Correlation Clustering Revisited: The "True" Cost of Error Minimization Problems

Nir Ailon* Edo Liberty†

Abstract

Correlation Clustering was defined by Bansal, Blum, and Chawla as the problem of clustering a set of elements based on a possibly inconsistent binary similarity function between element pairs. Their setting is agnostic in the sense that a ground truth clustering is not assumed to exist, and the cost of a solution is computed against the input similarity function. This problem has been studied in theory and in practice and has been subsequently proven to be APX-Hard.

In this work we assume that there does exist an unknown correct clustering of the data. This is the case in applications such as record linkage in databases. In this setting, we argue that it is more reasonable to measure accuracy of the output clustering against the unknown underlying true clustering. This corresponds to the intuition that in real life an action is penalized or rewarded based on actual outcome. The traditional combinatorial optimization version of the problem only offers an indirect solution to our revisited version via a triangle inequality argument applied to the distances between the output clustering, the input similarity function and the underlying ground truth. In the revisited version, we show that it is possible to shortcut the traditional optimization detour and obtain a factor 2 approximation. This factor could not have possibly been obtained by using a solution to the traditional problem as a black box, unless it was an exact optimal solution. Our result therefore shortcuts the APX-Hardness, and could be useful for revisiting other combinatorial optimization problems.

Our solution involves a novel way to continuously morph a general (non-metric) distance function into a metric. This technique is interesting in its own right and may be useful for other metric embedding problems. The resulting morphed solution is randomly rounded into a clustering. En route, in certain cases we obtain a certificate for the possibility of getting a solution of factor strictly less than 2. Finally, we show simple cases in which randomness is necessary for achieving a solution of factor strictly less than 2, thus justifying the use of randomization in our solution.

*Google Research
†Yale University and Google Research

1 Introduction

Correlation Clustering was defined by Bansal, Blum and Chawla [1] as the problem of agnostically learning how to cluster data based on a binary similarity function. The similarity function tells us, for each pair, whether they are similar or not. The reason they called the problem *agnostic* is because there is no real notion of an underlying true clustering of the data, and the algorithm tries to minimize the symmetric difference with respect to the given similarity function. This gives rise to an APX-hard optimization problem which is studied in their paper and in consequence work [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]. Applications include microarray and database clustering. In this paper we assume a setting in which *there is* an (unknown) correct way to cluster the data. Such a scenario is realistic and arises in duplicate detection and elimination in large data (also known as the record linkage), bioinformatics (there is an underlying assumption that proteins are actually grouped by function) and image segmentation.

The question is, how does there being an underlying true clustering help us if we do not know it? Our main result is a clustering algorithm taking the similarity function as input and outputting a clustering such that if the similarity function "respects" the true clustering, then so does the output. In other words, we care about the quality of the output w.r.t. the truth only insofar as we expect the similarity function to be "good".

We introduce some notation. Assume we are interested in an object τ, which is an unknown ground truth. We have access to an observation h. It is good to think of h as a noisy version of τ (see example below), although we assume here that h is adversarial and could possibly have no connection whatsoever with τ. Assume τ belongs to some restricted space P (for *property*) of consistent objects (in our example, clusterings). The observation h may violate the property and lie in a larger space $U \supseteq P$. This violation may be due to noise or due to computational difficulties in computing over P. We measure distance, or disagreement with respect to the truth τ using a cost function $f : P \times U \to \mathbb{R}^+$. Unlike the traditional setting considered in combinatorial optimization, property testing and reconstruction problems, where the target is to minimize $f(x, h)$, here the output x is measured against τ, i.e the target is to minimize $f(\tau, x)$. In both cases we divide by $f(\tau, h)$ to obtain a unit free approximation ratio. *Note that in the problem definition there is nothing optimal about the true clustering τ - it is any (adversarial) object.* The goal is to find a solution $x \in P$ such that $f(\tau, x) \leq Cf(\tau, h)$ for a small C (we make a precise defition in what follows). In the randomized setting we allow x to be drawn from a distribution, in which case we will want $E[f(\tau, x)] \leq Cf(\tau, h)$. We will see later that randomness can strictly reduce C. This fact is not a complexity theoretical result, but a claim related to the structure of the sets P, U and the geometry endowed by the distance function f (also see comment about Bregman divergence in Section 7).

In this work the object of interest is a clustering of a given dataset V, and the observation h is a pairwise similarity function (as in traditional Correlation Clustering) which has $h(u, v) = 0$ if u, v are deemed similar and 1 otherwise. A clustering is a $\{0, 1\}$ valued pairwise function that is a pseudometric. The ground truth τ is a consistent clustering, in other words, a $\{0, 1\}$ valued pseudometric on elements of V. The cost f is the Hamming distance on $\binom{n}{2}$ dimensional vectors. We call this corresponding problem of finding x such that $f(\tau, x) \leq Cf(\tau, h)$ (for small C) CorrelationClusteringX. We will definte it precisely lateer.

A Realistic Example. A search engine obtains information about businesses (e.g. restaurants, notary publics, hair salons) in a geographical location from multiple sources (Yellow Pages, web crawling etc.). Typically each business is represented many times, so there is need to identify the multiple information records for each business as one cluster. When a user sends a query for a pizzeria in a neighborhood, the search engine will return a list of possible (representatives from) clusters corresponding to matching businesses. If due to a clustering error a single pizzeria appears as two separate search results, the user will be upset. If due to yet a different type of clustering error, two competing pizzerias are hybridized in one search result, then the user will also be upset. This cost is directly related to the *true* unknown clustering.

Machine Learning Reductions ([11, 12, 13]). In order to solve the last realistic example, a practitioner may approach the question of whether two business records represent the same business or not as a classic machine learning binary classification problem. Thus the practitioner treats the clustering as a learning problem and divides it into small (binary) learning tasks, ignoring dependencies. Choosing a good binary

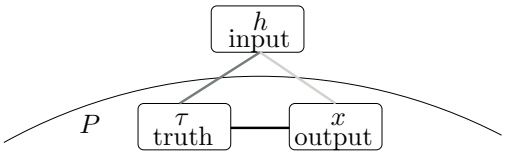

Figure 1: Given an input h, a traditional optimization approach tries to find x minimizing the maximal ratio between the length of the light gray line and the dark gray line over all τ . Our approach is to find x minimizing the maximal ratio between the black line and the dark gray line over all τ.

classifier will ensure that the classification error of the output hypothesis h is small, where the classification error is the number of pairs of businesses u, v for which $h(u, v)$ errs with respect to the ground truth.[1] The out-of-the-box binary classifier, however, does typically not "know" about transitivity constraints. The practitioner will then convert the possibly nontransitive hypothesis h to a transitive one (a clustering). The cost of interest is still against the ground truth - the search engine users do not experience or intereact with h in any way.

Connection to Random Noise Clustering Problems. A related problem that has been studied in the literature [14] is *planted clustering*. In this model, the observation h is given by random noise applied to the ground truth clustering τ. Returning to our above example, the observation h is obtained by flipping an independent coin with a fixed bias for each pair of businesses u, v, and either setting $h(u, v) = \tau(u, v)$ or $h(u, v) = 1 - \tau(u, v)$ based on the coin outcome. Traditional Correlation Clustering on input h thus obtained is precisely a *maximum likelihood* recovery from h. It is not clear, however, why this random noise model should be at all realistic. If for instance h is obtained as an output of a machine learned hypothesis (as stated above) then it is very reasonable to assume that the error will be highly structured and correlated. Also, it is often the case that h is obtained as a robust version (using e.g. spectral techniques [14] or dot product techniques [15][2]) of some raw input. In these cases, it is clear that any independence assumption that we may have had on the raw input would be lost in the process of obtaining h. Our approach is *adversarial*, and the practitioner may use it given h that is obtained using any preprocessing, even if heavy dependencies are introduced. The advantage of our work is that the practitioner need not worry about transitivity issues when preprocessing the data, and that unlike other techniques for obtaining a final (transitive) clustering (e.g. k-means over h obtained as a low dimensional Euclidean approximation of raw data using spectral techniques), we provide provable guarantees.

