Energy Clustering

Guilherme França* and Joshua T. Vogelstein†

Johns Hopkins University

Abstract

Energy statistics was proposed by Székely in the 80's inspired by the Newtonian gravitational potential from classical mechanics, and it provides a nonparametric test for equality of distributions. It was generalized to metric spaces of strong negative type, and more recently a connection with reproducing kernel Hilbert spaces was proposed. Here we consider the problem of clustering data from an energy statistics theory perspective. We provide a precise mathematical formulation yielding a quadratically constrained quadratic program (QCQP), which we show to be equivalent to kernel k-means optimization problem. Thus, our results imply a first principles derivation of kernel k-means from energy statistics. Moreover, we also consider a weighted version of energy statistics applied to clustering, which makes connection to graph partitioning problems. To find local optimizers of such QCQP we consider an iterative algorithm based on Hartigan's method, which in this case has the same computational cost as kernel k-means algorithm but usually with better clustering quality. We provide carefully designed numerical experiments showing the superiority of this method compared to kernel k-means, standard k-means, and gaussian mixture models in a variety of settings.

* guifranca@gmail.com

† jovo@jhu.edu

I. INTRODUCTION

Energy statistics [1] is based on a notion of statistical potential energy between probability distributions, in close analogy to Newton's gravitational potential in classical mechanics. It provides a nonparametric test for equality of distribution which is achieved under minimum energy. When probability distributions are different, this statistical potential energy diverges as sample size increases. Energy statistics has been applied to several goodness-of-fit hypothesis tests, multi-sample tests of equality of distributions, analysis of variance [2], nonlinear dependence tests through distance covariance and distance correlation, which generalizes the Pearson correlation coefficient, and hierarchical clustering [3] by extending Ward's method of minimum variance. We refer to [1] and references therein for an overview. Moreover, an energy statistics formulation to clustering was already proposed in [4] which greatly motivated this paper.

Recently, distance covariance was generalized from Euclidean spaces to metric spaces of strong negative type [5]. Furthermore, a unifying framework establishing an equivalence between generalized energy distances to maximum mean discrepancies (MMD), which are distances between embeddings of distributions in reproducing kernel Hilbert spaces (RKHS), was established [6]. This provides the link between concepts used in energy statistics and concepts commonly used in machine learning, and form the basis of our approach.

The clustering problem has a long history in machine learning. Perhaps the most used method is k-means [7–9], which is based on Lloyd's heuristic [7] of assigning a data point to the cluster with closest center. The only statistical information about each cluster comes from its mean and it is thus sensitive to outliers. Nevertheless, k-means works very well when data is linearly separable in Euclidean space. Gaussian mixture models (GMM) is also very commonly used for clustering, however it is strongly parametric, as k-means which is closely related.

To account for nonlinearities, kernel methods were introduced [10, 11]. A mercer kernel [12] is used to implicitly map data points to a RKHS, then clustering can be performed in the associated Hilbert space by using its inner product. However, the kernel choice remains the biggest challenge since there is no principled theory to construct a kernel for a given dataset, and usually a kernel introduces hyperparameters that need to be carefully chosen. The well-known kernel k-means optimization problem is nothing but k-means in the feature

space [11]. Furthermore, kernel k-means algorithm [13, 14] is still based on Loyd's heuristic [7] of using the mean of each cluster in the feature space. We refer the reader to [15] for a survey of clustering methods.

Although clustering from energy statistics was considered in [4], the precise optimization problem behind this approach remains elusive, as well as the connection to kernel methods. The main theoretical contribution of this paper is to fill this gap. Since the statistical potential energy is minimum when distributions are equal, the principle behind clustering is to maximize the statistical energy, enforcing probability distributions associated to each cluster to be different from one another. We provide a precise mathematical formulation to this statement leading to a quadratically constrained quadratic program (QCQP) in the associated RKHS. Our results immediately establish the connection to kernel methods used in machine learning, by showing that this QCQP is equivalent to kernel k-means optimization problem. The equivalence between kernel k-means, spectral clustering, and graph partitioning problems was already established [13, 14]. We show how these connections arise from a weighted version of energy statistics.

Our algorithmic contribution is to use Hartigan's method [16] to find local solutions of the above mentioned QCQP. This approach was also proposed in [4]. In this case, the resulting algorithm has the same time complexity as kernel k-means algorithm, however, the numerical results provide compelling evidence that this method is more accurate and robust than kernel k-means. This evidence is also supported by general theoretical results [17, 18]. Moreover, in our experiments we put in evidence the nonparametric aspect of energy statistics based clustering, which provides a family of default kernels, showing that it is able to perform accurately on datasets coming very different distributions, contrary to k-means and GMM for instance. More specifically, the proposed method performs closely to k-means and GMM on normally distributed data with balanced clusters, however, it performs considerably better on data that is not normally distributed. It also performs better than k-means and GMM in higher dimensions, even on gaussian settings, and it performs worse than GMM when we have highly unbalanced clusters.

Our work is organized as follows. In section II we introduce the necessary background on energy statistics and RKHS. Section III contains the main theoretical results of this paper, where we consider a clustering theory based on energy statistics leading to a QCQP, which is NP-hard. We also show the equivalence to kernel k-means problem. In Section IV

we generalize these results to a weighted version of energy statistics, which provides the connection with graph partitioning problems and spectral clustering. In Section V we consider a simple example in one dimension, where we propose an algorithm which requires no initialization. In section VI we briefly review kernel k-means algorithm, and propose a new iterative algorithm based on Hartigan's method to solve this QCQP. Section VII contains some carefully designed numerical experiments indicating that this algorithm outperforms kernel k-means, standard k-means, and GMM/EM algorithms. Our final conclusions are presented in section VIII.

II. BACKGROUND ON ENERGY STATISTICS AND RKHS

In this section we introduce the main concepts from energy statistics and its relation to RKHS which form the basis of our work. For more details we refer the reader to [1] and [6].

Consider random variables in $\mathbb{R}^D$ such that $X, X' \overset{iid}{\sim} P$ and $Y, Y' \overset{iid}{\sim} Q$, where P and Q are cumulative distribution functions with finite first moments. The quantity

$$\mathcal{E}(P,Q) \equiv 2\mathbb{E}\|X-Y\| - \mathbb{E}\|X-X'\| - \mathbb{E}\|Y-Y'\| \tag{1}$$

called *energy distance* [1] is rotationally invariant and nonnegative, $\mathcal{E}(P,Q) \geq 0$, where equality to zero holds if and only if $P = Q$. Above, $\|\cdot\|$ denotes the Euclidean norm in $\mathbb{R}^D$. Energy distance provides a characterization of equality of distributions, and $\mathcal{E}^{1/2}$ is a metric on the space of distributions.

The energy distance can be generalized as, for instance,

$$\mathcal{E}_\alpha(P,Q) \equiv 2\mathbb{E}\|X-Y\|^\alpha - \mathbb{E}\|X-X'\|^\alpha - \mathbb{E}\|Y-Y'\|^\alpha \tag{2}$$

where $0 < \alpha \leq 2$. This quantity is also nonnegative, $\mathcal{E}_\alpha(P,Q) \geq 0$. Furthermore, for $0 < \alpha < 2$ we have that $\mathcal{E}_\alpha(P,Q) = 0$ if and only if $P = Q$, while for $\alpha = 2$ we have $\mathcal{E}_2(P,Q) = 2\|\mathbb{E}(X) - \mathbb{E}(Y)\|^2$ which shows that equality to zero only requires equality of the means, and thus $\mathcal{E}_2(P,Q) = 0$ does not imply equality of distributions.

The energy distance (2) can be even further generalized. Let $X, Y \in \mathcal{X}$ where $\mathcal{X}$ is an arbitrary space endowed with a *semimetric of negative type* $\rho : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$, which is required to satisfy

$$\sum_{i,j=1}^{n} c_i c_j \rho(X_i, X_j) \leq 0, \tag{3}$$

where $X_i \in \mathcal{X}$ and $c_i \in \mathbb{R}$ such that $\sum_{i=1}^n c_i = 0$. Then, $\mathcal{X}$ is called a *space of negative type*. We can thus replace $\mathbb{R}^D \to \mathcal{X}$ and $\|X - Y\| \to \rho(X, Y)$ in the definition (1), obtaining the generalized energy distance

$$\mathcal{E}(P, Q) \equiv 2\mathbb{E}\rho(X, Y) - \mathbb{E}\rho(X, X') - \mathbb{E}\rho(Y, Y'). \tag{4}$$

For spaces of negative type there exists a Hilbert space $\mathcal{H}$ and a map $\varphi : \mathcal{X} \to \mathcal{H}$ such that $\rho(X, Y) = \|\varphi(X) - \varphi(Y)\|_{\mathcal{H}}^2$. This allows us to compute quantities related to probability distributions over $\mathcal{X}$ in the Hilbert space $\mathcal{H}$. Even though the semimetric ρ may not satisfy the triangle inequality, $\rho^{1/2}$ does since it can be shown to be a legitamate metric.

There is an equivalence between energy distance, commonly used in statistics, and distances between embeddings of distributions in RKHS, commonly used in machine learning. This equivalence was established in [6]. Let us first recall the definition of RKHS. Let $\mathcal{H}$ be a Hilbert space of real-valued functions over $\mathcal{X}$. A function $K : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is a reproducing kernel of $\mathcal{H}$ if it satisfies the following two conditions:

1. $h_x \equiv K(\cdot, x) \in \mathcal{H}$ for all $x \in \mathcal{X}$.

2. $\langle h_x, f \rangle_{\mathcal{H}} = f(x)$ for all $x \in \mathcal{X}$ and $f \in \mathcal{H}$.

In other words, for any $x \in \mathcal{X}$ and any function $f \in \mathcal{H}$, there is a unique $h_x \in \mathcal{H}$ that reproduces $f(x)$ through the inner product of $\mathcal{H}$. If such a *kernel* function K exists, then $\mathcal{H}$ is called a RKHS. The above two properties immediately imply that K is symmetric and positive definite. Indeed, notice that $\langle h_x, h_y \rangle = h_y(x) = K(x, y)$, and by definition $\langle h_x, h_y \rangle^* = \langle h_y, h_x \rangle$, but since the inner product is real we have $\langle h_y, h_x \rangle = \langle h_x, h_y \rangle$, or equivalently $K(y, x) = K(x, y)$. Moreover, for any $w \in \mathcal{H}$ we can write $w = \sum_{i=1}^n c_i h_{x_i}$ where $\{h_{x_i}\}_{i=1}^n$ is a basis of $\mathcal{H}$. It follows that $\langle w, w \rangle_{\mathcal{H}} = \sum_{i,j=1}^n c_i c_j K(x_i, x_j) \geq 0$, showing that the kernel is positive definite. If G is a matrix with elements $G_{ij} = K(x_i, x_j)$ this is equivalent to G being positive semidefinite, i.e. $v^\top G\, v \geq 0$ for any vector $v \in \mathbb{R}^n$.

