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 Newton's potential energy between massive bodies. The idea is to associate a statistical potential energy to observations such that minimum energy is achieved under the null hypothesis of equality of distributions. This was further generalized for probability distributions on arbitrary metric spaces, and more recently, a connection with kernels in RKHS was established. Nevertheless, although extensively used by the statistics community, it was not incorporated in machine learning problems. In this paper, we consider the problem of clustering data based energy statistics theory. We provide a precise mathematical formulation, obtaining a quadratically constrained optimization problem (QCQP). We show that this optimization problem is equivalent to kernel k-means optimization problem, however, energy statistics is able to fix the kernel choice. This method is nonparametric, and if prior information is available it can be easily incorporated in the kernel construction. We propose an algorithm to find local optimizers of this QCQP problem based on the energy cost of moving points to different clusters. We then compare this algorithm with kernel k-means, standard k-means, and GMM. We test our method on synthetic and real data experiments. Our results show that energy statistics based clustering outperforms these most used clustering algorithms.

* guifranca@gmail.com

† jovo@jhu.edu

I. INTRODUCTION

Mention why energy is important, main results, where it was applied, etc. Motivate how this can be used for clustering. Mention most important papers on this ... Explain main results of this paper and give a brief outline.

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 references therein) and also [2].

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 $\mathcal{E}_\alpha(P,Q) = 0$ 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 a space endowed with a *semimetric of negative type* $\rho : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$, which satisfies

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

where $X_i \in \mathcal{X}$ and $\alpha_i \in \mathbb{R}$ such that $\sum_{i=1}^n \alpha_i = 0$. 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

energy distance as

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

For spaces of negative type exists a Hilbert space $\mathcal{H}$ and a map $\varphi : \mathcal{X} \to \mathcal{H}$ such that $\rho(X,Y) = \|\varphi(X) - \varphi(Y)\|^2_{\mathcal{H}}$. 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 [2]. We 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 semi-definite, $\boldsymbol{w}^\top G \boldsymbol{w} \geq 0$ for any vector $\boldsymbol{w} \in \mathbb{R}^n$.

The Moore-Aronszajn theorem establishes the converse [3]. 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: $P \mapsto h_P \in \mathcal{H}_K$ such that $\int f(x) dP(x) = \langle f, h_P\rangle$ for all $f \in \mathcal{H}_K$, or alternatively, $h_P = \int K(\,\cdot\,, x) dP(x)$. 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 [4]

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

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

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

Therefore, in practice, we can estimate the inner product between the 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 [5]. Let $\rho : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$, and $x_0 \in \mathcal{X}$ an arbitrary but fixed point. Define

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

Thus, K is positive semidefinite 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

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

and the canonical feature map $\varphi : x \mapsto h_x$ is injective [2]. 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 [2]. 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 $\gamma_k^2(P,Q) = \|h_P - h_Q\|^2_{\mathcal{H}_K}$ 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 statistics of equality of distributions [1]. Assume we have data $\mathbb{X} = \{x_1, \ldots, x_n\}$, where $x_i \in \mathcal{X}$ and $\mathcal{X}$ is a space of negative type. Consider a 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

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

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. Next we show how we can use S for clustering data.

III. CLUSTERING BASED ON ENERGY STATISTICS

This section contains the main 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 intuitively 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 the points are correctly classified since S measures a cost between different clusters. However, the following straightforward result shows that maximizing (13) is equivalent to minimizing (12), which has a more convenient form.

Proposition 1. *Let $\mathbb{X} = \{x_1, \ldots, x_n\}$ 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*

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

where W is given by (12)*.*

Proof. It can be shown that the total dispersion of the data obeys [1]

$$W + S = \frac{n}{2} g(\mathbb{X}, \mathbb{X}). \tag{15}$$

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 is equivalent to minimizing W. □

Thus, 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.

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 $D = \operatorname{diag}(n_1, \ldots, n_k)$ 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

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

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 problem (QCQP).