Traditional optimization gives the following indirect solution to our problem: If f can be extended to $U \times U$ as a metric, then find $x \in P$ approximately minimizing $f(h, x)$ so $f(h, x) \leq C^* f(h, \tau)$ for some $C^* \geq 1$ and for all $\tau \in P$. By the triangle inequality $f(\tau, x) \leq f(\tau, h) + f(h, x) \leq (C^* + 1) f(\tau, h)$. Hence an approximation factor of C^* for the traditional corresponding combinatorial optimization problem gives an upper bound of $C = C^* + 1$ for our problem, which is at least 2 even if we could solve the optimization problem optimally. A similar argument can be made for randomized combinatorial optimization and an expected approximation ratio. This immediately raises the interesting question of whether we can go below 2 and shortcut the traditional optimization detour (often an obstruction under complexity theoretical assumptions).

Our main result is a morphing algorithm which finds a relaxed solution (in a sense that will be explained precisely) to CorrelationClusteringX. More precisely, our relaxed solution is a limit at infinity of a solution to a piecewise linear differential equation. This algorithm is interesting in its own right and may be useful for other problems on metric spaces. The intuitive idea behind the differential equation is a physical system in which edges "exert forces" on each other proportional to the size of triangle inequality violations. The main technical lemma (differed to the appendix) shows that different triangles interfere *constructively* with each other. We

[1] In machine learning, comparing against an unknown *ground truth* is the norm. It is usually assumed there though that the ground truth is drawn from a distribution. In this work the setting is adversarial. The fact that we can argue against an adversary is because the concept space of clusterings is highly structured.

[2] In this work, a Gaussian random noise model is assumed.

then show how to randomly convert the relaxed solution to a final output to CorrelationClusteringX with $C = 2$. Assuming P$\neq$ NP this is better than what we could have achieved efficiently using a black box analysis of the traditional optimization problem (because that problem is APX-Hard [5]). As a side effect, our algorithm allows computing an invariant $C' = C'(h) \leq 4/3$ which serves as a witness for getting a solution to CorrelationClusteringX with $C = 3C'/2$. In particular, if $C' < 4/3$ then we get $C < 2$.

Our work is related to recent work by Ailon and Mehryar [13] on Machine Learning reductions for ranking. Balcan et al. [16] also consider clustering problems in which a ground truth is assumed to exist. However, there are two main differences. First, they consider objective functions in which the cost is computed pointwise (here we consider pairwise costs). A second and more fundamental difference is that they make strong assumptions about the (input obsevation, ground truth) pair. Their assumptions, in some sense, exactly state that an approximation to the traditional optimization problem is "good" for the problem in which errors are computed against the truth. They show that under this assumption, NP-Hardness shortcutting is also possible. In our case, we make no assumptions about the input or the ground truth. Further investigation of the connection between the two results is an interesting research direction.

In Section 3 we present our morphing technique. In Section 4 we show how to randomly convert the solution to a final clustering solution, incurring another multiplicative cost of 3/2, resulting in a total of at most $4/3 \times 3/2 = 2$. The algorithm is similar to techniques used in Ailon et al's [2] work on standard Correlation Clustering, but (a) the analysis is different because a ground truth object is used, and (b) an extra "tweaking" step is needed. In Section 5 we discuss runtime issues. In Section 6 we show how to get strictly better solutions to CorrelationClusteringX using randomness (compared to deterministic solutions) for one specific input. Finally in Section 7 we discuss connection to other work and propose future directions. In particular, we note an interesting connection between our work and the well known theory of Bregman divergence in statistics and online optimization.

2 Definitions and Statement of Results

We are given a set V of n elements to cluster together with a symmetric distance function h serving as clustering information. We use the convention that $h(u, v) = h(v, u) = 1$ if u, v are believed to belong to separate clusters, and 0 otherwise.[3]

Let $\mathcal{K}$ denote the set of $[0, 1]$-valued symmetric functions on $V \times V$ (with a null diagonal). Let $\mathcal{I} \subseteq \mathcal{K}$ denote the subset of $\{0, 1\}$ valued functions in $\mathcal{K}$. Let $\Delta \subseteq \mathcal{K}$ denote the set of functions $k \in \mathcal{K}$ satisfying the triangle inequality $k(u, v) \leq k(v, w) + k(w, u)$ for all $u, v, w \in V$. Let $\mathcal{C}$ denote $\mathcal{I} \cap \Delta$. Clearly $c \in \mathcal{C}$ is an encoding of a clustering of V, with $c(u, v) = 1$ if u, v are separated and $c(u, v) = 0$ if they are co-clustered. [4]

Our input h lives in $\mathcal{I}$ but not in Δ, hence the function h encodes possibly inconsistent $\{0, 1\}$ clustering information. Indeed, it may tell us that $h(u, v) = h(v, w) = 0$ but $h(u, w) = 1$, hence violating transitivity. For a number $a \in [0, 1]$ let $\overline{a}$ denote $1-a$. Define the Correlation Clustering cost function [1] $f : \mathcal{K} \times \mathcal{K} \to \mathbb{R}^+$ as $f(k_1, k_2) = \sum_{u<v} (k_1(u, v)\overline{k_2(u, v)} + \overline{k_1(u, v)}k_2(u, v))$. For integer valued k_1, k_2 this is the Hamming distance.

Definition 2.1. *The problem of CorrelationClusteringX is defined as follows: Given $h \in \mathcal{I}$ and $C \geq 1$ output $x \in \mathcal{C}$ such that for all $\tau \in \mathcal{C}$, $f(\tau, x) \leq Cf(\tau, h)$ (assuming such an x exists). In the randomized setting, the goal is to output a sample x from a distribution $\mathcal{D}$ on $\mathcal{C}$, such that $E_{x\sim\mathcal{D}}[f(\tau, x)] \leq Cf(\tau, h)$ (assuming such $\mathcal{D}$ exists). An algorithm outputting x in the deterministic case or drawing it from $\mathcal{D}$ in the randomized case is called a C-approximation algorithm to CorrelationClusteringX.*

Deterministic CorrelationClusteringX has a corresponding integer program over the $\binom{n}{2}$ variables of $x \in \mathcal{C}$

[3] In other literature, h is a similarity measures, with higher values corresponding to higher belief in co-clustering. We find our convention easier to work with because a clustering is equivalently a metric over the values {0,1}.

[4] In what follows, $\binom{V}{b}$ denotes the collection of unordered b-tuples of the set V. When it is clear from the context, the notation (u, v) means an unordered tuple $\{u, v\} \in \binom{V}{2}$ and similarly (u, v, w) means an unordered tuple $\{u, v, w\} \in \binom{V}{3}$.

with an exponential number of constraints:

$$\text{IP: minimize } C \text{ s.t.} \qquad \begin{array}{l} f(x,\tau) \leq Cf(h,\tau) \text{ for all } \tau \in \mathcal{C} \\ x \in \mathcal{C}, C \geq 0 \end{array}$$

Note that in traditional correlation clustering, we would have used the constraint $f(x,h) \leq Cf(h,\tau)$ for all $\tau \in \mathcal{C}$ instead. IP can be relaxed by allowing $x \in \Delta$ and adding a constraint for each $\tau \in \Delta$.

$$\text{LP: minimize } C \text{ s.t.} \qquad \begin{array}{l} f(\tau,x) \leq Cf(\tau,h) \text{ for all } \tau \in \Delta \\ x \in \Delta, C \geq 1 \ . \end{array}$$

Clearly, an equivalent program can be obtained by using only constraints that correspond to vertices of Δ, of which there are exponentially many. Let (x_{LP}, C_{LP}) denote the minimizer of LP.

