# Learning with Jensen-Tsallis Kernels

Debarghya Ghoshdastidar, Ajay P. Adsul and Ambedkar Dukkipati

***Abstract*—Jensen-type (Jensen-Shannon and Jensen-Tsallis) kernels were first proposed by Martins et al. (2009). These kernels are based on Jensen-Shannon divergences that originated in information theory. In this paper, we extend the Jensen-type kernels on probability measures to define positive definite kernels on Euclidean space. We show that special cases of these kernels include dot-product kernels. Since Jensen-type divergences are multi-distribution divergences, we propose their multi-point variants, and study spectral clustering and kernel methods based on these. We also provide experimental studies on benchmark image database and gene expression database that show the benefits of the proposed kernels compared to existing kernels. The experiments on clustering also demonstrate the use of constructing multi-point similarities.**

***Index Terms*—Kernels, multi-distribution divergence, multi-point similarity, segmentation.**

## I. Introduction

Information theoretic divergences have been often employed as distance measures in the context of learning [2], [3]. Such distance measures tend to be a natural choice when the solution is computed in the probability simplex. To this end, Csiszár's $f$-divergences [4] and the Jensen-type divergences [5], [6], are quite special. This is primarily because these divergences are multi-distribution divergences, and hence, provide a measure of dissimilarity among more than two probability distributions. However, when one computes distances in the real domain, the obvious choice turns out to be the Bregman divergences [7] that generalizes the standard Euclidean distance along with other distance metrics. To this end, machine learning methods make little use of the fact that the square-root of the Jensen-Shannon (JS) divergence is a Hilbertian metric [8].

On the other hand, there is a completely new generalization of divergences that arose due to the introduction of Tsallis entropy [9] in physics. This entropy involves a parameter $q$ and as $q \to 1$ one can retrieve Shannon entropy. Suyari [10] generalized the Shannon-Khinchin axioms to this case, while Dukkipati et al. [11] provide a measure theoretic formulation of continuous form Tsallis entropy functional. Jensen-Shannon (JS) divergence in its generalized form are called Jensen-Tsallis divergence [12] and Jensen-Tsallis $q$-difference [13]. In this paper, we refer to all the above divergences as Jensen-type divergences.

The recent kernel connections of the square-root of JS-divergence [8] encouraged the machine learning community to view Jensen-type divergences as dissimilarity measures, and subsequently, the works in [14], [15] proposed new kernels on probability measures based on the JS-divergence. Studies by Martins et al. [13] extend the idea further to the Tsallis case to formulate the so-called Jensen-Tsallis (JT) kernel on the space of finite measures that has proved to be quite useful in text classification [13] and shape recognition [16].

Though the significance of JT-kernel has been well established in [13], [16], one still finds a lack of study in two directions.

- The JT-kernel retrieves the linear kernel ($x^T y$) in a special case, and hence, this kernel may have interesting properties even on the Euclidean space.
- The multi-distribution nature of the Jensen-type divergences provides an opportunity to construct multi-point variants of JT-kernel.

The notion of multi-point similarities has been often used in recent times for several vision tasks, such as face clustering [17], motion segmentation [18], [19] and image registration [20]. These approaches rely on a certain decomposition of higher-order tensors [21] by means of tensor flattening. However, till date, the use of multi-point similarities have been restricted to specific areas of computer vision as the multi-way similarities are constructed from certain geometric models. In this work, we broaden the use of multi-point similarities by presenting multi-point generalization of some positive definite kernels. We also present a general technique for constructing positive definite kernels from these multi-point similarities. Thus, we are able to extend the use of multi-point similarities to the widely varying applications addressed by kernel methods [22], [23] and spectral clustering [24], [25], that range from image processing [24] to the analysis of gene expressions [26].

### *Contributions in this paper*

The contributions in this paper are listed below:

1) We extend the JT-kernel on finite measures and its exponential variant to define similar kernels on the $d$-dimensional unit cube $[0, 1]^d$ that encompass the linear (dot-product) kernel. Further, we use the idea of multi-distribution divergences to define multi-point extensions of above kernels.
2) We develop a technique for constructing positive definite kernels from multi-point similarities based on tensor flattening approach used in tensor singular value decompositions [21].
3) As common in tensor based methods, the above kernel computation has high computational complexity. Hence, we discuss approximate methods for kernel computation, and also characterize special cases where the complexity can be significantly reduced.

This work is supported by Department of Science and Technology (DST:SB/S3/EECE/093/2014).

Part of this paper has been presented at IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2014 [1].

D. Ghoshdastidar, A. P. Adsul and A. Dukkipati are with the Department of Computer Science & Automation, Indian Institute of Science, Bangalore – 560012, India (e-mail: {debarghya.g,ajay.adsul,ad}@csa.iisc.ernet.in).

4) We study the performance of the proposed kernels in the context of classification and clustering.

*Organization of this paper*

In Section II, we briefly review kernels and in particular, the class of Jensen-Tsallis kernels. We study the JT-kernel and its exponential variant on Euclidean space in Section III, and then propose the multi-point variants of the same in Section IV. Section V presents a multi-point generalization of the linear kernel that is derived from the multi-point JT-kernel. Section VI provides an experimental evaluation of the proposed kernels with existing kernels. This comparison is performed using standard UCI datasets (Section VI-A), gene expression data (Section VI-B) and image segmentation database (Section VI-C). Finally, we provide concluding remarks in Section VII. Proofs of the theoretical results are given in the Appendix.

## II. Preliminaries and Background

### A. Kernels and similarities

One of the fundamental problems in machine learning is to obtain a map between an input space $\mathcal{X}$ and an output space $\mathcal{Y}$. The objective varies depending on the nature of the problem. In linear methods of learning, the Euclidean distance between data points is used to distinguish them. In other words, the dot product between two vectors is used as a measure of similarity between them. But this approach does not work well when the data is not linearly separable.

In such cases, a better method, known as kernel trick [23], is to transform the data into a higher dimensional space $\mathcal{H}$ through a mapping $\Phi : \mathcal{X} \mapsto \mathcal{H}$, such that the data is linearly separable in $\mathcal{H}$. The similarity between two points in this transformed space is given by a kernel function $K : \mathcal{X} \times \mathcal{X} \mapsto \mathbb{R}$ defined as

$$K(x, y) = \Phi(x)^T \Phi(y) \qquad x, y \in \mathcal{X}. \tag{1}$$

Berg et al. [27] has shown that for any symmetric function $K$, there exists a mapping $\Phi$ such that (1) holds if and only if $K$ is a positive definite kernel.

Kernel functions are also often used in the literature when achieving linear separability is not the primary concern. This is mostly found in graph based approaches [24], where one constructs graphs among the data instances such that the edges are weighted by kernel functions. Such a construction usually helps to obtain a low-dimensional embedding of the data points that is more suitable for data clustering [25]. Recent works in computer vision have generalized this technique to the case of hypergraphs, where one constructs edges that join multiple data instances, and are weighted by some multi-way similarity relation. The commonly used similarity relations are based on some geometric structure related to the data points that cannot be captured by pairwise similarities [18], [17].

In this paper, we propose multi-way similarities that have information theoretic connections.

### B. Jensen-type Divergences and Kernels

We briefly review the JT-kernel on the $d$-dimensional probability simplex

$$\Delta^{d-1} = \left\{ (p^{(1)}, \ldots, p^{(d)}) : p^{(j)} \in [0,1] \ \forall j, \ \sum_{j=1}^{d} p^{(j)} = 1 \right\}.$$

The Jensen-Tsallis $q$-difference among $n$ p.m.f.s $p_i = (p_i(1), \ldots, p_i(d)) \in \Delta^{d-1}$, $i = 1, \ldots, n$ is defined as [13]

$$T_q\left(p_1, \ldots, p_n\right) = H_q\left(\bar{p}\right) - \frac{1}{n^q} \sum_{i=1}^{n} H_q(p_i), \tag{2}$$

where $\bar{p} = (\bar{p}^{(1)}, \ldots, \bar{p}^{(d)})$ is the p.m.f. defined as $\bar{p}^{(i)} = \frac{1}{n} \sum_{j=1}^{n} p_j^{(i)}$, $i = 1, \ldots, d$, and $H_q$ is the Tsallis entropy [9] given by

$$H_q(p) = \frac{1}{(q-1)} \left( 1 - \sum_{j=1}^{d} \left( p^{(j)} \right)^q \right),$$

where $q \in \mathbb{R}$, $q \neq 1$ is a parameter related to the nature of the physical system. As $q \to 1$, the classical case of Shannon entropy [28]

$$H_1(p) = -\sum_{j=1}^{d} p^{(i)} \ln \left( p^{(j)} \right)$$

is retrieved, and the $q$-difference (2) in this case corresponds to the JS-divergence.

