Technical Report CS2008-0916

Department of Computer Science and Engineering
University of California, San Diego

The hardness of k-means clustering

Sanjoy Dasgupta[1]

Abstract

We show that k-means clustering is an NP-hard optimization problem, even if k is fixed to 2.

1 Introduction

In this brief note, we establish the hardness of the following optimization problem.

k-MEANS CLUSTERING

Input: A set of points $x_1, \ldots, x_n \in \mathbb{R}^d$; an integer k.

Output: A partition of the points into clusters $C_1, \ldots, C_k$, along with a center μ_j for each cluster, so as to minimize

$$\sum_{j=1}^{k} \sum_{i \in C_j} \|x_i - \mu_j\|^2.$$

(Here $\|\cdot\|$ is Euclidean distance.) It can be checked that in any optimal solution, μ_j is the mean of the points in C_j. Thus the $\{\mu_j\}$ can be removed entirely from the formulation of the problem. To this end, let X, Y be i.i.d. random draws from C_j. Simple algebra shows $\mathbb{E}\|X - Y\|^2 = 2\mathbb{E}\|X - \mathbb{E}X\|^2$, which implies

$$\sum_{i \in C_j} \|x_i - \mu_j\|^2 = \frac{1}{2|C_j|} \sum_{i,i' \in C_j} \|x_i - x_{i'}\|^2.$$

Therefore, the k-means cost function can equivalently be rewritten as

$$\sum_{j=1}^{k} \frac{1}{2|C_j|} \sum_{i,i' \in C_j} \|x_i - x_{i'}\|^2.$$

We consider the specific case when k is fixed to 2.

Theorem 1 *2-means clustering is an NP-hard optimization problem.*

This was recently asserted in [1], but the proof was flawed. Here, we use a sequence of reductions involving three problems, the first of which is a standard restriction of 3SAT, well known to be NP-complete.

3SAT

Input: A Boolean formula in 3CNF, where each clause has exactly three literals and each variable appears at least twice.

Output: `true` if formula is satisfiable, `false` if not.

[1]Email: dasgupta@cs.ucsd.edu. The author acknowledges the support of the National Science Foundation, under grants IIS-0347646 and IIS-0713540.

The second problem is a special case of NOT-ALL-EQUAL 3SAT, which can be shown to be hard by a straightforward variant of the usual reduction from 3SAT. For completeness, we lay out the details in the next section.

NAESAT*

Input: A Boolean formula $\phi(x_1, \ldots, x_n)$ in 3CNF, such that (i) every clause contains exactly three literals, and (ii) each pair of variables x_i, x_j appears together in at most two clauses, once as either $\{x_i, x_j\}$ or $\{\overline{x}_i, \overline{x}_j\}$, and once as either $\{\overline{x}_i, x_j\}$ or $\{x_i, \overline{x}_j\}$.

Output: `true` if there exists an assignment in which each clause contains exactly one or two satisfied literals; `false` otherwise.

The final problem we consider is a generalization of 2-MEANS.

GENERALIZED 2-MEANS

Input: An $n \times n$ matrix of interpoint distances D_{ij}.

Output: A partition of the points into two clusters C_1 and C_2, so as to minimize

$$\sum_{j=1}^{2} \frac{1}{2|C_j|} \sum_{i,i' \in C_j} D_{ii'}.$$

Theorem 1 is shown by reduction from NAESAT*. For any input ϕ to NAESAT*, we show how to efficiently produce a distance matrix $D(\phi)$ and a threshold $c(\phi)$ such that ϕ satisfies NAESAT* if and only if $D(\phi)$ admits a generalized 2-means clustering of cost $\leq c(\phi)$.

Thus GENERALIZED 2-MEANS CLUSTERING is hard. To get back to 2-MEANS, we prove that the distance matrix $D(\phi)$ can in fact be realized by squared Euclidean distances. This existential fact is also constructive, because in such cases, the embedding can be obtained in cubic time by classical multidimensional scaling [2].

2 Hardness of NAESAT*

Given an input $\phi(x_1, \ldots, x_n)$ to 3SAT, we first construct an intermediate formula ϕ' that is satisfiable if and only if ϕ is, and additionally has exactly three occurrences of each variable: one in a clause of size three, and two in clauses of size two. This ϕ' is then used to produce an input ϕ'' to NAESAT*.

