Approximate kernel clustering

Extended abstract*

Subhash Khot†
Courant Institute of Mathematical Sciences
khot@cims.nyu.edu

Assaf Naor‡
Courant Institute of Mathematical Sciences
naor@cims.nyu.edu

Abstract

In the kernel clustering problem we are given a large $n \times n$ positive semi-definite matrix $A = (a_{ij})$ with $\sum_{i,j=1}^n a_{ij} = 0$ and a small $k \times k$ positive semi-definite matrix $B = (b_{ij})$. The goal is to find a partition $S_1, \ldots, S_k$ of $\{1, \ldots n\}$ which maximizes the quantity

$$\sum_{i,j=1}^{k} \left(\sum_{(i,j) \in S_i \times S_j} a_{ij} \right) b_{ij}.$$

We study the computational complexity of this generic clustering problem which originates in the theory of machine learning. We design a constant factor polynomial time approximation algorithm for this problem, answering a question posed by Song, Smola, Gretton and Borgwardt. In some cases we manage to compute the sharp approximation threshold for this problem assuming the Unique Games Conjecture (UGC). In particular, when B is the 3×3 identity matrix the UGC hardness threshold of this problem is exactly $\frac{16\pi}{27}$. We present and study a geometric conjecture of independent interest which we show would imply that the UGC threshold when B is the $k \times k$ identity matrix is $\frac{8\pi}{9}\left(1 - \frac{1}{k}\right)$ for every $k \geq 3$.

1. Introduction

This paper is devoted to an investigation of the polynomial time approximability of a generic clustering problem which originates in the theory of machine learning. In doing so, we uncover a connection with a continuous geometric/analytic problem which is of independent interest. In [21] Song, Smola, Gretton and Borgwardt introduced the following framework for *kernel clustering problems*. Assume that we are given a centered kernel, i.e. an $n \times n$ positive semidefinite matrix $A = (a_{ij})$ with real entries such that $\sum_{i,j=1}^n a_{ij} = 0$ (the assumption that the kernel is centered is a commonly used normalization in learning theory—see [20] for more information on this topic). Such matrices arise, for example, as correlation matrices of random variables $(X_1, \ldots, X_n)$ that measure attributes of certain empirical data, i.e. $a_{ij} = \mathbb{E}[X_i X_j]$. We think of n as very large, and our goal is to "cluster" the matrix A to a much smaller $k \times k$ matrix in such a way that certain features could still be extracted from the clustered matrix. Formally, given a partition of $\{1, \ldots, n\}$ into k sets $S_1, \ldots, S_k$, define the clustering of A with respect to this partition to be the $k \times k$ matrix $A(S_1, \ldots, S_k)$ whose $(i,j)^{\text{th}}$ entry is $\sum_{(i,j) \in S_i \times S_j} a_{ij}$.

In the kernel clustering problem, we are given a positive semidefinite $k \times k$ matrix $B = (b_{ij})$, and we wish to find the clustering $A(S_1, \ldots, S_k) = C = (c_{ij})$ of A, which is most similar to B in the sense that $\sum_{i,j=1}^k c_{ij} b_{ij}$, i.e its scalar product with B, is as large as possible. In other words, our goal is to compute the number (and the corresponding partition):

$$\begin{aligned} \mathbf{Clust}(A|B) := \max \Bigg\{ & \sum_{i,j=1}^{k} \left(\sum_{(i,j) \in S_i \times S_j} a_{ij} \right) b_{ij} : \\ & \{S_1, \ldots, S_k\} \text{ is a partition of } \{1, \ldots, n\} \Bigg\} \qquad (1) \\ = \max \Bigg\{ & \sum_{i,j=1}^{k} A(S_1, \ldots, S_k)_{ij} \cdot b_{ij} : \\ & \{S_1, \ldots, S_k\} \text{ is a partition of } \{1, \ldots, n\} \Bigg\} \\ = \max \Bigg\{ & \sum_{i,j=1}^{n} a_{ij} b_{\sigma(i)\sigma(j)} : \sigma : \{1, \ldots, n\} \to \{1, \ldots, k\} \Bigg\}. \end{aligned}$$

*The full version of this paper is available at http://www.cims.nyu.edu/∼naor/homepage%20files/kernel.pdf

†Research supported in part by NSF CARREER award CCF-0643626, and a Microsoft New Faculty Fellowship.

‡Research supported by NSF grants CCF-0635078 and DMS-0528387.

The flexibility in the above formulation of the kernel clustering problem is clearly in the choice of comparison matrix B, which allows us to enforce a wide-range of clustering criteria. Using the statistical interpretation of (a_{ij}) as a correlation matrix, we can think of the matrix B as encoding our belief/hypothesis that the empirical data has a certain structure, and the kernel clustering problem aims to efficiently expose this structure.

Several explicit examples of useful "test matrices" B are discussed in [21], including hierarchical clustering and clustering data on certain manifolds. We refer to [21] for additional information which illustrates the versatility of this general clustering problem, including its relation to the Hilbert Schmidt Independence Criterion (HSIC) and various experimental results. In [21] it was asked if there is a polynomial time approximation algorithm for computing $\mathbf{Clust}(A|B)$. Here we obtain a constant factor approximation algorithm for this problem, and prove some computational hardness of approximation results.

Before stating our results in full generality we shall now present a few simple illustrative examples. If $B = I_k$ is the $k \times k$ identity matrix, then thinking once more of a_{ij} as correlations $\mathbb{E}[X_i X_j]$, our goal is to find a partition $S_1, \ldots, S_k$ of $\{1, \ldots, n\}$ which maximizes the quantity

$$\sum_{i=1}^{k} \sum_{p,q \in S_i} \mathbb{E}[X_p X_q],$$

i.e. we wish to cluster the variables so as to maximize the total intra-cluster correlations. As we shall see below, our results yield a polynomial time algorithm which approximates $\mathbf{Clust}(A|I_k)$ up to a factor of $\frac{8\pi}{9}\left(1 - \frac{1}{k}\right)$. In particular, when $k = 3$ we obtain a $\frac{16\pi}{27}$ approximation algorithm, and we show that assuming the Unique Games Conjecture (UGC) no polynomial time algorithm can achieve an approximation guarantee which is smaller than $\frac{16\pi}{27}$. The Unique Games Conjecture was posed by Khot in [13], and it will be described momentarily. For the readers who are not familiar with this computational hypothesis and its remarkable applications to hardness of approximation, it suffices to say that this hardness result should be viewed as strong evidence that $\frac{16\pi}{27}$ is the sharp threshold below which no polynomial time algorithm can solve the kernel clustering problem when $B = I_3$. Moreover, we conjecture that $\frac{8\pi}{9}\left(1 - \frac{1}{k}\right)$ is the sharp approximability threshold (assuming UGC) for $\mathbf{Clust}(A|I_k)$ for every $k \geq 3$. In this paper, we reduce this conjecture to a purely geometric/analytic conjecture, which we will describe in detail later, and prove some partial results about it.

