Weighted Kernel Deterministic Annealing: A Maximum-Entropy Principle Approach for Shape Clustering

Mayank Baranwal[1] and Srinivasa M. Salapaka[2]

***Abstract*— Kernel *k*-means and spectral clustering methods have both been used extensively to cluster data that are non-linearly separable in input space. While there has been significant research since their inceptions, both the methods have some drawbacks. Similar to the basic *k*-means algorithm, the Kernel *k*-means algorithm is sensitive to initialization. On the other hand, the spectral methods are based on finding eigenvectors and can be computationally prohibitive. In this paper, we propose a novel maximum-entropy principle (MEP) based weighted-kernel deterministic annealing (WKDA) algorithm, which is *independent* of initialization and has ability to avoid poor local minima. Additionally, we show that the WKDA approach reduces to Kernel *k*-means approach as a special case. Finally, we extend the proposed algorithm to include constrained-clustering and present the results for a variety of interesting data sets.**

I. INTRODUCTION

Cluster analysis or *clustering* is a key element of unsupervised learning and has emerged as one of the fundamental problems in data mining in the recent years. It is used for exploratory data analysis to find hidden patterns in data, where the clusters are modeled using similarity measures based upon metrics such as Euclidean, Manhattan and Bergman divergences. These similarity measures represent distances of data points from their corresponding cluster centroids, or pairwise distances between any two data points in the input space.

The task of clustering is computationally difficult (**NP**-hard). A particularly well known approximation method is Lloyd's algorithm [1], often actually referred to as "*k*-means algorithm". It does however only find a local optimum, and is commonly run multiple times with different random initializations. To overcome the curse of initialization, Rose [2] proposed an annealing-based algorithm, well described in terms of laws such as maximum entropy principle (MEP) [3] in statistical physics literature, and showed that the solutions obtained using this approach are totally independent of the choice of initial configurations. The algorithm is referred as *deterministic annealing* (DA) algorithm and is aimed to provide high-quality solutions to clustering problem with only marginal increase in computational complexity.

A major drawback of both *k*-means and DA algorithms is their incapability to separate clusters that are non-linearly separable in input space. Fig. 1 shows the performance of

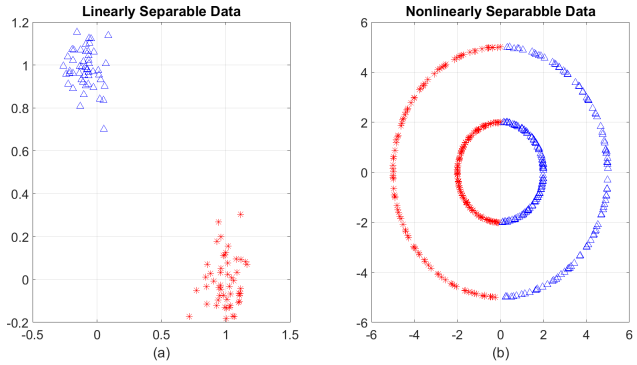

Fig. 1: Results of *k*-means clustering algorithm on (a) linearly separable input data, and (b) nonlinearly separable input data.

k-means algorithm in identifying natural clusters for two distinct distribution of data points. While the data points in Fig. 1a can be separated using a hyperplane in $\mathbb{R}^2$ (line), there is no such line that can separate data points distributed along two concentric circles in Fig. 1b. Thus, while *k*-means algorithm finds optimal linear separation of data points in Fig. 1b, such separations are indeed not natural and often undesired. Several approaches are proposed to tackle such a problem - (a) Agglomerative (or hierarchical) clustering [4], which uses linkage functions and distance thresholding on resulting drendograms, (b) Spectral clustering [5], [6], which requires computing eigenvectors of the associated graph Laplacian, and (c) Kernel *k*-means [7], which uses kernel-trick to map data points to higher-dimensional space and then clusters data points using linear separators in the new space. While performance of agglomerative clustering is sensitive to choice of linkage functions and thresholds on cutting the resulting drendograms, computation of eigenvectors of large sparse matrices in spectral clustering can have substantial computational overheads, especially when a large number of eigenvectors are to be computed. On the other hand, similar to the basic *k*-means algorithm, the kernel *k*-means algorithm is sensitive to initialization and a poor initialization may result in undesirable clustering performance.

