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

Nir Ailon[1] and Edo Liberty[2]

[1] Google Research. nailon@google.com
[2] Yale University and Google Research. edo.liberty@yale.edu

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. In this setting, we argue that it is more reasonable to measure the output clustering's accuracy against the unknown underlying true clustering.
We present two main results. The first is a novel method for continuously morphing a general (non-metric) function into a pseudometric. This technique may be useful for other metric embedding and clustering problems. The second is a simple algorithm for randomly rounding a pseudometric into a clustering. Combining the two, we obtain a certificate for the possibility of getting a solution of factor strictly less than 2 for our problem. This approximation coefficient could not have been achieved by considering the agnostic version of the problem unless $P = NP$.

1 Introduction

Correlation Clustering was defined by Bansal, Blum and Chawla [9] as the problem of producing a clustering x of data points based on a binary function, h, which tells us, for each pair, whether they are similar or not. The objective is to find the clustering x that minimizes $f(x, h)$, the number of disagreements between h and x. The problem is *agnostic* in the sense that the clustering of the data is not taken into account or even assumed to exist. This gives rise to an APX-hard optimization problem which is studied in their paper and in consequent work [9, 3, 10, 12, 11, 15, 2, 14, 13, 17]. In this paper we assume a setting in which *there is* an (unknown) correct way to cluster the data, τ. Such a scenario arises, for example, in duplicate detection and elimination in large data (also known as the record linkage). In this setting we argue that one should try to minimize $f(x, \tau)$, the number of disagreements between the output clustering and the ground truth clustering.

A related problem that has been studied in the literature [16] is *planted clustering*. In this model, the observation h is given by random noise applied to the ground truth clustering τ. Solving the traditional Correlation Clustering problem on h, thus obtained, gives precisely a *maximum likelihood* configuration for τ. 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, 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 [16] or dot product techniques [7][3]) 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 approximation guarantees.

In our case, the ground truth clustering τ is not only unknown but can also be arbitrarily different from the similarity function h. Since the algorithm can only access h, we can expect the output x to respect the ground truth only insofar as the input h does. We thus try to minimize C such that $f(x, \tau) \leq Cf(h, \tau)$ where f measures the distance between the different objects. In other words, the more h disagrees with the ground truth τ (larger $f(h, \tau)$) the weaker the requirements from the output x.

Traditional optimization gives the following indirect solution to our problem: Find x which approximately minimizes $f(x, h)$ so $f(x, h) \leq C^* f(h, \tau)$ for some $C^* \geq 1$ and for all possible clusterings τ. 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. Since $C^* > 1$ (Correlation Clustering is APX-hard) this approach would yield $C > 2$. 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, Section 3, is a morphing process which proves the existence of a good relaxed solution to our problem, which we name CorrelationClusteringX. More precisely, one can continuously change the values of the input h into a "soft" clustering x_{dif} which is $[0, 1]$ valued and a metric (satisfies all triangle inequalities). More importantly, we show that $f(x_{dif}, \tau) \leq 4/3 f(h, \tau)$. The relaxation x_{dif} is obtained as the limit at infinity of a solution to a piecewise linear differential equation. This algorithm, which we also refer to as a *morphing process*, is interesting in its own right and may be useful for other problems on

[3] In [7], a Gaussian random noise model is assumed.

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 shows that all triangle inequality violations decay exponentially in time. This fast decay allows us to bound the loss with respect to the ground truth from above.

In Section 4 we show how to randomly convert the relaxed solution x_{dif} into an integer solution x to CorrelationClusteringX such that $f(x, \tau) \leq 3/2 f(x_{dif}, \tau)$. Applying the rounding algorithm to x_{dif} gives a $C \leq 2$ approximation algorithm. 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 [6] on Machine Learning reductions for ranking. Balcan et al. [8] 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 observation, 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. 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.

We dedicate Section 2 to formally defining our notation and stating our results. Some proofs were omitted due to space limitations and can be found in [4]. The interested reader is also referred to the last reference for further discussion.

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.[4]

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.