Observation 2.1. *LP has a separation oracle and can therefore be solved optimally in polynomial time.*

To see Observation 2.1, note that given a candidate solution (x, C) it is possible to find $\tau \in \Delta$ satisfying $f(\tau,x) > Cf(\tau,h)$ (if one exists) using another simple standard linear program with $\tau \in \Delta$ as variable. Note that unlike in the usual case of combinatorial optimization LP relaxations, it is not immediate to compare between the values of IP and LP, because the relaxation is obtained by both adding constraints and removing others. The reason we enlarged the collection of constraints $\{f(\tau,x) \leq Cf(\tau,h)\}_\tau$ in LP is to give rise to an efficient separation oracle.

Our first result states that the optimal solution to LP is a (deterministic) fractional solution x_{LP} for CorrelationClusteringX with approximation factor C_{LP} of at most 4/3 (in the sense that $f(x_{LP},\tau) \leq C_{LP} f(h,\tau)$ for all $\tau \in \Delta$). The proof of the theorem is constructive. It is shown that the limit at infinity of a solution to a certain differential equation is a feasible solution to LP.

Theorem 2.1. *For any $h \in I$, the value of LP is at most 4/3.*

In the proof of Theorem 2.1 we will point to one particular solution (x_{dif}, C_{dif}) which is a limit at infinity of a solution to a piecewise linear differential equation. Finding this limit may be done exactly, but we omit the details because together with Observation 2.1, general purpose convex optimization may be used instead.

Our next theorems refer to the QuickCluster algorithm which is defined in Section 4. QuickCluster takes as input $h \in \mathcal{K}$ and outputs $x \in \mathcal{C}$. Let $QC(h)$ denote the distribution over outputs $x \in \mathcal{C}$ of QuickCluster for input h.

Theorem 2.2. *For any $\hat{h} \in \Delta$ and $\tau \in \mathcal{C}$ we have $E_{x \sim QC(\hat{h})}[f(x,\tau)] \leq \frac{3}{2} f(\hat{h},\tau)$.*

Combining Theorems 2.1 and Theorem 2.2 we get a randomized solution with $C = \frac{3}{2} C_{LP} \leq 2$ for CorrelationClusteringX. If $C_{LP} < 4/3$, we get a witness for achieving C strictly less than 2.

Theorem 2.3. *For any $h \in \mathcal{I}$ and $\tau \in \mathcal{C}$ we have $E_{x \sim QC(h)}[f(\tau,x)] \leq 2f(\tau,h)$.*

The running time of QuickCluster is analyzed for two representation dependent regimes. In the first, only pairwise queries to h are allowed, i.e, evaluating $h(u,v)$ for a pair $\{u,v\}$.

Theorem 2.4. *In the pairwise queries model, any constant factor randomized approximation algorithm for CorrelationClusteringX performs $\Omega(n^2)$ queries to h on expectation for some input h.*

In the second regime, the algorithm is allowed neighborhood-queries, returning for u its neighborhood $N(u) = \{u\} \cup \{v \in V \mid h(u,v) = 0\}$ as a linked list. We obtain the following bound on the running time of QuickCluster, depending on the distance of h to being a clustering.

Theorem 2.5. *In the neighborhood-queries model, the expected running time of QuickCluster is* $O(n + \min_{\tau \in \mathcal{C}} f(\tau, h))$.

The following is a lower bound on what a deterministic algorithm can do. For this hard case there is a strict gap between the randomized and deterministic cases.

Theorem 2.6. *There exists an input* h *for which any deterministic algorithm for CorrelationClusteringX incurs an approximation factor of at least* 2 *for some ground truth* $\tau \in \mathcal{C}$. *For the same input, a randomized algorithm can obtain a factor of at most* $4/3$.

3 Morphing h Into a metric: A Differential Program

In this section we prove Theorem 2.1. The idea is to "morph" $h \in \mathcal{K}$, which is not necessarily a metric, into a (pseudo)metric. The solution $x_{dif} \in \Delta$ is obtained by theoretically running a differential equation to infinity. More precisely, we define a differential morphing process such that $h_t(u, v)$ is the changed value of the $h(u, v)$ at time t and $h_0(u, v) = h(u, v)$ for all u and v. The solution is given by $x_{dif} = \lim_{t \to \infty} h_t$.

We look at a triangle created by the triplet $\{u, v, w\}$. For ease of notation we set $a = h(u, v)$, $b = h(v, w)$, and $c = h(w, u)$. First, we define the gap g_{uvw} of the triangle $\{u, v, w\}$ away from satisfying the triangle inequality as:

$$g_{uvw} = \max\{0,\ a - (b + c),\ b - (c + a),\ c - (a + b)\} \tag{1}$$

We define the *force* that triangle $\{u, v, w\}$ exerts on a as follows:

$$F(a; b, c) = \begin{cases} -g_{uvw} & \text{if } a > b + c \\ g_{uvw} & \text{otherwise.} \end{cases} \tag{2}$$

The morphing process is such that the contribution of the triangle $\{u, v, w\}$ to the change in a, $\frac{da}{dt}$, is the force $F(a; b, c)$. Intuitively, the force serves to reduce the gap. If a, b, and c satisfy the triangle inequality then no force is applied. If $a > b + c$ then $\frac{da}{dt}$ is negative and a is reduced. If $b > c + a$ or $c > a + b$ then $\frac{da}{dt}$ is positive a is increased. Averaging over all triangles containing u and v gives our differential equation in Figure 2. For ease of notation, we let $a(t)$, $b(t)$ and $c(t)$ denote $h_t(u, v), h_t(v, w)$ and $h_t(w, u)$ throughout.

$$\frac{dh_t(u,v)}{dt} = \frac{1}{n-2} \sum_{w \in V \setminus \{u,v\}} F(h_t(u, v); h_t(v, w), h_t(w, u)); \quad h_0(u, v) = h(u, v) \ \ \forall u, v \in V$$

Figure 2: The morphed input h_t is given by the solution to the above differential equation at time t. The initial starting point is the input $h_0 = h$. The solution x_{dif} is given by $x_{dif} = \lim_{t \to \infty} h_t$

The following is the main technical lemma of the proof. It asserts that the external forces applied to a triangle $\{u, v, w\}$ by other triangles only contribute to reducing the gap g_{uvw}. It implies both the exponential decay of all positive gaps and the stability of null gap.

Lemma 3.1. *Let* $g_{uvw}(t)$ *denote the gap of* h_t *on the triplet* $\{u, v, w\}$ *at time* t, *as defined in (1). Then* $\frac{dg_{uvw}(t)}{dt} \leq -3g_{uvw}(t)$ *for all* t.

Note: Clearly the lemma implies that $g_{uvw}(t) \leq g_{uvw}(t_0)e^{-3(t-t_0)}$ for any $t_0 \leq t$. The lemma is easy to prove if $|V| = 3$. For larger V, the difficulty is in showing that the interference between triangles is constructive.

