Sparsity in Large Scale Kernel Models

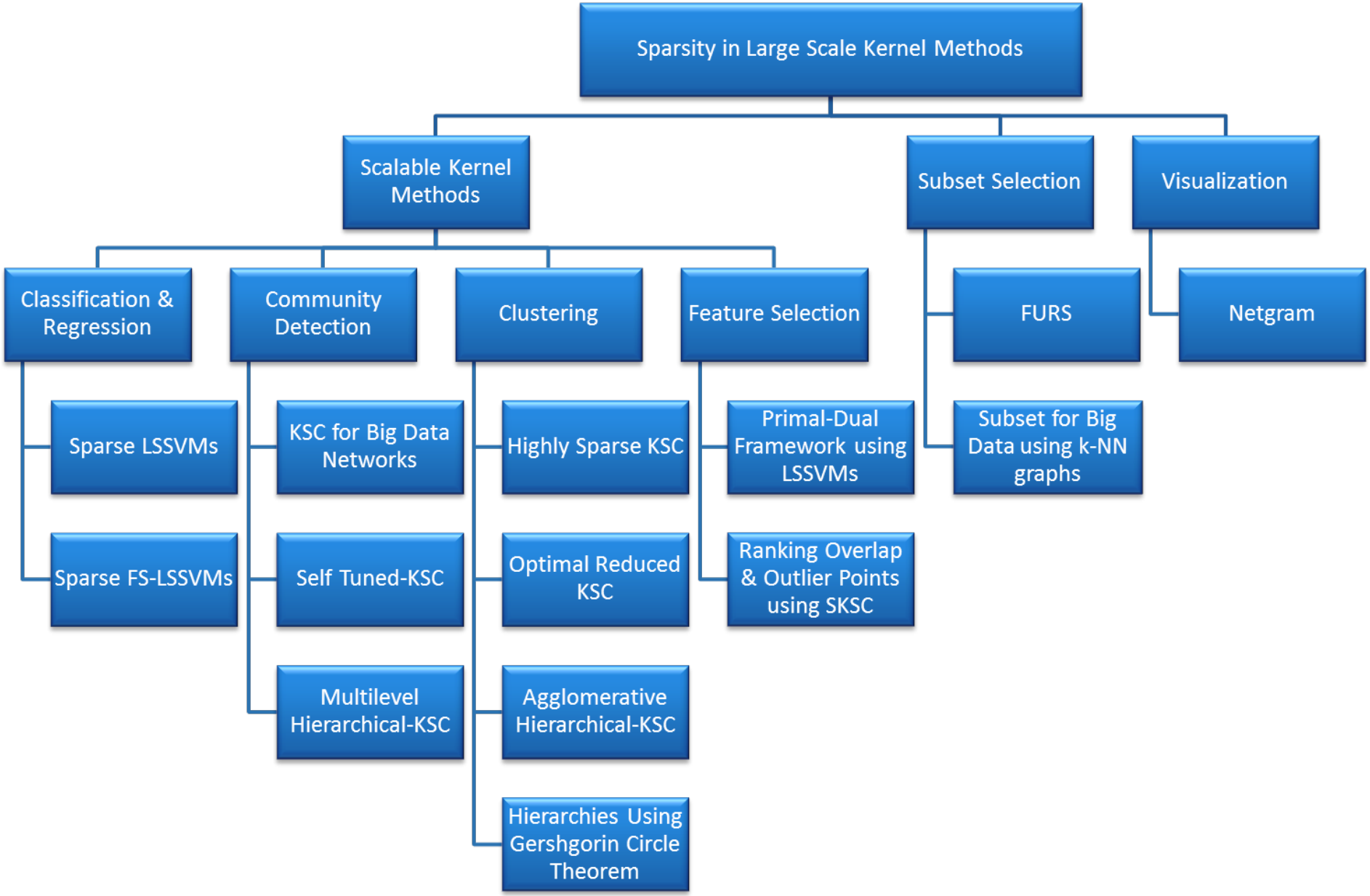
Raghvendra Mall
Promotor: Prof. Johan. A. K. Suykens

Dept: ESAT-STADIUS
KU Leuven, Belgium

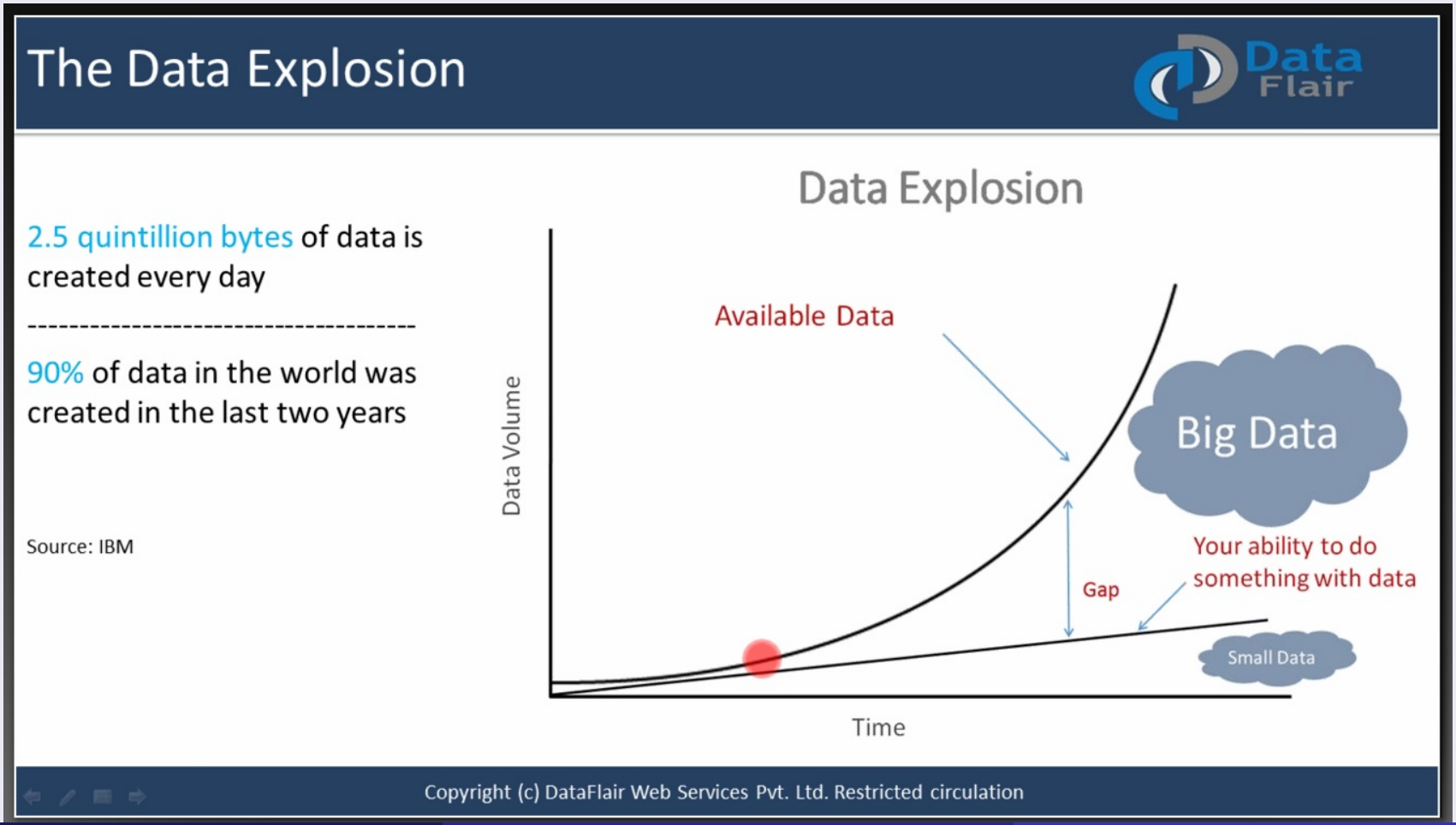


Pre-Defense Thesis Presentation
May 2014

Outline

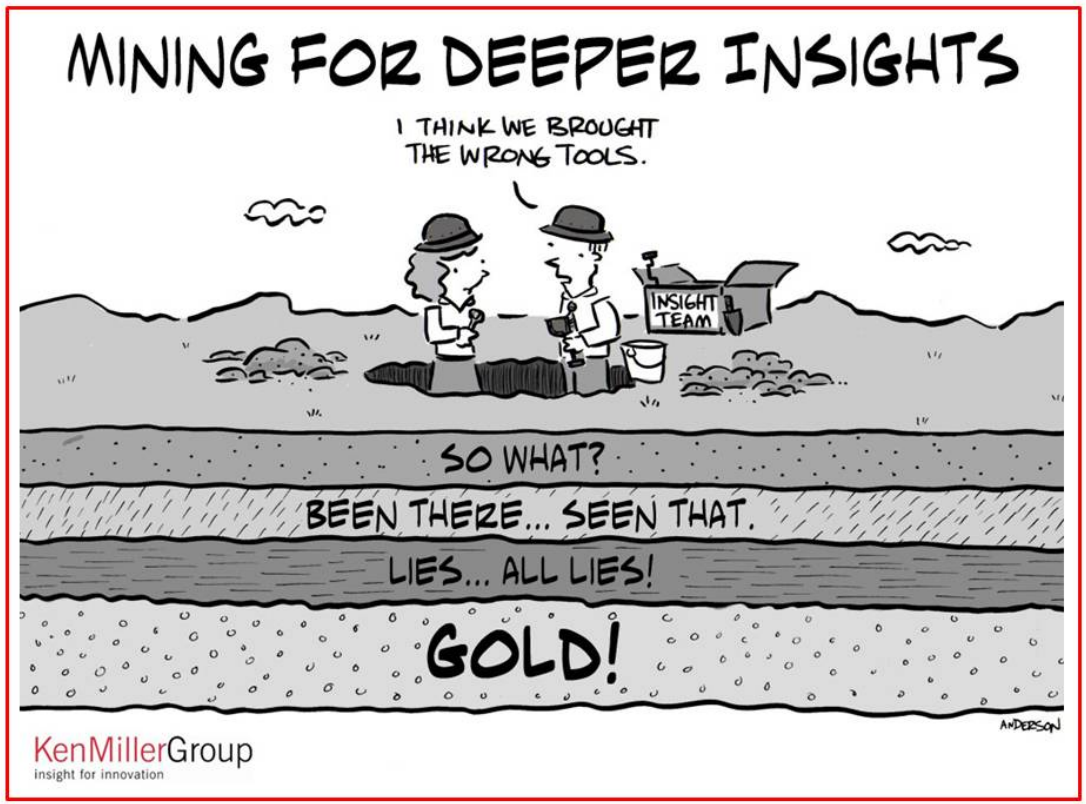

Sparsity in Large Scale Kernel Methods
Scalable Kernel Methods
Subset Selection
Visualization
Classification & Regression
Community Detection
Clustering
Feature Selection
FURS
Netgram
Sparse LSSVMs
KSC for Big Data Networks
Highly Sparse KSC
Primal-Dual Framework using LSSVMs
Subset for Big Data using k-NN graphs
Sparse FS-LSSVMs
Self Tuned-KSC
Optimal Reduced KSC
Ranking Overlap & Outlier Points using SKSC
Multilevel Hierarchical-KSC
Agglomerative Hierarchical-KSC
Hierarchies Using Gershgorin Circle Theorem

- Advent of technology & its widespread usage has resulted in proliferation of data.
- Data is a gold mine and the need is to find the right tools to mine the data.
- Real-world data is noisy & only small fraction contains the most information.
- Sparsity plays a big role in selection of this representative subset of data.
- Immense wealth of data has led to the emergence of the concept of Big Data.



Motivations

- Choices for selecting predictive models for Big Data learning are limited.
- One way is to design models which are fast, scalable and support parallelization or distributed computing.
- Kernel based models build on a subset of data with a proper training, validation and test phase make an ideal learning framework for large scale data.
- Kernel methods have powerful out-of-sample extension property which allows inference for unseen data and can exploit parallelism.



LSSVM formulation

- Formulation of Least Squares Support Vector Machines (Suykens et al, NPL, 1998 & World Scientific, 2002) is:

$$\min_{w,b,e} \quad \mathcal{J}(w,e) = \frac{1}{2} w^\intercal w + \frac{\gamma}{2} \sum_{i=1}^{N} e_i^2 \tag{1}$$

$$\text{s.t.} \quad [w^\intercal \phi(x_i) + b] = y_i - e_i, \ \forall i = 1, \ldots, N,$$

- Dual Solution:

$$\left[\begin{array}{c|c} 0 & 1_N^\intercal \\ \hline 1_N & \Omega + \frac{1}{\gamma} I_N \end{array}\right] \left[\begin{array}{c} b \\ \hline \alpha \end{array}\right] = \left[\begin{array}{c} 0 \\ \hline y \end{array}\right]. \tag{2}$$

- One KKT condition: $\alpha_i = \gamma e_i$ makes $\alpha_i \neq 0$ & results in lack of sparsity.