The Moore-Aronszajn theorem [19] establishes the converse of the above paragraph. For every symmetric and positive definite function $K : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$, there is an associated RKHS $\mathcal{H}_K$ with reproducing kernel K. The map $\varphi : x \mapsto h_x \in \mathcal{H}_K$ is called the canonical *feature map*. Given a kernel K, this theorem enables us to define an embedding of a probability measure P into the RKHS as follows: $P \mapsto h_P \in \mathcal{H}_K$ such that $\int f(x) dP(x) = \langle f, h_P \rangle$ for all $f \in \mathcal{H}_K$, or alternatively $h_P \equiv \int K(\,\cdot\,, x) dP(x)$. We can now introduce the notion of

distance between two probability measures using the inner product of $\mathcal{H}_K$, which is called the maximum mean discrepancy (MMD) and is given by

$$\gamma_K(P, Q) \equiv \|h_P - h_Q\|_{\mathcal{H}_K}. \tag{5}$$

This can also be written as [20]

$$\gamma_K^2(P, Q) = \mathbb{E}K(X, X') + \mathbb{E}K(Y, Y') - 2\mathbb{E}K(X, Y) \tag{6}$$

where $X, X' \overset{iid}{\sim} P$ and $Y, Y' \overset{iid}{\sim} Q$. From the equality between (5) and (6) we also have

$$\langle h_P, h_Q \rangle_{\mathcal{H}_K} = \mathbb{E}\, K(X, Y). \tag{7}$$

Thus, in practice, we can estimate the inner product between embedded distributions by averaging the kernel function over sampled data.

The following important result shows that semimetrics of negative type and symmetric positive definite kernels are closely related [21]. Let $\rho : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ and $x_0 \in \mathcal{X}$ an arbitrary but fixed point. Define

$$K(x, y) \equiv \tfrac{1}{2} \left[\rho(x, x_0) + \rho(y, x_0) - \rho(x, y)\right]. \tag{8}$$

Then, it can be shown that K is positive definite if and only if ρ is a semimetric of negative type (3). We have a family of kernels, one for each choice of x_0. Conversely, if ρ is a semimetric of negative type and K is a kernel in this family, then

$$\begin{aligned} \rho(x, y) &= K(x, x) + K(y, y) - 2K(x, y) \\ &= \|h_x - h_y\|_{\mathcal{H}_K}^2 \end{aligned} \tag{9}$$

and the canonical feature map $\varphi : x \mapsto h_x$ is injective [6]. When these conditions are satisfied we say that the kernel K generates the semimetric ρ. If two different kernels generate the same ρ they are equivalent kernels.

Now we can state the equivalence between energy distance $\mathcal{E}$ and inner products on RKHS, which is one of the main results of [6]. If ρ is a semimetric of negative type and K a kernel that generates ρ, then replacing (9) into (4), and using (6), yields

$$\mathcal{E}(P, Q) = 2\left[\mathbb{E}\, K(X, X') + \mathbb{E}\, K(Y, Y') - 2\mathbb{E}\, K(X, Y)\right] = 2\gamma_K^2(P, Q). \tag{10}$$

Due to (5) we can compute the energy distance using the inner product of $\mathcal{H}_K$.

Finally, let us recall the main formulas from energy statistics for the test statistic of equality of distributions [1]. Assume we have data $\mathbb{X} = \{x_1, \ldots, x_n\}$ where $x_i \in \mathcal{X}$, and $\mathcal{X}$ is a space of negative type. Consider a disjoint partition $\mathbb{X} = \bigcup_{j=1}^{k} \mathcal{C}_j$, with $\mathcal{C}_i \cap \mathcal{C}_j = \emptyset$. Each expectation in (4) can be computed through the function

$$g(\mathcal{C}_i, \mathcal{C}_j) \equiv \frac{1}{n_i n_j} \sum_{x \in \mathcal{C}_i} \sum_{y \in \mathcal{C}_j} \rho(x, y) \tag{11}$$

where $n_i = |\mathcal{C}_i|$ is the number of elements in $\mathcal{C}_i$. The *within energy dispersion* is defined by

$$W \equiv \sum_{j=1}^{k} \frac{n_j}{2} g(\mathcal{C}_j, \mathcal{C}_j) \tag{12}$$

and the *between-sample energy statistic* is defined by

$$S \equiv \sum_{1 \le i < j \le k} \frac{n_i n_j}{2n} \left[2g(\mathcal{C}_i, \mathcal{C}_j) - g(\mathcal{C}_i, \mathcal{C}_i) - g(\mathcal{C}_j, \mathcal{C}_j) \right] \tag{13}$$

where $n = \sum_{j=1}^{k} n_j$. Given a set of distributions $\{P_j\}_{j=1}^{k}$, where $x \in \mathcal{C}_j$ if and only if $x \sim P_j$, the quantity S provides a *nonparametric test statistic* for equality of distributions [1]. When the sample size is large enough, $n \to \infty$, under the null hypothesis $H_0 : P_1 = P_2 = \cdots = P_k$ we have that $S \to 0$, and under the alternative hypothesis $H_1 : P_i \neq P_j$ for at least two $i \neq j$, we have that $S \to \infty$. This test is nonparametric in the sense that it does not make any assumptions about the distributions P_j.

One can make the analogy that points $x \in \mathcal{C}_j$ form a massive body whose total mass is characterized by the distribution function P_j. The quantity S is thus a potential energy of the from $S(P_1, \ldots, P_k)$ which measures how different the distribution of these masses are, and achieves the ground state $S = 0$ when all bodies have the same mass distribution. The potential energy S increases as bodies have different mass distributions.

III. CLUSTERING BASED ON ENERGY STATISTICS

This section contains the main theoretical results of this paper, where we formulate an optimization problem for clustering based on energy statistics and RKHS introduced in the previous section.

Due to the test statistic (13) for equality of distributions, the obvious criterion for clustering data is to maximize S which makes each cluster as different as possible from

the other ones. In other words, given a set of points coming from different probability distributions, S should attain a maximum when each point is correctly classified as belonging to the cluster associated to its probability distribution. The following straightforward result shows that maximizing (13) is, however, equivalent to minimizing (12) which has a more convenient form.

Proposition 1. *Let $\mathbb{X} = \{x_1, \dots, x_n\}$ where each data point x_i lives in a space $\mathcal{X}$ endowed with a semimetric $\rho : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ of negative type* (3)*. For a fixed integer k, the partition $\mathbb{X} = \bigcup_{j=1}^{k} \mathcal{C}_j$, where $\mathcal{C}_i \cap C_j = \emptyset$ for all $i \neq j$, maximizes* (13) *if and only if*

$$\min_{\mathcal{C}_1,\dots,C_k} W(\mathcal{C}_1, \dots, \mathcal{C}_k), \tag{14}$$

where W is given by (12)*.*

Proof. From (12) and (13) we have

$$\begin{aligned} S + W &= \frac{1}{2n} \sum_{\substack{i,j=1 \\ i \neq j}}^{k} n_i n_j g(\mathcal{C}_i, \mathcal{C}_j) + \frac{1}{2n} \sum_{i=1}^{k} \left[n - \sum_{j \neq i = 1}^{k} n_j \right] n_i g(\mathcal{C}_i, \mathcal{C}_i) \\ &= \frac{1}{2n} \sum_{i,j=1}^{k} n_i n_j g(\mathcal{C}_i, \mathcal{C}_j) = \frac{1}{2n} \sum_{x \in \mathbb{X}} \sum_{y \in \mathbb{X}} \rho(x, y) = \frac{n}{2} g(\mathbb{X}, \mathbb{X}). \end{aligned} \tag{15}$$

Note that the right hand side of this equation only depends on the pooled data, so it is a constant independent of the choice of partition. Therefore, maximizing S over the choice of partition is equivalent to minimizing W. □

For a given k, the clustering problem amounts to finding the best partition of the data by minimizing W. Notice that this is a hard clustering problem as partitions are disjoint. The problem (14) based on energy statistics was already proposed in [4]. However, it is important to note that this is equivalent to maximizing (13) which is the test statistic for equality of distributions. In the following we show what is the explicit optimization problem behind (14).

We now formulate (14) in the corresponding RKHS. Based on (8) and (9), assume that the kernel $K : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ generates ρ. Let us define the Gram matrix

$$G \equiv \begin{pmatrix} K(x_1, x_1) & K(x_1, x_2) & \cdots & K(x_1, x_n) \\ K(x_2, x_1) & K(x_2, x_2) & \cdots & K(x_2, x_n) \\ \vdots & \vdots & \ddots & \vdots \\ K(x_n, x_1) & K(x_n, x_2) & \cdots & K(x_n, x_n) \end{pmatrix}. \tag{16}$$

Let $Z \in \{0,1\}^{n\times k}$ be the label matrix, with only one nonvanishing entry per row, indicating to which cluster (column) each point (row) belongs to. This matrix satisfies $Z^\top Z = D$ where the diagonal matrix $D = \mathrm{diag}(n_1, \ldots, n_k)$ contains the number of points in each cluster. Let us also introduce the rescaled matrix $Y \equiv ZD^{-1/2}$. In component form they are given by

$$Z_{ij} \equiv \begin{cases} 1 & \text{if } x_i \in \mathcal{C}_j \\ 0 & \text{otherwise} \end{cases} \qquad Y_{ij} \equiv \begin{cases} \frac{1}{\sqrt{n_j}} & \text{if } x_i \in \mathcal{C}_j \\ 0 & \text{otherwise} \end{cases}. \tag{17}$$

Throughout the paper, we use the notation $M_{i\bullet}$ to denote the ith row of a matrix M, and $M_{\bullet j}$ denotes its jth column. Our next result shows that the optimization problem (14) is NP-hard since it is a quadratically constrained quadratic program (QCQP).

Proposition 2. *The problem* (14) *is equivalent to*

$$\max_Y \mathrm{Tr}\left(Y^\top G\, Y\right) \qquad s.t.\ Y \geq 0,\ Y^\top Y = I,\ YY^\top e = e, \tag{18}$$

where $e = (1, 1, \ldots, 1)^\top \in \mathbb{R}^n$ *is the all-ones vector, and* G *is the Gram matrix* (16).