Based on (2), Martins et al. [13] defined a kernel $\widetilde{k}_q : \Delta^{d-1} \times \Delta^{d-1} \mapsto [0, \infty)$ as

$$\begin{aligned} \widetilde{k}_q\left(p_1, p_2\right) &= 2^q \left( \ln_q(2) - T_q\left(p_1, p_2\right) \right) \\ &= \frac{1}{(q-1)} \sum_{j=1}^{d} \left( \left( p_1^{(j)} + p_2^{(j)} \right)^q - \left( p_1^{(j)} \right)^q - \left( p_2^{(j)} \right)^q \right) \end{aligned} \tag{3}$$

for $q \neq 1$, which is the Jensen-Tsallis (JT) kernel between the two probability measures $p_1$ and $p_2$. The above class of kernels $\widetilde{k}_q$ is positive definite on $\Delta^{d-1}$ for $0 \leqslant q \leqslant 2$ [13]. For $q = 2$, we have a dot-product kernel on $\Delta^{d-1}$

$$\widetilde{k}_2\left(p_1, p_2\right) = 2 \sum_{j=1}^{d} p_1^{(j)} p_2^{(j)} = 2 \left( p_1^{(j)} \right)^T \left( p_2^{(j)} \right), \tag{4}$$

and in the limiting case of $q \to 1$, we have the JS-kernel

$$\begin{aligned} \widetilde{k}_1\left(p_1, p_2\right) = \sum_{j=1}^{d} \Big( & \left( p_1^{(j)} + p_2^{(j)} \right) \ln \left( p_1^{(j)} + p_2^{(j)} \right) \\ & - p_1(j) \ln \left( p_1^{(j)} \right) - p_2(j) \ln \left( p_2^{(j)} \right) \Big). \end{aligned} \tag{5}$$

Martins et al. [13] also indicated a similar possible generalization based on exponential JS-kernel [14]. This nonextensive kernel called exponential JT-(expJT) kernel is defined as

$$\begin{aligned} \tilde{k}_q^{(e)}\left(p_1, p_2\right) &= \exp\left( -t T_q(p_1, p_2) \right) \\ &= \exp\left( 2^q t \left( \tilde{k}_q(p_1, p_2) - \ln_q(2) \right) \right) \end{aligned} \tag{6}$$

for $t > 0$, $q \neq 1$, and it retrieves the exponential JS-kernel as $q \to 1$.

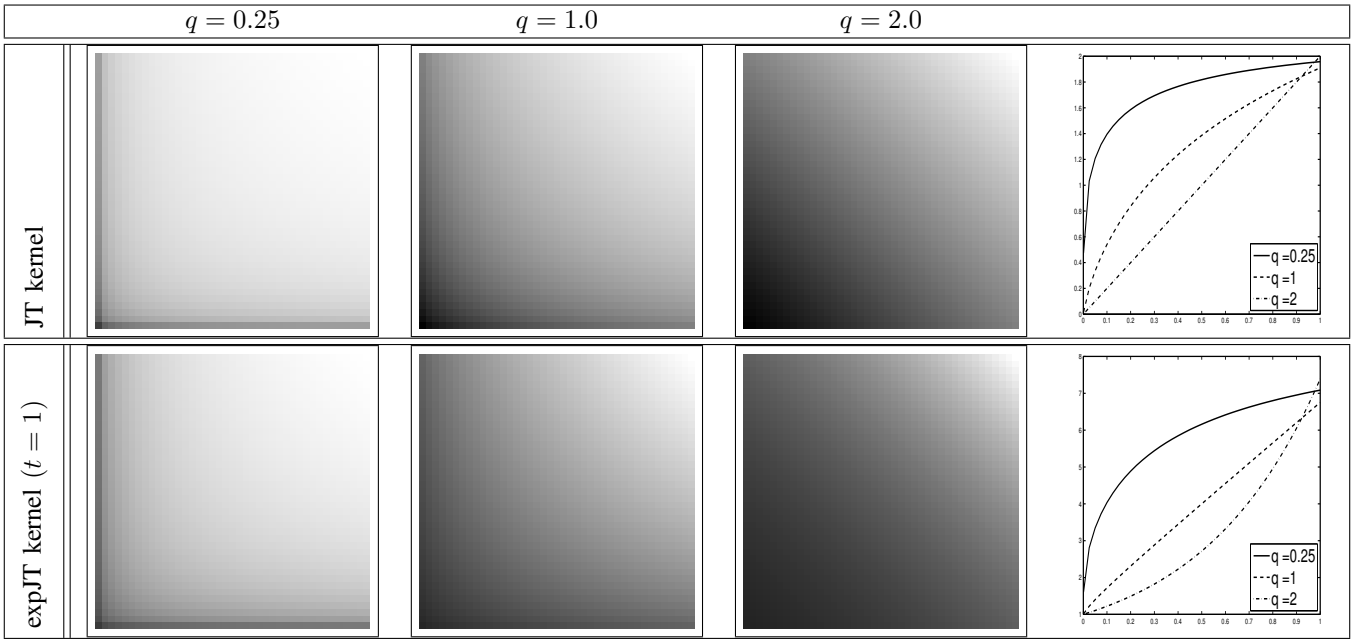


Fig. 1. Variation of the JT-kernel $k_q(x, x_0)$ and the expJT-kernel $k_q^{(e)}(x, x_0)$ defined over $[0, 1]^2$. In first three columns, magnitude of the kernels are displayed as function of $x$, when the second argument is fixed at $x_0 = (0.5, 0.5)$. The bottom left point of each figure denotes $(0, 0)$ and top right corresponds to $(1, 1)$. A brighter shade indicates a higher similarity of the point with $x_0$ as measured by the corresponding kernel. The last column corresponds to the case when the first argument $x$ varies along a line from $(0, 0)$ to $(1, 1)$. The horizontal axis denotes the abscissa of the first argument, and the vertical axis shows the value of the kernel function.

## III. Jensen kernels on Euclidean space

We first extend the above kernels to the Euclidean space. More specifically, we present extensions to the set $[0, 1]^d$. This is not restrictive since one usually normalizes features of data, and such a set suffices for most datasets. We proceed along the lines of the defined probability kernel (3), and define an extension of JT-kernel $k_q : [0, 1]^d \times [0, 1]^d \mapsto [0, \infty)$ of the form

$$k_q(x, y) = \begin{cases} \frac{1}{(q-1)} \sum_{j=1}^{d} \left( \left(x^{(j)} + y^{(j)}\right)^q - \left(x^{(j)}\right)^q - \left(y^{(j)}\right)^q \right), \\ \qquad\qquad \text{for } q \neq 1, \\ \sum_{j=1}^{d} \big( \left(x^{(j)} + y^{(j)}\right) \ln \left(x^{(j)} + y^{(j)}\right) \\ \qquad -x^{(j)} \ln \left(x^{(j)}\right) - y^{(j)} \ln \left(y^{(j)}\right) \big), \quad \text{for } q = 1, \end{cases} \tag{7}$$

where $x = (x^{(1)}, \ldots, x^{(d)}), y = (y^{(1)}, \ldots, y^{(d)}) \in [0, 1]^d$. The special case of linear kernel on $[0, 1]^d$ follows similar to (4). The same approach can be used to extend the expJT-kernel to define $k_q^{(e)} : [0, 1]^d \times [0, 1]^d \mapsto [0, \infty)$ as

$$k_q^{(e)}(x, y) = \exp\left(2^q t\big(k_q(x, y) - \ln_q(2)\big)\right) \quad \text{for } t > 0.$$

For simplified representation, we propose to define the exponential JT-kernel as

$$k_q^{(e)}(x, y) = \exp\left(t k_q(x, y)\right)) \quad \text{for } t > 0. \tag{8}$$

We note here that though one can define various measures to capture similarity among data instances, approaches such as kernel machines or kernel $k$-means can be used only when the similarity or kernel defines an inner product in a transformed space, as given in (1). It is well-known that this is ensured when the kernel function is positive definite [27]. We show that the proposed kernels in (7) and (8) are indeed positive definite.

**Proposition 1.** *JT-kernel $k_q$ and its exponential variant $k_q^{(e)}$ are positive definite for all $q \in [0, 2]$ and $t > 0$.*

The above result is proved in the Appendix. We do not delve into the reproducing kernel Hilbert space (RKHS) for these kernels, *i.e.*, the space where the kernels define an inner product. However, it intuitively seems that the RKHS of this class of kernels changes significantly with $q$. For instance, it is obvious that the RKHS of JT-kernel is same as the input space for $q = 2$, whereas one can argue that for $q \to 1$, the RKHS is an infinite dimensional space. We rather focus on the practical implication of the variation in RKHS.

Figure 1 illustrates the behavior of the proposed kernels. To be precise, we consider the kernels defined over $[0, 1]^2$. The first three columns in Figure 1 show the variation of $k_q(x, x_0)$ and $k_q^{(e)}(x, x_0)$ when $x_0$ is fixed at $x_0 = (0.5, 0.5)$ and $x$ varies over the entire domain. Note that the JT-kernel $k_2$ is the standard linear kernel, and clearly shows a linear variation in the similarity as $x$ changes. However, the other cases exhibit variations of different nature. In fact, for smaller values of $q$, both kernels tend towards a constant function over the entire domain. To further illustrate this effect, we have plotted the variation of the kernel values when $x$ is varied such that both its coordinates are equal. This is plotted in the last column of Figure 1, which shows how the JT-kernel deviates from the linear kernel. Similarly, the variation of the expJT-kernel also shows the effect of $q$ on the non-linear nature of the kernel.

We observe that in this case, $q = 1$ exhibits an almost linear nature, while the rate of exponential increase is higher for larger values of $q$. This suggests that the exponential variant has a more flexible structure, and can be expected to provide better performance.

