Nonparametric Clustering from Energy Statistics

Guilherme França[*] and Joshua T. Vogelstein[†]

Johns Hopkins University

Abstract

Energy statistics provides a nonparametric test for equality of distributions. It was proposed by Székely in the 80's inspired by the Newtonian gravitational potential from classical mechanics. Energy statistics was further generalized to probability distributions on arbitrary metric spaces, and more recently a connection with reproducing kernel Hilbert spaces was established. Nevertheless, although extensively used by the statistics community, it has not been previously incorporated in machine learning problems. In this paper, 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). We show the equivalence between the energy statistics clustering formulation and the kernel k-means optimization problem (not to be confused with kernel k-means algorithm). Therefore, our results imply a first principles derivation of kernel k-means problem from energy statistics theory, thus bringing this important clustering method into a formal statistical framework. Moreover, energy statistics is nonparametric, it usually fixes the kernel choice, and if prior information is available it can be easily incorporated in the kernel construction. We propose an iterative algorithm to find local optimizers of the aforementioned QCQP based on the energy change of moving points to different clusters. This algorithm is different but has the same computational cost as kernel k-means algorithm. We compare it to kernel k-means, standard k-means, and GMM/EM algorithms by providing carefully designed numerical experiments. The results show that, in general, this algorithm outperforms these most used clustering algorithms.

[*] guifranca@gmail.com

[†] jovo@jhu.edu

I. INTRODUCTION

Energy statistics is based on the energy distance between probability distributions, which provides a notion of statistical potential energy to statistical observations, in close analogy to Newton's gravitational potential in classical mechanics. It provides a nonparametric test for equality of distribution, such that minimum energy is achieved under equality of distributions. We refer the reader to [1], and references therein, for an overview. Energy statistics has been applied to several goodness-of-fit hypothesis tests, multi-sample tests of equality of distributions, analysis of variance [2], and (nonlinear) dependence tests through distance covariance and distance correlation, which generalizes the Pearson correlation coefficient. Moreover, energy statistics was applied to hierarchical clustering [3] by extending Ward's method of minimum variance.

More recently, distance covariance was generalized from Euclidean spaces to metric spaces of strong negative type [4]. 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 [5]. This important work provides the missing link between techniques commonly used in the statistics literature, regarding energy statistics, and techniques commonly used in machine learning.

Given a dataset $\mathbb{X} = \{x_1, \dots, x_n\}$, where each data point x_i lives in a space $\mathcal{X}$, the clustering problem consists in grouping these points into k groups $\{\mathcal{C}_1, \mathcal{C}_2, \dots, \mathcal{C}_k\}$, such that points belonging to the same group are more "similar" to each other than to points in other groups. This intuitive notion already assumes a "metric" able to measure the similarity between points. Clustering is an unsupervised method, and one of the most important problems in machine learning since it provides the first step towards automatically constructing labels for previously unseen data. In practice k-means is arguably the most used algorithm, and gaussian mixture models (GMM) through the expectation maximization (EM) algorithm is also often used. Both are parametric, making strong assumptions about the distribution of the data (isotropy and normality), and it assumes that data points lie in Euclidean space, $\mathcal{X} = \mathbb{R}^D$, hence the similarity is based on Euclidean metric, $\|x_i - x_j\|^2 = (x_i - x_j)^\top (x_i - x_j) = \sum_{\ell=1}^{D} (x_{i,\ell} - x_{j,\ell})^2$. These algorithms provide a good quality clustering when data is linearly separable in Euclidean space, i.e. there is a clear resolution in the cluster centers and points from different clusters do not overlap considerably.

To account for nonlinearly separable data, which may live in an arbitrary non-Euclidean space, kernel methods are usually employed. If a so-called Mercer kernel $K : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is available, which is a symmetric and positive definite function, it guarantees that there exists a map $\varphi : \mathcal{X} \to \mathcal{H}$ where $\mathcal{H}$ is a Hilbert space [6]. One can thus compute similarities between data point through the inner product of the feature space $\mathcal{H}$ and perform clustering in $\mathcal{H}$. The convenience of a Mercer kernel is that the map φ is only used implicitly, i.e. one does not need to know φ, since the inner of product of $\mathcal{H}$ can be computed only based on the kernel function $K(\cdot, \cdot)$ which acts on the original space $\mathcal{X}$. This technique is known as the kernel trick. Kernel k-means problem is exactly k-means in the feature space [7, 8] (see also [9] for a survey of clustering methods). To cluster data that is not linearly separable in $\mathcal{X}$, kernel based methods exploit the nonlinear structure provided by the kernel function, which implicitly maps data points to a usually higher dimensional feature space, with the hope that they becomes separable by forming well-defined groups. The choice of kernel is crucial, and yet there is no principled method for this. One must rely on ad hoc methods to find an appropriate kernel which is data dependent. As it stands, kernel k-means problem is an heuristic approach to clustering in the sense that it is not derived from a statistical theory.