Primal FS-LSSVM

- Select M prototype vectors ($\mathcal{S}_{PV}$) using quadratic Rènyi entropy.
- Performs Nyström approximation to obtain explicit feature map $\{\hat{\phi}\} \in \mathbb{R}^M$.
- Applicable to large scale datasets but not the sparsest.
- Primal Fixed-Size LSSVM (Suykens et al, 2002 & De Brabanter et al, CSDA, 2010) is formulated as:

$$\min_{\hat{w},\hat{b}} \quad \mathcal{J}(\hat{w},\hat{b}) = \frac{1}{2} \hat{w}^\intercal \hat{w} + \frac{\gamma}{2} \sum_{i=1}^{N} (y_i - [\hat{w}^\intercal \hat{\phi}(x_i) + \hat{b}])^2 \tag{3}$$

- Primal FS-LSSVM solution:

$$\left[\begin{array}{c|c} \hat{\Phi}^\intercal \hat{\Phi} + \frac{1}{\gamma} I_N & \hat{\Phi}^\intercal 1_N \\ \hline 1_N^\intercal \hat{\Phi} & 1_N^\intercal 1_N \end{array}\right] \left[\begin{array}{c} \hat{w} \\ \hline \hat{b} \end{array}\right] = \left[\begin{array}{c} \hat{\Phi}^\intercal y \\ \hline 1_N^\intercal y \end{array}\right] \tag{4}$$

Two stages of sparsity (SPFS-LSSVM)

	Stage 1	Stage 2
Primal	Primal FS model estimation $\longrightarrow$	Re-weighted L_1
	$\uparrow$	
	subset selection Nyström approximation	

Synergy between parametric & kernel-based models
(Mall & Suykens, IEEE-TNNLS, 2015), Re-weighted L_1 (Candes et al, JFAA, 2008)

Sparse Primal FS-LSSVM - SPFS-LSSVM

	Initial		1^{st} **Reduction**		2^{nd} **Reduction**
SV/Train	N/N		M/N		M'/N
Primal	$w, \mathcal{D}$	**Step 1** $\rightarrow$	$\hat{w}, \mathcal{S}_{\mathcal{PV}}$	**Step 2** $\rightarrow$	$\beta', b', \mathcal{S}_{\mathcal{SV}}$
	$\phi(x) \in \mathbb{R}^{n_h}$	**PFS-LSSVM**	$\hat{\phi}(x) \in \mathbb{R}^{M}$	**SPFS-LSSVM**	

Table: In Step 2 we perform an iterative sparsifying re-weighted L_1 procedure in the primal.

- Final predictive model is: $f(x^*) = \sum_{i=1}^{M'} \beta'_i \hat{\phi}(z_i)^\intercal \hat{\phi}(x^*) + b'$.

Two stages of sparsity (SSD-LSSVM)

	Dual		
Stage 2			Re-weighted L_1
			$\uparrow$
Stage 1	subset selection Nyström approximation	$\longrightarrow$	Subsampled dual model

Synergy between parametric & kernel-based models
(Mall & Suykens, IEEE-TNNLS, 2015), Re-weighted L_1 (Candes et al, JFAA, 2008)

Sparse Subsampled Dual LSSVM - SSD-LSSVM

	Initial		1^{st} **Reduction**		2^{nd} **Reduction**
SV/Train	N/N		M/M		M'/N
Dual	$\alpha, \mathcal{D}$	**Step 1** $\rightarrow$	$\bar{\alpha}, \mathcal{S}_{PV}$	**Step 2** $\rightarrow$	$\alpha', b', \mathcal{S}_{SV}$
	$\phi(x) \in \mathbb{R}^{n_h}$	**SD-LSSVM**	$\phi(x) \in \mathbb{R}^{n_h}$	**SSD-LSSVM**	

Table: In Step 2 we perform an iterative sparsifying re-weighted L_1 procedure in the dual.

- Final predictive model is: $f(x^*) = \sum_{i=1}^{M'} \alpha'_i K(z_i, x^*) + b'$.

Experimental Results (Scales upto 10^6 points on a laptop)

Test Classification(Error and Mean SV)								
	BC		MGT		ADU		FC	
Algorithm	Error	SV	Error	SV	Error	SV	Error	SV
PFS-LSSVM [1]	0.023 ± 0.004	153	$\mathbf{0.134 \pm 0.001}$	414	0.147 ± 0.0	664	0.2048 ± 0.026	763
C-SVC [2]	0.0348 ± 0.02	414	0.144 ± 0.015	7000	$0.151(*)$	11085(*)	$0.185(*)$	185000(*)
ν-SVC [2]	0.028 ± 0.011	398	0.158 ± 0.014	7252	$0.161(*)$	12205(*)	$184(*)$	165205(*)
Lopez [3]	0.04 ± 0.011	17	-	-	-	-	-	-
Keerthi [4]	0.0463 ± 0.037	30	0.135 ± 0.002	260	$\mathbf{0.145 \pm 0.002}$	494	0.2054(*)	762(*)
SV_L_0-norm [5]	0.023 ± 0.01	27	0.15 ± 0.009	75	0.149 ± 0.001	236	0.2346 ± 0.025	352
SPFS-LSSVM	$\mathbf{0.021 \pm 0.01}$	26	0.141 ± 0.004	163	0.148 ± 0.0	464	0.2195 ± 0.03	505
SSD-LSSVM	0.0256 ± 0.007	8	0.135 ± 0.0	280	0.1488 ± 0.001	102	$\mathbf{0.1905 \pm 0.009}$	635
Test Regression(Error and Mean SV)								
	Boston Housing		Concrete		Slice Localization		Year Prediction	
Algorithm	Error	SV	Error	SV	Error	SV	Error	SV
PFS-LSSVM [1]	$\mathbf{0.133 \pm 0.002}$	113	$\mathbf{0.112 \pm 0.006}$	193	$\mathbf{0.0554 \pm 0.0}$	694	$\mathbf{0.402 \pm 0.034}$	718
ϵ-SVR [2]	0.16 ± 0.05	226	0.23 ± 0.02	670	0.102(*)	13012(*)	0.465(*)	192420(*)
ν-SVR [2]	0.16 ± 0.04	195	0.22 ± 0.02	330	0.092(*)	12524	0.472(*)	175250(*)
Lopez [3]	0.16 ± 0.05	65	0.165 ± 0.07	215	-	-	-	-
SV_L_0-norm [5]	0.192 ± 0.015	38	0.265 ± 0.02	25	0.156 ± 0.13	381	0.495 ± 0.03	422
SPFS-LSSVM	0.176 ± 0.03	50	0.2383 ± 0.04	46	0.058 ± 0.002	651	0.44 ± 0.05	595
SSD-LSSVM	0.189 ± 0.022	43	0.1645 ± 0.017	54	0.056 ± 0.0	577	0.42 ± 0.01	688

Table: Comparison in performance of different methods for UCI repository datasets. The (*) represents no cross-validation and performance on a fixed value of tuning parameters due to computational burden and (-) represents cannot scale for this dataset.

Motivation Example

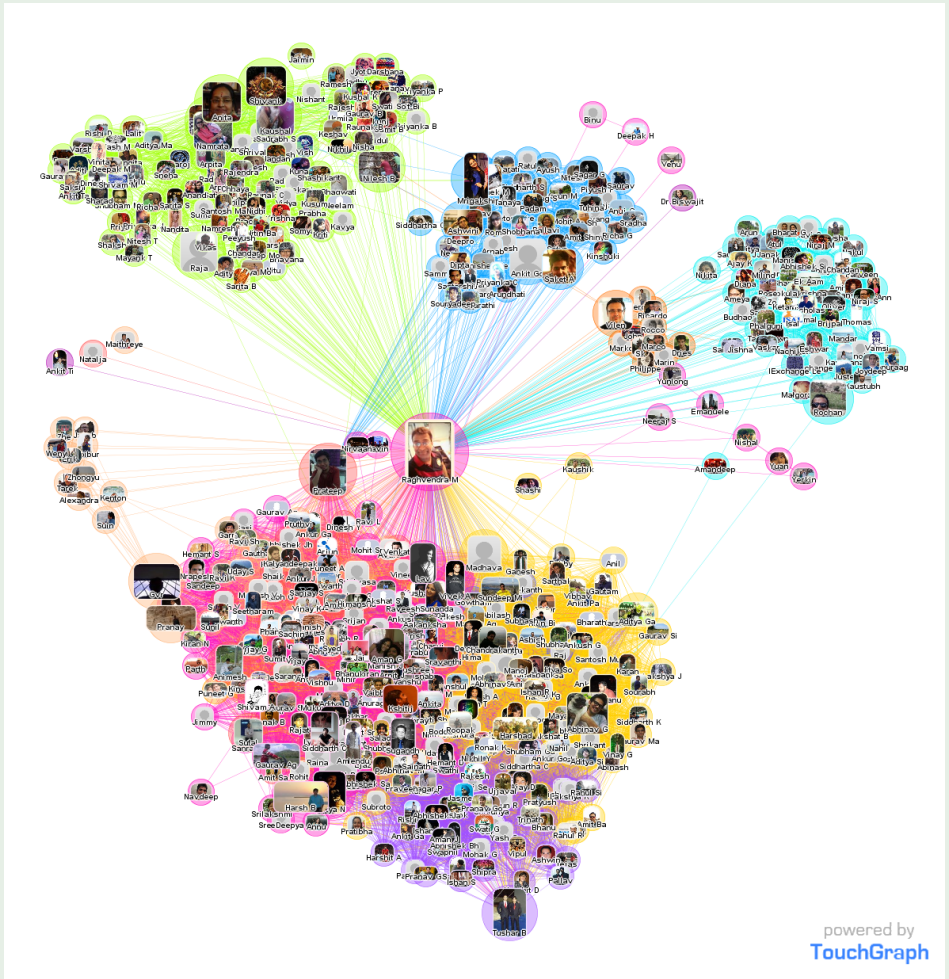


Figure: Identify communities in a flat & hierarchical manner.

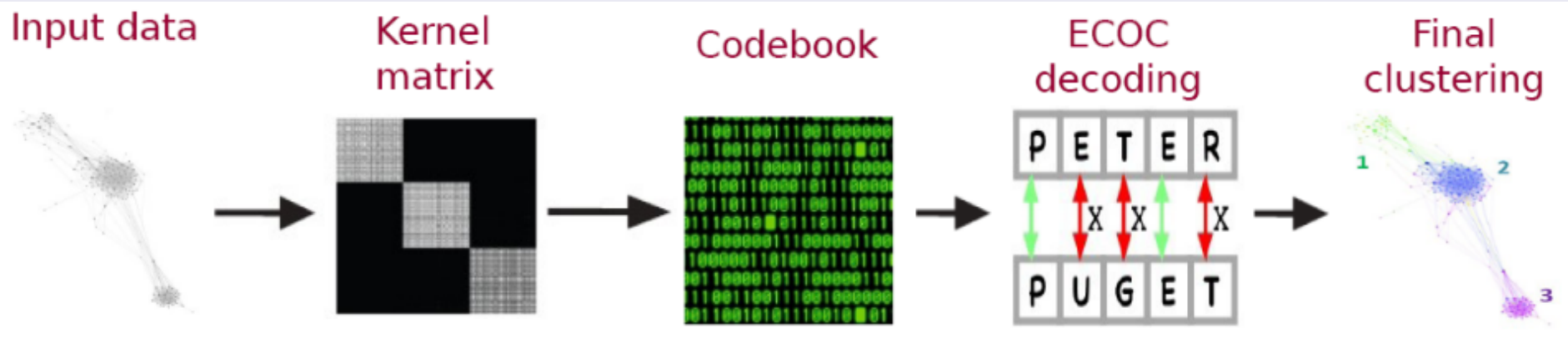
Kernel Spectral Clustering (KSC) Formulation

Given $\mathcal{D} = \{x_i\}_{i=1}^{N_{tr}}$ and *maxk*, the primal formulation of the weighted kPCA (Alzate & Suykens, IEEE TPAMI, 2010) is:

$$\min_{w^{(l)}, e^{(l)}, b_l} \quad \frac{1}{2} \sum_{l=1}^{maxk-1} w^{(l)\intercal} w^{(l)} - \frac{1}{2N_{tr}} \sum_{l=1}^{maxk-1} \gamma_l e^{(l)\intercal} D_\Omega^{-1} e^{(l)} \tag{5}$$

$$\text{such that} \quad e^{(l)} = \Phi w^{(l)} + b_l 1_{N_{tr}}, l = 1, \ldots, maxk - 1$$

KSC Steps



KSC Requirements

- Requires a representative subset of data to build the KSC model.
- A good model selection criterion to estimate k and σ.
- Need to fine-tune the technique and demonstrate its applicability to Big Data.