## IV. Multi-point kernels

### A. Multi-point Jensen-type kernels

We present multi-point extensions of the JT-kernel (7) and the expJT kernel (8). The idea is based on the multi-distribution definition of Jensen-Tsallis $q$-difference (2), where $n$ need not be equal to 2. We extend the JT-kernel for arbitrary number of points in $\mathcal{X} = [0,1]^d$ to obtain a class of multi-point kernels $\{K_{q,n}\}_{n\in\mathbb{N}}$ with $K_{q,n} : \mathcal{X}^n \mapsto [0,\infty)$ defined as

$$K_{q,n}(x_1,\ldots,x_n) = \frac{1}{(q-1)}\sum_{j=1}^{d}\left[\left(\sum_{i=1}^{n} x_i^{(j)}\right)^q - \sum_{i=1}^{n}\left(x_i^{(j)}\right)^q\right] \quad (9)$$

for $q \neq 1$. In the case of $q \to 1$, we define the kernel as

$$K_{1,n}(x_1,\ldots,x_n) = \sum_{j=1}^{d}\left[\left(\sum_{i=1}^{n} x_i^{(j)}\right)\ln\left(\sum_{i=1}^{n} x_i^{(j)}\right) - \sum_{i=1}^{n} x_i^{(j)}\ln x_i^{(j)}\right]. \quad (10)$$

The above definition is consistent with multi-distribution extensions of the JT $q$-difference. Since it naturally extends a positive definite kernel, we refer to it as a kernel. In the sequel, we discuss a method for constructing a positive definite kernel from above multi-point kernel (see Proposition 2). A similar multi-point extension holds for the expJT-kernel $K_{q,n}^{(e)} : \mathcal{X}^n \mapsto [0,\infty)$ defined as

$$K_{q,n}^{(e)}(x_1,...,x_n) = \exp\left(tK_{q,n}(x_1,...,x_n)\right) \quad (11)$$

for $t > 0$. The above extension of two-point kernels captures information about similarity among multiple points, and is capable of providing a more global measure of similarity. Further, the proposed multi-point similarity is not dependent on any geometric model, unlike the ones in [18], [20], and hence, it is applicable in a more general framework.

The standard tools in machine learning often depend on the use of a symmetric similarity or kernel matrix. On the other hand, the above multi-point kernels lead to symmetric higher order tensors [21]. In this section, we comment on how one can interface these kernels with standard learning algorithms.

We begin our discussion with spectral clustering [25], [24]. To this end, in order to incorporate multi-point kernels into spectral clustering, we can rely on the higher order singular value decomposition of tensors [21]. According to this decomposition, given a symmetric tensor, one can derive an orthonormal matrix $U$ that generalizes the notion of eigenvector matrix. Furthermore, $U$ can be obtained from the eigen decomposition of a certain matrix $V$ computed from the entries of the tensor. Hence, we propose to use $V$ in our kernel computation.

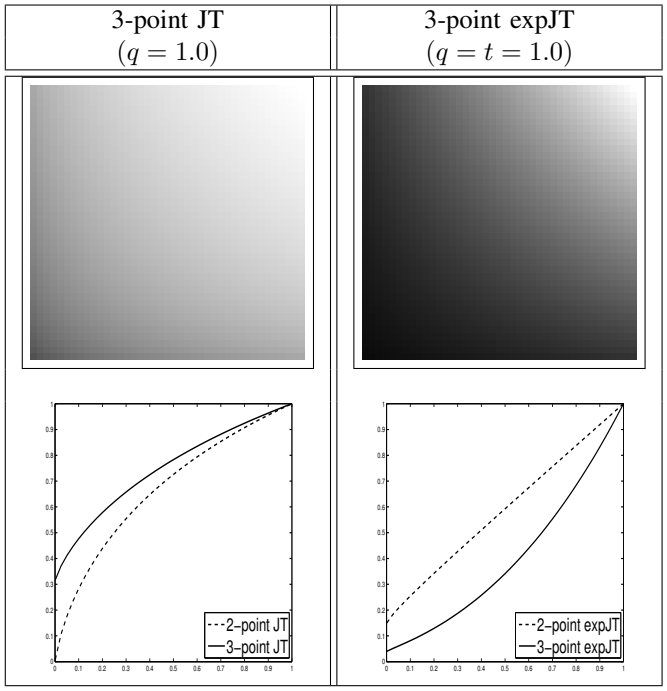


Fig. 2. Nature of the 3-point JT and expJT-kernels defined over $[0,1]^2$. The setting is same as that of Figure 1. The top row shows the variation of the 3-point kernels as a function of the first argument. The bottom row compares the nature of these kernels with corresponding 2-point kernels, when the first argument varies along a line from $(0,0)$ to $(1,1)$. The maximum value of the kernel functions have been normalized to unit value for better illustration.

Formally, given $N$ data points $x_1,\ldots,x_N \in [0,1]^d$ and some $n$-point kernel $K$, the matrix $V \in \mathbb{R}^{N\times N}$ is given by

$$V_{ij} = \sum_{i_2,\ldots,i_n=1}^{N} K(x_i, x_{i_2},...,x_{i_n})K(x_j, x_{i_2},...,x_{i_n}) \quad (12)$$

for $i,j = 1,\ldots,N$. One may also view $V$ as a pairwise similarity matrix, where $V_{ij}$ denotes certain similarity between data $x_i$ and $x_j$. Following the methods in [18], one can directly use the matrix $V$ as the affinity matrix in normalized spectral clustering [25]. Theoretical analysis for this technique can be found in [29].

While spectral clustering allows one to use any similarity measure, methods such as kernel $k$-means [22] or kernel support vector machines (SVM) [30] require the function to be such that the constructed similarity matrix is positive semi-definite. We show that the construction shown in (12) defines such a matrix.

**Proposition 2.** *Any multi-point kernel along with the computation in* (12) *defines a positive definite pairwise kernel.*

In Figure 2, we illustrate the nature of the 3-point Jensen-type kernels by conducting a similar exercise as in the previous section. The result shows that the structure of the kernels does not change drastically when we use multi-point variants. However, one can still expect a more flexible nature for the expJT kernel, as shown in the right column of Figure 2.

### B. Approximate multi-point kernels

It is easy to observe that if one uses (12), then the computation of the kernel matrix requires a time complexity of $O(N^{n+1})$. For large datasets, such a time complexity for kernel computation turns out to be quite inefficient, thereby limiting the applicability of multi-point kernels only to small data. In this section, we present an approximate method for computing the kernel matrix in (12). This approach is based on the column sampling technique for tensor flattening suggested in [18], [19]. The method is listed below.

1) For some given $C$, randomly select $C$ sets of data instances, each of size $(n-1)$.
2) Let $\{x_{1,l}, ..., x_{(n-1),l}\}$ denote the $l^{th}$ set of instances for $l = 1, \ldots, C$. Compute each entry of kernel matrix $V$ as

$$V_{ij} = \sum_{l=1}^{C} K(x_i, x_{1,l}, ..., x_{(n-1),l}) K(x_j, x_{i,l}, ..., x_{(n-1),l}). \tag{13}$$

Based on the proof of Proposition 2 and observing the structure of (13), one directly arrives at the following conclusion.

**Corollary 3.** *For a fixed $C$, any multi-point kernel along with the computation in* (13) *defines a random positive definite pairwise kernel matrix, that can be computed in $O(N^2)$ time.*

This complexity is same as that of computing standard 2-point kernels. In Section VI, we present the empirical performance of above approximation. To this end, we also note that in the case of large datasets, such as the ones encountered in image segmentation, one can easily combine the approximation of (13) with divide and conquer approaches [31] or Nyström approximation [32] to achieve a further reduction in computational complexity.

## V. The multi-point linear kernel

Here, we demonstrate a special case of the multi-point JT kernel, which is a natural generalization of the linear kernel. The interesting feature of this kernel is that in this case, one can compute $V$, defined in (12), exactly in cubic time complexity irrespective of the value of $n$. Observe that in the linear case, *i.e.*, for $q = 2$, the multi-point JT-kernel retrieves a multi-point version of the linear kernel as

$$K_{2,n}(x_1, \ldots, x_n) = 2 \sum_{i=1}^{n} \sum_{j=i+1}^{n} x_i^T x_j \ , \tag{14}$$

Henceforth, we call this the $n$-point linear kernel. The structure of this kernel helps to compute the matrix $V$ explicitly in cubic time as shown below.

**Proposition 4.** *Let $X = (x_1, x_2, \ldots, x_N) \in [0,1]^{d \times N}$ represent the given data matrix and $\bar{x} := \sum_{i=1}^{N} x_i$ be the component-wise addition of the vectors. Then, the matrix $V$ computed in* (12) *using the $n$-point linear kernel $K_{2,n}$* (14) *can be written as*

$$\begin{aligned} V = {} & 4\tbinom{n-1}{1} N^{n-2} \left(X^T X\right)^2 + 8\tbinom{n-1}{2} N^{n-3} \left(X^T \bar{x} \bar{x}^T X\right) \\ & + 8\tbinom{n-1}{2} N^{n-3} \left(X^T X X^T \bar{x} \mathbf{1}_{1\times N} + \mathbf{1}_{N\times 1} \bar{x}^T X X^T X\right) \\ & + 12\tbinom{n-1}{3} N^{n-4} \|\bar{x}\|_2^2 \left(X^T \bar{x} \mathbf{1}_{1\times N} + \mathbf{1}_{N\times 1} \bar{x}^T X\right) \\ & + 4\tbinom{n-1}{2} N^{n-5} \Big( N^2 \left\|X^T X\right\|_F^2 + 2(n-3) N \left\|X^T \bar{x}\right\|_2^2 \\ & \qquad + 2\tbinom{n-3}{2} \|\bar{x}\|_2^4 \Big) \mathbf{1}_{N\times N} \end{aligned} \tag{15}$$

*where $\mathbf{1}_{r\times s}$ denotes a $r \times s$ matrix of all 1's, $\|.\|_F$ is the Frobenius norm.*

The key fact in above result is that all computations in (15) are at most $O(N^3)$, which implies that $V$ is computable exactly in cubic time. Further, though the above result holds for any $n \in \mathbb{N}$, few simplifications are possible for $n \leqslant 4$. For instance, if $n = 2$, all terms vanish except first, giving $V = 4(X^T X)^2$, which has the same eigen structure as $X^T X$. Hence, spectral clustering with $V$ is equivalent to the case of constructing affinity using the Gram matrix.