The main contribution of this paper is to provide a consistent formulation to clustering based on energy statistics theory. The original goal is to provide a nonparametric clustering method, since energy distance is nonparametric. The basis of our work is the theory developed in [5] which establishes an equivalence between energy distance and RKHS. Based on this and on the multi-sample test for equality of distributions, we show that there is only one possible clustering optimization problem consistent with the energy statistics formulation. We then show that this optimization problem is equivalent to a quadratically constrained quadratic program (QCQP), which has the same trace maximization form as kernel k-means, spectral clustering, and some well-known graph partitioning problems [10, 11]. We show the equivalence between this approach and kernel k-means optimization problem, implying that the later can actually be derived from energy statistics, thus bringing this important clustering method into a unified statistical theory. However, it is important to note that energy statistics fixes a kernel to begin with. From an algorithmic perspective, our contribution is to provide an iterative algorithm, which is different but has the same complexity as kernel k-means algorithm. Our numerical results provide compelling evidence that this algorithm is more accurate and robust than kernel k-means algorithm. Moreover, these experiments puts in evidence the nonparametric aspect of energy statistics based clustering, since it is able to perform accurately on data having very different distributions, contrary to k-means and GMM for instance.

Our work is organized as follows. In section II we review 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. In section IV we consider a simple example in one dimension, where we propose an algorithm which requires no initialization. In section V we briefly review kernel k-means algorithm, and propose a new iterative algorithm to solve this QCQP. Section VI 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 VII.

II. BACKGROUND ON ENERGY STATISTICS AND RKHS

In this section we briefly review the main concepts from energy statistics and its relation to reproducing kernel Hilbert spaces (RKHS) which form the basis of our work. For more details we refer the reader to [1] and [5].

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 [1]

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

called *energy distance*, is rotational 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

$\mathcal{E}_2(P,Q) = 2\|\mathbb{E}(X) - \mathbb{E}(Y)\|^2$, showing that equality to zero only requires equality of the means, and thus $\mathcal{E}_2(P,Q) = 0$ does not imply equality of distributions.

It is important to mention that (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 $\rho(X,Y) = \|\varphi(X) - \varphi(Y)\|^2_{\mathcal{H}}$, which 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 legit 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 [5]. 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 since this inner product is real, $\langle h_x, h_y \rangle^* = \langle h_y, h_x \rangle = \langle h_x, h_y \rangle$, we immediately have that the kernel is symmetric, $K(y,x) = K(y,x)$. 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; $v^\top G\, v \geq 0$ for any vector $v \in \mathbb{R}^n$.

The Moore-Aronszajn theorem [12] 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 $\mathcal{H}_K$. This is called the maximum mean discrepancy (MMD) and is given by

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

which can also be written as [13]

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

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 semidefinite kernels are closely related [14]. 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). Here 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 [5]. 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 [5]. If ρ is a semimetric of negative type and K a kernel that generates ρ, then replacing (9) into (4), and using (6), yields

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

Since $\gamma_k^2(P, Q) = \|h_P - h_Q\|_{\mathcal{H}_K}^2$ we can compute the energy distance using the inner product of $\mathcal{H}_K$. Moreover, this can be computed from the data according to (7).

Finally, let us recall the main formulas for test statistic of equality of distributions [1]. Assume we have data $\mathbb{X} = \{x_1, \dots, x_n\}$, where $x_i \in \mathcal{X}$, and $\mathcal{X}$ is a space of negative type. Consider a 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 every data point $x \in \mathcal{C}_j$ 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 $S(P_1, \dots, P_k)$ which measures how different these mass distributions 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. □

Therefore, for a given k, the clustering problem amounts to finding the best partition of the data by solving (14). Notice that this is a hard assignment clustering problem as partitions are disjoint.

Now we show how to formulate problem (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 satisfy $Z^\top Z = D$ where the diagonal matrix $D = \operatorname{diag}(n_1, \dots, n_k)$ contains the number of points in each cluster. Let us also introduce the rescaled matrix $Y \equiv Z D^{-1/2}$. 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 reveals the optimization problem behind (14), which is NP-hard since it is a quadratically constrained quadratic program (QCQP).