Proof. From (9), (11), and (12) we have

$$W(\mathcal{C}_1, \ldots, \mathcal{C}_k) = \frac{1}{2}\sum_{j=1}^{k} \frac{1}{n_j} \sum_{x,y\in\mathcal{C}_j} \rho(x,y) = \sum_{j=1}^{k}\sum_{x\in\mathcal{C}_j}\left(K(x,x) - \frac{1}{n_j}\sum_{y\in\mathcal{C}_j} K(x,y)\right). \tag{19}$$

Note that the first term is global so it does not contribute to the optimization problem. Therefore, minimizing (19) is equivalent to

$$\max_{\mathcal{C}_1,\ldots,\mathcal{C}_k} \sum_{j=1}^{k} \frac{1}{n_j} \sum_{x,y\in C_j} K(x,y). \tag{20}$$

But

$$\sum_{x,y\in\mathcal{C}_j} K(x,y) = \sum_{p=1}^{n}\sum_{q=1}^{n} Z_{pj} Z_{qj} G_{pq} = (Z^\top G\, Z)_{jj}, \tag{21}$$

where we used the definitions (16) and (17). Thus, the objective function in (20) is equal to $\mathrm{Tr}\left(D^{-1}Z^\top G Z\right)$. Now we can use the cyclic property of the trace, and by the definition of the matrix Z in (17), we obtain the following integer programing problem:

$$\max_Z \mathrm{Tr}\left(\left(ZD^{-1/2}\right)^\top G\left(ZD^{-1/2}\right)\right) \quad \text{s.t. } Z_{ij} \in \{0,1\},\ \textstyle\sum_{j=1}^{k} Z_{ij} = 1,\ \sum_{i=1}^{n} Z_{ij} = n_j. \tag{22}$$

Now we write this in terms of the matrix $Y = ZD^{-1/2}$. The objective function immediately becomes $\mathrm{Tr}\left(Y^\top G\, Y\right)$. Notice that the above constraints imply that $Z^T Z = D$,

where $D = \text{diag}(n_1, \ldots, n_k)$, which in turn gives $D^{-1/2}Y^TYD^{-1/2} = D$, or $Y^\top Y = I$. Also, every entry of Y is positive by definition, $Y \geq 0$. Now it only remains to show the last constraint in (18), which comes from the last constraint in (22). In matrix form this reads $Z^Te = De$. Replacing $Z = YD^{1/2}$ we have $Y^\top e = D^{1/2}e$. Multiplying this last equation on the left by Y, and noticing that $YD^{1/2}e = Ze = e$, we finally obtain $YY^\top e = e$. Therefore, the optimization problem (22) is equivalent to (18) . □

Based on Proposition 2, to group data $\mathbb{X} = \{x_1, \ldots, x_n\}$ into k clusters we first compute the Gram matrix G and then solve the optimization problem (18) for $Y \in \mathbb{R}^{n\times k}$. The ith row of Y will contain a single nonzero element in some jth column, indicating that $x_i \in \mathcal{C}_j$. Problem (18) is NP-hard and there are few methods available to solve it directly, which is computational prohibitive even for small datasets. However, one can find approximate solutions by relaxing some of the constraints, or obtaining a relaxed SDP version of it. For instance, the relaxed problem

$$\max_{Y} \text{Tr}\left(Y^\top G\, Y\right) \quad \text{s.t. } Y^\top Y = I \tag{23}$$

has a well-known closed form solution $Y^\star = UR$, where the columns of U contain the leading k eigenvectors of G corresponding to the k largest eigenvalues $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_k$, and $R \in \mathbb{R}^{k\times k}$ is an arbitrary orthogonal matrix. The resulting optimal objective function is given by $\max \text{Tr}\left(Y^{\star\top} G\, Y^\star\right) = \sum_{i=1}^k \lambda_i$. One might then normalize and threshold the rows of $Y^\star$, or better, following [22] one can normalize the rows of $Y^\star$ and apply standard k-means on this matrix where each row is considered as a data point. This is the procedure done in spectral clustering on a graph Laplacian matrix obtained through a similarity matrix. However, computing eigenvectors of very large matrices can be prohibitively expensive and usually iterative methods are preferred. Moreover, this procedure is only guaranteed to converge if the normalized $Y^\star$ is positive semidefinite. In Section VI we will propose an iterative method to directly find approximate solutions to (18) that is guaranteed to converge and does not require that the similarities $K(x_i, x_j)$ be nonnegative.

The clustering problem (18) based on energy statistics is valid for data living in an *arbitrary* space of negative type where a semimetric ρ, and thus the kernel (8), are assumed to be known. This clustering formulation from energy statistics is *nonparametric* in the sense that it does not make assumptions about the distribution of the data, contrary to k-means and GMM formulations, for example. If one uses the standard energy distance

defined in (1) we have $\rho(x, y) = \|x - y\|$ and this fixes the kernel through (8). In the same way, for data living in a more general metric space $(\mathcal{X}, \rho)$ the corresponding semimetric ρ fixes the kernel. In practice, however, the clustering quality strongly depend on the choice of a suitable ρ which measures the similarity between data points, and is equivalent to choosing an appropriate kernel. The energy distance (2) fixes a family of choices $\rho(x, y) = \|x - y\|^\alpha$ with $0 < \alpha \leq 2$. Nevertheless, if prior information is available one may choose a more appropriate ρ.

Relation to Kernel k-Means

One may wonder how energy statistics clustering relates to the well-known kernel k-means problem[1] which is extensively used in machine learning. For a positive semidefinite Gram matrix G, as defined in (16), there exists a map $\varphi : \mathcal{X} \to \mathcal{H}_K$ such that $K(x, y) = \varphi(x)^\top \varphi(y)$. The kernel k-means optimization problem, in feature space, is defined by

$$\min_{\mathcal{C}_1,\ldots,\mathcal{C}_k} \left\{ J(\mathcal{C}_1, \ldots, \mathcal{C}_k) \equiv \sum_{j=1}^{k} \sum_{x \in \mathcal{C}_j} \|\varphi(x) - \varphi(\mu_j)\|^2 \right\} \tag{24}$$

where $\mu_j = \frac{1}{n_j} \sum_{x \in \mathcal{C}_j} x$ is the mean of cluster $\mathcal{C}_j$ in the ambient space. Notice that the above objective function is strongly tied to the idea of minimizing distances between points and cluster centers, which arises from k-means objective function. [13, 14] show that problem (24) is equivalent to a QCQP in the same form as (18). The next result makes this explicit, showing that (14) and (24) are actually equivalent.

Proposition 3. *For a fixed kernel, the clustering optimization problem* (14) *based on energy statistics is equivalent to the kernel k-means optimization problem* (24)*, and both are equivalent to* (18)*.*

Proof. Notice that $\|\varphi(x) - \varphi(\mu_j)\|^2 = \varphi(x)^\top \varphi(x) - 2\varphi(x)^\top \varphi(\mu_j) + \varphi(\mu_j)^\top \varphi(\mu_j)$, therefore

$$J = \sum_{j=1}^{k} \sum_{x \in \mathcal{C}_j} \left(K(x, x) - \frac{2}{n_j} \sum_{y \in \mathcal{C}_j} K(x, y) + \frac{1}{n_j^2} \sum_{y,z \in \mathcal{C}_j} K(y, z) \right). \tag{25}$$

The first term is global so it does not contribute to the optimization problem. Notice that the third term gives $\sum_{x \in \mathcal{C}_j} \frac{1}{n_j^2} \sum_{y,z \in \mathcal{C}_j} K(y, z) = \frac{1}{n_j} \sum_{y,z \in \mathcal{C}_j} K(y, z)$, which is the same as the

[1] When we refer to kernel k-means problem we mean specifically the optimization problem (24), which should not be confused with kernel k-means algorithm that is just one possible recipe to solve (24).

second term. Thus, problem (24) is equivalent to

$$\min_{\mathcal{C}_1,\ldots,\mathcal{C}_k} \left\{ J(\mathcal{C}_1,\ldots,\mathcal{C}_k) = \max_{\mathcal{C}_1,\ldots,\mathcal{C}_k} \sum_{j=1}^{k} \frac{1}{n_j} \sum_{x,y\in\mathcal{C}_j} K(x,y) \right\} \tag{26}$$

which is exactly the same as (20) from the energy statistics formulation. Therefore, once the kernel K is fixed, the function W given by (12) is the same as J in (24). The remaining of the proof proceeds as already shown in the proof of Proposition 2, leading to (18). □

The above result shows that kernel k-means problem is equivalent to the clustering problem formulated in the energy statistics framework. In this vein, kernel k-means is part of a statistical framework where distances between probability distributions are maximized. The kernel function K arises from a semimetric ρ of negative type which defines the energy distance between distributions.

As shown in [13, 14], kernel k-means, spectral clustering, and graph partitioning problems such as ratio association, ratio cut, and normalized cut are all equivalent to a QCQP of the form (18). Thus one can use kernel k-means algorithm to solve these problems as well. This correspondence involves a weighted version of (18) that we demonstrate in the following.

IV. CLUSTERING BASED ON WEIGHTED ENERGY STATISTICS

We generalize the formulas from energy statistics to incorporate weights associated to each data point. Let $w(x)$ be a weight function associated to point $x \in \mathcal{X}$. We can generalize (11) as follows:

$$g(\mathcal{C}_i,\mathcal{C}_j) \equiv \frac{1}{s_i s_j} \sum_{x\in\mathcal{C}_i} \sum_{y\in\mathcal{C}_j} w(x)w(y)\rho(x,y), \qquad s_i \equiv \sum_{x\in\mathcal{C}_i} w(x). \tag{27}$$

Now we replace this function in the formulas (12) and (13), with $n_i \to s_i$ and $n \to s$ where $s = \sum_{j=1}^{k} s_j$, to obtain a weighted version of energy test statistic. With these changes, Proposition 1 remains the unaltered, so the clustering problem becomes

$$\min_{\mathcal{C}_1,\ldots,\mathcal{C}_k} \left\{ W(\mathcal{C}_1,\ldots,\mathcal{C}_k) \equiv \sum_{j=1}^{k} \frac{s_j}{2} g(\mathcal{C}_j,\mathcal{C}_j) \right\} \tag{28}$$

where now g is given by (27). Let us define the following matrices and vector:

$$Y_{ij} \equiv \begin{cases} \frac{1}{\sqrt{s_j}} & \text{if } x_i \in \mathcal{C}_j \\ 0 & \text{otherwise} \end{cases}, \qquad \mathcal{W} \equiv \operatorname{diag}(w_1,\ldots,w_n), \qquad H \equiv \mathcal{W}^{1/2}Y, \qquad \omega \equiv \mathcal{W}e, \tag{29}$$

where $w_i = w(x_i)$ and $e \in \mathbb{R}^n$ is the all-ones vector. Now we can show the analogous of Proposition 2 to the case of (28).

Proposition 4. *The weighted version of energy statistics clustering given by problem* (28) *is equivalent to*

$$\max_{H} \operatorname{Tr}\left\{H^\top (\mathcal{W}^{1/2} G \mathcal{W}^{1/2}) H\right\} \qquad s.t.\ H \geq 0,\ H^\top H = I,\ HH^\top \omega = \omega, \tag{30}$$

where G is the Gram matrix (16) *and the other quantities are defined in* (29).