One can also have a more general case of Proposition 4, which also leads to cubic time complexity. This is given in the following result, whose proof is similar to the above proof.

**Proposition 5.** *Let $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ be a symmetric kernel map, which defines a symmetric kernel matrix $\mathbf{K} \in \mathbb{R}^{N\times N}$ on $N$ given data points. Let $K : \mathcal{X}^n \to \mathbb{R}$ be a $n$-point extension of $k$ defined as*

$$K(x_1, \ldots, x_n) = \sum_{i=1}^{n} \sum_{j=i+1}^{n} k(x_i, x_j). \tag{16}$$

*From the above multi-point kernel, the matrix $V$ computed from* (12) *can be expressed as*

$$\begin{aligned} V = {} & \tbinom{n-1}{1} N^{n-2} \mathbf{K}^2 + 2\tbinom{n-1}{2} N^{n-3} \mathbf{K} \mathbf{1}_{N\times N} \mathbf{K} \\ & + 2\tbinom{n-1}{2} N^{n-3} \left(\mathbf{K}^2 \mathbf{1}_{N\times N} + \mathbf{1}_{N\times N} \mathbf{K}^2\right) \\ & + 3\tbinom{n-1}{3} N^{n-4} \mathbf{1}_{1\times N} \mathbf{K} \mathbf{1}_{N\times 1} \left(\mathbf{K} \mathbf{1}_{N\times N} + \mathbf{1}_{N\times N} \mathbf{K}\right) \\ & + \tbinom{n-1}{2} N^{n-5} \Big( N^2 \|\mathbf{K}\|_F^2 + 2(n-3) N \mathbf{1}_{1\times N} \mathbf{K}^2 \mathbf{1}_{N\times 1} \\ & \qquad + 2\tbinom{n-3}{2} \mathbf{1}_{1\times N} \mathbf{K} \mathbf{1}_{N\times N} \mathbf{K} \mathbf{1}_{N\times 1} \Big) \mathbf{1}_{N\times N}. \end{aligned}$$

Following the discussion in Section IV-B, we can also approximate the computation of multi-point linear kernel in $O(N^2)$ time. In the following result, we state this approximation in the general setting of Proposition 5.

**Corollary 6.** *If the approximate method of Section IV-B is used in the case of a $n$-point extension of a kernel, as defined in* (16)*, then the kernel matrix $V \in \mathbb{R}^{N\times N}$ is of the form*

$$V = (S + \mathbf{1}_{N\times 1} s^T)(S + \mathbf{1}_{N\times 1} s^T)^T,$$

*where $S \in \mathbb{R}^{N\times C}$ such that $S_{pl} = \sum_{j=1}^{n-1} k(x_p, x_{j,l})$, and $s \in \mathbb{R}^C$ such that $s^{(l)} = \sum_{p<j} k(x_{i_p,l}, x_{i_j,l})$.*

The above result makes it clear that the computation of $V$ can in done in $O(N^2)$ time.

## VI. Experimental results

In this section, we compare the proposed kernels with some popular kernels used in practice. Before presenting the numerical results, we list the computational complexity of computing the proposed kernels which we use for experiments. This is presented in Table I, where $N$ denotes the number of data instances. The table also lists the abbreviations that will be later used to refer to different kernels. The 2-point kernels have a time complexity of $O(N^2)$, which is standard in the literature. In case of large datasets, one can further reduce this complexity by considering various sampling strategies [32]. Table I shows that sampled variants of the multi-point kernels can be computed as efficiently as standard pairwise kernels.

TABLE I
COMPLEXITY OF COMPUTING DIFFERENT KERNELS.

| Kernel | Abbreviation | Complexity |
|---|---|---|
| 2-point Jensen-Tsallis | JT2 | $O(N^2)$ |
| 3-point Jensen-Tsallis | JT3 | $O(N^4)$ |
| Sampled variant of JT3 | JT3+ | $O(N^2)$ |
| 2-point exponential Jensen-Tsallis | expJT2 | $O(N^2)$ |
| 3-point exponential Jensen-Tsallis | expJT3 | $O(N^4)$ |
| Sampled variant of expJT3 | expJT3+ | $O(N^2)$ |
| $n$-point Linear | nLin | $O(N^3)$ |
| Sampled variant of nLin | nLin+ | $O(N^2)$ |

### A. Experiments on UCI datasets

We first study the performance of the proposed kernels in clustering and classification, in particular, when they are used to define similarities in algorithms such as spectral clustering, kernel $k$-means or kernel support vector machines (SVM). We conduct our experiments on some benchmark datasets from UCI repository [33][1]. The datasets and their characteristics are listed in Table II. These datasets have been previously used for comparative study of some clustering algorithms in [34], and kernel SVMs in [35].

TABLE II
LIST OF UCI DATA SETS CONSIDERED FOR COMPARATIVE STUDY. THE TOP 8 DATASETS HAVE BEEN USED IN CLUSTERING EXPERIMENTS, AND THE BOTTOM 5 FOR CLASSIFICATION.

| Data set | # instances | # attributes | # classes |
|---|---|---|---|
| Balance | 625 | 4 | 3 |
| Breast | 569 | 30 | 2 |
| Diabetes | 768 | 8 | 2 |
| German | 1000 | 24 | 2 |
| Ionosphere | 351 | 34 | 2 |
| Heart | 270 | 13 | 2 |
| Iris | 150 | 4 | 2 |
| Wine | 178 | 13 | 3 |
| Glass | 214 | 10 | 6 |
| Sonar | 208 | 60 | 2 |

We compare the performance of the Jensen-Tsallis kernels with some existing kernels such as Gaussian and polynomial kernels. In case of clustering, we also compare the results with the performance of some other clustering algorithms such as standard $k$-means algorithm (KM), spectral clustering with $k$ nearest neighbor based adjacency (SCNN), mean shift algorithm (MS), variants of maximum margin clustering (MMC), and minimal entropy encoding (MEE).

The performance measure considered is the Adjusted Rand index of the obtained clusters, defined as [34]

$$ARI = \frac{2(N_{00}N_{11}-N_{01}N_{10})}{(N_{00}+N_{01})(N_{01}+N_{11})+(N_{00}+N_{10})(N_{10}+N_{11})}$$

where $N_{11}$ denotes the number of pairs which are in the same clusters according to both true labels as well as obtained clusters, and $N_{00}$ is the number of pairs which have different labels and are also in different clusters. On the other hand, $N_{01}$ and $N_{10}$ are the number of pairs for which there is disagreement in the true and obtained clusters, where the former denotes the case of clustering pairs with different labels into the same cluster. $ARI$, also sometimes termed as corrected Rand index [26], is bounded above by 1, with larger value of $ARI$ indicating better clustering. The results for KM, SCNN, MS, MMC and MEE have been taken from [34]. For the different variants of MMC, we mention only the best reported result for each dataset.

In the experiments[2], we tune the parameters of the proposed and existing kernels, and report the best result in each case. This way of presenting the results have been adapted from [34]. For nLin, we vary $n \in \{2, 3, \ldots, 10\}$, while for other Jensen-type kernels $q$ is tuned as $q = 0.01$ or in the range $[0.25, 2]$ in steps of $0.25$. Both $t$ (for expJT) and $\sigma^2$ (for Gaussian) is varied from 0.01 to 100 in multiplicative step with a factor of 10, and for Polynomial kernel, we vary the degrees as $1, 2, \ldots, 10$. Moreover, for sampled variants of multi-point kernels, we sample only 50 columns to compute the approximate kernels. To account for the randomness in $k$-means initialization, as well as sampling, we average the results over 20 independent runs, as considered in [34].

Table III shows that Jensen-type kernels, particularly the exponential variety, perform quite well compared to other methods. Relative merits of the 2-point and 3-point kernels depend mostly on the data under study. We also observe that, except for the *balance* dataset, approximate kernel computation performs give performance quite close to that given using explicit kernel computation. In few cases, the nLin kernel is also observed to work with reasonable accuracy, particularly in comparison with KM, which is based on the linear kernel. We also study the variation in performance of the Jensen-type kernels as $q$ varies. Figure 3 shows the variation of ARI obtained when spectral clustering is performed with JT2 and JT3 kernels with varying $q$. It is observed that the nature of variation in ARI is mostly dependent on the data, but trends for 2-point and 3-point kernels are mostly similar.