To overcome these limitations, a novel weighted kernel deterministic annealing (WKDA) approach is presented in this paper. The WKDA algorithm enjoys the best of both worlds. On one hand, the algorithm is independent of initialization much similar to the basic DA algorithm; and on the other hand, WKDA does not require computing eigenvectors. Furthermore similar to kernel *k*-means, by choosing the weights in particular ways, the WKDA objective function is

[1]Mayank Baranwal is with the Department of Mechanical Engineering, University of Illinois, Urbana-Champaign, USA baranwa2@illinois.edu

[2]Srinivasa M. Salapaka is with the Faculty of Mechanical Engineering, University of Illinois, Urbana-Champaign, USA salapaka@illinois.edu

identical to the normalized cut. Thus we can use WKDA-like iterative algorithms for directly minimizing the normalized-cut of a graph.

The rest of the paper is organized as follows. Section II introduces the basic DA algorithm by Rose, which is modified for shape-clustering applications using kernel trick in Section III. We then specify WKDA's equivalence to kernel *k*-means approach and spectral clustering in Section IV, followed by evaluation of the WKDA algorithm on few example scenarios in Section V. We finally conclude the paper with directions to future work in Section VI.

Notations: We use capital letters such as X, Y to denote matrices; and lower case bold letters such as $\mathbf{x}, \mathbf{y}$ to denote column vectors. $N, k \in \mathbb{N}$ denote the number of data points and number of desired clusters, respectively. M denotes the number of attributes (or dimensions) of input data point. Script letters such as $\mathcal{X}, \mathcal{Y}$ represent sets; $\|\mathbf{x}\|$ denotes the L^2-norm of $\mathbf{x}$; and $\|X\|_F$ denotes the Frobenius norm of matrix X, and is given by $\|X\|_F = \left(\sum_{i,j} X_{ij}^2\right)^{1/2}$.

II. DETERMINISTIC ANNEALING (DA) ALGORITHM

The deterministic annealing (DA) algorithm views the task of clustering as an equivalent facility location problem (FLP), which concerns with optimal placement of facilities to minimize transportation costs from a given set of points to their nearest facilities. More precisely, given a set of $N \in \mathbb{N}$ points $\mathcal{X} = \{\mathbf{x_i} : \mathbf{x_i} \in \mathbb{R}^M, 1 \leq i \leq N\}$, the objective of an FLP is to find optimal locations of $k \in \mathbb{N}$ facilities denoted by $\mathcal{Y} = \{\mathbf{y_j} : \mathbf{y_j} \in \mathbb{R}^M, 1 \leq j \leq k\}$ such that the *aggregate weighted sum of distances of each point from its nearest facility location is minimized.* If $p(\mathbf{x_i})$ denotes the relative significance of point $\mathbf{x_i}$, then an FLP considers the following objective

$$\min_{\substack{\mathcal{Y}=\{\mathbf{y_j}\} \\ \mathcal{T}=\{t_{ij}\}}} \underbrace{\sum_{j=1}^{k}\sum_{i=1}^{N} t_{ij}\, p(\mathbf{x_i})\, d(\mathbf{x_i}, \mathbf{y_j})}_{D(\mathcal{X},\mathcal{Y})}, \tag{1}$$

where $\mathcal{T} = \{t_{ij} : t_{ij} \in \{0, 1\}\}$ is a set of associations with $t_{ij} = 1$ if facility $\mathbf{y_j}$ is allocated to point $\mathbf{x_i}$, otherwise $t_{ij} = 0$, and $d(\mathbf{x_i}, \mathbf{y_j}) = \|\mathbf{x_i} - \mathbf{y_j}\|^2$. Borrowing from data compression literature [8], the quantity $D(\mathcal{X}, \mathcal{Y})$ in (1) is often referred as *distortion* between set of data points $\mathcal{X}$ and facility locations $\mathcal{Y}$. Then the equivalent optimization problem is to minimize the distortion function. Solution to an FLP results in a set of clusters, where facility j is located at the centroid $\mathbf{y_j}$ of the j^{th} cluster, and each data point is associated only to its nearest facility (*Voronoi* partitions).

