Denoised Kernel Spectral Data Clustering

Raghvendra Mall
and Halima Bensmail
Qatar Computing Research Institute
Hamad Bin Khalifa University
Doha, Qatar
Email: rmall@qf.org.qa

Rocco Langone, Carolina Varon
and Johan A.K. Suykens
KU Leuven, ESAT-STADIUS
Kasteelpark Arenberg 10
B-3001 Leuven, Belgium

***Abstract*—Kernel Spectral Clustering (KSC) solves a weighted kernel principal component analysis problem in a primal-dual optimization framework. It builds an unsupervised model on a small subset of data using the dual solution of the optimization problem. This allows KSC to have a powerful out-of-sample extension property leading to good cluster generalization w.r.t. unseen data points. However, in the presence of noise that causes overlapping data, the technique often fails to provide good generalization capability. In this paper, we propose a two-step process for clustering noisy data. We first denoise the data using kernel principal component analysis (KPCA) with a recently proposed Model selection criterion based on point-wise Distance Distributions (MDD) to obtain the underlying information in the data. We then use the KSC technique on this denoised data to obtain good quality clusters. One advantage of model based techniques is that we can use the same training and validation set for denoising and for clustering. We discovered that using the same kernel bandwidth parameter obtained from MDD for KPCA works efficiently with KSC in combination with the optimal number of clusters k to produce good quality clusters. We compare the proposed approach with normal KSC and KSC with KPCA using a heuristic method based on reconstruction error for several synthetic and real-world datasets to showcase the effectiveness of the proposed approach.**

I. Introduction

Clustering algorithms have been widely used in fields like machine learning, data mining, graph compression, market segmentation etc. The objective in clustering is to divide the data into natural groups present in a given dataset. Clusters are defined such that the data present within a group are more similar to each other in comparison to the data between the clusters. Spectral clustering methods [2], [3], [4],[5] have attracted much attention off-late [6], [7]. In spectral clustering the eigenvectors of a normalized graph Laplacian L reveal the natural groups in the data. One of the issues with spectral clustering techniques is scalability to larger datasets as it requires to store an $N \times N$ matrix and perform a computationally expensive eigen-decomposition step ($O(N^3)$).

Recently, a new Kernel Spectral Clustering (KSC) algorithm was proposed in [8]. The method was based on a model built on a small subset of data in a primal-dual optimization framework. The model had a powerful out-of-sample extension property which allows to infer cluster affiliation for unseen data and helps to overcome the issue of scalability. The KSC methodology has been extensively applied for task of data clustering [8], [9], [10] and community detection [11], [12], [13] in large scale networks.

The KSC technique performs well in case of well-separated noise-free data. However, in the presence of noise, the KSC techniques fails because the model selection techniques like BAF [8] and BAF [11] cannot identify clear line structures. Thus, it becomes difficult to identify good quality clusters. There has been some literature related to clustering noisy data [14], [15], [16], [17]. The authors in [14], [15] perform clustering but in the presence of background noise, not necessarily noise that causes overlap in the data, using a robust centroid based clustering algorithm. A noise robust spectral clustering was proposed in [16] where the authors first warp the data to a new space using a regularization which includes a graph kernel inversion and then perform clustering.

In this paper we propose a two step-process to overcome the problem of clustering noisy overlapping data. In the first step, we perform denoising using kernel PCA (KPCA). We use the least squares support vector machines (LS-SVM) based formulation to KPCA [18]. An advantage of this primal-dual framework is that we have a proper training phase to build the model, validation phase to obtain the model parameters (σ and n_{pc}) and a test phase for an unsupervised problem. We use a recently proposed Model selection criterion based on Distance Distributions (MDD) [1] to obtain the model parameters for KPCA. In the second stage using the denoised version of the training and validation sets, we perform kernel spectral clustering to obtain clusters with good generalizations for noisy data. A major advantage is that we can use the Gaussian kernel parameter σ obtained as a result of KPCA with MDD as the model parameter for KSC to obtain good quality clusters.

978-1-5090-0620-5/16/$31.00 ©2016 IEEE

The remainder of the paper is structured as follows. Section II describes the LS-SVM formulation to kernel PCA. A brief explanation of the MDD technique is provided in Section III. Section IV describes the Kernel Spectral Clustering technique. In Section V we explain the two-step process proposed for handling noisy overlap data. Experimental results and analysis is provided in Section VI and finally we conclude in Section VII.

II. LS-SVM FORMULATION OF KPCA

LS-SVM formulation to unsupervised learning problems such as kernel PCA (KPCA) can be found in [18]. This formulation is set-up in a primal-dual optimization framework where the dual problem corresponds to the original formulation of kernel PCA [19]. Given a dataset $\mathcal{D} = \{x_i\}_{i=1}^{N}$, where $x_i \in \mathbb{R}^d$ and N represents the total number of points in the dataset, and $\phi(\cdot) : \mathbb{R}^d \rightarrow \mathbb{R}^{d_h}$ is the mapping to high dimensional space $\mathcal{F}$ of dimension d_h, the primal formulation to KPCA [18] is given as follows:

$$\begin{aligned} \min_{w,e,b} \quad & \frac{1}{2}w^\intercal w - \frac{\gamma}{2N_{tr}}\sum_{i=1}^{N_{tr}} e_i^2 \\ \text{such that} \quad & e_i = w^\intercal\phi(x_i) + b, i = 1,\ldots,N_{tr}, \end{aligned} \tag{1}$$

where γ is a regularization term and b is the bias term. Here N_{tr} represents the training points s.t. $N_{tr} < N$. The Lagrangian of this primal problem is defined as

$$\mathcal{L}(w,e,b;\alpha) = \frac{1}{2}w^\intercal w - \frac{\gamma}{2}\sum_{i=1}^{N_{tr}} e_i^2 + \sum_{i=1}^{N_{tr}} \alpha_i(e_i - w^\intercal\phi(x_i) - b), \tag{2}$$