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 $a \in [0, 1]$

[4] 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 pseudometric over the values {0,1}.

let $\overline{a}$ denote $1-a$. Define the Correlation Clustering cost function [9] $f : \mathcal{K} \times \mathcal{K} \rightarrow \mathbb{R}^+$ 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.
The problem of CorrelationClusteringX is given in the following:

Definition 1. *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 $x \in \mathcal{C}$ with an exponential number of constraints:

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

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 $x \in \varDelta$ and adding a constraint for each $\tau \in \varDelta$.

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

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

Observation 1 *LP has a separation oracle and can therefore be solved optimally in polynomial time.*

To see Observation 1, note that given a candidate solution (x, C) it is possible to find $\tau \in \varDelta$ satisfying $f(\tau, x) > Cf(\tau, h)$ (if one exists) using another simple standard linear program with $\tau \in \varDelta$ 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 $\tau \in \varDelta$). 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 1. *For any $h \in I$, the value of LP is at most $4/3$.*

In the proof of Theorem 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 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. *For any $\hat{h} \in \varDelta$ 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 1 and Theorem 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 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 following theorems are proved in [4].

The running time of QuickCluster is analyzed for two representation dependent regimes. In the pairwise-queries model, only pairwise queries to h are allowed, i.e, evaluating $h(u,v)$ for a pair $\{u,v\}$. In the neighborhood-queries regime, the algorithm is allowed neighborhood queries, returning for a query u its neighborhood $N(u) = \{u\} \cup \{v \in V \mid h(u,v) = 0\}$ as a linked list. We obtain the following bounds.

Theorem 4. *In the pairwise-queries model, any constant factor randomized approximation algorithm for CorrelationClusteringX performs $\varOmega(n^2)$ queries to h in expectation for some input h.*

Trivially, QuickCluster performers $O(n^2)$ queries to h for any input h and thus performs optimally for the inputs h in Theorem 4.

Theorem 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 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 1. The idea is to "morph" $h \in \mathcal{K}$, which is not necessarily a metric, into a pseudometric. The solution $x_{dif} \in \varDelta$ 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 $h(u, v)$ 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, $\frac{d\,a}{dt}$, is the force $F(a; b, c)$. Intuitively, the force serves to reduce the gap. If a, b, and c satisfy the triangle inequality then no force is applied. If $a > b + c$ then $\frac{d\,a}{dt}$ is negative and a is reduced. If $b > c + a$ or $c > a + b$ then $\frac{d\,a}{dt}$ is positive a is increased. Averaging over all triangles containing u and v gives our differential equation in Figure 1. With the starting boundary condition

$$\boxed{\frac{dh_t(u,v)}{dt} = \sum_{w\in V\setminus\{u,v\}} F(h_t(u, v); h_t(v, w), h_t(w, u)).}$$

Fig. 1. 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 $x_{dif} = \lim_{t\to\infty} h_t$

$h_0(u, v) = h(u, v)\ \ \forall u, v \in V$. Similar to our previous notation, let $a(t)$, $b(t)$ and $c(t)$ denote $h_t(u, v), h_t(v, w)$ and $h_t(w, u)$ throughout.

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 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 $\frac{dg_{uvw}(t)}{dt} \le -3g_{uvw}(t)$ 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 $g_{uvw}(t) = a(t) - b(t) - c(t)$).

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

where $x_s(t) = h_t(u,s)$, $y_s(t) = h_t(v,s)$, and $z_s(t) = h_t(w,s)$. 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 [4] appendix-A.

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 is resolved.

Lemma 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 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 $\tau \in \mathit{\Delta}$. 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 $\frac{d\ f(\tau,h_t)}{dt} = \frac{1}{n-2}\sum_{uvw} G_{uvw}(t)$[5], where

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

The cost at time t is $f(\tau,h_t) = \frac{1}{n-2}\sum_{uvw} C_{uvw}(0) + \frac{1}{n-2}\sum_{uvw}\int_0^t G_{uvw}(s)ds$. We concentrate on the contribution of one triangle to this sum: $H_{uvw}(t) = C_{uvw}(0) + \int_0^t G_{uvw}(s)ds$.