FURS Illustration

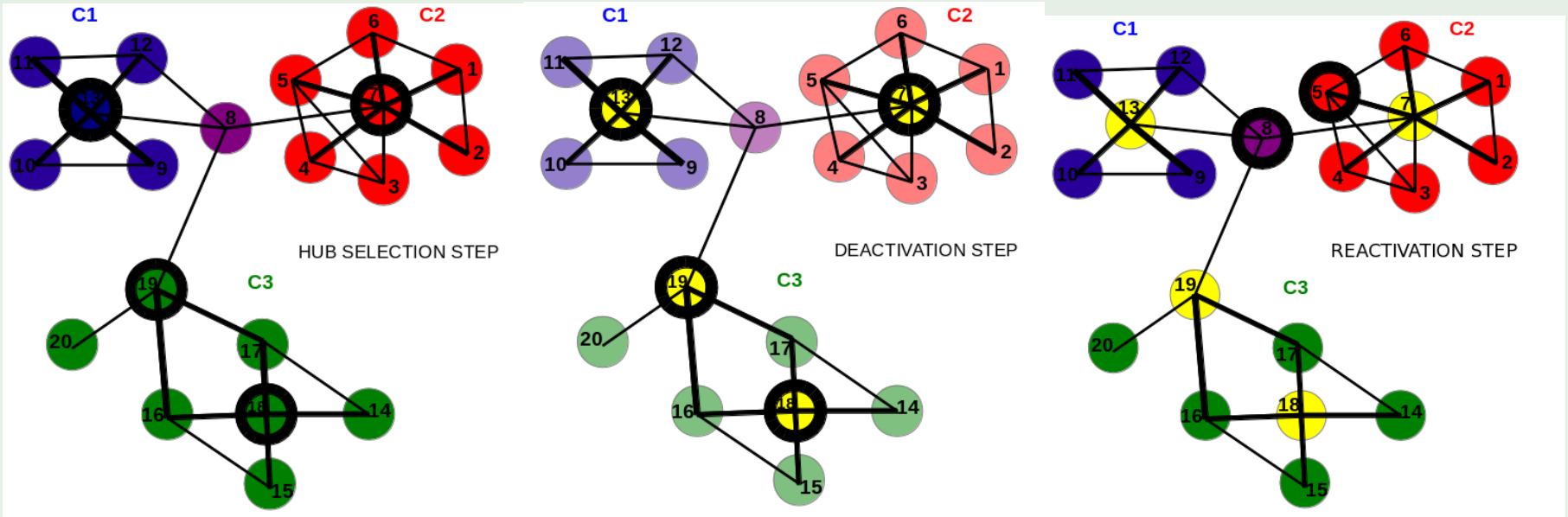


Figure: Steps of FURS selection technique for an example with 20 nodes. The required subset size is $M = 6$. We first selected v_7 and deactivate its neighborhood, followed by selection of v_{13}, v_{18} and v_{19}. However, by this time all the nodes in the network in deactivated. We then activate all the nodes and then select vertex v_5 and v_8 respectively to obtain the require subset.

Experiments on Real World Networks

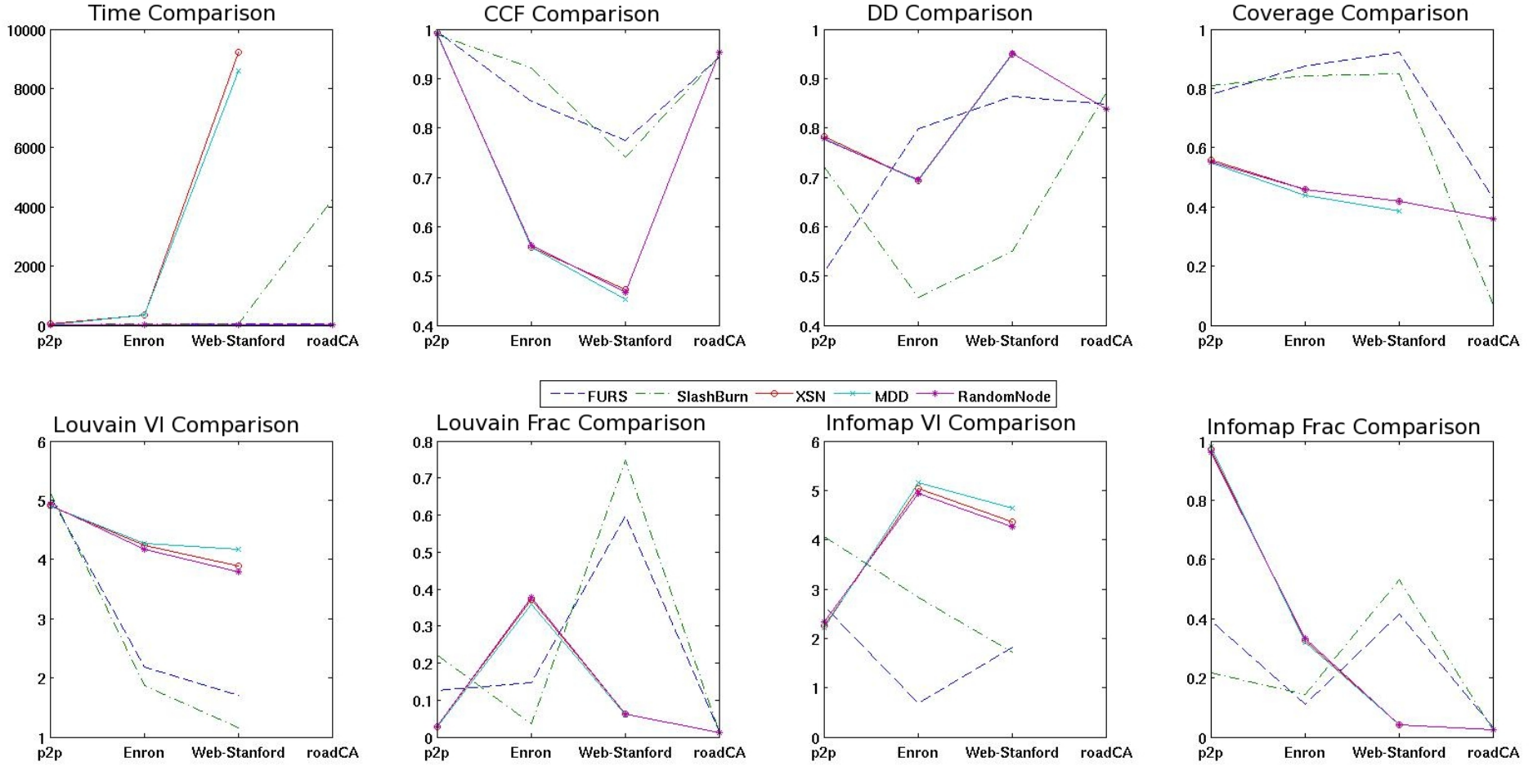
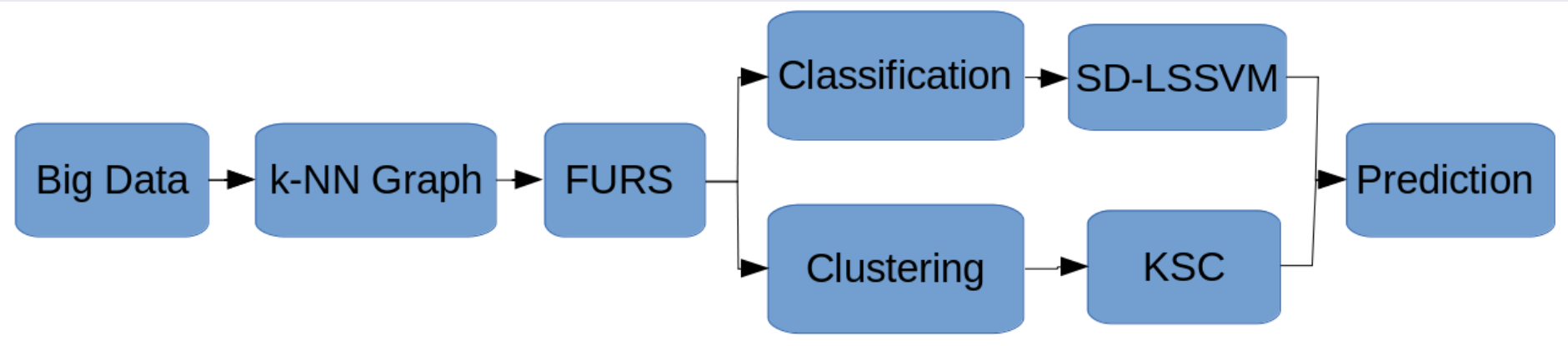


Figure: Evaluation of various subset selection methods.

Dataset	**Nodes**	**Edges**	**Time**	**Coverage**	**CCF**	**DD**
Synthetic	500,000	23,563,412	103.49	0.44	0.906	0.786
YouTube	1,134,890	2,987,624	20.546	0.28	0.975	0.959
roadCA	1,965,206	5,533,214	30.816	0.135	0.89	0.40
Livejournal	3,997,962	34,681,189	181.21	0.173	0.92	0.953

Proposed Framework (Mall et al, IEEE BigData, 2014)



- k-NN = k-Nearest Neighbor
- KSC = Kernel Spectral Clustering
- LSSVM = Least Squares Support Vector Machines
- SD-LSSVM = Subsampled-Dual LSSVM
- Applicable for Big Data (datasets & images).
- Applied on SUSY dataset (Particle Accelerator, 5 million points) available at http://archive.ics.uci.edu/ml/

Distributed k-NN Graph Generation

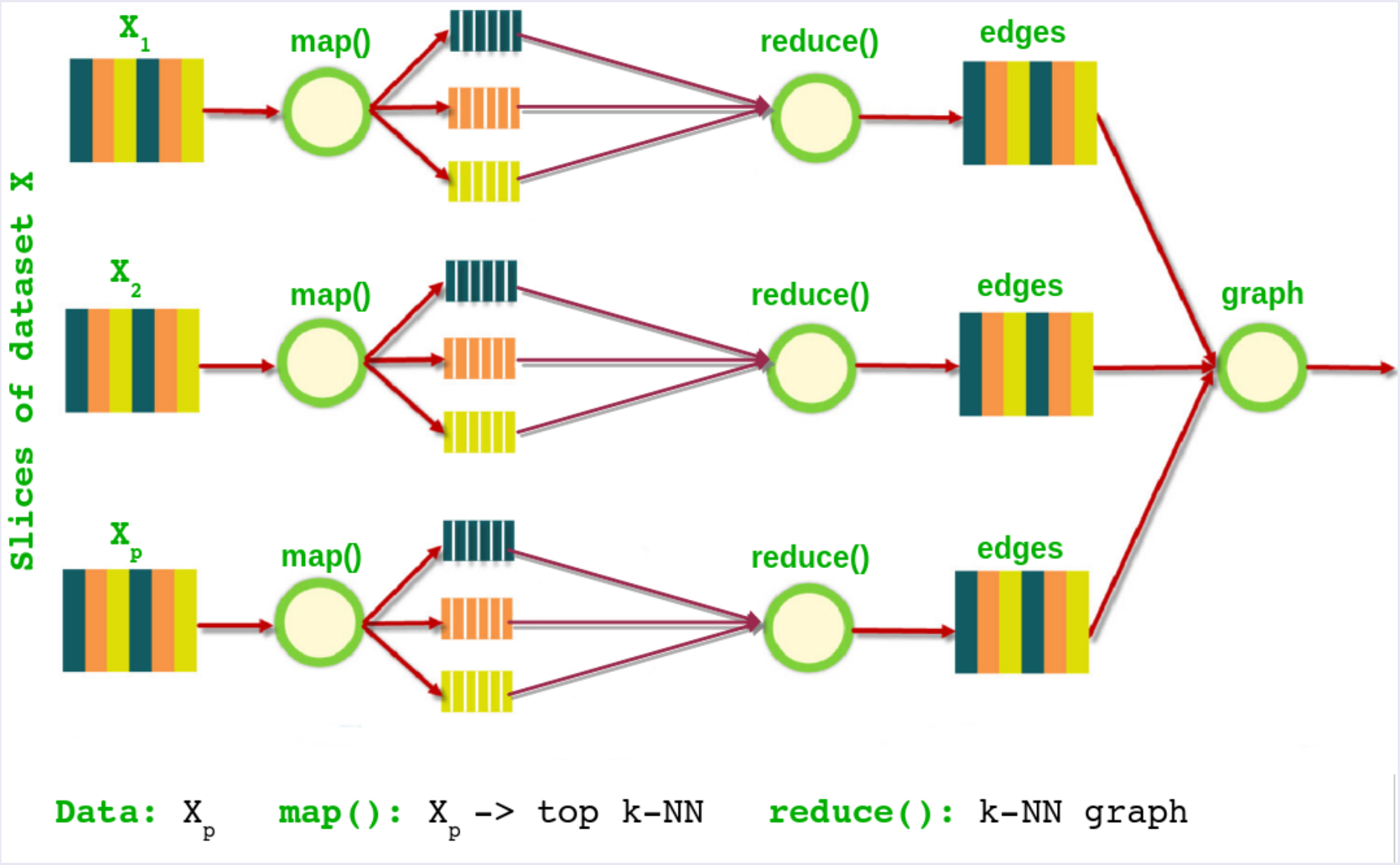


Figure: Creating the k-NN graph using MapReduce framework.