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

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

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

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

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

Note that the first term 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

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

But

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

where we used the definitions (16) and (17). Thus the objective function in (20) is equal to $\operatorname{Tr}\left(D^{-1}Z^\top GZ\right)$. 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:

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

Now we write this in terms of the matrix $Y=ZD^{-1/2}$. The objective function immediately becomes $\operatorname{Tr}\left(Y^\top GY\right)$. Notice that the above constraints imply that $Z^TZ=D$, where $D=\operatorname{diag}(n_1,\ldots,n_k)$, which in turn gives $D^{-1/2}Y^TYD^{-1/2}=D$, or $Y^\top Y=I$. Also, every entry of Y is positive by definition, $Y\geq 0$. Now it only remains to show the last constraint in (18), which comes from the last constraint in (22). In matrix form this reads $Z^T\boldsymbol{e}=D\boldsymbol{e}$. 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 $YY^\top\boldsymbol{e}=\boldsymbol{e}$. Thus, the optimization problem (22) is equivalent to (18) . □

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

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

has a well-known closed form solution given by $Y^\star = UR$, where the columns of U contain the leading k eigenvectors of G corresponding to the k largest eigenvalues $\{\lambda_1, \ldots, \lambda_k\}$, and $R \in \mathbb{R}^{k\times k}$ is an arbitrary orthogonal matrix. The resulting optimal objective function is thus given by $\max \operatorname{Tr}\left(Y^{\star\top} G\, Y^\star\right) = \sum_{i=1}^k \lambda_i$. One might then normalize and threshold the rows of $Y^\star$, or even better, apply standard k-means to the rows of Y to provide the final clusters. This procedure is known as spectral clustering. Nonetheless, computing eigenvectors of a very large matrix can be problematic, and usually iterative methods are preferred.

It is important to note that clustering based on energy statistics holds 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, 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. Thus, this method is quite general and makes very few assumptions about the data.

Although (18) is nonparametric, in practice, 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, or equivalently K, this can easily be taken into account.

Now let us consider the relation to the well-known kernel k-means optimization problem. For 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 problem, in the feature space, is given by

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

where $\mu_j = \frac{1}{n_j} \sum_{x\in\mathcal{C}_j} x$ is the mean of cluster $\mathcal{C}_j$. It is well-known [6] that problem (24) is equivalent to the QCQP (18). The next result makes this explicit, showing its relation with the energy statistics based clustering considered thus far.

Proposition 3. *The clustering problem* (14) *based on energy statistics is equivalent to the kernel k-means 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

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

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

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

which is exactly the same as (20). The remaining of the proof proceeds as already shown in the proof of Proposition 2. □

It was shown [6] that 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). Actually, in general, this corresponds to a weighted version of (18) which reads $\operatorname{Tr}\left(Y^\top W^{1/2}G\,W^{1/2}Y\right)$, where $W=\operatorname{diag}(w(x_1),\ldots,w(x_n))$ and $w(\cdot)$ is a weight attributed to each data point. Our previous results show that energy statistics based clustering is also equivalent to these problems. The advantage of energy statistics is that the semimetric ρ fixes the kernel through (8).

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 also be useful later for comparison with the more general iterative algorithm.

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

$$\begin{aligned}|x-y| &= \mathbb{1}_{x\geq y}(x-y)-\mathbb{1}_{x<y}(x-y)\\ &= \left(\mathbb{1}_{x\geq y}-\mathbb{1}_{x<y}\right)x+\left(\mathbb{1}_{y>x}-\mathbb{1}_{y\leq x}\right)y,\end{aligned} \tag{27}$$

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

$$g\left(\mathcal{C},\mathcal{C}\right)=\frac{1}{n^2}\sum_{x\in\mathcal{C}}\sum_{y\in\mathcal{C}}\left(\mathbb{1}_{x\geq y}+\mathbb{1}_{y>x}-\mathbb{1}_{x\geq y}-\mathbb{1}_{x<y}\right)x. \tag{28}$$