The corresponding dual problem is:

$$\Omega_c\alpha = \lambda\alpha \tag{3}$$

where $\Omega_c = M_c\Omega M_c$ is a centered kernel matrix where $M_c = I - \frac{1}{1_{N_{tr}}}1_{N_{tr}}1_{N_{tr}}^\intercal$ and $\Omega_{ij} = K(x_i,x_j) = \phi(x_i)^\intercal\phi(x_j)$ for $i,j = 1,\ldots,N_{tr}$. Here $K(\cdot,\cdot)$ is the Mercer kernel function which generates the kernel matrix and $1_{N_{tr}} = [1;1;\ldots;1]$ is a vector of ones. For all the experiments in this work, we use the radial-basis function (RBF) kernel.

The projection of a new data point x onto the subspace spanned by the most relevant principal axes in the feature space is depicted as:

$$\hat{e}(x) = \sum_{i=1}^{N_{tr}} \alpha_i K(x,x_i) + b \tag{4}$$

These projected variables are called score variables. The eigenvalue problem in equation 3 satisfies the KKT conditions of optimality and each eigenvalue-eigenvector pair of the kernel matrix is a solution to equation 1. The eigenvector of Ω_c corresponding to the largest eigenvalue is the one corresponding to direction of maximum variance. The maximum number of eigenvectors that can be obtained is equivalent to the size of the training set i.e. N_{tr}.

For denoising it is essential to map the transformed points from the feature space back to the input space. This is the so called pre-image problem and it is well know that a unique solution will not always exist [20], [21]. Several methodologies have been proposed to find approximate pre-images. In this work we use the iterative fixed point algorithm for RBF kernels [20] implemented in the LS-SVMLab Matlab http://www.esat.kuleuven.be/stadius/lssvmlab/ toolbox.

The two model parameters that need to be tuned for KPCA are the kernel bandwidth parameter σ and the number of principal components n_{pc}. One simple heuristic is to select as few components as possible along with the right σ such that they minimize the reconstruction error i.e. $||x_i - h(e_i)||_2^2$, where $h(\cdot)$ maps the score variables to reconstructed input variables. However, this techniques does not work well in case of noisy overlapping data as it tries to even fit the noise so as to have a minimal reconstruction error. We need to devise/utilize a metric which allows to remove the noise information and capture the inherent information content in the dataset.

III. MDD:MODEL SELECTION USING DISTANCE DISTRIBUTION

In order to select the model parameters n_{pc} and σ^2, the methodology proposed in [1] is implemented. This methodology is based on the distance distribution of the dataset, and it assumes that the probability distribution of the pair-wise distances of an unstructured dataset is unimodal. In addition, it takes into account the relationship between the statistical moments of such distributions and the dimensionality of the data. An example of a structured dataset and its distance distribution are shown in Figures 1(a) and 1(b), respectively.

Given the training set $\mathcal{D}_{tr} = \{x_i\}_{i=1}^{N_{tr}}$, and the validation set $\mathcal{D}_v = \{x_m\}_{m=1}^{N_{valid}}$, with $x_i, x_m \in \mathbb{R}^d$, the Euclidean distances $r_{ij} = \|x_i - x_j\|_2$ are calculated between $x_i \in \mathcal{D}_{tr}$ and $x_j \in (\mathcal{D}_{tr} \cup \mathcal{D}_v)$, where $i = 1,\ldots,N_{tr}$, $j = 1,\ldots,N_{tr} + N_{valid}$, and $0 \leq \hat{r} \leq r_{max}$. The maximum distance in the dataset is denoted by r_{max}. The pairwise distances result in a distribution $\hat{f}_{\mathcal{D}}(r)$, with mean μ, standard deviation σ, skewness $\hat{y}$, and kurtosis β. These statistical moments are then used to generate a "noise" distribution $\hat{f}_{\mathcal{N}}(\hat{r})$, from which new values are drawn and used as entries of a new distance matrix $\hat{R} \in \mathbb{R}^{N_{tr}\times N_{tr}}$. This matrix $\hat{R}$ is organized in such a way that the conditions of a metric are satisfied. For details on the computation of $\hat{R}$, the reader is referred to [1]. Figure 1(b) shows the distance distributions of both the data and the estimated noise.