Proof. It is enough to prove the lemma for the case $\{a(t) \geq b(t) + c(t)\} \cup \{b(t) \geq c(t) + a(t)\} \cup \{c(t) \geq a(t) + b(t)\}$. Indeed, in the open set $\{a(t) < b(t) + c(t)\} \cap \{b(t) < c(t) + a(t)\} \cap \{c(t) < a(t) + b(t)\}$ the value

of g is 0 identically. Assume w.l.o.g. therefore that $a(t) \geq b(t) + c(t)$ (hence $g_{uvw}(t) = a(t) - b(t) - c(t)$).

$$\frac{d\ g_{uvw}(t)}{dt} = \frac{d\ (a(t) - b(t) - c(t))}{dt} = \frac{1}{n-2}\Big[\ (F(a(t); b(t), c(t)) - F(b(t); c(t), a(t)) - F(c(t); a(t), b(t))) + \sum_{s \in V \setminus \{u,v,w\}} (F(a(t); x_s(t), y_s(t)) - F(b(t); z_s(t), y_s(t)) - F(c(t); x_s(t), z_s(t)))\Big]\ ,$$

where $x_s(t) = h_t(u, s)$, $y_s(t) = h_t(v, s)$, and $z_s(t) = h_t(w, s)$ as depicted in Figure 3. The first term gives exactly $F(a(t); b(t), c(t)) - F(b(t); c(t), a(t)) - F(c(t); a(t), b(t)) = -3g_{uvw}$. It suffices to prove that for any $s \in V \setminus \{u, v, w\}$, $F(a(t); x_s(t), y_s(t)) - F(b(t); z_s(t), y_s(t)) - F(c(t); x_s(t), z_s(t)) \leq 0$. This is proved by enumerating over all possible configurations of the three triangles $\{u, v, s\}$, $\{v, w, s\}$ and $\{w, u, s\}$ and is deferred to the appendix (Lemma A.1). □

The following lemma tells us that if $a(0)$, $b(0)$, and $c(0)$ violate the triangle inequality then at each moment $t > 0$ they either violate the same inequality or the violation disappears.

Lemma 3.2. *Let $a(t)$, $b(t)$, and $c(t)$ denote $h_t(u, v)$, $h_t(v, w)$, and $h_t(w, u)$ respectively. If $a(0) \geq b(0)+c(0)$ then for all $t \geq 0$ either $a(t) \geq b(t) + c(t)$ or $a(t)$, $b(t)$, and $c(t)$ satisfy the triangle inequality.*

Proof. First note that if for some time t_0 the triplet $\{a(t_0), b(t_0), c(t_0)\}$ satisfies the triangle inequality, then this will continue to hold for all $t \geq t_0$ in virtue of the note following Lemma 3.1. Also note that $a(t) > b(t) + c(t)$ and ($b(t) > c(t) + a(t)$ or $c(t) > a(t) + b(t)$) cannot hold simultaneously. Let t' be the infimum of t such that $a(t) \leq b(t) + c(t)$, or ∞ if no such t exists. If $t' = \infty$ then the lemma is proved. Otherwise by continuity and the first note above, $a(t') = b(t') + c(t')$, $b(t') \leq a(t') + c(t')$ and $c(t') \leq a(t') + b(t')$, hence $a(t')$, $b(t')$, and $c(t')$ satisfy the triangle inequality and thus continue to do so for all $t > t'$, completing the proof of the lemma.

□

Now fix a ground truth clustering $\tau \in \Delta$. Consider the cost $f(\tau, h_t)$ as a function of t. Letting $L_t(u, v) = h_t(u, v)\overline{\tau(u, v)} + \overline{h_t(u, v)}\tau(u, v)$, we get $f(\tau, h_t) = \sum_{u<v} L_t(u, v) = \frac{1}{n-2}\sum_{u<v<w} C_{uvw}(t)$, where $C_{uvw}(t) := L_t(u, v) + L_t(v, w) + L_t(w, u)$. The derivative of the cost is $\frac{d\ f(\tau,h_t)}{dt} = \frac{1}{n-2}\sum_{uvw} G_{uvw}(t)$, where

$$G_{uvw}(t) := \Big[\ (1 - 2\tau(u, v))F(h_t(u, v); h_t(v, w), h_t(w, u)) + (1 - 2\tau(v, w))F(h_t(v, w); h_t(w, u), h_t(u, v)) + (1 - 2\tau(w, u))F(h_t(w, u); h_t(u, v), h_t(v, w))\Big]\ .$$

(Note that G_{uvw} is *not* the derivative of C_{uvw}, but the sum $\sum_{uvw} G_{uvw}$ is the derivative of $\sum_{uvw} C_{uvw}$.) The cost at time t is hence $f(\tau, h_t) = \frac{1}{n-2}\sum_{uvw} C_{uvw}(0) + \frac{1}{n-2}\sum_{uvw}\int_0^t G_{uvw}(s)ds$. We concentrate on the contribution of one triangle to this sum: $H_{uvw}(t) = C_{uvw}(0) + \int_0^t G_{uvw}(s)ds$.

Let us consider the possible values of the term $G_{uvw}(t)$. If the values $h_t(u, v)$, $h_t(v, w)$, and $h_t(w, u)$ satisfy the triangle inequality then $G_{uvw}(t) = 0$ since the forces F are all zero. Assume then w.l.o.g. that $h_t(u, v) \geq h_t(v, w)+h_t(w, u)$ and so by the definition of F, $G_{uvw}(t) = [2(\tau(u, v)-\tau(v, w)-\tau^*(w, u))+1]g_{uvw}(t)$. Notice that $G_{uvw}(t) \leq g_{uvw}(t)$ since $\tau \in \Delta$. Therefore $G_{uvw}(t) \leq g_{uvw}(t)$ and by Lemma 3.1 $G_{uvw}(t) \leq g_{uvw}(0)e^{-3t}$.

Lemma 3.3. *Set $\tau \in \Delta$. Given the above process, let $x_{dif} = \lim_{t\to\infty} h_t$. Then $f(x_{dif}, \tau) \leq \frac{4}{3}f(h, \tau)$. Additionally, $x_{dif} \in \Delta$.*

Proof. In what follows we use the facts that $f(x_{dif}, \tau) = \lim_{t\to\infty} \frac{1}{n-2}\sum_{uvw} H_{uvw}(t)$ and that $\int_0^\infty G_{uvw}(t)dt \leq \int_0^\infty g_{uvw}(0)e^{-3t}dt \leq \frac{1}{3}g_{uvw}(0)$.

$$f(x_{dif}, \tau) = \lim_{t\to\infty} \sum_{u<v<w} H_{uvw}(t) = \sum_{u<v<w} \left(C_{uvw}(0) + \int_0^\infty G_{uvw}(t)\right) \leq \sum_{u<v<w} \left(C_{uvw}(0) + \frac{1}{3}g_{uvw}(0)\right)\ .$$

It suffices to show that $C_{uvw}(0) \geq g_{uvw}(0)$. Indeed, that would imply $C_{uvw}(0) + \frac{1}{3}g_{uvw}(0) \leq \frac{4}{3}C_{uvw}(0)$. To see that, it suffices to check that $C_{uvw}(0) \geq 0$ and $C_{uvw}(0) \geq h(u,v) - h(v,w) - h(w,u)$ for any $h(u,v)$, $h(v,w)$ and $h(w,u)$ in $[0,1]$. Notice that $C_{uvw}(0) - [h(u,v) - h(v,w) - h(w,u)]$ is a linear function in h defined on the convex set $[0,1]^3$ and thus attains its maximal values at its extreme points, i.e. integer values of h. Enumerating these cases and validating the statement is straightforward. □

Lemma 3.3 immediately implies that $(x_{dif} = \lim_{t\to\infty} h_t, C_{dif} = 4/3)$ is a feasible solution to LP.

