$, a cluster of $\widehat{\mathrm{OPT}_{i-1}}$ can only be half-contained in one cluster of OPT'_i. Therefore given $\widehat{\mathrm{OPT}_{i-1}}$, at most $1/\epsilon^2$ clusters of OPT'_i are marked. Thus $\widehat{\mathrm{OPT}_i}$ has at most $1/\epsilon^2$ clusters. ■

We can now prove Lemma 2.8.

Proof of Lemma 2.8. By Lemma 2.14, clusterings $\mathcal{F}_i$ and $\mathcal{F}_{i-1}$ only differ in their partition of $[1, t_{i-1}]$. Then the set of the vertices that are classified differently in $\mathcal{F}_i$ and $\mathcal{F}_{i-1}$ is $\cup_i C_i \setminus C_{i-1}$. Each of these vertices creates at most t_K disagreements:

$$\mathrm{cost}(\mathcal{F}_{i-1}) - \mathrm{cost}(\mathcal{F}_i) \leq \sum_{C_i \in \mathcal{F}_i} |C_i \setminus C_{i-1}| t_K \tag{2.3}$$

By Lemmas 2.15 and 2.16,

$$\sum_{C_i \in \mathcal{F}_i} |C_i \setminus C_{i-1}| t_K \leq \left(2\sqrt{2}\left(\sum_{\text{large } B \in \widehat{\mathrm{OPT}_i}} \gamma_B^{i,i-1}\right) + |\mathcal{S}^i| + |\mathcal{T}^i|\right) t_K. \tag{2.4}$$

By Lemmas 2.11 and 2.13,

$$|\mathcal{S}^i| \leq \frac{\epsilon}{1-\epsilon} t_{i-1} \text{ and } |\mathcal{T}^i| \leq \sqrt{6} \sum_{\text{large } B \in \widehat{\mathrm{OPT}_{i-1}}} \gamma_B^{i-1,i} \tag{2.5}$$

The term $\sum_{\text{large } B \in \widehat{\mathrm{OPT}_{i-1}}} \gamma_B^{i-1,i}$ can be seen as the ℓ_1 norm of the vector $(\gamma_B^{i-1,i})_{\text{large } B}$. Since $\widehat{\mathrm{OPT}_{i-1}}$ has at most $1/\epsilon^2$ large clusters by Lemma 2.17, we can use Hölder's inequality:

$$\sum_{\text{large } B \in \widehat{\mathrm{OPT}_{i-1}}} \gamma_B^{i-1,i} = \|(\gamma_B^{i-1,i})_{\text{large } B}\|_1 \leq 1/\epsilon \|(\gamma_B^{i-1,i})_{\text{large } B}\|_2.$$

By definition we have $\|(\gamma_B^{i-1,i})_{\text{large B}}\|_2 \leq \sqrt{2(\mathrm{cost}(\mathrm{OPT}'_{i-1}) + \mathrm{cost}(\mathrm{OPT}'_i))}$. Thus

$$\sum_{\text{large } B \in \widehat{\mathrm{OPT}_{i-1}}} \gamma_B^{i-1,i} \leq 1/\epsilon \sqrt{2(\alpha t_{i-1}^2/2 + \alpha t_i^2/2)} \leq \frac{\sqrt{2\alpha}}{\epsilon} t_i. \tag{2.6}$$

Similarly, we have

$$\sum_{\text{large } B \in \widehat{\text{OPT}_i}} \gamma_B^{i,i-1} \leq \frac{\sqrt{2\alpha}}{\epsilon} t_i. \tag{2.7}$$

Combining equations (2.3) through (2.7) and $\alpha = \epsilon^4$ yields

$$\text{cost}(\mathcal{F}_{i-1}) - \text{cost}(\mathcal{F}_i) \leq \left((4 + 2\sqrt{3})\epsilon + \frac{\epsilon}{1-\epsilon} \right) t_i t_K$$

■

3. Minimizing Disagreements Online

Theorem 3.1. *Algorithm* Greedy *is* $(2n+1)$*-competitive for* MinDisAgree.

To prove Theorem 3.1, we need to compare the cost of the optimal clustering to the cost of the clustering constructed by the algorithm. The following lemma reduces this to, roughly, analyzing the number of vertices classified differently.

Lemma 3.2. *Let* $\mathcal{W}$ *and* $\mathcal{W}'$ *be two clusterings such that there is an injection* $W_i' \in \mathcal{W}' \to W_i \in \mathcal{W}$*. Then cost*$(\mathcal{W}')$ *−* $cost(\mathcal{W}) \leq n \sum_i |W_i' \setminus W_i|$.

For subsets of vertices $S_1, \ldots, S_m$, we will write, with a slight abuse of notation, $\Gamma^+(S_1, \ldots, S_m)$ for the set of edges in $\Gamma^+(S_i, S_j)$ for any $i \neq j$: $\Gamma^+(S_1, \ldots, S_m) = \cup_{i \neq j} \Gamma^+(S_i, S_j)$.