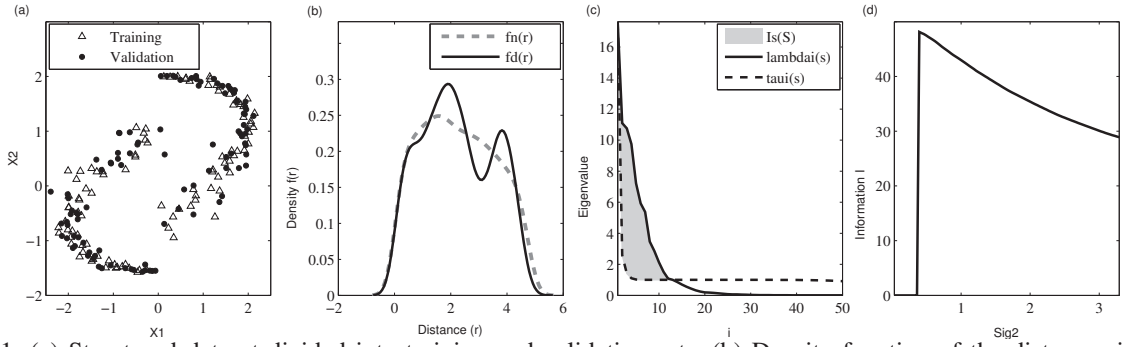


Fig. 1: (a) Structured dataset divided into training and validation sets. (b) Density function of the distances in the dataset $\hat{f}_{\mathcal{D}}(r)$, and the noise distribution $\hat{f}_{\mathcal{N}}(r)$ created using the statistical moments of $\hat{f}_{\mathcal{D}}(r)$. (c) Eigenvalue spectra of the data (λ_i) and noise ($\hat{\tau}_i$) for a given σ^2. The shaded area indicates the amount of information $I_s(\sigma^2)$ contained in first 13 components. (d) Information content I_s as a function of the kernel parameter σ^2.

After creating $\hat{R}$, a new kernel matrix $\Omega_{\mathcal{N}_{ij}} = K_{\mathcal{N}}(x_i, x_j) = \exp(-\hat{r}(x_i, x_j)^2/2\sigma^2)$, is computed. Next, the eigenvalue problem

$$\Omega_{\mathcal{N}_c}\nu = \hat{\tau}\nu, \tag{5}$$

with $\Omega_{\mathcal{N}_c} = M_c\Omega_{\mathcal{N}}M_c$, is used to compute the eigenspectrum $\hat{\tau}_i$ of the noise for a given kernel parameter σ^2. Here M_c refers to the centering matrix. This eigenspectrum is then compared with the eigenspectrum λ_i, calculated using (3). Examples of these two eigenspectra are shown in Figure 1(c). The number of components, for a given σ^2, that describe the underlying structure of the data can be selected as

$$n_{pc}(\sigma^2) = \max_{\lambda_i - \hat{\tau}_i > 0} i, \tag{6}$$

and the amount of information I_s contained in the selected components is defined as

$$I_s(\sigma^2) = \sum_{i=1}^{n_{pc}(\sigma^2)} \lambda_i - \hat{\tau}_i, \tag{7}$$

and indicated by the shaded area in Figure 1(c). Figure 1(d) shows the value of I_s for different values of σ^2. The maximum value of this curve is then used to select the kernel parameter σ^2, which corresponds to the value that maximizes the information content in the components.

IV. Kernel Spectral Clustering

A. Primal-Dual Weighted Kernel PCA framework

Given a dataset $\mathcal{D} = \{x_i\}_{i=1}^N$, $x_i \in \mathbb{R}^d$, the training points are selected by maximizing the quadratic Rènyi criterion as depicted in [9], [18] and [22]. Given $\mathcal{D}$ and the number of clusters k, the primal problem of the spectral clustering via weighted kernel PCA is formulated as follows [8]:

$$\begin{aligned} \min_{w^{(l)}, e^{(l)}, b_l} \quad & \frac{1}{2}\sum_{l=1}^{k-1} w^{(l)\intercal} w^{(l)} - \frac{1}{2N_{tr}}\sum_{l=1}^{k-1}\gamma_l e^{(l)\intercal} D_\Omega^{-1} e^{(l)} \\ \text{such that} \quad & e^{(l)} = \Phi w^{(l)} + b_l 1_{N_{tr}}, l = 1, \ldots, k-1, \end{aligned} \tag{8}$$

where $e^{(l)} = [e_1^{(l)}, \ldots, e_{N_{tr}}^{(l)}]^\intercal$ are the projections onto the eigenspace, $l = 1, \ldots, k-1$ indicates the number of score variables required to encode the k clusters, $D_\Omega^{-1} \in \mathbb{R}^{N_{tr}\times N_{tr}}$ is the inverse of the degree matrix associated to the kernel matrix Ω. Φ is the $N_{tr} \times d_h$ feature matrix, $\Phi = [\phi(x_1)^\intercal; \ldots; \phi(x_{N_{tr}})^\intercal]$ and $\gamma_l \in \mathbb{R}^+$ are the regularization constants. We note that $N_{tr} < N$ i.e. the number of points in the training set is less than the total number of data points in the dataset. The kernel matrix Ω is obtained by calculating the similarity between each pair of data points in the training set. Each element of Ω, denoted as $\Omega_{ij} = K(x_i, x_j) = \phi(x_i)^\intercal\phi(x_j)$ is obtained by using the RBF kernel. The clustering model is given as:

$$e_i^{(l)} = w^{(l)\intercal}\phi(x_i) + b_l, i = 1, \ldots, N_{tr}, \tag{9}$$

where $\phi : \mathbb{R}^d \to \mathbb{R}^{d_h}$ is the mapping to a high-dimensional feature space d_h, b_l are the bias terms, $l = 1, \ldots, k-1$. The projections $e_i^{(l)}$ represent the latent variables of a set of $k-1$ binary cluster indicators given by $\text{sign}(e_i^{(l)})$ which can be combined with the final groups using an encoding/decoding scheme. The decoding consists of comparing the binarized projections w.r.t. codewords in the codebook and assigning cluster membership based on minimal Hamming distance. The dual problem corresponding to this primal formulation is:

$$D_\Omega^{-1} M_D \Omega \alpha^{(l)} = \lambda_l \alpha^{(l)}, \tag{10}$$

where M_D is the centering matrix which is defined as $M_D = \mathrm{I}_{N_{tr}} - (\frac{(1_{N_{tr}} 1_{N_{tr}}^\intercal D_\Omega^{-1})}{1_{N_{tr}}^\intercal D_\Omega^{-1} 1_{N_{tr}}})$. The $\alpha^{(l)}$ are the

dual variables and the positive definite kernel function $K : \mathbb{R}^d \times \mathbb{R}^d \rightarrow \mathbb{R}$ plays the role of similarity function. This dual problem is closely related to the random walk model as shown in [8]. The KSC formulation is just a weighted version of KPCA where the weighting term is D_Ω^{-1}.

