Optimal Reduced Sets for Sparse Kernel Spectral Clustering

Raghvendra Mall
ESAT/SCD
Kasteelpark Arenberg 10, bus 2446
3001 Heverlee
Email: rmall@esat.kuleuven.be

Siamak Mehrkanoon
and Rocco Langone
ESAT/SCD
Kasteelpark Arenberg 10, bus 2446
3001 Heverlee

Johan A.K. Suykens
ESAT/SCD
Kasteelpark Arenberg 10, bus 2446
3001 Heverlee
Email: johan.suykens@esat.kuleuven.be

***Abstract*—Kernel spectral clustering (KSC) solves a weighted kernel principal component analysis problem in a primal-dual optimization framework. It results in a clustering model using the dual solution of the problem. It has a powerful out-of-sample extension property leading to good clustering generalization w.r.t. the unseen data points. The out-of-sample extension property allows to build a sparse model on a small training set and introduces the first level of sparsity. The clustering dual model is expressed in terms of non-sparse kernel expansions where every point in the training set contributes. The goal is to find reduced set of training points which can best approximate the original solution. In this paper a second level of sparsity is introduced in order to reduce the time complexity of the computationally expensive out-of-sample extension. In this paper we investigate various penalty based reduced set techniques including the Group Lasso, L_0, $L_1 + L_0$ penalization and compare the amount of sparsity gained w.r.t. a previous L_1 penalization technique. We observe that the optimal results in terms of sparsity corresponds to the Group Lasso penalization technique in majority of the cases. We showcase the effectiveness of the proposed approaches on several real world datasets and an image segmentation dataset.**

I. Introduction

Clustering algorithms are widely used tools in fields like data mining, machine learning, graph compression and many other tasks. The aim of clustering is to divide data into natural groups present in a given dataset. Clusters are defined such that the data present within the group are more similar to each other in comparison to the data between clusters. Spectral clustering methods [1], [2] and [3] are generally better than the traditional k-means techniques. A new Kernel Spectral Clustering (KSC) algorithm based on weighted kernel PCA formulation was proposed in [4]. The method was based on a model built 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. The KSC methodology has been extensively applied for task of data clustering [4], [5], [6], [7] and community detection [8], [9], [10] in large scale networks.

The data points are projected to the eigenspace and the projections are expressed in terms of non-sparse kernel expansions. In [5], a method to sparsify the clustering model was proposed by exploiting the line structure of the projections when the clusters are well formed and well separated. However, the method fails when the clusters are overlapping and for real world datasets where the projections in the eigenspace do not follow a line structure as mentioned in [6]. In [6], the authors used an $L_2 + L_1$ penalization to produce a reduced set to approximate the original solution vector. Although the authors propose it as an $L_2 + L_1$ penalization technique, the actual penalty on the weight vectors is L_1 penalty and the loss function is squared loss function and hence the name. Therefore in this paper we refer to the previous proposed approach as L_1 penalization technique. It is well known that the L_1 regularization introduces sparsity as shown in [11]. However, the resulting reduced set is neither the sparsest nor the most optimal w.r.t. the quality of clustering for the entire dataset. In this paper we propose alternative penalization techniques like Group Lasso [12] and [13], L_0 and $L_1 + L_0$ penalizations. The Group Lasso penalty is ideal for clusters as it results in groups of relevant data points. The L_0 regularization calculates the number of non-zero terms in the vector. The L_0-norm results in a non-convex and NP-hard optimization problem. We modify the convex relaxation of L_0-norm based iterative sparsification procedure introduced in [14] for classification. We apply it to obtain the optimal reduced sets for sparse kernel spectral clustering.

The main advantage of these sparse reductions is that it results in much simpler and faster predictive models. It allows to reduce the time complexity for the computationally expensive out-of-sample extensions and also reduces the memory requirements for building the test kernel matrix.

II. Kernel Spectral Clustering

We first provide a brief description of the kernel spectral clustering methodology according to [4].

A. Primal-Dual Weighted Kernel PCA framework

Given a dataset $\mathcal{D} = \{x_i\}_{i=1}^{N_{tr}}$, $x_i \in \mathbb{R}^d$, the training points are selected by maximizing the quadratic Rènyi criterion as depicted in [6], [15] and [18]. This introduces the first level of sparsity by building the model on a subset of the dataset. Here x_i represents the i^{th} training data point and the training set is represented by X_{tr}. The number of data points in the training set is N_{tr}. Given $\mathcal{D}$ and the number of clusters k, the primal problem of the spectral clustering via weighted kernel

PCA is formulated as follows [4]:

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

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 $N_{tr} \times n_h$ 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 $N_{tr} \ll N$ i.e. the number of points in the training set is much 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 for example by using the radial basis function (RBF) kernel. The clustering model is then represented by:

$$e_i^{(l)} = w^{(l)\intercal}\phi(x_i) + b_l, i = 1,\ldots,N_{tr}, \tag{2}$$

where $\phi : \mathbb{R}^d \to \mathbb{R}^{n_h}$ is the mapping to a high-dimensional feature space n_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:

$$D_\Omega^{-1} M_D \Omega \alpha^{(l)} = \lambda_l \alpha^{(l)}, \tag{3}$$

where M_D is the centering matrix which is defined as $M_D = \text{I}_{N_{tr}} - (\frac{(1_{N_{tr}} 1_{N_{tr}}^\intercal D_\Omega^{-1})}{1_{N_{tr}}^\intercal D_\Omega^{-1} 1_{N_{tr}}})$. The $\alpha^{(l)}$ are the dual variables and the positive definite kernel function $K : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ plays the role of similarity function. This dual problem is closely related to the random walk model as shown in [4].