Next, we turn to kernel SVMs, and compare the performance of the Jensen-type kernels with the kernels studied in [35]. The results are presented in Table IV, where we report the accuracy of kernel SVMs with various kernels. The experiments have been conducted along the lines of [35], where each data is randomly partitioned into two sets of equal size for training and testing. The kernel parameters are chosen

[1]Available at: `http://archive.ics.uci.edu/ml/datasets/`

[2]The MATLAB codes for our implementations are available at `http://sml.csa.iisc.ernet.in/SML/code/TNNLS15_code.zip`

TABLE III
*ARI* OBTAINED FROM DIFFERENT METHODS FOR CLUSTERING UCI DATASETS. FOR EACH DATASET, THE BEST RESULT IS IN BOLD FACE, AND THE BEST RESULTS OBTAINED FROM KERNEL $k$-MEANS OR SPECTRAL CLUSTERING ARE UNDERLINED.

| Method | | Balance | Breast | Diabetes | German | Heart | Ionosphere | Iris | Wine |
|---|---|---|---|---|---|---|---|---|---|
| | KM | 0.14 | 0.73 | 0.07 | 0.03 | 0.29 | 0.18 | 0.64 | 0.36 |
| | SCNN | 0.09 | **0.80** | 0.00 | 0.03 | 0.33 | 0.17 | 0.79 | 0.38 |
| | MS | 0.16 | 0.74 | 0.02 | 0.01 | 0.34 | 0.00 | 0.71 | 0.39 |
| | MMC | 0.18 | 0.74 | 0.10 | 0.02 | 0.31 | 0.30 | 0.73 | 0.37 |
| | MEE | 0.20 | 0.74 | 0.08 | **0.06** | 0.31 | **0.58** | **0.90** | 0.42 |
| Kernel $k$-means | Gaussian | 0.13 | 0.73 | 0.10 | 0.02 | 0.29 | 0.18 | 0.74 | 0.89 |
| | Polynomial | 0.14 | 0.73 | **0.18** | 0.02 | 0.38 | 0.17 | 0.71 | 0.84 |
| | JT2 | 0.16 | 0.76 | 0.10 | 0.02 | 0.30 | 0.29 | 0.70 | 0.87 |
| | JT3 | 0.02 | 0.69 | 0.15 | **0.06** | 0.37 | 0.27 | 0.62 | 0.40 |
| | JT3+ | 0.01 | 0.69 | 0.15 | 0.05 | 0.36 | 0.27 | 0.62 | 0.40 |
| | expJT2 | 0.16 | 0.76 | 0.17 | 0.03 | 0.41 | 0.37 | 0.73 | 0.88 |
| | expJT3 | 0.02 | 0.69 | **0.18** | **0.06** | **0.43** | 0.32 | 0.64 | 0.40 |
| | expJT3+ | 0.02 | 0.69 | **0.18** | 0.05 | 0.41 | 0.32 | 0.64 | 0.40 |
| | nLin | 0.01 | 0.66 | 0.15 | **0.06** | 0.33 | 0.15 | 0.63 | 0.30 |
| | nLin+ | 0.01 | 0.66 | 0.15 | **0.06** | 0.32 | 0.14 | 0.62 | 0.31 |
| Spectral clustering | Gaussian | 0.26 | 0.68 | 0.10 | 0.03 | 0.13 | 0.17 | 0.74 | 0.87 |
| | Polynomial | 0.26 | 0.57 | 0.09 | 0.03 | 0.21 | 0.06 | 0.58 | 0.80 |
| | JT2 | 0.43 | 0.57 | 0.05 | 0.03 | 0.30 | 0.14 | 0.65 | 0.81 |
| | JT3 | **0.54** | 0.52 | 0.05 | 0.03 | 0.26 | 0.16 | 0.61 | 0.87 |
| | JT3+ | 0.26 | 0.55 | 0.05 | 0.02 | 0.23 | 0.16 | 0.62 | 0.82 |
| | expJT2 | 0.49 | 0.79 | 0.10 | 0.04 | 0.25 | 0.19 | 0.60 | **0.95** |
| | expJT3 | 0.47 | 0.68 | 0.10 | 0.05 | 0.27 | 0.17 | 0.62 | 0.93 |
| | expJT3+ | 0.28 | 0.67 | 0.09 | 0.03 | 0.26 | 0.17 | 0.63 | 0.92 |
| | nLin | 0.43 | 0.67 | 0.09 | **0.06** | 0.20 | 0.15 | 0.55 | 0.88 |
| | nLin+ | 0.22 | 0.62 | 0.08 | 0.02 | 0.18 | 0.12 | 0.55 | 0.83 |

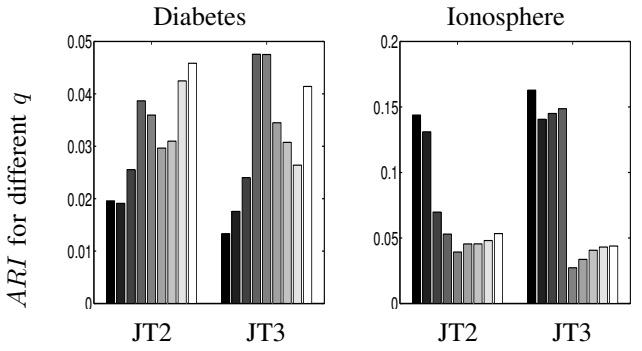


Fig. 3. Variation in performance of spectral clustering with 2-point and 3-point JT-kernels as parameter ($q$) varies. Each group of bars denote the ARI values for 9 different $q$ values in a particular case, where the bars are arranged from left to right in increasing order of $q$ values.

from the previously mentioned ranges using a two-fold cross validation in the training set. The chosen kernels are then used to evaluate accuracy on the test set. We have used standard implementations in LIBSVM [36][3] for our experiments.

The kernels considered for comparison include Gaussian and diffusion kernels, along with some geometry-aware kernels such as geometry-aware Gaussian (GA-Gaussian), geometry aware ideal kernel (GA-Ideal), order spectral kernel (OSK), simple non-parametric kernel (SimpleNPK) and geometry-aware metric learning (GML). The results for these kernels have been reported in [35]. In Table IV, we conduct 10 independent runs of the experiment, and report the mean and standard deviation of the percentage accuracy. The results show that while Jensen-type kernels give best performance in some cases, it does not work well in others. For instance, both in clustering and classification of iris dataset, this class of kernels is not able to extract the structure of clusters. In case of SVM, we also observe that multi-point kernels usually perform worse than the two-point kernels.

### B. Clustering cancer gene expressions

We also conduct experiments on clustering gene expressions. The cancer gene expression database [26][4] contains data sets related to two types of gene expressions: cDNA type and Affymetrix data sets. Some statistics of the data sets are provided in Table V. Further details are available in [26].

TABLE V
STATISTICS OF CANCER GENE EXPRESSION DATASETS.

| Data type | # datasets | # instances | | # attributes | | # classes | |
|---|---|---|---|---|---|---|---|
| | | min | max | min | max | min | max |
| cDNA | 14 | 37 | 179 | 86 | 4554 | 2 | 5 |
| Affy. | 21 | 22 | 248 | 183 | 2527 | 2 | 14 |

The performance of a number of clustering algorithms and proximity measures have been compared in [26]. The study concluded that best performance is usually obtained from $k$-means or mixture models, and spectral clustering works well in certain cases. We restrict our comparisons only to $k$-means and spectral clustering, but with different proximity measures or kernels. In [26], both algorithms were performed with proximity measures such as Pearson's correlation, cosine, Spearman correlation coefficient and Euclidean distance, where different data normalizations have been considered in the last case.

[3]Software available at: `http://www.csie.ntu.edu.tw/~cjlin/libsvm`

[4]Available at: `http://bioinformatics.rutgers.edu/Static/Supplements/CompCancer/index.htm`

TABLE IV
MEAN AND STANDARD DEVIATION OF ACCURACY FOR KERNEL SVM WITH DIFFERENT KERNELS. THE HIGHEST AVERAGE ACCURACY FOR EACH DATASET IS IN BOLD.

| Kernels | Glass | Heart | Iris | Sonar | Wine |
|---|---|---|---|---|---|
| Gaussian | 39.0±10.7 | 67.7±4.2 | 93.6±3.3 | 74.1±3.9 | 78.5±4.5 |
| Diffusion | 56.7±8.3 | 65.1±3.9 | 95.1±4.0 | 75.7±4.4 | 67.5±4.0 |
| OSK | 57.0±6.6 | 66.0±2.9 | 91.2±4.8 | **79.8**±3.4 | 68.1±5.3 |
| SimpleNPK | 56.4±3.7 | 63.2±3.3 | 94.5±4.3 | 75.4±5.5 | 70.7±4.5 |
| GML | 51.5±5.8 | 62.1±2.9 | 94.9±3.8 | 77.1±3.6 | 65.2±3.1 |
| GA-Gaussian | 55.4±8.4 | 73.0±3.3 | 93.6±4.3 | 77.0±3.2 | **97.4**±1.1 |
| GA-Ideal | 58.7±3.1 | **83.0**±2.7 | **95.5**±4.2 | 78.5±3.3 | 97.0±1.4 |
| JT2 | **76.6**±3.7 | 82.1±2.1 | 87.6±11.4 | 66.3±9.3 | **97.4**±1.1 |
| JT3 | 34.0±9.3 | 79.5±2.0 | 66.8±17.4 | 51.4±6.6 | 35.5±4.9 |
| JT3+ | 33.7±8.5 | 79.5±2.2 | 65.9±17.8 | 51.4±6.6 | 35.5±4.9 |
| expJT2 | 60.3±8.7 | **83.0**±1.6 | 85.1±5.1 | 61.2±8.4 | **97.4**±2.4 |
| expJT3 | 52.5±8.6 | 80.1±2.3 | 76.0±7.5 | 56.0±6.4 | 66.0±3.4 |
| expJT3+ | 50.6±10.7 | 80.0±2.3 | 75.7±7.1 | 55.8±6.3 | 65.7±3.4 |
| nLin | 34.7±11.3 | 79.7±2.9 | 84.4±2.8 | 53.6±6.0 | 57.5±8.7 |
| nLin+ | 34.3±10.2 | 79.4±2.4 | 84.1±2.9 | 53.5±6.0 | 57.8±9.5 |

Along with the reported results for above proximity measures, we also present the performance of the algorithms when we use the kernels listed in Table I. For Euclidean distance measure, we report only the result when data is normalized to the unit cube (termed as $Z_2$ in [26]), same as that considered for the other kernels. To account for the random initializations in $k$-means, [26] considered best result over 30 independent runs. We follow the same approach to obtain a fair comparison. However, to account for randomness in sampling in case of multi-point kernels, we still average over the results over 30 runs in this case.