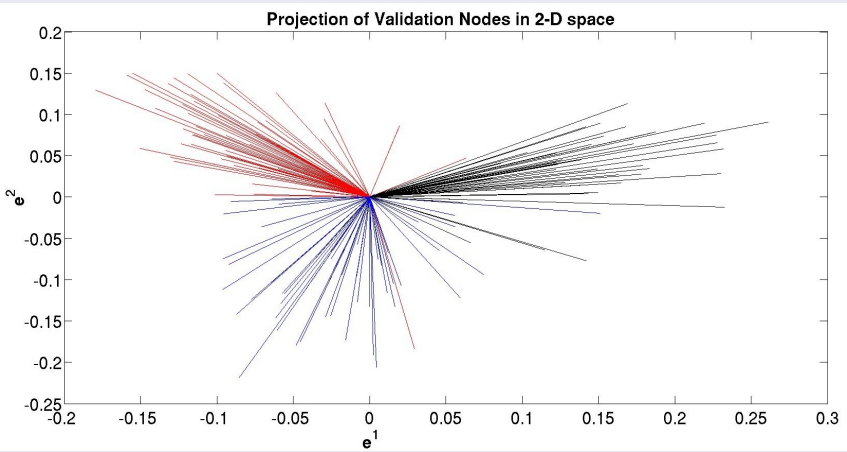
KSC-net (Mall & Suykens, CRC Press, Taylor & Francis, 2015)

- KSC-net software available for community detection at `http://www.esat.kuleuven.be/stadius/ADB/mall/softwareKSCnet.php`.
- Subset selection done using FURS
- Model selection possibilities include 'BAF' or 'Self-Tuned' criterion.
- Current version of software can perform community detection on graphs with 10^6-10^7 nodes and 10^8-10^9 edges in 5-6 minutes on 12Gb Ram laptop.

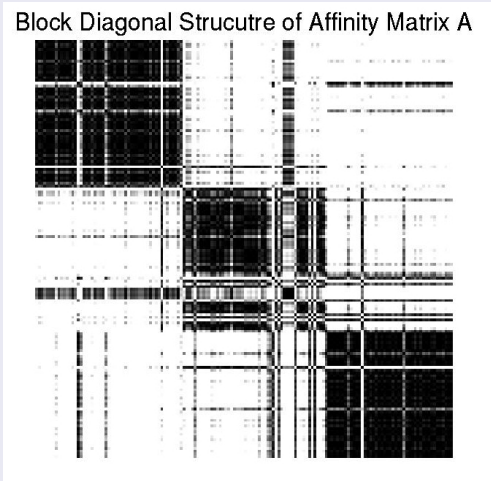
Self-Tuned Approach for Graphs (Mall et al, IEEE BigData 2013)

A new symmetric affinity matrix A is created from E s.t.

$A(i,j) = CosDist(e_i, e_j) = 1 - \cos(e_i, e_j) = 1 - \frac{e_i^T e_j}{\|e_i\| \|e_j\|}$.



(a) Validation node projections



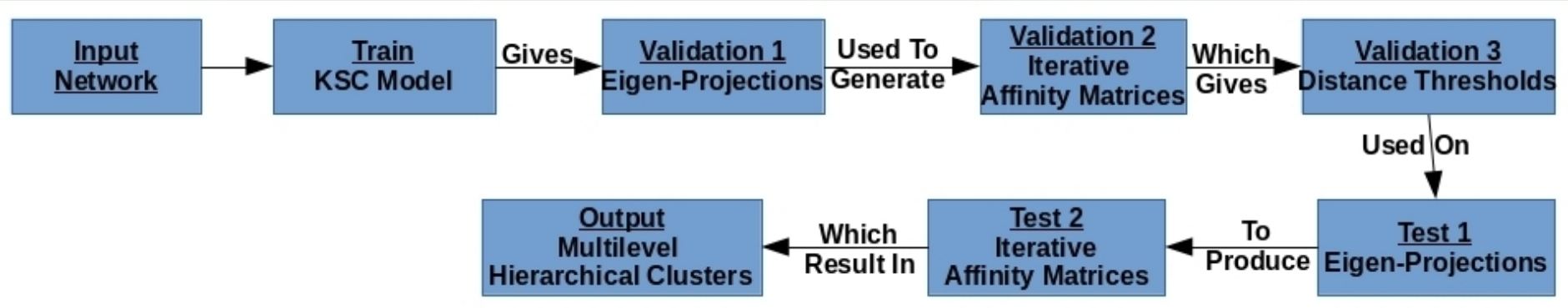
(b) Affinity matrix

ST-KSC (Mall et al, IEEE BigData 2013) Real World Results

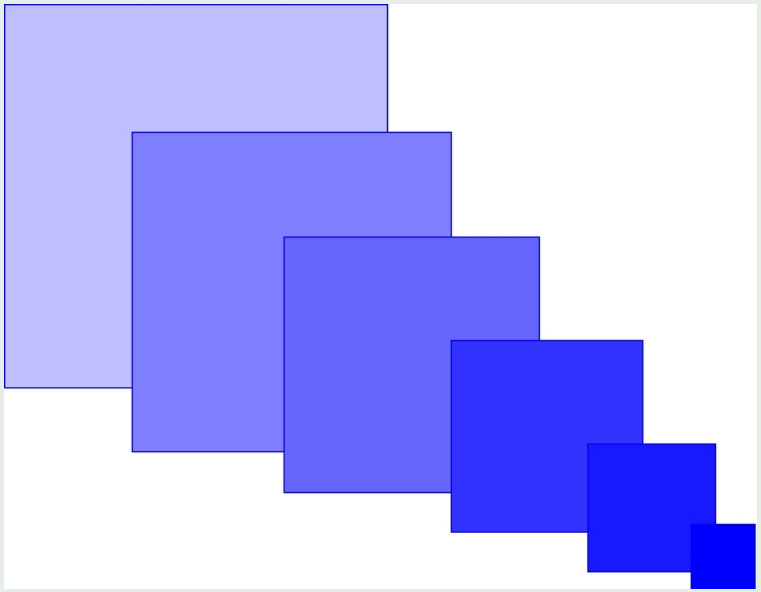
		Original KSC				ST-KSC				ST k-means KSC			
		k	VI	ARI	MI	k	VI	ARI	MI	k	VI	ARI	MI
PGPnet	Louvain	4	4.022	0.004	0.124	30	**3.94**	**0.073**	**1.102**	30	4.08	0.041	0.87
	Infomap	4	**3.583**	0.006	0.1	30	3.7	**0.1**	**0.98**	30	3.77	0.063	0.78
	Bigclam	4	**1.57**	0.011	0.029	30	3.4	**0.12**	**1.026**	30	3.462	0.07	0.78
Cond-mat	Louvain	3	4.021	0.01	0.225	52	**3.91**	**0.011**	0.324	52	4.5	0.04	**0.68**
	Infomap	3	3.132	0.05	0.222	52	**3.1**	0.051	0.32	52	3.63	**0.224**	**0.68**
	Bigclam	3	**0.5**	**0.49**	0.562	52	1.47	0.155	**2.49**	52	1.22	0.331	1.99
HepPh	Louvain	3	**3.22**	0.028	0.093	76	5.7395	0.014	0.216	76	5.23	**0.09**	**0.64**
	Infomap	3	**5.72**	0.006	0.166	76	7.224	0.014	0.78	76	6.3	**0.1**	**1.4**
	Bigclam	3	**0.95**	0.0004	0.0005	76	6.44	0.001	0.51	76	6.25	**0.09**	**0.8**
Enron	Louvain	3	3.6	0.038	0.274	16	**3.57**	**0.053**	**0.442**	16	3.51	0.0441	0.37
	Infomap	3	**1.6**	0.467	0.261	16	1.73	**0.53**	**0.4**	16	1.62	0.51	0.344
	Bigclam	3	**0.7**	0.48	0.51	16	0.91	**0.51**	**1.97**	16	0.94	0.50	1.3
Epinion	Louvain	3	3.1	0.0006	0.005	24	**3.08**	0.001	0.015	24	3.451	**0.03**	**0.06**
	Infomap	3	4.163	0.017	0.007	24	**4.16**	**0.031**	0.013	24	4.47	0.014	**0.1**
	Bigclam	3	**0.11**	**0.1**	0.007	24	0.84	0.02	**0.02**	24	0.74	0.014	0.016
Imdb-actor	Louvain	3	**3.05**	0.0005	0.006	59	4.4	**0.114**	**1.09**	59	3.84	0.093	0.673
	Infomap	3	**2.7**	0.0006	0.004	59	4.2	**0.11**	**0.99**	59	3.58	0.08	0.63
	Bigclam	3	**1.1**	0.0	0.0	59	3.0405	**0.025**	**1.07**	59	3.14	0.02	0.83
Youtube	Louvain	3	**3.64**	0.0003	0.004	21	4.1	0.016	0.047	21	4.55	**0.017**	**0.096**
	Infomap	3	**3.71**	0.0003	0.0035	21	4.16	0.014	0.03	21	4.7	**0.016**	**0.07**
	Bigclam	3	**0.056**	0.0	2.6e-06	21	1.6	0.035	0.06	21	1.25	**0.09**	**0.1**
roadCA	Louvain	3	**6.12**	0.0004	0.141	39	6.38	0.0005	**0.41**	39	6.42	**0.0006**	0.242
	Infomap	3	8.0	7.6e-05	0.21	39	**7.9**	0.0001	**0.64**	39	8.14	**0.00011**	0.40206
	Bigclam	3	**1.075**	0.0	0.0	39	2.62	0.044	**1.26**	39	1.56	**0.17**	1.01

Table: Evaluation of the cluster memberships for Original KSC, ST-KSC and k-means KSC for several real world networks w.r.t. Louvain (Blondel et al, JSM, 2008), Infomap (Rosvall et al, PNAS, 2008) and Bigclam (Leskovec et al, WSDM, 2013) methods.

Building Blocks with Illustrations



(a) Steps undertaken by the AH-KSC/MH-KSC (Mall et al, PLoS One, 2014 & Mall et al, IEEE SSCI CIDM, 2014) algorithm.



(b) Generating a series of affinity matrices over different levels. Communities at level h become nodes for next level $h+1$.

Determining Distance Thresholds

1. Create $S^{(0)}_{valid}$ for the validation nodes as:
$$S^{(0)}_{valid}(i,j) = CosDist(e_i, e_j) = 1 - \cos(e_i, e_j) = 1 - \frac{e_i^T e_j}{\|e_i\|\|e_j\|}.$$
2. Set initial distance threshold $t^{(0)} \in [0.1, 0.2]$ & $h = 0$.
3. *GreedyMaxOrder* gives $C^{(h)}$ and k at level h.
4. Generate affinity matrix at level h of hierarchy as:
$$S^{(h)}_{valid}(i,j) = \frac{\sum_{m \in C_i^{(h-1)}} \sum_{l \in C_j^{(h-1)}} S^{(h-1)}_{valid}(m,l)}{|C_i^{(h-1)}| \times |C_j^{(h-1)}|} \quad (6)$$
5. Threshold $t^{(h)} = \text{mean}(\min_j(S^{(h)}_{valid}(i,j))), i \neq j$ at level h.
6. Repeat Steps 4,5,6 till $k = 1$.
7. Obtain $\mathcal{T} = \{t^{(0)}, \ldots, t^{(maxh)}\}$.