1. Constructing ϕ'.

   Suppose variable x_i appears $k \geq 2$ times in ϕ. Create k variables $x_{i1}, \ldots, x_{ik}$ for use in ϕ': use the same clauses, but replace each occurrence of x_i by one of the x_{ij}. To enforce agreement between the different copies x_{ij}, add k additional clauses $(\overline{x}_{i1} \vee x_{i2}), (\overline{x}_{i2} \vee x_{i3}), \ldots, (\overline{x}_{ik}, x_{i1})$. These correspond to the implications $x_1 \Rightarrow x_2, x_2 \Rightarrow x_3, \ldots, x_k \Rightarrow x_1$.

   By design, ϕ is satisfiable if and only if ϕ' is satisfiable.

2. Constructing ϕ''.

   Now we construct an input ϕ'' for NAESAT*. Suppose ϕ' has m clauses with three literals and m' clauses with two literals. Create $2m + m' + 1$ new variables: $s_1, \ldots, s_m$ and $f_1, \ldots, f_{m+m'}$ and f.

   If the jth three-literal clause in ϕ' is $(\alpha \vee \beta \vee \gamma)$, replace it with two clauses in ϕ'': $(\alpha \vee \beta \vee s_j)$ and $(\overline{s}_j \vee \gamma \vee f_j)$. If the jth two-literal clause in ϕ' is $(\alpha \vee \beta)$, replace it with $(\alpha \vee \beta \vee f_{m+j})$ in ϕ''. Finally, add $m + m'$ clauses that enforce agreement among the f_i: $(\overline{f}_1 \vee f_2 \vee f), (\overline{f}_2 \vee f_3 \vee f), \ldots, (\overline{f}_{m+m'} \vee f_1 \vee f)$.

   All clauses in ϕ'' have exactly three literals. Moreover, the only pairs of variables that occur together (in clauses) more than once are $\{f_i, f\}$ pairs. Each such pair occurs twice, as $\{f_i, f\}$ and $\{\overline{f}_i, f\}$.

Lemma 2 *ϕ' is satisfiable if and only if ϕ'' is not-all-equal satisfiable.*

Proof. First suppose that ϕ' is satisfiable. Use the same settings of the variables for ϕ''. Set $f = f_1 = \cdots = f_{m+m'} = \texttt{false}$. For the jth three-literal clause $(\alpha \vee \beta \vee \gamma)$ of ϕ', if $\alpha = \beta = \texttt{false}$ then set s_j to $\texttt{true}$, otherwise set s_j to $\texttt{false}$. The resulting assignment satisfies exactly one or two literals of each clause in ϕ''.

Conversely, suppose ϕ'' is not-all-equal satisfiable. Without loss of generality, the satisfying assignment has f set to $\texttt{false}$ (otherwise flip all assignments). The clauses of the form $(\overline{f}_i \vee f_{i+1} \vee f)$ then enforce agreement among all the f_i variables. We can assume they are all $\texttt{false}$ (otherwise, once again, flip all assignments). This means the two-literal clauses of ϕ' must be satisfied. Finally, consider any three-literal clause $(\alpha \vee \beta \vee \gamma)$ of ϕ'. This was replaced by $(\alpha \vee \beta \vee s_j)$ and $(\overline{s}_j \vee \gamma \vee f_j)$ in ϕ''. Since f_j is $\texttt{false}$, it follows that one of the literals α, β, γ must be satisfied. Thus ϕ' is satisfied. ∎

3 Hardness of Generalized 2-means

Given an instance $\phi(x_1, \ldots, x_n)$ of NaeSat*, we construct a $2n \times 2n$ distance matrix $D = D(\phi)$ where the (implicit) $2n$ points correspond to literals. Entries of this matrix will be indexed as $D_{\alpha,\beta}$, for $\alpha, \beta \in \{x_1, \ldots, x_n, \overline{x}_1, \ldots, \overline{x}_n\}$. Another bit of notation: we write $\alpha \sim \beta$ to mean that either α and β occur together in a clause or $\overline{\alpha}$ and $\overline{\beta}$ occur together in a clause. For instance, the clause $(x \vee \overline{y} \vee z)$ allows one to assert $\overline{x} \sim y$ but not $x \sim y$. The input restrictions on NaeSat* ensure that every relationship $\alpha \sim \beta$ is generated by a unique clause; it is not possible to have two different clauses that both contain either $\{\alpha, \beta\}$ or $\{\overline{\alpha}, \overline{\beta}\}$.