4 QuickCluster with a tweak

We prove Theorem 2.2 and Theorem 2.3. The QuickCluster algorithm described here is very similar to the one used in [2] but the new analysis provides a shortcut that allows us to directly argue about the cost of the algorithm against an unknown truth $\tau \in \mathcal{C}$ which we hold fixed. To describe our algorithm we need to define a piecewise linear tweaking function $\psi : [0,1] \to [0,1]$ as follows: $\psi(a) = 0$ for $a \leq 1/6$, $\psi(a) = 1$ for $a \geq 5/6$, and in the middle section $a \in [1/6, 5/6]$ ψ is obtained by linear interpolation as $\psi(a) = (6a-1)/4$. Moreover, for convenience we overload the definition of ψ such that $\psi(u,v) \equiv \psi(h(u,v))$. The algorithm begins by setting all nodes $u \in V$ as *free*. In each iteration one node is chosen uniformly at random from all *free* nodes, say u, to serve as a cluster center. Then, each node $v \neq u$ is added to the cluster centered at u with probability $\overline{\psi(u,v)}$ (and set as not-*free*). The algorithm terminates when there are no *free* nodes left. Note that QuickCluster is defined for all $h \in \mathcal{K}$ and that for $h \in \mathcal{I}$ QuickCluster is identical to the algorithm in [2]. Also, Ailon [17] used a similar tweaking idea to improve rounding of a ranking LP in a traditional combinatorial optimization setting.

4.1 The Expected Cost of QuickCluster

Let x be an output of QuickCluster on h. By definition of f and the fact that τ is fixed we have that $E[f(x,\tau)] = \sum_{u<v} E[x(u,v)]\overline{\tau(u,v)} + E[\overline{x(u,v)}]\tau(u,v)$. Since $x(u,v)$ is a binary r.v. its expectation is equal to the probability of it being equal 1 which is equal to the probability of QuickCluster separating (cross-clustering) u and v. This happens if either u or v are chosen as centers and then not co-clustered (w.p. $\psi(u,v)$). This also happens if a third node w is chosen as a center and and it co-clusters either u or v but not both. Similarly $E[\overline{x(u,v)}]$ is equal to the co-clustering probability of u and v which occurs if either u or v are chosen as centers and joined or if a third node, w, co-clusters both of them. Define p_{uv} as the probability that during the execution of the algorithm v and u are both free and one of them is chosen as a center. Define p_{uvw} as the probability that during the execution of QuickCluster, u, v and w are all free and one of them is chosen as center. Also note that the relation of u and v in the output of QuickCluster is determined exactly once.

Lemma 4.1. *Fix $\tau \in \mathcal{C}$. Let $L_\psi : \binom{V}{2} \to \mathbb{R}^+, \beta : \binom{V}{3} \to \mathbb{R}^+$ and $B : \binom{V}{2} \times V \to \mathbb{R}^+$ be defined as*

$$\begin{aligned} L_\psi(u,v) &:= \psi(u,v)\overline{\tau(u,v)} + \overline{\psi(u,v)}\tau(u,v) \\ \beta(u,v;w) &:= \overline{\psi(w,u)}\ \overline{\psi(w,v)}\tau(u,v) + \psi(w,u)\overline{\psi(w,v)}\ \overline{\tau(u,v)} + \overline{\psi(w,u)}\psi(w,v)\ \overline{\tau(u,v)} \\ B(u,v,w) &:= \frac{1}{3}[\beta(u,v;w) + \beta(v,w;u) + \beta(w,u;v)]\ . \end{aligned}$$

Then $E_{x\sim QC(h)}[f(\tau,x)] = \sum_{u<v} p_{uv}L_\psi(u,v) + \sum_{u<v<w} p_{uvw}B(u,v,w)$, where $x \in \mathcal{C}$ is a random clustering obtained as the output of QuickCluster.

Proof. Following the above discussion:

$$\begin{aligned} E_{x\sim QC(h)}[x(u,v)] &= p_{uv}\psi(u,v) + \sum_{w\neq u,v} \frac{1}{3} p_{uvw}[\psi(w,u)\overline{\psi(w,v)} + \overline{\psi(w,u)}\psi(w,v)] \\ E_{x\sim QC(h)}[\overline{x(u,v)}] &= p_{uv}\overline{\psi(u,v)} + \sum_{w\neq u,v} \frac{1}{3} p_{uvw}[\overline{\psi(w,u)}\ \overline{\psi(w,v)}] \ . \end{aligned}$$

And so by linearity of expectation $E[\tau(u,v)\overline{x(u,v)} + \overline{\tau(u,v)}x(u,v)] = p_{uv}L_\psi(u,v) + \sum_{w\neq u,v} \frac{1}{3} p_{uvw}\beta(u,v;w)$.

$$\begin{aligned} E_{x\sim QC(h)}[f(\tau,x)] &= \sum_{u<v} p_{uv}L_\psi(u,v) + \sum_{u<v}\sum_{w\neq u,v} \frac{1}{3} p_{uvw}\beta(u,v;w) \\ &= \sum_{u<v} p_{uv}L_\psi(u,v) + \sum_{u<v<w} p_{uvw}\frac{1}{3}[\beta(u,w;v) + \beta(u,v;w) + \beta(v,w;u)] \\ &= \sum_{u<v} p_{uv}L_\psi(u,v) + \sum_{u<v<w} p_{uvw}B(u,v;w) \ , \end{aligned}$$

as required. □

4.2 QuickCluster decomposition

In order to compute $f(h,\tau)$ we introduce a general decomposition for the sum $\sum_{u<v} Z(u,v)$ for any function $Z : \binom{V}{2} \to \mathbb{R}$. Then, we apply our decomposition to $Z(u,v) = L_h(u,v) = h(u,v)\overline{\tau(u,v)} + \overline{h(u,v)}\tau(u,v)$.

Lemma 4.2. *Let Z be any function $Z : \binom{V}{2} \to \mathbb{R}$. Let $C(u,v;w) := \overline{\psi(w,u)}\ \overline{\psi(w,v)} + \psi(w,u)\overline{\psi(w,v)} + \overline{\psi(w,u)}\psi(w,v)$. Define the operator $A_Z : (\binom{V}{2} \to \mathbb{R}) \to (\binom{V}{3} \to \mathbb{R})$ on Z as:*

$$A_Z(u,v,w) := \frac{1}{3}\Big[\ C(u,v;w)Z(u,v) + C(v,w;u)Z(v,w) + C(w,u;v)Z(w,u)\Big] \ . \tag{3}$$

Then one has:

$$\sum_{u<v} Z(u,v) = \sum_{u<v} p_{uv}Z(u,v) + \sum_{u<v<w} p_{uvw}A_Z(u,v,w) \ .$$