B. Out-of-Sample Extensions Model

The projections $e^{(l)}$ define the cluster indicators for the training data. In the case of an unseen data point x, the predictive model becomes:

$$\hat{e}^{(l)}(x) = \sum_{i=1}^{N_{tr}} \alpha_i^{(l)} K(x, x_i) + b_l. \qquad (11)$$

This out-of-sample extension property allows kernel spectral clustering to be formulated in a learning framework with training, validation and test stages for better generalization. The validation stage is used to obtain the model parameters like the kernel parameter (σ for RBF kernel) and the number of clusters k in the dataset. The data points corresponding to the validation set are also selected by maximizing the quadratic Rènyi entropy criterion.

C. Model Selection

Several model selection criterion [8], [11] have been proposed to estimate the right number of clusters k and the σ for the RBF function. However, these criterion work on the assumption on piece-wise or nearly piece-wise constant eigenvectors and using line structure (BLF [8]) or angular similarity (BAF [11]) of the projections of the validation points in the eigenspace. However, in case of noisy overlapping data, the inherent cluster structure is lost and the user has to provide the optimal number of groups k to search for in the dataset.

V. Proposed Approach

Here we put together all the pieces of the puzzle. The proposed idea is that in order to handle noisy overlapping data we first perform denoising using LS-SVM formulation to KPCA. We obtain the optimal model parameters σ and n_{pcs} for KPCA using the Model selection criterion based on Distance Distributions (MDD). This technique works well for both low and high dimensional datasets as depicted in [1]. After denoising, we cluster the data using the same model parameter σ as obtained from MDD along with user provided number of clusters k to obtain groups with better cluster generalizations. This two-step process is depicted in Fig 2.

We illustrate the various steps taken through an example synthetic dataset comprising 4 Gaussians with 800 points for each Gaussian. We add random Gaussian noise with $\mu = 0$ and $\sigma = 1.3$ to the data

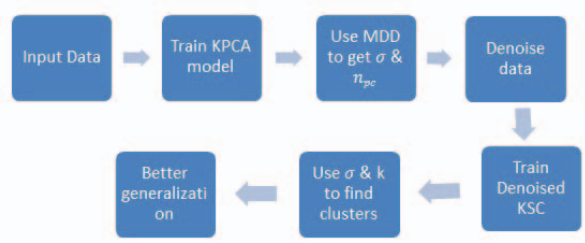
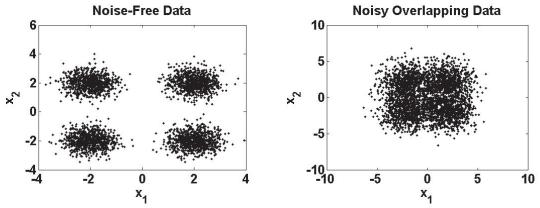


Fig. 2: Two step-process for clustering noisy data

to create noisy overlap data. The noise-free and the noisy datasets are showcased in Figure 3.



(a) Noise-Free Data (b) Noisy Overlap Data

Fig. 3: Synthetic 4 Gaussian datasets

We now select training and validation set to perform denoising using KPCA with MDD model selection criterion. The training ($\mathcal{D}_{tr}$) and validation ($\mathcal{D}_v$) set are selected by maximizing the quadratic Rényi entropy criterion and the cardinality of these sets i.e. N_{tr} and N_{valid} is smaller than total number of points N in the dataset. The training and validation set before and after denoising are illustrated in Figure 4.

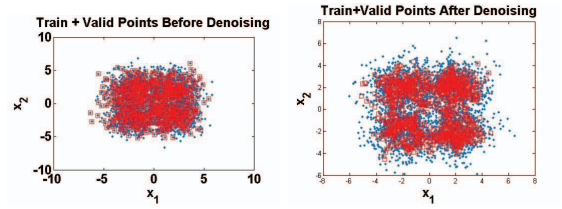


Fig. 4: Train and Validation set coloured in red and original data coloured in blue. Outlook of these sets before and after denoising with KPCA using MDD model selection criterion.

The KPCA technique with MDD optimizes for the model parameter σ and the number of principal components n_{pc} and works well in case of low dimensional datasets as shown in [1]. The MDD selection criterion selects σ and n_{pc} corresponding to which the structural information I_s is maximum. As mentioned earlier in Section III, it only selects those components corresponding to which the difference between the eigenvalues of the original data and the noise distribution is greater than zero. For our noisy 4

Gaussian dataset the MDD selection criterion selects $n_{pc} = 72$ as shown in Figure 5.