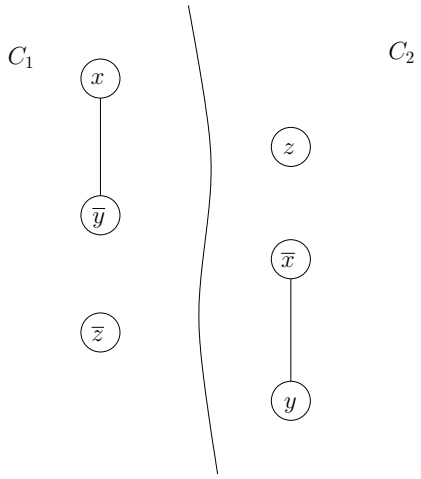
Define

$$D_{\alpha,\beta} = \begin{cases} 0 & \text{if } \alpha = \beta \\ 1 + \Delta & \text{if } \alpha = \overline{\beta} \\ 1 + \delta & \text{if } \alpha \sim \beta \\ 1 & \text{otherwise} \end{cases}$$

Here $0 < \delta < \Delta < 1$ are constants such that $4\delta m < \Delta \leq 1 - 2\delta n$, where m is the number of clauses of ϕ. One valid setting is $\delta = 1/(5m + 2n)$ and $\Delta = 5\delta m$.

Lemma 3 *If ϕ is a satisfiable instance of* NaeSat*, *then $D(\phi)$ admits a generalized 2-means clustering of cost $c(\phi) = n - 1 + 2\delta m/n$, where m is the number of clauses of ϕ.*

Proof. The obvious clustering is to make one cluster (say C_1) consist of the positive literals in the satisfying not-all-equal assignment and the other cluster (C_2) the negative literals. Each cluster has n points, and the distance between any two distinct points α, β within a cluster is either 1 or, if $\alpha \sim \beta$, $1 + \delta$. Each clause of ϕ has at least one literal in C_1 and at least one literal in C_2, since it is a not-all-equal assignment. Hence it contributes exactly one $\sim$ pair to C_1 and one $\sim$ pair to C_2. The figure below shows an example with a clause $(x \vee \overline{y} \vee z)$ and assignment $x = \texttt{true}, y = z = \texttt{false}$.



Thus the clustering cost is

$$\frac{1}{2n}\sum_{i,i'\in C_1} D_{ii'} + \frac{1}{2n}\sum_{i,i'\in C_2} D_{ii'} = 2\cdot\frac{1}{n}\left(\binom{n}{2} + m\delta\right) = n - 1 + \frac{2\delta m}{n}.$$

∎

Lemma 4 *Let C_1, C_2 be any 2-clustering of $D(\phi)$. If C_1 contains both a variable and its negation, then the cost of this clustering is at least $n - 1 + \Delta/(2n) > c(\phi)$.*

Proof. Suppose C_1 has n' points while C_2 has $2n - n'$ points. Since all distances are at least 1, and since C_1 contains a pair of points at distance $1 + \Delta$, the total clustering cost is at least

$$\frac{1}{n'}\left(\binom{n'}{2} + \Delta\right) + \frac{1}{2n - n'}\binom{2n - n'}{2} = n - 1 + \frac{\Delta}{n'} \geq n - 1 + \frac{\Delta}{2n}.$$

Since $\Delta > 4\delta m$, this is always more than $c(\phi)$. ∎

Lemma 5 *If $D(\phi)$ admits a 2-clustering of cost $\leq c(\phi)$, then ϕ is a satisfiable instance of* NAESAT*.

Proof. Let C_1, C_2 be a 2-clustering of cost $\leq c(\phi)$. By the previous lemma, neither C_1 nor C_2 contain both a variable and its negation. Thus $|C_1| = |C_2| = n$. The cost of the clustering can be written as

$$\frac{2}{n}\left(\binom{n}{2} + \delta \sum_{\text{clauses}} (1 \text{ if clause is split between } C_1,\, C_2;\, 3 \text{ otherwise})\right).$$

Since the cost is $\leq c(\phi)$, it follows that *all* clauses are split between C_1 and C_2, that is, every clause has at least one literal in C_1 and one literal in C_2. Therefore, the assignment that sets all of C_1 to `true` and all of C_2 to `false` is a valid NAESAT* assignment for ϕ. ∎

4 Embeddability of $D(\phi)$

We now show that $D(\phi)$ can be embedded into l_2^2, in the sense that there exist points $x_\alpha \in \mathbb{R}^{2n}$ such that $D_{\alpha,\beta} = \|x_\alpha - x_\beta\|^2$ for all α, β. We rely upon the following classical result [3].