The sum over y can be eliminated since the term in parenthesis is simply counting the number of elements in $\mathcal{C}$ that satisfy the conditions of the indicator functions. Assuming

that we first order the data in the partition, obtaining $\bar{\mathcal{C}} = [x_j \in \mathcal{C} : x_1 \leq x_2 \leq \cdots \leq x_n]$, we can write (28) in the following simple form:

$$g\left(\bar{\mathcal{C}}, \bar{\mathcal{C}}\right) = \frac{2}{n^2} \sum_{\ell=1}^{n} (2\ell - 1 - n) x_\ell. \tag{29}$$

Note that the cost of computing this 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

$$W\left(\bar{\mathcal{C}}_1, \ldots, \bar{\mathcal{C}}_k\right) = \sum_{j=1}^{k} \sum_{\ell=1}^{n_j} \frac{2\ell - 1 - n_j}{n_j} x_\ell. \tag{30}$$

For a two-class problem we can use (30) to cluster data through a simple algorithm as follows. We first order the entire dataset $\mathbb{X} \to \bar{\mathbb{X}}$. Then we compute (30) 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 below. 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 $\mathcal{O}(n \log n + n^2) = \mathcal{O}(n^2)$.

Algorithm 1 Procedure to find an approximate solution to (18) for a two-class problem in one dimension and with Euclidean distance.

input Data $\mathbb{X}$

output Label matrix Z

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

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

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

4: $W^{(j)} \leftarrow W\left(\bar{\mathcal{C}}_1^{(j)}, \bar{\mathcal{C}}_2^{(j)}\right)$ from (30)

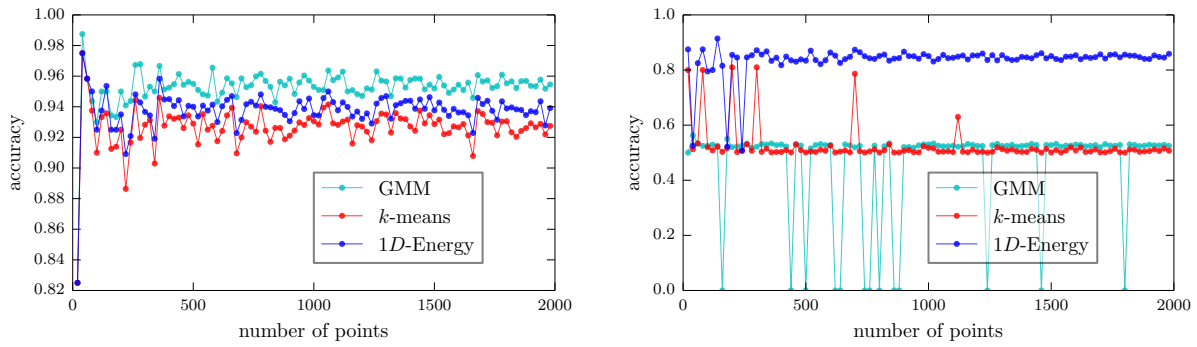
5: **end for**

6: $j^\star \leftarrow \arg\min_j W^{(j)}$ determines the best split

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

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

$$\text{accuracy}(\hat{Z}) = \max_{\sigma} \frac{1}{n} \sum_{i=1}^{n} \sum_{j=1}^{k} \hat{Z}_{i\sigma(j)} Z_{ij} \tag{31}$$



(a) (b)

FIG. 1. We cluster data using Algorithm 1 ($1D$-Energy in the plots), GMM, and k-means. We use (31) to evaluate cluster quality. Both clusters have the same number of points, which are increased in each experiment. (a) $x \sim \frac{1}{2}\left(\mathcal{N}(\mu_1, \sigma_1) + \mathcal{N}(\mu_2, \sigma_2)\right)$ with $\mu_1 = 0$, $\mu_2 = 5$, $\sigma_1 = 1$, and $\sigma_2 = 2$. (b) $x \sim \frac{1}{2}\left(e^{\mathcal{N}(\mu_1,\sigma_1)} + e^{\mathcal{N}(\mu_2,\sigma_2)}\right)$ with $\mu_1 = 0$, $\mu_2 = -1.5$, $\sigma_1 = 0.3$, and $\sigma_2 = 1.5$.

