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 only reasonable way to measure the cost of a solution is by comparing it with the input similarity function. This problem has been studied in theory and application 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 reality and not on our noisy perception thereof. 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 many other combinatorial optimization problems.

Our analysis consists of two solutions. The first gives a simple 2-approximation algorithm. The second 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.

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 "do its best" with respect to the given similarity function. This gives rise to an NP-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 cases such as duplicate detection and elimination in large data (also known as the record linkage problem).

The question is, how does there being an underlying true clustering help us if we don't know what that truth is? Our main result is a clustering algorithm taking the similarity function as input in such a way that guarantees that if the similarity function "respects" the true clustering, then the output will respect the true clustering. 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". Traditional combinatorial optimization gives an indirect solution to the problem. In this work we show how to "shortcut" that approach and obtain better results.

To make the story clear, we introduce some notation. Assume we are interested in (some function of) 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^*. We are interested in cases where τ^* comes from 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 $U \supseteq P$. 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.

Our problem is described precisely as follows. In the deterministic case, the goal is to output $x \in P$, depending on h only, such that $f(\tau^*, x) \leq Cf(\tau^*, h)$ for some global constant C (as small as possible). In the randomized case, the goal is to output a sample from a distribution $\mathcal{D}$ depending only on h such that $E_{x\sim\mathcal{D}} f(\tau^*, x) \leq Cf(\tau^*, h)$ for the smallest possible C. Note that we have to argue against an adversarial choice of τ^*, since we assume no knowledge on it. The corresponding decision problems are, given $h \in U, C \geq 1$:

$$\text{Randomized:} \quad \exists \mathcal{D}.\ \forall \tau^* \in P\ .\ E_{x\sim\mathcal{D}} f(\tau^*, x) \leq Cf(\tau^*, h)\ ? \tag{1}$$

$$\text{Deterministic:} \quad \exists x \in P.\ \forall \tau^* \in P\ .\ f(\tau^*, x) \leq Cf(\tau^*, h)\ ? \tag{2}$$

(Where $\mathcal{D}$ is a distribution over the elements of P). Note that unlike in traditional combinatorial optimization, where sufficient computational power always enables us to output a solution with approximation ratio of 1, obtaining $C = 1$ here may not always be possible: this problem is primarily of combinatorial and information theoretical and not computational nature. Being able to output x (deterministically or as a sample from $\mathcal{D}$) *efficiently* is of separate, independent interest for practical purposes.

Note that this setting is akin to the task of statistical prediction in the sense that the cost of the output (the action we take based on our observation), is determined by reality (such as the actual future price of a stock) regardless of how much computational power was used (to predict the future price). On the other hand, a bad model or insufficient observations are never assumed to be good for prediction. The hope of a good output therefore always relies on the premise of having a "good" or "close" model (though possibly incomplete or approximate). This is the intuition behind comparing $f(\tau^*, x)$ with $f(\tau^*, h)$ in (1) and (2): we cannot hope to get more information than that offered by h. In prediction problems, however, the ground truth is often assumed to be stochastic.[1] Here we assume the game is played against an adversary who knows our algorithm and picks the worst possible ground truth. Hence our approach can be considered to be a combination of machine learning and combinatorial optimization.

In *machine learning reductions* (see [11, 12, 13] and references therein), a complex, structured learning problem is divided into small (ideally binary) learning tasks, ignoring dependencies. Invoking the collection of small learners on data gives rise to an ensemble of "small" hypotheses, which need to be converted somehow into a consistent output. The cost of the output is not measured against the hypothesis ensemble, but against the true (unknown) structure.[2]

As claimed above, traditional optimization gives the following indirect solution to problems (1) and (2): 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 the above decision 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 question of whether we can go below 2 by shortcutting the traditional optimization detour (which is often an obstruction under certain complexity theoretical assumptions).