Proof. Replacing (9) and eliminating the global terms which do not contribute, the optimization problem (28) becomes

$$\max_{\mathcal{C}_1,\ldots,\mathcal{C}_k} \sum_{j=1}^{k} \frac{1}{s_j} \sum_{x\in\mathcal{C}_j} \sum_{y\in\mathcal{C}_j} w(x)w(y)K(x,y). \tag{31}$$

This objective function can be written as

$$\begin{aligned}
\sum_{j=1}^{k} \frac{1}{s_j} \sum_{p=1}^{n}\sum_{q=1}^{n} w_p w_q Z_{pj} Z_{qj} G_{pq} &= \sum_{j=1}^{k}\sum_{p=1}^{n}\sum_{q=1}^{n} \frac{Z_{jp}^\top \sqrt{w_p}}{\sqrt{s_j}} w_p^{1/2} G_{pq} w_q^{1/2} \frac{\sqrt{w_q} Z_{qj}}{\sqrt{s_j}} \\
&= \sum_{j=1}^{k} \left(H^\top \mathcal{W}^{1/2} G \mathcal{W}^{1/2} H\right)_{jj} \\
&= \operatorname{Tr}\left(H^\top \mathcal{W}^{1/2} G \mathcal{W}^{1/2} H\right).
\end{aligned} \tag{32}$$

To obtain the constraints, note that $H_{ij} \geq 0$ by definition, and

$$(H^\top H)_{ij} = \sum_{\ell=1}^{n} Y_{\ell i} \mathcal{W}_{\ell\ell} Y_{\ell j} = \frac{1}{\sqrt{s_i}\sqrt{s_j}} \sum_{\ell=1}^{n} w_\ell Z_{\ell i} Z_{\ell j} = \frac{\delta_{ij}}{s_i} \sum_{\ell=1}^{n} w_\ell Z_{\ell i} = \delta_{ij}, \tag{33}$$

therefore $H^\top H = I$. This is a constraint on the rows of H. To obtain a condition on its columns observe that

$$\left(H^\top H\right)_{pq} = \sqrt{w_p w_q} \sum_{j=1}^{k} \frac{Z_{pj} Z_{qj}}{s_j} = \begin{cases} \dfrac{\sqrt{w_p w_q}}{s_i} & \text{if both } x_p, x_q \in \mathcal{C}_i \\ 0 & \text{otherwise.} \end{cases} \tag{34}$$

Therefore, $(H^\top H \mathcal{W}^{1/2})_{pq} = \sqrt{w_p}\, w_q s_i^{-1}$ if both points x_p and x_q belong to the same cluster, which we denote by $\mathcal{C}_i$ for some $i \in \{1,\ldots,k\}$, and $(H^\top H \mathcal{W}^{1/2})_{pq} = 0$ otherwise. Thus, the pth line of this matrix is nonzero only on entries corresponding to points that are in the same cluster as x_p. If we sum over the columns of this line we obtain $\sqrt{w_p} s_i^{-1} \sum_{q=1}^{n} w_q Z_{qi} = \sqrt{w_p}$, or equivalently $HH^\top \mathcal{W}^{1/2} e = \mathcal{W}^{1/2} e$, which gives the constraint $HH^\top \omega = \omega$. □

Connection with Graph Partitioning

The relation between kernel k-means and graph partitioning problems is known [13, 14]. For conciseness, we repeat a similar analysis due to the relation of these problems to energy statistics and RKHS, which provides a different perspective.

Consider a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{A})$ where $\mathcal{V}$ is the set of vertices, $\mathcal{E}$ the set of edges, and $\mathcal{A}$ is an affinity matrix of the graph that measures the similarities between pairs of nodes. Thus, $\mathcal{A}_{ij} \neq 0$ if $(i,j) \in \mathcal{E}$, and $\mathcal{A}_{ij} = 0$ otherwise. We also associate weights to every vertex, $w_i = w(i)$ for $i \in \mathcal{V}$, and let $s_j = \sum_{i \in \mathcal{C}_j} w_i$, where $\mathcal{C}_j \subseteq \mathcal{V}$ is one partition of $\mathcal{V}$. Let

$$\text{links}(\mathcal{C}_\ell, \mathcal{C}_m) \equiv \sum_{i \in \mathcal{C}_\ell, j \in \mathcal{C}_m} A_{ij}. \tag{35}$$

We want to partition the set of vertices $\mathcal{V}$ into k disjoint subsets, $\mathcal{V} = \bigcup_{j=1}^{k} \mathcal{C}_j$. The generalized ratio association problem is given by

$$\max_{\mathcal{C}_i, \ldots, \mathcal{C}_k} \sum_{j=1}^{k} \frac{\text{links}(\mathcal{C}_j, \mathcal{C}_j)}{s_j} \tag{36}$$

and maximizes the within cluster association. The generalized ratio cut problem

$$\min_{\mathcal{C}_i, \ldots, \mathcal{C}_k} \sum_{j=1}^{k} \frac{\text{links}(\mathcal{C}_j, \mathcal{V} \backslash \mathcal{C}_j)}{s_j} \tag{37}$$

minimizes the cut between clusters. These two problems are equivalent, in analogous way as minimizing (12) is equivalent to maximizing (13) as shown in Proposition 1. Here this is due to the equality $\text{links}(\mathcal{C}_j, \mathcal{V} \backslash \mathcal{C}_j) = \text{links}(\mathcal{C}_j, \mathcal{V}) - \text{links}(\mathcal{C}_j, \mathcal{C}_j)$. Several graph partitioning methods [23–26] can be seen as a particular case of (36) or (37).

Consider (36), whose objective function can be written as

$$\sum_{j=1}^{k} \frac{1}{s_j} \sum_{p \in \mathcal{C}_j} \sum_{q \in \mathcal{C}_j} \mathcal{A}_{pq} = \sum_{j=1}^{k} \sum_{p=1}^{n} \sum_{q=1}^{n} \frac{Z_{jp}^\top}{\sqrt{s_j}} \mathcal{A}_{pq} \frac{Z_{qj}}{\sqrt{s_j}} = \text{Tr}\left(Y^\top \mathcal{A} Y\right), \tag{38}$$

with Z defined in (17) and Y in (29). To make the analogy with (30) explicit, problem (36) is equivalent to

$$\max_{H} \text{Tr}\left(H^\top \mathcal{W}^{-1/2} \mathcal{A} \mathcal{W}^{-1/2} H\right) \qquad \text{s.t. } H \geq 0,\, H^\top H = I,\, H H^\top \omega = \omega. \tag{39}$$

Therefore, this is exactly the same problem as weighted energy statistics clustering (30) with $G = \mathcal{W}^{-1} \mathcal{A} \mathcal{W}^{-1}$. Assuming this matrix is positive semidefinite, this generates a semimetric

(9) for graphs given by

$$\rho(i,j) = \frac{\mathcal{A}_{ii}}{w_i^2} + \frac{\mathcal{A}_{jj}}{w_j^2} - \frac{2\mathcal{A}_{ij}}{w_i w_j} \qquad \text{or} \qquad \rho(i,j) = -\frac{2\mathcal{A}_{ij}}{w_i w_j} \tag{40}$$

for vertices $i, j \in \mathcal{V}$, and where in the second equation we assume the graph has no self-loops, i.e. $\mathcal{A}_{ii} = 0$. Using (40) in (11)–(13) allows one to use energy statistics theory for inference on graphs. Above, the weight $w_i = w(i)$ of node $i \in \mathcal{V}$ can be, for instance, its degree $w_i = d(i)$.

V. TWO-CLASS PROBLEM IN ONE DIMENSION

Before stating a general algorithm to solve (18) let us first consider the simplest possible case which is one-dimensional data and a two-class problem. This will be useful to test energy statistics clustering on a simple setting.

Fixing $\rho(x,y) = |x-y|$, according to (1), we can actually compute (11) in $\mathcal{O}(n \log n)$ and find a direct solution to (14). This is done by noticing that

$$\begin{aligned} |x-y| &= (x-y)\mathbb{1}_{x \geq y} - (x-y)\mathbb{1}_{x<y} \\ &= x\left(\mathbb{1}_{x \geq y} - \mathbb{1}_{x<y}\right) + y\left(\mathbb{1}_{y>x} - \mathbb{1}_{y \leq x}\right) \end{aligned} \tag{41}$$

where we have the indicator function defined by $\mathbb{1}_A = 1$ if A is true, and $\mathbb{1}_A = 0$ otherwise. Let $\mathcal{C}$ be a partition with n elements. Using the above distance in (11) we have

$$g\left(\mathcal{C}, \mathcal{C}\right) = \frac{1}{n^2} \sum_{x \in \mathcal{C}} \sum_{y \in \mathcal{C}} x\left(\mathbb{1}_{x \geq y} + \mathbb{1}_{y>x} - \mathbb{1}_{x \geq y} - \mathbb{1}_{x<y}\right). \tag{42}$$

The sum over y can be eliminated since each term in the parenthesis is simply counting the number of elements in $\mathcal{C}$ that satisfy the condition of the indicator function. Assuming that we first order the data in $\mathcal{C}$, obtaining $\bar{\mathcal{C}} = [x_j \in \mathcal{C} : x_1 \leq x_2 \leq \cdots \leq x_n]$, we can write (42) in the following simple form:

$$g\left(\bar{\mathcal{C}}, \bar{\mathcal{C}}\right) = \frac{2}{n^2} \sum_{\ell=1}^{n} (2\ell - 1 - n) x_\ell. \tag{43}$$

Note that the cost of computing (43) is $\mathcal{O}(n)$ and the cost of sorting the data is at the most $\mathcal{O}(n \log n)$. Assuming that each partition is ordered, $\mathbb{X} = \bigcup_{j=1}^{k} \bar{\mathcal{C}}_j$, the within energy dispersion (12) can be written as

$$W\left(\bar{\mathcal{C}}_1, \ldots, \bar{\mathcal{C}}_k\right) = \sum_{j=1}^{k} \sum_{\ell=1}^{n_j} \frac{2\ell - 1 - n_j}{n_j} x_\ell. \tag{44}$$

Algorithm 1 Approximate solution to (14) for a two-class problem in one dimension.

input data $\mathbb{X}$

output label matrix Z

1: sort $\mathbb{X}$ obtaining $\bar{\mathbb{X}} = [x_1, \ldots, x_n]$

2: **for** $j \in [1, \ldots, n]$ **do**

3: Let $\bar{\mathcal{C}}_1^{(j)} = [x_i : i = 1, \ldots, j]$ and $\bar{\mathcal{C}}_2^{(j)} = [x_i : i = j+1, \ldots, n]$

4: $W^{(j)} \leftarrow W\big(\bar{\mathcal{C}}_1^{(j)}, \bar{\mathcal{C}}_2^{(j)}\big)$ from (44)

5: **end for**

6: $j^\star \leftarrow \arg\min_j W^{(j)}$

7: $Z_{j\bullet} \leftarrow (1, 0)$ if $j \leq j^\star$, and $Z_{j\bullet} \leftarrow (0, 1)$ otherwise, for $j = 1, \ldots, n$

For a two-class problem we can use (44) to cluster the data through a simple algorithm as follows. We first order the entire dataset, $\mathbb{X} \to \bar{\mathbb{X}}$. Then we compute (44) for each possible split of $\bar{\mathbb{X}}$ and pick the point which gives the minimum value of W. This procedure is described in Algorithm 1. Note that this method does not require any initialization, however, it only works for one-dimensional data with Euclidean distance. The total complexity of the algorithm is $\mathcal{O}(n \log n + n^2) = \mathcal{O}(n^2)$.

Assuming the true label matrix Z is available, a direct measure of how different the estimated matrix $\hat{Z}$ is from Z, up to label permutations, is given by

$$\text{accuracy}(\hat{Z}) \equiv \max_{\sigma} \frac{1}{n} \sum_{i=1}^{n} \sum_{j=1}^{k} \hat{Z}_{i\sigma(j)} Z_{ij} \tag{45}$$

where σ is a permutation of the k cluster groups. The accuracy is always between $[0, 1]$, where 1 corresponds to all points correctly clustered, and 0 to all points wrongly clustered. For a two-class problem with balanced clusters, the value $1/2$ correspond to chance.