Most algorithms for FLP (such as Lloyd's [1]) start with some initial distribution of facility locations $\mathcal{Y}$ and iteratively optimize over them as the algorithm proceeds. However, such approaches are sensitive to the choice of initial facility locations, primarily die to the distributed aspect of the FLPs, where any change in the location of $\mathbf{x_i}$ affects $d(\mathbf{x_i}, \mathbf{y_j})$ only with respect to the *nearest* facility located at $\mathbf{y_j}$. The DA algorithm suggested by Rose [2], overcomes this sensitivity by allowing *fuzzy* association of every data point to each facility through an association probabilities $\{p(\mathbf{y_j}|\mathbf{x_i})\}$. This results in a modified *distortion* measure to reflect the weighted average distance of data points to *all* the facilities:

$$\bar{D}(\mathcal{X}, \mathcal{Y}) = \sum_{i=1}^{N} p(\mathbf{x_i}) \sum_{j=1}^{k} p(\mathbf{y_j}|\mathbf{x_i})\, d(\mathbf{x_i}, \mathbf{y_j}). \tag{2}$$

The probability distribution $\{p(\mathbf{y_j}|\mathbf{x_i})\}$ assesses the trade-off between decreasing the *local* influence and the deviation of the modified distortion $\bar{D}$ from the original distortion measure D. The uncertainty in associating facility locations $\mathcal{Y} = \{\mathbf{y_j}\}$ to locations of data points $\mathcal{X} = \{\mathbf{x_i}\}$ is captured by Shannon *entropy* term, widely used in data compression literature:

$$H(\mathcal{Y}|\mathcal{X}) = -\sum_{i=1}^{N} p(\mathbf{x_i}) \sum_{j=1}^{k} p(\mathbf{y_j}|\mathbf{x_i}) \log\left(p(\mathbf{y_j}|\mathbf{x_i})\right). \tag{3}$$

Note that maximizing the entropy is commensurate with decreasing the *local* influence. The trade-off between maximizing the entropy and minimizing the modified distortion in (2) is addressed by seeking the probability distribution $\{(p(\mathbf{y_j}|\mathbf{x_i})\}$ that minimize the *free-energy* function (or equivalent Lagrangian) given by

$$F \triangleq \bar{D}(\mathcal{X}, \mathcal{Y}) - \frac{1}{\beta} H(\mathcal{Y}|\mathcal{X}) + \sum_{i=1}^{N} \mu_i \left(\sum_{j=1}^{k} p(\mathbf{y_j}|\mathbf{x_i}) - 1\right), \tag{4}$$

where the last term corresponds to $\{p(\mathbf{y_j}|\mathbf{x_i})\}$ being a valid probability distribution. The Lagrange multiplier β bears a direct analogy to the inverse of the *temperature* variable in an annealing process [3]. Minimizing F at small values of β is equivalent to maximizing entropy H (a convex optimization problem). As β is increased gradually, minimization of F lays more emphasis on minimization of the distortion function. The association weights $\{p(\mathbf{y_j}|\mathbf{x_i})\}$ that minimize the free-energy function are given by the *Gibbs* distribution

$$p(\mathbf{y_j}|\mathbf{x_i}) = \frac{e^{-\beta d(\mathbf{x_i}, \mathbf{y_j})}}{\sum_{j'=1}^{k} e^{-\beta d(\mathbf{x_i}, \mathbf{y_{j'}})}}. \tag{5}$$

By substituting the Gibbs distribution into (4), the corresponding *free-energy* function is obtained as

$$F(\mathcal{Y}) = -\frac{1}{\beta} \sum_{i=1}^{N} p(\mathbf{x_i}) \log\left(\sum_{j=1}^{k} e^{-\beta d(\mathbf{x_i}, \mathbf{y_j})}\right). \tag{6}$$

In the DA algorithm, the *free-energy* function is *deterministically* optimized at successively increased values of the annealing parameter β. The optimal facility locations $\mathcal{Y}$ are obtained by setting the derivative of $F(\mathcal{Y})$ with respect to $\mathbf{y_j}$ to zero, thereby resulting in following update equation

$$\mathbf{y_j} = \frac{\sum_{i=1}^{N} p(\mathbf{x_i}) p(\mathbf{y_j}|\mathbf{x_i}) \mathbf{x_i}}{\sum_{i=1}^{N} p(\mathbf{x_i}) p(\mathbf{y_j}|\mathbf{x_i})}. \tag{7}$$

Note that the above equation has a form similar to computing centroids in *k*-means clustering algorithm. However in *k*-means clustering, the association between $\mathbf{x_i}$ and $\mathbf{y_j}$ are hard (0-1). The DA algorithm alternates between (5) and (7) at each β until convergence. In fact, the convergence of (6) is guaranteed as a consequence of coordinate descent on the free-energy function [9].

Since its inception, DA has been successfully applied to larger class of optimization problems such as, pattern classification [10], image segmentation [11], graph aggregation [12], robust speech recognition [13], expectation-maximization [14], coverage control [15] and scheduling problems [16].