Very recently Ailon and Mehryar [13], improving pioneering work in the context of machine learning reductions by Balcan et al [11] showed that it is possible to achieve $C = 2$ for a ranking problem, with an

[1] In addition to these examples, it is worth noting that much effort has been devoted in recent years to study stochastic versions of combinatorial optimization problems traditionally studied as worst case problems. This is different from what we do here.

[2] There are exceptions. For example Cohen et al [14] consider minimizing the distance of a ranking output to a collection of comparison hypotheses, and prove that they stumble upon an NP-Hard optimization problem.

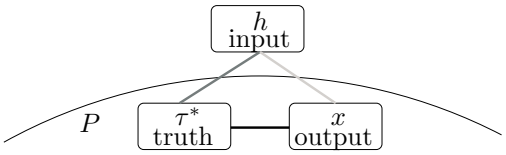

Figure 1: Given an input h, a traditional optimization approach tries to find x minimizing the maximal ratio between the length of the light gray line and the dark gray line over all τ^* . Our approach is to find x minimizing the maximal ratio between the black line and the dark gray line over all τ^*.

improvement to $C = 1$ for a particular case related to a standard information retrieval cost function (in the randomized setting).

Here we study Correlation Clustering in this setting. We show that $C = 2$ can be obtained in the randomized setting, using distribution $\mathcal{D}$ on clusterings from which we can efficiently sample. This is better than what the traditional optimization approach can achieve because the corresponding optimization problem is APX-Hard [5]. Additionally, for any input h, we show that it is possible to compute, in polynomial time, a number $C_{LP} = C_{LP}(h) \leq 4/3$, such that the answer to (2) is affirmative with $C = 3C_{LP}/2$, using a distribution $\mathcal{D}$ on clusterings from which we can sample efficiently. If $C_{LP} < 4/3$ then this serves as a witness for the possibility of getting $C < 2$. The invariant C_{LP} is the value of a convex optimization problem which we write as an LP with an exponential number of constraints.

Interestingly, the first algorithm we present in Section 4, giving the 2-approximation (in the sense of decision problem (2)) is the same one used in [2] for traditional Correlation Clustering optimization. The shortcut is done in the analysis. The running time is analyzed in Section 7. In Section 5 we then present a new algorithm based on *differential programming*: We show that the limit at infinity of a solution to a piecewise linear differential equation in $\binom{n}{2}$ dimensions (where n is the number of elements we wish to cluster) is a fractional solution to the deterministic decision problem (1) with a factor of at most $4/3$. We do not need to explicitly solve this differential equation, but just need its existence. The actual best fractional solution can be obtained by solving a certain LP with an exponential number of constraints, for which a separation oracle exists. The fractional solution satisfies a desirable triangle inequality property, which makes it amenable to a clustering algorithm (Section 6) with a factor of at most $3/2$ (in the sense of decision problem (2)). The combined solution gives a factor of at most $4/3 \times 3/2 = 2$ again, but offers the additional bonus of obtaining a witness from the LP proving that the actual factor is strictly less than 2 in some cases. Finally, in Section 8 we show that 2 is the best possible constant in the deterministic case, and that there is a strict gap between the deterministic and randomized cases.

2 Definitions

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 symmetric $[0, 1]$-valued symmetric functions on $V \times V$ (with a null diagonal). Let $\mathcal{I} \subseteq \mathcal{K}$ denote the set of integer values 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.

Our input h lives in $\mathcal{I}$ but not in Δ, hence the function h only serves as noisy and incomplete 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] $f : \mathcal{C} \times \mathcal{K} \to \mathbb{R}^+$ as $f(c, k) = \sum_{u<v}(c(u, v)\overline{k(u, v)} + \overline{c(u, v)}k(u, v))$.

[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 a metric over the values {0,1}.