Proof. The term $C(u,v;w)$ gives the probability that the node w determines the relation between u and v given that u, v and w are free and w is chosen as center. Since the relation between u and v is determined only once either indirectly (via w) or directly (either u or v are centers) we have:

$$p_{uv} + \sum_{w\neq u,v} \frac{1}{3} p_{uvw}C(u,v;w) = 1. \tag{4}$$

By (4), $Z(u,v) = 1\cdot Z(u,v) = \Big[p_{uv} + \sum_{w\neq u,v} \frac{1}{3} p_{uvw}C(u,v;w)\Big]\ Z(u,v)$. Hence,

$$\begin{aligned} \sum_{u<v} Z(u,v) &= \sum_{u<v} p_{uv}Z(u,v) + \sum_{u<v}\sum_{w\neq u,v} \frac{1}{3} p_{uvw}C(u,v;w)Z(u,v) \\ &= \sum_{u<v} p_{uv}Z(u,v) + \sum_{u<v<w} \frac{1}{3} p_{uvw}C(u,v;w)Z(u,v) \\ &\quad + \sum_{u<w<v} \frac{1}{3} p_{uvw}C(u,v;w)Z(u,v) + \sum_{w<u<v} \frac{1}{3} p_{uvw}C(u,v;w)Z(u,v) \\ &= \sum_{u<v} p_{uv}Z(u,v) + \sum_{u<w<v} p_{uvw}A_Z(u,v,w) \ . \end{aligned}$$

□

Applying Lemma 4.2 to the cost function $f(h,\tau)$ we gain:

$$f(h,\tau) = \sum_{u<v} L_h(u,v) = \sum_{u<v} p_{uv} L_h(u,v) + \sum_{u<w<v} p_{uvw} A_{L_h}(u,v,w) \tag{5}$$

4.3 Bounded Ratio Argument

To bound the ratio $f(x,\tau)/f(h,\tau)$ using Equation (5) and Lemma 4.1 it suffices to bound $L_\psi(u,v)/L_h(u,v)$ for every pair $\{u,v\}$ and $B(u,v,w)/A_{L_h}(u,v,w)$ for every triplet $\{u,v,w\}$.

In the case where $h \in \Delta$ we have that $L_\psi(u,v)/L_h(u,v) \leq 6/5$ and $B(u,v,w)/A_{L_h}(u,v,w) \leq 3/2$. Showing this entails breaking the polytope defining $(h(u,v), h(v,w), h(w,u))$ into 27 smaller polytopes in which each $h(\cdot,\cdot)$ is constrained to lie in $[0,1/6]$, $(1/6,5/6]$, or $(5/6,1]$. On each of these smaller polytopes and for each one of 5 possibilities for τ on u,v,w, the functions L_h, L_ψ are linear, and B and A_{L_h} are multinomials of total degree two and three respectively.[5] A computer aided proof was used to obtain the bound of $3/2$ using standard polynomial maximization techniques on each one of the polytopes. We refer the reader to [18] for details. This proves Theorem 2.2.

When $h \in \mathcal{I}$, enumerating over all possible choices of h and τ gives that $L_\psi(u,v)/L_h(u,v) = 1$ and $B(u,v,w)/A_{L_\psi}(u,v,w) \leq 2$. This shows that performing QuickCluster directly on h without solving the LP gives a $C = 2$ approximation ratio. This proves Theorem 2.3.

5 Running Time

Running time with pairwise queries: We prove Theorem 2.4 using Yao's minimax Lemma [19]. To use the lemma it is enough to show that there exists one distribution $\mathbf{h}$ on inputs $h \in \mathcal{I}$ for which any *deterministic* algorithm A which is a C-approximation for CorrelationClusteringX makes $\Omega(n^2)$ queries into h on expectation, for some constant C.

We choose $\mathbf{h}$ to be the uniform distribution over inputs $h \in \mathcal{C}$ such that only two elements are clustered together and the rest are singletons. In other words, for some u_0, v_0, $h(u_0,v_0) = 0$ and $h(u,v) = 1$ for $\{u,v\} \neq \{u_0,v_0\}$. Notice that for all h in the support of $\mathbf{h}$, $h \in \mathcal{C}$. Therefore, there exists a unique $\tau \in \mathcal{C}$ for which $f(\tau,h) = 0$, namely $\tau = h$. The algorithm A must therefore output h on input h. The problem of finding the unique null coordinate of an $\binom{n}{2}$ dimensional vector (where the null coordinate is chosen uniformly at random) clearly reduces to this task, and it is well known that the expected number of queries into the vector must be $\Omega(n^2)$, as required.

Running time with neighborhood queries: We prove Theorem 2.5. We claim that given a stronger oracle, the expected running time of QuickCluster can be reduced to $O(n + \min_{\tau\in\mathcal{C}} f(\tau,h))$. We restrict our proof to the case where $h \in \mathcal{I}$ (and hence the tweaking function is not necessary), although it is very simple to extend the proof to the fractional case as well. Fix an arbitrary $\tau \in \mathcal{C}$. The stronger oracle receives as query a single element $u \in V$ and returns $N(u)$ where $N(u) = \{u\} \cup \{v \mid h(u,v) = 0\}$ as a linked list.

Let $u_1, \ldots, u_k \in V$ be the k centers chosen by QuickCluster. Since for each center u_i QuickCluster performs $O(|N(u_i)|)$ operations we have that $T(\mathit{QuickCluster}, h) \leq O(\sum_{u_i} |N(u_i)|)$, where $T(\mathit{QuickCluster}, h)$ is the running time of *QuickCluster* on input h. Let us count $\sum_{u_i} |N(u_i)|$ in differently. For every element $v \in N(u_i)$ one of two events occur. One, v is free when u_i is chosen and thus v is assigned to cluster i, this event cannot occur more than $|V| = n$ times. Two, v is already assigned and thus $h(u_i,v) = 0$ and $x(u_i,v) = 1$ (x is the clustering output of QuickCluster). The second event occurs at most $f(x,h)$ times, the number of disagreements between x and h. We have that $\sum_{u_i} |N(u_i)| \leq n + f(x,h)$. Moreover, due to the triangle inequality on the function f and Theorem 2.2, $E[f(x,h)] \leq E[f(x,\tau^\dagger)] + f(\tau^\dagger,h) \leq 3[f(\tau^\dagger,h)]$ for all $\tau^\dagger$. Choosing $\tau^\dagger = \arg\min_{\tau\in\mathcal{C}} f(\tau,h)$ yields $E[T(\mathit{QuickCluster}, h)] \leq O(n + \min_{\tau\in\mathcal{C}} f(\tau,h))$ as required.

Remark: The running time above assumes that choosing an index from $1,\ldots,n$ uniformly at random requires $O(1)$ operations. Depending on the computational model, this might require $\theta(\log(n))$ operations which would add an $O(k\log(n))$ term to the above running time.

[5] The 5 possibilities for τ are: One single cluster, 3 singleton clusters, and the 3 ways to get a singleton and a pair.

6 Randomness is Necessary

It is clear by definition that randomized CorrelationClusteringX can achieve a bound which is at least as good as deterministic CorrelationClusteringX. Here we show a simple case that illustrates that the gap between the two cases is nonzero. Consider an input to CorrelationClusteringX consisting of three elements $V = \{u, v, w\}$ and an inconsistent instance of h, $h(u, v) = h(v, w) = 0$, $h(w, u) = 1$. In what follows we enumerate the possible outputs, x, of a deterministic algorithm and the adversarial choice of τ which gives $f(\tau, x) \geq 2f(\tau, h)$

1. $x(u, v) = x(v, w) = x(w, u) = 0$; $\tau(u, v) = \tau(w, u) = 1, \tau(v, w) = 0$ giving $f(\tau, x) = 2f(\tau, h)$;
2. $x(u, v) = x(v, w) = x(w, u) = 1$; $\tau(u, v) = \tau(v, w) = \tau(w, u) = 0$ resulting in $f(\tau, x) = 3f(\tau, h)$
3. $x(u, v) = 0, x(v, w) = (w, u) = 1$; $\tau(u, v) = \tau(v, w) = \tau(w, u) = 0$ resulting in $f(\tau, x) = 2f(\tau, h)$
4. $x(w, u) = 0, x(u, v) = x(v, w) = 1.$; $\tau(u, v) = \tau(v, w) = \tau(w, u) = 0$ yielding $f(\tau, x) = 2f(\tau, h)$

This means that the best approximation we can get for the deterministic case is 2. However, if we allow randomness, consider an algorithm outputting $(x(u, v), x(v, w), x(w, u))$ either $(0, 0, 0)$, $(0, 1, 1)$, or $(1, 0, 1)$ each with probability $1/3$. It is easy to see, by testing all 6 cases of τ that the worst the adversary can do for the algorithm is to choose $\tau = (0, 0, 0)$, for which $f(\tau, h) = 1$ and $E[f\tau, x] = 4/3$, resulting in a factor of $4/3$. This proves Theorem 2.6.