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

$$\max_{Y} \operatorname{Tr}\left(Y^\top G Y\right) \qquad \text{s.t. } Y \ge 0,\ Y^\top Y = I,\ YY^\top \boldsymbol{e} = \boldsymbol{e}, \tag{18}$$

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

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

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

Note that the first term does not contribute to the optimization problem, since it is a global term that does not depend which partition is chosen. Therefore, minimizing (19) is equivalent to

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

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 $\mathrm{Tr}\left(D^{-1}Z^{\top}GZ\right)$. Now we can use the cyclic property of the trace, and by the own 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 $\mathrm{Tr}\left(Y^{\top}GY\right)$. Notice that the above constraints imply that $Z^T Z = D$, where $D = \mathrm{diag}(n_1, \dots, n_k)$, which in turn gives $D^{-1/2}Y^T Y D^{-1/2} = D$, 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 $Z^T\boldsymbol{e} = D\boldsymbol{e}$. Replacing $Z = YD^{1/2}$ we have $Y^{\top}\boldsymbol{e} = D^{1/2}\boldsymbol{e}$. Multiplying this last equation on the left by Y, and noticing that $YD^{1/2}\boldsymbol{e} = Z\boldsymbol{e} = \boldsymbol{e}$, we finally obtain $YY^{\top}\boldsymbol{e} = \boldsymbol{e}$. Therefore, the optimization problem (22) is equivalent to (18) . □

Based on Proposition 2, to group data $\{x_1, \dots, x_n\}$ 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

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

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 $\max \mathrm{Tr}\left(Y^{\star\top}GY^{\star}\right) = \sum_{i=1}^{k}\lambda_i$. One might then normalize and threshold the rows of $Y^{\star}$, or even better, following [15] we 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 same procedure done in spectral clustering on the (normalized) Laplacian of the graph defined by a similarity matrix. However, computing eigenvectors of a very large matrix can be problematic, and usually iterative methods are preferred. We will propose an iterative method to find approximate solutions to (18) in section V.

It is important to note that the previous theory for clustering based on energy statistics is valid for data living in an *arbitrary* space of negative type. This clustering method is *nonparametric* since it does not make any assumptions about the distribution of the data, nor the metric space where data belongs to, contrary to k-means and gaussian mixture models (GMM), for example. Moreover, this approach *does not* require the concept of the *cluster mean*, which can be ill-defined for some types of data, such as images for instance. Since no mean is involved, the method should also be robust to outliers. If one uses the traditional energy distance defined in (1), then $\rho(x, y) = \|x - y\|$, and this fixes the kernel through (8). In the same way, for data living in more general metric spaces $(\mathcal{X}, \rho)$, the corresponding semimetric ρ fixes the kernel. In practice, however, the clustering quality strongly depend on the choice of a suitable ρ which is what measures the similarity between different data points, and is equivalent to choosing an appropriate kernel. Nevertheless, if prior knowledge is available for choosing ρ, this can easily be taken into account.

One may wonder how energy statistics clustering relates to the well-known kernel k-means problem[1]. We now address this question. For a positive semidefinite G, 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

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

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. It is known [10, 11] 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. *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

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

[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, which is just one possible recipe to solve (24).

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 second term. Thus, the kernel k-means optimization problem (24) is equivalent to

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

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 the optimization problem in the form (18). □

The above result shows that kernel k-means problem is actually a consequence of energy statistics. In this vein, kernel k-means is part of a statistical framework where we are maximizing distances between probability distributions.

As shown in [11], 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 for the sake of completeness demonstrate next.

A. Clustering Based on Weighted Energy Statistics

Now we generalize the formulas from energy statistics to incorporate weights associated to each data point. Let $w(x)$ be a weight function for the point $x \in \mathbb{X}$. 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 unnaltered, so the clustering problem becomes

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

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

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

where $w_i = w(x_i)$ and $\boldsymbol{e} \in \mathbb{R}^n$ 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*

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

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

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

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

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. Now we obtain a condition on the columns of H. 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 $i = 1, \dots, k$, 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 $H H^\top \mathcal{W}^{1/2} \boldsymbol{e} = \mathcal{W}^{1/2} \boldsymbol{e}$, which gives the constraint $H H^\top \boldsymbol{\omega} = \boldsymbol{\omega}$. □

B. Connection with Graph Partioning

The relation between kernel k-means and graph partitioning problems was already established [11]. We repeat a similar analysis due to the relation of these problems to energy statistics and RKHS, which may provide 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 measure 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

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

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

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

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

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

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 $\operatorname{links}(\mathcal{C}_j, \mathcal{V} \backslash \mathcal{C}_j) = \operatorname{links}(\mathcal{C}_j, \mathcal{V}) - \operatorname{links}(\mathcal{C}_j, \mathcal{C}_j)$. Several graph partitioning methods [16–19] can be seen as a particular case of (36) or (37).

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

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

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

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

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, $\mathcal{A}_{ii} = 0$. Using (40) in (11)–(13) allows one to use energy statistics theory for inference over populations of graphs.