Definition 2.1. *The problem of CorrelationClusterX is defined as follows: Given $h \in \mathcal{I}$ and $C \geq 1$ output $x \in \mathcal{C}$ such that for all $\tau^* \in \mathcal{C}$, $f(\tau^*, x) \leq Cf(\tau^*, h)$ (assume promise of existence of x). In the randomized setting, the goal is to output a sample x from a distribution $\mathcal{D}$ on $\mathcal{C}$, such that $E_{x\sim\mathcal{D}}[f(\tau^*, x)] \leq Cf(\tau^*, h)$ (assume promise of existence of $\mathcal{D}$). 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 CorrelationClusterX.*

Deterministic CorrelationClusterX has a corresponding integer program over the $\binom{n}{2}$ variables of $x \in \mathcal{C}$ with an exponential number of constraints:

$$\begin{aligned} \text{IP: minimize } C \text{ s.t.} \quad & f(\tau^*, x) \leq Cf(\tau^*, h) \text{ for all } \tau^* \in \mathcal{C} \\ & x \in \mathcal{C}, C \geq 1 \end{aligned}$$

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

$$\begin{aligned} \text{LP: minimize } C \text{ s.t.} \quad & f(\tau^*, x) \leq Cf(\tau^*, h) \text{ for all } \tau^* \in \Delta \\ & x \in \Delta, C \geq 1 \ . \end{aligned}$$

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

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. In this work we will show the existence of a feasible solution (x_{LP}, C_{LP}) with $C_{LP} \leq 4/3$ by driving a solution to a piecewise linear differential equation to infinity.

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 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.

Definition 2.2. QuickCluster: *The algorithm recives as input a set of n elements and a fuction $h \in \mathcal{I}$. It proceeds as follows: Set all elements as* free*. Pick one element uniformly at random from the* free *elements, say u, to serve as a cluster center. Let $N(u) = \{u\} \cup \{v \in V \mid h(u, v) = 0\}$ denote the neighborhood of u. Create a new cluster consisting of all the* free *elements in $N(u)$ (set these elements to no longer be* free*). Recurse until there are no more* free *elements. Return $x \in \mathcal{C}$ corresponding to the collection of created clusters.*

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 $\{u, v, w\} \in \binom{V}{3}$.

3 Statement of Results

Our first theorem states that a randomized 2-approximation algorithm for CorrelationClusterX exists. Moreover, sampling $x \sim \mathcal{D}$ can be done in polynomial time. Traditional combinatorial optimization cannot achieve this because Correlation Clustering is NP-Hard (in fact APX-Hard [5]).

Theorem 3.1. *Let QC denote the distribution over the outputs $x \in \mathcal{C}$ of QuickCluster. For any $h \in \mathcal{I}$ and $\tau^* \in \mathcal{C}$ QuickCluster on h outputs $x \in \mathcal{C}$ with $E_{x\sim QC}[f(\tau^*, x)] \leq 2f(\tau^*, h)$.*

The following theorem states that the optimal solution to LP is a (deterministic) fractional solution for CorrelationClusterX with approximation factor of at most 4/3. 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 3.2. *For any $h \in \mathcal{I}$ the corresponding value of LP is $C_{LP} \leq 4/3$. Additionally, a feasible solution of value at most $4/3$ can be obtained in polynomial time.*

Theorem 3.2 together with Observation 2.1 implies that using standard convex optimization techniques it is possible to obtain an optimal solution (x_{LP}, C_{LP}) to LP, with $C_{LP} \leq 4/3$. The next theorem shows how to randomly convert this solution to a $\frac{3}{2}C_{LP}$ approximation for CorrelationClusterX.

