Kernel based methods for microarray and mass spectrometry data analysis

Fabian Ojeda

ESAT-SCD-SISTA Division
Department of Electrical Engineering
Katholieke Universiteit Leuven
Leuven, Belgium

May 20, 2011

Prof. dr. ir. B. De Moor, promotor
Prof. dr. ir. J.A.K. Suykens, co-promotor
Prof. dr. ir. P. Sas, chairman
Prof. dr. ir. Y. Moreau
Prof. dr. J. Rozenski
Prof. dr. ir. M. Van Barel
Prof. dr. ir. G. Bontempi, ULB

Outline

1. Background
2. Low rank updated LS-SVM
3. Sparse linear models
4. Entropy based spectral clustering
5. Conclusions

Outline

1 Background

2 Low rank updated LS-SVM

3 Sparse linear models

4 Entropy based spectral clustering

5 Conclusions

Motivation and problem description

Goal

Application of regularization/kernel based methods and adaptation to the areas of high dimensional and low sample/large scale data.

Methods

Prediction models, model selection, variable selection, clustering.

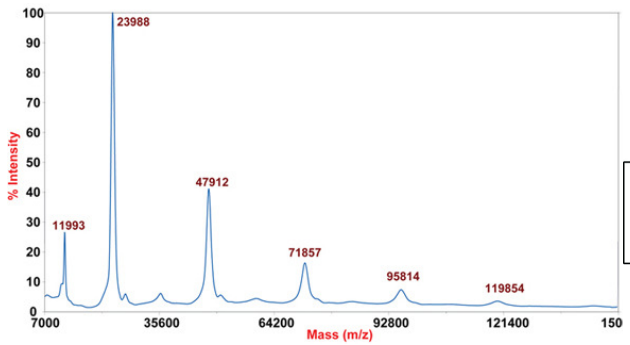
Application

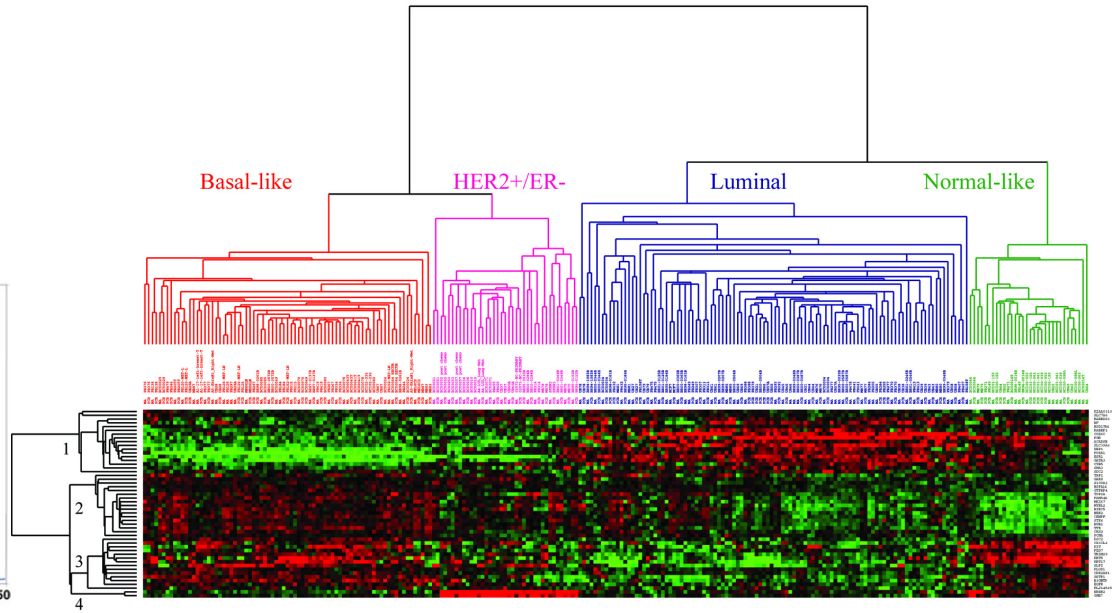
- Efficient variable selection algorithms for microarray data.
- Incorporate structural information of MSI data.
- Clustering methods for large scale gene clustering.

Motivation and problem description

Microarray / mass spectrometry data

- Simultaneous measure of thousands of genes / proteins.
- Structural and prior information.
- Large number of variables, low sample size.
- Irrelevant variables.
- Lack of labeled data.