[5] Note that G_{uvw} is *not* the derivative of C_{uvw}, but the sum $\sum_{uvw} G_{uvw}$ is the derivative of $\sum_{uvw} C_{uvw}$.

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 1 $G_{uvw}(t) \leq g_{uvw}(0)e^{-3t}$.

Lemma 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 $\int_0^\infty G_{uvw}(t)dt \leq \int_0^\infty g_{uvw}(0)e^{-3t}dt \leq \frac{1}{3} g_{uvw}(0)$.

$$\begin{aligned} f(x_{dif}, \tau) &= \frac{1}{n-2} \lim_{t\to\infty} \sum_{u<v<w} H_{uvw}(t) = \frac{1}{n-2} \sum_{u<v<w} C_{uvw}(0) + \int_0^\infty G_{uvw}(t) \\ &\leq \frac{1}{n-2} \sum_{u<v<w} C_{uvw}(0) + \frac{1}{3} g_{uvw}(0) \leq \frac{1}{n-2} \sum_{u<v<w} \frac{4}{3} C_{uvw}(0) \leq \frac{4}{3} f(h, \tau) \end{aligned}$$

The last equation relies on the fact 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 immediately implies that $(x_{dif} = \lim_{t\to\infty} h_t, C_{dif} = 4/3)$ is a feasible solution to LP.

4 QuickCluster

We prove Theorem 2 and Theorem 3. The QuickCluster algorithm described here is very similar to the one used in [3] 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 $\psi(u,v)$ (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 [3]. Also, Ailon [1] 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 $QC(h)$ be the distribution over outputs produced by QuickCluster for input h. By definition of f and the fact that τ is fixed we have that $E_{x\sim QC(h)}[f(x,\tau)] = \sum_{u<v} E_{x\sim QC(h)}[x(u,v)]\overline{\tau(u,v)} + E_{x\sim QC(h)}[\overline{x(u,v)}]\tau(u,v)$. Since each $x(u,v)$ is a binary random variable 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 $E_{x\sim QC(h)}[\overline{x(u,v)}]$ 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. 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 $\{u,v,w\} \in \binom{V}{3}$.

Lemma 4. *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 $E_{x\sim QC(h)}[\tau(u,v)\overline{x(u,v)} + \overline{\tau(u,v)}x(u,v)] = p_{uv}L_\psi(u,v) + \sum_{w\neq u,v}\frac{1}{3}p_{uvw}\beta(u,v;w)$.

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

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 5. *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 5 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 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 $h \in \mathit{\Delta}$ we have that $L_\psi(u, v)/L_h(u, v) \leq 6/5$ and that $B(u, v, w)/A_{L_h}(u, v, w) \leq 3/2$. 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.[6] 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 [5] for details. This proves Theorem 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 3.

5 Short discussion

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 [3], raising the question of whether it is possible to obtain a 1.5 approximation for CorrelationClusteringX.

Finding a specific instance h for which our algorithm 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).

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

References

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

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

2. 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.
3. 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.
4. Nir Ailon and Edo Liberty. Correlation clustering revisited: The "true" cost of error minimization problems. *Yale University Tecnical Report 1214*, 2008.
5. Nir Ailon and Edo Liberty. Mathematica program, 2008. `http://www.cs.yale.edu/homes/el327/public/prove32/`.
6. 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.
7. J. Aslam, A. Leblanc, and C. Stein. A new approach to clustering. In *4th International Workshop on Algorithm Engineering*, 2000.
8. Maria-Florina Balcan, Avrim Blum, and Anupam Gupta. Approximate clustering without the approximation. In *SODA'09*, New York, NY, 2009.
9. 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.
10. 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.
11. 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.
12. 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.
13. 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.
14. Aristides Gionis, Heikki Mannila, and Panayiotis Tsaparas. Clustering aggregation. In *Proceedings of the 21st International Conference on Data Engineering (ICDE)*, Tokyo, 2005. To appear.
15. 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.
16. 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.
17. Alexander Strehl. Relationship-based clustering and cluster ensembles for high-dimensional data mining. *PhD Dissertation, University of Texas at Austin*, May 2002.