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:

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

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

The original KSC formulation in [4] works well assuming piece-wise constant eigenvectors and using the line structure of the projections of the validation points in the eigenspace. It uses an evaluation criterion called Balanced Line Fit (BLF) for model selection i.e. for selection of k and σ for the RBF function. However, this criterion works well only in case of well separated clusters. So, we use the Balanced Angular Fit (BAF) criterion proposed in [8] and [7] for cluster evaluation. This criterion works on the principle of angular similarity and is efficient when the clusters are either well separated or overlapping. The BAF criterion varies from [-1, 1] and higher values are better for a given value of k.

III. Sparse reductions to KSC model

A. Related Work

In classical spectral clustering one needs to store the $N \times N$ matrix where N is the total number of points in the dataset. One then has to perform an eigen-decomposition of this matrix. The time complexity of this eigen-decomposition is $O(N^3)$. In the case of KSC we can build the training model using a training set ($N_{tr} \ll N$) and use the out-of-sample extension property to predict the cluster affiliation for unseen data. This leads to the first level of sparsity. However, the projections of the data points in the eigenspace are expressed in terms of non-sparse kernel expansions as reflected in (4). This non-sparsity is a result of the KKT condition: $w^{(l)} = \sum_{i=1}^{N_{tr}} \alpha_i^{(l)}\phi(x_i)$. Here $w^{(l)}$ represents the optimal representation of the primal weight vectors and comprises of linear combination of the mapped training data points in the feature space. When using a universal kernel like the RBF kernel the feature space comprises infinite dimensions. Thus, we first create an explicit feature map using the Nyström approximation as in [16] and [17]. This explicit feature map is created using the training points X_{tr} and the feature mapping becomes: $\phi : \mathbb{R}^d \to \mathbb{R}^{N_{tr}}$.

The objective is to find a reduced set of training points $\mathcal{RS} = \{\tilde{x}_i\}_{i=1}^R$ such that it approximates $w^{(l)}$ by a new weight vector $\tilde{w}^{(l)} = \sum_{i=1}^R \beta_i^{(l)}\phi(\tilde{x}_i)$ while minimizing the reconstruction error $||w^{(l)} - \tilde{w}^{(l)}||_2^2$ where $\tilde{x}_i$ is the i^{th} point in the reduced set $\mathcal{RS}$ whose cardinality is R. In [5], it was shown by the authors that if the reduced set $\mathcal{RS}$ is known then the $\beta^{(l)}$ co-efficients can be obtained by solving the linear system:

$$\Omega^{\psi\psi}\beta^{(l)} = \Omega^{\psi\phi}\alpha^{(l)}, \tag{5}$$

where $\Omega_{mn}^{\psi\psi} = K(\tilde{x}_m,\tilde{x}_n)$, $\Omega_{mi}^{\psi\phi} = K(\tilde{x}_m,x_i)$, $m,n = 1,\ldots,R, i = 1,\ldots,N_{tr}$ and $l = 1,\ldots,k-1$.

In the past literature including the works in [5] and [6], it was shown that this reduced set can be built by selecting points whose projections in the eigenspace occupy certain positions or by using an L_1 penalization. The first method works only when the clusters are well formed and well separated and cannot be generalized to real world datasets. The second method using L_1 penalization cannot introduce significant sparsity. In this paper, we investigate other penalization techniques including the Group Lasso [12] and [13], L_0 and $L_1 + L_0$ penalizations.

B. Group Lasso Penalization

The Group Lasso was first proposed for regression in [12] where it solves the convex optimization problem:

$$\min_{\beta \in \mathbb{R}^p} \quad \|y - \sum_{l=1}^{L} X_l \beta_l\|_2^2 + \lambda \sum_{l=1}^{L} \sqrt{\rho_l} \|\beta_l\|_2,$$

where the $\sqrt{\rho_l}$ accounts for the varying group sizes, $\| \, . \, \|_2$ is the Euclidean norm. This procedure acts like Lasso [11] at a group level: depending on λ, an entire group of predictors may drop out of the model. We now utilize this to obtain the formulation for our optimization problem as:

$$\min_{\beta \in \mathbb{R}^{N_{tr} \times (k-1)}} \quad \|\Phi^\intercal \alpha - \Phi^\intercal \beta\|_2^2 + \lambda \sum_{l=1}^{N_{tr}} \sqrt{\rho_l} \|\beta_l\|_2, \tag{6}$$

where $\Phi = [\phi(x_1), \ldots, \phi(x_{N_{tr}})]$, $\alpha = [\alpha^{(1)}, \ldots, \alpha^{(k-1)}]$, $\alpha \in \mathbb{R}^{N_{tr} \times (k-1)}$ and $\beta = [\beta_1, \ldots, \beta_{N_{tr}}]$, $\beta \in \mathbb{R}^{N_{tr} \times (k-1)}$. Here $\alpha^{(i)} \in \mathbb{R}^{N_{tr}}$ while $\beta_j \in \mathbb{R}^{k-1}$ and we set $\sqrt{\rho_l}$ as the fraction of training points belonging to the cluster to which the l^{th} training point belongs. By varying the value of λ we control the amount of sparsity introduced in the model as it acts as a regularization parameter. In [13], the authors show that if the initial solutions are $\hat{\beta}_1, \hat{\beta}_2, \ldots, \hat{\beta}_{N_{tr}}$ then if $\|X_l^\intercal (y - \sum_{i \neq l} X_i \hat{\beta}_i)\| < \lambda$, then $\hat{\beta}_l$ is zero otherwise it satisfies: $\hat{\beta}_l = (X_l^\intercal X_l + \lambda / \|\hat{\beta}_l\|)^{-1} X_l^\intercal r_l$ where $r_l = y - \sum_{i \neq l} X_i \hat{\beta}_i$.