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.

Before proposing a more general iterative algorithm to (18), 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 k-means. In both of these cases we use the initialization from k-means++ and we run the algorithms a couple of times with different initializations and choose the answer with best objective function value. We use (31) to measure the clustering quality. The results are shown in Fig. 1. 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 right model for the data. However, as shown in Fig 1b, for lognormal distributions Algorithm 1 provides a huge improvement compared to both GMM and k-means, which basically cluster at chance. The zero accuracy values for GMM happened when EM algorithm was unable to estimate the parameters. These two simple experiments illustrate how energy statistics based clustering is nonparametric.

V. ITERATIVE ALGORITHM FOR ENERGY CLUSTERING

In this section we will introduce a new iterative algorithm to find a local maximizer of (18), however, due to Proposition 3 we can also find an approximate solution by the well-known kernel k-means algorithm. Thus, to be self-contained let us first recall kernel k-means algorithm in this context.

A. Kernel k-Means Algorithm

Consider this optimization problem as written in the form (20),

$$\max_{\{\mathcal{C}_1,\ldots,\mathcal{C}_k\}} \left\{ Q = \sum_{j=1}^{k} \frac{Q_j}{n_j} \right\}, \qquad Q_j = \sum_{x,y\in\mathcal{C}_j} K(x,y), \tag{32}$$

where Q_j represents the internal cost of cluster $\mathcal{C}_j$, and Q is the total cost where each cluster cost is weighted by the inverse of the number of its elements. For a data point x_i its cost with cluster $\mathcal{C}_j$ is denoted by

$$Q_j(x_i) \equiv \sum_{y\in\mathcal{C}_j} K(x_i,y) = G_{i\bullet}\cdot Z_{\bullet j}. \tag{33}$$

Now for kernel k-means consider (25) where we define the function

$$J^{(\ell)}(x_i) \equiv -\frac{2}{n_\ell} Q_\ell(x_i) + \frac{1}{n_\ell^2} Q_\ell \tag{34}$$

which represents the cost of x_i with cluster $\mathcal{C}_\ell$. Thus, one assigns point x_i to the cluster $\mathcal{C}_{j^\star}$ according to $j^\star = \arg\min_\ell J^{(\ell)}(x_i)$, for $\ell = 1,\ldots,k$. This procedure is performed for every data point, and repeated until convergence, i.e. until no new assignments are made. The complete algorithm is shown in Algorithm 2. Although our formulation looks a little bit different than the standard kernel k-means found in the literature [6], this is precisely the same algorithm.

Notice that to compute the first term in (34) 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 the loop through data points. Therefore, the time complexity of this algorithm is $\mathcal{O}(nk\max_\ell n_\ell) = \mathcal{O}(kn^2)$. If G is sparse with n' nonzero elements, this complexity can be further reduced to $\mathcal{O}(kn')$.

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

Let $\boldsymbol{q} = (Q_1, \ldots, Q_k)^\top$ have the costs of each cluster, according to (32)
Let $\boldsymbol{n} = (n_1, \ldots, n_k)^\top$ have the number of elements in each cluster
repeat
  for $i = 1, \ldots, n$ **do**
    Let j be such that $x_i \in \mathcal{C}_j$
    $j^\star \leftarrow \arg\min_\ell J^{(\ell)}(x_i)$ according to (34), for $\ell = 1, 2, \ldots, k$
    if $j^\star \neq j$ **then**
      Move x_i to $\mathcal{C}_{j^\star}$: $Z_{ij} \leftarrow 0$, $Z_{ij^\star} \leftarrow 1$
      Update $\boldsymbol{n}$: $n_j \leftarrow n_j - 1$, $n_{j^\star} \leftarrow n_{j^\star} + 1$
      Update $\boldsymbol{q}$: $q_j \leftarrow q_j - 2Q_j(x_i)$, $q_{j^\star} \leftarrow q_{j^\star} + 2Q_{j^\star}(x_i)$
    end if
  end for
until convergence