Let us consider two simple experiments with equal number of points in each cluster. We plot the accuracy (45) versus the number of points in each cluster. The data is clustered using Algorithm 1, GMM through EM algorithm, and k-means++ algorithm. We use the initialization from k-means++ [27] also for GMM. We run the algorithms 100 times choosing the result with best objective function value. Notice that Algorithm 1 requires no initialization so we only run it once. Moreover, for each case we sample 100 times and show

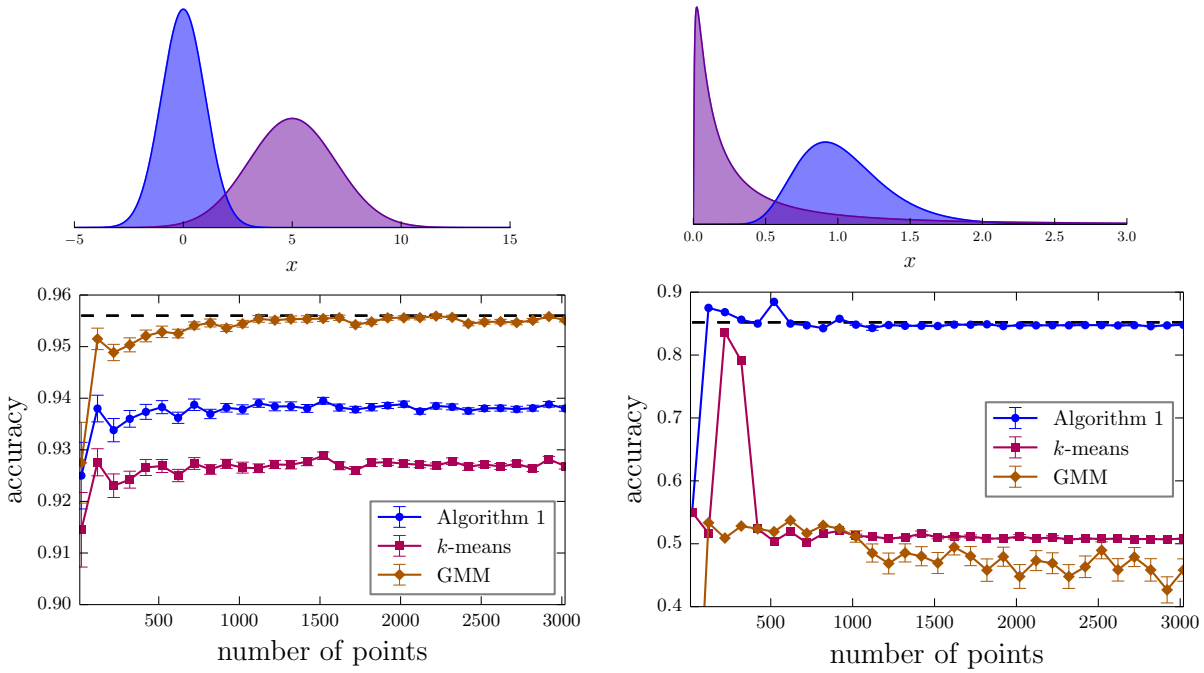


FIG. 1. Energy statistics clustering by Algorithm 1 compared to k-means and GMM/EM. We have the same number of points in both clusters, and for each case we sample 100 times from the distributions shown in the histograms. We plot the average value of (45) versus the total number of points (error bars are standard error). The dashed line indicates the best possible classification accuracy computed from Bayes error. (a) Data coming from (46), where the optimal accuracy is ≈ 0.956. (b) Data from (47), where the optimal accuracy is ≈ 0.852.

the average accuracy with error bars indicating the standard error. In Fig. 1a we have data sampled from two normal distributions with equal number of points in each cluster,

$$x \sim \mathcal{N}\big(\mu_i, \sigma_i^2\big) \quad \text{with } \mu_1 = 0,\ \sigma_1 = 1 \text{ and } \mu_2 = 5,\ \sigma_2 = 2. \tag{46}$$

For these distributions the optimal accuracy obtained from Bayes classification error is ≈ 0.956 which is indicated by the dashed line in the plot. We see that the three methods perform closely. As expected, GMM has a slight advantage over the other methods since it corresponds to the true model of the data. Energy statistics performs slightly better than k-means. On the other hand, in Fig. 1b we consider two clusters with lognormal distributions,

$$\log x \sim \mathcal{N}\big(\mu_i, \sigma_i^2\big) \quad \text{with } \mu_1 = 0,\ \sigma_1 = 0.3 \text{ and } \mu_2 = -1.5,\ \sigma_2 = 1.5. \tag{47}$$

The optimal classification accuracy from Bayes error is ≈ 0.852. In this case, Algorithm 1 provides a very accurate clustering while GMM and k-means basically cluster at chance. Sometimes GMM/EM was unable to estimate the parameters thus giving zero accuracy. The two simple experiments of Fig. 1 illustrate how energy statistics clustering is nonparametric, being able to provide high quality clustering in settings where data comes from very different distributions.

VI. ITERATIVE ALGORITHMS FOR ENERGY STATISTICS CLUSTERING

In this section we introduce an iterative algorithm to find a local maximizer of (18). Due to Proposition 3 we can also find an approximate solution by the well-known kernel k-means algorithm (see [13, 14]), which for convenience will also be restated in the present context.

Consider the optimization problem written in the form (20) as follows:

$$\max_{\{\mathcal{C}_1,\ldots,\mathcal{C}_k\}} \left\{ Q = \sum_{j=1}^{k} \frac{Q_j}{n_j} \right\}, \qquad Q_j = \sum_{x,y \in \mathcal{C}_j} K(x,y), \tag{48}$$

where Q_j represents an internal energy cost of cluster $\mathcal{C}_j$, and Q is the total energy cost where each Q_j is weighted by the inverse of the number of its elements. For a data point x_i we denote its own energy cost with the entire cluster $\mathcal{C}_\ell$ by

$$Q_\ell(x_i) \equiv \sum_{y \in \mathcal{C}_\ell} K(x_i, y) = G_{i\bullet} \cdot Z_{\bullet \ell}, \tag{49}$$

where we recall that $G_{i\bullet}$ ($G_{\bullet i}$) denotes the ith row (column) of matrix G.

Kernel k-Means Algorithm

To optimize the kernel k-means objective function (25) we remove the global term and define the function

$$J^{(\ell)}(x_i) \equiv \frac{1}{n_\ell^2} Q_\ell - \frac{2}{n_\ell} Q_\ell(x_i) \tag{50}$$

which represents a cost depending on point x_i and cluster $\mathcal{C}_\ell$. One thus assigns x_i to cluster $\mathcal{C}_{j^\star}$ according to $j^\star = \arg\min_\ell J^{(\ell)}(x_i)$, for $\ell = 1, \ldots, k$. This procedure is performed for every data point and repeated until convergence, i.e. until no new assignments are made. The entire procedure is described in Algorithm 2. It can be shown that this algorithm converges provided G is positive semidefinite.

Algorithm 2 Kernel k-means algorithm to find a local solution to (18).

input number of clusters k, Gram matrix G, initial label matrix $Z = Z_0$

output label matrix Z

1: $q \leftarrow (Q_1, \ldots, Q_k)^\top$ have the costs of each cluster, according to (48)
2: $n \leftarrow (n_1, \ldots, n_k)^\top$ have the number of points in each cluster, obtained from $D = Z^\top Z$
3: **repeat**
4: **for** $i = 1, \ldots, n$ **do**
5: let j be such that $x_i \in \mathcal{C}_j$
6: $j^\star \leftarrow \arg\min_\ell J^{(\ell)}(x_i)$ according to (50), for $\ell = 1, 2, \ldots, k$
7: **if** $j^\star \neq j$ **then**
8: move x_i to $\mathcal{C}_{j^\star}$: $Z_{ij} \leftarrow 0$ and $Z_{ij^\star} \leftarrow 1$
9: update n: $n_j \leftarrow n_j - 1$ and $n_{j^\star} \leftarrow n_{j^\star} + 1$
10: update q: $q_j \leftarrow q_j - 2Q_j(x_i)$ and $q_{j^\star} \leftarrow q_{j^\star} + 2Q_{j^\star}(x_i)$
11: **end if**
12: **end for**
13: **until** convergence

Notice that to compute the first term in (50) requires $\mathcal{O}(n_\ell)$ operations, and although the second term requires $\mathcal{O}(n_\ell^2)$ it only needs to be computed once outside loop through data points (step 1). Therefore, the time complexity Algorithm 2 is $\mathcal{O}(nk \max_\ell n_\ell) = \mathcal{O}(kn^2)$. For a sparse Gram matrix G having n' nonzero elements this complexity can be further reduced to $\mathcal{O}(kn')$.

Hartigan's Method for Energy Statistics Clustering

We now consider an algorithm based on Hartigan's method [16] applied to (48). This gives a local solution to (18). The method is based on the change in the within energy statistic when moving a given data point to another cluster. Suppose point $x_i \in \mathcal{X}$ is currently assigned to cluster $\mathcal{C}_j$, yielding a total energy cost function (48) denoted by $Q^{(j)}$. Let us consider the change in the total within energy by moving x_i to $\mathcal{C}_\ell$. Denote this new

Algorithm 3 Hartigan's method to find a local solution to (18).

input number of clusters k, Gram matrix G, initial label matrix $Z = Z_0$

output label matrix Z

$q \leftarrow (Q_1, \ldots, Q_k)^\top$ have the energy costs of each cluster, according to (48)
$n \leftarrow (n_1, \ldots, n_k)^\top$ have the number of points in each cluster, obtained from $D = Z^\top Z$
repeat
  for $i = 1, \ldots, n$ **do**
    let j be such that $x_i \in \mathcal{C}_j$
    $j^\star \leftarrow \arg\max_\ell \Delta Q^{j \to \ell}(x_i)$, for $\ell = 1, 2, \ldots, k$ and $\ell \neq j$
    if $\Delta Q^{j \to j^\star}(x_i) > 0$ **then**
      move x_i to $\mathcal{C}_{j^\star}$: $Z_{ij} \leftarrow 0$ and $Z_{ij^\star} \leftarrow 1$
      update n: $n_j \leftarrow n_j - 1$ and $n_{j^\star} \leftarrow n_{j^\star} + 1$
      update q: $q_j \leftarrow q_j - 2Q_j(x_i)$ and $q_{j^\star} \leftarrow q_{j^\star} + 2\left(Q_{j^\star}(x_i) + G_{ii}\right)$
    end if
  end for
until convergence

energy cost by $Q^{(\ell)}$. A straightforward computation gives

$$\begin{aligned} \Delta Q^{j \to \ell}(x_i) &\equiv Q^{(\ell)} - Q^{(j)} \\ &= \frac{1}{n_j - 1}\left[\frac{Q_j}{n_j} - 2Q_j(x_i)\right] - \frac{1}{n_\ell + 1}\left[\frac{Q_\ell}{n_\ell} - 2\big(Q_\ell(x_i) + K(x_i, x_i)\big)\right]. \end{aligned} \tag{51}$$

Thus, if $\Delta Q^{j \to \ell}(x_i) > 0$ we get closer to a maximum of (48) by moving x_i to $\mathcal{C}_\ell$, otherwise we keep x_i in $\mathcal{C}_j$. Based on this the algorithm goes as follows. We start with an initial configuration for the label matrix Z, then for each point x_i we compute the cost of moving it to another cluster $\mathcal{C}_\ell$, $\Delta Q^{j \to \ell}(x_i)$ for $\ell = 1, \ldots, k$ with $\ell \neq j$. Hence, let us choose $j^\star = \arg\max_\ell \Delta^{j \to \ell}(x_i)$. If $\Delta Q^{j \to j^\star}(x_i) > 0$ we move x_i to cluster $\mathcal{C}_{j^\star}$, otherwise we keep x_i in its original cluster $\mathcal{C}_j$. We update Z accordingly. The process is repeated until no points are assigned to new clusters. This whole procedure is described in Algorithm 3. Note that this ensures that the objective function is monotonically increasing at each iteration and consequently the algorithm converges in a finite number of steps.