Theorem 3.3. *Given $h \in \mathcal{I}$ and a corresponding optimal solution (x_{LP}, C_{LP}) to LP, there exists a polynomial time algorithm outputting $x \in \mathcal{C}$ with $E_x[f(\tau^*, x)] \leq \frac{3}{2}f(\tau^*, x_{LP})$ for all $\tau^* \in \mathcal{C}$.*

Corollary 3.1. *Given $h \in \mathcal{I}$ there exists a polynomial time randomized algorithm outputting $x \in \mathcal{C}$ such that $E_x[f(\tau^*, x)] \leq \frac{3}{2}C_{LP}f(\tau^*, h)$ for all $\tau^* \in \mathcal{C}$, where $C_{LP} = C_{LP}(h)$ is the value of the corresponding LP. In particular, if $C_{LP} < 4/3$ then it serves as a witness for achieving an approximation strictly less than 2 for CorrelationClusterX.*

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 3.4. *In the pairwise queries model, any constant factor randomized approximation algorithm for CorrelationClusterX performs $\Omega(n^2)$ queries to h in 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 3.5. *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 determinisic algorithm can do. For this hard case there is a strict gap between the randomized and deterministic cases.

Theorem 3.6. *There exists an input h for which any deterministic algorithm for CorrelationClusterX incurrs an approximation factor of at least 2 for some gound truth $\tau^* \in \mathcal{C}$. For the same input, a randomized algorithm can obtain a factor of at most $4/3$.*

4 A Randomized 2 Approximation Algorithm for CorrelationClusterX

We prove Theorem 3.1. The algorithm is the same as 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. Let QC denote the distribution on the output clustering of QuickCluster on h. We show that:

$$E_{x \in QC}[f(\tau^*, x)] \leq 2f(\tau^*, h) \ . \tag{3}$$

The proof of (3) is obtained using a local ratio argument based on a decomposition of the QC probability space. We present this very useful, general decomposition principle which was used in the past for traditional clustering optimization purposes [2, 7] and apply it to the ground truth cost considered in CorrelationClusterX, instead of the cost used in Correlation Clustering. Fix $u, v \in V$. Consider how the relation (either co- or cross-cluster) between u, v is determined during the execution of QuickCluster. 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. If w is chosen and u and v are free, the relation between u and v is determined as co-clustered if w decides to co-cluster both u, v to itself, and cross-clustered if w decides to co-cluster exactly one of the two. This can be expressed using the following indicator variable:

$$C(u, v; w) := \overline{h(w, u)}\ \overline{h(w, v)} + h(w, u)\overline{h(w, v)} + \overline{h(w, u)}h(w, v) \tag{4}$$

Since the relation between u and v is determined exactly once we have the following decomposition of the probability space:

$$p_{uv} + \sum_{w \neq u,v} \frac{1}{3} p_{uvw} C_h(u, v; w) = 1. \tag{5}$$

The 1/3 comes from the fact that conditioned on one of u, v, w being chosen as center when the other two are in the same recursive input to QuickCluster, the probability of w being that center is exactly 1/3 (due to the uniform center selection).

Lemma 4.1. *Let Z be any function $Z : \binom{V}{2} \to \mathbb{R}$. Define the operator $A_Z : (\binom{V}{2} \to \mathbb{R}) \to (\binom{V}{3} \to \mathbb{R})$ on Z as:*

$$A_Z(u,v,w) := \frac{1}{3}\left[C(u,v;w)Z(u,v) + C(v,w;u)Z(v,w) + C(w,u;v)Z(w,u)\right] \ . \tag{6}$$

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. By (5), $Z(u,v) = 1 \cdot Z(u,v) = \left[p_{uv} + \sum_{w \neq u,v} \frac{1}{3} p_{uvw} C(u,v;w)\right] Z(u,v)$. Hence,

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

□

Lemma 4.2. *Fix $\tau^* \in \mathcal{C}$. Let $L : \binom{V}{2} \to \mathbb{R}^+, \beta : \binom{V}{3} \to \mathbb{R}^+$ and $B : \binom{V}{2} \times V \to \mathbb{R}^+$ be defined as*

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

Then $E_{x \in QC}[f(\tau^*, x)] = \sum_{u<v} p_{uv} L(u,v) + \sum_{u<v<w} p_{uvw} B(u,v,w)$.