Regularized learning models

Microarray/MSI data representation

$\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^n$, $\mathbf{x}_i \in \mathbb{R}^d$, $y_i \in \mathbb{R}$, n samples measured over d variables.

$$y_i = \sum_{k=1}^{d} w_k x_i^k + \varepsilon_i \qquad \varepsilon_i \sim \mathcal{N}(0, \sigma^2) \ , \tag{1}$$

- x_i^k: k-th component of $\mathbf{x}_i$.

Solve for $\hat{\mathbf{w}} = (\hat{w}_1, \ldots, \hat{w}_d)^\top \in \mathbb{R}^d$

$$\hat{\mathbf{w}} = \arg\min_{\mathbf{w}} \|\mathbf{y} - \mathbf{X}\mathbf{w}\|_2^2 + \lambda P(\mathbf{w}) \ . \tag{2}$$

- $\lambda > 0$, regularization parameter ($\lambda = 0$, OLS)
- Ridge regression $P(\mathbf{w}) = \|\mathbf{w}\|_2^2$, LASSO $P(\mathbf{w}) = \|\mathbf{w}\|_1$

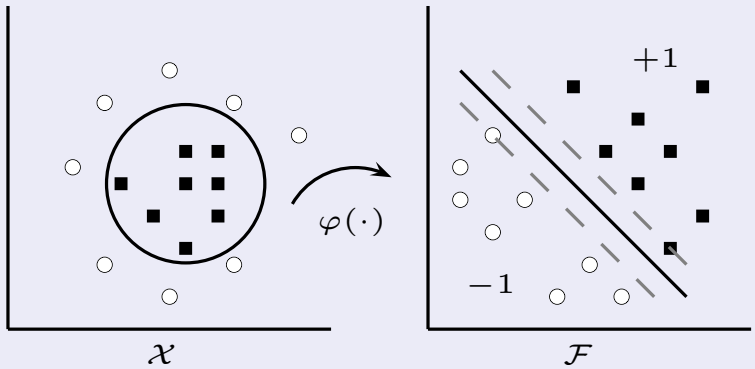
$P(\cdot)$ encodes *a priori* assumptions to make problem well-posed.

Kernel methods

- No assumptions about data structure.
- Allow introduction of prior knowledge.
- Input space $\mathcal{X}$ mapped to high dimensional space $\mathcal{F}$.
- Non-linear general versions of linear algorithms.

Kernel trick

Mapping: $\mathbf{x} \rightarrow \varphi(\mathbf{x})$. Kernel: $K(\mathbf{x}, \mathbf{z}) = \varphi(\mathbf{x})^\top \varphi(\mathbf{z})$.



- Linear: $K(\mathbf{x}, \mathbf{z}) = \mathbf{x}^\top \mathbf{z}$.
- Polynomial: $K(\mathbf{x}, \mathbf{z}) = (\mathbf{x}^\top \mathbf{z} + \tau)^p$, $p \in \mathbb{N}$, $\tau \geq 0$.
- Gaussian: $K(\mathbf{x}, \mathbf{z}) = \exp(-||\mathbf{x} - \mathbf{z}||_2^2 / \sigma^2)$, $\sigma \in \mathbb{R}$ kernel width.

Least Squares Support Vector Machines (LS-SVM)

Optimization problem

Model: $f(\mathbf{x}) = \mathbf{w}^\top \varphi(\mathbf{x}_i) + b$

$$\min_{\mathbf{w},b,e} \frac{1}{2}\mathbf{w}^\top\mathbf{w} + \gamma\frac{1}{2}\sum_{i=1}^{n} e_i^2$$
$$\text{s.t.} y_i = \mathbf{w}^\top\varphi(\mathbf{x}_i) + b + e_i \ , \ i = 1,\ldots,n \ ,$$

Estimate parameters $\mathbf{w} \in \mathbb{R}^{d_h}$ and feature map $\varphi(\cdot) : \mathbb{R}^d \to \mathbb{R}^{d_h}$.

Linear equations: Dual Solve in $\boldsymbol{\alpha} \in \mathbb{R}^n$, via kernel trick

$$\left[\begin{array}{c|c} \boldsymbol{\Omega} + \gamma^{-1}\mathbf{I}_n & \mathbf{1} \\ \hline \mathbf{1}^\top & 0 \end{array}\right]\left[\begin{array}{c} \boldsymbol{\alpha} \\ \hline b \end{array}\right] = \left[\begin{array}{c} \mathbf{y} \\ \hline 0 \end{array}\right] \ .$$

Model: $f(\mathbf{x}) = \text{sign}(\sum_{i=1}^{n} \alpha_i K(\mathbf{x},\mathbf{x}_i) + b)$.