III. WEIGHTED KERNEL DA

We propose weighted kernel deterministic annealing (WKDA) as an extension of the basic DA algorithm for shape clustering scenarios as shown in Fig. 1b. This is achieved by mapping the data $\mathcal{X}$ in the input space to a higher-dimensional feature space through an appropriate choice of *kernel* functions. This approach is referred as "kernel trick" and enables learning algorithms to operate in higher-dimensional feature space without ever explicitly computing the coordinates of data points in that space. The mapping allows to use linear separators to extract clusters in the *implicit* feature space.

Note that the squared Euclidean distance $d(\mathbf{x_i}, \mathbf{y_j})$ can be expressed using inner-products as

$$d(\mathbf{x_i}, \mathbf{y_j}) = < \mathbf{x_i}, \mathbf{x_i} > + < \mathbf{y_j}, \mathbf{y_j} > -2 < \mathbf{x_i}, \mathbf{y_j} >, \quad (8)$$

where $\mathbf{y_j}$ is defined in (7). For all $\mathbf{x_i}$ and $\mathbf{x_{i'}}$ in the input space $\mathcal{X}$, *kernel* functions $\kappa(\mathbf{x_i}, \mathbf{x_{i'}})$ can be expressed as an inner product in higher-dimensional, *implicit* feature space $\mathcal{H}$ using non-linear feature maps $\phi : \mathcal{X} \to \mathcal{H}$ which satisfies

$$\kappa(\mathbf{x_i}, \mathbf{x_{i'}}) = < \phi(\mathbf{x_i}), \phi(\mathbf{x_{i'}}) >_{\mathcal{H}} . \quad (9)$$

While explicit representation of ϕ is not necessary, its existence is guaranteed as long as κ satisfies *Mercer's* condition [17]. For a given set of data points $\mathcal{X} = \{\mathbf{x_1}, \ldots, \mathbf{x_N}\}$ in the input space, a *kernel matrix* $K \in \mathbb{R}^{N \times N}$ is given by $K_{ii'} = \kappa(\mathbf{x_i}, \mathbf{x_{i'}})$. Mercer's condition requires that K must be positive semi-definite (PSD). Empirically, for kernel-based algorithms, choices of kernel function κ that do not satisfy Mercer's condition may still perform reasonably if κ at least approximates the intuitive idea of similarity [18]. Many popular choices of κ exist, such as Gaussian, polynomial or radial basis function kernels. Using the non-linear distortion function ϕ, the distance between data point $\phi(\mathbf{x_i})$ and facility location $\mathbf{y_j}$ in the *implicit* feature space is given as

$$\underbrace{< \phi(\mathbf{x_i}), \phi(\mathbf{x_i}) > + < \mathbf{y_j}, \mathbf{y_j} > -2 < \phi(\mathbf{x_i}), \mathbf{y_j} >}_{d(\phi(\mathbf{x_i}), \mathbf{y_j})}, \quad (10)$$

with

$$\mathbf{y_j} = \frac{\sum_i p(\mathbf{x_i}) p(\mathbf{y_j}|\mathbf{x_i}) \phi(\mathbf{x_i})}{\sum_i p(\mathbf{x_i}) p(\mathbf{y_j}|\mathbf{x_i})}, \quad p(\mathbf{y_j}|\mathbf{x_i}) = \frac{e^{-\beta d(\phi(\mathbf{x_i}), \mathbf{y_j})}}{\sum_{j'} e^{-\beta d(\phi(\mathbf{x_i}), \mathbf{y_{j'}})}}. \quad (11)$$

Here (11) is a consequence of the DA algorithm with modified distance function $d(\phi(\mathbf{x_i}), \mathbf{y_{j'}})$. All computations in (10) are in the form of inner products, hence we can replace all inner products by entries of the kernel matrix, i.e.,

$$d(\phi(\mathbf{x_i}), \mathbf{y_j}) = K_{ii} - 2 \frac{\sum_l p(\mathbf{x_l}) p(\mathbf{y_j}|\mathbf{x_l}) K_{il}}{\sum_l p(\mathbf{x_l}) p(\mathbf{y_j}|\mathbf{x_l})} + \frac{\sum_{l,m} p(\mathbf{x_l}) p(\mathbf{x_m}) p(\mathbf{y_j}|\mathbf{x_l}) p(\mathbf{y_j}|\mathbf{x_m}) K_{lm}}{(\sum_l p(\mathbf{x_l}) p(\mathbf{y_j}|\mathbf{x_l}))^2}. \quad (12)$$

In the WKDA algorithm, the Euclidean distance in (12) is iteratively computed until convergence at each β value.

