Energy Clustering

Guilherme França, Maria Rizzo and Joshua Vogelstein

Abstract—Energy statistics was proposed by Székely in the 80's inspired by the Newtonian gravitational potential from classical mechanics and it provides a hypothesis test for equality of distributions. It was further generalized from Euclidean spaces to metric spaces of strong negative type, and more recently, a connection with reproducing kernel Hilbert spaces (RKHS) was established. In this paper, we consider the problem of clustering data from the perspective of energy statistics theory. We provide a precise mathematical formulation yielding a quadratically constrained quadratic program (QCQP) in the associated RKHS, thus establishing the connection with kernel methods. Moreover, we also propose a weighted version of energy statistics applied to clustering, making connection to graph partitioning problems. To find local solutions of such QCQP, we propose an iterative algorithm based on Hartigan's method, which in this case has the same computational cost as kernel k-means algorithm, which is on Lloyd's heuristic, but usually with better clustering quality. We provide carefully designed numerical experiments showing the superiority of the proposed method compared to kernel k-means, spectral clustering, standard k-means, and Gaussian mixture models in a variety of settings, specially in high dimensions. We also employ the proposed method to an important real dataset describing protein expressions of neural synapses.

Index Terms—Clustering, Energy Statistics, Kernel Methods.

1 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 model-free hypothesis test for equality of distributions which is achieved under minimum energy. When probability distributions are different the "statistical potential energy" diverges as sample size increases, while tends to a nondegenerate limit distribution when probability distributions are equal. 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. Moreover, in Euclidean spaces, an application of energy statistics to clustering was already proposed [4]. We refer the reader to [1], and references therein, for an overview of energy statistics theory and its applications.

In its original formulation, energy statistics has a compact representation in terms of expectations of pairwise Euclidean distances, providing straightforward empirical estimates. The notion of distance covariance was further generalized from Euclidean spaces to metric spaces of strong negative type [5]. Furthermore, the link between energy distance based tests and kernel based tests has been recently established [6] through an equivalence between generalized energy distances to maximum mean discrepancies (MMD), which are distances between embeddings of distributions in reproducing kernel Hilbert spaces (RKHS). This equivalence immediately relates energy statistics to kernel methods often used in machine learning, and form the basis of our approach.

Clustering has such a long history in machine learning, making it impossible to mention all important contributions in a short space. Perhaps, the most used method is k-means [7], [8], [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, making it sensitive to outliers. Nevertheless, k-means works very well when data is linearly separable in Euclidean space. Gaussian mixture models (GMM) is another very common approach, providing more flexibility than k-means, however, it still makes strong assumptions about the distribution of the data.

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. A well-known kernel based clustering method is kernel k-means, which is precisely k-means formulated in the feature space [11]. Furthermore, kernel k-means algorithm [13], [14] is still based on Loyd's heuristic [7] of grouping points that are closer to a cluster center. We refer the reader to [15] for a survey of clustering methods.

Although clustering from energy statistics, in Euclidean spaces, was considered in [4], the precise optimization problem behind this approach remains obscure, as well as the connection with 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

- *G. França is with the Center for Imaging Science, Johns Hopkins University; J. Vogelstein is with the Center for Imaging Science, Department of Biomedical Engineering and Institute for Computational Medicine, Johns Hopkins University.*
E-mail: guifranca@jhu.edu; jovo@jhu.edu
- *M. Rizzo is with the Department of Mathematics and Statistics, Bowling Green State University.*
E-mail: mrizzo@bgsu.edu

Manuscript received March 15, 2018; revised March 15, 2018.

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. This immediately establishes the connection between energy statistics based clustering, or *energy clustering* for short, with kernel methods. Moreover, our formulation holds beyond the Euclidean case, more precisely, it holds for general semimetric spaces of negative type. We also show that such QCQP is equivalent to kernel k-means optimization problem when the kernel is fixed. However, energy statistics is able to fix a family of kernels. Therefore, one can see kernel k-means as a consequence of energy statistics theory. The equivalence between kernel k-means, spectral clustering, and graph partitioning problems is well-known [13], [14]. We further demonstrate how these relations arise from a weighted version of energy statistics.

Our main algorithmic contribution is to use Hartigan's method [16] to find local solutions of the above mentioned QCQP, which is NP-hard in general. Hartigan's method was also used in [4], but without any connection to kernels. More importantly, the advantages of Hartigan's over Lloyd's method was already demonstrated in some simple settings [17], [18], but apparently this method did not receive the deserved attention. To the best of our knowledge, Hartigan's method was not previously employed together with kernel methods. We provide a fully kernel based Hartigan's algorithm for clustering, where the kernel is fixed by energy statistics. We make clear the advantages of this proposal versus Lloyd's method, which kernel k-means algorithm is based upon and will also be used to solve our QCQP. We show that both algorithms have the same time complexity, but Hartigan's method in kernel spaces offer several advantages since it is able to escape local minima of Lloyd's method. Furthermore, in the examples considered in this paper, the proposed method also provides superior performance compared spectral clustering, which is more expensive and in fact solves a relaxed version of our QCQP.

Our numerical results provide compelling evidence that Hartigan's method applied to energy clustering is more accurate and robust than kernel k-means algorithm. Furthermore, our experiments illustrate the flexibility of energy clustering, showing that it is able to perform accurately on data coming from 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, however, it is significantly better on data that is not normally distributed. Its superiority in high dimensions is striking, being more accurate than k-means and GMM even on Gaussian settings.

2 Background

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

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

called *energy distance* [1], is rotationally invariant and nonnegative, $\mathcal{E}(P,Q) \ge 0$, 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,

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

where $0 < \alpha \le 2$. This quantity is also nonnegative, $\mathcal{E}_\alpha(P,Q) \ge 0$. 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 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

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

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*

$$\mathcal{E}(P,Q) \equiv 2\mathbb{E}\rho(X,Y) - \mathbb{E}\rho(X,X') - \mathbb{E}\rho(Y,Y'). \quad (3)$$

For spaces of negative type there exists a Hilbert space $\mathcal{H}$ and a map $\varphi : \mathcal{X} \to \mathcal{H}$ such that $\rho(X,Y) = \|\varphi(X) - \varphi(Y)\|^2_{\mathcal{H}}$. This allows us to compute quantities related to probability distributions over $\mathcal{X}$ in the associated 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 proper metric. Our energy clustering formulation, proposed in the next section, will be based on the generalized energy distance (3).

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. Defining the Gram matrix G with elements $G_{ij} = K(x_i, x_j)$, this is equivalent to $G = G^\top$ being positive semidefinite, i.e. $v^\top G\, v \ge 0$ 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

$$\gamma_K(P,Q) \equiv \| h_P - h_Q \|_{\mathcal{H}_K}. \tag{4}$$

This can also be written as [20]

$$\gamma_K^2(P,Q) = \mathbb{E}K(X,X') + \mathbb{E}K(Y,Y') - 2\mathbb{E}K(X,Y) \tag{5}$$

where $X, X' \overset{iid}{\sim} P$ and $Y, Y' \overset{iid}{\sim} Q$. From the equality between (4) and (5) we also have $\langle h_P, h_Q \rangle_{\mathcal{H}_K} = \mathbb{E}\, K(X,Y)$.

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

$$K(x,y) \equiv \tfrac{1}{2}\left[\rho(x,x_0) + \rho(y,x_0) - \rho(x,y) \right]. \tag{6}$$

Then, it can be shown that K is positive definite if and only if ρ is a semimetric of negative type. 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

$$\begin{aligned} \rho(x,y) &= K(x,x) + K(y,y) - 2K(x,y) \\ &= \| h_x - h_y \|^2_{\mathcal{H}_K} \end{aligned} \tag{7}$$

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 said to be equivalent kernels.

Now we can state the equivalence between the generalized energy distance (3) 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 (7) into (3), and using (5), yields

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

Due to (4) we can compute the energy distance $\mathcal{E}(P,Q)$ between two probability distributions using the inner product of $\mathcal{H}_K$.