Another illustrative example of the kernel clustering problem is the case

$$B = \begin{pmatrix} 1 & -1 \\ -1 & 1 \end{pmatrix}.$$

In this case, we clearly have

$$\mathbf{Clust}\left(A \middle| \begin{pmatrix} 1 & -1 \\ -1 & 1 \end{pmatrix}\right) = \max\left\{ \sum_{i,j=1}^{n} : a_{ij}\varepsilon_i\varepsilon_j : \varepsilon_1, \ldots, \varepsilon_n \in \{-1, 1\} \right\}. \quad (2)$$

The optimization problem in (2) is well known as the positive semi-definite Grothendieck problem and has several algorithmic applications (see [19, 17, 2, 5]). It has been shown by Rietz [19] that the natural semidefinite relaxation of (2) has integrality gap $\frac{\pi}{2}$ (see also Nesterov's work [17]). Our results imply that assuming the UGC $\frac{\pi}{2}$ is the sharp approximation threshold for the positive-semidefinite Grothendieck problem. Note that without the assumption that A is positive semidefinite the natural semidefinite relaxation of (2) has integrality gap $\Theta(\log n)$. See [16, 6, 1] for more information, and [3] for hardness results for this problem.

We can also view the problem (2) as a generalization of the MaxCut problem. Indeed, let $G = (V = \{1, \ldots, n\}, E)$ be an n-vertex loop-free graph. For every vertex $i \in V$ let d_i denote its degree in G. Let A be the Laplacian of G, i.e. A is the $n \times n$ matrix given by

$$a_{ij} = \begin{cases} d_i & \text{if } i = j, \\ -1 & \text{if } i \neq j \ \wedge \ ij \in E, \\ 0 & \text{if } i \neq j \ \wedge \ ij \notin E. \end{cases} \quad (3)$$

Then A is positive semi-definite since it is diagonally dominant. For every $\varepsilon_1, \ldots, \varepsilon_n \in \{-1, 1\}$ let $S \subseteq V$ be the set $S{:=}\{i \in V : \varepsilon_i = 1\}$. Then:

$$\begin{aligned} & \sum_{i,j=1}^{n} a_{ij}\varepsilon_i\varepsilon_j \\ = \ & \sum_{i=1}^{n} d_i - 2|E(S,S)| - 2|E(V \setminus S, V \setminus S)| \\ & +2|E(S, V \setminus S)| \\ = \ & 2|E| - 2\left(|E| - |E(S, V \setminus S)|\right) + 2|E(S, V \setminus S)| \\ = \ & 4|E(S, V \setminus S)|. \end{aligned} \quad (4)$$

Hence

$$\mathbf{Clust}\left(A \middle| \begin{pmatrix} 1 & -1 \\ -1 & 1 \end{pmatrix}\right) = 4\mathrm{MaxCut}(G).$$

Using Hastad's inapproximability result for MaxCut [11] it follows that if $P \neq NP$ there is no polynomial time algorithm which approximates (2) up to a factor smaller than $\frac{17}{16}$.

Our algorithmic results. For a fixed positive semidefinite matrix B, the approximability threshold for the problem of

computing $\mathbf{Clust}(A|B)$ depends on B. It is therefore of interest to study the performance of our algorithms in terms of the matrix B. We do obtain bounds which depend on B (which are probably suboptimal in general)—the precise statements are contained in Theorem 1 and Theorem 3. For the sake of simplicity, in the introduction we state bounds which are independent of B. We believe that the problem of computing the approximation threshold (perhaps under UGC) for each fixed B is an interesting problem which deserves further research.

If A is centered, i.e. $\sum_{i,j=1}^{n} a_{ij} = 0$, then for every $k \times k$ positive semi-definite matrix B our algorithm achieves an approximation ratio of $\pi\left(1 - \frac{1}{k}\right)$. If, in addition, B is centered and spherical, i.e. $\sum_{i,j=1}^{k} b_{ij} = 0$ and $b_{ii} = 1$ for all i, then our algorithm achieves an approximation ratio of $\frac{8\pi}{9}\left(1 - \frac{1}{k}\right)$. This ratio is also valid if B is the identity matrix, and as we mentioned above, we believe that this approximation guarantee cannot be improved assuming the UGC (and here we prove this conjecture for $k = 3$). When A is not necessarily centered (note that this case is of lesser interest in terms of the applications in machine learning) we obtain an algorithm which achieves an approximation ratio of $1 + \frac{3\pi}{2}$ (this is probably sub-optimal). All of our algorithms, which are described in Section 2, use semi-definite programming in a perhaps non-obvious way.

The Unique Games Conjecture, hardness of approximation, and the propeller problem. Our hardness result for kernel clustering problem is based on the Unique Games Conjecture which was put forth by Khot in [13]. We shall now describe this conjecture. A *Unique Game* is an optimization problem with an instance $\mathscr{L} = \mathscr{L}(G(V, W, E), n, \{\pi_{vw}\}_{(v,w)\in W})$. Here $G(V, W, E)$ is a regular bipartite graph with vertex sets V and W and edge set E. Each vertex is supposed to receive a label from the set $\{1, \ldots, n\}$. For every edge $(v, w) \in E$ with $v \in V$ and $w \in W$, there is a given permutation $\pi_{vw} : \{1, \ldots, n\} \to \{1, \ldots, n\}$. A labeling to the Unique Game instance is an assignment $\rho : V \cup W \to \{1, \ldots, n\}$. An edge (v, w) is satisfied by a labeling ρ if and only if $\rho(v) = \pi_{vw}(\rho(w))$. The goal is to find a labeling that maximizes the fraction of edges satisfied (call this maximum $\mathrm{OPT}(\mathscr{L})$). We think of the number of labels n as a constant and the size of the graph $G(V, W, E)$ as the size of the problem instance.

The Unique Games Conjecture asserts that for arbitrarily small constants $\varepsilon, \delta > 0$, there exists a constant $n = n(\varepsilon, \delta)$ such that no polynomial time algorithm can distinguish whether a Unique Games instance $\mathscr{L}$ with n labels satisfies $\mathrm{OPT}(\mathscr{L}) \geq 1 - \varepsilon$ or $\mathrm{OPT}(\mathscr{L}) \leq \delta$[1]. This conjecture is (by now) a commonly used complexity assumption to prove hardness of approximation results. Despite several recent attempts to get better polynomial time approximation algorithms for the Unique Game problem (see the table in [4] for a description of known results), the unique games conjecture still stands.

Our UGC hardness result for kernel clustering is based at heart on the "dictatorship vs. low-influence" paradigm that is recurrent in UGC hardness results (for example [13, 15]). In order to apply this paradigm one usually designs a probabilistic test on a given Boolean function on the Boolean hypercube and then analyzes the acceptance probability of this test in the two extremes of dictatorship functions and functions without influential variables. The gap between these two acceptance probabilities translates into the hardness of approximation factor. In our case, instead of a probabilistic test we need to design a positive semidefinite quadratic form on the truth table of the function. Our form is the sum of the squares of the Fourier coefficients of level 1. This already yields $\frac{\pi}{2}$ UGC hardness when $k = 2$. For larger k we need to work with functions from $\{1, \ldots, k\}^n$ to $\{1, \ldots, k\}$. The analysis of this approach leads to the "propeller problem" which we now describe.