Algorithm 1 WKDA Algorithm

Input: $\mathcal{X} = \{\mathbf{x_1}, \ldots, \mathbf{x_N}\}$; No. clusters: k; Kernel Matrix K; Weight Matrix $W \triangleq \text{diag}\{p(\mathbf{x_1}), \ldots, p(\mathbf{x_N})\}$
Output: Cluster associations : $\{p(\mathbf{y_j}|\mathbf{x_i})\}$

```
Initialization:
p(y_j|x_i) ← 1/k        ∀  x_i ∈ X, y_j ∈ Y
β ← β_min
Annealing Process
while β < β_max do
    β Iterations
    while until convergence do
        Evaluate d(φ(x_i), y_j) as in (12)    ∀i, j
        Evaluate p(y_j|x_i) as in (11)    ∀i, j
    end while
    Increment β
end while
return {p(y_j|x_l)}
```

IV. CONNECTION WITH KERNEL k-MEANS AND SPECTRAL CLUSTERING ALGORITHMS

The WKDA algorithm (Algorithm 1) shares many properties with the *kernel k-means* algorithm described in [7]. For instance, when the association probabilities $\{p(\mathbf{y_j}|\mathbf{x_i})\}$ are hard (0-1), the distance function in (12) reduces to distance function for kernel *k*-means algorithm. Moreover similar to the DA algorithm, the WKDA algorithm decreases the objective function (modified free-energy function) in each β iteration.

For implementing the WKDA algorithm, we must compute the distance matrix $[d(\phi(\mathbf{x_i}), \mathbf{y_j})]$ during each iteration. The complexity of the WKDA algorithm can be analyzed using (12). The main complexity arises from computing the numerator of the third term in (12). The complexity is $O(N^4 k)$ scalar operations per iteration of computing the distance matrix. Thus if the total number of iterations is τ, then the complexity of the WKDA algorithm is $O(\tau N^4 k)$. The complexity can be significantly reduced using scalable implementation of the WKDA [19]. Such scalable implementation uses thresholding on association weights to minimize the number of scalar computations arising from associating every data point in the input space to *all* the clusters.

Similar to kernel *k*-means algorithm, the WKDA algorithm too exhibits connections with spectral clustering clustering

algorithms such as normalized cut and ratio cut methods. Note that the WKDA algorithm aims to optimize the expected distortion $\bar{D}(\phi(\mathcal{X}), \mathcal{Y})$ given by

$$\bar{D}(\phi(\mathcal{X}), \mathcal{Y}) = \sum_{i=1}^{N} p(\mathbf{x_i}) \sum_{j=1}^{k} p(\mathbf{y_j}|\mathbf{x_i}) d(\phi(\mathbf{x_i}), \mathbf{y_j}). \quad (13)$$

Let W_j be the diagonal matrix of all the $p(\mathbf{x_i})$ weights in the j^{th} cluster C_j, i.e., $W_j \triangleq \text{diag}\left\{p(\mathbf{x_{i_1}}), \ldots, p(\mathbf{x_{i_{|C_j|}}})\right\}$ $\forall i_l \in C_j$ and $W \triangleq \text{diag}\{W_1, \ldots, W_k\}$, then the minimization of total-distortion $\bar{D}(\phi(\mathcal{X}), \mathcal{Y})$ in the limiting case (i.e. $p(\mathbf{y_j}|\mathbf{x_i}) \in \{0, 1\}$) is equivalent to the following trace maximization problem [7]

$$\min_{\mathcal{Y}, \{C_j\}} \bar{D}(\phi(\mathcal{X}), \mathcal{Y}) \equiv \max_{U \in \mathbb{R}^{N \times k}} \text{Tr}\left(U^T W^{1/2} \underbrace{\Phi^T \Phi}_{K} W^{1/2} U\right), \quad (14)$$

where $\Phi = [\mathbf{\Phi_1}, \ldots, \mathbf{\Phi_k}]^T$ and $\mathbf{\Phi_j}$ is a matrix of points of the form $\phi(\mathbf{x_i})$ associated with cluster C_j, i.e., $\mathbf{\Phi_j} \triangleq [\phi(\mathbf{x_i})]$ $\forall i \in C_j$. The matrix U is of the form given by