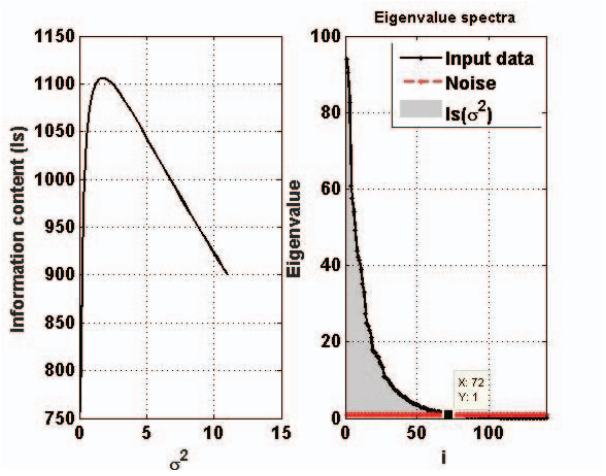
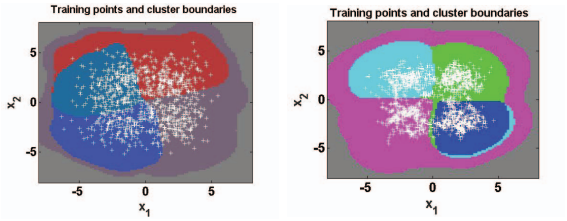


Fig. 5: Structural information (I_s) captured by the right value of σ and optimal number of components n_{pc} obtained from the eigenvalue spectrum.

Now, we use the denoised training set to build the KSC model. We use the kernel parameter σ obtained from KPCA with MDD along with a user-provided number of clusters k to obtain good cluster generalization. *The reason we are able to use the same σ as that obtained from KPCA with MDD for KSC is that MDD selects that σ corresponding to which the structural information in the data is maximal.* Thus, it is able to capture the inherent cluster structure present in the data when provided with the right number of clusters k. Figure 6 illustrates the cluster generalizations corresponding to a normal KSC and the denoised KSC.



(a) KSC Generalization (b) Denoised KSC Result

Fig. 6: Generalization for normal and denoised KSC

VI. Experiments and Analysis

A. Experimental Setup

We conducted experiments on several synthetic and real-world datasets which are available at [25]. We provide a brief description of these datasets in Table I. The cluster memberships of some of these datasets are known beforehand while it is not known for others. So, we use external quality matrix adjusted rand index (ARI) [24] when groundtruth is present and internal clustering quality metrics like the widely used silhouette (Sil) index and the Davies Bouldin (DB) index as described in [24]. Larger the values of ARI, Sil better the clustering quality and lower the value of DB better the clustering quality.

Dataset	Points	Dimensions	Clusters
A1	3000	2	20
Aggregation	788	2	7
D31	3100	2	31
Europe	169308	2	-
Glass	214	9	7
Iris	150	4	3
Mopsi Location Finland (MF)	13467	2	-
Mopsi Location Joensuu (MJ)	6014	2	-
S1	5000	2	15
S2	5000	2	15
S3	5000	2	15
S4	5000	2	15
Seeds	210	7	-
Wdbc	569	32	2
Wine	178	13	3
Yeast	1484	8	10

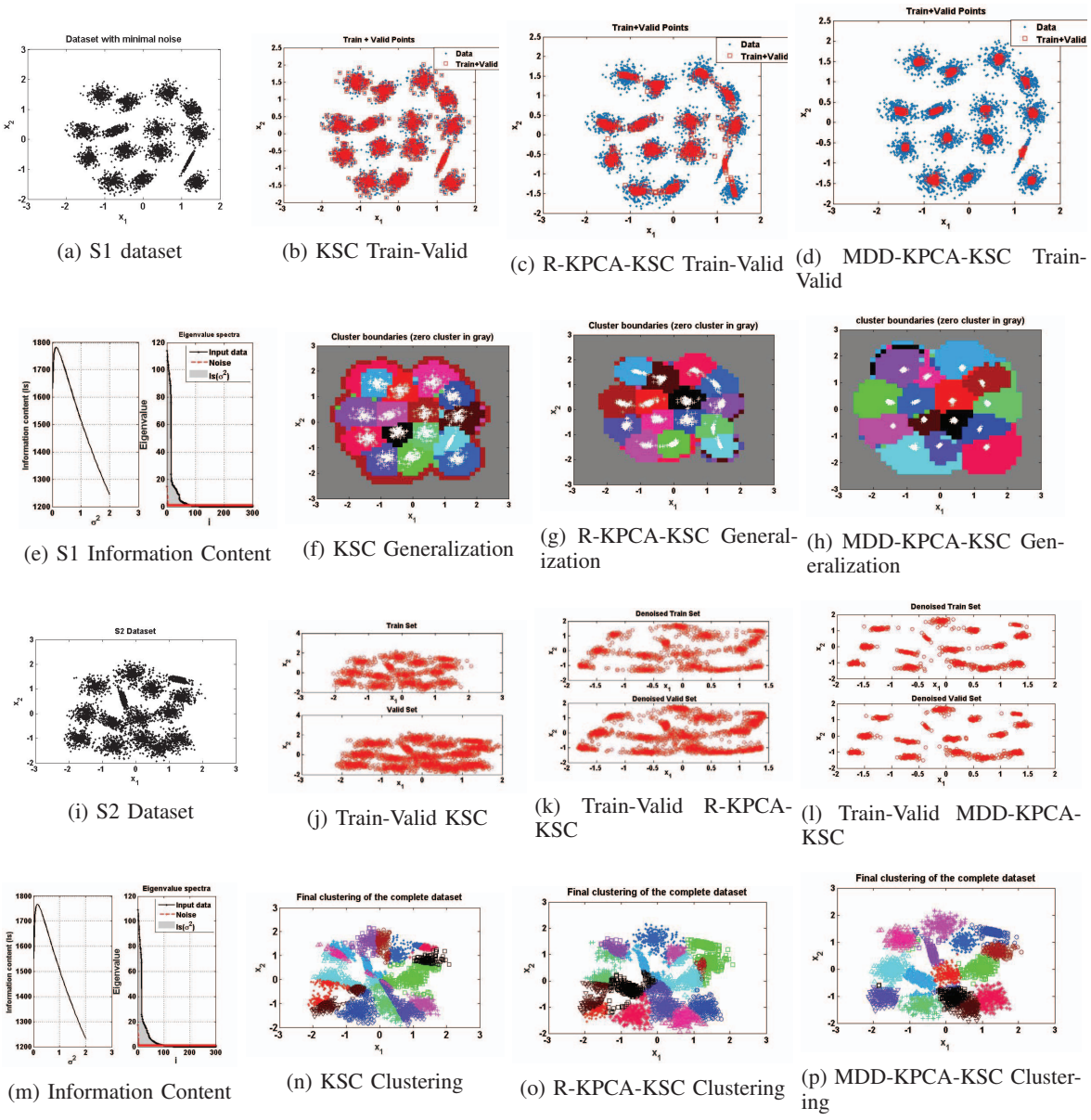
TABLE I: Experiments were performed on these datasets. Here '-' means number of clusters are not known previously.