Finally, let us recall the main formulas from generalized energy statistics for the 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 disjoint partition $\mathbb{X} = \bigcup_{j=1}^k \mathcal{C}_j$, with $\mathcal{C}_i \cap \mathcal{C}_j = \emptyset$. Each expectation in the generalized energy distance (3) can be computed through the function

$$g(\mathcal{C}_i, \mathcal{C}_j) \equiv \frac{1}{n_i n_j} \sum_{x \in \mathcal{C}_i} \sum_{y \in \mathcal{C}_j} \rho(x,y), \tag{8}$$

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

$$W \equiv \sum_{j=1}^k \frac{n_j}{2} g(\mathcal{C}_j, \mathcal{C}_j), \tag{9}$$

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

$$S \equiv \sum_{1 \le i < j \le k} \frac{n_i n_j}{2n} \left[2g(\mathcal{C}_i, \mathcal{C}_j) - g(\mathcal{C}_i, \mathcal{C}_i) - g(\mathcal{C}_j, \mathcal{C}_j) \right], \tag{10}$$

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 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 $H_1 : P_i \ne P_j$ for at least two $i \ne j$, we have that $S \to \infty$. Note that this test does not make any assumptions about the form of distribution P_j, thus it is said to be distribution-free.

One can make a physical analogy by thinking 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 $S(P_1, \dots, P_k)$ 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.

3 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 in the associated RKHS.

Due to the previous test statistic for equality of distributions, the obvious criterion for clustering data is to maximize S, given by (10), 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, the test statistic 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 S is, however, equivalent to minimizing W, given by (9), which has a more convenient form.

Lemma 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. For a fixed integer k, the partition $\mathbb{X} = \bigcup_{j=1}^k \mathcal{C}_j^\star$, where $C_i^\star \cap C_j^\star = \emptyset$ for all $i \ne j$, maximizes the between-sample statistic S, defined in equation* (10), *if and only if*

$$\{\mathcal{C}_1^\star, \dots, \mathcal{C}_k^\star\} = \operatorname*{arg\,min}_{\mathcal{C}_1, \dots, C_k} W(\mathcal{C}_1, \dots, \mathcal{C}_k) \tag{11}$$

where the within energy dispersion W is defined by (9).

Proof. From (9) and (10) we have that

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

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. In the Euclidean case, the optimization problem (11) was already proposed in [4]. Thus, Lemma 1 provides a generalization to arbirary spaces of negative type. Moreover, it is important to note that this problem is equivalent to maximizing S which is the test statistic for equality of distributions. More importantly, in this current form, the relation of problem (11) with kernels and other clustering methods is obscure. In the following, we show what is the explicit optimization problem behind (11) in the corresponding RKHS, establishing the connection with kernel methods.

Based on the relation between kernels and semimetrics of negative type, assume that the kernel $K : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ generates ρ. Define the Gram matrix

$$G \equiv \begin{pmatrix} K(x_1,x_1) & K(x_1,x_2) & \cdots & K(x_1,x_n) \\ K(x_2,x_1) & K(x_2,x_2) & \cdots & K(x_2,x_n) \\ \vdots & \vdots & \ddots & \vdots \\ K(x_n,x_1) & K(x_n,x_2) & \cdots & K(x_n,x_n) \end{pmatrix}. \quad (12)$$

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 $D = \operatorname{diag}(n_1,\dots,n_k)$ contains the number of points in each cluster. We also introduce the rescaled matrix $Y \equiv ZD^{-1/2}$. In component form they are given by

$$Z_{ij} \equiv \begin{cases} 1 & \text{if } x_i \in \mathcal{C}_j \\ 0 & \text{otherwise} \end{cases} \qquad Y_{ij} \equiv \begin{cases} \frac{1}{\sqrt{n_j}} & \text{if } x_i \in \mathcal{C}_j \\ 0 & \text{otherwise} \end{cases}. \quad (13)$$

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 (11) is NP-hard since it is a quadratically constrained quadratic program (QCQP) in the RKHS.

Proposition 2. *The optimization problem* (11) *is equivalent to*

$$\max_Y \operatorname{Tr}\left(Y^\top G Y\right) \quad \text{s.t. } Y \geq 0, Y^\top Y = I, YY^\top e = e, \quad (14)$$

where $e = (1,1,\dots,1)^\top \in \mathbb{R}^n$ *is the all-ones vector, and* G *is the Gram matrix* (12).

Proof. From (7), (8), and (9) we have

$$\begin{aligned} W &= \frac{1}{2}\sum_{j=1}^k \frac{1}{n_j} \sum_{x,y\in\mathcal{C}_j} \rho(x,y) \\ &= \sum_{j=1}^k \sum_{x\in\mathcal{C}_j} \left(K(x,x) - \frac{1}{n_j}\sum_{y\in\mathcal{C}_j} K(x,y) \right). \end{aligned} \quad (15)$$

Note that the first term is global so it does not contribute to the optimization problem. Therefore, minimizing (15) is equivalent to

$$\max_{\mathcal{C}_1,\dots,\mathcal{C}_k} \sum_{j=1}^k \frac{1}{n_j} \sum_{x,y\in C_j} K(x,y). \quad (16)$$

But

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

where we used the definitions (12) and (13). Notice that $n_j^{-1} = D_{jj}^{-1}$, where $D = \operatorname{diag}(n_1,\dots,n_k)$ contains the number of points in each cluster, thus the objective function in (16) is equal to $\sum_{j=1}^k D_{jj}^{-1}\left(Z^\top G Z\right)_{jj} = \operatorname{Tr}\left(D^{-1}Z^\top G Z\right)$. Now we can use the cyclic property of the trace, and by the definition of the matrix Z in (13), we obtain the following integer programing problem:

$$\begin{aligned} &\max_Z \operatorname{Tr}\left(\left(ZD^{-1/2}\right)^\top G \left(ZD^{-1/2}\right) \right) \\ &\text{s.t. } Z_{ij} \in \{0,1\}, \textstyle\sum_{j=1}^k Z_{ij} = 1, \sum_{i=1}^n Z_{ij} = n_j. \end{aligned} \quad (17)$$

Now we write this in terms of the matrix $Y = ZD^{-1/2}$. The objective function immediately becomes $\operatorname{Tr}\left(Y^\top G Y\right)$. Notice that the above constraints imply that $Z^T Z = D$, 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 (14), which comes from the last constraint in (17). In matrix form this reads $Z^T e = De$. 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, problem (17) is equivalent to (14) . □

Based on Proposition 2, to group data $\mathbb{X} = \{x_1,\dots,x_n\}$ into k clusters we first compute the Gram matrix G and then solve the optimization problem (14) 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 $x_i \in \mathcal{C}_j$. This optimization problem is nonconvex, and also NP-hard, thus a direct approach 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

$$\max_Y \operatorname{Tr}\left(Y^\top G Y\right) \qquad \text{s.t. } Y^\top Y = I$$

has a well-known closed form solution $Y^\star = UR$, where the columns of $U \in \mathbb{R}^{n\times k}$ contain the top 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 assumes the value $\max \operatorname{Tr}\left(Y^{\star\top} G Y^\star\right) = \sum_{i=1}^k \lambda_i$. Spectral clustering is based on the above approach, where one further normalize the rows of $Y^\star$, then cluster the resulting rows as data points. A procedure on these lines was proposed in the seminal papers [22], [23].

We remark that problem (14) is valid for data living in an *arbitrary* space of negative type. The standard energy distance (2) fixes a family of choices in Euclidean spaces given by

$$\begin{aligned} &\rho_\alpha(x,y) = \|x-y\|^\alpha, \\ &K_\alpha(x,y) = \tfrac{1}{2}\left(\|x\|^\alpha + \|y\|^\alpha - \|x-y\|^\alpha\right), \end{aligned}$$

for $0 < \alpha \leq 2$ and we fix $x_0 = 0$ in (6). The same would be valid for data living in a more general semimetric space $(\mathcal{X},\rho)$ where ρ fixes the kernel. In practice, the clustering quality strongly depends on the choice of a suitable ρ. Nevertheless, if prior information is available to make this choice, it can be immediately incorporated into the optimization problem (14).

One may wonder how energy 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 (12), there exists a map $\varphi : \mathcal{X} \to \mathcal{H}_K$ such that $K(x,y) = \langle \varphi(x), \varphi(y) \rangle$. The kernel k-means optimization problem is defined by

$$\min_{\mathcal{C}_1,\dots,\mathcal{C}_k} \left\{ J(\mathcal{C}_1,\dots,\mathcal{C}_k) \equiv \sum_{j=1}^{k} \sum_{x\in\mathcal{C}_j} \|\varphi(x) - \varphi(\mu_j)\|^2 \right\} \tag{18}$$

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 is Lloyd's method [7]. It is known [13], [14] that problem (18) can be cast into a trace maximization in the same form as (14). The next result makes this explicit, showing that (11) and (18) are actually equivalent.