$$U = \begin{bmatrix} \frac{W_1^{1/2}\mathbf{e_1}}{\sqrt{s_1}} & & \\ & \ldots & \\ & & \frac{W_k^{1/2}\mathbf{e_k}}{\sqrt{s_k}} \end{bmatrix}, \quad (15)$$

where $s_j = \sum_{i \in C_j} p(\mathbf{x_i})$ and $\mathbf{e_j}$ is a vector of ones of appropriate dimension. Note that U is an orthonormal matrix, i.e., $U^T U = I$. This discrete optimization problem is relaxed by allowing U to be an arbitrary orthonormal matrix. Using Rayleigh-Ritz theorem, the optimal U for the relaxed problem is obtained by taking the top k eigenvectors of $W^{1/2} K W^{1/2}$. Similarly, the sum of the top k eigenvalues of $W^{1/2} K W^{1/2}$ gives the optimal trace value.

On the other hand, for a graph $\mathcal{G}$ with edge-weight matrix A and degree-matrix D, the optimization problem for the relaxed normalized cut problem is given by

$$\max_{U \in \mathbb{R}^{N \times k}} \text{Tr}\left(U^T D^{-1/2} A D^{-1/2} U\right) \quad \text{s.t.} \quad U^T U = I. \quad (16)$$

Thus if we consider WKDA with $W = D$ and $K = D^{-1/2} A D^{-1/2}$, then the optimization problem in (16) is identical to the one in (14). Similarly, the optimization problem for the relaxed ratio cut problem is given by

$$\max_{U \in \mathbb{R}^{N \times k}} \text{Tr}\left(U^T A U\right) \quad \text{s.t.} \quad U^T U = I. \quad (17)$$

Choosing $W = D^{1/2}$ and $K = D^{-1/2} A D^{-1/2}$ establishes the equivalence between the WKDA algorithm and the ratio cut approach.

Thus, if the affinity matrix K is positive definite, we can use the WKDA procedure in order to minimize the normalized (or ratio) cut, without the need to compute eigenvectors.

Remark (Semi-supervised shape clustering): Semi-supervised clustering methods aim to improve clustering results using pairwise constraints, such as *must-link* and *cannot-link* constraints. These constraints can be incorporated into our framework through an appropriate modification of the kernel matrix. For every *cannot-link* constraint between $\mathbf{x_i}$ and $\mathbf{x_{i'}}$, the corresponding entry in the kernel matrix is set to zero, i.e., $K(i, i') = 0$. This can be understood as follows. The WKDA algorithm replicates normalized cut (or ratio cut) and aims to minimize the associated cut value. Setting $K(i, i') = 0$ is equivalent to setting the edge-weight between i i' to zero in the associated graph. Thus any cut separating i and i' incurs zero cost. Hence, such a choice of kernel matrix favors viability of *cannot-link* constraints.

Must-link constraints are relatively straight forward to handle. For every pair i, i' with *must-link* constraint between them, we require that the two points must be associated to the same cluster. This can be easily addressed in our framework by enforcing $p(\mathbf{y_j}|\mathbf{x_{i'}}) = p(\mathbf{y_j}|\mathbf{x_i})$ during each β iteration of the WKDA algorithm.

V. EXPERIMENTAL RESULTS

We now provide experimental results to validate the usefulness of the proposed WKDA algorithm. Our WKDA algorithm is implemented in MATLAB and all experiments are done on a PC (Windows, Intel i7-4790 CPU @ 3.6GHz processor, 8 GB RAM). Note that a geometric scheduling rate of β update (i.e. $\beta_{t+1} = 1.05\beta_t$) is employed and thus results in fast clustering performance. The kernel matrices are generated using Gaussian kernels.

We first present the results on standard shape data sets - 'flame', 'pathbased' and 'R15', with $2, 3$ and 15 natural clusters, respectively. These examples are downloaded from *https://cs.joensuu.fi/sipu/datasets/* under the shape sets category. The results are shown in Figs. 2a, 2b and 2c, respectively. Our WKDA algorithm successfully finds the underlying natural clusters in each of these examples. Similar performances are obtained for other standard shape data sets, too. However, the corresponding results are excluded for the sake of brevity.

Fig. 2d presents an artificially generated generated data set composed of the string - 'ICC2018'. The example contains 7 natural clusters in the form of individual characters of the string. Our WKDA algorithm correctly finds the underlying clusters in this example.