We compare our proposed approach with normal KSC methodology and KSC technique combined with KPCA. For this approach we use the number of components obtained from KPCA with MDD criterion but tune the σ parameter by optimizing the reconstruction error of KPCA (R-KPCA). We refer to this technique as R-KPCA-KSC for further reference and refer to the propose approach as MDD-KPCA-KSC.

From Table I it can be observed that we perform extensive comparison on low dimensional datasets to obtain good visualization and test the applicability of MDD-KPCA technique. In case of datasets where the optimal number of clusters is not known beforehand, we tune the clustering algorithms by means of BAF [11] criterion. For datasets whose size was less than 1000 points we use 40% of the data for training and another 40% for validation. However, for larger datasets we use a maximum of 15% of the data for training and validation. All the experiments were performed on a 2.4 GHz Core 2 Duo, 12 Gb RAM machine using MATLAB 2011a.

B. Experimental Results

We illustrate the efficiency and effectiveness of our proposed MDD-KPCA-KSC technique on S1, S2, S3 and S4 datasets in Figures 7 and 8. These datasets obtained from [25] depict that as the amount of noise that causes the overlap in the data to increase, the normal KSC and R-KPCA-KSC technique start to deteriorate. However, the MDD-KPCA-KSC technique still provides good cluster generalization.



(a) S1 dataset (b) KSC Train-Valid (c) R-KPCA-KSC Train-Valid (d) MDD-KPCA-KSC Train-Valid

(e) S1 Information Content (f) KSC Generalization (g) R-KPCA-KSC Generalization (h) MDD-KPCA-KSC Generalization

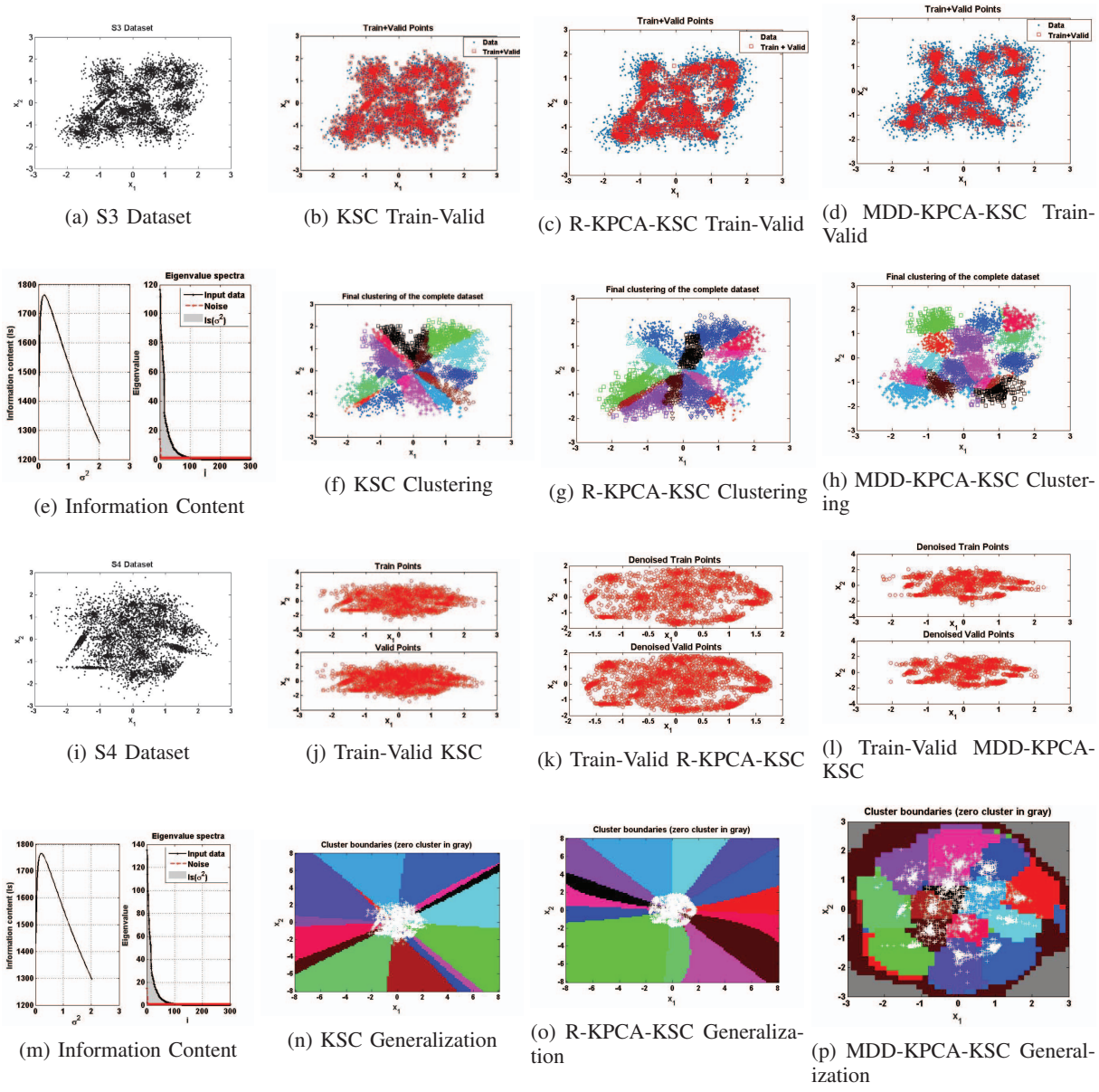
(i) S2 Dataset (j) Train-Valid KSC (k) Train-Valid R-KPCA-KSC (l) Train-Valid MDD-KPCA-KSC