7 Discussion

7.1 Connection to Bregman divergence.

Bregman divergence [20] is a family of distance functions used in statistics and in online optimization. Most notable examples include the Euclidean metric and KL-divergence (also referred to as entropy loss). If $B(x, y)$ is a Bregman divergence function on $x, y \in \mathbb{R}^n$, $P \subseteq \mathbb{R}^n$ is a convex set and $w \in \mathbb{R}^n$ is some point, then the P-projection w^* of w onto P (defined as $w^* = \text{argmin}_{u \in P} B(u, w)$) satisfies that for all $\tau \in P$, $P(\tau, w) \geq P(\tau, w^*) + P(w^*, w)$. This is called the *Pythagorean* property. It implies, in particular, that $B(\tau, w) \geq B(\tau, w^*)$ for all $\tau \in W$. In other words, the solution w^* to a distance minimization (projection) problem "minimize $P(u, w), u \in B$" is also a solution to the following "Pythagorean" problem: "Find $w' \in P$ such that $B(\tau, w') \leq B(\tau, w)$ for all $\tau \in P$". Bregman divergence functions are often used as loss functions in online optimization problems, where the cost is against an unknown ground truth τ (such as the true price of stocks, revealed only after our portfolio is chosen). A generalized gradient descent step (e.g. multiplicative weights) is taken, giving rise to a solution w possibly *outside* the feasible set P (e.g. the simplex in the portfolio management case). A Bregman projection to P fixes this problem without compromising the loss against the ground truth, in virtue of the Pythagorean property. Our work can therefore be considered as a generalization of Bregman divergence theory, where the optimization is here over a discrete (hence, non convex) set, and the Pythagorean inequality is relaxed by a constant. The analogue of Correlation Clustering is clear: The original problem definition [1] is analogous to geometric *projection*, and our problem CorrelationClusteringX is the "Pythagorean" version.

7.2 Other

- The objective of traditional Correlation Clustering is to minimize the loss of the output clustering with respect to h and not with respect to τ as we do here. Our algorithm trivially also gives an expected factor of $2 + 1 = 3$ approximation to the traditional problem by triangle inequality of f. Note that the best known approximation factor for Correlation Clustering is 2.5 [2], raising the question of whether it is possible to obtain 1.5 for CorrelationClusteringX.

- Finding a specific instance h for which QuickCluster achieves the 2 approximation bound for CorrelationClusteringX will show that our analysis is tight. The worst input known to the authors is h corresponding to the balanced complete bipartite graph ($h(u, v) = 0$ if $\{u, v\} \in e$) for which QuickCluster gives a 1.5 approximation factor (for τ which puts all of V into one cluster).

- Optimizing with respect to an unknown truth gives us a new regime in which to design and analyze algorithms, between combinatorial optimization and machine learning. It will be interesting to apply this notion to other traditional combinatorial optimization problems.

Acknowledgments: The authors would like to thanks Eyal Even-Dar, Mehryar Mohri, and Elad Hazan for sharing their insights and expertise.

References

[1] Nikhil Bansal, Avrim Blum, and Shuchi Chawla. Correlation clustering. *Machine Learning Journal (Special Issue on Theoretical Advances in Data Clustering)*, 56(1–3):89–113, 2004. Extended abstract appeared in FOCS 2002, pages 238–247.

[2] Nir Ailon, Moses Charikar, and Alantha Newman. Aggregating inconsistent information: ranking and clustering. In *Proceedings of the 37th Annual ACM Symposium on Theory of Computing (STOC)*, pages 684–693, 2005.

[3] Paola Bonizzoni, Gianluca Della Vedova, Riccardo Dondi, and Tao Jiang. On the approximation of correlation clustering and consensus clustering. *Journal of Computer and System Sciences*, 74(5):671–696, 2008.

[4] D. Emanuel and A. Fiat. Correlation clustering – minimizing disagreements on arbitrary weighted graphs. In *In Proc. of 11th ESA, volume 2832 of LNCS, pages 208–220. Springer.*, 2003.

[5] Moses Charikar, Venkatesan Guruswami, and Anthony Wirth. Clustering with qualitative information. In *Proceedings of the 44th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 524–533, Boston, 2003.

[6] Ioannis Giotis and Venkatesan Guruswami. Correlation clustering with a fixed number of clusters. In *SODA '06: Proceedings of the seventeenth annual ACM-SIAM symposium on Discrete algorithm*, pages 1167–1176, New York, NY, USA, 2006. ACM.

[7] Nir Ailon and Moses Charikar. Fitting tree metrics: Hierarchical clustering and phylogeny. In *Proceedings of the 46th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, 2005.

[8] Aristides Gionis, Heikki Mannila, and Panayiotis Tsaparas. Clustering aggregation. In *Proceedings of the 21st International Conference on Data Engineering (ICDE)*, Tokyo, 2005. To appear.

[9] Vladimir Filkov and Steven Skiena. Integrating microarray data by consensus clustering. In *Proceedings of International Conference on Tools with Artificial Intelligence (ICTAI)*, pages 418–425, Sacramento, 2003.

[10] Alexander Strehl. Relationship-based clustering and cluster ensembles for high-dimensional data mining. *PhD Dissertation, University of Texas at Austin*, May 2002.

[11] Maria-Florina Balcan, Nikhil Bansal, Alina Beygelzimer, Don Coppersmith, John Langford, and Gregory B. Sorkin. Robust reductions from ranking to classification. In Nader H. Bshouty and Claudio Gentile, editors, *COLT*, volume 4539 of *Lecture Notes in Computer Science*, pages 604–619. Springer, 2007.

[12] John Langford and Alina Beygelzimer. Sensitive error correcting output codes. In *The 18th Annual Conference on Learning Theory (COLT)*, 2005.

[13] Nir Ailon and Mehryar Mohri. Efficient reduction of ranking to classification. In *To appear: The 21st Annual Conference on Learning Theory (COLT) , Helsinki, Finland*, 2008.

[14] F. McSherry. Spectral partitioning of random graphs. In *FOCS '01: Proceedings of the 42nd IEEE symposium on Foundations of Computer Science*, page 529, Washington, DC, USA, 2001.

[15] J. Aslam, A. Leblanc, and C. Stein. A new approach to clustering. In *4th International Workshop on Algorithm Engineering*, 2000.

[16] Maria-Florina Balcan, Avrim Blum, and Anupam Gupta. Approximate clustering without the approximation. In *SODA'09*, New York, NY, 2009.

[17] Nir Ailon. Aggregation of partial rankings, p-ratings and top-m lists. In *SODA*, 2007.

[18] Nir Ailon and Edo Liberty. Mathematica program, 2008. `http://www.cs.yale.edu/homes/el327/public/prove32/`.