Computing G requires $\mathcal{O}(Dn^2)$ operations, where D is the dimension of each data point and n is the data size. However, both Algorithms 2 and 3 assume that G is given. There are more efficient methods to compute G, specially if it is sparse, but we will not consider this further and just assume that G is given. The computation of each cluster cost Q_j has complexity $\mathcal{O}(n_j^2)$, and overall to compute q we have $\mathcal{O}(n_1^2 + \cdots + n_k^2) = \mathcal{O}(k \max_j n_j^2)$. These operations only need to be performed a single time. For each point x_i we need to compute $Q_j(x_i)$ once, which is $\mathcal{O}(n_j)$, and we need to compute $Q_\ell(x_i)$ for each $\ell \neq j$. The cost of computing (49) is $\mathcal{O}(n_j)$, thus the cost of step 8 in Algorithm 3 is $\mathcal{O}(k \max_j n_j)$ for $j = 1, \ldots, k$. For the entire dataset this gives a time complexity of $\mathcal{O}(nk \max_j n_j) = \mathcal{O}(kn^2)$. This is the same cost as in kernel k-means, Algorithm 2. Again, if G is sparse this can be reduced to $\mathcal{O}(kn')$ where n' is the number of nonzero entries of G.

In the following mention some important results about Hartigan's method.

Theorem 5 (Telgarsky-Vattani [17])**.** *Hartigan's method has the cost function strictly decreasing in each iteration. Moreover, if $n > k$ then*

1. *the resulting partition has no empty clusters, and*

2. *the resulting partition has distinct means.*

Neither of these two conditions are satisfied by Lloyd's method, and consequently by (kernel) k-means algorithm. The next result indicates that Hartigan's can potentially escape local optima of Lloyd's method.

Theorem 6 (Telgarsky-Vattani [17])**.** *The set of local optima of Hartigan's method is a (possibly strict) subset of local optima of Lloyd's method.*

This means that Algorithm 2 cannot improve on a local optima of Algorithm 3. On the other hand, Algorithm 3 might improve on a local optima of Algorithm 2. Lloyd's method forms Voronoi partitions, while Hartigan's method groups data in regions formed by the intersection of spheres called circlonoi cells. It can be shown that the circlonoi cells are contained within a smaller volume of a Voronoi cell, and this excess volume grows exponentially with the dimension of $\mathcal{X}$ [17, Theorems 2.4 and 3.1]. Points in this excess volume force Hartigan's method to iterate, contrary to Lloyd's method. Therefore, Hartigan's can escape local optima of Lloyd's. Moreover, this improvement should be

more prominent as dimension increases. Also, the improvement grows as k increases. The empirical results of [17] show that an implementation of Hartigan's method has comparable execution time as an implementation of Lloyd's method, but no explicit complexity was provided. In our case, we showed that both Algorithms 2 and 3 have the same time complexity.

In [18] Hartigan's method was applied to k-means problem with any Bregman divergence. They showed that the number of Hartigan's local minima is upper bounded by $\mathcal{O}(1/k)$ [18, Proposition 5.1]. In addition, they provide examples where *any* initial partition correspond to a local optima of Lloyd's method, while the number of local optima in Hartigan's method is small and correspond to true partitions of the data. Empirically, the number of Hartigan's local optima was considerably smaller than the number of Lloyd's local optima. These results indicate that Hartigan's method provides several advantages over Lloyd's method. We will see this explicitly in the following numerical results where Algorithm 3 outperforms Algorithm 2 when using the same kernel.

VII. NUMERICAL EXPERIMENTS

In the experiments below we fix the semimetric according to the traditional energy distance (1), and the point $x_0 = 0$ is chosen in the corresponding kernel (8). Therefore,

$$\rho(x, y) = \|x - y\| \qquad \text{and} \qquad K(x, y) = \tfrac{1}{2} \left(\|x\| + \|y\| - \|x - y\| \right). \tag{52}$$

We will consider other kernels as well but (52) will be the standard kernel for energy statistics and will always be present in the experiments as a reference.

Let us briefly mention that we compared Algorithm 3 and Algorithm 1 for several univariate distributions and both perform almost indistinguishable regarding the clustering quality. However, we omit these results since we will analyse more interesting scenarios in high dimensions and $k > 2$ number of clusters.

The main goal of the experiments to follow is to compare Algorithm 3 based on Hartigan's method to kernel k-means, as described in Algorithm 2, which is based on Lloyd's method. From the discussion in the end of the previous section we can anticipate that Algorithm 3 will be superior than Algorithm 2. Another goal is to illustrate the nonparametric aspect of energy statistics. To this end we also compare Algorithm 3 to standard k-means and

GMM/EM since these are reference clustering algorithms in practice. Moreover, for every algorithm, we always choose the initialization procedure from k-means++[2] [27]. Our measure of clustering quality will be the accuracy (45) based on the ground truth. Furthermore, in every experiment, we sample data many times and show the average value of the accuracy with error bars indicating the standard error. Whenever possible, we also indicate the optimal accuracy computed from Bayes classification error.

From the results of [17], we expect the improvement of Hartigan's over Lloyd's method to be more accentuated in high dimensions. Thus, we analyze how the algorithms degrade as the number of dimensions increase, while keeping the number of points in each cluster fixed. Consider two clusters with multivariate normal distributions given by

$$\begin{aligned} &x \in \mathcal{C}_i \sim \mathcal{N}(\mu_i, \Sigma_i) \qquad (i = 1, 2), \\ &\mu_1 = (\underbrace{0, \dots, 0}_{\times D})^\top, \quad \mu_2 = 0.7 \times (\underbrace{1, \dots, 1}_{\times 10}, \underbrace{0, \dots, 0}_{\times (D-10)})^\top, \quad \Sigma_1 = \Sigma_2 = I_D. \end{aligned} \tag{53}$$

Note that the Bayes error is fixed as D increases, giving an optimal accuracy of ≈ 0.86. For each D we generate 100 Monte Carlo runs, sampling 100 points for each cluster. We apply each algorithm to the resulting dataset and compute the average of the accuracy (45) (error bars are standard error). Algorithm 3 and Algorithm 2 both use the standard kernel (52). The results are shown in Fig. 2a. Note that GMM/EM is unable to estimate the covariance matrices when the number of dimensions exceeds the number of points in each cluster, i.e. when $D \gtrsim 100$. We see that Algorithm 3 performs better than all the other ones, and in particular it outperforms kernel k-means, Algorithm 2, as the number of dimensions increase.

Note that in this case GMM and k-means are consistent estimators.

Consider now the following setting:

$$x \in \mathcal{C}_i \sim \mathcal{N}(\mu_i, \Sigma_i) \quad (i = 1, 2), \quad \mu_1 = (\underbrace{0, \dots, 0}_{\times D})^\top, \quad \mu_2 = (\underbrace{1, \dots, 1}_{\times 10}, \underbrace{0, \dots, 0}_{\times (D-10)})^\top, \tag{54}$$

where

$$(\Sigma_1)_{ij} = \begin{cases} i^{-q}\delta_{ij} & \text{if } i \leq 10 \\ \delta_{ij} & \text{if } 10 < i \leq D \end{cases} \qquad (\Sigma_2)_{ij} = \begin{cases} i^{\,q}\delta_{ij} & \text{if } i \leq 10 \\ \delta_{ij} & \text{if } 10 < i \leq D \end{cases} \tag{55}$$

[2] Notice that we just use the initialization procedure and not the full k-means++ algorithm.

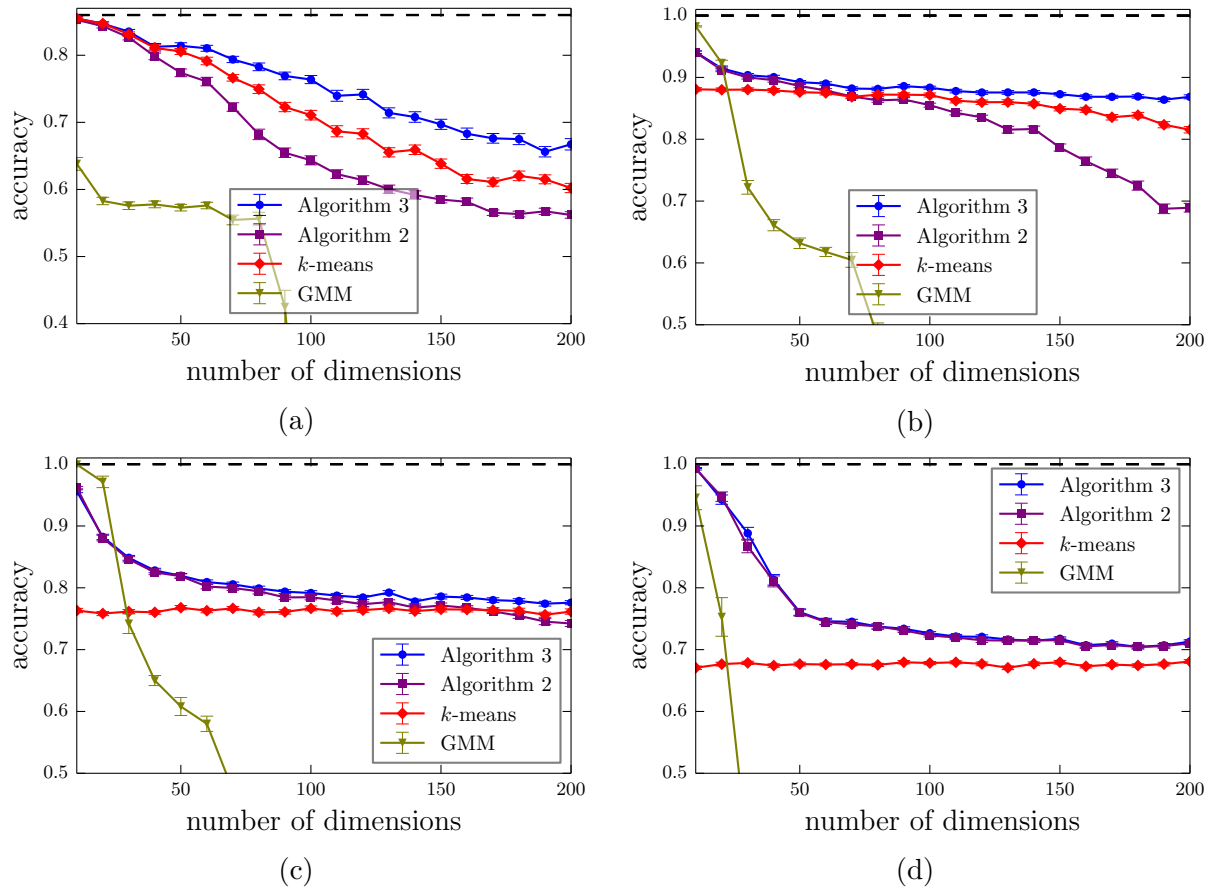


FIG. 2. Comparison of Algorithm 3, Algorithm 2, standard k-means and GMM/EM as the number of dimensions increase in Gaussian settings. We compute the average of (45) over 100 samples with error bars being standard error. We have two clusters with 100 points each. (a) Data as in (53), where the optimal accuracy from Bayes error is the dashed line equal to ≈ 0.86. (b) Data from (54) with $q = 1/2$ in (55). (c) Data from (54) with $q = 1$ in (55). (d) Data from (54) with $q = 2$ in (55). The optimal accuracy from Bayes error in (b–d) is ≈ 1.

and we choose $q \in \{1/2, 1, 2\}$. Above $\delta_{ij} = 1$ if $i = j$ and $\delta_{ij} = 0$ if $i \neq j$ is the Kronecker delta. In these cases, the best possible accuracy from Bayes classification error is ≈ 1. In Fig. 2b–d we have $q = 1/2$, $q = 1$, and $q = 2$, respectivelly. Again, GMM/EM is unable to estimate the covariance matrices as dimensions get larger than $\gtrsim 100$, and it gives poor results even for number of dimensions much lower than this. Note that GMM requires a larger number of points to estimate the parameters accurately. Algorithm 3 outperforms Algorithm 2, and k-means degrades faster as q increases.