Proposition 3. *For a fixed kernel, the optimization problem* (11) *based on energy statistics is equivalent to the kernel k-means optimization problem* (18)*, and both are equivalent to* (14)*.*

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

$$J = \sum_{j=1}^{k}\sum_{x\in\mathcal{C}_j} \left(K(x,x) - \frac{2}{n_j}\sum_{y\in\mathcal{C}_j} K(x,y) + \frac{1}{n_j^2}\sum_{y,z\in\mathcal{C}_j} K(y,z) \right). \tag{19}$$

The first term is global so it does not contribute. 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, problem (18) is equivalent to

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

which is exactly the same as (16) from the energy statistics formulation. Therefore, once the kernel K is fixed, the function W given by (9) is the same as J in (18). The remaining of the proof proceeds as already shown in the proof of Proposition 2. □

The above result shows that kernel k-means optimization problem is equivalent to the clustering problem formulated in the energy statistics framework, when operating on the same kernel. This result could not have been anticipated a priori. It is important to note, however, that kernel k-means formulation (18) is completelly heuristic, while problem (21) was derived from first-principles using energy statistics theory, which includes the kernel construction. Therefore, one may view kernel k-means optimization problem as a consequence of energy statistics.

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 (14) [13], [14]. One can thus use kernel k-means algorithm to solve these problems as well. This correspondence involves a weighted version of problem (14), that will be demonstrated in the following from the perspective of energy statistics.

4 A Weighted Generalization

We now generalize energy statistics to incorporate weights associated to each data point. Let $w(x)$ be a weight function associated to point $x \in \mathcal{X}$. Define

$$g(\mathcal{C}_i,\mathcal{C}_j) \equiv \frac{1}{s_i s_j}\sum_{x\in\mathcal{C}_i}\sum_{y\in\mathcal{C}_j} w(x)w(y)\rho(x,y), \tag{20}$$

$$s_i \equiv \sum_{x\in\mathcal{C}_i} w(x), \qquad s \equiv \sum_{j=1}^{k} s_j .$$

Replace (20) in the formulas (9) and (10) with $n_i \to s_i$ and $n \to s$. With these changes, Proposition 1 remains the unaltered, so the clustering problem becomes

$$\min_{\mathcal{C}_1,\dots,\mathcal{C}_k} \left\{ W(\mathcal{C}_1,\dots,\mathcal{C}_k) \equiv \sum_{j=1}^{k} \frac{s_j}{2} g(\mathcal{C}_j,\mathcal{C}_j) \right\} \tag{21}$$

where now g is given by (20). Define the following:

$$Y_{ij} \equiv \begin{cases} \frac{1}{\sqrt{s_j}} & \text{if } x_i \in \mathcal{C}_j \\ 0 & \text{otherwise} \end{cases}, \qquad \mathcal{W} \equiv \operatorname{diag}(w_1,\dots,w_n),$$
$$H \equiv \mathcal{W}^{1/2} Y, \qquad\qquad \omega \equiv \mathcal{W} e, \tag{22}$$

where $w_i = w(x_i)$ and $e \in \mathbb{R}^n$ is the all-ones vector. The analogous of Proposition 2 is as follows.

Proposition 4. *The weighted energy clustering given by problem* (21) *is equivalent to*

$$\begin{aligned} &\max_{H} \operatorname{Tr}\left\{ H^\top (\mathcal{W}^{1/2} G \mathcal{W}^{1/2}) H \right\} \\ &\text{s.t. } H \ge 0,\ H^\top H = I,\ H H^\top \omega = \omega, \end{aligned} \tag{23}$$

where G is the Gram matrix (12)*, $\omega = (w_1,\dots,w_n)^T$ contains the weights of each point, and $\mathcal{W} = \operatorname{diag}(\omega)$.*

Proof. Replacing (7) and eliminating the global terms which do not contribute, the optimization problem (21) becomes

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

This objective function can be written as

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

[1] When we refer to kernel k-means *problem* we mean specifically the optimization problem (18), which should not be confused with kernel k-means *algorithm* which is just one possible recipe to solve (18). The distinction should also be clear from the context.

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

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

where $\delta_{ij} = 1$ if $i = j$ and $\delta_{ij} = 0$ if $i \neq j$ is the Kronecker delta. Therefore, $H^\top H = I$. This is a constraint on the rows of H. To obtain a condition on its columns, observe that

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

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 $i \in \{1, \dots, k\}$, 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 $H H^\top \mathcal{W}^{1/2} e = \mathcal{W}^{1/2} e$, which gives the constraint $H H^\top \omega = \omega$. □

4.1 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, which 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

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

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

$$\max_{\mathcal{C}_i, \dots, \mathcal{C}_k} \sum_{j=1}^{k} \frac{\text{links}(\mathcal{C}_j, \mathcal{C}_j)}{s_j} \tag{24}$$

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

$$\min_{\mathcal{C}_i, \dots, \mathcal{C}_k} \sum_{j=1}^{k} \frac{\text{links}(\mathcal{C}_j, \mathcal{V} \backslash \mathcal{C}_j)}{s_j} \tag{25}$$

minimizes the cut between clusters. These two problems are equivalent, in analogous way as minimizing (9) is equivalent to maximizing (10) 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 [22], [24], [25], [26] can be seen as a particular case of (24) or (25).

Consider the ratio association problem (24), whose objective function can be written as

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

with Z defined in (13) and Y in (22). Therefore, the ratio association problem can be written in the form (23), i.e.

$$\begin{aligned}&\max_{H} \operatorname{Tr}\left(H^\top \mathcal{W}^{-1/2} \mathcal{A} \mathcal{W}^{-1/2} H\right) \\ &\text{s.t. } H \geq 0,\, H^\top H = I,\, H H^\top \omega = \omega.\end{aligned}$$

This is exactly the same problem as weighted energy clustering with $G = \mathcal{W}^{-1} \mathcal{A} \mathcal{W}^{-1}$. Assuming this matrix is positive semidefinite, this generates a semimetric (7) for graphs given by

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

for vertices $i, j \in \mathcal{V}$. If we assume the graph has no self-loops we must replace $\mathcal{A}_{ii} = 0$ above. Above, the weight of node $i \in \mathcal{V}$ can be, for instance, its degree, $w_i = w(i) = d(i)$.

5 Two-Class Problem in One Dimension

Before stating a general algorithm to solve the optimization problem (14) we first consider the simplest possible case which is one-dimensional data and a two-class problem. This will be useful to test energy clustering on a simple setting.

Fixing $\rho(x, y) = |x - y|$ according to the standard energy distance, we can actually compute the function (8) in $\mathcal{O}(n \log n)$ and minimize W directly. This is done by noting that

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

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 we have

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

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 $\widetilde{\mathcal{C}} = [x_j \in \mathcal{C} : x_1 \leq x_2 \leq \cdots \leq x_n]$, we get

$$g(\widetilde{\mathcal{C}}, \widetilde{\mathcal{C}}) = \frac{2}{n^2} \sum_{\ell=1}^{n} (2\ell - 1 - n) x_\ell.$$

Note that the cost of computing $g(\widetilde{\mathcal{C}}, \widetilde{\mathcal{C}})$ 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, $\mathbb{X} = \bigcup_{j=1}^{k} \widetilde{\mathcal{C}}_j$, the within energy dispersion can be written explicitly as

$$W\big(\widetilde{\mathcal{C}}_1, \dots, \widetilde{\mathcal{C}}_k\big) = \sum_{j=1}^{k} \sum_{\ell=1}^{n_j} \frac{2\ell - 1 - n_j}{n_j}\, x_\ell. \tag{26}$$

For a two-class problem we can use the formula (26) to cluster the data through a simple algorithm as follows. We first order the entire dataset, $\mathbb{X} \to \widetilde{\mathbb{X}}$. Then we compute (26) for each possible split of $\widetilde{\mathbb{X}}$ and pick the point which gives the minimum value of W. This procedure is described in Algorithm 1 and called $\mathcal{E}^{1D}$-clustering. Note that this algorithm is deterministic, however, it only works for one-dimensional data with Euclidean distance. The total complexity of $\mathcal{E}^{1D}$-clustering is $\mathcal{O}(n \log n + n^2) = \mathcal{O}(n^2)$.