Table VI presents a comparison of the different algorithms in terms of the corrected or adjusted Rand index $(ARI)$ as considered in [26]. The $ARI$ is averaged over all datasets over each type, and the best result for each type and each algorithm is underlined. The results show that in most cases, the expJT kernels surpass other similarity measures by some margin. While JT2 and expJT2 give good results in case of kernel $k$-means, their 3-point counterparts dominate in case of spectral clustering. However, unlike previous clustering experiments, here we find that for JT3 and expJT3, approximations lead to significant reduction in performance.

### C. Image segmentation

We also studied the performance of the proposed kernels in the context of image segmentation, where the similarities are constructed based on the pixel intensities and positions, and spectral clustering [24] is used to segment the image. The study is performed on the benchmark single object image segmentation database [37][5].

To improve the space and time complexity of segmentation, we follow the approach in [31] and divide each image into blocks of size $16 \times 16$ or $32 \times 32$. Segmentation of each block is performed based on the RGB values of the pixels. For 2-point kernels, such as Gaussian [24], JT-kernel (7) and expJT-kernel (8), the intensity based affinity matrix $V$ is directly computed from the kernel functions, whereas for the multi-point kernels as in (9) and (11), we use the computation in (12)

[5]Available at: http://www.wisdom.weizmann.ac.il/~vision/Seg_Evaluation_DB/1obj/index.html

TABLE VI
$ARI$ OBTAINED FROM $k$-MEANS AND SPECTRAL CLUSTERING FOR CLUSTERING GENE EXPRESSION DATASETS. THE RESULTS ARE AVERAGED OVER ALL DATASETS OF EACH TYPE. THE BEST RESULTS OBTAINED FROM KERNEL K-MEANS OR SPECTRAL CLUSTERING FOR EACH DATA TYPE ARE UNDERLINED.

| Method | | cDNA | Affymetrix |
|---|---|---|---|
| $k$-means | Pearson | <u>0.51</u> | 0.44 |
| | Cosine | 0.46 | 0.44 |
| | Spearman | – | – |
| | Euclidean | 0.38 | 0.39 |
| | Gaussian | 0.34 | 0.42 |
| | Polynomial | 0.44 | 0.49 |
| | JT2 | 0.45 | 0.56 |
| | JT3 | 0.14 | 0.33 |
| | JT3+ | 0.09 | 0.21 |
| | expJT2 | 0.49 | <u>0.62</u> |
| | expJT3 | 0.17 | 0.37 |
| | expJT3+ | 0.11 | 0.22 |
| | nLin | 0.14 | 0.33 |
| | nLin+ | 0.14 | 0.31 |
| Spectral clustering | Pearson | 0.33 | 0.39 |
| | Cosine | 0.32 | 0.42 |
| | Spearman | 0.27 | 0.40 |
| | Euclidean | 0.10 | 0.11 |
| | Gaussian | 0.27 | 0.27 |
| | Polynomial | 0.38 | 0.47 |
| | JT2 | 0.36 | 0.45 |
| | JT3 | 0.40 | 0.47 |
| | JT3+ | 0.24 | 0.29 |
| | expJT2 | 0.44 | 0.54 |
| | expJT3 | <u>0.47</u> | <u>0.57</u> |
| | expJT3+ | 0.28 | 0.35 |
| | nLin | 0.39 | 0.44 |
| | nLin+ | 0.34 | 0.39 |

to obtain the intensity based affinity matrix $V$. As shown in (15), this computation can be simplified for $n$-point linear kernel. We do not report the result for approximate multi-point kernels, which were observed to perform worse than their exact counterparts. Further to incorporate the proximity information for pixels, we scaled each entry in $V$ based on the distance among pixels as

$$\overline{V}_{ij} = V_{ij}^{(1-\lambda)} \exp\left(\frac{-\lambda \|c_i - c_j\|_2^2}{\sigma^2}\right), \qquad (17)$$

where $c_i$ and $c_j$ denote the coordinates of $i^{th}$ and $j^{th}$ pixels,


Fig. 4. Segmentation of images using spectral clustering with Gaussian and Jensen-type kernels.

and $\lambda = 0.08$ and $\sigma = 2$ are two constants. The pixels in each block is segmented into 8 initial segments. In the second stage, all the segments are further grouped into the desired number of output segments. This is again achieved by spectral clustering, where only adjacent segments are provided with a non-zero affinity computed as the Bhattacharya coefficient of their histogram of pixel intensities. The number of bins for the histogram are varied over the range $2^b$ for $b = 1, \ldots, 5$, and the Bhattacharya coefficient is raised by a factor that varies as 4, 8, 16 or 32.

TABLE VII
PERFORMANCE OF DIFFERENT KERNELS ON SINGLE OBJECT IMAGE SEGMENTATION DATABASE.

| Kernel | F-score | |
|---|---|---|
| | $16 \times 16$ blocks | $32 \times 32$ blocks |
| Gaussian | 0.76 | 0.78 |
| JT2 | 0.78 | **0.80** |
| JT3 | 0.71 | – |
| expJT2 | **0.80** | **0.80** |
| expJT3 | 0.74 | – |
| $n$-point linear | 0.74 | 0.75 |

Table VII shows the F-score for each kernel, averaged over all the images, when we use blocks of size $16 \times 16$ or $32 \times 32$. The F-score computation is performed using the code provided along with the database [37], and the parameters are tuned to improve the performance. We note that since the complexity for 3-point kernels is high, segmentation was not performed with them for $32 \times 32$ block size. The results show that 2-point expJT gives the best performance, which is also achieved by 2-point JT-kernel for $32 \times 32$ block size. Table VII clearly shows that the 2-point JT and expJT-kernels provide better performance than other kernels. The segmentation of some sample images are also provided in Figure 4.

## VII. DISCUSSIONS AND CONCLUDING REMARKS

In this paper, we studied the Jensen-Tsallis kernels on the Euclidean space that generalize the linear kernel. We proved that these kernels are positive semi-definite for $q \in [0, 2]$, and show that the nature of these kernels differ from standard distance based kernels. We also mentioned the possibility of defining multi-point extensions of these kernels. We showed a technique of generating positive-definite kernel from such multi-point similarities by the method of tensor flattening [21]. In the broad sense, the proposed kernels generalize the linear kernel, and hence, are expected to provide more flexibility in applications where the linear kernel is usually preferred over Gaussian or other related kernels.

Our elaborate empirical study on UCI datasets, gene expression dataset and image segmentation also provide some insight into the advantages and applicability of the proposed kernels. We observed that though considering multi-point extensions may be useful in the case of clustering, it does not provide further improvement in kernel SVMs. To this end, we also presented approximate computations of multi-point kernels in order to reduce the time complexity.

An interesting discussion related to multi-point linear kernels has been presented. Though the complexity of explicitly computing multi-point kernels is usually high, one can achieve cubic complexity in this case. The result can be stated in more general form for any positive definite pairwise kernel (Proposition 5). Hence, a study of multi-point extensions of this form may provide an interesting direction of research. To

motivate such study, we illustrate the nature of 4 and 6-point extensions of the Gaussian kernel in Figure 5.

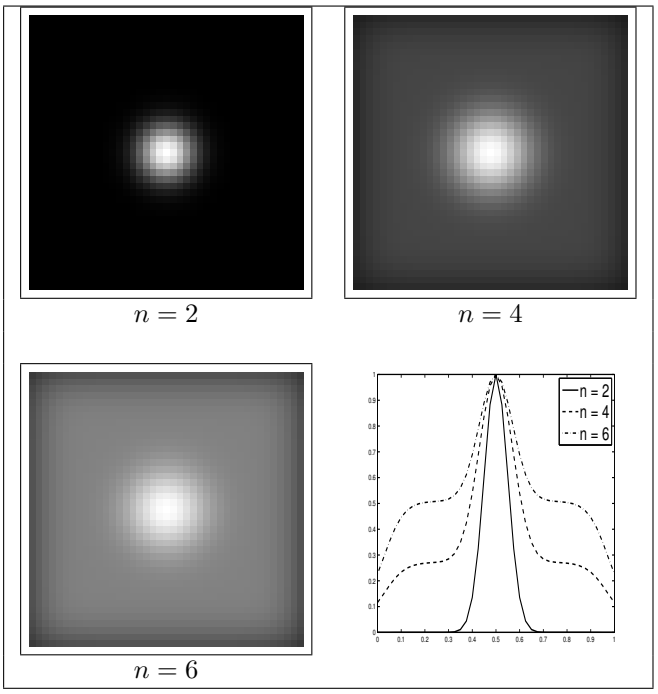


Fig. 5. Variation of the linear multi-point extension of the RBF kernel with $\sigma = 0.1$. The setting is same as Figure 1. The kernel function values have been normalized.