Proof. By the definitions of p_{uv}, p_{uvw} and the fact that the relation of u and v in the output of QuickCluster is determined exactly once,

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

And so $E[\tau^*(u,v)\overline{x(u,v)} + \overline{\tau^*(u,v)}x(u,v)] = p_{uv}L(u,v) + \sum_{w \neq u,v} \frac{1}{3} p_{uvw} \beta(u,v;w)$.

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

as required. □

We now compare between $f(\tau^*, h)$ and $E_{x\sim QC}[f(\tau^*, x)]$. Decompose $Z(u,v) = L(u,v)$ using Lemma 4.1.

$$f(\tau^*, h) = \sum_{u<v} L(u,v) = \sum_{u<v} p_{uv} L(u,v) + \sum_{u<v<w} p_{uvw} A_L(u,v,w) \ . \tag{7}$$

From Lemma 4.2 we have that: $E_{x\sim QC}[f(\tau^*, x)] = \sum_{u<v} p_{uv} L(u,v) + \sum_{u<v<w} p_{uvw} B(u,v,w)$. The theorem is proved if we show that for any u, v and w: $B(u,v,w) \leq 2A_L(u,v,w)$. Due to the symmetry with respect to u, v and w and w.l.o.g. it is sufficient to check four cases.

1. $h(u,v) = h(v,w) = h(w,u) = 1$, yields $B(u,v,w) = 0$, giving the required result due to the nonnegativity of A_L.
2. $h(u,v) = h(v,w) = h(w,u) = 0$, implies $B^h(u,v,w) = \tau^*(u,v) + \tau^*(u,v) + \tau^*(w,u) = A_L(u,v,w)$ as required.
3. $h(u,v) = 0$, $h(v,w) = h(w,u) = 1$, implies $B(u,v,w) = \overline{\tau^*(v,w)} + \overline{\tau^*(w,u)} = A_L(u,v,w)$ as required.
4. $h(u,v) = h(v,w) = 0$, $h(w,u) = 1$, gives

$$\begin{aligned} B(u,v,w) &= \overline{\tau^*(u,v)} + \overline{\tau^*(v,w)} + \tau^*(w,u) \\ A_L(u,v,w) &= \tau^*(u,v) + \tau^*(v,w) + \overline{\tau^*(w,u)} \ . \end{aligned}$$

And so the lemma is true if for any u, v and w, $\overline{\tau^*(u,v)} + \overline{\tau^*(v,w)} + \tau^*(w,u) \leq 2\tau^*(u,v) + \tau^*(v,w) + \overline{\tau^*(w,u)}$. This is clearly the case for any clustering $\tau^* \in \mathcal{C}$ and can be verified easily by restricting to clustering to 3 elements. This concludes the proof of Theorem 3.1.

5 Morphing h Into a metric: A Differential Program

In this section we prove Theorem 3.2. The idea is to "morph" h, which is not necessarily a metric, into a (pseudo)metric (a function in Δ). The solution $\hat{h} \in \Delta$ is obtained by theoretically running the differential equation to infinity. More precisely, $\hat{h} = \lim_{t\to\infty} h_t(u,v)$ where h_t is the morphed state of h in time 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:

$$g_{uvw} = \max\{0,\ a-(b+c),\ b-(c+a),\ c-(a+b)\} \tag{8}$$

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

$$F(a;b,c) = \begin{cases} -g_{uvw} & \text{if } a > b+c \\ g_{uvw} & \text{otherwise.} \end{cases} \tag{9}$$

The morphing process is such that the contribution of the triangle $\{u,v,w\}$ to the change in a, $\frac{d\ a}{dt}$, is the force $F(a;b,c)$. Intuitively, the force serves to reduce the gap. If a, b, and c satisfy the triangle in equality then no force is applied. If $a > b+c$ then a is reduced and if $b > c+a$ or $c > a+b$ then a is increased. Averaging over all triangles containing u, v gives our differential equation in Figure 2. For each 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 differential equation morphing h to a metric. The initial starting point is the input h.