Algorithm 1 $\mathcal{E}^{1D}$-clustering algorithm to find local solutions to the optimization problem (11) for a two-class problem in one dimension.

```
input $\mathbb{X}$
output $Z$
sort $\mathbb{X}$, obtaining $\widetilde{\mathbb{X}} = [x_1, \dots, x_n]$
for $j \in [1, \dots, n]$ do
    $\widetilde{\mathcal{C}}_{1,j} \leftarrow [x_i : i = 1, \dots, j]$
    $\widetilde{\mathcal{C}}_{2,j} \leftarrow [x_i : i = j+1, \dots, n]$
    $W^{(j)} \leftarrow W\big(\widetilde{\mathcal{C}}_{1,j}, \widetilde{\mathcal{C}}_{2,j}\big)$ (see (26))
end for
$j^\star \leftarrow \arg\min_j W^{(j)}$
for $j = 1, \dots, n$ assign $Z_{j\bullet} \leftarrow \begin{cases} (1,0) & \text{if } j \le j^\star \\ (0,1) & \text{otherwise} \end{cases}$
```

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

$$\text{accuracy}(\hat{Z}) \equiv \max_{\sigma} \frac{1}{n} \sum_{i=1}^{n} \sum_{j=1}^{k} \hat{Z}_{i\sigma(j)} Z_{ij} \tag{27}$$

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 balanced two-class problem the value $1/2$ correspond to chance.

We now consider two simple experiments where we sample n points from a two-class mixture. We plot the average accuracy (27) versus n, with error bars indicating standard error. The data is clustered using $\mathcal{E}^{1D}$-clustering algorithm, GMM and k-means. For GMM and k-means we use the implementations from the well-known *scikit-learn* library in Python [27], where k-means is initialized through k-means++ procedure [28], and GMM is initialized with the output of k-means. We run both algorithms 5 times with different initializations and pick the answer with best objective function value. Notice that $\mathcal{E}^{1D}$-clustering does not require random initialization so we only run it once. For each n we use use 100 Monte Carlo runs. In Fig. 1a we have the results for data sampled from the Gaussian mixture

$$\begin{gathered} x \overset{iid}{\sim} \tfrac{1}{2}\mathcal{N}(\mu_1, \sigma_1^2) + \tfrac{1}{2}\mathcal{N}(\mu_2, \sigma_2^2), \\ \mu_1 = 1.5, \quad \sigma_1 = 0.3, \quad \mu_2 = 0, \quad \sigma_2 = 1.5. \end{gathered} \tag{28}$$

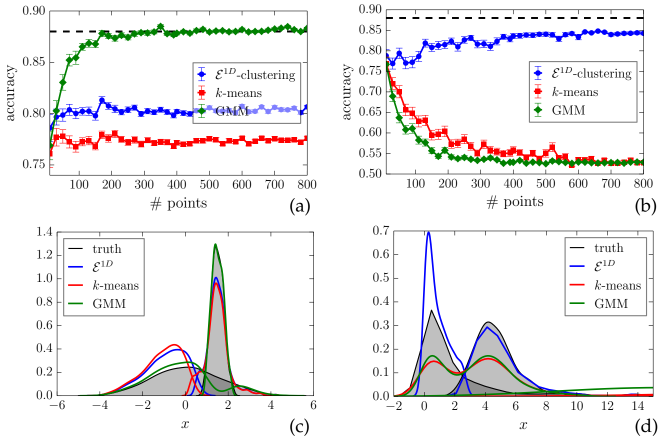

Fig. 1. $\mathcal{E}^{1D}$-clustering versus k-means and GMM. (a,b) We plot the mean accuracy (27) over 100 Monte Carlo trials, versus the number of sampled points. Error bars are standard error. The dashed line indicates Bayes accuracy (≈ 0.88 in both cases). (a) Clustering results for data normally distributed as in (28). (b) Data lognormally distributed as in (29). (c) Density estimation of each component in the mixture (28) after clustering 1000 sampled points using the three algorithms, compared to the ground truth. (d) The same but for lognormal data (29).

In this case the optimal accuracy obtained from Bayes classification error is ≈ 0.88, indicated by the dashed line in the plot. The three methods perform closely, with a slight advantage of GMM, as expected since it is a consistent model to the data, and $\mathcal{E}^{1D}$-clustering performs slightly better than k-means. In Fig. 1c we show a density estimation from clustering 1000 points from this mixture using the three algorithms. Notice that all of them are able to distinguish the two classes. On the other hand, in Fig. 1b we consider a mixture of lognormal distributions,

$$\begin{gathered} x \overset{iid}{\sim} \tfrac{1}{2}\exp\left\{\mathcal{N}(\mu_1, \sigma_1^2)\right\} + \tfrac{1}{2}\exp\left\{\mathcal{N}(\mu_2, \sigma_2^2)\right\}, \\ \mu_1 = 1.5, \quad \sigma_1 = 0.3, \quad \mu_2 = 0, \quad \sigma_2 = 1.5. \end{gathered} \tag{29}$$

The optimal Bayes accuracy is again ≈ 0.88. We can now see that $\mathcal{E}^{1D}$-clustering is still very accurate, while GMM and k-means basically cluster at chance. Density estimation after clustering 1000 points this mixture using the three algorithms are is shown in Fig. 1d. Note that only $\mathcal{E}^{1D}$-clustering was able to distinguish the two classes. k-means and GMM put most of the points in a single cluster, and points on the tail of the second component of (29) in the other cluster. The experiments of Fig. 1 illustrate how energy clustering is more flexible compared to k-means and GMM.

6 Iterative Algorithms

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

Consider the optimization problem (16) written as

$$\max_{\{\mathcal{C}_1, \dots, \mathcal{C}_k\}} \left\{ Q = \sum_{j=1}^{k} \frac{Q_j}{n_j} \right\}, \qquad Q_j \equiv \sum_{x,y \in \mathcal{C}_j} K(x, y), \tag{30}$$

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 points in $\mathcal{C}_j$. For a data point x_i we denote its own energy cost with the entire cluster $\mathcal{C}_\ell$ by

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

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

6.1 Lloyd's Method for Energy Clustering

To optimize kernel k-means objective function (19) we remove the global term and define the function

$$J^{(\ell)}(x_i) \equiv \frac{1}{n_\ell^2} Q_\ell - \frac{2}{n_\ell} Q_\ell(x_i). \tag{31}$$

We are thus solving

$$\min_Z \sum_{i=1}^{n} \sum_{\ell=1}^{k} Z_{i\ell} J^{(\ell)}(x_i).$$

One possible strategy is to assign x_i to cluster $\mathcal{C}_{j^\star}$ according to

$$j^\star = \operatorname*{arg\,min}_{\ell=1,\dots,k} J^{(\ell)}(x_i).$$

This is done for every data point x_i and repeated until convergence, i.e. until no new assignments are made. The entire procedure is described in Algorithm 2, which we name $\mathcal{E}^L$-clustering to emphasize that we are optimizing the within energy function W based on Lloyd's method [7]. It can be shown that this algorithm converges provided G is positive semidefinite.

$\mathcal{E}^L$-clustering is precisely kernel k-means algorithm [13], [14] but written more concisely and with the kernel induced by energy statistics. Indeed, recalling that $K(x, y) = \langle \varphi(x), \varphi(y) \rangle$ where $\varphi : \mathcal{X} \to \mathcal{H}_K$ is the feature map, we have from (31) that

$$\begin{aligned} J^{(\ell)}(x_i) &= \langle \varphi(\mu_\ell), \varphi(\mu_\ell) \rangle - 2 \langle \varphi(x_i), \varphi(\mu_\ell) \rangle \\ &= \| \varphi(x_i) - \varphi(\mu_\ell) \|^2 - \| \varphi(x_i) \|^2, \end{aligned}$$

where $\mu_\ell = \frac{1}{n_\ell} \sum_{x \in \mathcal{C}_\ell} x$ is the mean of cluster $\mathcal{C}_\ell$. Therefore, $\min_\ell J^{(\ell)}(x_i) = \min_\ell \| \varphi(x_i) - \varphi(\mu_\ell) \|^2$, i.e. we are assigning x_i to the cluster with closest center (in feature space), which is the familiar Lloyd's heuristic approach that kernel k-means is based upon.

To check the complexity of $\mathcal{E}^L$-clustering, notice that to compute the second term of $J^{(\ell)}(x_i)$ in (31) requires $\mathcal{O}(n_\ell)$ operations, and although the first term requires $\mathcal{O}(n_\ell^2)$ it only needs to be computed once outside loop through data points (step 1 of Algorithm 2). Therefore, the time complexity of $\mathcal{E}^L$-clustering 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 $\mathcal{O}(kn')$.