Analogous to this, the solution to the group lasso penalization for our problem can be defined as: $\|\phi(x_l)(\Phi^\intercal \alpha - \sum_{i \neq l} \phi(x_i)\hat{\beta}_i)\| < \lambda$ then $\hat{\beta}_l$ is zero otherwise it satisfies: $\hat{\beta}_l = (\Phi^\intercal \Phi + \lambda / \|\hat{\beta}_l\|)^{-1} \phi(x_l) r_l$ where $r_l = \Phi^\intercal \alpha - \sum_{i \neq l} \phi(x_i)\hat{\beta}_i$. The Group Lasso penalization technique can be solved by a blockwise co-ordinate descent procedure as shown in [12]. The time complexity of the approach is $O(\text{maxiter} * k^2 N_{tr}^2)$ where maxiter is the maximum number of iterations specified for the co-ordinate descent procedure and k is the number of clusters obtained via KSC. From our experiments we observed that on an average 10 iterations suffice for convergence.

An important point to remember here is that $\hat{\beta}_l \in \mathbb{R}^{k-1}$ and is a vector. When this $\hat{\beta}_l$ is zero it means that it is equivalent to zero vector or the corresponding l^{th} training point is not part of the reduced set $\mathcal{RS}$. In our experiments, we set the initial value of β as $\hat{\beta}_{ij} = \alpha_{ij} + \mathcal{N}(0,1)$ where $\mathcal{N}(0,1)$ represents Gaussian noise with mean 0 and standard deviation 1.

C. L_0 Penalization

We modify the iterative sparsification procedure for classification as shown in [14] and use it for obtaining the reduced set. The optimization problem ($\mathcal{J}$) which is solved iteratively is formulated as:

$$\begin{aligned} \min_{\beta \in \mathbb{R}^{N_{tr} \times (k-1)}} \quad & \|\Phi^\intercal \alpha - \Phi^\intercal \beta\|_2^2 + \rho \sum_{i=1}^{N_{tr}} \epsilon_i + \|\Lambda . \beta\|_2^2 \\ \text{such that} \quad & \|\beta_i\|_2^2 \leq \epsilon_i, i = 1, \ldots, N_{tr} \\ & \epsilon_i \geq 0, \end{aligned} \tag{7}$$

where Λ is matrix of the same size as the β matrix i.e. $\Lambda \in \mathbb{R}^{N_{tr} \times (k-1)}$. The term $\|\Lambda . \beta\|_2^2$ along with the constraint $\|\beta_i\|_2^2 \leq \epsilon_i$ corresponds to the L_0-norm penalty on β matrix. Λ matrix is initially defined as a matrix of ones so that it gives equal chance to each element of β matrix to reduce to zero. The constraints on the optimization problem forces each element of $\beta_i \in \mathbb{R}^{(k-1)}$ to reduce to zero. This helps to overcome the problem of sparsity per component which is explained in [6]. The ρ variable is a regularizer which controls the amount of sparsity that is introduced by solving this optimization problem.

The optimization problem stated in (6) is a convex Quadratically Constrained Quadratic Programming (QCQP) problem. Its computational complexity is $O(k^3 N_{tr}^3)$ and we solve it iteratively using the CVX software:[20]. We obtain a β matrix as a solution for each iteration such that $\beta_{ij}^{t+1} = \arg\min_\beta \mathcal{J}(\Lambda_{ij}^t)$. For each iteration, the Λ matrix is re-weighted as: $\Lambda_{ij}^t = \frac{1}{\beta_{ij}^t}$, $\forall \; i = 1, \ldots, N_{tr}, j = 1, \ldots, k-1$. It was shown in [14] that this iterative procedure results in a convex approximation to the L_0-norm. But as the L_0-norm is a non-convex problem it results in a local minimum. We stop this iterative procedure when the rate of change of the β matrix is below a threshold such that $\|\beta^{t+1} - \beta^t\|_2^2 / N_{tr} < 10^{-4}$. We then select those indices i for which $\|\beta_i^{t+1}\|_2^2 > 10^{-6}$ and put the corresponding training points in the reduced set $\mathcal{RS}$. In our experiments we observe that the number of iterations required to reach this convergence is usually less than 20.

D. $L_1 + L_0$ Penalization

The $L_1 + L_0$ penalization formulation is quite similar to the formulation of L_1 penalization as defined in [6]. We add an additional regularization matrix Λ on the β matrix and the problem formulation becomes:

$$\begin{aligned} \min_{\beta \in \mathbb{R}^{N_{tr} \times (k-1)}} \quad & \|\Phi^\intercal \alpha - \Phi^\intercal \beta\|_2^2 + \rho \sum_{i=1}^{N_{tr}} \epsilon_i + \|\Lambda . \beta\|_2^2 \\ \text{such that} \quad & |\beta_i| \leq \epsilon_i, i = 1, \ldots, N_{tr} \\ & \epsilon_i \geq 0, \end{aligned} \tag{8}$$

The difference between (7) and (8) is the set of constraints for both the optimization problems. In (8) the constraint $|\beta_i| \leq \epsilon_i$ corresponds to the L_1 penalization.