IV. 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 usefull 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

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

where we have the indicator function defined as $\mathbb{1}_A = 1$ 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

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

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 inside the partition, obtaining $\bar{\mathcal{C}} = [x_j \in \mathcal{C} : x_1 \le x_2 \le \cdots \le x_n]$, we can write (42) in the following simple form:

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

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

```
input  data $\mathbb{X}$
output  label matrix $Z$
sort $\mathbb{X}$ obtaining $\bar{\mathbb{X}} = [x_1, \dots, x_n]$
for $j \in [1, \dots, n]$ do
    Let $\bar{\mathcal{C}}_1^{(j)} = [x_i : i = 1, \dots, j]$ and $\bar{\mathcal{C}}_2^{(j)} = [x_i : i = j+1, \dots, n]$
    $W^{(j)} \leftarrow W\big(\bar{\mathcal{C}}_1^{(j)}, \bar{\mathcal{C}}_2^{(j)}\big)$ from (44)
end for
$j^\star \leftarrow \arg\min_j W^{(j)}$
$Z_{j\bullet} \leftarrow (1,0)$ if $j \le j^\star$, and $Z_{j\bullet} \leftarrow (0,1)$ otherwise, for $j = 1, \dots, n$
```

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 $\mathbb{X} = \bigcup_{j=1}^{k} \bar{\mathcal{C}}_j$, but notice that the entire data set $\mathbb{X}$ does not need to be necessarily ordered, the within energy dispersion (12) can be written as

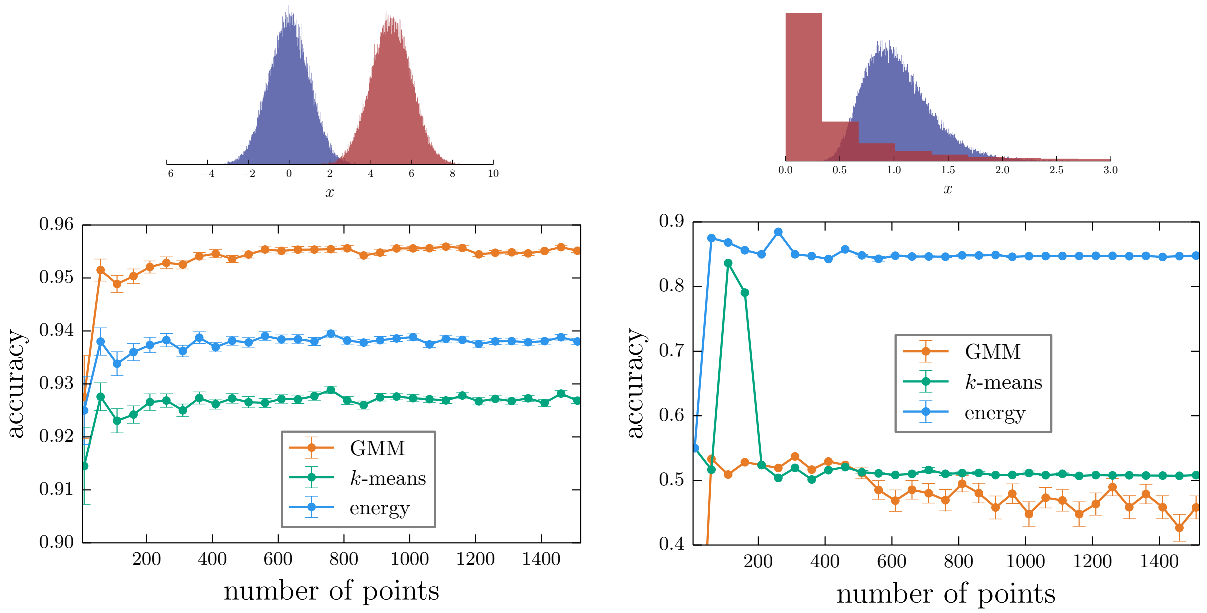
$$W\left(\bar{\mathcal{C}}_1, \dots, \bar{\mathcal{C}}_k\right) = \sum_{j=1}^{k} \sum_{\ell=1}^{n_j} \frac{2\ell - 1 - n_j}{n_j} x_\ell. \tag{44}$$

For a two-class problem, we can use (44) to cluster 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. Notice 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

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

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 equal number of points in each cluster, the value $1/2$ correspond to chance.



(a)

(b)

FIG. 1. Energy statistics clustering by Algorithm 1, compared to k-means and GMM. Both clusters have the same number of points (x-axis), which are increased in each experiment. We sample 100 times from the distributions in the histograms and the y-axis is the average value of the clustering accuracy (45) (errorbars are standard error). (a) $x \sim \frac{1}{2}\left[\mathcal{N}(\mu_1, \sigma_1) + \mathcal{N}(\mu_2, \sigma_2)\right]$, $\mu_1 = 0$, $\mu_2 = 5$, $\sigma_1 = 1$, and $\sigma_2 = 2$. (b) $x \sim \frac{1}{2}\left[e^{\mathcal{N}(\mu_1,\sigma_1)} + e^{\mathcal{N}(\mu_2,\sigma_2)}\right]$, $\mu_1 = 0$, $\mu_2 = -1.5$, $\sigma_1 = 0.3$, and $\sigma_2 = 1.5$.