B. Energy Cost Algorithm

Now let us consider a different algorithm, which is based on computing the difference in the energy cost when assigning a given data point to a different partition. Suppose we have a data point $x_i \in \mathcal{X}$ which is currently classified as being in cluster $\mathcal{C}_j$, yielding a total cost function (32) denoted by $Q^{(j)}$. Let us consider the change in the total cost by moving x_i to cluster $\mathcal{C}_\ell$. Denote the new total cost after moving x_i to $\mathcal{C}_\ell$ by $Q^{(\ell)}$. It is straightforward to see that

$$\begin{aligned} \Delta Q^{j\to\ell}(x_i) &\equiv Q^{(\ell)} - Q^{(j)} \\ &= \frac{1}{n_j - 1}\left[\frac{Q_j}{n_j} - 2Q_j(x_i)\right] - \frac{1}{n_\ell + 1}\left[\frac{Q_\ell}{n_\ell} - 2\big(Q_\ell(x_i) + K(x_i, x_i)\big)\right]. \end{aligned} \tag{35}$$

Thus, if $\Delta Q^{j\to\ell}(x_i) > 0$ we get closer to a maximum of (32) by moving x_i to $\mathcal{C}_\ell$, otherwise we 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 procedure is described in Algorithm 3 below.

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

1: Let $\boldsymbol{q} = (Q_1, \dots, Q_k)^\top$ have the energy costs of each cluster, according to (32)
2: Let $\boldsymbol{n} = (n_1, \dots, n_k)^\top$ have the number of elements in each cluster
3: **repeat**
4: **for** $i = 1, \dots, n$ **do**
5: Let j be such that $x_i \in \mathcal{C}_j$
6: $j^\star \leftarrow \arg\max_\ell \Delta Q^{j\to\ell}(x_i)$, for $\ell = 1, 2, \dots, k$ and $\ell \neq j$
7: **if** $\Delta Q^{j\to j^\star}(x_i) > 0$ **then**
8: Move x_i to $\mathcal{C}_{j^\star}$: $Z_{ij} \leftarrow 0$, $Z_{ij^\star} \leftarrow 1$
9: Update $\boldsymbol{n}$: $n_j \leftarrow n_j - 1$, $n_{j^\star} \leftarrow n_{j^\star} + 1$
10: Update $\boldsymbol{q}$: $q_j \leftarrow q_j - 2Q_j(x_i)$, $q_{j^\star} \leftarrow q_{j^\star} + 2\left(Q_{j^\star}(x_i) + G_{ii}\right)$
11: **end if**
12: **end for**
13: **until** convergence

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 (33) 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. In the numerical experiments below we choose the initial Z from the initialization procedure of k-means++ [7].

VI. NUMERICAL EXPERIMENTS

In the following experiments we fix the semimetric as $\rho(x, y) = \|x - y\|$, with induced kernel (8). We will compare Algorithm 3 with kernel k-means Algorithm 2, and also with standard k-means and GMM through the Expectation Maximization (EM) algorithm as well. For synthetic data our measure of clustering performance will be (31). Moreover, for all the algorithms, we always choose the initialization from k-means++ [7].

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. In Figure 2a we have data generated from normal distributions in D-dimensions:

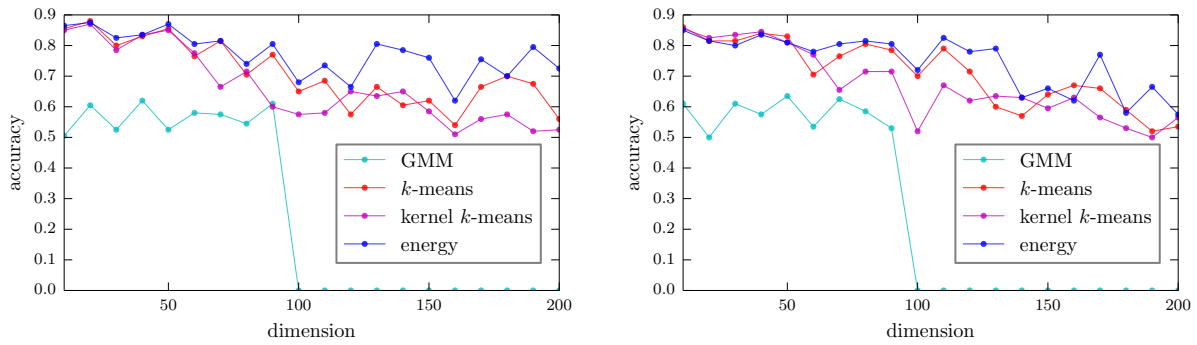
$$\begin{aligned} &x \sim \tfrac{1}{2}\left\{\mathcal{N}(\mu_1, \Sigma_1) + \mathcal{N}(\mu_2, \Sigma_2)\right\}, \\ &\mu_1 = (\underbrace{0, \ldots, 0}_{\times D})^\top, \quad \mu_2 = 0.7 \times (\underbrace{1, \ldots, 1}_{\times 10}, \underbrace{0, \ldots, 0}_{\times (D-10)})^\top, \quad \Sigma_1 =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

For each experiment we only keep signal in μ_2 in the first 10 dimensions, and keep increasing the ambient dimension D. For each data set, we run the algorithms 10 times and pick the result with the best objective function value. In Figure 2a one sees that GMM is not able to estimate the covariance matrix when the number of dimensions exceeds the number of points in each cluster. One can see a slightly better performance of Algorithm 3 compared to the other ones. In Figure 2b we have the same type of experiment but with

$$\begin{aligned} &x \sim \tfrac{1}{2}\left(\mathcal{N}(\mu_1, \Sigma_1) + \mathcal{N}(\mu_2, \Sigma_2)\right), \\ &\mu_1 = (\underbrace{0, \ldots, 0}_{\times D})^\top, \, \mu_2 = 0.7 \times (\underbrace{1, \ldots, 1}_{\times 10}, \underbrace{0, \ldots, 0}_{\times (D-10)})^\top, \, \Sigma_1 = I_D, \, \Sigma_2 = \begin{pmatrix} \frac{1}{2} I_{10} & 0 \\ 0 & I_{D-10} \end{pmatrix}. \end{aligned} \tag{37}$$

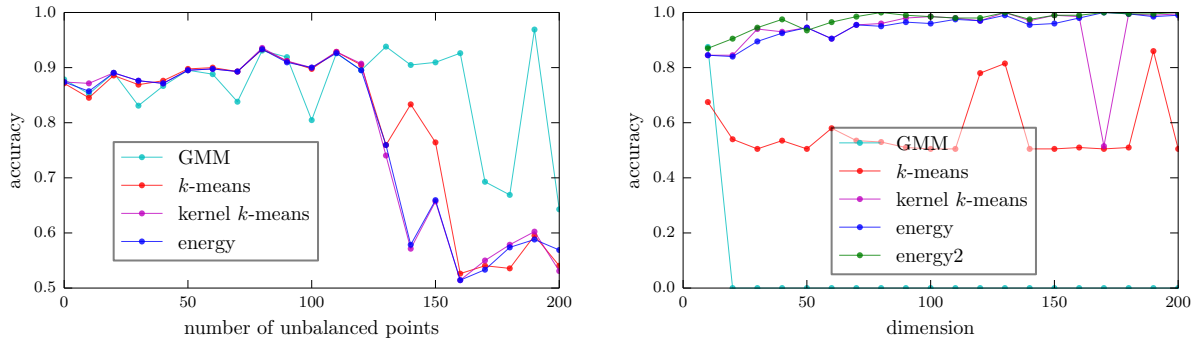
In Figure 3a we consider how the previous algorithms behave for unbalanced clusters. We generate data as

$$\begin{aligned} &x \sim \frac{n_1}{N}\mathcal{N}(\mu_1, \Sigma_1) + \frac{n_2}{N}\mathcal{N}(\mu_2, \Sigma_2), \quad \mu_1 = (0,0,0,0)^\top, \, \mu_2 = 1.5 \times (1,1,0,0)^\top, \\ &\Sigma_1 = I_4, \quad \Sigma_2 = \begin{pmatrix} 1/2 & 0 & 0 & 0 \\ 0 & 1/2 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}, \quad n_1 = N - m, \quad n_2 = N + m, \quad N = 200, \end{aligned} \tag{38}$$