Theorem 6 (Schoenberg) *Let H denote the matrix $I - (1/N)\mathbf{1}\mathbf{1}^T$. An $N \times N$ symmetric matrix D can be embedded into l_2^2 if and only if $-HDH$ is positive semidefinite.*

The following corollary is immediate.

Corollary 7 *An $N \times N$ symmetric matrix D can be embedded into l_2^2 if and only if $u^T D u \leq 0$ for all $u \in \mathbb{R}^N$ with $u \cdot \mathbf{1} = 0$.*

Proof. Since the range of the map $v \mapsto Hv$ is precisely $\{u \in \mathbb{R}^N : u \cdot \mathbf{1} = 0\}$, we have

$$\begin{aligned} -HDH \text{ is positive semidefinite} &\Leftrightarrow v^T HDHv \leq 0 \text{ for all } v \in \mathbb{R}^N \\ &\Leftrightarrow u^T Du \leq 0 \text{ for all } u \in \mathbb{R}^N \text{ with } u \cdot \mathbf{1} = 0. \end{aligned}$$

∎

Lemma 8 *$D(\phi)$ can be embedded into l_2^2.*

Proof. If ϕ is a formula with variables $x_1, \ldots, x_n$, then $D = D(\phi)$ is a $2n \times 2n$ matrix whose first n rows/columns correspond to $x_1, \ldots, x_n$ and remaining rows/columns correspond to $\overline{x}_1, \ldots, \overline{x}_n$. The entry for literals (α, β) is

$$D_{\alpha\beta} = 1 - \mathbf{1}(\alpha = \beta) + \Delta \cdot \mathbf{1}(\alpha = \overline{\beta}) + \delta \cdot \mathbf{1}(\alpha \sim \beta),$$

where $\mathbf{1}(\cdot)$ denotes the indicator function.

Now, pick any $u \in \mathbb{R}^{2n}$ with $u \cdot \mathbf{1} = 0$. Let u^+ denote the first n coordinates of u and u^- the last n coordinates.

$$\begin{aligned} u^T Du &= \sum_{\alpha,\beta} D_{\alpha\beta} u_\alpha u_\beta \\ &= \sum_{\alpha,\beta} u_\alpha u_\beta \left(1 - \mathbf{1}(\alpha = \beta) + \Delta \cdot \mathbf{1}(\alpha = \overline{\beta}) + \delta \cdot \mathbf{1}(\alpha \sim \beta)\right) \\ &= \sum_{\alpha,\beta} u_\alpha u_\beta - \sum_\alpha u_\alpha^2 + \Delta \sum_\alpha u_\alpha u_{\overline{\alpha}} + \delta \sum_{\alpha,\beta} u_\alpha u_\beta \mathbf{1}(\alpha \sim \beta) \\ &\leq \left(\sum_\alpha u_\alpha\right)^2 - \|u\|^2 + 2\Delta(u^+ \cdot u^-) + \delta \sum_{\alpha,\beta} |u_\alpha||u_\beta| \\ &\leq -\|u\|^2 + \Delta(\|u^+\|^2 + \|u^-\|^2) + \delta \left(\sum_\alpha |u_\alpha|\right)^2 \\ &\leq -(1 - \Delta)\|u\|^2 + 2\delta\|u\|^2 n \end{aligned}$$

where the last step uses the Cauchy-Schwarz inequality. Since $2\delta n \leq 1 - \Delta$, this quantity is always ≤ 0. ∎

5 Open problems

Our reduction constructs instances of 2-means with n points in $d = 2n$ dimensions. To what extent can the dependence of the dimensionality on n be reduced? Since d-dimensional 2-means can always be solved in $O(n^{d+1})$ time by enumerating all possible hyperplane separators between the two clusters, we would certainly expect a hardness result to have $d = \omega(\text{poly}\log(n))$.

Can the hardness of k-means, for general k, be established in low dimension? For $d = 1$ an efficient dynamic programming solution can be given; what about $d = 2$?

And finally, what about hardness of approximation?

References

[1] P. Drineas, A. Frieze, R. Kannan, S. Vempala, and V. Vinay. Clustering large graphs via the singular value decomposition. *Machine Learning*, 56:9–33, 2004.

[2] J.B. Kruskal and M. Wish. *Multidimensional Scaling*. Sage University Paper series on Quantitative Application in the Social Sciences, 07-011, 1978.

[3] I.J. Schoenberg. Metric spaces and positive definite functions. *Transactions of the American Mathematical Society*, 44:522–553, 1938.