Let us consider two simple experiments with equal number of points in each cluster. We keep increasing the number of points in the clusters for each experiment, and cluster the data using Algorithm 1. We also cluster the same data set with GMM, through EM algorithm, and with standard k-means. In both of these cases we use the initialization from k-means++ [20] and we run the algorithms few times with different initializations and choose the answer with best objective function value. We use (45) to measure the clustering quality. Furthermore, we pick 100 different samples for each configuration, and show the average accuracy with errorbars indicating the standard error. In Fig. 1a we have data from normal distributions, where we can see that all the three methods perform closely, with a slight advantage of GMM, as expected, since it is the true model for the data. Energy statistics performs slightly better than k-means in this experiment. Now in Fig 1b we consider one example of data from lognormal distributions. We can see that energy statistics clustering through Algorithm 1 provides a huge improvement compared to both GMM and k-means, which basically cluster at chance in this case. GMM is worst than chance since sometimes it is not able to estimate the parameters, thus giving zero accuracy. These two simple experiments illustrate how energy statistics based clustering is nonparametric, being able to provide high quality clustering in settings where data comes from very different distributions.

V. ITERATIVE ALGORITHM FOR ENERGY STATISTICS CLUSTERING

In this section we will introduce a new iterative algorithm to find a local maximizer of the QCQP (18), however, due to Proposition 3 we can also find an approximate solution by the well-known kernel k-means algorithm, which for convenience will also be restated in the present context. First, let us introduce some base notation.

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

$$\max_{\{\mathcal{C}_1,\dots,\mathcal{C}_k\}} \left\{ Q = \sum_{j=1}^{k} \frac{Q_j}{n_j} \right\}, \qquad Q_j = \sum_{x,y\in\mathcal{C}_j} K(x,y), \tag{46}$$

where Q_j represents an internal energy cost of cluster $\mathcal{C}_j$, and Q is the total energy cost where each individual cluster cost 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

$$Q_\ell(x_i) \equiv \sum_{y\in\mathcal{C}_\ell} K(x_i,y) = G_{i\bullet}\cdot Z_{\bullet\ell}, \tag{47}$$

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

A. Kernel *k*-Means Algorithm

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

$$J^{(\ell)}(x_i) \equiv -\frac{2}{n_\ell} Q_\ell(x_i) + \frac{1}{n_\ell^2} Q_\ell, \tag{48}$$

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 $\ell = 1,\dots,k$. This procedure is performed for every data point, and repeated until convergence, i.e. until no new assignments are made. The whole procedure shown in Algorithm 2. It can be shown that this algorithm converges provided G is positive semidefinite. Although our notation looks a little different than the

Algorithm 2 Kernel k-means algorithm to find an approximate solution to (18).

```
input  number of clusters $k$, Gram matrix $G$, initial label matrix $Z = Z_0$
output  label matrix $Z$
 1: $\boldsymbol{q} \leftarrow (Q_1, \dots, Q_k)^\top$ have the costs of each cluster, according to (46)
 2: $\boldsymbol{n} \leftarrow (n_1, \dots, n_k)^\top$ have the number of points in each cluster, obtained from $D = Z^\top Z$
 3: repeat
 4:    for $i = 1, \dots, n$ 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 (48), for $\ell = 1, 2, \dots, k$
 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 $\boldsymbol{n}$: $n_j \leftarrow n_j - 1$ and $n_{j^\star} \leftarrow n_{j^\star} + 1$
10:          update $\boldsymbol{q}$: $q_j \leftarrow q_j - 2Q_j(x_i)$ and $q_{j^\star} \leftarrow q_{j^\star} + 2Q_{j^\star}(x_i)$
11:       end if
12:    end for
13: until convergence
```

standard kernel k-means found in the literature [10, 11], this is precisely the same algorithm but written in a more concise and explicit way.

Notice that to compute the first term in (48) 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 $\mathcal{O}(kn')$.

B. Energy Cost Algorithm

Now let us consider a different algorithm, which is based on the change in the within energy statistics when moving a given data point to a different cluster. Suppose we have a data point $x_i \in \mathcal{X}$ which is currently assigned to cluster $\mathcal{C}_j$, yielding a total energy cost function (46) denoted by $Q^{(j)}$. Let us consider the change in the total energy by moving x_i to cluster $\mathcal{C}_\ell$. Denote the new energy cost after moving x_i to $\mathcal{C}_\ell$ by $Q^{(\ell)}$. It is straightforward to see that

$$
\begin{aligned}
\Delta Q^{j\to\ell}(x_i) &\equiv Q^{(\ell)} - Q^{(j)} \\
&= \frac{1}{n_j - 1}\left[\frac{Q_j}{n_j} - 2Q_j(x_i)\right] - \frac{1}{n_\ell + 1}\left[\frac{Q_\ell}{n_\ell} - 2\big(Q_\ell(x_i) + K(x_i, x_i)\big)\right].
\end{aligned} \tag{49}
$$

Thus, if $\Delta Q^{j\to\ell}(x_i) > 0$ we get closer to a maximum of (46) by moving x_i to $\mathcal{C}_\ell$, otherwise we better keep x_i in $\mathcal{C}_j$. Based on this we propose an algorithm where the iterates are performed 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, $\Delta Q^{j\to\ell}(x_i)$ for $\ell = 1, \dots, k$ with $\ell \neq j$. We then 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 convergence, i.e. until no points are assigned to new clusters. This whole procedure is described in Algorithm 3. Note that (49) assures that the objective function is monotonically increasing at each iteration, therefore the algorithm converges in a finite number of steps.