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 5.1. *Let $g_{uvw}(t)$ denote the gap of h_t on the triplet $\{u,v,w\}$ in time t, as defined in (8). Then $\frac{dg_{uvw}(t)}{dt} \leq -3g_{uvw}(t)$ for all t.*

Note: Clearly the lemma implies that $g_{uvw}(t) \le g_{uvw}(t_0)e^{-3(t-t_0)}$ for any $t_0 \le t$. 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 $\{a(t) \ge b(t) + c(t)\} \cup \{b(t) \ge c(t) + a(t)\} \cup \{c(t) \ge a(t) + b(t)\}$. 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 $a(t) \ge b(t) + c(t)$ (hence $g_{uvw}(t) = a(t) - b(t) - c(t)$).

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

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 A. 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 $s \in V \setminus \{u, v, w\}$, $F(a(t); x_s(t), y_s(t)) - F(b(t); z_s(t), y_s(t)) - F(c(t); x_s(t), z_s(t)) \le 0$. 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 $a(0), b(0), c(0)$ violates the triangle inequality then at each moment $t > 0$ it either violates the same inequality or the violation disappears.

Lemma 5.2. *Let $a(t)$, $b(t)$, and $c(t)$ denote $h_t(u, v)$, $h_t(v, w)$, and $h_t(w, u)$ respectively. If $a(0) \ge b(0) + c(0)$ then for all $t \ge 0$ either $a(t) \ge b(t) + c(t)$ 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 $t \ge t_0$ in virtue of the note following Lemma 5.1. 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 $a(t) \le b(t) + c(t)$, or ∞ if no such t exists. If $t' = \infty$ then the lemma is proved. Otherwise by continuity and the first note above, $a(t') = b(t') + c(t')$, $b(t') \le a(t') + c(t')$ and $c(t') \le a(t') + b(t')$, 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 $\tau^* \in \mathcal{C}$ of V. 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 $C_{uvw}(t) := \sum_{u<v<w} L_t(u, v) + L_t(v, w) + L_t(w, u)$. The derivative of the cost is $\frac{d\ f(\tau^*,h_t)}{dt} = \frac{1}{n-2}\sum_{uvw} G_{uvw}(t)$, where

$$\begin{aligned}G_{uvw}(t) &:= [(1 - 2\tau^*(u, v))F(h_t(u, v); h_t(v, w), h_t(w, u)) + (1 - 2\tau^*(v, w))F(h_t(v, w); h_t(w, u), h_t(u, v)) \\ &\quad + (1 - 2\tau^*(w, u))F(h_t(w, u); h_t(u, v), h_t(v, w))] .\end{aligned}$$

(Note that G_{uvw} is *not* the derivative of C_{uvw}, but the sum is.) 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 $h_t(u, v) \ge h_t(v, w) + h_t(w, u)$ and so by the definition of F, $G_{uvw}(t) = [2\tau^*(u, v) - 2\tau^*(v, w) - 2\tau^*(w, u) + 1]g_{uvw}(t)$. Notice that $G_{uvw}(t) > g_{uvw}(t)$ can occur only if $\tau^*(u, v) = 1$, $\tau^*(v, w) = 0$ and $\tau^*(w, u) = 0$, a contradiction to $\tau^* \in \Delta$. Therefore $G_{uvw}(t) \le g_{uvw}(t)$ and by Lemma 5.1 $G_{uvw}(t) \le g_{uvw}(0)e^{-3t}$.