We believe that one of the interesting aspects of the present paper is that complexity considerations lead to geometric/analytic problems which are of independent interest. Similar such connections have been recently discovered in [14, 8]. In our case the reduction from UGC to kernel clusterings leads to the following question, which we call the "propeller problem" for reasons that will become clear presently. Let γ_{k-1} denote the standard Gaussian measure on $\mathbb{R}^{k-1}$, i.e. the density of γ_{k-1} is $(2\pi)^{-(k-1)/2} e^{-\|x\|_2^2/2}$. Let $A_1, \ldots, A_k$ be a partition of $\mathbb{R}^{k-1}$ into measurable sets. For each $i \in \{1, \ldots, k\}$ consider the Gaussian moment of the set A_i, i.e. the vector

$$z_i := \int_{A_i} x d\gamma_{k-1}(x) \in \mathbb{R}^{k-1}.$$

Our goal is to find the partition which maximizes the sum of the squared Euclidean lengths of the Gaussian moments of the elements of the partition, i.e. $\sum_{i=1}^{k} \|z_i\|_2^2$. Let $C(k)$ denote the value of this maximum (we prove that this is indeed a maximum and not just a supremum). We show that assuming the UGC there is no polynomial time algorithm which approximates $\mathbf{Clust}(A|I_k)$ to a factor smaller than $\frac{1-1/k}{C(k)}$. We show that $C(2) = \frac{2}{\pi}$ and $C(3) = \frac{9}{8\pi}$. The value of $C(3)$ comes from the partition of the plane $\mathbb{R}^2$ into a "propeller", i.e. three cones of angle $\frac{2\pi}{3}$ with cusp at the origin. We also show that $C(k)$ is attained at a *simplicial conical partition*, i.e. a partition $A_1, \ldots, A_k$ of $\mathbb{R}^{k-1}$ of the following form: let $A_1, \ldots, A_m$ be the elements of the partition which are non-empty. Then $A_j = B_j \times \mathbb{R}^{k-m}$ where $B_j \subseteq \mathbb{R}^{m-1}$ is a cone with cusp at 0 whose base is a simplex. We conjecture that the optimal partition of this type for every $k \geq 3$ is actually $\{C_1 \times \mathbb{R}^{k-3}, C_2 \times \mathbb{R}^{k-3}, C_3 \times \mathbb{R}^{k-3}\}$, where

[1] As stated in [13], the conjecture says that it is NP-hard to distinguish between these two cases. However if one only wants to rule out polynomial time algorithms, the conjecture as stated here suffices.

$\{C_1, C_2, C_3\}$ is the propeller partition of $\mathbb{R}^2$ —see Figure 1. If so then it would follow that the approximation algorithms described above are optimal assuming the UGC for every $k \geq 4$, and not just for $k \in \{2, 3\}$. We give the following evidence for this conjecture: it is tempting to believe that the optimal simplicial conical partition described above occurs when the cones $B_1, \ldots, B_m$ are generated by the regular simplex. However, we show that among such *regular simplicial conical partitions* the one which maximizes the sum of the squared lengths of its Gaussian moments is when $m = 3$. The full propeller conjecture seems to be a challenging geometric problem of independent interest, not just due to the connection that we establish between it and the study of hardness of approximation for kernel clustering.

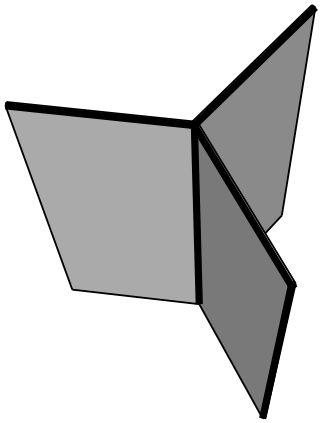

Figure 1. *The conjectured optimal partition for the "sum of squares of Gaussian moments problem" described above consists of a partition of R^{k-1} into 3 parts, and the remaining $k-3$ parts are empty. This partition corresponds to a planar 120° "propeller" multiplied by an orthogonal copy of $\mathbb{R}^{k-3}$.*

We end this introduction with an explanation of how our work relates to the recent result of Raghavendra [18] which shows that for any generalized constraint satisfaction problem[2] (CSP) there is a generic way of writing a semidefinate relaxation that achieves an optimal approximation ratio assuming the Unique Games Conjecture. Our clustering problem fits in the framework of [18] as follows: we wish to compute

$$\max \left\{ \sum_{i,j=1}^{n} a_{ij} b_{\sigma(i)\sigma(j)} : \sigma : \{1, \ldots, n\} \to \{1, \ldots, k\} \right\}, \tag{5}$$

where (a_{ij}) is a centered positive semi-definite matrix and (b_{ij}) is a positive semi-definite matrix. One can think of this problem as a CSP (with an extra global constraint corresponding to the positive semi-definiteness) where the set of variables is $\{1, \ldots, n\}$ and we wish to assign each variable a value from the domain $\{1, \ldots, k\}$. For every pair $(i, j) \in \{1, \ldots, n\} \times \{1, \ldots, n\}$, there is a constraint with weight a_{ij}. We get a payoff of b_{st} if variables i and j are assigned $s \in \{1, \ldots, k\}$ and $t \in \{1, \ldots, k\}$ respectively.

Raghavendra shows that every integrality gap instance for his generic SDP relaxation can be translated into a UGC-hardness of approximation result with the hardness factor (essentially) the same as the integrality gap. We make here the non-trivial observation that in the reduction of [18], starting with an integrality gap instance for (the generic SDP relaxation of) the clustering problem (5), the matrix of the constraint weights (a_{ij}) indeed turns out to be positive semi-definite as required in the kernel clustering problem (this requires proof—the details are omitted since this is a digression from the topic of this paper). Thus Raghavendra's result can be made to apply to the kernel clustering problem (i.e. the generic SDP achieves the optimum approximation ratio assuming UGC).

Nevertheless, it is also useful to look at different relaxations and rounding procedures for the following reasons. Firstly, for a given problem there could be an SDP relaxation that is more *natural* than the generic one and might be easier to work with. Secondly, Raghavendra's result (that the integrality gap is same as the hardness factor) applies only when the integrality gap is a constant. This is a priori not clear for the kernel clustering problem. For instance, a priori the integrality gap could be $\Omega(\log n)$ (as is the case for Grothendieck problem on a general graph—see [1]). So before applying the result of [18], one would need to show that the integrality gap of the generic SDP is indeed a constant. Thirdly, for CSPs with negative payoffs (as is the case in the kernel clustering problem), Raghavendra shows that the *value* computed by the generic SDP achieves the optimal approximation ratio (modulo UGC), but the paper does not give a rounding procedure. Finally, Raghavendra's result does not really shed light on the exact hardness threshold in the sense that it shows how to translate integrality gap instances into a UGC hardness result, but gives no idea as to how to construct an integrality gap instance in the first place. Constructing the integrality gap instance in general amounts to answering certain isoperimetric type geometric question (naturally leading to a dictatorship test, or the other way round. In other words, the geometric question itself

[2] In a generalized CSP, every assignment to variables in a constraint has a real-valued (possibly negative) pay-off instead of a simple decision saying that the assignment is a satisfying assignment or not.

might be inspired by the dictatorship test that we have in mind). Thus as far as we know, we cannot avoid designing an explicit dictatorship test and answering an isoperimetric type question, whether or not we start with Raghavendra's generic SDP that is guaranteed to be optimal. As mentioned before, in the clustering problem where $B = (b_{st})$ is centered and spherical, we show that the UGC-hardness threshold is at least $\frac{1-1/k}{C(k)}$ and characterizing $C(k)$ seems to be a challenging geometric question.

For lack of space in this extended abstract we will only describe our algorithmic results. The UGC hardness and the details of the connection with the propeller problem are presented in the full version of this paper which available at http://www.cims.nyu.edu/∼naor/homepage%20files/kernel.pdf.