Lemma 3.3. *Let* C *be a cluster created by* Greedy*, and* $\mathcal{W} = \{W_1, \ldots, W_K\}$ *denote the clusters of OPT. Then* $|C| \leq \max_i |C \cap W_i| + 2|\Gamma^+(C \cap W_1, \ldots, C \cap W_K)|$.*. We call* $i_0 = \arg\max_i |C \cap W_i|$ *the* leader *of* C*.*

Proof of Theorem 3.1. Let $\mathcal{C}$ denote the clustering given by Greedy. For every cluster W_i of OPT, merge all the clusters of $\mathcal{C}$ that have i as their leaders. Let $\mathcal{C}' = (W_i')$ be this new clustering. By definition of the greedy algorithm, this operation can only increase the cost since every pair of clusters have a negative-majority cut at the end of the algorithm:$\text{cost}(\mathcal{C}) \leq \text{cost}(\mathcal{C}')$. We apply Lemma 3.2 to $\mathcal{W}$ =OPT and $\mathcal{W}' = \mathcal{C}'$, and obtain: $\text{cost}(\mathcal{C}') \leq \text{cost}(\text{OPT}) + n \sum_i |W_i' \setminus W_i|$. By definition of $\mathcal{C}'$ we have $|W_i' \setminus W_i| = \sum_{C \in \mathcal{C}: \text{leader}(C) = i} \sum_{j \neq i} |C \cap W_j|$, hence

$$\sum_i |W_i' \setminus W_i| = \sum_{C \in \mathcal{C}} \sum_{j \neq \text{leader}(C)} |C \cap W_j|.$$

By Lemma 3.3, $\sum_{j \neq \text{leader}(C)} |C \cap W_j| \leq 2|\Gamma^+(C \cap W_1, \ldots, C \cap W_K)|$. Finally, to bound OPT from below, we observe that, for any two clusterings $\mathcal{C}$ and $\mathcal{W}$, it holds that the sum over $C \in \mathcal{C}$ of $|\Gamma^+(C \cap W_1, \ldots, C \cap W_K)|$ is less than $\text{cost}(\mathcal{W})$. Combining these inequalities yields the theorem. ■

Theorem 3.4. *Let ALG be a randomized algorithm for* MinDisAgree. *Then there exists an instance on which ALG has cost at least* $n - 1 - cost(OPT)$ *where OPT is the offline optimum. If OPT is constant then* $cost(ALG) = \Omega(n)cost(OPT)$.

Proof. Consider two cliques A and B, each of size m, where all the internal edges of A and B are positive. Choose a vertex a in A, and a set of vertices $b_1, \ldots, b_k$ in B. Define the edge labels of ab_i as positive, for all $1 \leq i \leq k$ and the rest of the edges between A and B as negative. Define an input sequence starting with $a, b_1, \ldots, b_k$, followed by the rest of the vertices in any order.

■

References

[1] Nir Ailon, Moses Charikar, and Alantha Newman. Aggregating inconsistent information: ranking and clustering. In *STOC '05: Proceedings of the thirty-seventh annual ACM symposium on Theory of computing*, pages 684–693, New York, NY, USA, 2005. ACM Press.

[2] Nikhil Bansal, Avrim Blum, and Shuchi Chawla. Correlation clustering. *Mach. Learn.*, 56(1-3):89–113, 2004.

[3] Amir Ben-Dor, Ron Shamir, and Zohar Yakhini. Clustering gene expression patterns. *Journal of Computational Biology*, 6(3-4):281–297, 1999.

[4] Allan Borodin and Ran El-Yaniv. *Online computation and competitive analysis*. Cambridge University Press, New York, NY, USA, 1998.

[5] Moses Charikar, Chandra Chekuri, Tomas Feder, and Rajeev Motwani. Incremental clustering and dynamic information retrieval. *SIAM J. Comput.*, 33(6):1417–1440, 2004.

[6] Moses Charikar, Venkatesan Guruswami, and Anthony Wirth. Clustering with qualitative information. In *focs*, volume 00, page 524, Los Alamitos, CA, USA, 2003. IEEE Computer Society.

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

[8] William W. Cohen and Jacob Richman. Learning to match and cluster large high-dimensional data sets for data integration. In *KDD '02: Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 475–480, New York, NY, USA, 2002. ACM.

[9] Erik D. Demaine, Dotan Emanuel, Amos Fiat, and Nicole Immorlica. Correlation clustering in general weighted graphs. *Theor. Comput. Sci.*, 361(2):172–187, 2006.

[10] Jenny Rose Finkel and Christopher D. Manning. Enforcing transitivity in coreference resolution. In *Proceedings of ACL-08: HLT, Short Papers*, pages 45–48, Columbus, Ohio, June 2008. Association for Computational Linguistics.

[11] Ioannis Giotis and Venkatesan Guruswami. Correlation clustering with a fixed number of clusters. *Theory of Computing*, 2(1):249–266, 2006.

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

[13] Marek Karpinski and Warren Schudy. Linear time approximation schemes for the Gale-Berlekamp game and related minimization problems. In *STOC '09: Proceedings of the 41st annual ACM symposium on Theory of computing*, pages 313–322, 2009.

[14] Claire Mathieu and Warren Schudy. Correlation clustering with noisy input. In *To appear in Procs. 21st SODA, preprint: http://www.cs.brown.edu/∼ws/papers/cluster.pdf*, 2010.

[15] Ron Shamir, Roded Sharan, and Dekel Tsur. Cluster graph modification problems. *Discrete Appl. Math.*, 144(1-2):173–182, 2004.