AH-KSC Comparison

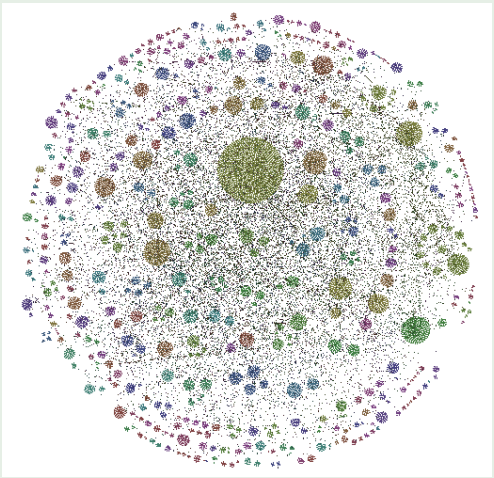
Method	Net_1						Net_2					
	Level	k	*ARI*	*VI*	*Q*	*CC*	**Level**	k	*ARI*	*VI*	*Q*	*CC*
Louvain [7]	1	32	0.84	0.215	**0.693**	4.87e-05	1	135	0.853	0.396	**0.687**	**1.98e-05**
	2	9	**1.0**	**0.0**	**0.786**	**5.01e-04**	3	13	**1.0**	**0.0**	**0.82**	**2.0e-05**
Infomap [8]	1	8	0.915	0.132	0.771	5.03e-04	1	590	0.003	8.58	0.003	**1.98e-05**
	2	6	0.192	1.965	0.487	5.07e-04	3	13	**1.0**	**0.0**	**0.82**	**2.0e-05**
OSLOM [9]	1	38	0.988	0.037	0.655	4.839e-04	1	141	**0.96**	**0.214**	0.64	2.07e-05
	2	9	**1.0**	**0.0**	**0.786**	**5.01e-04**	2	29	0.74	0.633	076	2.08e-05
AH-KSC [10]	2	39	**0.996**	**0.016**	0.67	**4.83e-04**	3	134	0.685	0.612	0.66	**1.98e-05**
	5	9	**1.0**	**0.0**	**0.786**	**5.01e-04**	10	13	**1.0**	**0.0**	**0.82**	**2.0e-05**

Table: 2 best level of hierarchy obtained by Louvain (Blondel et al, JSM, 2008), Infomap (Rosvall et al, PNAS, 2008), OSLOM (Lancichinetti et al, PLoS One, 2011) and AH-KSC (Mall et al, PLoS One, 2014) methods on Net_1 and Net_2 networks.

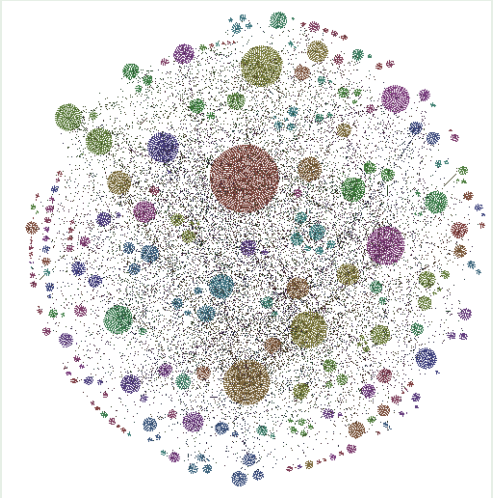
AH-KSC/MH-KSC Implementation

- AH-KSC code for networks is available at `http://www.esat.kuleuven.be/stadius/ADB/mall/softwareMHKSC.php`.
- Current version of software can scale to graphs with 10^6 nodes and 10^8 edges in 8-10 minutes on 12Gb Ram laptop.

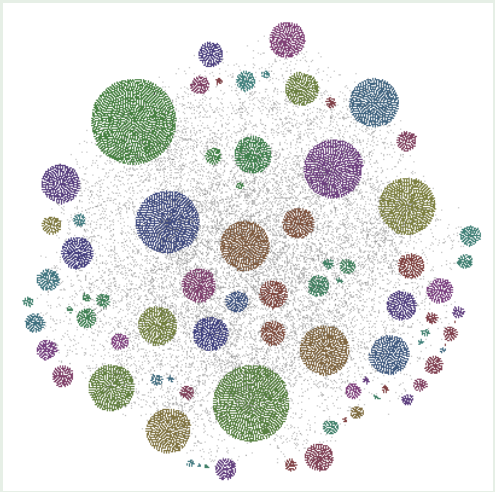
AH-KSC Illustration



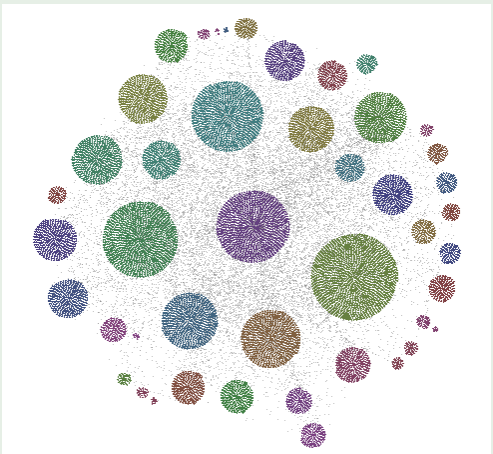
(a) Fine



(b) Intermediate



(c) Intermediate



(d) Coarse

Figure: Results of the **AH-KSC** algorithm for the PGP network.

Hierarchical Tree Organization using AH-KSC

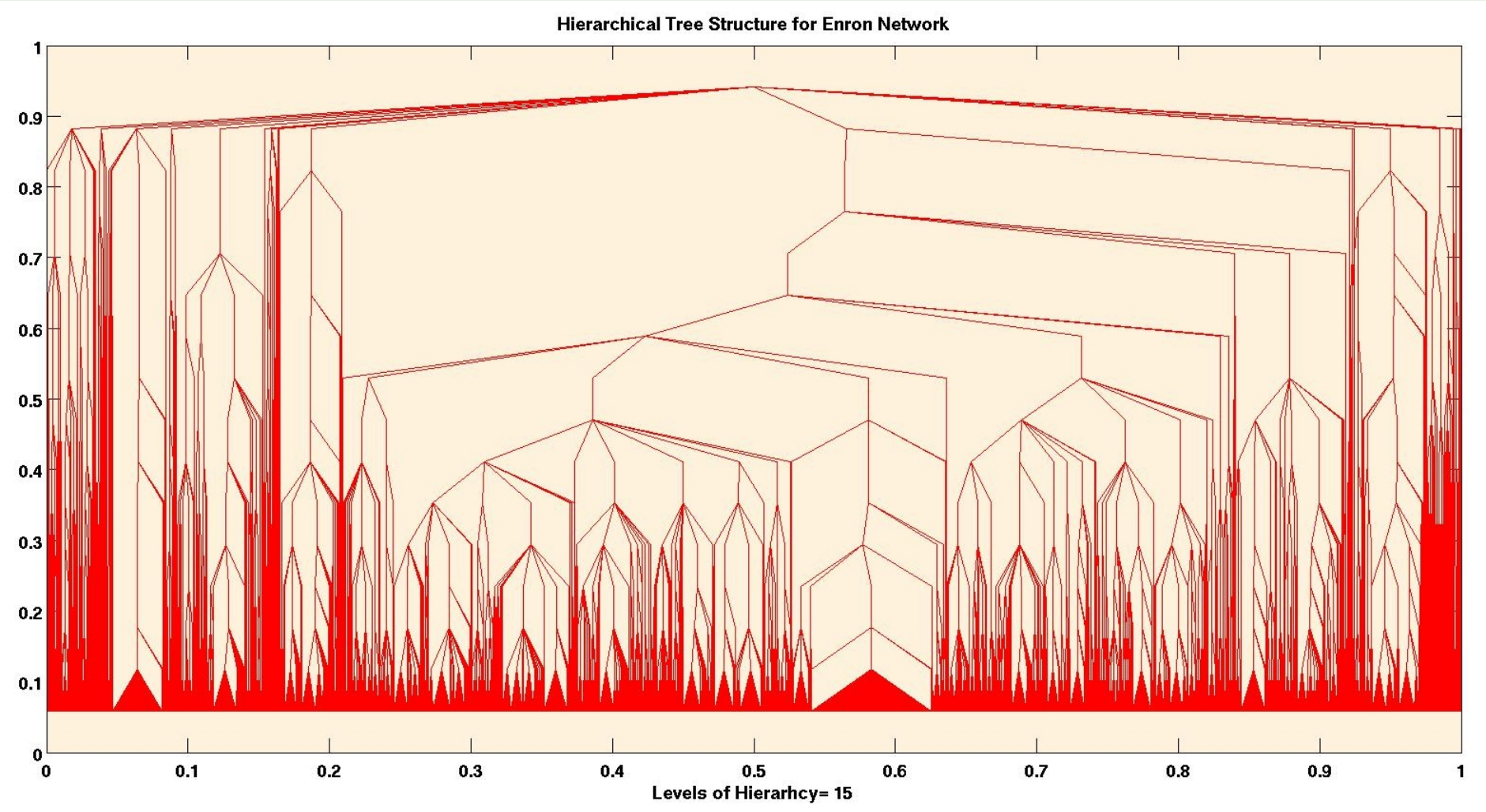


Figure: Result on Enron Network

AH-KSC steps for Images (Mall et al, IEEE CIDM, 2014)

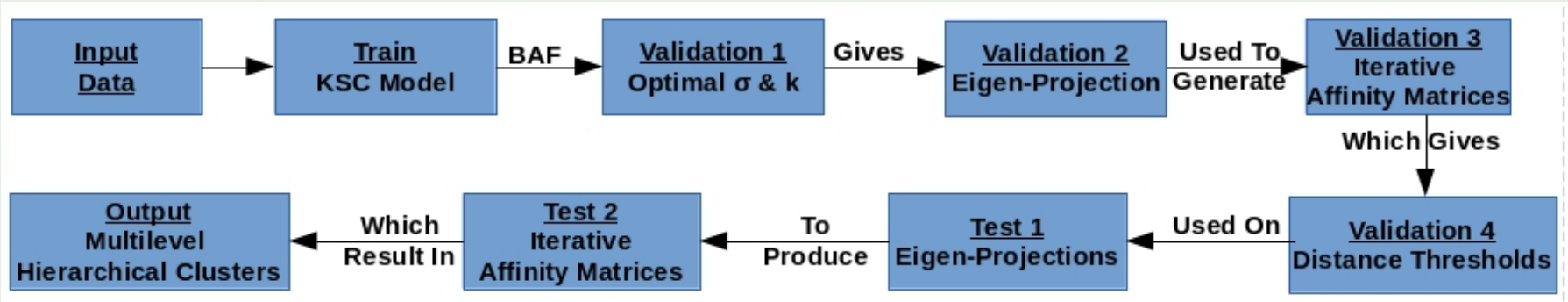


Figure: Steps of Agglomerative Hierarchical-KSC method with additional step where the optimal σ and k are estimated using Balanced Angular Fitting criterion.