This problem formulation results in a convex Quadratic Programming (QP) problem due to linear constraints. Its computational complexity is $O(k^3 N_{tr}^3)$. It is also solved iteratively using the CVX software. We initialize Λ matrix as ones and after each iteration we modify each element of Λ^t matrix such that $\Lambda_{ij}^t = \frac{1}{\beta_{ij}^t}$, $\forall \; i = 1, \ldots, N_{tr}$, $j = 1, \ldots, k-1$. We show in the experimental results that this penalization often results similarly to the L_1 penalization outcomes. This suggests that the L_1 penalization is driving the amount of sparsity in this penalization to obtain the reduced set $\mathcal{RS}$.

E. Choice of Tuning parameter

The choice of the right tuning parameter is essential to obtain optimal reduced set. The tuning parameter λ influences the amount of sparsity in the model for the Group Lasso penalization technique while in case of other penalization techniques this is handled by the tuning parameter ρ. Sparsity is defined as $1 - \frac{|\mathcal{RS}|}{N_{tr}}$.

The procedure for selection of this tuning parameter is quite simple. For Group Lasso penalization technique, we obtain the $\lambda_{\max}$ initially which is defined as: $\arg_{\max}\|\phi(x_l)(\Phi^\intercal\alpha - \sum_{i\neq l}\phi(x_i)\hat{\beta}_i)\|$, $\forall\ l = 1,\ldots,N_{tr}$. In order to tune the value of the regularizer λ for varying the amount of sparsity for the reduced set $\mathcal{RS}$, we use different fractional values of $\lambda_{\max}$ as λ. The values of λ are set such that the value of sparsity covers the entire range $[0,1]$ i.e. we vary the value of λ such that there is no sparsity (sparsity$=0$) in the model to the case when there is no data point in the reduced set $\mathcal{RS}$ (sparsity $=1$).

For the L_1, L_0 and L_1+L_0 penalization techniques we have to tune the parameter ρ. To have a fair comparison we use the same range and same values for tuning parameter ρ in case of these techniques. However, the best results for different penalization techniques can occur for different value of tuning parameter ρ. The choice of ρ is again dependent on the amount of sparsity it generates. We aim to select the smallest range of values for ρ such that the value of sparsity covers the entire range $[0,1]$. From our experiments, we observe that the smallest possible range for ρ corresponding to which sparsity varies from $[0,1]$ is $[1,10]$. Thus, we vary the value of ρ in logarithmic steps between the range $[1,10]$ to obtain the optimal reduced sets.

F. Out-of-Sample Extension Time Complexity

For the KSC method [4] we consider the entire dataset as the test set. The cardinality of the entire dataset is N. The computational complexity for the out-of-sample predictions for KSC method is $O(N_{tr}N)$ where $N_{tr} \ll N$. This is because for the out-of-sample extension we need to create the test kernel matrix of size $N_{tr}\times N$. When this kernel matrix is too large to be stored in memory then we divide the test data into chunks such that each chunk can fit in memory. Test cluster membership prediction is then done for each chunk.

For the reduced set based methods we can greatly reduce the computational cost for out-of-sample extensions. Let the cardinality of the reduced set corresponding to the Group Lasso, L_0 and L_1+L_0 penalization methods be R_1, R_2, R_3 respectively. Since these methods introduce sparsity, the amount of sparsity introduced corresponding to these penalization methods can be defined as: R_1/N_{tr}, R_2/N_{tr} and R_3/N_{tr} respectively. The cardinality of the reduced set $\mathcal{RS}$ is much lesser than the size of the training set i.e. $R_i \ll N_{tr}$, $i=1,2,3$. Thus the time complexity for the out-of-sample extension corresponding to the three proposed reduced sets is $O(R_iN)$, $i=1,2,3$. This also reduces the constraint on the memory as the size of the test kernel matrix for the reduced sets becomes $R_i\times N$, $i=1,2,3$ which is much less than the size of the original test kernel matrix ($N_{tr}\times N$).

G. Synthetic Example

We show the results of an experiment on a synthetic dataset using RBF kernel in Figure 1. The dataset consists of 3 overlapping Gaussian clouds in 2-dimensions for a total number of $1,500$ data points. We select 450 data points for training and 600 data points for validation using the quadratic Rènyi entropy criterion.

Figure 1 shows the results on this synthetic dataset corresponding to Group Lasso, L0, L1, L1+L0 penalizations. We vary the regularization parameter λ for Group Lasso and ρ for the other penalization methods. In Figures 1b, 1d, 1f and 1h, the 'o'-shaped, red-bodied black-outlined points correspond to the reduced set. In these Figures the training set is constant but the reduced set changes in accordance to the penalization technique. Since the dataset is synthetic, the groundtruth is known beforehand, the quality of the clusters are evaluated using an external quality metric - Adjusted Rand Index (ARI) as defined in [21]. The ARI metric compares the cluster memberships obtained using the reduced set w.r.t. the groundtruth of the test points and higher value of ARI signifies better match between the cluster memberships.