Notice that 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 previous algorithms assume that G is given. There are more efficient methods to compute G, specially if it is sparse. 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 $\boldsymbol{q}$ we have $\mathcal{O}(n_1^2 + \cdots + n_k^2) = \mathcal{O}(k \max_j n_j^2)$. These operations, however, only need to be performed a single time. Now 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 (47) is $\mathcal{O}(n_j)$, thus the cost of step 8 in Algorithm 3 is $\mathcal{O}(k \max_j n_j)$ for $j = 1, \dots, k$. 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.

Algorithm 3 Energy cost algorithm to find an approximate solution to (18).

```
input number of clusters $k$, Gram matrix $G$, initial label matrix $Z = Z_0$
output label matrix $Z$
$\boldsymbol{q} \leftarrow (Q_1, \dots, Q_k)^\top$ have the energy costs of each cluster, according to (46)
$\boldsymbol{n} \leftarrow (n_1, \dots, n_k)^\top$ have the number of points in each cluster, obtained from $D = Z^\top Z$
repeat
  for $i = 1, \dots, n$ 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 $\ell = 1, 2, \dots, k$ 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 $\boldsymbol{n}$: $n_j \leftarrow n_j - 1$ and $n_{j^\star} \leftarrow n_{j^\star} + 1$
      update $\boldsymbol{q}$: $q_j \leftarrow q_j - 2Q_j(x_i)$ and $q_{j^\star} \leftarrow q_{j^\star} + 2\left(Q_{j^\star}(x_i) + G_{ii}\right)$
    end if
  end for
until convergence
```

VI. 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 associated kernel (8). We thus have

$$\rho(x, y) = \|x - y\|, \qquad K(x, y) = \tfrac{1}{2}\left(\|x\| + \|y\| - \|x - y\|\right). \tag{50}$$

We will consider other semimetrics/kernels as well, but the above will be considered the standard kernel for energy statistics and will always be present in every experiment as a reference. Notice that this is a convention, we could have chosen any other semimetric as the standard. One of the main goals of the following experiments is to compare Algorithm 3 to kernel k-means algorithm, described in Algorithm 2. Thus, for every kernel used in Algorithm 3, we also use the same kernel in Algorithm 2. Another goal is to compare Algorithm 3 with k-means and GMM (through expectation maximization algorithm), as these are the most used clustering algorithms in practice. Since for synthetic data the true labels are available, our measure of clustering quality will be (45). Moreover, for all algorithms, we always choose the initialization from k-means++ [20].

We first consider clustering in high dimensions and analyze how the algorithms degrade as the number of dimensions increase, while keeping the number of points in each cluster fixed. The Bayes error is also kept fixed as ambient dimension increases. In Figure 2a we have data generated from D-variate normal distributions as follows:

$$\begin{aligned} & x \sim \tfrac{1}{2}\left[\mathcal{N}(\mu_1, \Sigma_1) + \mathcal{N}(\mu_2, \Sigma_2)\right], \\ & \mu_1 = (\underbrace{0, \dots, 0}_{\times D})^\top, \quad \mu_2 = 0.7 \times (\underbrace{1, \dots, 1}_{\times 10}, \underbrace{0, \dots, 0}_{\times (D-10)})^\top, \quad \Sigma_1 = \Sigma_2 = I_D. \end{aligned} \tag{51}$$

We only keep signal in in the first 10 dimensions of μ_2, and keep increasing the ambient dimension D. For each D, we perform 100 experiments, obtaining the clustering accuracy for each algorithm. We can see that GMM is not able to estimate the covariance matrix when the number of dimensions exceeds the number of points in each cluster, so it gives zero accuracy for $D \gtrsim 100$. In Figure 2b we have the same type of experiment but with

$$\begin{aligned} & x \sim \tfrac{1}{2}\left[\mathcal{N}(\mu_1, \Sigma_1) + \mathcal{N}(\mu_2, \Sigma_2)\right], \\ & \mu_1 = (\underbrace{0, \dots, 0}_{\times D})^\top, \, \mu_2 = 0.7 \times (\underbrace{1, \dots, 1}_{\times 10}, \underbrace{0, \dots, 0}_{\times (D-10)})^\top, \, \Sigma_1 = I_D, \, \Sigma_2 = \begin{pmatrix} \frac{1}{2} I_{10} & 0 \\ 0 & I_{D-10} \end{pmatrix}. \end{aligned} \tag{52}$$

Therefore, for both experiments shown in Figure 2 we can see a better performance of Algorithm 3 compared to the other ones, in particular compared to kernel k-means algorithm, where we recall that both aim at optimizing the same problem (see Proposition 3). Also, notice that k-means and GMM are consistently the right model for this dataset, so it is hard to perform better than these algorithms in this current setting. Notice that Algorithm 3 is more robust as the ambient dimension increases.