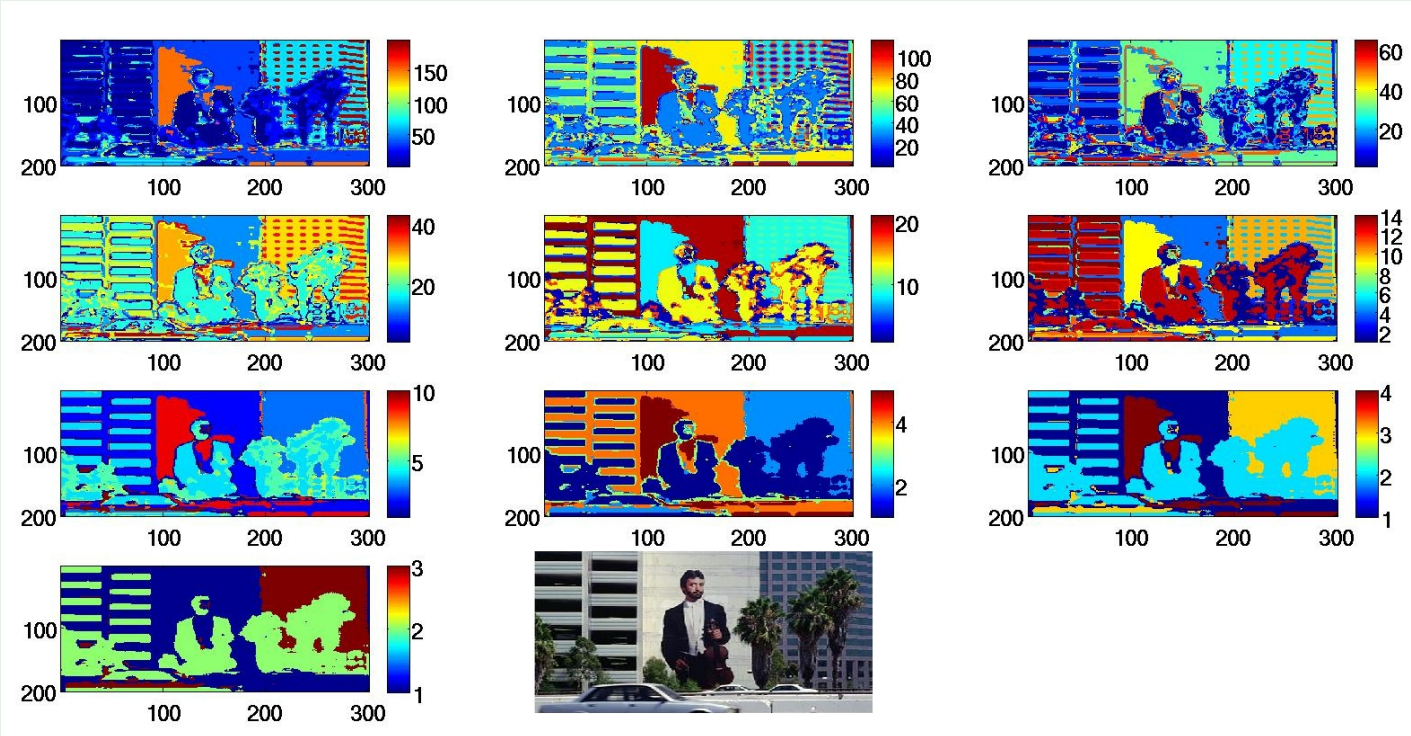
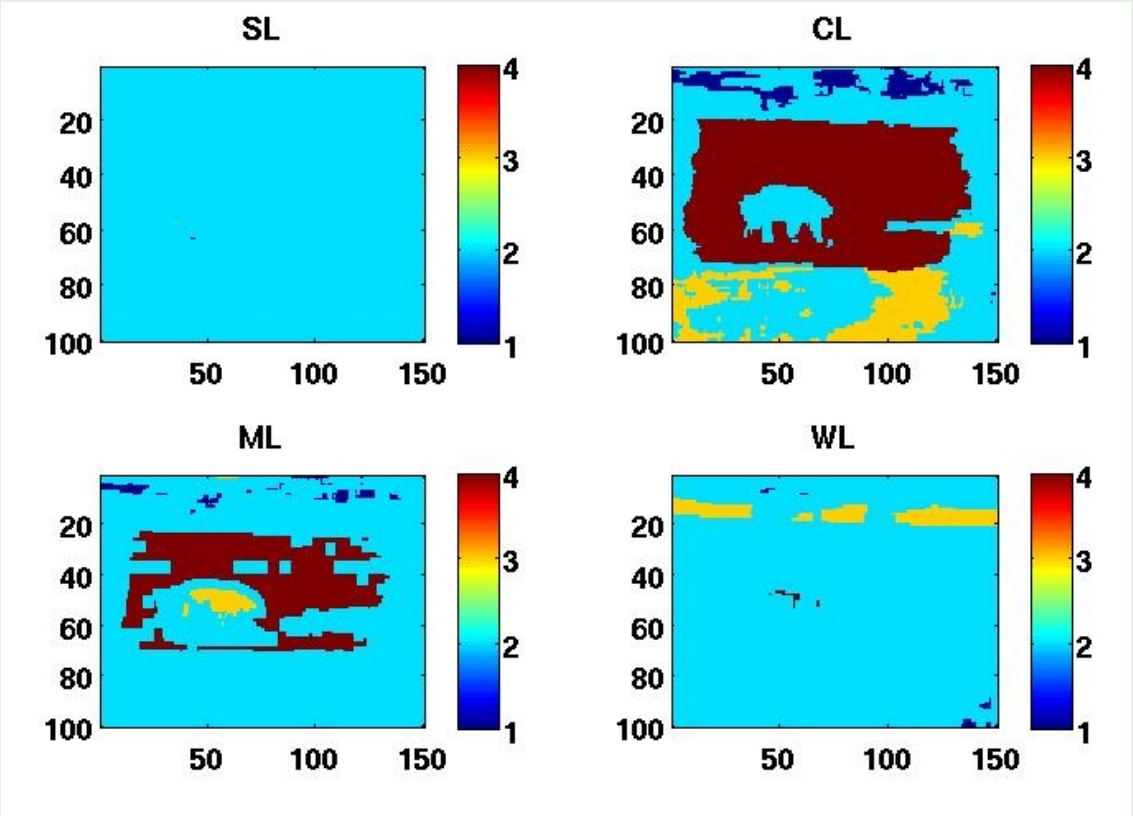
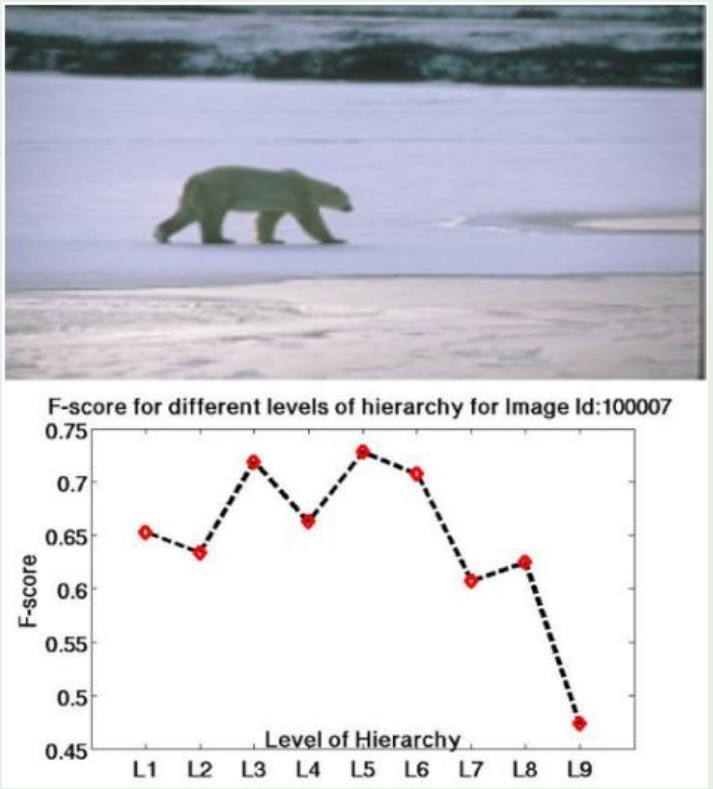


Figure: Color histograms with a 5×5 pixels window around each pixel.

Image Segmentation Result



(a) Best results for other hierarchical techniques where F-score is maximum at $k = 4$.



(b) Original Image & Best F-score at L5 when $k = 19$ for AH-KSC method.

Figure: Hierarchical segmentation results by different methods on image Id:100007.

Sparse KSC (Mall et al, PREMI, 2013 & IJCNN, 2014)

- Data points are projected in the eigenspace & expressed in terms of non-sparse kernel expansions i.e. $\hat{e}^{(l)}(x) = \sum_{i=1}^{N_{tr}} \alpha_i^{(l)} K(x, x_i) + b_l$.
- Goal is to achieve sparse & efficient clustering models using a reduced set $\mathcal{RS}$.
- Propose penalization techniques like **Group Lasso** and L_0 penalizations.
- Result in simpler and faster predictive clustering models.
- Allows to investigate a smaller reduced set to obtain deeper insights.
- **Reduce the time complexity of out-of-sample extensions procedure**.

Sparse Formulations

- $w^{(l)} = \sum_{i=1}^{N_{tr}} \alpha_i^{(l)} \phi(x_i)$ can be approximated by reduced set.
- Objective is to approximate $w^{(l)}$ by a new weight vector $\tilde{w}^{(l)} = \sum_{i=1}^{R} \beta_i^{(l)} \phi(\tilde{x}_i)$ minimizing $||w^{(l)} - \tilde{w}^{(l)}||_2^2$.
- Locating the reduced set $\mathcal{RS}$, $\beta^{(l)}$ co-efficients are obtained by solving the linear system: $\Omega^{\psi\psi}\beta^{(l)} = \Omega^{\psi\phi}\alpha^{(l)}$.
- $\Omega_{mn}^{\psi\psi} = K(\tilde{x}_m, \tilde{x}_n)$, $\Omega_{mi}^{\psi\phi} = K(\tilde{x}_m, x_i)$.

Group Lasso (Yuan et al, JRSS, 2006) Penalization

- The **Group Lasso** formulation for sparse KSC problem is:

$$\min_{\beta \in \mathbb{R}^{N_{tr} \times (k-1)}} \quad \|\Phi^\intercal \alpha - \Phi^\intercal \beta\|_2^2 + \lambda \sum_{l=1}^{N_{tr}} \sqrt{\rho_l} \|\beta_l\|_2, \tag{7}$$

- Regularizer λ controls the amount of sparsity in the model.
- $\hat{\beta}_l$ is zero if $\|\phi(x_l)(\Phi^\intercal \alpha - \sum_{i \neq l} \phi(x_i)\hat{\beta}_i)\| < \lambda$ else $\hat{\beta}_l = (\Phi^\intercal \Phi + \lambda / \|\hat{\beta}_l\|)^{-1} \phi(x_l) r_l$ where $r_l = \Phi^\intercal \alpha - \sum_{i \neq l} \phi(x_i)\hat{\beta}_i$.
- **Group Lasso** is be solved by blockwise co-ordinate descent algorithm with time complexity $O(\text{maxiter} * k^2 N_{tr}^2)$.

Out-of-Sample Prediction Complexity

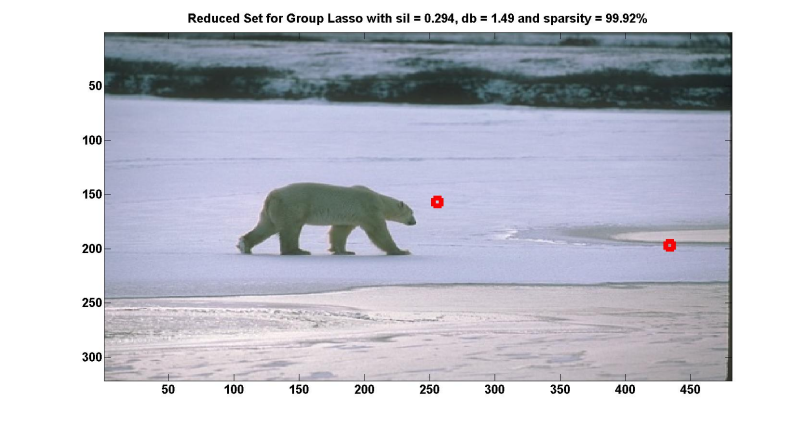
- Normal KSC out-of-sample prediction complexity is $O(N_{tr} N)$.
- Cardinality of reduced set $\mathcal{RS}$ corresponding to **Group Lasso** and L_0-norm be R_1 and R_2 respectively.
- Since $R_i \ll N_{tr}$, the out-of-sample prediction complexity of proposed approaches i.e. $O(R_i N) \ll$ normal KSC.

Sparse KSC (Mall et al, PREMI, 2013 & IJCNN, 2014) Real-Life Experiments

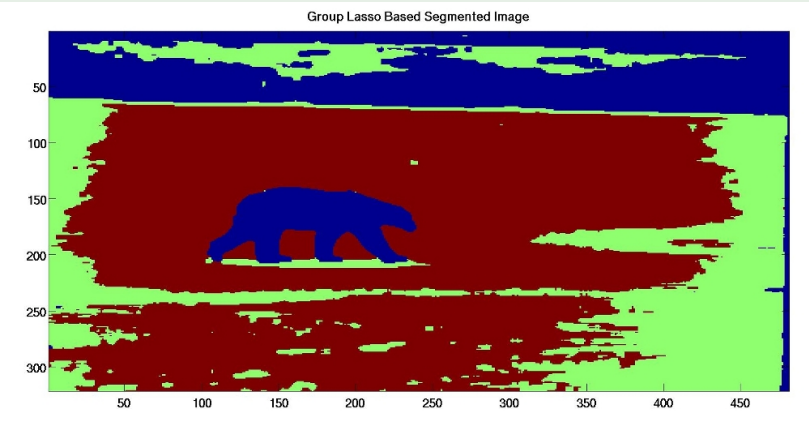
	Group Lasso					L_0 Penalization					L_1 Penalization				
Dataset	*sil*	*db*	Sparsity	λ	Time	*sil*	*db*	Sparsity	ρ	Time	*sil*	*db*	Sparsity	ρ	Time
Breast	0.6824	0.85	**99.5**%	$0.7\lambda_{max}$	**0.99**	0.6898	0.833	96.6%	7.74	19.5	0.6898	0.833	96.6%	7.74	6.7
Bridge	0.423	1.436	**99.0**%	$0.7\lambda_{max}$	**34.8**	**0.596**	1.3	97.1%	4.642	701.0	0.559	1.72	98.5%	5.995	238.4
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
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
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
MLF	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
MLJ	**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
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
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
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
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

Table: Comparison of the sparsest feasible solution for the different penalization methods. Here we highlight the unique best results and measure time in seconds.