Algorithm 2 $\mathcal{E}^L$-clustering is Lloyd's method for energy clustering, which is precisely kernel k-means algorithm, with the kernel induced by energy statistics. This procedure finds local solutions to the optimization problem (14).

```
input  $k, G, Z \leftarrow Z_0$
output  $Z$
 1: $q \leftarrow (Q_1, \dots, Q_k)^\top$ (see (30))
 2: $n \leftarrow (n_1, \dots, n_k)^\top$
 3: repeat
 4:   for $i = 1, \dots, n$ do
 5:     let $j$ be such that $x_i \in \mathcal{C}_j$
 6:     $j^\star \leftarrow \arg\min_{\ell=1,\dots,k} J^{(\ell)}(x_i)$ (see (31))
 7:     if $j^\star \neq j$ then
 8:       $Z_{ij} \leftarrow 0$
 9:       $Z_{ij^\star} \leftarrow 1$
10:       $n_j \leftarrow n_j - 1$
11:       $n_{j^\star} \leftarrow n_{j^\star} + 1$
12:       $q_j \leftarrow q_j - 2Q_j(x_i)$
13:       $q_{j^\star} \leftarrow q_{j^\star} + 2Q_{j^\star}(x_i)$
14:     end if
15:   end for
16: until convergence
```

6.2 Hartigan's Method for Energy Clustering

We now consider Hartigan's method [16] applied to the optimization problem in the form (30), which gives a local solution to the QCQP defined in (14). The method is based in computing the maximum change in the total cost function Q when moving each data point to another cluster. More specifically, suppose point x_i is currently assigned to cluster $\mathcal{C}_j$ yielding a total cost function denoted by $Q^{(j)}$. Moving x_i to cluster $\mathcal{C}_\ell$ yields another total cost function denoted by $Q^{(\ell)}$. We are interested in computing the maximum change $\Delta Q^{j \to \ell}(x_i) \equiv Q^{(\ell)} - Q^{(j)}$, for $\ell \neq j$. From (30), by explicitly writing the costs related to these two cluster we obtain

$$\Delta Q^{j \to \ell}(x_i) = \frac{Q_\ell^+}{n_\ell + 1} + \frac{Q_j^-}{n_j - 1} - \frac{Q_j}{n_j} - \frac{Q_\ell}{n_\ell}$$

where Q_ℓ^+ denote the cost of the new ℓth cluster with the point x_i added to it, and Q_j^- is the cost of new jth cluster with x_i removed from it. Noting that $Q_\ell^+ = Q_\ell + 2Q_\ell(x_i) + G_{ii}$ and $Q_j^- = Q_j - 2Q_j(x_i) + G_{ii}$, we get the formula

$$\begin{aligned} \Delta Q^{j \to \ell}(x_i) = &\frac{1}{n_j - 1} \left[\frac{Q_j}{n_j} - 2Q_j(x_i) + G_{ii} \right] \\ &- \frac{1}{n_\ell + 1} \left[\frac{Q_\ell}{n_\ell} - 2Q_\ell(x_i) - G_{ii} \right]. \end{aligned} \tag{32}$$

Therefore, if $\Delta Q^{j \to \ell}(x_i) > 0$ we get closer to a maximum of (30) by moving x_i to $\mathcal{C}_\ell$, otherwise we keep x_i in $\mathcal{C}_j$.

We thus propose the following algorithm. 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 $\mathcal{C}_\ell$, i.e. $\Delta Q^{j \to \ell}(x_i)$ for $\ell = 1, \dots, k$ with $\ell \neq j$, where j denotes the index of its current partition, $x \in \mathcal{C}_j$. Hence, we choose

$$j^\star = \operatorname*{arg\,max}_{\ell=1,\dots,k \,|\, \ell \neq j} \Delta^{j \to \ell}(x_i).$$

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 $\mathcal{C}_j$. This process is repeated until no points are assigned to new clusters. The entire procedure is explicitly described in Algorithm 3, which we denote $\mathcal{E}^H$-clustering to emphasize that it is based on Hartigan's method. This method automatically ensures that the objective function is monotonically increasing at each iteration, and consequently the algorithm converges in a finite number of steps.

Algorithm 3 $\mathcal{E}^H$-clustering is Hartigan's method for energy clustering. This algorithm finds local solutions to the optimization problem (14). The steps 6, 12 and 13 are different than $\mathcal{E}^L$-clustering described in Algorithm 2.

```
input  $k$, $G$, $Z \leftarrow Z_0$
output  $Z$
 1: $q \leftarrow (Q_1, \dots, Q_k)^\top$ (see (30))
 2: $n \leftarrow (n_1, \dots, n_k)^\top$
 3: repeat
 4:   for  $i = 1, \dots, n$ do
 5:     let $j$ be such that $x_i \in \mathcal{C}_j$
 6:     $j^\star \leftarrow \arg\max_{\ell=1,\dots,k \,|\, \ell \neq j} \Delta Q^{j \to \ell}(x_i)$ (see (32))
 7:     if  $\Delta Q^{j \to j^\star}(x_i) > 0$  then
 8:       $Z_{ij} \leftarrow 0$
 9:       $Z_{ij^\star} \leftarrow 1$
10:       $n_j \leftarrow n_j - 1$
11:       $n_{j^\star} \leftarrow n_{j^\star} + 1$
12:       $q_j \leftarrow q_j - 2Q_j(x_i) + G_{ii}$
13:       $q_{j^\star} \leftarrow q_{j^\star} + 2Q_{j^\star}(x_i) + G_{ii}$
14:     end if
15:   end for
16: until convergence
```

The complexity analysis of $\mathcal{E}^H$-clustering is the following. Computing the Gram matrix 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 $\mathcal{E}^L$- and $\mathcal{E}^H$-clustering 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 $Q_\ell(x_i)$ is $\mathcal{O}(n_\ell)$, thus the cost of step 6 in Algorithm 3 is $\mathcal{O}(k \max_\ell n_\ell)$ for $\ell = 1, \dots, k$. For the entire dataset this gives a time complexity of $\mathcal{O}(nk \max_\ell n_\ell) = \mathcal{O}(kn^2)$. Note that this is the same cost as in $\mathcal{E}^L$-clustering, or kernel k-means algorithm. 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 we mention some important known 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 guaranteed to be satisfied by Lloyd's method, and consequently by $\mathcal{E}^L$-clustering algorithm. The next result indicates that Hartigan's method 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.*

The above theorem implies that $\mathcal{E}^L$-clustering cannot improve on a local optima of $\mathcal{E}^H$-clustering. On the other hand, $\mathcal{E}^H$ might improve on a local optima of $\mathcal{E}^L$. 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 the number of clusters k increases. The empirical results of [17] show that an implementation of Hartigan's method has comparable execution time to an implementation of Lloyd's method, but no explicit complexity was provided. We show that both $\mathcal{E}^L$- and $\mathcal{E}^H$-clustering have the same time complexity. To the best of our knowledge, Hartigan's method was not previously considered together with kernels, as we are proposing in $\mathcal{E}^H$-clustering algorithm.

In [18], Hartigan's method was applied to k-means problem with any Bregman divergence. It was shown that the number of Hartigan's local optima is upper bounded by $\mathcal{O}(1/k)$ [18, Proposition 5.1]. In addition, it was provided 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.

The above results indicate that Hartigan's method provides several advantages over Lloyd's method, a fact that will also be supported by our numerical experiments in the next section where $\mathcal{E}^H$ outperforms of $\mathcal{E}^L$ (kernel k-means) in several settings, specially in high dimensions.

7 Numerical Experiments

The main goal of this section is threefold. First, we want to compare $\mathcal{E}^H$-clustering in Euclidean space to k-means and GMM. Second, we want to compare $\mathcal{E}^H$-clustering, based on Hartigan's method, to $\mathcal{E}^L$-clustering or kernel k-means, based on Lloyd's method, and also to spectral clustering, when they all operate on the same kernel. Third, we want to illustrate the flexibility provided by energy clustering, which is able to cluster accurately in different settings while keeping the same kernel.

The following experimental setup holds unless specified otherwise. We consider $\mathcal{E}^H$-clustering, $\mathcal{E}^L$-clustering and spectral clustering with the following metrics and corresponding generating kernels:

$$\rho_\alpha(x, y) = \|x - y\|^\alpha, \tag{33}$$

$$\widetilde{\rho}_\sigma(x, y) = 2 - 2e^{-\frac{\|x-y\|}{2\sigma}}, \tag{34}$$

$$\widehat{\rho}_\sigma(x, y) = 2 - 2e^{-\frac{\|x-y\|^2}{2\sigma^2}}. \tag{35}$$

The relation between kernel and semimetric is given by formula (6) where we fix $x_0 = 0$. The standard ρ_1, from the original energy distance (1), will always be present in the experiments as a reference, being the implied choice unless explicitly mentioned. For k-means, GMM and spectral clustering we use the robust implementations of *scikit-learn* library [27], where k-means is initialized with k-means++ [28], and GMM with the output of k-means, making it more robust and preventing it from breaking in high dimensions. Spectral clustering implementation is based on [22]. We implemented $\mathcal{E}^L$-clustering as described in Algorithm 2, and $\mathcal{E}^H$-clustering as described in Algorithm 3. Both will also be initialized with k-means++. We run the algorithms 5 times with different initializations, picking the result with best objective function value. We evaluate clustering quality by the accuracy (27) based on the true labels. For each setting we show the average accuracy over 100 Monte Carlo trials, with error bars indicating standard error.