(m) Information Content (n) KSC Clustering (o) R-KPCA-KSC Clustering (p) MDD-KPCA-KSC Clustering

Fig. 7: Comparison of KSC, R-KPCA-KSC and proposed MDD-KPCA-KSC technique on S1 and S2 datasets.

We can observe from Figure 7 that KSC technique works well for S1 dataset as observed from Figure 7f however the clustering results become poor for S2 dataset where KSC finds linear decision boundaries as shown in Figure 7n. The R-KPCA-KSC technique can handle small amount of noise as observed from Figure 7o. However, as the amount of noise increases as shown in Figure 8, the R-KPCA-KSC technique fails as highlighted in Figures 8g and 8o. The R-KPCA-KSC method starts to discover nearly linear decision boundaries as the amount of overlap increases. But the MDD-KPCA-KSC technique is efficient and discovers non-linear decision boundaries with good generalization as observed from Figures 7h, 7p, 8h and 8p. The grey zone in Figures 7f, 7g, 7h and 8p represents the zero cluster or the part for which the model cannot generalize and provides no cluster label.

Table II showcases the clustering results on various synthetic and real-world datasets. We can observe that the MDD-KPCA-KSC technique generally works better than KSC and R-KPCA-KSC technique. For datasets like D31, Glass, Wine, A1, S2, S3, S4, Seeds and Wdbc, the MDD-KPCA-KSC technique easily outperforms KSC and R-KPCA-KSC methods thereby



(a) S3 Dataset (b) KSC Train-Valid (c) R-KPCA-KSC Train-Valid (d) MDD-KPCA-KSC Train-Valid

(e) Information Content (f) KSC Clustering (g) R-KPCA-KSC Clustering (h) MDD-KPCA-KSC Clustering

(i) S4 Dataset (j) Train-Valid KSC (k) Train-Valid R-KPCA-KSC (l) Train-Valid MDD-KPCA-KSC

(m) Information Content (n) KSC Generalization (o) R-KPCA-KSC Generalization (p) MDD-KPCA-KSC Generalization

Fig. 8: Comparison of KSC, R-KPCA-KSC and proposed MDD-KPCA-KSC technique on S3 and S4 datasets.

indicating that it can make the clustering process more efficient in presence of noisy data. However, for datasets like S1 and Aggregation where the clusters are well-separated it can be observed from Table II that the KSC technique is efficient and identifies good quality clusters.

VII. Conclusion

In this paper we proposed a two-step process to cluster noisy data where noise causes overlap in the data. We proposed a combination of KPCA with MDD model selection criterion as it retains the structural information in the data which can then be utilized by the KSC technique to identify good quality clusters with better generalization capability for noisy data. An important aspect to work on in future would be to scale the proposed methodology to larger datasets using techniques proposed in [1] and [11].

Acknowledgments

Co-authors at KU Leuven are supported by EU: ERC AdG A-DATADRIVE-B (290923), Research Council KUL: GOA/10/-/09 MaNet , CoE PFV/10/002 (OPTEC), BIL12/11T; PhD/Postdoc grants-Flemish Government; FWO: projects: G.0377.12 (Structured systems), G.088114N (Tensor based data similarity); PhD/Postdoc grants; IWT: projects: SBO POM (100031); PhD/Postdoc grants: PDM/15/147; iMinds Medical Information Technologies SBO 2014-Belgian Federal Science Policy Office: IUAP P7/19 (DYSCO, Dynamical systems, control and optimization, 2012-2017).

		KSC			R-KPCA-KSC			MDD-KPCA-KSC		
Groundtruth Available										
Dataset	k	ARI	Sil	DB	ARI	Sil	DB	ARI	Sil	DB
Aggregation	7	0.558	0.310	**0.692**	0.805	0.625	0.552	**0.903**	0.626	**0.540**
D31	31	0.553	0.299	0.855	0.599	0.304	0.844	**0.693**	**0.333**	**0.834**
Glass	7	0.242	0.238	3.224	0.298	0.177	1.480	**0.342**	**0.525**	**1.325**
Iris	3	0.812	0.483	1.187	**0.844**	0.527	**0.752**	0.841	**0.564**	0.920
Wine	3	0.434	0.206	2.072	0.482	0.366	**1.165**	**0.510**	**0.390**	1.429
Yeast	10	0.645	**0.765**	1.046	0.662	0.747	1.279	**0.677**	0.723	**0.981**