To summarize, in the experiments of Fig. 2 we see a better performance of Algorithm 3

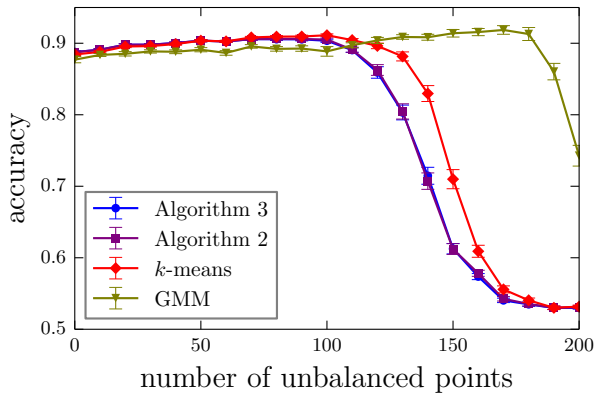


FIG. 3. Comparison of Algorithm 3, Algorithm 2, k-means, and GMM/EM. The data is distributed as (56) where we make the clusters progressively more unbalanced.

compared to the other ones, and in particular to kernel k-means, where we recall that both find local solutions to the same optimization problem (18). Algorithm 3 is more robust as the number of dimensions increase.

Consider the effect of having unbalanced clusters accordint to

$$\begin{aligned} &x \in \mathcal{C}_i \sim \frac{n_i}{N}\mathcal{N}(\mu_i, \Sigma_i) \quad (i=1,2), \quad \mu_1 = (0,0,0,0)^\top, \quad \mu_2 = 1.5 \times (1,1,0,0)^\top, \\ &\Sigma_1 = I_4, \quad \Sigma_2 = \begin{pmatrix} 1/2 & 0 & 0 & 0 \\ 0 & 1/2 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}, \quad n_1 = N - m, \quad n_2 = N + m, \quad N = 200. \end{aligned} \tag{56}$$

We then increase m, that is we make the clusters progressively more unbalanced, and plot the average of (45) over 100 samples for each m (error bars are standard error). The results are in Fig. 3. For highly unbalanced clusters we see that GMM performs better than the other methods which have similar performance.

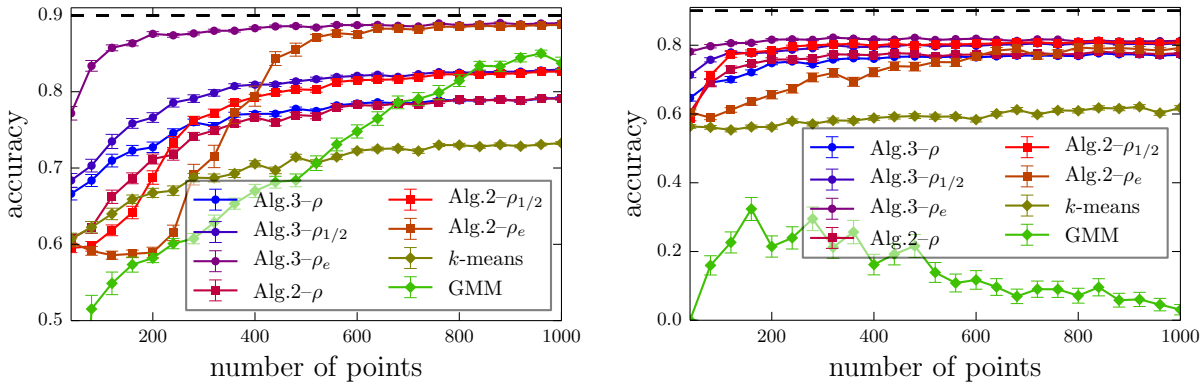
Besides the standard kernel from energy statistics (52) consider the following two other semimetrics with their respective generating kernels:

$$\rho_{1/2}(x,y) = \|x-y\|^{1/2} \qquad K_{1/2}(x,y) = \tfrac{1}{2}\left(\|x\|^{1/2} + \|y\|^{1/2} - \|x-y\|^{1/2}\right), \tag{57}$$

$$\rho_e(x,y) = 2 - 2e^{-\frac{1}{2}\|x-y\|} \qquad K_e(x,y) = e^{-\frac{1}{2}\|x-y\|}. \tag{58}$$

The kernel $K_{1/2}(x,y)$ corresponds to the energy distance (2) with $\alpha = 1/2$. We sample data from the following normal distribution in $D = 20$:

$$\begin{aligned} &x \in \mathcal{C}_i \sim \mathcal{N}(\mu_i, \Sigma_i) \qquad (i=1,2), \\ &\mu_1 = (\underbrace{0,\ldots,0}_{\times 20})^\top, \quad \mu_2 = \tfrac{1}{2}(\underbrace{1,\ldots,1}_{5},\underbrace{0,\ldots,0}_{15})^\top, \quad \Sigma_1 = \tfrac{1}{2}I_{20}, \quad \Sigma_2 = I_{20}. \end{aligned} \tag{59}$$



(a) (b)

FIG. 4. Algorithm 3 and Algorithm 2 with kernels (52), (57) and (58), k-means, and GMM. The optimal accuracy in both cases is ≈ 0.9. We show the average of (45) over 100 samples with standard error. (a) Data distributed as in (59). (b) Data distributed as in (60).

We sample an equal number of points for each cluster, which is progressively increased. The optimal accuracy based on Bayes classification error is ≈ 0.90. Clustering results are shown in Fig. 4a. Algorithm 3 outperforms the other ones, and in particular the kernel (58) provides better results. As the number of points get large enough GMM starts to approach optimal Bayes, as it should since it is a consistent model to the data. However, Algorithm 3 with kernel (58) approach optimal Bayes with a much smaller number of points. Moreover, Algorithm 3 outperforms Algorithm 2 for any of the kernel choices.

Now consider the same parameters as in (59) but with lognormal distributions,

$$\log x \in \mathcal{C}_i \sim \mathcal{N}(\mu_i, \Sigma_i) \qquad (i = 1, 2). \tag{60}$$

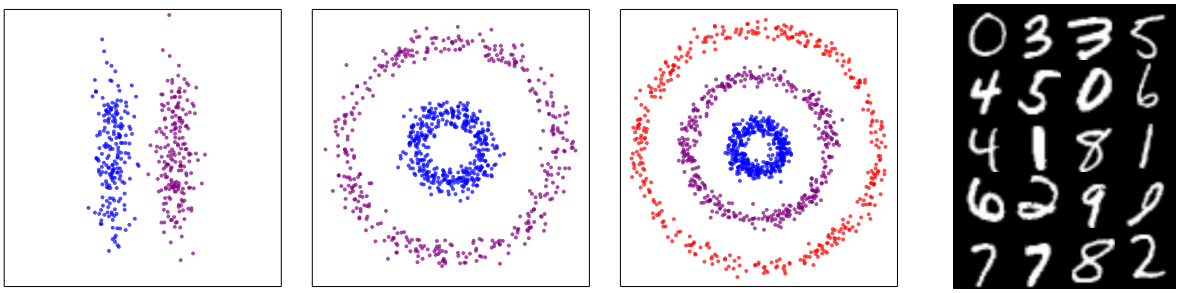
The same previous experiment is shown in Fig. 4b. Note that Algorithm 3 still performs accurately, while k-means works almost at chance, and GMM is not even able to estimate the parameters. Again, the kernel (58) provides better results than (52) or (57). The experiments in Fig. 4 illustrate how energy statistics clustering is nonparametric.

Consider the following choices of semimetric and corresponding generating kernel:

$$\rho_\alpha(x, y) = \|x - y\|^\alpha \qquad K_\alpha(x, y) = \tfrac{1}{2}\left(\|x\|^\alpha + \|y\|^\alpha - \|x - y\|^\alpha\right), \tag{61}$$

$$\widetilde{\rho}_\sigma(x, y) = 2 - 2e^{-\frac{\|x-y\|}{2\sigma}} \qquad \widetilde{K}_\sigma(x, y) = e^{-\frac{\|x-y\|}{2\sigma}}, \tag{62}$$

$$\widehat{\rho}_\sigma(x, y) = 2 - 2e^{-\frac{\|x-y\|^2}{2\sigma^2}} \qquad \widehat{K}_\sigma(x, y) = e^{-\frac{\|x-y\|^2}{2\sigma^2}}. \tag{63}$$



(a) (b) (c) (d)

FIG. 5. (a) Parallel cigars. (b) Two concentric circles with noise. (c) Three concentric circles with noise. (d) MNIST handwritten digits. Clustering results are in Table I and Table II.

In Fig. 5 we have examples of complex two dimensional datasets. The two parallel cigars of Fig. 5a have 200 points each. For the concentric circles of Fig. 5b and Fig. 5c we sample 400 points for each class. We apply Algorithm 3 and Algorithm 2, using the above kernels (61)–(63), as well as k-means and GMM. The results are shown in Table I where the respective choice of parameters for the kernels are indicated. For the data in Fig. 5a the semimetrics ρ_1 and $\rho_{1/2}$ are able to provide more accurate results compared to k-means. However, the gaussian kernel $\widetilde{\rho}_2$ gives very accurate results, similar to GMM, which is a consistent estimator for this data. For the data shown in Fig. 5b we see that the clustering quality is highly sensitive to the choice of kernel, and only (62) was able to cluster accurately. The same kernel choice to the case of Fig. 5c still provides better results than the other kernels, but the results are less accurate compared to the data in Fig. 5b.