2 Constant factor approximation algorithms for kernel clustering

Let $A \in M_n(\mathbb{R})$ and $B \in M_k(\mathbb{R})$ be positive semidefinite matrices. Then there are $u_1, \ldots, u_n \in \mathbb{R}^n$ and $v_1, \ldots, v_k \in \mathbb{R}^k$ such that $a_{ij} = \langle u_i, u_j \rangle$ and $b_{ij} = \langle v_i, v_j \rangle$. Such vectors can be found in polynomial time (this is simply the Cholesky decomposition). The instance of the kernel clustering problem will be called centered if $\sum_{i,j=1}^n a_{ij} = 0$, or equivalently $\sum_{i=1}^n u_i = 0$. The instance will be called spherical if $b_{ii} = 1 = \|v_i\|_2^2$ for all $i \in \{1, \ldots, k\}$. Let $R(B)$ be the radius of the smallest Euclidean ball containing $\{v_1, \ldots, v_k\}$. Note that $R(B)$ is indeed only a function of B, i.e. it does not depend on the particular representation of B as a Gram matrix. Moreover, it is possible to compute $R(B)$, and given the decomposition $b_{ij} = \langle v_i, v_j \rangle$ a vector $w \in \mathbb{R}^k$ such that $\max_{j \in \{1,\ldots,k\}} \|v_j - w\|_2 = R(B)$, in polynomial time (see [10]).

Our goal is to compute in polynomial time the quantity:

$$\begin{aligned} \mathbf{Clust}(A|B) &:= \max_{\sigma:\{1,\ldots,n\}\to\{1,\ldots,k\}} \sum_{i,j=1}^n a_{ij} b_{\sigma(i)\sigma(j)} \\ &= \max_{\sigma:\{1,\ldots,n\}\to\{1,\ldots,k\}} \sum_{i,j=1}^n \langle u_i, u_j \rangle \langle v_{\sigma(i)}, v_{\sigma(j)} \rangle \end{aligned}$$

Our algorithm, which is based on semidefinite programming, proceeds via the following steps:

1. Compute a Cholesky decomposition of B, i.e. $v_1, \ldots, v_k \in \mathbb{R}^k$ with $b_{ij} = \langle v_i, v_j \rangle$.

2. Compute (using for example [10]) $R(B)$ and a vector $w \in \mathbb{R}^k$ such that

$$\max_{j \in \{1,\ldots,k\}} \|v_j - w\|_2 = R(B).$$

3. Solve the semidefinite program

$$\max \left\{ \sum_{i,j=1}^n a_{ij} \cdot \langle \|w\|_2 u + R(B) x_i,\ \|w\|_2 u + R(B) x_j \rangle : \right.$$
$$\left. u, x_1, \ldots, x_n \in \mathbb{R}^{n+1} \ \wedge\ \|u\|_2 = 1 \ \wedge\ \forall i\ \|x_i\|_2 \le 1 \right\}.$$

4. Choose $p, q \in \{1, \ldots, k\}$ such that $\|v_p - v_q\|_2 = \max_{i,j \in \{1,\ldots,k\}} \|v_i - v_j\|_2$. Let $g_1, g_2 \in \mathbb{R}^{n+1}$ be i.i.d. standard Gaussian vectors and define $\sigma : \{1, \ldots, n\} \to \{1, \ldots, k\}$ by

$$\sigma(r) = \begin{cases} p & \text{if } \langle g_1, x_r \rangle \ge \langle g_2, x_r \rangle, \\ q & \text{if } \langle g_2, x_r \rangle \ge \langle g_1, x_r \rangle. \end{cases} \tag{6}$$

5. Choose distinct $\alpha, \beta, \gamma \in \{1, \ldots, k\}$ such that

$$\left\| v_\alpha - \frac{v_\alpha + v_\beta + v_\gamma}{3} \right\|_2^2 + \left\| v_\beta - \frac{v_\alpha + v_\beta + v_\gamma}{3} \right\|_2^2 + \left\| v_\gamma - \frac{v_\alpha + v_\beta + v_\gamma}{3} \right\|_2^2$$

is maximized among all such choices of α, β, γ. Let $g_1, g_2, g_3 \in \mathbb{R}^{n+1}$ be i.i.d. standard Gaussian vectors and define $\tau : \{1, \ldots, n\} \to \{1, \ldots, k\}$ by

$$\tau(r) = \begin{cases} \alpha & \text{if } \langle g_1, x_r \rangle \ge \max\{\langle g_2, x_r \rangle, \langle g_3, x_r \rangle\}, \\ \beta & \text{if } \langle g_2, x_r \rangle \ge \max\{\langle g_1, x_r \rangle, \langle g_3, x_r \rangle\}, \\ \gamma & \text{if } \langle g_3, x_r \rangle \ge \max\{\langle g_1, x_r \rangle, \langle g_2, x_r \rangle\}. \end{cases} \tag{7}$$

6. Output σ if $\sum_{i,j=1}^n a_{ij} b_{\sigma(i)\sigma(j)} \ge \sum_{i,j=1}^n a_{ij} b_{\tau(i)\tau(j)}$. Otherwise output τ.

Remark 1. The astute reader might notice that there is an obvious generalization of the above algorithm. Namely for every fixed integer $s \in [2, k]$ we can choose a subset $S \subseteq \{1, .., k\}$ of cardinality s which maximizes the quantity

$$\sum_{i \in S} \left\| v_i - \frac{1}{s} \sum_{j \in S} v_j \right\|_2^2.$$

Then, we can choose s i.i.d. standard Gaussians $\{g_i\}_{i \in S} \subseteq \mathbb{R}^{n+1}$ and define $\sigma_s : \{1, \ldots, n\} \to \{1, \ldots, k\}$ analogously to the above, namely $\sigma_s(r) = i$ if

$$\langle g_i, x_r \rangle = \max_{j \in S} \langle g_j, x_r \rangle.$$

Then, we can consider the assignments $\sigma_2, \sigma_3, \ldots, \sigma_s$ and choose the one which maximizes the objective $\sum_{i,j=1}^n a_{ij} b_{\sigma_\ell(i)\sigma_\ell(j)}$. In spite of this flexibility, it turns out that the the rounding method described above does not improve if we take $s \ge 4$. In order to demonstrate this fact we will proceed below to analyze the algorithm for general s, and then optimize over s.

Bounds on the performance of the above algorithm are contained in the following theorem:

Theorem 1. *Assume that A is centered, i.e. that $\sum_{i,j=1}^{n} a_{ij} = 0$. Let $p, q, \alpha, \beta, \gamma \in \{1, \ldots, k\}$ and $v_1, \ldots, v_k$ be as in the description above. Then the algorithm outputs in polynomial time a random assignment $\lambda : \{1, \ldots, n\} \to \{1, \ldots, k\}$ satisfying*

$$\mathbf{Clust}(A|B) \leq \min \left\{ \frac{2\pi R(B)^2}{\|v_p - v_q\|_2^2}, \frac{16\pi R(B)^2/9}{\|v_\alpha - v_{\alpha,\beta,\gamma}\|_2^2 + \|v_\beta - v_{\alpha,\beta,\gamma}\|_2^2 + \|v_\gamma - v_{\alpha,\beta,\gamma}\|_2^2} \right\} \cdot \mathbb{E}\left[\sum_{i,j=1}^{n} a_{ij} b_{\lambda(i)\lambda(j)} \right], \quad (8)$$

where $v_{\alpha,\beta,\gamma} = \frac{v_\alpha + v_\beta + v_\gamma}{3}$. In particular we always have

$$\mathbf{Clust}(A|B) \leq \pi \left(1 - \frac{1}{k}\right) \mathbb{E}\left[\sum_{i,j=1}^{n} a_{ij} b_{\lambda(i)\lambda(j)} \right], \quad (9)$$

and if B is centered and spherical, i.e. $\sum_{i,j=1}^{k} b_{ij} = 0$ and $b_{ii} = 1$ for all i, then

$$\mathbf{Clust}(A|B) \leq \frac{8\pi}{9} \left(1 - \frac{1}{k}\right) \mathbb{E}\left[\sum_{i,j=1}^{n} a_{ij} b_{\lambda(i)\lambda(j)} \right]. \quad (10)$$

The same bound in (10) *holds true if B is the identity matrix.*

We single out in the next theorem the case $k \in \{2, 3\}$, since in these cases we have matching UGC hardness results. Note that for general k we obtain a factor π approximation algorithm, answering positively the question posed by Song, Smola, Gretton and Borgwardt in [21].