[19] Andrew Chi-Chih Yao. Probabilistic computations: Toward a unified measure of complexity. In *FOCS*, pages 222–227, 1977.

[20] L. M. Bregman. The relaxation method of finding the common point of convex sets and its application to the solution of problems in convex programming. *USSR Computational Mathematics and Mathematical Physics*, 7:200–217, 1967.

A

Lemma A.1. *For any $\{s,u,v,w\} \in V$ we have $F(a;x,y)-F(b;z,y)-F(c;x,z) \leq 0$, $a = h(u,v)$, $b = h(v,w)$, $c = h(w,u)$, $x = h(u,s)$, $y = h(v,s)$, $z = h(w,s)$, $a \geq b+c$ and F is as defined in Equation (2).*

Proof. Let $\gamma := F(a;x,y) - F(b;y,z) - F(c;x,z)$. We need to show $\gamma \leq 0$. We analyze each of the following

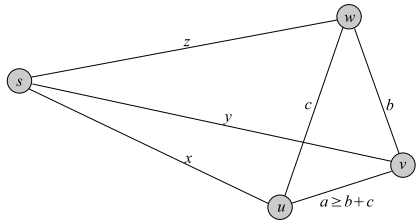

Figure 3: The tetrahedral structure of the triangle $\{u,v,w\}$ and the three triangles adjacent to it which contain s. Lemma A.1 claims that the sum of forces applied by the triangles $\{u,v,s\}$, $\{v,w,s\}$ and $\{w,u,s\}$ cannot act to increase the gap g_{uvw}.

cases separately.

1. $F(a;x,y) < 0$. This means that $a \geq x+y$ and $F(a;x,y) = (x+y-a)$. If both $F(b;x,z)$ and $F(c;y,z)$ are nonnegative then the inequality is trivially satisfied. We thus assume w.l.o.g. that $b \geq z+y$ and $F(b;y,z) = z+y-b \leq 0$. Hence $\gamma \leq 0$ if $F(c;x,z) \geq (x+y-a)-(z+y-b)$. Due to $a \geq b+c$ it suffices to show that $F(c;x,z) \geq x-z-c$. If $x \geq z+c$ this holds with equality. If $z \geq x+c$ then

$F(c;x,z) = z - x - c \geq x - z - c$. Finally, if $c \geq x + z$, $F(c;x,z) = x + z - c \geq x - z - c$. The last case is $z < x + c, x < c + z, c < z + x$, resulting in $F(c;x,z) = 0 \geq x - z - c$ by definition of F and by assumption.

2. $F(a;y,x) \geq 0$. This means that $a \leq x + y$. Assume w.l.o.g. that $x \leq y$. This implies that $F(a;x,y) = \max\{0, y - x - a\}$. We distinguish between 3 subcases.

   - $z \leq x \leq y$. We distinguish between two cases.
     - $b \leq y + z$. In this case $F(b;y,z) = \max\{0, y - z - b\}$. This implies $\gamma = \max\{y - x - a, 0\} - \max\{y - z - b, 0\} - F(c;x,z)$. Clearly $y - x - a \leq y - z - b$ (because $z \leq x$ and $b \leq a$), hence $\max\{y - x - a, 0\} \leq \max\{y - z - b, 0\}$. Therefore, $\gamma \leq 0$ trivially if $F(c;x,z) \geq 0$. Assume otherwise that $F(c;x,z) < 0$. This implies that $c > x + z$ and $F(c;x,z) = x + z - c$, or $\gamma = \max\{y - x - a, 0\} - \max\{y - z - b, 0\} - (x + z - c)$. If $y \geq x + a$ (and hence also $y \geq z + b$) then $\gamma = (y - x - a) - (y - z - b) - (x + z - c) = -2x - a + b + c \leq 0$ by the assumption $a \geq b + c$. Otherwise $y < x + a$ and $\gamma = -\max\{y - z - b, 0\} - (x + z - c)$. If $y \geq z + b$ This equals $y - z - b - x - z + c = b + c - y - x$, which is ≤ 0 by our assumption that $b + c \leq a \leq x + y$, as required. The remaining case $y \leq z + b$ cannot happen, because $y \geq a - x \geq b + c - x > b + x + z - x = b + z$.
     - $b > y + z$. In this case $F(b;y,z) = y + z - c$. This imples $\gamma = \max\{y - x - a, 0\} - y - z + c - F(c;x,z)$. Our assumptions also imply $c < x - z$ (indeed, by assumptions $c \leq a - b \leq x + y - b < x + y - y - z = x - z$), consequently $F(c;x,z) = x - z - c$. Hence, $\gamma = \max\{y - x - a, 0\} - y - z + c - x + z + c = \max\{y - x - a, 0\} - y - x + 2c$. Also notice that by assumptions $y < b - z \leq a - c - z$ which is trivially $\leq a + x$, hence $\max\{y - x - a, 0\} = 0$ and $\gamma = -y - x + 2c \leq -2x + 2c \leq -2x + 2(a - b) = 2(-x + a - b) \leq 2(y - b)$, where the last inequality is due to the assumption $a \leq x + y$. The last expression $2(y - b)$ in the last chain is ≤ 0 by our assumption that $b > y + z$ in this case, as required.
   - $x \leq z \leq y$. In this case we have that $b \leq y + z$ (otherwise we would have $a \geq b > y + z \geq y + x$, a contradiction to the assumption $a \leq x + y$). This implies that $\gamma = \max\{y - x - a, 0\} - \max\{y - z - b, 0\} - F(c;x,z)$. We distinguish two subcases.
     - $c \leq x + z$. This implies $\gamma = \max\{y - x - a, 0\} - \max\{y - z - b, 0\} - \max\{z - x - c, 0\}$. If $y \geq x + a$ then $\gamma = y - x - a - \max\{y - z - b, 0\} - \max\{z - x - c, 0\} \leq (y - x - a) - (y - z - b) - (z - x - c) = b + c - a$ which is ≤ 0 by assumption, as required. Otherwise, $y < x + a$ implying $\gamma = 0 - \max\{z - y - b, 0\} - \max\{x - z - c, 0\} \leq 0$ trivially, as required.
     - $c > x + z$, implying $F(c;x,z) = x + z - c$ and hence $\gamma = \max\{y - x - a, 0\} - \max\{y - z - b, 0\} - (x + z - c)$. Our current assumptions are $y + x \geq a$, $c > z + x$, $a \geq b + c$. Summing them, we conclude $y > z + b$, implying that $\gamma = \max\{y - x - a, 0\} - (y - z - b) - (x + z - c) = \max\{y - x - a, 0\} - y - x + b + c$. If $y \geq x + a$ this equals $y - x - a - y - x + b + c = b + c - a - 2x$, which is ≤ 0 by our assumption $a \geq b + c$. Otherwise it equals $0 - y - x + b + c \leq -y - x + a$ which is again ≤ 0 by the assumption $a \leq x + y$.
   - $x \leq y \leq z$. First, we have that $b \leq z + y$. Otherwise $b > z + y$ implying $a > z + y \geq x + y$. Similarly $c \leq x + z$. Thus $\gamma = \max\{y - x - a, 0\} - \max\{z - y - b, 0\} - \max\{z - x - c, 0\}$. If $y \geq x + a$ then also $z \geq x + b$ (because $a \geq b$ by assumption) and hence $\gamma \leq (y - x - a) - (z - y - b) - \max\{z - x - c, 0\} \leq (y - x) - (a - b) - (z - y) \leq 0$ due to the assumptions. If $y \leq x + a$ then γ is the sum of two nonpositive numbers making it nonpositive as well.

□