In Figure 3 we consider the effect of having unbalanced clusters. We generate data as

$$\begin{aligned} & x \sim \frac{n_1}{N} \mathcal{N}(\mu_1, \Sigma_1) + \frac{n_2}{N} \mathcal{N}(\mu_2, \Sigma_2), \quad \mu_1 = (0,0,0,0)^\top, \, \mu_2 = 1.5 \times (1,1,0,0)^\top, \\ & \Sigma_1 = I_4, \quad \Sigma_2 = \begin{pmatrix} 1/2 & 0 & 0 & 0 \\ 0 & 1/2 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}, \quad n_1 = N - m, \quad n_2 = N + m, \quad N = 200. \end{aligned} \tag{53}$$

We then increase m, i.e. we make the clusters progressively more unbalanced. We generate 100 experiments for each m, and plot the clustering accuracy versus m. As expected, GMM works better than the other algorithms in the case of unbalanced clusters. This is mostly due to its soft assignments. We can see that the other methods based on hard assignments

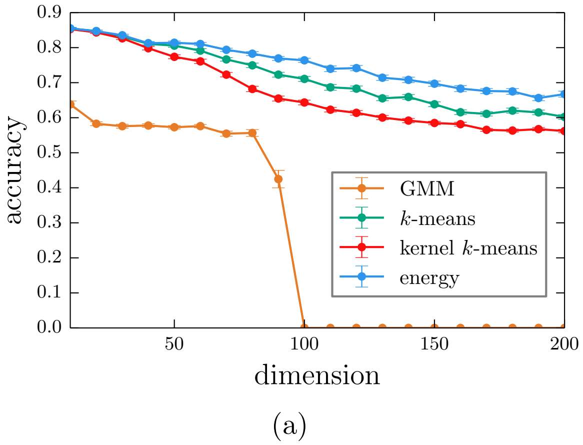


FIG. 2. Effect of increasing the ambient dimension while keeping Bayes error fixed, for two clusters with normally distributed data with 100 points in each cluster. (a) We increase D as described in (51). The blue line correspond to Algorithm 3, while the magenta line corresponds to kernel k-means, Algorithm 2. (b) The same but with data following (52). One notices that Algorithm 3 is more robust than the other ones.

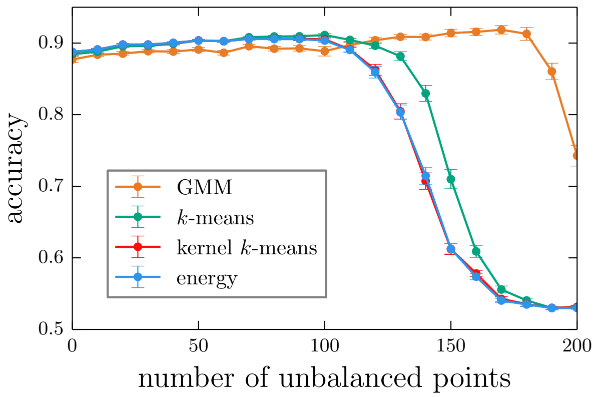


FIG. 3. Previous algorithms for unbalanced clusters, according to (53).

degrade similarly, and more rapidly than GMM. This indicates that a fuzzy version of energy statistics clustering should compensate for this effect.

Now, besides (50) we consider two other semimetrics:

$$\rho_{1/2}(x,y) = \|x-y\|^{1/2}, \qquad K(x,y) = \tfrac{1}{2}\left(\|x\|^{1/2} + \|y\|^{1/2} - \|x-y\|^{1/2}\right), \tag{54}$$

$$\rho_e(x,y) = 2 - 2e^{-\|x-y\|/2}, \qquad K(x,y) = e^{-\|x-y\|/2}. \tag{55}$$

In Figure 4a we have data in 20 dimensions distributed as

$$
\begin{aligned}
&x \sim \tfrac{1}{2}\left[\mathcal{N}(\mu_1,\Sigma_1)+\mathcal{N}(\mu_2,\Sigma_2)\right], \\
&\mu_1 = (\underbrace{0,\dots,0}_{\times 20})^\top, \quad \mu_2 = \tfrac{1}{2}(\underbrace{1,\dots,1}_{5},\underbrace{0,\dots,0}_{15})^\top, \quad \Sigma_1 = \tfrac{1}{2}I_{20}, \quad \Sigma_2 = I_{20}.
\end{aligned}
\tag{56}
$$