Theorem 2. *Assume that A is centered and B is a 2×2 matrix. Then our algorithm achieves a $\frac{\pi}{2}$ approximation factor. Assuming the Unique Games Conjecture no polynomial time algorithm achieves an approximation guarantee smaller than $\frac{\pi}{2}$ in this case.*

Assume that A is centered, $k = 3$ and B is centered and spherical (since $k = 3$ this forces B to be the Gram matrix of the-degree three roots of unity in the complex plane). Then our algorithm achieves an approximation factor of $\frac{16\pi}{27}$. Assuming the Unique Games Conjecture no polynomial time algorithm achieves an approximation guarantee smaller than $\frac{16\pi}{27}$ in this case.

In fact, we believe that the UGC hardness threshold for the kernel clustering problem when A is centered and B is spherical and centered is exactly

$$\frac{8\pi}{9} \left(1 - \frac{1}{k}\right).$$

In the introduction we described a geometric conjecture which we show (in the full version of this abstract) implies this tight UGC threshold for general k.

We end the discussion by stating a (probably suboptimal) constant factor approximation result when A is not necessarily centered (note that this case is of lesser interest in terms of the applications in machine learning). In this case the above algorithm gives a constant factor approximation. The slightly better bound on the approximation factor in Theorem 3 below follows from a variant of the above algorithm which will be described in its proof.

Theorem 3. *For general A and B (not necessarily centered) there exists a polynomial time algorithm that achieves an approximation factor of*

$$1 + \frac{2\pi}{\|v_p - v_q\|_2^2} \cdot \max_{i \in \{1,\ldots,k\}} \left\| v_i - \frac{v_p + v_q}{2} \right\|_2^2 \leq 1 + \frac{3\pi}{2}.$$

The proof of Theorem 2 is contained in the full version. We shall now proceed to prove Theorem 1. Before doing so we will show how the general bound in (8) implies the bounds (9) and (10). The proof of Theorem 3 is deferred to the end of this section.

To prove that (8) implies (9) let D denote the diameter of the set $\{v_1, \ldots, v_k\}$, i.e. $D = \|v_p - v_q\|_2$. A classical theorem of Jung [12] (see [7]) says that

$$R(B) \leq D \cdot \sqrt{\frac{k-1}{2k}},$$

and (9) follows immediately by taking the first term in the minimum in (8).

We shall now show that (8) implies (10) when B is either centered and spherical or the identity matrix. Assume first of all that B is centered and spherical. Note that since $v_1, \ldots, v_k$ are unit vectors, $R(B) \leq 1$. Hence, by considering the second term in the minimum in (8) we see that it is enough to show that there exist $\alpha, \beta, \gamma \in \{1, \ldots, k\}$ for which

$$\left\| v_\alpha - \frac{v_\alpha + v_\beta + v_\gamma}{3} \right\|_2^2 + \left\| v_\beta - \frac{v_\alpha + v_\beta + v_\gamma}{3} \right\|_2^2 + \left\| v_\gamma - \frac{v_\alpha + v_\beta + v_\gamma}{3} \right\|_2^2 \geq \frac{2k}{k-1}.$$

This follows from an averaging argument. Indeed,

$$\begin{aligned}
&\frac{1}{\binom{k}{3}} \sum_{\substack{\alpha,\beta,\gamma\in\{1,\ldots,k\}\\ \alpha<\beta<\gamma}} \left(\left\| v_\alpha - \frac{v_\alpha+v_\beta+v_\gamma}{3} \right\|_2^2 \right. \\
&\qquad \left. + \left\| v_\beta - \frac{v_\alpha+v_\beta+v_\gamma}{3} \right\|_2^2 + \left\| v_\gamma - \frac{v_\alpha+v_\beta+v_\gamma}{3} \right\|_2^2 \right) \\
&= \frac{2}{k}\sum_{i=1}^k \|v_i\|_2^2 - \frac{2}{k(k-1)} \sum_{\substack{i,j\in\{1,\ldots,k\}\\ i\neq j}} \langle v_i, v_j\rangle \\
&= \frac{2}{k}\sum_{i=1}^k b_{ii} - \frac{2}{k(k-1)} \left(\sum_{i,j=1}^k b_{ij} - \sum_{i=1}^k b_{ii} \right) \\
&= \frac{2k}{k-1}.
\end{aligned}$$

This complete the proof of (10) when B is spherical and centered. The same bound holds true when $B = I_k$ is the identity matrix since in this case if we denote by $e_1, \ldots, e_k$ the standard unit basis of $\mathbb{R}^k$ and $e = \frac{1}{k}\sum_{i=1}^k e_i$ then for every assignment $\lambda : \{1,\ldots,n\} \to \{1,\ldots,k\}$ we have

$$\begin{aligned}
&\sum_{i,j=1}^n a_{ij} (I_k)_{\lambda(i)\lambda(j)} \\
&= \sum_{i,j=1}^n \langle u_i, u_j\rangle \langle e_{\lambda(i)}, e_{\lambda(j)}\rangle \\
&= \sum_{i,j=1}^n \langle u_i, u_j\rangle \langle e_{\lambda(i)} - e, e_{\lambda(j)} - e\rangle \\
&\quad + 2\left\langle \sum_{i=1}^n u_i, \sum_{j=1}^n \langle e, e_{\lambda(j)}\rangle u_j \right\rangle - \|e\|_2^2 \left\| \sum_{i=1}^k u_i \right\|_2^2.
\end{aligned}$$

The last two terms in (11) vanish since A is centered. Thus

$$\sum_{i,j=1}^n a_{ij} (I_k)_{\lambda(i)\lambda(j)} = \frac{k-1}{k} \sum_{i,j=1}^n a_{ij} c_{\lambda(i)\lambda(j)},$$

where $C = (c_{ij}) = \frac{k}{k-1}(\langle e_i - e, e_j - e\rangle)$ is spherical and centered. Thus the case of the identity matrix reduces to the previous analysis.