Lemma 5.3. *Set $\tau^* \in \mathcal{C}$. Given the above process, let $\hat{h} = \lim_{t\to\infty} h_t$. Then $f(\tau^*, \hat{h}) \le \frac{4}{3}f(\tau^*, h)$. Additionally, $\hat{h} \in \Delta$.*

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

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

It suffices to show that $g_{uvw}(0) \le C_{uvw}(0)$. Indeed, that would imply $C_{uvw}(0) + \frac{1}{3}g_{uvw}(0) \le \frac{4}{3}C_{uvw}(0)$. If $h_t(u, v), h_t(v, w), h_t(w, u)$ satisfy the triangle inequality at time $t = 0$ for some $u, v, w \in V$ then $g_{uvw}(0) = 0$ and the claim is trivial due to the nonnegativity of $C_{uvw}(0)$. Since h is integer, the only assignment for h which does not satisfy the triangle inequality is (w.l.o.g.) $h(u, v) = 1$, $h(v, w) = 0$, and $h(w, u) = 0$ which gives $g_{uvw}(0) = 1$. Such an assignment is inconsistent with any $\tau^* \in \Delta$ and thus $C_{uvw}(0) \ge 1$ as required. □

Lemma 5.3 immediately implies that $(x = \hat{h}, C = 4/3)$ is a feasible solution to the following LP', which is obtained from IP by allowing $x \in \Delta$:

$$\text{LP': minimize } C \text{ s.t.} \qquad f(\tau^*, x) \leq Cf(\tau^*, h) \text{ for all } \tau^* \in \mathcal{C}$$
$$x \in \Delta, C \geq 1 \ .$$

To prove Theorem 3.2, we need $f(\tau^*, x) \leq Cf(\tau^*, h)$ for all $\tau^* \in \Delta$, where Δ is a strict superset of $\mathcal{C}$. To see that this holds, observe that the proof of Lemma 5.3 is done by showing a 4/3 ratio locally for all $u, v, w \in V$ and for all choices of an integral τ^*. It turns out (proof omitted) that for $\tau^* \in \Delta$ and for all u, v, w the vector $(\tau^*(u, v), \tau^*(v, w), \tau^*(w, u))$ is a convex combination of the 6 ways to cluster the 3 elements u, v, w, and hence the 4/3 ratio is obtained with respect to τ^* as required. This completes the proof of Theorem 3.2.

6 QuickCluster with a Tweak

We prove Theorem 3.3. Let (x_{LP}, C_{LP}) be an optimal solution of LP for input h. 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 $\psi(a) = (6a - 1)/4$.

Note that Ailon [15] used a similar tweaking idea to improve rounding of a ranking LP in a traditional combinatorial optimization setting. Let $x_\psi(u, v)$ denote $\psi(X_{LP}(u, v))$. The algorithm (called QuickCluster$_\psi$) is as follows: Set all elements as *free*. Pick one element uniformly at random from the *free* elements, say u, to serve as a cluster center. For all $v \neq u$, add v to u with probability $\overline{x_\psi(u, v)}$ (setting it as not-free). Return the collection of clusters created during the execution as the solution. Recurse until there are no more *free* elements.

Let QC_ψ denote the distribution of the output clustering $x \in \mathcal{C}$ of QuickCluster$_\psi$. We need to show that for all $\tau^* \in \mathcal{C}$, $E_{x \in QC_\psi}[f(\tau^*, x)] \leq \frac{3}{2} f(\tau^*, x_{LP})$. The proof of this is analogous to the proof of Theorem 3.1. First we define $L_\psi(u, v)$, $C_\psi(u, v; w)$, $B_\psi(u, v, w)$, and $A^\psi_Z(u, v, w)$ exactly as $L(u, v)$, $C(u, v; w)$, $B(u, v, w)$, and $A^\psi_Z(u, v, w)$ under the substitution $h(u, v) \to x_\psi(u, v)$. Note that for L_ψ, C_ψ, B_ψ, and A^ψ_Z Lemma 4.1 and Lemma 4.2 both still hold. From the analog to Lemma 4.1 we get that:

$$f(\tau^*, x_{LP}) = \sum_{u<v} L(u, v) = \sum_{u<v} p_{uv} L(u, v) + \sum_{u<v<w} p_{uvw} A^\psi_L(u, v, w) \ ,$$