From Figures 1b, 1c and 1j, we observe that the best result for Group Lasso penalization occurs when the regularization parameter $\lambda = 0.8\lambda_{\max}$. It introduces maximal amount of sparsity (sparsity = 0.9933, cardinality of reduced set is 4) while obtaining the best generalization (ARI = 0.56). The best result for L_0 penalization technique takes place for $\rho = 10$ and produces a sparse reduced set (sparsity = 0.9911). But the generalization (ARI = 0.478) is not as good as Group Lasso. This can be observed from Figures 1d, 1e and 1k.

The L_1 and L_1+L_0 penalization techniques produce the same generalization and sparsity for several values of regularizer ρ as depicted in Figures 1k and 1l. Figure 1l indicates that as we increase the value of ρ the amount of sparsity increases. However, when we increase the value of ρ from 8 to 10, then the quality of the clusters decrease as observed from Figure 1k. The best result for the L_1 and L_1+L_0 penalization techniques (sparsity = 0.93, ARI = 0.44) is worse than Group Lasso and L_0 penalization technique both in terms of quality (ARI) and amount of sparsity introduced.

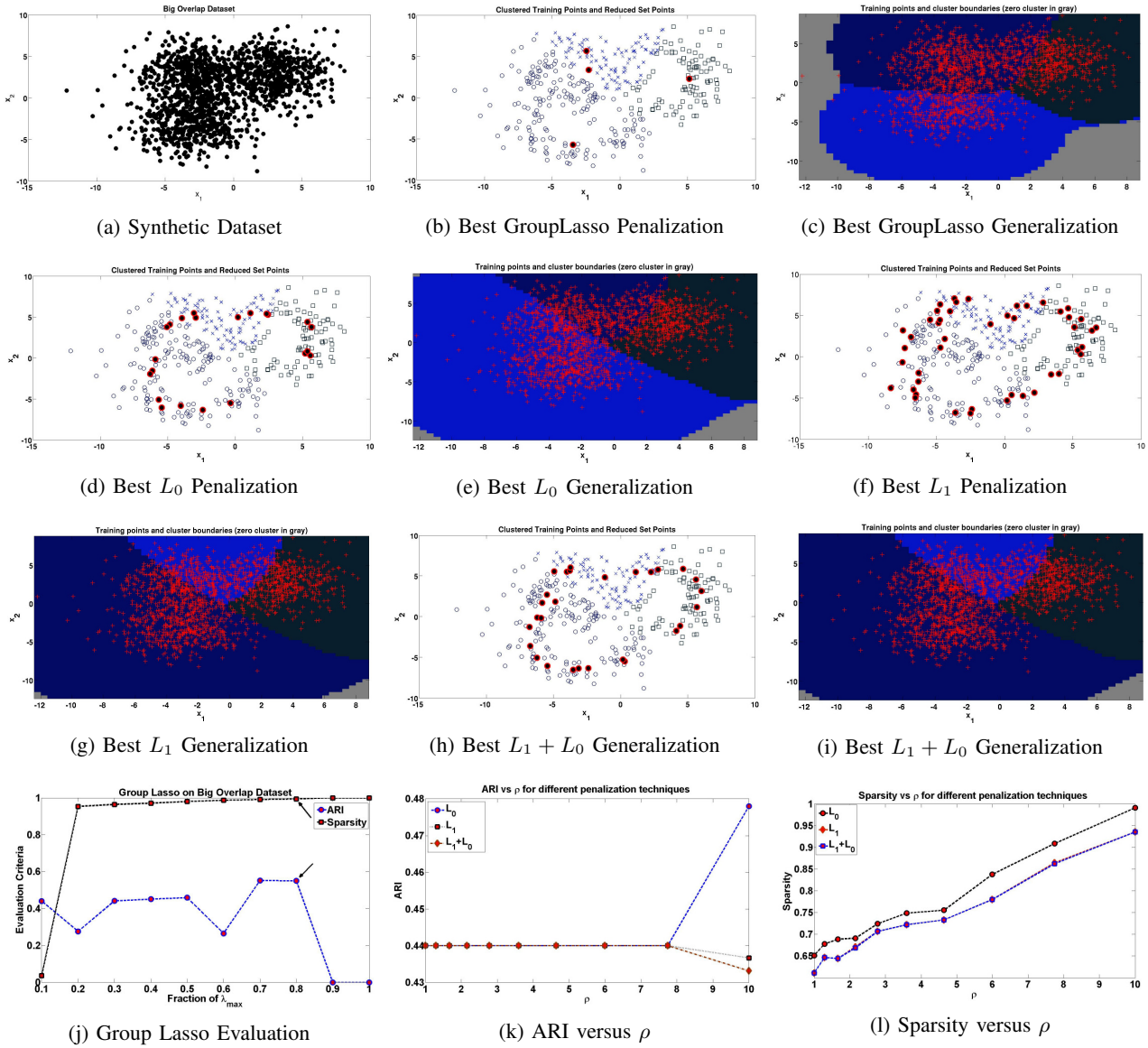
IV. Experiments on Real World Datasets

A. Experimental Setup

We conducted experiments on several real world datasets which are available at [22]. We provide a brief description of these datasets in Table I. Since the cluster memberships of these datasets are not known beforehand, we use internal clustering quality metrics for evaluation of the resulting clusters. These internal quality metrics include the widely used silhouette (sil) index and the Davies Bouldin (db) index as described in [21]. Larger the values of sil better the clustering quality and lower the value of db better the clustering quality.