VI. CONCLUSIONS AND FUTURE WORK

In this paper, we present an innovative WKDA algorithm for shape clustering. The algorithm combines the kernel trick with the distributed aspect of the deterministic annealing algorithm to produce effective clustering solutions that are independent of initialization. We also present a theoretical connection between the WKDA algorithm and other existing approaches, such as kernel *k*-means and spectral clustering approaches. The algorithm is implemented on a variety of interesting examples and is shown to find the underlying natural shapes (clusters) in each of the example scenarios.

In future work, we would like to implement a scalable version of our algorithm with intelligent thresholding schemes to further minimize the run-time complexity of the WKDA algorithm. Another interesting aspect of the WKDA

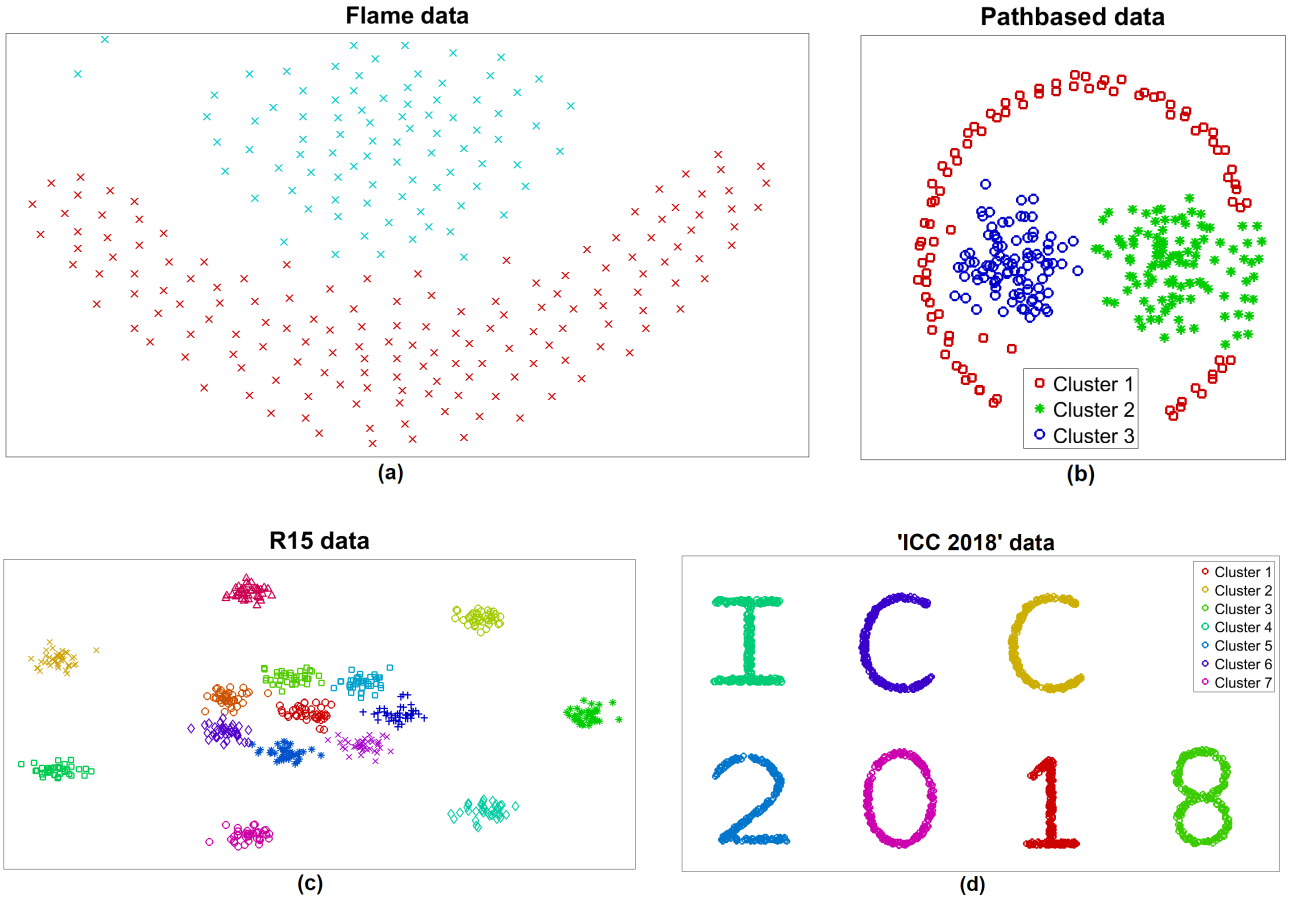

Fig. 2: Implementation of the proposed WKDA algorithm on some interesting data sets. The data sets in examples (a), (b) and (c) are obtained from *https://cs.joensuu.fi/sipu/datasets/* under the shape sets category. The example presented in (d) is an artificially generated data set.