(a) (b)

FIG. 2. Experiments where data is normally distributed. (a) We keep 100 points on each cluster and increase the ambient dimension, as described in (36). The blue dots correspond to Algorithm 3, while the red dots corresponds to Algorithm 2. (b) The same but with data following (37). In both experiments one notice a slightly better accuracy of Algorithm 3 compared to the other ones.



(a) (b)

FIG. 3. (a) Previous algorithms for unbalanced clusters, according to (38). (b) The same experiment as in Figure 2a but for lognormal data as $x \sim \frac{1}{2}\left(e^{\mathcal{N}(\mu_1,\Sigma_1)} + e^{\mathcal{N}(\mu_2,\Sigma_2)}\right)$ with $\mu_1 = (0,\ldots,0)^\top$, $\mu_2 = 0.5 \times (1,\ldots,1,0,\ldots,0)^\top$ with signal in $d = 10$, $\Sigma_1 = 0.3 \times I_D$, and $\Sigma_2 = I_D$. One see a clear advantage of energy statistics clustering. The extra green line correspond to $\rho(x,y) = \|x-y\|^{1/2}$.

and we increase m, i.e. we make the clusters progressively more unbalanced. As well-known, one an see that GMM works well for unbalanced clusters, mostly because it is a soft clustering algorithm. We can see that all the other kernel based methods degrade more rapidly than GMM for highly unbalanced clusters. This is unsurprising since Algorithms 3 and 2 both make hard assignments. Finally, Figure 3b considers the same experiment as in Figure 2a but for lognormal data. With $\rho(x,y) = \|x-y\|^{1/2}$ we even gain a slight improvement. Energy statistics clustering outperforms all the other methods since it is nonparametric.

In Figure 4 we consider other types of data in two dimensions. In each case we perform 10 experiments and compute the accuracy based on the true labels. The plots show the average results of 10 runs of each experiment (dots), and the maximum and minimum values as well (shaded area). Although energy clustering is nonparametric, these datasets are very specific so we can take advantage and make a suitable choice of the semimetric ρ. This illustrates that energy clustering is also flexible enough to incorporate prior information about the data. In Figure 4a we have parallel cigars from

$$\begin{aligned} &x \sim \tfrac{1}{2}\left(\mathcal{N}(\mu_1, \Sigma_1) + \mathcal{N}(\mu_2, \Sigma_2)\right), \\ &\mu_1 = (0,0)^\top, \quad \mu_2 = (6.5, 0)^\top, \quad \Sigma_1 = \Sigma_2 = \left(\begin{smallmatrix} 1 & 0 \\ 0 & 20 \end{smallmatrix}\right) \end{aligned} \tag{39}$$

with 100 points in each cluster. Notice that k-means is unable to cluster this data, while energy clustering performs pretty much as well as GMM, which is the optimal algorithm suited to this kind of data. We used the semimetric $\rho(x, y) = \|x - y\|^{1/2}$ in this case. In Figure 4b we generate two concentric circles with a small gaussian noise. To make the semimetric a little bit more localized we include a decaying exponential in the form:

$$\rho(x, y) = \|x - y\|^2 e^{-\frac{1}{2}\|x-y\|}. \tag{40}$$

Notice that energy clustering is able to cluster this data almost perfectly in most cases. In Figure 4c we consider two spirals, also with a small gaussian noise. Due to the geometry of the data we consider

$$\rho(x, y) = \|x - y\|^2 \theta \sin\left(\tfrac{\|x-y\|}{\theta}\right), \qquad \theta = 0.9. \tag{41}$$

Again, energy clustering performs much better than k-means and GMM which are unable to cluster this kind of dataset.