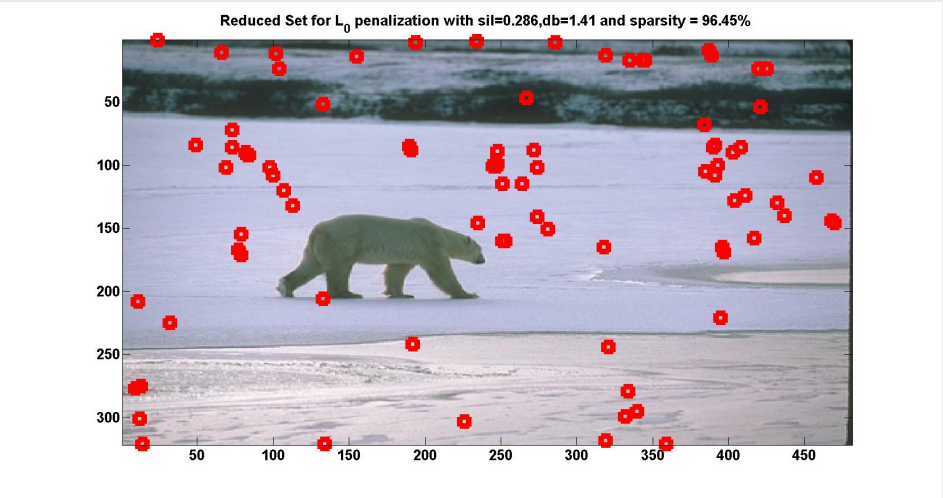
Sparse KSC Image Segmentation



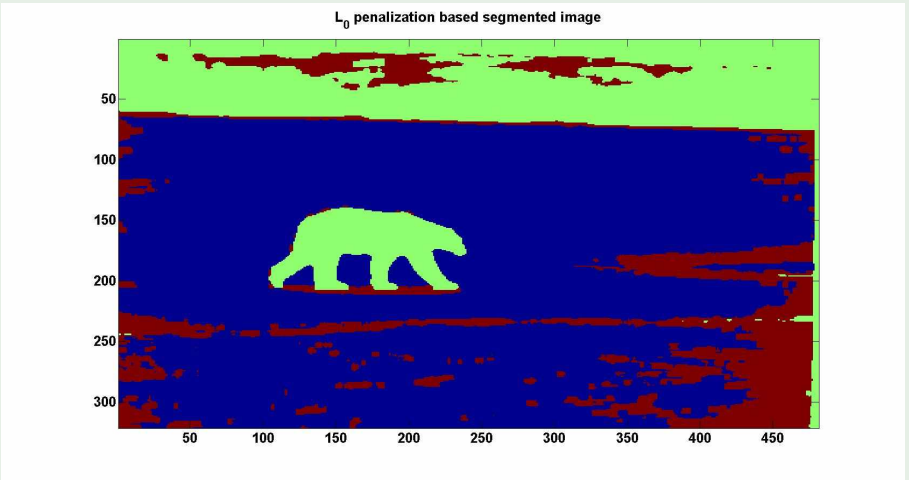
(a) Group Lasso based $\mathcal{RS}$



(b) Best **Group Lasso** Result



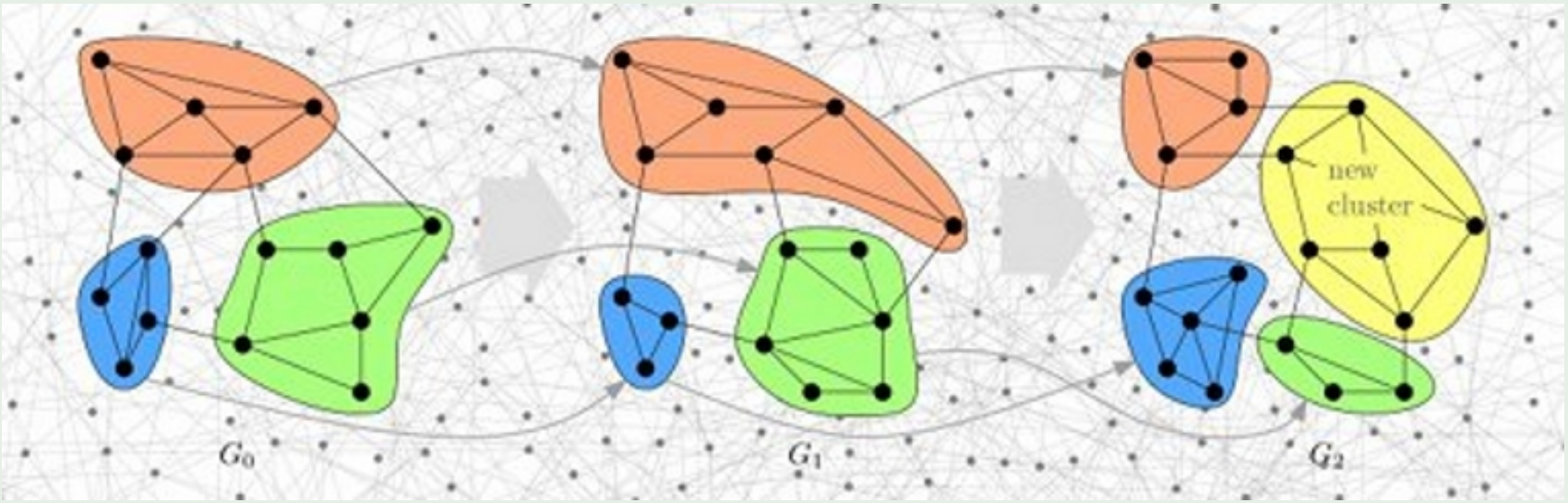
(c) Best L_0 penalization based $\mathcal{RS}$



(d) Best L_0 penalization Result

Figure: Red points are part of reduced set. Fewer points in **Group Lasso** than L_0 due to more pruning by larger regularization constant.

Motivation Example



Design Principles for Netgram

- Each community is represented by a circle whose size is proportional to the number of nodes in that community at time t.
- The evolution of communities between 2 time-stamps is represented by a dashed line.
- Lines follow a straight path if there is no merge or split event during the entire course of evolution for a given community.
- If a part of a community merges into another community then a dashed line is drawn to the community it merges.
- If a part of a community splits into another community then a dashed line is drawn from to the community it merges at time $t + 1$.

Aesthetic Principles (Kwan Ma et al, IEEE TVCG, 2015)

- Minimize line cross-overs.
- Minimize screen space.

Flowchart of Netgram (Mall et al, PLoS One, in submission)

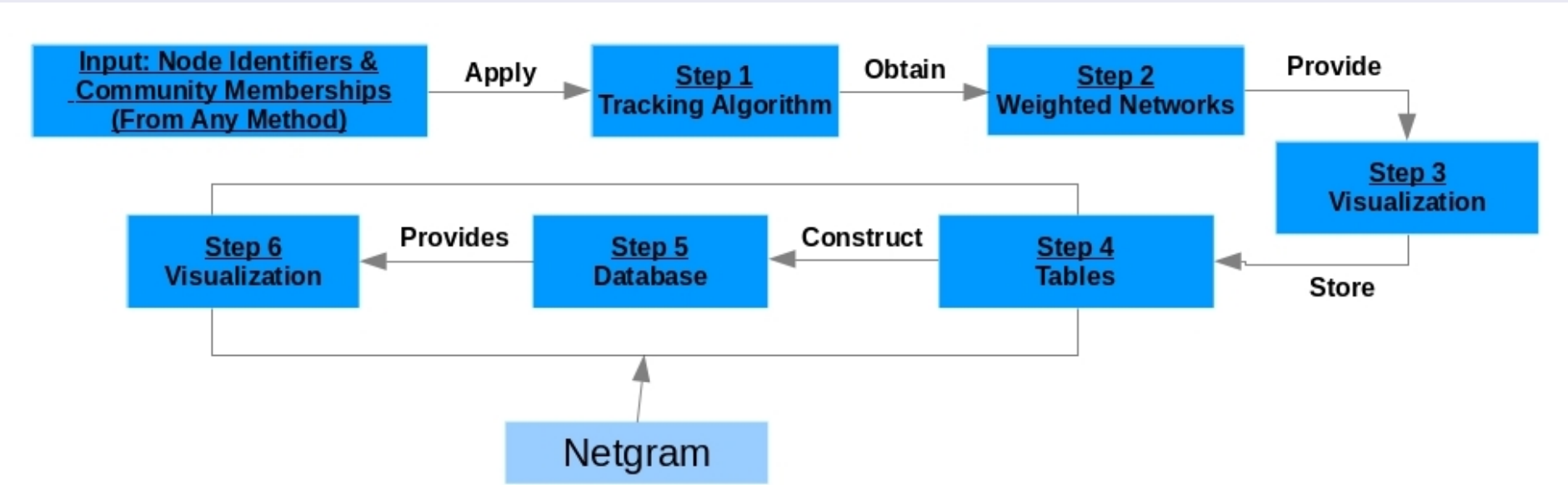


Figure: Steps undertaken by Netgram for visualizing evolution of communities in dynamic networks.

Netgram tool is available at
`ftp://ftp.esat.kuleuven.be/pub/SISTA/rmall/Netgram_Tool.zip`.

Line-Tracking Communities for NIPS Dataset

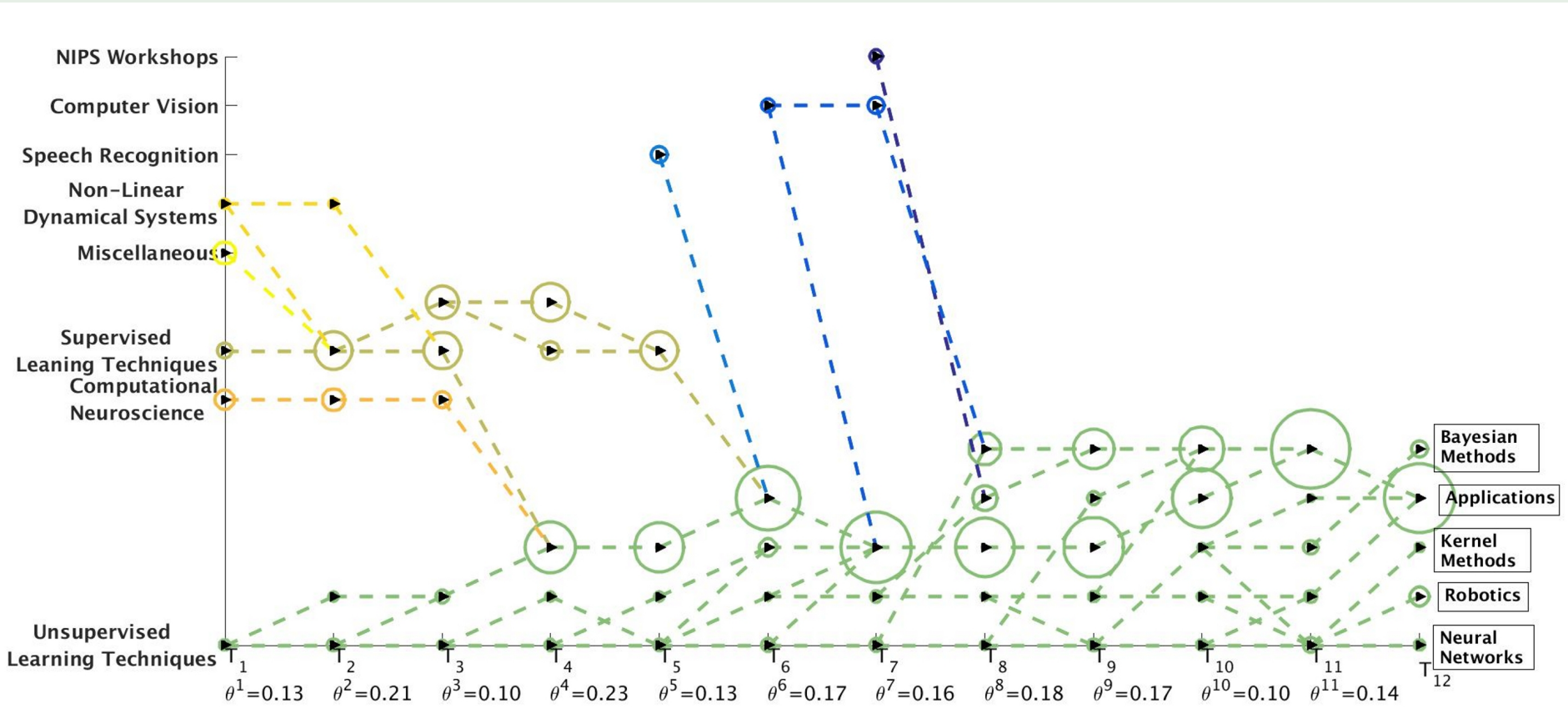
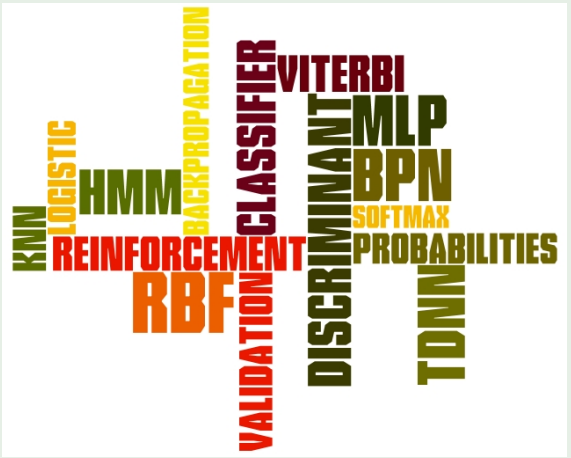