here $L(u, v) = x_{LP}(u, v)\overline{\tau^*(u, v)} + \overline{x_{LP}(u, v)}\tau^*(u, v)$ is the contribution of u, v to $f(\tau^*, x_{LP})$.[4] From the revised Lemma 4.2 we obtain $E_{x \in QC_\psi}[f(\tau^*, x)] = \sum_{u<v} p_{uv} L_\psi(u, v) + \sum_{u<v<w} p_{uvw} B_\psi(u, v, w)$.
Showing that $L_\psi(u, v) \leq \frac{3}{2} L(u, v)$ for all $u, v \in V$, and that $B_\psi(u, v, w) \leq A^\psi_L(u, v, w)$ for all $u, v, w \in V$ completes the proof. This entails breaking the polytope defining $(x_{LP}(u, v), x_{LP}(v, w), x_{LP}(w, u))$ into 27 smaller polytopes in which each x_{LP} is constrained to lay in $[0, 1/6]$, $(1/6, 5/6]$, or $(5/6, 1]$. On each of these smaller polytopes and for each one of 3 nonsymmetric possibilities for τ^* on u, v, w, the functions L, L_ψ are linear, and B_ψ and A^ψ_L are multinomials of total degree two and three respectively. 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 [16] for details.

7 Running Time

Running time with pairwise queries: We prove Theorem 3.4 using Yao's minimax Lemma [17]. 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 CorrelationClusterX makes $\Omega(n^2)$ queries into h in 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

[4] Not to be confused with prior definition of L.

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 3.5. We claim that given a stronger oracle, the expected running time of QuickCluster can be reduced to $O(n + \min_{\tau^* \in \mathcal{C}} f(\tau^*, h))$. 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 $T(QuickCluster, h) \leq O(\sum_{u_i} |N(u_i)|)$. Let us count $\sum_{u_i} |N(u_i)|$ in a different method. 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 $\sum_{u_i} |N(u)| \leq n + f(x, h)$. Moreover, due to the triangle inequality on the function f and Theorem 3.1, $E[f(x, h)] \leq E[f(x, \tau^\dagger)] + f(\tau^\dagger, h) \leq 3[f(\tau^\dagger, h)]$ for all $\tau^\dagger$. Choosing $\tau^\dagger = \arg\min_{\tau \in \mathcal{C}} f(\tau, h)$ yields $E[T(QuickCluster, h)] \leq O(n + \min_{\tau \in \mathcal{C}} f(\tau, h))$ 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.

8 Randomness is Necessary

It is clear by definitions that randomized CorrelationClusterX can achieve a bound which is at least as good as deterministic CorrelationClusterX. Here we show a simple case that illustrates that the gap between the two cases is nonzero. Consider an input to CorrelationClusterX 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 choise 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 3.6.

9 Future Work

- 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 CorrelationClusterX.

- Finding a specific instance h for which QuickCluster achieves the 2 approximation bound for CorrelationClusterX 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] William W. Cohen, Robert E. Schapire, and Yoram Singer. Learning to order things. *J. Artif. Intell. Res. (JAIR)*, 10:243–270, 1999.

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

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

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

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 (9).*

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

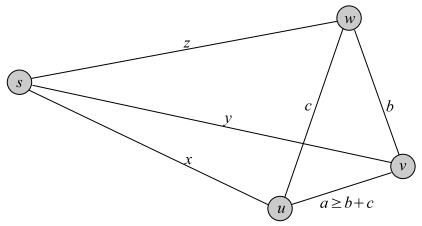

Figure 3: The tetrahedral structure of the triangle $\{u,v,w\}$ and the three triangles adjacent to it which contain s. Lemma A.1 claims that the sum of forces applied by the triangles $\{u,v,s\}$, $\{v,w,s\}$ and $\{w,u,s\}$ cannot act to increase the gap g_{uvw}.

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.

□