Proof of Theorem 1. Denote

$$\mathrm{SDP} := \max \sum_{i,j=1}^n a_{ij} \cdot \langle \|w\|_2 u + R(B) x_i,\ \|w\|_2 u + R(B) x_j \rangle,$$

where the maximum is taken over all $u, x_1, \ldots, x_n \in \mathbb{R}^{n+1}$ such that $\|u\|_2 = 1$ and $\|x_i\|_2 \le 1$ for all i. Observe that

$$\mathrm{SDP} \ge \mathbf{Clust}(A|B). \tag{11}$$

Indeed, for every $\lambda : \{1,\ldots,n\} \to \{1,\ldots,k\}$ define $u = \frac{w}{\|w\|_2}$ and $x_i = \frac{v_{\lambda(i)} - w}{R(B)}$ and note that in this case

$$\sum_{i,j=1}^n a_{ij} \cdot \langle \|w\|_2 u + R(B) x_i,\ \|w\|_2 u + R(B) x_j \rangle = \sum_{i,j=1}^n a_{ij} b_{\lambda(i)\lambda(j)}.$$

Let $u^*, x_1^*, \ldots, x_n^*$ be the optimal solution to the SDP. It will be convenient to think of the SDP solution as being split into two parts. So we rewrite

$$\begin{aligned}
\mathrm{SDP} &= \sum_{i,j=1}^n a_{ij} \cdot \left\langle \|w\|_2 u^* + R(B) x_i^*,\ \|w\|_2 u^* + R(B) x_j^* \right\rangle \\
&= \sum_{i,j=1}^n \langle u_i, u_j\rangle \cdot \langle \|w\|_2 u^* + R(B) x_i^*,\ \|w\|_2 u^* + R(B) x_j^* \rangle \\
&= \left\| \sum_{i=1}^n u_i \otimes (\|w\|_2 u^* + R(B) x_i^*) \right\|_2^2 \qquad (12) \\
&= \left\| \left(\|w\|_2 \left(\sum_{i=1}^n u_i \right) \otimes u^* \right) + \left(R(B) \sum_{i=1}^n u_i \otimes x_i^* \right) \right\|_2^2 \\
&= \|P + Q\|_2^2, \qquad (13)
\end{aligned}$$

where

$$P := \|w\|_2 \sum_{i=1}^n u_i \otimes u^*, \tag{14}$$

and

$$Q := R(B) \sum_{i=1}^n u_i \otimes x_i^*. \tag{15}$$

Observe in passing that (13) implies that the objective function of the SDP is convex as a function of $u, x_1, \ldots, x_n$, and therefore we may assume that $\|u^*\|_2 = 1$ and $\|x_i^*\|_2 = 1$ for all i.

We shall now proceed with the analysis of our algorithm while using the variant described in Remark 1. This will not create any additional complication, and will allow us to explain why there is no advantage in working with subsets of size $s \ge 4$. Recall the setting: for a fixed integer $s \in [2,k]$ we choose a subset $S \subseteq \{1,..,k\}$ of cardinality s which maximizes the quantity

$$\sum_{i\in S} \left\| v_i - \frac{1}{s}\sum_{j\in S} v_j \right\|_2^2.$$

Then, we choose s i.i.d. standard Gaussians $\{g_i\}_{i\in S} \subseteq \mathbb{R}^{n+1}$ and define $\sigma : \{1,\ldots,n\} \to \{1,\ldots,k\}$ by setting $\sigma(r) = i$ if

$$\langle g_i, x_r^*\rangle = \max_{j\in S} \langle g_j, x_r^*\rangle.$$

Fix $i,j \in \{1,\ldots,n\}$. As proved by Frieze and Jerrum in [9] (see Lemma 5 there), we have[3]:

$$\Pr\left[\sigma(i)=\sigma(j)\right] = \sum_{m=0}^{\infty} R_m(s)\langle x_i^*, x_j^*\rangle^m,$$

where the power series converges on $[-1,1]$ and all the coefficients $R_m(s)$ are non-negative. Moreover $R_0(s) = \frac{1}{s}$ and

$$\begin{aligned} R_1(s) &= \frac{1}{s-1}\left(\mathbb{E}\left[\max_{j\in S} g_j\right]\right)^2 \\ &= \frac{s}{(2\pi)^{s/2}}\int_{-\infty}^{\infty} x e^{-x^2/2}\left(\int_{-\infty}^{x} e^{-y^2/2}dy\right)^{s-1} dx. \end{aligned}$$

Note that conditioned on the event $\sigma(i)=\sigma(j)$, the random index $\sigma(i)$ is uniformly distributed over S. Also, conditioned on the event $\sigma(i)\neq\sigma(j)$, the pair $(\sigma(i),\sigma(j))$ is uniformly distributed over all $s(s-1)$ pairs of distinct indices in S. Thus

$$\begin{aligned} \mathbb{E}[b_{\sigma(i)\sigma(j)}] = \Pr[\sigma(i)=\sigma(j)]\cdot\left(\frac{1}{s}\sum_{\ell\in S} b_{\ell\ell}\right) \\ + \Pr[\sigma(i)\neq\sigma(j)]\cdot\left(\frac{1}{s(s-1)}\sum_{\substack{\ell,t\in S\\ \ell\neq t}} b_{\ell t}\right). \end{aligned}$$

Denote $\Phi = \frac{1}{s}\sum_{\ell\in S} b_{\ell\ell}$ and $\Psi = \frac{1}{s(s-1)}\sum_{\substack{\ell,t\in S\\ \ell\neq t}} b_{\ell t}$. (note that Φ, Ψ depend on the matrix B as well as the choice of the subset $S\subseteq\{1,\ldots,k\}$. Thus

$$\begin{aligned} \mathbb{E}[b_{\sigma(i)\sigma(j)}] &= \left(\sum_{m=0}^{\infty} R_m(s)\langle x_i^*, x_j^*\rangle^m\right)\cdot\Phi \\ &\quad + \left(1-\sum_{m=0}^{\infty} R_m(s)\langle x_i^*, x_j^*\rangle^m\right)\cdot\Psi \\ &= (\Psi + (\Phi-\Psi)R_0(s)) \\ &\quad +(\Phi-\Psi)\sum_{m=1}^{\infty} R_m(s)\langle x_i^*, x_j^*\rangle^m. \quad (16) \end{aligned}$$

Write $v := \frac{1}{s}\sum_{\ell\in S} v_\ell$. Observe that

$$\Psi + (\Phi-\Psi)R_0(s) = \|v\|_2^2. \quad (17)$$

[3]We are using here the fact that $x_1^*,\ldots,x_n^*$ are unit vectors.

Indeed, since $R_0(s) = 1/s$ we have

$$\begin{aligned} &\Psi + (\Phi-\Psi)R_0(s) \\ &= \left(1-\frac{1}{s}\right)\left(\frac{1}{s(s-1)}\sum_{\substack{\ell,t\in S\\ \ell\neq t}} b_{\ell t}\right) + \frac{1}{s}\left(\frac{1}{s}\sum_{\ell\in S} b_{\ell\ell}\right) \\ &= \frac{1}{s^2}\sum_{\ell,t\in S} b_{\ell t} = \left\|\frac{1}{s}\sum_{\ell\in S}^{s} v_\ell\right\|_2^2 = \|v\|_2^2. \end{aligned}$$

Moreover,

$$(s-1)(\Phi-\Psi) = \sum_{\ell\in S}\|v_\ell - v\|_2^2. \quad (18)$$

In particular $\Phi-\Psi\geq 0$. To prove (18) we simply expand:

$$\begin{aligned} \sum_{\ell\in S}\|v_\ell - v\|_2^2 &= \sum_{\ell\in S}\|v_\ell\|_2^2 - s\|v\|_2^2 \\ &= s\Phi - \frac{1}{s}\sum_{\ell,t\in S} b_{\ell t} \\ &= s\Phi - \frac{1}{s}\left(s\Phi + s(s-1)\Psi\right) \\ &= (s-1)(\Phi-\Psi). \end{aligned}$$