Figure: Visualization of evolution of communities obtained by MKSC algorithm (Rocco et al, Physica A, 2014 & IEEE CIDM, 2014) for the NIPS dataset by Netgram toolkit.

Netgram NIPS Results



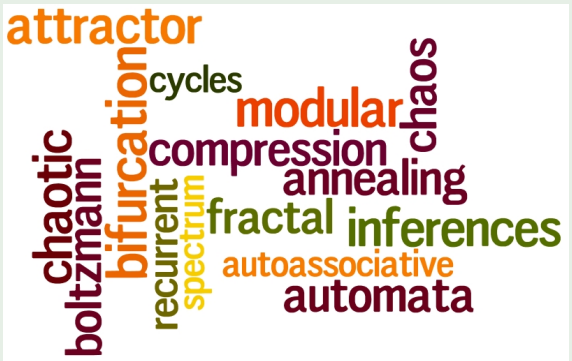
(a) Top words for Unsupervised Learning Techniques cluster



(b) Top words for Supervised Learning Techniques cluster



(c) Top words for Computational Neuroscience cluster



(d) Top words for Non-Linear Dynamics cluster



(e) Top words for Speech Recognition cluster



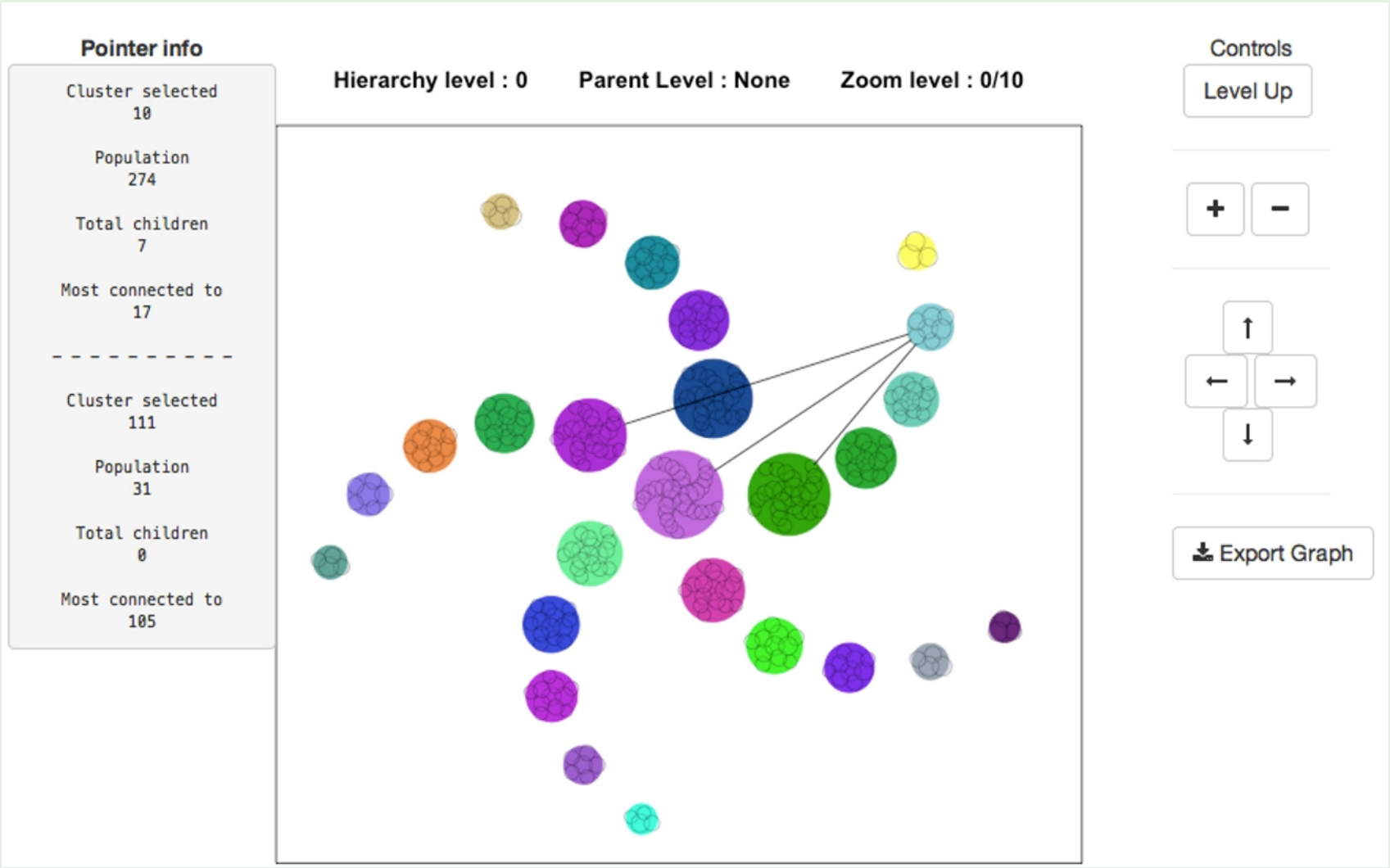
(f) Top words for Computer Vision cluster

Figure: Word clouds corresponding to the top words in communities detected by MKSC algorithm over the first few time-stamps for NIPS dataset.

Future Work

- For supervised tasks, use other methods like Group Lasso in combination with FS-LSSVM to obtain more structured sparse models.
- Extend community detection techniques to networks beyond topographic information & devise methods to handle richer data like social, habitual and professional information.
- Explore the role of kernel based methods in a deep architecture with several layers of non-linearities.
- Multilevel Hierarchical Kernel Spectral Clustering provides an initial direction for this purpose.
- Extend the generic models developed in this thesis to distributed frameworks.

Sample Visualization Scheme



Publications

- Mall R., Suykens J.A.K., "Very Sparse LSSVM Reductions for Large Scale Data", IEEE Transactions on Neural Networks and Learning Systems, vol. 26, no. 5, Mar. 2015, pp. 1086 - 1097.
- Mall R., Langone R., Suykens J.A.K., "Multilevel Hierarchical Kernel Spectral Clustering for Real-Life Large Scale Complex Networks", PLOS One, e99966, vol. 9, no. 6, Jun. 2014, pp. 1-20.
- Mall R., Mehrkanoon S., Suykens J.A.K., "Identifying Intervals for Hierarchical Clustering using the Gershgorin Circle Theorem", Pattern Recognition Letters, vol. 55, no. 1, Apr. 2015, pp. 1-7.
- Mall R., Langone R., Suykens J.A.K., "Kernel Spectral Clustering for Big Data Networks", Entropy, Special Issue: Big Data, vol. 15, no. 5, May 2013, pp. 1567-1586.
- 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, vol. 3, no. 4, Oct. 2013, pp. 1075-1095.
- Mall R., Langone R., Suykens J.A.K., "Ranking Overlap and Outlier Points in Data using Soft Kernel Spectal Clustering", in proc. of ESANN , Brugges, Belgium, 2015.
- Mall R., Jumutc V., Langone R., Suykens J.A.K., "Representative Subsets For Big Data Learning using kNN graphs", in Proc. of the IEEE BigData (BigData), Washington DC, U.S.A., Oct. 2014, pp. 37-42.
- Mall R., Langone R., Suykens J.A.K., "Agglomerative Hierarchical Kernel Spectral Data Clustering", in Proc. of the IEEE Symposium on Computational Intelligence and Data Mining (SSCI CIDM), Orlando, U.S.A, Dec. 2014, pp. 9 - 16.
- Mall R., Langone R., Suykens J.A.K., "Agglomerative Hierarchical Kernel Spectral Clustering for Large Scale Networks", in Proc. of ESANN, Brugges, Belgium, Apr. 2014.
- Mall R., El Anbari M., Bensmail H., Suykens J.A.K., "Primal-Dual Framework for Feature Selection using Least Squares Support Vector Machines", in Proc. of the 19th International Conference on Management of Data (COMAD), Ahmedabad, India, Dec. 2013, pp. 105-108.
- Mall R., Mehrkanoon S., Langone R., Suykens J.A.K., "Optimal Reduced Sets for Sparse Kernel Spectral Clustering", in Proc. of the International Joint Conference on Neural Networks (IJCNN), Beijing, China, Jul. 2014, pp. 2436 - 2443.
- Mall R., Langone R., Suykens J.A.K., "Self-Tuned Kernel Spectral Clustering for Large Scale Networks", in Proc. of the IEEE International Conference on Big Data (IEEE BigData 2013), Santa Clara, United States of America, Oct. 2013.
- Mall R., Suykens J.A.K., "Sparse Reductions for Fixed-Size Least Squares Support Vector Machines on Large Scale Data", in Proc. of PAKDD 2013, GoldCoast, Australia, Apr. 2013, pp. 161-173


J.A.K. Suykens K. De Brabanter, J. De Brabanter and Bart De Moor.

Optimized fixed-size kernel models for large data sets.
Computational Statistics & Data Analysis, 54(6):1484–1504, 2010.


C.C. Chang and C.J. Lin.

Libsvm : a library for support vector machines.
ACM Transactions on Intelligent Systems and Technology, 2(27):1–27, 2011.


J. Lopez, K. De Brabanter, J.R. Dorronsoro, and J.A.K. Suykens.

Sparse lssvms with l_0-norm minimization.
In *Proc. of ESANN 2011*, pages 189–194, 2011.


S.S. Keerthi, O. Chapelle, and D. DeCoste.

Building support vector machines with reduced classifier complexity.
Journal of Machine Learning Research, 7:1493–1515, 2006.


R. Mall and J. A. K. Suykens.

Sparse variations to fixed-size least squares support vector machines for large scale data.
In *Proc. of* 17^{th} *Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD2013)*, pages 161–173, 2013.


R. Mall, R. Langone, and J.A.K. Suykens.

FURS: Fast and unique representative subset selection retaining large scale community structure.
Social Network Analysis and Mining, 3(4):1075–1095, 2013.


V. Blondel, J. Guillaume, R. Lambiotte, and L. Lefebvre.

Fast unfolding of communities in large networks.
Journal of Statistical Mechanics: Theory and Experiment, 10(P10008), 2008.


M. Rosvall and C. Bergstrom.

Maps of random walks on complex networks reveal community structure.
PNAS, 105:1118–1123, 2008.


A. Lancichinetti, F. Radicchi F, J. Ramasco, and S. Fortunato.

Finding statistically significant communities in networks.
PLOS ONE, 6(e18961), 2011.



R. Mall, R. Langone, and J.A.K. Suykens.

Multilevel hierarchical kernel spectral clustering for real-life large scale complex networks.

PLOS One, 9(6), 2014.