Dataset	Points	Dimensions	Clusters
Breast	699	9	2
Bridge	4096	16	-
Europe	169308	2	-
Glass	214	9	7
Iris	150	4	3
Mopsi Location Finland	13467	2	-
Mopsi Location Joensuu	6014	2	-
Thyroid	215	5	2
Wdbc	569	32	2
Wine	178	13	3
Yeast	1484	8	10

TABLE I: Real world datasets. Here '-' means the number of clusters are not known previously



(a) Synthetic Dataset (b) Best GroupLasso Penalization (c) Best GroupLasso Generalization

(d) Best L_0 Penalization (e) Best L_0 Generalization (f) Best L_1 Penalization

(g) Best L_1 Generalization (h) Best $L_1 + L_0$ Generalization (i) Best $L_1 + L_0$ Generalization

(j) Group Lasso Evaluation (k) ARI versus ρ (l) Sparsity versus ρ

Fig. 1: Results on Synthetic Dataset corresponding to the reduced sets obtained for different penalization techniques.

	Group Lasso					L_0 **Penalization**					L_1 **Penalization**					$L_1 + L_0$ **Penalization**				
Dataset	sil	db	Sparsity	λ	Time(secs)	sil	db	Sparsity	ρ	Time(secs)	sil	db	Sparsity	ρ	Time(secs)	sil	db	Sparsity	ρ	Time(secs)
Breast	0.6824	0.85	**99.5%**	$0.7\lambda_{max}$	**0.99**	0.6898	0.833	96.6%	7.74	19.5	0.6898	0.833	96.6%	7.74	6.7	0.6898	0.833	96.6%	7.74	20.1
Bridge	0.423	1.436	**99.0%**	$0.7\lambda_{max}$	**34.8**	**0.596**	1.3	97.1%	4.642	701.0	0.559	1.72	98.5%	5.995	238.4	0.559	1.72	98.5%	5.995	702.4
Europe	**0.437**	2.1	**99.9%**	$0.9\lambda_{max}$	**4509**	0.352	**1.145**	99.75%	10	210,512	0.352	1.148	99.7%	10	61,456	0.352	1.148	99.7%	10	220,456
Glass	0.33	3.133	14.0%	$1.0\lambda_{max}$	**0.12**	0.3223	1.913	93.75%	2.155	2.42	0.408	**1.813**	**96.9%**	4.64	0.68	**0.547**	2.037	89.1%	3.593	2.64
Iris	0.611	1.333	**93.33%**	$0.9\lambda_{max}$	**0.03**	0.605	1.323	84.45%	2.78	1.02	0.6184	1.3063	86.67%	3.598	0.28	0.6184	1.3063	86.67%	3.598	1.03
Mopsi Location Finland	0.7219	1.2735	**99.7%**	$0.6\lambda_{max}$	**301.2**	**0.7976**	**1.142**	99.6%	10	6,586	0.7946	1.158	99.5%	10	2,315	0.7946	1.158	99.5%	10	6,671
Mopsi Location Joensuu	**0.911**	**0.64**	**99.5%**	$0.7\lambda_{max}$	**80.2**	0.88	0.67	96.5%	10	1,720	0.8814	0.6684	95.5%	10	612	0.8814	0.6684	95.5%	10	1,734
Thyroid	**0.6345**	**0.844**	**98.4%**	$6\lambda_{max}$	**0.13**	0.538	1.04	93.75%	5.995	2.48	0.538	1.04	93.75%	5.995	0.7	0.538	1.04	93.75%	5.995	2.7
Wdbc	0.5585	1.304	96.5%	$0.9\lambda_{max}$	**0.78**	0.56	1.303	97.6%	5.995	13.2	0.56	1.303	97.6%	5.995	4.11	0.56	1.303	97.6%	5.995	14.3
Wine	**0.291**	**1.943**	5.6%	$1.0\lambda_{max}$	**0.05**	0.29	1.96	85.0%	1.668	1.28	0.29	1.96	86.8%	2.154	0.31	0.29	1.96	86.8%	2.154	1.3
Yeast	**0.81**	2.7	97.76%	$0.9\lambda_{max}$	**4.01**	0.258	2.3	**97.9%**	7.74	79.1	0.2637	**2.2**	83.0%	10	25.2	0.2775	2.214	83.37%	10	82.32

TABLE II: Comparison of the sparsest feasible solution for the different penalization methods. Here we highlight the unique best results i.e. the best results are highlighted if they correspond to a single penalization method. In most of the cases the L_1 and $L_1 + L_0$ penalization result in the same sparsest solution.

B. Experimental Results

Table II showcases the sparsest feasible solution for the different penalization methods and evaluates it on quality

metrics like sil and db. We represent the amount of sparsity as percentage of sparsity rather than fractions (i.e. fraction of sparsity $\times 100$). By feasible solutions we refer to the cases when the cardinality of the reduced set $|\mathcal{RS}| > 0$. For higher values of regularization parameters the cardinality of the reduced set can become zero and these solutions are not part of the feasible solutions.

From Table II we observe that the Group Lasso penalization introduces the maximum amount of sparsity in general and in the obtained cases the cluster quality by corresponding reduced set is better than the other penalization methods. The Group Lasso penalization performs best for the Europe, Mopsi Location Joensuu (MLJ), Thyroid, Wine and Yeast datasets. We also observe that the proposed L_0 penalization technique generally results in sparser solution than the L_1 penalization method. In some cases it also results in better quality clusters, for example in the cases of Bridge, Europe and Mopsi Location Finland (MLF) datasets. An important observation is that the results corresponding to the proposed $L_1 + L_0$ penalization are quite similar to the results of L_1 penalization. This suggests that the L_1 penalization dominates in each step of the iterative sparsification procedure for the $L_1 + L_0$ penalization method.