Next we consider the well-known MNIST handwritten digit dataset as illustrated in Fig. 5d. Each data point is an 8-bit gray scale image forming a 784-dimensional vector corresponding to the digits $\{0, 1, \ldots, 9\}$. Besides the kernel (61), we consider the gaussian kernel (63) with

$$\sigma^2 = \frac{1}{n^2} \sum_{i,j=1}^{n} \|x_i - x_j\|^2, \tag{64}$$

which is computed from a sample $\{x_i\}_{i=1}^n$. We consider subsets of the classes $\{0, 1, \ldots, 9\}$, where we sample 100 points for each class. We perform clustering through Algorithm 3, Algorithm 2, and k-means. The results are shown in Table II where the kernel and its parameter for each case is indicated. Algorithm 3 performed slightly better than k-means, except for the last column where all the methods are comparable. Unsupervised clustering on MNIST dataset without any feature extraction or dimensionality reduction is not an

		Fig. 5a		*Fig. 5b*		*Fig. 5c*
Algorithm 3	ρ_1	0.766 ± 0.066	ρ_1	0.522 ± 0.006	ρ_1	0.437 ± 0.030
	$\rho_{1/2}$	0.859 ± 0.062	$\rho_{1/2}$	0.524 ± 0.007	$\rho_{1/2}$	0.547 ± 0.026
	$\widetilde{\rho}_2$	0.971 ± 0.015	$\widetilde{\rho}_1$	$\mathbf{0.9999 \pm 0.0001}$	$\widetilde{\rho}_2$	$\mathbf{0.677 \pm 0.003}$
	$\widehat{\rho}_2$	$\mathbf{0.998 \pm 0.001}$	$\widehat{\rho}_1$	0.597 ± 0.052	$\widehat{\rho}_2$	0.645 ± 0.012
Algorithm 2	ρ_1	0.758 ± 0.069	ρ_1	0.516 ± 0.002	ρ_1	0.452 ± 0.030
	$\rho_{1/2}$	0.901 ± 0.060	$\rho_{1/2}$	0.524 ± 0.007	$\rho_{1/2}$	0.570 ± 0.016
	$\widetilde{\rho}_2$	0.971 ± 0.015	$\widetilde{\rho}_1$	$\mathbf{0.9999 \pm 0.0001}$	$\widetilde{\rho}_2$	$\mathbf{0.673 \pm 0.002}$
	$\widehat{\rho}_2$	$\mathbf{0.998 \pm 0.001}$	$\widehat{\rho}_1$	0.528 ± 0.008	$\widehat{\rho}_2$	0.640 ± 0.013
k-means		0.599 ± 0.046		0.521 ± 0.005		0.360 ± 0.004
GMM		0.9995 ± 0.0003		0.598 ± 0.018		0.479 ± 0.021

TABLE I. Clustering the data shown in Fig. 5 with Algorithm 3 and Algorithm 2, with kernels (61)–(63), as well as k-means and GMM. We sample 10 times and show the average accuracy (45) with standard error.

easy task. For instance, the same experiment was performed in [28] where a low-rank transformation is learned then subsequently used in subspace clustering, providing very accurate results. One could explore analogous methods for learning a better representation of the data and subsequently apply Algorithm 3 for clustering.

Class Subset		$\{0,1,2\}$	$\{0,1,\ldots,4\}$	$\{0,1,\ldots,6\}$	$\{0,1,\ldots,8\}$
Algorithm 3	ρ_1	0.907 ± 0.007	0.866 ± 0.006	0.715 ± 0.013	0.616 ± 0.019
	$\rho_{1/2}$	$\mathbf{0.918 \pm 0.006}$	0.849 ± 0.025	0.711 ± 0.010	0.642 ± 0.009
	$\widetilde{\rho}_\sigma$	0.900 ± 0.007	$\mathbf{0.871 \pm 0.005}$	$\mathbf{0.719 \pm 0.010}$	0.630 ± 0.016
Algorithm 2	ρ_1	0.914 ± 0.006	0.845 ± 0.023	0.664 ± 0.022	0.614 ± 0.014
	$\rho_{1/2}$	0.895 ± 0.011	0.822 ± 0.026	0.669 ± 0.021	0.591 ± 0.019
	$\widetilde{\rho}_\sigma$	0.896 ± 0.007	0.869 ± 0.006	0.705 ± 0.016	$\mathbf{0.646 \pm 0.020}$
k-means		0.871 ± 0.015	0.840 ± 0.022	0.707 ± 0.012	0.634 ± 0.011

TABLE II. Clustering the data shown in Fig. 5d with Algorithm 3, Algorithm 2, and k-means. We use the kernel (61) with $\alpha \in \{1,2\}$ and the gaussian kernel (63) with σ given by (64). For each subset of digits we sample 10 times and show the average accuracy (45) with standard error. We sample 100 points for each class.

VIII. DISCUSSION

We considered clustering from the perspective of energy statistics theory which provides a nonparametric test for equality of distributions. We showed that the clustering problem reduces to a quadratically constrained quadratic program, as described in Proposition 2. Moreover, this problem is equivalent to kernel k-means optimization problem once the kernel is fixed; see Proposition 3. Energy statistics, however, fixes a family of standard kernels consistent with (2), and more general kernels related to (4) can be obtained. Our results imply that kernel k-means optimization problem can be interpreted as a consequence of energy statistics based clustering. We also considered a weighted version of energy statistics whose clustering formulation establishes connections with spectral clustering and graph partitioning problems.

We considered Algorithm 3 based on Hartigan's method and compared with kernel k-means, described in Algorithm 2, which is based on Lloyd's heuristic. Both have the same time complexity, however, the numerical results provide compelling evidence that Algorithm 3 is more robust with a superior clustering performance. This is also theoretically supported as described in the end of Section VI.

Finally, kernel methods can benefit from sparsity and fixed-rank approximations of the Gram matrix. For instance, an approach to make kernel k-means scalable was recently proposed [29], and a modified technique for rank reduction in Nyström method was also recently considered [30]. It would be interesting to apply these and related techniques to the case of Algorithm 3. Another fruitful avenue of exploration would be to find better methods to tackle the optimization problem (18).

Acknowledgements

We would like to thank Carey Priebe for discussions. This work was supported by NIH TRA grant.

---

[1] G. J. Székely and M. L. Rizzo. Energy Statistics: A Class of Statistics Based on Distances. *Journal of Statistical Planning and Inference*, 143:1249–1272, 2013.

[2] M. L. Rizzo and G. J. Székely. DISCO Analysis: A Nonparametric Extension of Analysis of Variance. *The Annals of Applied Statistics*, 4(2):1034–1055, 2010.

[3] G. J. Székely and M. L. Rizzo. Hierarchical Clustering via Joint Between-Within Distances: Extending Ward's Minimum Variance Method. *Journal of Classification*, 22(2):151–183, 2005.

[4] S. Li. k-Groups: A Generalization of k-Means by Energy Distance. PhD Thesis, Bowling Green State University, 2015.

[5] R. Lyons. Distance Covariance in Metric Spaces. *The Annals of Probability*, 41(5):3284–3305, 2013.

[6] D. Sejdinovic, B. Sriperumbudur, A. Gretton, and K. Fukumizu. Equivalence of Distance-Based and RKHS-Based Statistic in Hypothesis Testing. *The Annals of Statistics*, 41(5):2263–2291, 2013.

[7] S. P. Lloyd. Least Squares Quantization in PCM. *IEEE Transactions on Information Theory*, 28(2):129–137, 1982.

[8] J. B. MacQueen. Some Methods for Classification and Analysis of Multivariate Observations. In *Proceedings of the 5th Berkeley Symposium on Mathematical Statistics and Probability*, volume 1, pages 281–297. University of California Press, 1967.

[9] E. Forgy. Cluster Analysis of Multivariate Data: Efficiency versus Interpretabiliby of Classification. *Biometrics*, 21(3):768–769, 1965.

[10] B. Schölkopf, A. J. Smola, and K. R. Müller. Nonlinear Component Analysis as a Kernel Eigenvalue Problem. *Neural Computation*, 10:1299–1319, 1998.

[11] M. Girolami. Kernel Based Clustering in Feature Space. *Neural Networks*, 13(3):780–784, 2002.

[12] J. Mercer. Functions of Positive and Negative Type and their Connection with the Theory of Integral Equations. *Proceedings of the Royal Society of London*, 209:415–446, 1909.

[13] I. S. Dhillon, Y. Guan, and B. Kulis. Kernel K-means: Spectral Clustering and Normalized Cuts. In *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '04, pages 551–556, New York, NY, USA, 2004. ACM.

[14] I. S. Dhillon, Y. Guan, and B. Kulis. Weighted Graph Cuts without Eigenvectors: A Multilevel Approach. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(11):1944–1957, 2007.

[15] M. Filippone, F. Camastra, F. Masulli, and S. Rovetta. A Survey of Kernel and Spectral Methods for Clustering. *Pattern Recognition*, 41:176–190, 2008.

[16] J. A. Hartigan and M. A. Wong. Algorithm AS 136: A k-Means Clustering Algorithm. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, 28(1):100–108, 1979.

[17] M. Telgarsky and A. Vattani. Hartigan's Method: k-Means Clustering without Voronoi. In *Proceedings of the 13th International Conference on Artificial Intelligence and Statistics (AISTATS)*, volume 9, pages 313–319. JMLR, 2010.

[18] N. Slonim, E. Aharoni, and K. Crammer. Hartigan's k-Means versus Lloyd's k-Means — Is it Time for a Change? In *Proceedings of the 20th International Conference on Artificial Intelligence*, pages 1677–1684. AAI Press, 2013.

[19] N. Aronszajn. Theory of Reproducing Kernels. *Transactions of the American Mathematical Society*, 68(3):337–404, 1950.

[20] A. Gretton, K. M. Borgwardt, M. J. Rasch, B. Schölkopf, and A. Smola. A Kernel Two-Sample Test. *Journal of Machine Learning Research*, 13:723–773, 2012.

[21] C. Berg, J. P. R. Christensen, and P. Ressel. *Harmonic Analysis on Semigroups: Theory of Positive Definite and Related Functions*. Graduate Text in Mathematics 100. Springer, New York, 1984.

[22] A. Y. Ng, M. I. Jordan, and Y. Weiss. On Spectral Clustering: Analysis and an Algorithm. In *Advances in Neural Information Processing Systems*, volume 14, pages 849–856, Cambridge, MA, 2001. MIT Press.

[23] B. Kernighan and S. Lin. An Efficient Heuristic Procedure for Partitioning Graphs. *The Bell System Technical Journal*, 49(2):291–307, 1970.

[24] J. Shi and J. Malik. Normalized Cust and Image Segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):888–905, 2000.

[25] P. Chan, M. Schlag, and J. Zien. Spectral k-Way Ratio Cut Partitioning. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 13:1088–1096, 1994.

[26] S. X. Yu and J. Shi. Multiclass Spectral Clustering. In *Proceedings Ninth IEEE International Conference on Computer Vision*, volume 1, pages 313–319, 2003.

[27] D. Arthur and S. Vassilvitskii. k-means++: The Advantage of Careful Seeding. In *Proceedings of the Eighteenth annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1027–1035, Philadelphia, PA, USA, 2007. Society for Industrial and Applied Mathematics.

[28] Q. Qui and G. Sapiro. Learning Transformations for Clustering and Classification. *Journal of Machine Learning Research*, 16:187–225, 2015.

[29] S. Wang, A. Gittens, and M. W. Mahoney. Scalable Kernel k-Means Clustering with Nyström Approximation: Relative-Error Bounds. arXiv:1706.02803v2 [cs.LG], 2017.

[30] F. Pourkamali-Anaraki and S. Becker. Improved Fixed-Rank Nyström Approximation via QR Decomposition: Practical and Theoretical Aspects. arXiv:1708.03218v2 [stat.ML], 2017.