We briefly mention that we compared $\mathcal{E}^H$-clustering, as described in Algorithm 3, to $\mathcal{E}^{1D}$-clustering, described in Algorithm 1, for several univariate distributions. Both perform very closely. However, we omit these results since we will analyse more interesting scenarios in high dimensions.

From the results of [17], summarized in the end of the previous section, 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 data from the Gaussian mixture

$$\begin{aligned} &x \overset{iid}{\sim} \tfrac{1}{2}\mathcal{N}(\mu_1, \Sigma_1) + \tfrac{1}{2}\mathcal{N}(\mu_2, \Sigma_2), \quad \Sigma_1 = \Sigma_2 = I_D, \\ &\mu_1 = (\underbrace{0, \dots, 0}_{\times D})^\top, \qquad \mu_2 = 0.7(\underbrace{1, \dots, 1}_{\times 10}, \underbrace{0, \dots, 0}_{\times (D-10)})^\top. \end{aligned} \tag{36}$$

To get some intuition about how separated data points from each class are, we show scatter plots between the first 5 dimensions in Fig. 2a. Note that the Bayes error is fixed as D increases, yielding an optimal accuracy of ≈ 0.86. We sample 200 points on each trial. The results are shown in Fig. 3a. We can see that $\mathcal{E}^H$ and spectral clustering have practically the same performance, which is higher than $\mathcal{E}^L$-clustering (kernel k-means). Moreover, $\mathcal{E}^H$ outperforms k-means and GMM, where the improvement is noticeable specially in high dimensions. Note that in this setting k-means and GMM are consistent models to the data, however, energy clustering degrades much less as dimension increases.

Still for a two-class Gaussian mixture, we now allow the diagonal entries of one of the covariances to have different values by choosing

$$\begin{aligned} &x \overset{iid}{\sim} \tfrac{1}{2}\mathcal{N}(\mu_1, \Sigma_1) + \tfrac{1}{2}\mathcal{N}(\mu_2, \Sigma_2), \\ &\mu_1 = (\underbrace{0, \dots, 0}_{\times D})^\top, \qquad \mu_2 = (\underbrace{1, \dots, 1}_{\times 10}, \underbrace{0, \dots, 0}_{\times (D-10)})^\top, \\ &\Sigma_1 = I_D, \qquad \Sigma_2 = \left(\begin{array}{c|c} \widetilde{\Sigma}_{10} & 0 \\ \hline 0 & I_{D-10} \end{array} \right), \\ &\widetilde{\Sigma}_{10} = \operatorname{diag}(1.367, 3.175, 3.247, 4.403, 1.249, 1.969, \\ &\qquad 4.035, 4.237, 2.813, 3.637). \end{aligned} \tag{37}$$

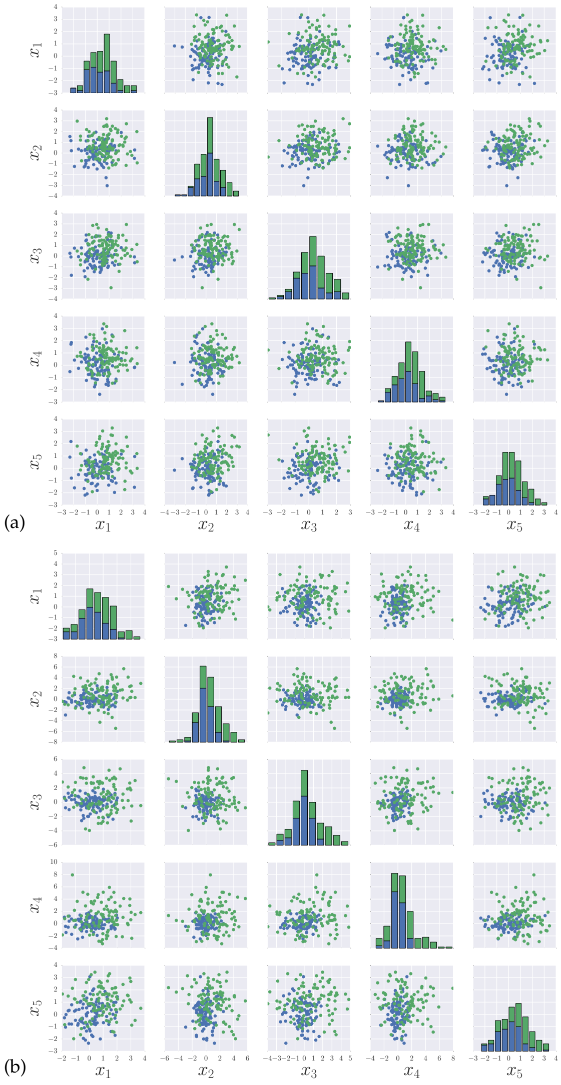

Fig. 2. Pair plots for the first 5 dimensions. (a) Data normally distributed as in (36). (b) Data normally distributed as in (37). We sample 200 points for both cases. We can see that there is a considerable overlap between the clusters.

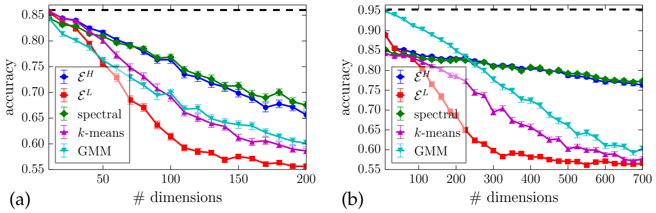

Fig. 3. Comparison of $\mathcal{E}^H$-clustering, $\mathcal{E}^L$-clustering (kernel k-means), spectral clustering, k-means and GMM in high dimensional Gaussian settings. We plot the mean accuracy versus the number of dimensions, with error bars indicating standard error from 100 Monte Carlo runs. (a) Data normally distributed as in (36), with Bayes accuracy ≈ 0.86, over the range $D \in [10, 200]$. (b) Data normally distributed as in (37), with Bayes accuracy ≈ 0.95, over the range $D \in [10, 700]$.

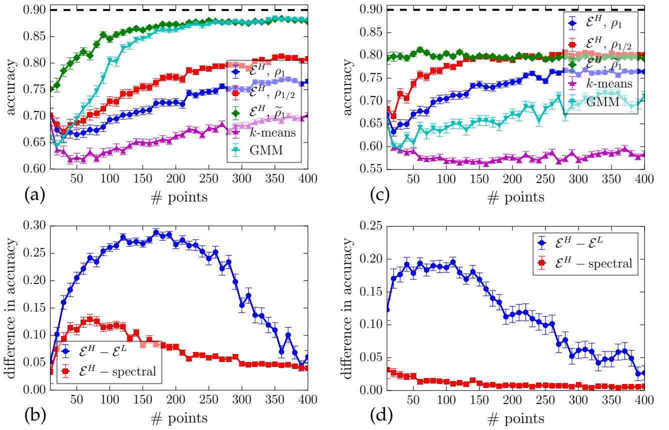

Fig. 4. $\mathcal{E}^H$-clustering with metrics (33) and (34) versus k-means and GMM. In both settings Bayes accuracy is ≈ 0.9. We show average accuracy (error bars are standard error) versus number of points for 100 Monte Carlo trials. (a,b) Gaussian mixture (38). (c,d) Lognormal mixture (39). The plots in (c) and (d) consider the difference in accuracy between $\mathcal{E}^H$ versus $\mathcal{E}^L$ (kernel k-means) and spectral clustering, with metric $\widetilde{\rho}_1$.