## APPENDIX

### *Proof of Proposition 1*

We mention few results on kernels [27], which will be used to arrive at the claim. For any space $\mathcal{X}$, the following results hold. We use the acronyms p.d. for positive definite kernel, and n.d. for negative definite kernel.

(R1) $\varphi : \mathcal{X} \times \mathcal{X} \mapsto \mathbb{R}$ is n.d. if and only if $\exp(-t\varphi)$ is p.d. for all $t > 0$.

(R2) If $\varphi : \mathcal{X} \times \mathcal{X} \mapsto \mathbb{R}$ is n.d. and $x_0 \in \mathcal{X}$, then $\psi : \mathcal{X} \times \mathcal{X} \mapsto \mathbb{R}$ defined as $\psi(x, y) = \varphi(x, x_0) + \varphi(y, x_0) - \varphi(x, y) - \varphi(x_0, x_0)$ is a p.d. kernel.

(R3) $\varphi : \mathcal{X} \times \mathcal{X} \mapsto \mathbb{R}$ is n.d. and $\varphi(x, x) \geqslant 0$ for all $x \in \mathcal{X}$ implies $\varphi^\alpha$ is n.d. for $\alpha \in [0, 1]$.

(R4) If $f : \mathcal{X} \mapsto \mathbb{R}$ is such that $f(x) \geqslant 0$ for all $x \in \mathcal{X}$, then for any $\alpha \in [1, 2]$, the map $\varphi_\alpha : \mathcal{X} \times \mathcal{X} \mapsto \mathbb{R}$ defined as $\varphi_\alpha(x, y) = -\big(f(x) + f(y)\big)^\alpha$ is a n.d. kernel.

(R5) $\psi_1, \psi_2 : \mathcal{X} \times \mathcal{X} \mapsto \mathbb{R}$ are p.d. kernels, then so are $(\psi_1 + \psi_2)$ and $c\psi_1$ for any $c > 0$.

(R6) $\{\varphi_n : \mathcal{X} \times \mathcal{X} \mapsto \mathbb{R}\}_{n \in \mathbb{N}}$ be a sequence of n.d. kernels, then $\lim\limits_{n \to \infty} \varphi_n$ is a n.d. kernel.

We first prove positive definiteness of $k_q$. In our case, the domain space is $\mathcal{X} = [0, 1]^d$. We prove the claim separately for three cases: $q \in (1, 2]$, $q \in [0, 1)$ and $q = 1$.

Case 1 - $q \in (1, 2]$ : Consider the functions $f_i : [0, 1]^d \mapsto \mathbb{R}$, $i = 1, \ldots, d$ as $f_i(x) = x(i)$ for $x = (x_1, \ldots, x_d) \in [0, 1]^d$. Then $f_i(x) \geqslant 0$ for all $x \in [0, 1]^d$, $i = 1, \ldots, d$. Hence, by (R4), $\varphi_{q,i}(x, y) = -(f_i(x) + f_i(y))^q = -(x(i) + y(i))^q$ is n.d. for $q \in (1, 2]$ and $i = 1, \ldots, d$. Applying (R2) with $x_0 = (0, \ldots, 0) \in [0, 1]^d$, we have $\psi_{q,i} = (x(i) + y(i))^q - x(i)^q - y(i)^q$ is p.d. for each $i = 1, \ldots, d$. Finally, by Result (R5), as $q > 1$, $k_q = \frac{1}{q-1} \sum_{i=1}^d \psi_{q,i}$ is p.d.

Case 2 - $q = 1$ : This is a consequence of the above case and an application of Result (R6). Consider for any $n \in \mathbb{N}$, $q_n = (1 + \frac{1}{n})$. As shown above, $\varphi_{q_n,i}(x, y) = -(x(i) + y(i))^{q_n}$ is n.d. for all $i = 1, \ldots, d$, $n \in \mathbb{N}$. Hence by (R6), $\varphi_{1,i} = \lim_{n \to \infty} \varphi_{q_n,i}$ is n.d., and using (R2) and (R5) as before, we have $k_1$ is p.d.

Case 3 - $q \in [0, 1)$ : We begin by noting that $\varphi_i(x, y) = (x(i) + y(i))$, $i = 1, \ldots, d$ are n.d. kernels. This can be derived from the definition of n.d. kernel. Then use of Result (R3) leads to the fact that $\varphi_i^q = (x(i) + y(i))^q$ is n.d. for $q \in [0, 1)$, $i = 1, \ldots, d$. Following the same procedure as before, we use (R2) and (R5) to claim that

$$k_q = \frac{1}{1 - q} \sum_{i=1}^d [x(i)^q + y(i)^q - (x(i) + y(i))^q]$$

is p.d. for $q \in [0, 1)$.

Thus, we have $k_q$ is p.d. Now, to claim $k_q^{(e)}$ is p.d., by (R1) and (8), it suffices to show that $\ln_q(2) - k_q$ is n.d., which is obvious from the fact that $k_q$ is p.d.

### *Proof of Proposition 2*

Note that for any $N$ data points, the values of the $n$-point kernel can be stored in a $n^{th}$-order tensor of dimension $N$. One can verify that The computation in (12) can be restated as $V = AA^T$, where $A \in \mathbb{R}^{N \times N^{n-1}}$ is a flattened matrix [21] obtained from the tensor whose each column contains the values of the $n$-point kernel when one data is varied and others are held fixed. From the representation $V = AA^T$, it is obvious the $V$ is positive semi-definite for any set of given data points. Hence, this process defines a positive definite kernel.

### *Proof of Corollary 3*

Follows from observing that $V = \sum_i a_i a_i^T$, where $a_i$ denotes $i^{th}$ column of $A$. Instead of summing over all columns, if we only sum over a fixed number of columns, $V$ is still positive semidefinite, but computation is only $O(N^2)$.

### *Proof of Proposition 4*

We provide a brief sketch of the proof. Note that $V$ can be written as

$$\begin{aligned} V = 4 \sum_{i_2, \ldots, i_n = 1}^N \Bigg[ & X^T \left( \sum_{l=2}^n \sum_{r=2}^n x_{il} x_{i_r}^T \right) X \\ & + X^T \left( \sum_{l=2}^n \sum_{r=2}^n \sum_{s=r+1}^n x_{il} x_{i_r}^T x_{i_s} \right) \mathbf{1}_{1 \times N} \\ & + \mathbf{1}_{1 \times N} \left( \sum_{r=2}^n \sum_{l=2}^n \sum_{k=l+1}^n x_{i_r} x_{il}^T x_{i_k} \right) X \\ & + \left( \sum_{l=2}^n \sum_{r=2}^n \sum_{k=l+1}^n \sum_{s=r+1}^n x_{il}^T x_{i_s} x_{i_r}^T x_{i_s} \right) \mathbf{1}_{N \times N} \Bigg]. \end{aligned} \quad (18)$$

Here we use the fact that given $i_2, \ldots, i_n$ and

$$j = \left(1 + \sum_{l=2}^{n} (i_l - 1) N^{l-2}\right),$$

$j^{th}$ column of $A$, is

$$2X^T \left(\sum_{l=2}^{n} x_{i_l}\right) + 2\left(\sum_{l=2}^{n}\sum_{k=l+1}^{n} x_{i_l}^T x_{i_k}\right)\mathbf{1}_{N\times 1}.$$

Comparing (18) and (15), we observe that the first term in (18) decomposes into the first two terms of (15). The second and third terms of (18) contribute to the third and fourth terms of (15), while the last term of (18) is equal to the last term in (15). Also, the outer summation in (18) may be pushed inside to simplify the results of the inner summations as shown below. For the first term, we consider the outer product of same and distinct vectors separately as

$$\begin{aligned}\sum_{i_2,\ldots,i_n=1}^{N} \sum_{l=2}^{n}\sum_{r=2}^{n} x_{i_l} x_{i_r}^T \\ = N^{n-2}\sum_{l=2}^{n}\sum_{i_l=1}^{N} x_{i_l} x_{i_l}^T + N^{n-3}\sum_{r,l=2, r\neq l}^{n}\sum_{i_l,i_r=1}^{N} x_{i_l} x_{i_r}^T\end{aligned} \tag{19}$$

since the terms act as constants while summing over all indices other than $i_l$ and $i_r$, and each such summation adds up $N$ similar terms, leading to the constants outside the summations. Now, one can verify that $XX^T = \sum_{i=1}^{N} x_i x_i^T$ and $\bar{x}\bar{x}^T = \sum_{i,j=1}^{N} x_i x_j^T$. Plugging this in (19), and noting that there are $(n-1)$ terms in the first summation and $2\binom{n-1}{2}$ terms in the second leads to the first two terms of (15). To deal with the second term of (18), it is enough to show that

$$\begin{aligned}\sum_{i_2,\ldots,i_n=1}^{N} \sum_{l=2}^{n}\sum_{r=2}^{n}\sum_{s=r+1}^{n} x_{i_l} x_{i_r}^T x_{i_s} \\ = 2\tbinom{n-1}{2} N^{n-3} XX^T\bar{x} + 3\tbinom{n-1}{3} N^{n-4}\|\bar{x}\|_2^2 \bar{x}\,.\end{aligned} \tag{20}$$

The constants $N^{n-3}$ and $N^{n-4}$ appear as before due to summation over indices, which are absent from the terms involved. We consider the cases $r = l$ and $r \neq l$ separately. For $r = l$, we obtain half of the first term in (20) since

$$\sum_{r=2}^{n}\sum_{s=r+1}^{n}\sum_{i_r,i_s=1}^{N} x_{i_r} x_{i_r}^T x_{i_s} = \tbinom{n-1}{2} XX^T\bar{x}.$$

For $r \neq l$, the situation becomes complicated as we may have $s = l$. But this happens only in $\binom{n-1}{2}$ cases, which adds up to give the remaining half of the first term in (20). The rest of the terms on the left in (20) have distinct indices, and hence, summing over them gives a term of the form $\sum_{i,j,k=1}^{N} x_i x_j^T x_k = \|\bar{x}\|_2^2\bar{x}$. But, there are $3\binom{n-1}{3}$ such terms, and hence, the result. Similarly, computing the other terms in (18), one can derive the expression in (15).