C. Real World Image Segmentation Dataset

We also perform an image segmentation experiment such that each pixel is transformed into a histogram and the χ^2 distance is used in the RBF kernel with bandwidth σ_χ as shown in Figure 2. The total number of pixels is $154,401$ (321×481). The training set consists of $N_{tr} = 7,500$ pixels and the validation set consists of $10,000$ pixels. Both these set are selected by maximizing the quadratic Rènyi entropy. After validation we obtain $k = 3$ and kernel parameter $\sigma_\chi = 2.807$. We performed this experiment on a 2.4 GHz Core 2 Duo, 12 Gb RAM machine using MATLAB 2012b.

The Group Lasso based penalization method reduces the reduced set to just two data points for $\lambda = 0.7\lambda_{max}$ and still has the best $sil = 0.294$ value as shown in Figure 2a. In KSC [4] there is a possibility of a null cluster i.e. the cluster which is beyond the generalization boundary of all the clusters. The Group Lasso penalization technique produces 2 points in the reduced set, one corresponding to each cluster. The third cluster corresponds to the null cluster. Hence, it results in good segmentation as observed from Figure 2b.

The L_0 penalization also results in a highly sparse model (sparsity = 0.9645) and has the smallest $db = 0.141$ value as observed from Figure 2c.

The results corresponding to L_1 and $L_1 + L_0$ penalization techniques are same for this image dataset. Thus, we only show the result for L_1 penalization technique in Figures 2e and 2f. The optimal value of the tuning parameter ρ for these penalization techniques was $\rho = 10$. From Figures 2d and 2f, we observe that the best image segmentation results for the L_0 and L_1 penalization technique is the same. However, the L_0 penalization technique produces more sparsity (0.9645) than L_1 penalization technique (0.948) to obtain the same segmentation. We obtain good image segmentation in case of both the Group Lasso and the L_0 penalization technique.

D. Discussion

We have used several penalization techniques to obtain optimal reduced sets for kernel spectral clustering. We observe that the Group Lasso based penalization technique results in maximum sparsity in many cases and is computationally the most efficient as shown in Table II. The Group Lasso based penalization technique is also ideal for clustering as it retains groups of relevant data points. The L_0 penalization technique results in sparser solution than L_1 penalization technique in general but at the expense of more computational time. This is because it iteratively solves a QCQP for each sparsification step whereas there is no such iterative procedure for L_1 penalization technique. This can also be concluded from the computation time shown for the two methods in Table II. We also observe that as the size of the dataset increases the L_0, L_1 and $L_1 + L_0$ penalization based techniques become less feasible. This is because CVX is meant for smaller size optimization problems and cannot handle very large scale problems efficiently.

V. Conclusion

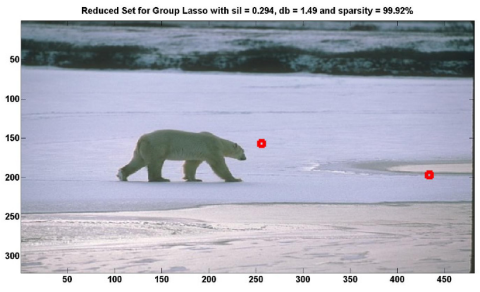
We proposed several methods for obtaining sparse optimal reduced sets for kernel spectral clustering. The formulation is based on weighted kernel PCA for a specific choice of weights. Several techniques like Group Lasso, L_0, $L_1 + L_0$ penalization methods had been proposed to obtain the reduced set along with the modified weight vectors. The methodologies were aimed to tackle different datasets in a computationally and memory efficient way. We observed that the Group Lasso resulted in the sparsest models with good clustering quality in least computation time followed by reduced models by the L_0 penalization. The reduced models obtained by $L_1 + L_0$ penalization technique are quite similar to the reduced models obtained by a previous L_1 penalization method.

Acknowledgments

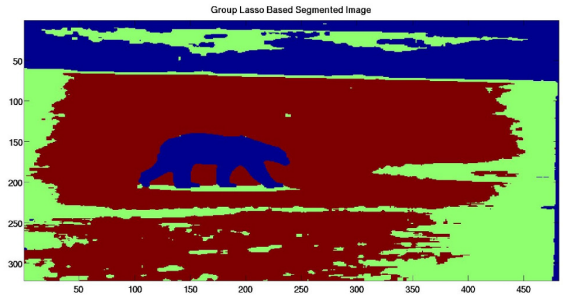
This work was supported by Research Council KUL: ERC AdG A-DATADRIVE-B, GOA/11/05 Ambiorics, GOA/10/09MaNet, CoE EF/05/006 Optimization in Engineering(OPTEC), IOF-SCORES4CHEM, several PhD/postdoc and fellow grants; Flemish Government:FWO: PhD/postdoc grants, projects: G0226.06 (cooperative systems & optimization), G0321.06 (Tensors), G.0302.07 (SVM/Kernel), G.0320.08 (convex MPC), G.0558.08 (Robust MHE), G.0557.08 (Glycemia2), G.0588.09 (Brain-machine) G.0377. 12 (structured models) research communities (WOG:ICCoS, ANMMM, MLDM); G.0377.09 (Mechatronics MPC) IWT: PhD Grants, Eureka-Flite+, SBO LeCoPro, SBO Climaqs, SBO POM, O&O-Dsquare; Belgian Federal Science Policy Office: IUAP P6/04 (DYSCO, Dynamical systems, control and optimization, 2007-2011); EU: ERNSI; FP7-HD-MPC (INFSO-ICT-223854), COST intelliCIS, FP7-EMBOCON (ICT-248940); Contract Research: AMINAL; Other:Helmholtz: viCERP, ACCM, Bauknecht, Hoerbiger. Johan Suykens is a professor at the KU Leuven, Belgium.