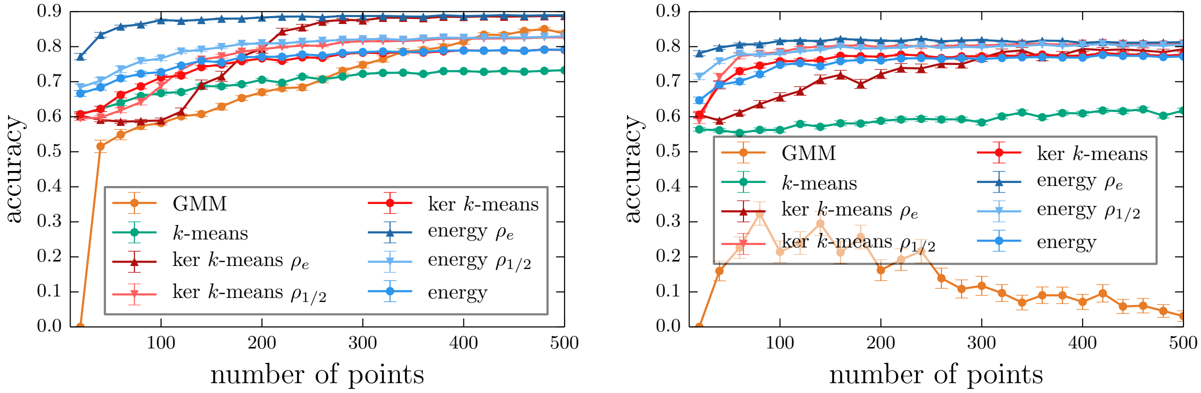
We increase the number of points in each cluster and show the clustering accuracy with different algorithms. The new semimetrics (54) and (55) are indicated in the legend. One can see that Algorithm 3 performs better than all the other ones, and in particular (55) provides better results. As the number of datapoints get large enough, GMM starts to be as accurate as clustering based on energy statistics, as it should since it is consistent model to the data. In Figure 4b, however, we use the same parameters as in (56) but now with data log-normally distributed:

$$
x \sim \tfrac{1}{2}\left[e^{\mathcal{N}(\mu_1,\Sigma_1)}+e^{\mathcal{N}(\mu_2,\Sigma_2)}\right]. \tag{57}
$$

We see that clustering based on energy statistics still performs accurately for this kind of data, while k-means works a little bit better than chance, and GMM is not even able to estimate the parameters. Again, (55) provides slightly better results than (50) or (54). Notice also that Algorithm 3 performs better than Algorithm 2. Both experiments in Figure 4 shows that energy statistics clustering is nonparametric, since it is able to cluster data coming from very different distributions.

VII. CONCLUSION

In this paper we have considered clustering from the perspective of energy statistics, which provides a nonparametric test for equality of distributions. Based on this, we showed that the clustering problem reduces to a quadratically constrained optimization problem (QCQP), as described in Proposition 2. Moreover, we showed that clustering based on energy statistics is equivalent to kernel k-means optimization problem, once the kernel is fixed; see Proposition 3. Our results imply that kernel k-means approach to clustering is actually a consequence of energy statistics theory, and thus place this method into a principled statistical basis. As already known [11], this approach is related to spectral clustering, and graph partitioning problems. Therefore, all these problems may be seen as arising naturally from energy statistics clustering. It is important to mention that energy



(a) (b)

FIG. 4. (a) Data normally distributed as in (56). We increase the number of points in each cluster to illustrate the statistical consistency of the algorithms. (b) The same experiment but for data following (57). In both experiments, for each case we run every algorithm 100 times and show the average results. One can see the better performance of energy statistics clustering, Algorithm 3, and in particular by using the semimetric (55). These two figures illustrate that energy statistics clustering is nonparametric since it works well for very different distributions.

statistics clustering, as formulated here, is valid for arbitrary metric spaces of negative type, and makes no assumptions about the distribution of the data. Moreover, it does not rely on the concept of a cluster mean, even implicitly.

We also proposed Algorithm 3 as an alternative to the well-known kernel k-means algorithm (see Algorithm 2), where both have the same time complexity. The numerical results show that Algorithm 3 might provide better clustering accuracy and is more robust than kernel k-means algorithm. Since there exists a huge literature about kernel k-means, and approximation methods to make it faster, with applications to several artificial and real data, we limited ourselves to analyze few but carefully designed experiments, which illustrates the advantages of Algorithm 3.

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

[5] 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.

[6] 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.

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

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

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

[10] 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.

[11] 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.

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

[13] A. Gretton, K. M. Borgwardt, M. J. Rasch, B. Schölkopf, and A. Smola. A Kernel Two-Sample Test. *J. Mach. Learn. Res.*, 13:723–773, 2012.

[14] 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.

[15] 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.

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

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

[18] 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.

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

[20] 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.