*Proof of Proposition 5*

Similar to above proof.

*Proof of Corollary 6*

We only need to show that the subsampled version of $A$ is of the form $(S+\mathbf{1}_{N\times 1}s^T)$. Consider the $l^{th}$ randomly sampled subset $\{x_{1,l}, \ldots, x_{(n-1),l}\}$. Then for any point $x$,

$$\begin{aligned}K(x, x_{1,l}, \ldots, x_{(n-1),l}) \\ = \sum_{i=1}^{n-1} k(x, x_{i,l}) + \sum_{i=1}^{n-1}\sum_{j=i+1}^{n=1} k(x_{i,l}, x_{j,l}).\end{aligned}$$

The first summation is a function of $x$, and contributes to the $l^{th}$ column of $S$, while the second term is constant for each chosen subset and contributes to $l^{th}$ entry of $s$.

## REFERENCES


[1] D. Ghoshdastidar, A. Dukkipati, A. P. Adsul, and A. Vijayan., "Spectral clustering with jensen-type kernels and their multi-point extensions," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*. Columbus, Ohio: IEEE, June 2014.

[2] D. Garcia-Garcia and R. C. Williamson, "Divergences and risks for multiclass experiments," in *25th Annual Conference on Learning Theory*, Edinburgh, Scotland, June 2012.

[3] F. Nielsen and R. Nock, "Total jensen divergences: Definition, properties and k-means++ clustering," *arXiv preprint*, vol. arXiv:1309.7109, 2013.

[4] I. Csiszár, "Information-type measures of difference of probability distributions and indirect observation," *Studia Scientiarum Mathematicarum Hungarica*, vol. 2, pp. 229–318, 1967.

[5] R. Sibson, "Information radius," *Probability Theory and Related Fields*, vol. 14, pp. 149–160, 1969.

[6] J. Lin, "Divergence measures based on the Shannon entropy," *IEEE Transactions on Info. Theory*, vol. 37, pp. 145–151, 1991.

[7] A. Banerjee, S. Merugu, I. S. Dhillon, and J. Ghosh, "Clustering with Bregman divergences," *Journal of Machine Learning Research*, vol. 6, pp. 1705–1749, 2005.

[8] D. M. Endres and J. E. Schindelin, "A new metric for probability distributions," *IEEE Transactions on Information Theory*, vol. 49, no. 7, pp. 1858–1860, 2003.

[9] C. Tsallis, "Possible generalization of Boltzmann-Gibbs statistics," *Journal of Statiscal Physics*, vol. 52, pp. 479–87, 1988.

[10] H. Suyari, "Generalization of Shannon-Khinchin axioms to nonextensive systems and the uniqueness theorem for the nonextensive entropy," *IEEE Transactions on Information Theory*, vol. 50, 2004.

[11] A. Dukkipati, M. N. Murty, and S. Bhatnagar, "Nonextensive triangle equality and their properties of Tsallis relative-entropy minimization," *Physica A: Statistical Mechanics and its Applications*, vol. 361, no. 1, pp. 124–138, 2006.

[12] J. C. Angulo, J. Antolín, S. López-Rosa, and R. O. Esquivel, "Jensen-Tsallis divergence and atomic dissimilarity for ionized systems in conjugated spaces," *Physica A: Statistical Mechanics and its Applications*, vol. 390, no. 4, pp. 769–780, 2011.

[13] A. F. T. Martins, N. A. Smith, E. P. Xing, P. M. Q. Aguiar, and M. A. T. Figueiredo, "Nonextensive information theoretic kernels on measures," *Journal of Machine Learning Research*, vol. 10, pp. 935–975, 2009.

[14] M. Cuturi, K. Fukumizu, and J. P. Vert, "Semigroup kernels on measures," *Journal of Machine Learning Research*, vol. 6, pp. 1169–1198, 2005.

[15] B. Fuglede and F. Topsoe, "Jensen-Shannon divergence and Hilbert space embedding," in *IEEE International Symposium on Information Theory*, Chicago, Illinois, USA, June 2004, p. 31.

[16] M. Bicego, A. F. T. Martins, V. Murino, P. M. Q. Aguiar, and M. A. T. Figueiredo, "2D shape recognition using information theoretic kernels," in *IEEE International Conference on Pattern Recognition (ICPR)*, Istanbul, Turkey, August 2010, pp. 25–28.

[17] S. Rota Bulo and M. Pelillo, "A game-theoretic approach to hypergraph clustering," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 6, pp. 1312–1327, 2013.

[18] V. M. Govindu, "A tensor decomposition for geometric grouping and segmentation," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 1, San Diego, CA, USA, June 2005, pp. 1150–1157.

[19] D. Ghoshdastidar and A. Dukkipati, "Spectral clustering using multilinear SVD: Analysis, approximations and applications," in *Proceedings of the 29th AAAI Conference on Artificial Intelligence*, Austin, Texas, USA, January 2015.
[20] M. Chertok and Y. Keller, "Efficient high order matching," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 32, no. 12, pp. 2205–2215, 2010.
[21] L. D. Lathauwer, B. D. Moor, and J. Vandewalle, "A multilinear singular value decomposition," *SIAM Journal on Matrix Analysis and Applications*, vol. 21, no. 4, pp. 1253–1278, 2000.
[22] B. Schölkopf, A. Smola, and K.-R. Müller, "Nonlinear component analysis as a kernel eigenvalue problem," *Neural Computation*, vol. 10, pp. 1299–1319, 1998.
[23] B. Schölkopf and A. J. Smola, *Learning with kernels: support vector machines, regularization, optimization, and beyond.* MIT Press, 2001.
[24] J. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, pp. 888–905, 2000.
[25] A. Y. Ng, M. I. Jordan, and Y. Weiss, "On spectral clustering: Analysis and an algorithm," in *Advances in Neural Information Processing Systems*, Vancouver, British Columbia, Canada, December 2001, pp. 849–856.
[26] M. C. P. de Souto, I. G. Costa, D. S. A. de Araujo, T. B. Ludermir, and A. Schliep, "Clustering cancer gene expression data: a comparative study," *BMC bioinformatics*, vol. 9, no. 1, p. 497, 2008.
[27] C. Berg, J. P. R. Christensen, and P. Ressel, *Harmonic Analysis on Semigroups: Theory of Positive Definite and Related Functions*. Springer, 1984.
[28] C. E. Shannon, "A mathematical theory of communication," *Bell System Technical Journal*, vol. 27, p. 379, 1948.
[29] D. Ghoshdastidar and A. Dukkipati, "Consistency of spectral partitioning of uniform hypergraphs under planted partition model," in *Advances in Neural Information Processing Systems*, Montreal, Quebec, Canada, December 2014.
[30] C. Cortes and V. Vapnik, "Support-vector networks," *Machine Learning*, vol. 20, pp. 273–297, 1995.
[31] F. Tung, A. Wong, and D. A. Clausi, "Enabling scalable spectral clustering for image segmentation," *Pattern Recognition*, vol. 43, no. 12, pp. 4069–4076, 2010.
[32] C. Fowlkes, S. Belongie, F. Chung, and J. Malik, "Spectral grouping using the Nyström method," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 2, pp. 214–225, 2004.
[33] M. Lichman, *UCI Machine Learning Repository*. University of California, Irvine: http://archive.ics.uci.edu/ml, 2013.
[34] S. Melacci and M. Gori, "Unsupervised learning by minimal entropy encoding," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 23, no. 12, pp. 1849–1861, 2012.
[35] B. Pan, W.-S. Chen, C. Xu, and B. Chen, "A novel framework for learning geometry-aware kernels," *IEEE Transactions on Neural Networks and Learning Systems*, 2015.
[36] C.-C. Chang and C.-J. Lin, "LIBSVM: A library for support vector machines," *ACM Transactions on Intelligent Systems and Technology*, vol. 2, no. 3, pp. 27:1–27, 2011.
[37] S. Alpert, M. Galun, R. Basri, and A. Brandt, "Image segmentation by probabilistic bottom-up aggregation and cue integration." in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Minneapolis, Minnesota, USA, June 2007.