Multiplying both sides of equation (16) by a_{ij} and summing over $i,j\in\{1,\ldots,n\}$ while using (17) we get that

$$\begin{aligned} &\mathbb{E}\left[\sum_{i,j=1}^{n} a_{ij} b_{\sigma(i)\sigma(j)}\right] \\ &= \|v\|_2^2\sum_{i,j=1}^{n} a_{ij} + (\Phi-\Psi)R_1(s)\sum_{i,j=1}^{n} a_{ij}\langle x_i^*, x_j^*\rangle \\ &\quad +(\Phi-\Psi)\sum_{m=2}^{\infty} R_m(s)\sum_{i,j=1}^{n} a_{ij}\langle x_i^*, x_j^*\rangle^m. \quad (19) \end{aligned}$$

Note that for every $m\geq 1$ we have

$$\begin{aligned} \sum_{i,j=1}^{n} a_{ij}\langle x_i^*, x_j^*\rangle^m &= \sum_{i,j=1}^{n}\langle u_i, u_j\rangle\left\langle (x_i^*)^{\otimes m}, (x_j^*)^{\otimes m}\right\rangle \\ &= \left\|\sum_{i=1}^{n} u_i\otimes(x_i^*)^{\otimes m}\right\|_2^2 \geq 0. \quad (20) \end{aligned}$$

Plugging (20) into (19), and using the fact that $\Phi-\Psi\geq 0$ and the positivity of $\mathbb{R}_m(s)$, we conclude that

$$\begin{aligned} &\mathbb{E}\left[\sum_{i,j=1}^{n} a_{ij} b_{\sigma(i)\sigma(j)}\right] \\ &\geq \|v\|_2^2\sum_{i,j=1}^{n} a_{ij} + (\Phi-\Psi)R_1(s)\sum_{i,j=1}^{n} a_{ij}\langle x_i^*, x_j^*\rangle. \quad (21) \end{aligned}$$

We shall now use the fact that $\sum_{i,j=1}^n a_{ij} = 0$ for the first time. In this case $P = 0$ (see equations (13) and (14)) so that

$$\mathrm{SDP} = R(B)^2 \sum_{i,j=1}^n a_{ij} \langle x_i^*, x_j^* \rangle. \tag{22}$$

Hence, using (21) and (17) we get the bound

$$\begin{aligned}\mathbb{E}\left[\sum_{i,j=1}^n a_{ij} b_{\sigma(i)\sigma(j)}\right] &\geq \frac{R_1(s)\sum_{\ell\in S}\|v_\ell - v\|_2^2}{(s-1)R(B)^2}\cdot \mathrm{SDP} \\ &\overset{(11)}{\geq} \frac{R_1(s)\sum_{\ell\in S}\|v_\ell - v\|_2^2}{(s-1)R(B)^2}\cdot \mathbf{Clust}(A|B). \end{aligned} \tag{23}$$

The term $R_1(s)$ is studied in the full version, where its geometric interpretation is explained. In particular, it is shown in the full version that $R_1(s) < R_1(3)$ for every $s \geq 4$ and that $R_1(2) = \frac{1}{\pi}$ and $R_1(3) = \frac{9}{8\pi}$. Hence the cases $s \in \{2, 3\}$ in (23) conclude the proof of Theorem 1. Moreover, we see that for $s \geq 4$ the lower bound in (23) is worse than the lower bound obtained when case $s = 3$. Indeed, we have already noted that in this case $R_1(s) < R_1(3)$. In addition,

$$\begin{aligned}&\frac{1}{\binom{s}{3}}\sum_{\substack{T\subseteq S\\|T|=3}}\frac{1}{2}\sum_{\ell\in T}\left\|v_\ell - \frac{1}{3}\sum_{t\in T} v_t\right\|_2^2 \\ &= \frac{1}{s}\sum_{\ell\in S}\|v_\ell\|_2^2 - \frac{1}{s(s-1)}\sum_{\substack{\ell,t\in S\\ \ell\neq t}}\langle v_\ell, v_t\rangle \\ &= \frac{1}{s-1}\sum_{\ell\in S}\left\|v_\ell - \frac{1}{s}\sum_{t\in S} v_t\right\|_2^2.\end{aligned}$$

This implies that there exists $T \subseteq S$ with $|T| = 3$ for which

$$\frac{1}{2}\sum_{\ell\in T}\left\|v_\ell - \frac{1}{3}\sum_{t\in T} v_t\right\|_2^2 \geq \frac{1}{s-1}\sum_{\ell\in S}\|v_\ell - v\|_2^2,$$

so that when $s \geq 4$ the lower bound in (23) is inferior to the same lower bound when $s = 3$. □

It remains to deal with the case $\sum_{i,j=1}^n a_{ij} > 0$, i.e. to prove Theorem 3.

Proof of Theorem 3. We slightly modify the algorithm that was studied in Theorem 1. Let $v_1, \ldots, v_k$ and $p, q \in \{1, \ldots, k\}$ be as before, that is $b_{ij} = \langle v_i, v_j\rangle$ and $\|v_p - v_q\|_2 = \max_{i,j\in\{1,\ldots,k\}}\|v_i - v_j\|_2 = D$, the diameter of the set $\{v_1, \ldots, v_k\} \in \mathbb{R}^k$. Denote $w' := \frac{v_p+v_q}{2}$ and

$$R'(B) := \max_{i\in\{1,\ldots,k\}}\|v_i - w'\|_2.$$

We now consider the modified semidefinite program

$$\mathrm{SDP} := \max \sum_{i,j=1}^n a_{ij}\cdot\langle \|w'\|_2 u + R'(B)x_i,\ \|w'\|_2 u + R'(B)x_j\rangle,$$

where the maximum is taken over all $u, x_1, \ldots, x_n \in \mathbb{R}^{n+1}$ such that $\|u\|_2 = 1$ and $\|x_i\|_2 \leq 1$ for all i. From now on we will use the notation of the proof of Theorem 1 with w replaced by w' and $R(B)$ replaced by $R'(B)$ (this slight abuse of notation will not create any confusion). As before, we let $g_1, g_2 \in \mathbb{R}^{n+1}$ be i.i.d. standard Gaussian vectors and define $\sigma : \{1, \ldots, n\} \to \{1, \ldots, k\}$ by

$$\sigma(r) = \begin{cases} p & \text{if } \langle g_1, x_r\rangle \geq \langle g_2, x_r\rangle, \\ q & \text{if } \langle g_2, x_r\rangle \geq \langle g_1, x_r\rangle. \end{cases} \tag{24}$$

Note that the first place in the proof of Theorem 1 where the assumption that A is centered was used in equation (22). Hence, in the present setting we still have the bounds

$$\mathbf{Clust}(A|B) \leq \mathrm{SDP} = \|P + Q\|_2^2 \leq \left(\|P\|_2 + \|Q\|_2\right)^2, \tag{25}$$

where P and Q are defined in (14) and (15) (with w and $R(B)$ replaced by w' and $R'(B)$, respectively). Also, it follows from (21) that

$$\begin{aligned}\mathbb{E}\left[\sum_{i,j=1}^n a_{ij} b_{\sigma(i)\sigma(j)}\right] &\geq \|v\|_2^2 \sum_{i,j=1}^n a_{ij} \\ &+ \left(\|v_p - v\|_2^2 + \|v_q - v\|_2^2\right) R_1(2)\sum_{i,j=1}^n a_{ij}\langle x_i^*, x_j^*\rangle,\end{aligned} \tag{26}$$

where $v = \frac{v_p+v_q}{2} = w'$. Note that $\|v_p - v\|_2^2 + \|v_q - v\|_2^2 = \frac{D^2}{2}$, and recall that $R_1(2) = \frac{1}{\pi}$. Thus (26) becomes:

$$\mathbb{E}\left[\sum_{i,j=1}^n a_{ij} b_{\sigma(i)\sigma(j)}\right] \geq \|w'\|_2^2 \sum_{i,j=1}^n a_{ij} + \frac{D^2}{2\pi}\sum_{i,j=1}^n a_{ij}\langle x_i^*, x_j^*\rangle, \tag{27}$$