We simply chose a fixed set of 10 numbers uniformly at random on the interval $[1, 5]$ for the diagonal of $\widetilde{\Sigma}_{10}$, and any other choice would give analogous results. We show pair plots of this data in Fig. 2b. We sample a total of 200 points from (37) on each trial. The Bayes error is kept fixed when increasing D yielding an optimal accuracy ≈ 0.95. In Fig. 3b we see that GMM performs better in low dimensions, but it quickly degenerates as D increases. The same is true for k-means and $\mathcal{E}^L$-clustering. However, $\mathcal{E}^H$ and spectral clustering remains much more stable in high dimensions. Notice that a naive implementation of GMM should not be able to estimate the covariances when $D \gtrsim 100$, however, scikit-learn library uses k-means output as initialization, therefore the output of GMM in this implementation is at least as good as k-means and the algorithm is more robust in high dimensions.

Consider sampling data from the following Gaussian mixture in $\mathbb{R}^{20}$:

$$
\begin{aligned}
&x \overset{iid}{\sim} \tfrac{1}{2}\mathcal{N}(\mu_1, \Sigma_1) + \tfrac{1}{2}\mathcal{N}(\mu_2, \Sigma_2), \qquad 2\Sigma_1 = \Sigma_2 = I_{20} \\
&\mu_1 = (\underbrace{0, \dots, 0}_{\times 20})^\top, \quad \mu_2 = \tfrac{1}{2}(\underbrace{1, \dots, 1}_{5}, \underbrace{0, \dots, 0}_{15})^\top.
\end{aligned} \tag{38}
$$

The optimal accuracy based on Bayes classification error is ≈ 0.90. We increase the sample size $n \in [10, 400]$ and show the accuracy versus n for the different kernels (33) and (34) within $\mathcal{E}^H$-clustering algorithm, which are compared to k-means and GMM. The results are in Fig. 4a. Note that for small n all methods are superior than GMM, which slowly catches up and tend to optimal Bayes, as expected since it is a consistent model to the data. Note also that $\mathcal{E}^H$-clustering with kernel $\widetilde{K}_1$ is as accurate as GMM for large number of points, however, it is superior for small number of points. Still for the same setting, in Fig. 4b we show the difference in accuracy provided by $\mathcal{E}^H$ minus $\mathcal{E}^L$ and $\mathcal{E}^H$ minus spectral clustering, when using the kernel $\widetilde{K}_1$. Note that $\mathcal{E}^H$ was always superior than kernel k-means and spectral clustering, otherwise there would be points negative values on the y-axis.

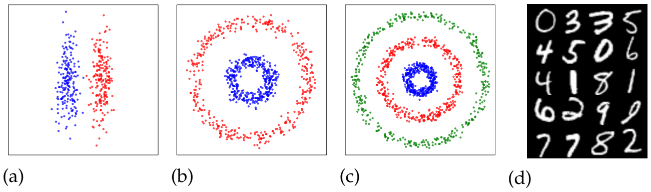

Fig. 5. (a) Parallel cigars. (b) Two concentric circles with noise. (c) Three concentric circles with noise. (d) MNIST handwritten digits. Clustering results are in Table 1 and Table 2.

Consider the same experiment but now with a lognormal mixture,

$$
\begin{aligned}
&x \overset{iid}{\sim} \tfrac{1}{2}e^{\mathcal{N}(\mu_1, \Sigma_1)} + \tfrac{1}{2}e^{\mathcal{N}(\mu_2, \Sigma_2)}, \quad 2\Sigma_1 = \Sigma_2 = I_{20}, \\
&\mu_1 = (\underbrace{0, \dots, 0}_{\times 20})^\top, \quad \mu_2 = \tfrac{1}{2}(\underbrace{1, \dots, 1}_{5}, \underbrace{0, \dots, 0}_{15})^\top.
\end{aligned} \tag{39}
$$

The results are in Fig. 4c. Energy clustering still performs accurately, with any of the utilized kernels, providing better results than k-means and GMM on this non-normal data. The kernel $\widetilde{K}_1$ still provides the best results for small number of points, but its performance is eventually achieved by $K_{1/2}$, indicating that $\alpha \approx 1/2$ in the standard energy distance should be more appropriate for skewed distributions. In Fig. 4d we show the difference between $\mathcal{E}^H$-clustering to kernel k-means and spectral clustering, with the kernel $\widetilde{K}_1$. Again, the accuracy provided by $\mathcal{E}^H$ is higher than the other methods, although not much higher than spectral clustering in this example. The two experiments of Fig. 4 illustrate how energy clustering is more flexible, performing well in different settings with the same kernel, contrary to k-means and GMM.

In Fig. 5a–c we have complex two dimensional datasets. The two parallel cigars in (a) have 200 points each. The concentric circles in (b) and (c) have 400 points for each class. We apply $\mathcal{E}^H$-clustering with the kernels (33), (34) and (35). We also consider the best kernel choice for each example for spectral clustering. Moreover, we consider k-means and GMM. We perform 10 Monte Carlo runs for each example. The results are in Table 1. For (a) we initialize all algorithms with k-means++, and for (b) and (c) we initialize at random. $\mathcal{E}^H$ has superior performance in every example, and in particular better than the spectral clustering. In (a) the standard kernel from energy statistics in Euclidean space, K_1 and $K_{1/2}$, are able to provide accurate results, however, for the examples in (b) and (c) the kernel choice is more sensitive, where $\widehat{K}_1$ and $\widehat{K}_2$ provide a significant improvement.

Next, we consider the infamous MNIST handwritten digits as illustrated in Fig. 5d. Each data point is an 8-bit gray scale image forming a 784-dimensional vector corresponding to the digits $\{0, 1, \dots, 9\}$. We compute the parameter

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

from a separate training set, to be used in the kernels (34) and (35). We consider subsets of $\{0, 1, \dots, 9\}$, sampling 100 points for each class. The results are shown in Table 2, where kernels and parameters are indicated. $\mathcal{E}^H$-clustering

TABLE 1
Clustering data from Fig. 5a–c.

		Fig. 5a		*Fig. 5b*		*Fig. 5c*
$\mathcal{E}^H$*-clustering*	ρ_1	0.705 ± 0.065	ρ_1	0.521 ± 0.005	ρ_1	0.393 ± 0.020
	$\rho_{1/2}$	0.952 ± 0.048	$\rho_{1/2}$	0.522 ± 0.004	$\rho_{1/2}$	0.486 ± 0.040
	$\widetilde{\rho}_2$	$\mathbf{0.9987 \pm 0.0008}$	$\widetilde{\rho}_1$	0.778 ± 0.075	$\widetilde{\rho}_2$	0.666 ± 0.007
	$\widehat{\rho}_2$	0.956 ± 0.020	$\widehat{\rho}_1$	$\mathbf{1.0 \pm 0.0}$	$\widehat{\rho}_2$	$\mathbf{0.676 \pm 0.002}$
spectral-clustering	$\widehat{\rho}_2$	0.557 ± 0.014	$\widehat{\rho}_1$	0.732 ± 0.002	$\widehat{\rho}_2$	0.364 ± 0.004
k-means	✗	0.550 ± 0.011	✗	0.522 ± 0.004	✗	0.368 ± 0.005
GMM	✗	0.903 ± 0.064	✗	0.595 ± 0.011	✗	0.465 ± 0.030

TABLE 2
Clustering MNIST data from Fig. 5d.

Class Subset		$\{0,1,\dots,4\}$	$\{0,1,\dots,6\}$	$\{0,1,\dots,8\}$	$\{0,1,\dots,9\}$
parameter	σ	10.41	10.41	10.37	10.19
$\mathcal{E}^H$*-clustering*	ρ_1	0.873 ± 0.025	0.731 ± 0.016	0.687 ± 0.016	0.581 ± 0.011
	$\rho_{1/2}$	0.874 ± 0.027	0.722 ± 0.017	0.647 ± 0.017	$\mathbf{0.600 \pm 0.009}$
	$\widetilde{\rho}_\sigma$	0.847 ± 0.031	0.695 ± 0.023	0.657 ± 0.014	0.584 ± 0.013
	$\widehat{\rho}_\sigma$	$\mathbf{0.891 \pm 0.009}$	$\mathbf{0.759 \pm 0.011}$	$\mathbf{0.704 \pm 0.011}$	$\mathbf{0.591 \pm 0.012}$
spectral-clustering	$\widehat{\rho}_\sigma$	0.769 ± 0.012	0.678 ± 0.014	0.649 ± 0.018	0.565 ± 0.009
k-means	✗	0.878 ± 0.010	0.744 ± 0.008	0.695 ± 0.012	0.557 ± 0.012
GMM	✗	0.839 ± 0.015	0.694 ± 0.010	0.621 ± 0.009	0.540 ± 0.009

performs slightly better than k-means and GMM, however the difference is not considerable. Unsupervised clustering on MNIST without any feature extraction is not trivial. For instance, the same experiment was performed in [29] where a low-rank transformation is learned then subsequently used in subspace clustering, providing very accurate results. It would be interesting to explore analogous methods for learning a better representation of the data and subsequently apply $\mathcal{E}^H$-clustering.