References

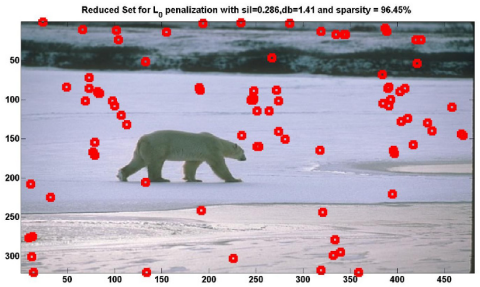
[1] Ng, A.Y., Jordan, M.I., Weiss, Y. On spectral clustering: analysis and an algorithm, *In proceedings of the Advances in Neural Information Processing Systems*; Dietterich, T.G., Becker, S., Ghahramani, Z., editors, MIT Press: Cambridge, MA, **2002**; pp. 849-856.



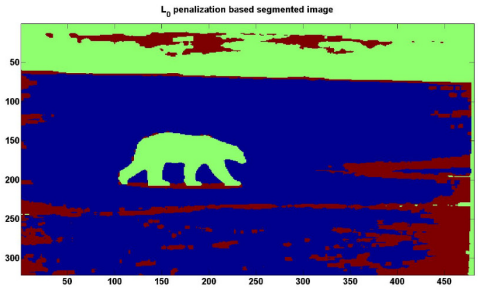
(a) Group Lasso based Reduced Set



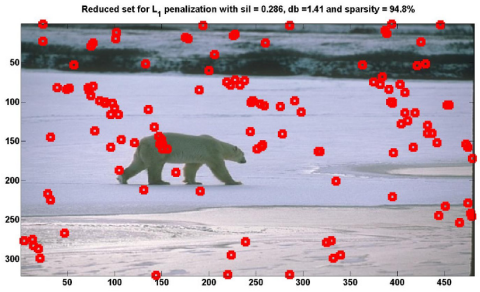
(b) Best GroupLasso based Image segmentation



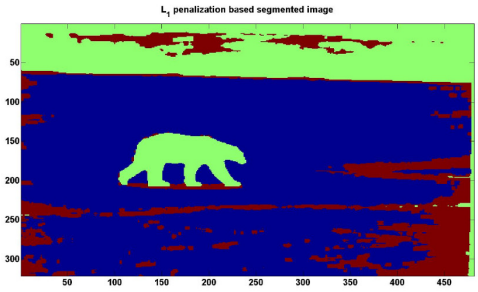
(c) Best L_0 penalization based Reduced Set



(d) Best L_0 penalization based Image segmentation



(e) Best L_1 penalization based Reduced Set [6]



(f) Best L_1 penalization based Image segmentation [6]

Fig. 2: Results on Image Dataset corresponding to the reduced sets obtained via different penalization techniques. The red-colored circular boxes represents the points selected as reduced set points

[2] von Luxburg, U. A tutorial on Spectral clustering. *Stat. Comput*, *17*, 395-416.

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

[4] Alzate, C., 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)*, 335-347 .

[5] Alzate, C., Suykens, J.A.K. Highly Sparse Kernel Spectral Clustering with Predictive Out-of-sample extensions. *ESANN*, **2010**, 235-240.

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

[7] Langone, R., Mall, R., Suykens, J.A.K. Soft Kernel Spectral Clustering. *IJCNN*, **2013**.

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

[9] Mall, R., Langone, R., 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.

[10] Mall, R., Langone, R., Suykens, J.A.K. Self-Tuned Kernel Spectral Clustering for Large Scale Networks, *IEEE International Conference on Big Data (IEEE BigData)*, **2013**, Santa Clara, U.S.A.

[11] Tibshirani, R. Regression shrinkage and Selection via the Lasso. *Journal of Royal Statistical Society*, **1996** *58(1)*, 267-288.

[12] Yuan, M., Lin, Y. Model selection and estimation in regression with grouped variables. *Journal of Royal Statistical Society*, **2006**, *68(1)*, 49-67.

[13] Friedman, J., Hastie, T., Tibshirani, R. A note on the group lasso and a sparse group lasso. *arXiv:1001.0736*, **2010**.

[14] K. Huang, D. Zheng, J. Sun, Y. Hotta, K. Fujimoto and S. Naoi. Sparse Learning for Support Vector Classification. *Pattern Recognition Letters*, **2010**, *31(13)*, 1944-1951.

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

[16] E. J. Nyström. Über die praktische Auflösung von Integralgleichungen mit Anwendungen auf Randwertaufgaben. *Acta Mathematica*, **1930**, *54*, 185-204.

[17] C. K. I. Williams and M. Seeger. Using the Nyström method to speed up kernel machines. *Advances in Neural Information Processing Systems*,

2001, *13*, 682-688.

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

[19] Kenney, J.F., Keeping, E.S. Linear Regression and Correlation. *Chapter 15 in Mathematics of Statistics*, *3(1)*, 252-285.

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

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

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