algorithm is the existence of critical (pseudo)-temperature (β_{critical}). It can be shown that the location of facilities $\mathcal{Y}$ do not change significantly until β reaches β_{critical}. The value of β_{critical} can be obtained analytically using variational approach. This enables using even faster annealing schedule. We will address this aspect in our subsequent work. Finally, we would like to extend this approach to other combinatorial optimization problems such as traveling salesman problem (TSP).

ACKNOWLEDGMENT

The authors would like to acknowledge NSF grants ECCS 15-09302 and CNS 15-44635 for supporting this work.

REFERENCES

[1] S. P. Lloyd, "Least squares quantization in pcm," *Information Theory, IEEE Transactions on*, vol. 28, no. 2, pp. 129–137, 1982.

[2] K. Rose, "Deterministic annealing for clustering, compression, classification, regression, and related optimization problems," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2210–2239, 1998.

[3] E. T. Jaynes, "Information theory and statistical mechanics," *Physical review*, vol. 106, no. 4, p. 620, 1957.

[4] W. H. Day and H. Edelsbrunner, "Efficient algorithms for agglomerative hierarchical clustering methods," *Journal of classification*, vol. 1, no. 1, pp. 7–24, 1984.

[5] U. Von Luxburg, "A tutorial on spectral clustering," *Statistics and computing*, vol. 17, no. 4, pp. 395–416, 2007.

[6] 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.

[7] 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*. ACM, 2004, pp. 551–556.

[8] T. M. Cover and J. A. Thomas, *Elements of information theory*. John Wiley & Sons, 2012.

[9] M. Baranwal and S. M. Salapaka, "Clustering with capacity and size constraints: A deterministic approach," in *2017 Indian Control Conference (ICC)*, Jan 2017, pp. 251–256.

[10] P. V. Gehler and O. Chapelle, "Deterministic annealing for multiple-instance learning." in *AIStats*, 2007, pp. 123–130.

[11] S. Mitra, R. Castellanos, and S. Joshi, "An adaptive deterministic annealing approach for medical image segmentation," in *Fuzzy Information Processing Society, 2000. NAFIPS. 19th International Conference of the North American*. IEEE, 2000, pp. 82–84.

[12] Y. Xu, S. M. Salapaka, and C. L. Beck, "Aggregation of graph models and markov chains by deterministic annealing," *Automatic Control, IEEE Transactions on*, vol. 59, no. 10, pp. 2807–2812, 2014.

[13] A. Rao, K. Rose, and A. Gersho, "Design of robust hmm speech recognizers using deterministic annealing," in *Automatic Speech Recognition and Understanding, 1997. Proceedings., 1997 IEEE Workshop on*. IEEE, 1997, pp. 466–473.

[14] N. UDEA, "Deterministic annealing variant of the EM algorithm,"

Advances in Neural Information Processing Systems 7 (NIPS7), pp. 69–77, 1995.

[15] Y. Xu, S. Salapaka, and C. L. Beck, "Dynamic maximum entropy algorithms for clustering and coverage control," in *Decision and Control (CDC), 2010 49th IEEE Conference on*. IEEE, 2010, pp. 1836–1841.

[16] M. Baranwal, P. M. Parekh, L. Marla, S. M. Salapaka, and C. L. Beck, "Vehicle routing problem with time windows: A deterministic annealing approach," in *2016 American Control Conference (ACC)*, July 2016, pp. 790–795.

[17] C. Carmeli, E. De Vito, and A. Toigo, "Reproducing kernel hilbert spaces and mercer theorem," *arXiv preprint math/0504071*, 2005.

[18] B. Kulis, S. Basu, I. Dhillon, and R. Mooney, "Semi-supervised graph clustering: a kernel approach," in *Proceedings of the 22nd international conference on Machine learning*. ACM, 2005, pp. 457–464.

[19] P. Sharma, S. Salapaka, and C. Beck, "A scalable deterministic annealing algorithm for resource allocation problems," in *American Control Conference, 2006*. IEEE, 2006, pp. 6–pp.