7.1 Real Data Experiment

We now consider a dataset that describes protein expression of neural synapses — chemical connections between neurons — which is crucial for understanding the diversity of synapses. This is part of a large NIH funded consortium within the BRAIN initiative. The details about this detaset, including experimental methods, are fully described in [30]. Here we have 1025 data points with 11 features, as described in Table 3. Each point corresponds to a (x, y, z) location in the brain. We normalize this dataset before applying $\mathcal{E}^H$-clustering.

We use the two metrics ρ_1 and $\widehat{\rho}_2$; see (33) and (35). We compute the difference between eigenvalues of the random walk Laplacian obtained from the kernel matrix G constructed from these metrics; see (12) and (6). The choice of k corresponds to the first time we see a meaningful peak in the plot, which occurs at $k = 6$ as illustrated in Fig. 6a. This procedure to find k is common in the literature [31].

Having found k, we now cluster the data. First we use the metric ρ_1, which is the standard Euclidean choice in energy statistics. In Fig. 6b and Fig. 6c we show a heat map of the cluster means versus features and the clustered data points projected into the 2 principal components, respectively. The plots of Fig. 6d and Fig. 6e show the same experiment but using the metric $\widehat{\rho}_2$ corresponding to a Gaussian kernel.

We remark that we also used the gap statistic [32] based on k-means to determine the number of clusters as a comparison. The results slightly suggest $k = 6$, but it does not pass the gap statistic test.

8 Discussion

We proposed clustering from the perspective of generalized energy statistics, valid for arbitrary spaces of negative type. Our mathematical formulation of energy clustering reduces to a QCQP in the associated RKHS, as demonstrated in Proposition 2. We showed that the optimization problem is equivalent to kernel k-means, once the kernel is fixed; see Proposition 3. Energy statistics, however, fixes a family of standard kernels in Euclidean space, and more general kernels on spaces of negative type can also be obtained. We also considered a weighted version of energy statistics, whose clustering formulation establishes connections with graph partitioning. We proposed the iterative $\mathcal{E}^H$-clustering algorithm based on Hartigan's method, which was compared to kernel k-means algorithm based on Lloyd's heuristic. Both have the same time complexity, however, numerical and theoretical results provide compelling evidence that $\mathcal{E}^H$-clustering is more robust with a superior performance, specially in high dimensions. Furthermore, energy clustering, with standard kernels from energy statistics, outperformed k-means and GMM on several settings, illustrating the flexibility of the proposed method which is model-free. In many settings, the iterative $\mathcal{E}^H$-clustering also surpassed spectral clustering, which is solves a relaxation of the original QCQP, and in other settings performed closely but never worse. Note that spectral clustering is more expensive than our iterative method, going up to $\mathcal{O}(n^3)$, and finding eigenvectors of very large matrices is problematic.

TABLE 3
Features of the synapse dataset according to [30].

x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}	x_{11}
bIIItubulin	DAPI-2nd	DAPI-3rd	GABA	GAD2	VGAT	gephyrin	NR1	VGluT1	synapsin	PSDr

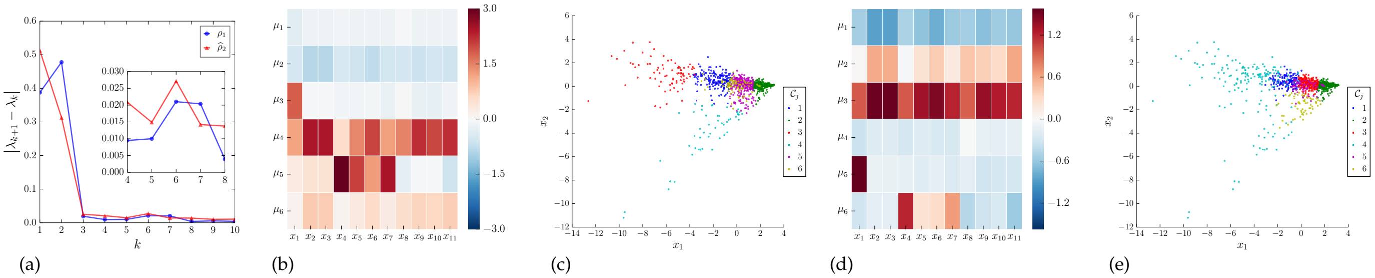

Fig. 6. Clustering synapse dataset. (a) Eigenvalues of random walk Laplacian. We choose $k = 6$ since its the first time we see a peak. We use energy clustering with the two metrics ρ_1 and $\widehat{\rho}_2$. (b) Heat map of the cluster means versus features using energy clustering with standard metric ρ_1 from energy statistics. (c) Clustered points projected into the two principal components, using ρ_1. (d) Heat map of the cluster means versus features using energy clustering with Gaussian metric $\widehat{\rho}_2$. (e) Clustered points projected into the two principal components, using $\widehat{\rho}_2$.

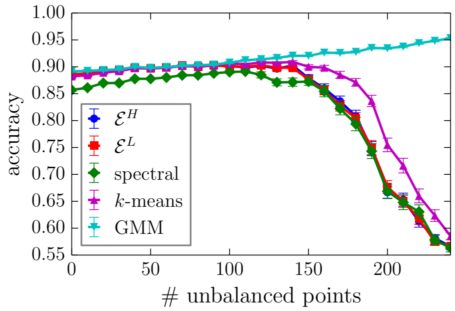

Fig. 7. Comparison of energy clustering algorithms to k-means and GMM on unbalanced clusters. The data is normally distributed as (40), where we vary $m \in [0, 240]$, and in each case we do 100 Monte Carlo runs showing the average accuracy with standard error.

A limitation of the proposed methods for energy clustering is that it cannot handle accurately highly unbalanced clusters. As an illustration, consider the following Gaussian mixture:

$$
\begin{aligned}
& x \overset{iid}{\sim} \frac{n_1}{2N}\mathcal{N}(\mu_1, \Sigma_1) + \frac{n_1}{2N}\mathcal{N}(\mu_1, \Sigma_1), \\
& \mu_1 = (0,0,0,0)^\top, \quad \mu_2 = 1.5 \times (1,1,0,0)^\top, \\
& \Sigma_1 = I_4, \quad \Sigma_2 = \left(\begin{array}{c|c} \frac{1}{2}I_2 & 0 \\ \hline 0 & I_2 \end{array} \right), \\
& n_1 = N - m, \quad n_2 = N + m, \quad N = 300.
\end{aligned} \tag{40}
$$

We then increase $m \in [0, 240]$ making the clusters progressively more unbalanced. We plot the average accuracy over 100 Monte Carlo runs for each m, with error bars indicating standard error. The results are shown in Fig. 7. For highly unbalanced clusters, we see that GMM performs better than the other methods, which have basically similar performance. Based on this experiment, an interesting problem would be to extend $\mathcal{E}^H$-clustering algorithm to account for highly unbalanced clusters.

Moreover, it would be interesting to formally demonstrate cases where energy clustering is a consistent in the large n limit. A soft version of energy clustering is also an interesting extension. Finally, kernel methods can benefit from sparsity and fixed-rank approximations of the Gram matrix, and there is plenty of room to make $\mathcal{E}^H$-clustering algorithm more scalable.

Acknowledgments

We would like to thank Carey Priebe for discussions. We would like to acknowledge the support of the Transformative Research Award (NIH #R01NS092474) and the Defense Advanced Research Projects Agencys (DARPA) SIMPLEX program through SPAWAR contract N66001-15-C-4041.

References

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

[23] 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.

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

[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] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[28] 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.

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

[30] F. Collman, J. Buchanan, K. D. Phend, K. D. Micheva, R. J. Weinberg, and S. J. Smith. "mapping synapses by conjugate light-electron array tomography". *The Journal of Neuroscience*, 34(14):5792–5807, 2015.

[31] U. von Luxburg. "a tutorial on spectral clustering". *Statistics and Computing*, 17(4):395–416, 2007.

[32] R. Tibshirani, G. Walther, and T. Hastie. "estimating the number of clusters in a data set via the gap statistic". *J. Royal Statistical Society B*, 63(2):411–423, 2001.

PLACE PHOTO HERE

Guilherme França Biography text here.

Maria Rizzo Biography text here.

Joshua Vogelstein Biography text here.