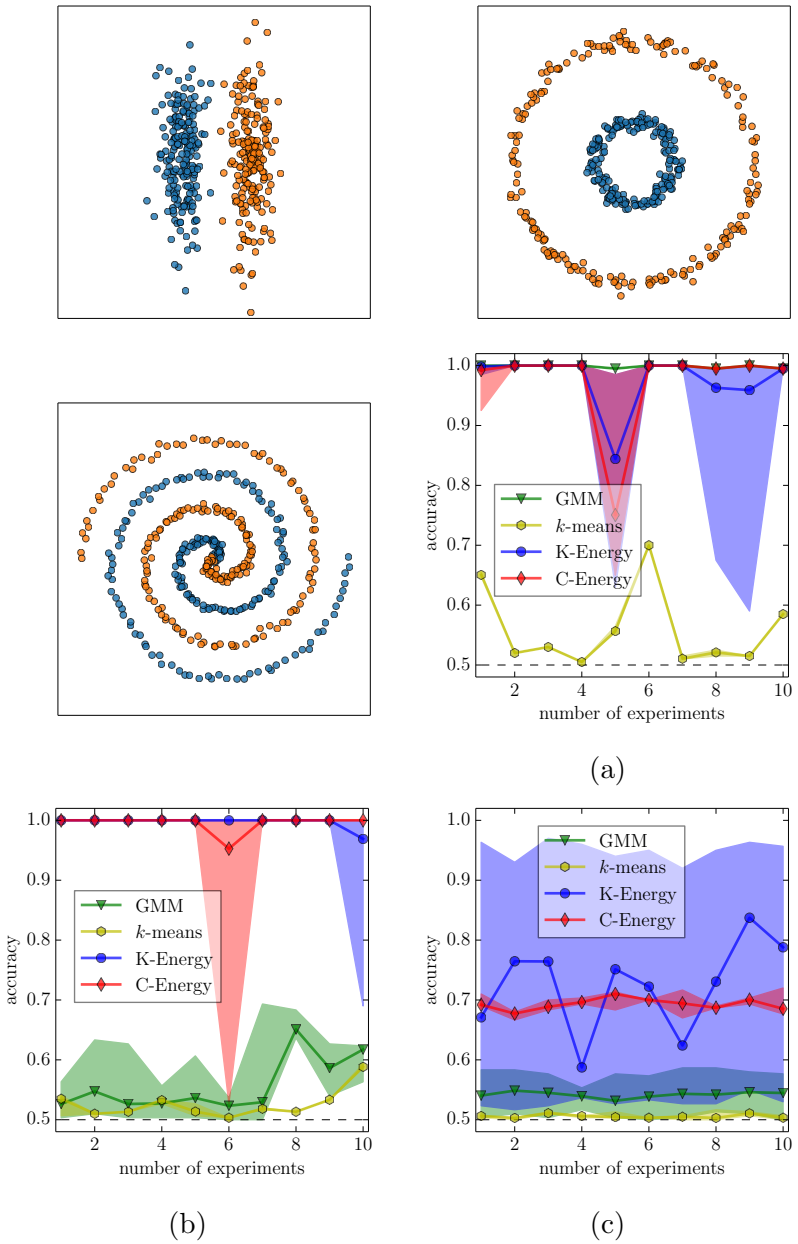
VII. CONCLUSION

Acknowledgements

We thank ...

---

[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.



(a)

(b)

(c)

FIG. 4. (a) Parallel cigars generated from (39) with 100 points in each class. Use use energy clustering with $\rho = \|x - y\|^{1/2}$. (b) Two concentric circles with 150 points in each class. We use the semimetric (40). (c) Two concentric spirals with 150 points in each class. We use the semimetric (41).

[2] 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.

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

[4] 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.

[5] 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.

[6] 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.

[7] 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.