Note that

$$\begin{aligned}\|P\|_2^2 = \|w'\|_2^2 \cdot \left\|\sum_{i=1}^n u_i \otimes u^*\right\|_2^2 &= \|w'\|_2^2 \cdot \|u^*\|_2^2 \sum_{i,j=1}^n \langle u_i, u_j\rangle \\ &= \|w'\|_2^2 \sum_{i,j=1}^n a_{ij}.\end{aligned} \tag{28}$$

and

$$\|Q\|_2^2 = R'(B)^2 \cdot \left\|\sum_{i=1}^n u_i \otimes x_i^*\right\|_2^2 = R'(B)^2 \sum_{i,j=1}^n a_{ij}\langle x_i^*, x_j^*\rangle. \tag{29}$$

Combining (25) and (27) with (28) and (29) we see that

$$\mathbf{Clust}(A|B) \leq \frac{\left(\|P\|_2 + \|Q\|_2\right)^2}{\|P_2\|_2^2 + c\|Q\|_2^2} \cdot \mathbb{E}\left[\sum_{i,j=1}^{n} a_{ij} b_{\sigma(i)\sigma(j)}\right], \quad (30)$$

where $c = \frac{D^2}{2\pi R'(B)^2}$. The convexity of the function $x \to x^2$ implies that

$$\begin{aligned}
&\left(\|P\|_2 + \|Q\|_2\right)^2 \\
&= \left(\frac{c}{c+1} \cdot \frac{c+1}{c}\|P\|_2^2 + \left(1 - \frac{c}{c+1}\right)(c+1)\|Q\|_2^2\right)^2 \\
&\leq \frac{c+1}{c}\|P\|_2^2 + (c+1)\|Q\|_2^2 \\
&= \left(1 + \frac{1}{c}\right)\left(\|P\|_2^2 + c\|Q\|_2^2\right).
\end{aligned}$$

Thus (30) implies that our algorithm achieves an approximation guarantee bounded above by

$$\begin{aligned}
1 + \frac{1}{c} &= 1 + \frac{2\pi R'(B)^2}{D^2} \\
&= 1 + \frac{2\pi}{\|v_p - v_q\|_2^2} \cdot \max_{i \in \{1,\ldots,k\}} \left\| v_i - \frac{v_p + v_q}{2} \right\|_2^2.
\end{aligned}$$

It remains to note that for every $i \in \{1, \ldots, k\}$ we know that $\|v_i - v_p\|_2, \|v_i - v_q\|_2 \leq D$ and therefore $\|v_i - w'\|_2 \leq \frac{\sqrt{3}}{2}D$. This implies that our approximation guarantee is bounded from above by $1 + \frac{3\pi}{2}$. □

References

[1] N. Alon, K. Makarychev, Y. Makarychev, and A. Naor. Quadratic forms on graphs. *Invent. Math.*, 163(3):499–522, 2006.

[2] N. Alon and A. Naor. Approximating the cut-norm via Grothendieck's inequality. *SIAM J. Comput.*, 35(4):787–803 (electronic), 2006.

[3] S. Arora, E. Berger, G. Kindler, E. Hazan, and S. Safra. On non-approximability for quadratic programs. In *46th Annual Symposium on Foundations of Computer Science*, pages 206–215. IEEE Computer Society, 2005.

[4] M. Charikar, K. Makarychev, and Y. Makarychev. Near-optimal algorithms for unique games (extended abstract). In *STOC'06: Proceedings of the 38th Annual ACM Symposium on Theory of Computing*, pages 205–214, New York, 2006. ACM.

[5] M. Charikar, K. Makarychev, and Y. Makarychev. Near-optimal algorithms for maximum constraint satisfaction problems. In *SODA '07: Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 62–68, Philadelphia, PA, USA, 2007. Society for Industrial and Applied Mathematics.

[6] M. Charikar and A. Wirth. Maximizing quadratic programs: extending Grothendieck's inequality. In *45th Annual Symposium on Foundations of Computer Science*, pages 54–60. IEEE Computer Society, 2004.

[7] L. Danzer, B. Grünbaum, and V. Klee. Helly's theorem and its relatives. In *Proc. Sympos. Pure Math., Vol. VII*, pages 101–180. Amer. Math. Soc., Providence, R.I., 1963.

[8] U. Feige, G. Kindler, and R. O'Donnell. Understanding parallel repetition requires understanding foams. In *IEEE Conference on Computational Complexity*, pages 179–192. IEEE Computer Society, 2007.

[9] A. Frieze and M. Jerrum. Improved approximation algorithms for MAX k-CUT and MAX BISECTION. *Algorithmica*, 18(1):67–81, 1997.

[10] P. Gritzmann and V. Klee. Inner and outer j-radii of convex bodies in finite-dimensional normed spaces. *Discrete Comput. Geom.*, 7(3):255–280, 1992.

[11] J. Håstad. Some optimal inapproximability results. *J. ACM*, 48(4):798–859 (electronic), 2001.

[12] H. W. E. Jung. über die kleinste kügel, die einerumliche figureinschlisst. *J. Reine Angew. Math.*, 123:241–257, 1901.

[13] S. Khot. On the power of unique 2-prover 1-round games. In *Proceedings of the Thirty-Fourth Annual ACM Symposium on Theory of Computing*, pages 767–775 (electronic), New York, 2002. ACM.

[14] S. Khot, G. Kindler, E. Mossel, and R. O'Donnell. Optimal inapproximability results for max-cut and other 2-variable csps? In *45th Annual Symposium on Foundations of Computer Science*, pages 146–154. IEEE Computer Society, 2004.

[15] S. Khot, G. Kindler, E. Mossel, and R. O'Donnell. Optimal inapproximability results for MAX-CUT and other 2-variable CSPs? *SIAM J. Comput.*, 37(1):319–357 (electronic), 2007.

[16] A. Nemirovski, C. Roos, and T. Terlaky. On maximization of quadratic form over intersection of ellipsoids with common center. *Math. Program.*, 86(3, Ser. A):463–473, 1999.

[17] Y. Nesterov. Semidefinite relaxation and nonconvex quadratic optimization. *Optim. Methods Softw.*, 9(1-3):141–160, 1998.

[18] P. Raghavendra. Optimal algorithms and inapproximability results for every csp? In *Proceedings of the 40th Annual ACM Symposium on Theory of Computing*, pages 245–254, 2008.

[19] R. E. Rietz. A proof of the Grothendieck inequality. *Israel J. Math.*, 19:271–276, 1974.

[20] B. Scholkopf and A. J. Smola. *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. MIT Press, Cambridge, MA, USA, 2001.

[21] L. Song, A. Smola, A. Gretton, and K. A. Borgwardt. A dependence maximization view of clustering. In *Proceedings of the 24th international conference on Machine learning*, pages 815 – 822, 2007.