		Groundtruth Unavailable					
Dataset	Optimal k	Sil	DB	Sil	DB	Sil	DB
A1	20	0.257	1.322	0.573	0.791	**0.626**	**0.536**
Europe	3	0.312	1.534	0.555	**0.874**	**0.632**	1.122
MF	8	0.777	1.335	**0.781**	1.155	0.6947	**0.7393**
MJ	3	0.880	1.156	**0.885**	**0.960**	0.881	1.543
S1	15	0.877	**0.308**	0.877	0.316	0.877	0.320
S2	15	0.311	0.951	0.455	0.586	**0.744**	**0.432**
S3	15	0.277	1.077	0.289	0.877	**0.605**	**0.550**
S4	15	0.185	1.328	0.217	1.280	**0.521**	**0.695**
Seeds	3	0.130	2.704	0.257	1.707	**0.568**	**0.746**
Wdbc	3	0.415	2.646	0.182	2.666	**0.429**	**2.350**

TABLE II: Comparison of KSC, R-KPCA-KSC and MDD-KPCA-KSC techniques on various synthetic and real-world datasets. For datasets where groundtruth is not known we tune for optimal number of clusters k.

REFERENCES

[1] Varon, C., Alzate, C. and Suykens, J.A.K. Noise level estimation for model selection in kernel PCA denoising, *IEEE Transactions on Neural Networks and Learning Systems*, 2015, *26(11)*, 2650-2663.

[2] Ng, A.Y., Jordan, M.I. and Weiss, Y. On spectral clustering: analysis and an algorithm, *In proceedings of the Advances in Neural Information Processing Systems*; Dietterich, T.G., Becker, S., Ghahramani, Z., editors, MIT Press: Cambridge, MA, 2002; pp. 849-856.

[3] von Luxburg, U. A tutorial on Spectral clustering, *Statistics and Computing*, 2007, *17(4)*, pp. 395-416.

[4] Chung, F.R.K. Spectral Graph Theory, *American Mathematical Society*, 1997.

[5] Shi, J. and Malik, J. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Intelligence*, 2000, *22(8)*, pp. 888-905.

[6] Bensmail, H., Celeux, G., Raftery, A.E. and Robert, C.P. Inference in model-based cluster analysis, *Statistics and Computing*, 1997, *7(1)*, pp. 1-10.

[7] Meila, M. and Shi, J. Learning segmentation by random walks, *In Proc. of Neural Information Processing Systems*, 2000, pp. 873-879.

[8] Alzate, C. and Suykens, J.A.K. Multiway spectral clustering with out-of-sample extensions through weighted kernel PCA. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2010, *32(2)*, pp. 335-347 .

[9] Alzate, C. and Suykens, J.A.K. Sparse kernel spectral clustering models for large-scale data analysis. *Neurocomputing*, 2011, *74(9)*, 1382-1390.

[10] Langone, R., Mall, R. and Suykens, J.A.K. Soft Kernel Spectral Clustering. *In Proc. of IEEE IJCNN*, 2013, pp. 1-8.

[11] Mall, R., Langone, R. and Suykens, J.A.K. Kernel Spectral Clustering for Big Data Networks, *Entropy*, 2013, *15(5)*, 1567-1586.

[12] Mall, R., Langone, R. and Suykens, J.A.K. FURS:Fast and Unique Representative Subset selection retaining large scale community structure, *Social Network Analysis and Mining*, 2013, *3(4)*, 1075-1095.

[13] Mall, R., Langone, R. and Suykens, J.A.K. Self-Tuned Kernel Spectral Clustering for Large Scale Networks, *IEEE International Conference on Big Data (IEEE BigData)*, 2013, pp. 385-393.

[14] Haghtalab, N. Clustering in the presence of noise, *Master Thesis*, 2013.

[15] Ben-David, S. and Haghtalab, N. Clustering in the presence of background noise, *In Proc. of* 31st *International Conference on Machine Learning*, 2014, pp. 280-288.

[16] Li, Z., Liu, J., Chen, S. and Tang, X. Noise Robust Spectral Clustering, *In Proc. of* 11th *IEEE International Conference on Computer Vision*, 2007, pp. 1-8.

[17] Bensmail, H. and Meulman, J.J. Model-based clustering with noise: Bayesian inference and information, *Journal of Classification*, 2003, *20*, pp. 49-76.

[18] Suykens, J.A.K., Van Gestel, T., De Brabanter, J., De Moor, B. and Vandewalle, J. Least Squares Support Vector Machines, 2002, *World Scientific, Singapore*.

[19] Schölkopf, B., Smola, A. and Müller, K.R. Nonlinear component analysis as a kernel eigenvalue problem, *Neural Computation*, 1998, *10(5)*, pp. 1299-1319.

[20] Mika, S., Schölkopf, B., Smola, A. J., Müller, K. R., Scholz, M. and Rätsch, G. Kernel PCA and de-noising in feature spaces, *In Advances in Neural Information Processing Systems*, 1999, pp. 536-542.

[21] Kwok, J.Y. and Tsang, I.W. The pre-image problem in kernel methods, *IEEE Trans. on Neural Networks*, 2004, *15*, pp. 1517-1525.

[22] Girolami, M. Orthogonal series density estimation and the kernel eigenvalue problem. *Neural Computation*, 2002, *14(3)*, pp. 1000-1017.

[23] Grant, M., Boyd and S. CVX: Matlab software for disciplined convex programming. 2010, *http://cvxr.com/cvx*.

[24] Rabbany, R., Takaffoli, M., Fagnan, J., Zaiane, O.R. and Campello R.J.G.B. Relative Validity Criteria for Community Mining Algorithms, 2012, *International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*, pp. 258-265.

[25] *http://cs.joensuu.fi/sipu/datasets/*