Outline

1 Background

2 Low rank updated LS-SVM

3 Sparse linear models

4 Entropy based spectral clustering

5 Conclusions

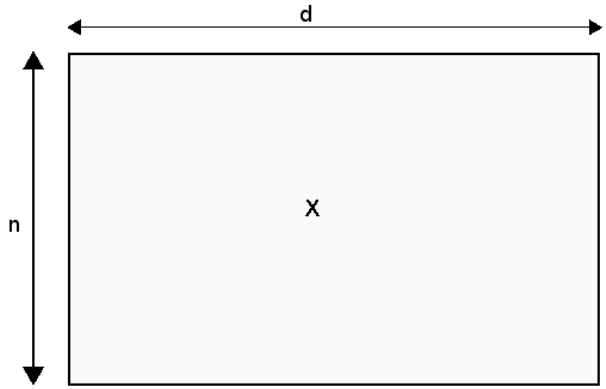
Variable selection problem

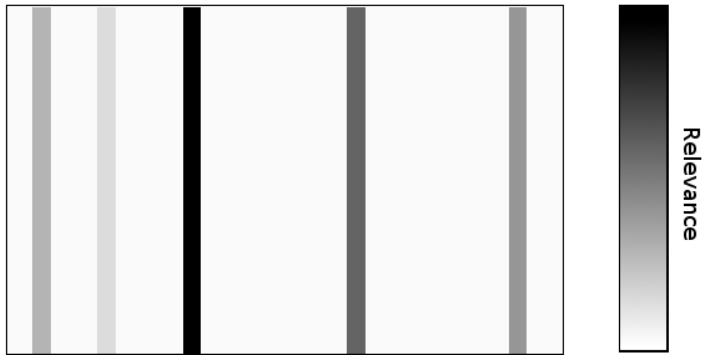
Definition

Given $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^{n}$, $\mathbf{x}_i \in \mathbb{R}^d$, let $\mathcal{S} = \{\mathbf{x}^1, \ldots, \mathbf{x}^k, \ldots, \mathbf{x}^d\}$.
Find $\mathcal{S}^* \subset \mathcal{S}, \mathcal{S}^* \in \mathbb{R}^m$, $m < d$, minimizing $\mathcal{J}_{\mathcal{S}^*} \leq \mathcal{J}_{\mathcal{S}}$, e.g. LOO error.

Elements

- $\mathcal{J}_{\mathcal{S}^*} \rightarrow$ easy/cheap to evaluate.
- Exploit any (if possible) structure of the predictor.
- Reduce computational complexity.





Rank-one updates

Linear kernels can be written in outer product form

$$\boldsymbol{\Omega} = \left[\mathbf{x}^1, \ldots, \mathbf{x}^d\right] \begin{bmatrix} \mathbf{x}^1 \\ \vdots \\ \mathbf{x}^d \end{bmatrix}^\top = \sum_{k=1}^{d} \mathbf{x}^k \mathbf{x}^{k\top}$$

$$\mathbf{H} = \boldsymbol{\Omega} + \gamma^{-1}\mathbf{I}_n = \sum_{k=1}^{d} \mathbf{x}^k \mathbf{x}^{k\top} + \gamma^{-1}\mathbf{I}_n \ .$$

At the level of variable $\mathbf{x}^k$

$$\mathbf{H}_k = \sum_{j=1}^{k-1} \mathbf{x}^j \mathbf{x}^{j\top} + \gamma^{-1}\mathbf{I}_n + \mathbf{x}^k \mathbf{x}^{k\top}$$

$$\mathbf{H}_k = \mathbf{H}_{k-1} + \mathbf{x}^k \mathbf{x}^{k\top} \ . \tag{3}$$

Key point compute $\mathbf{H}_k^{-1}$ from $\mathbf{H}_{k-1}^{-1}$ and obtain $\boldsymbol{\alpha}^*, b^*$.

Low rank updates

With Cholesky factorization $\mathbf{L}\mathbf{L}^\top = \mathbf{\Omega} + \gamma^{-1}\mathbf{I}_n$, then adding new variable $\mathbf{x}^k$ results in a *rank-1* modification to $\mathbf{L}$

$$\tilde{\mathbf{L}}\tilde{\mathbf{L}}^\top = \mathbf{L}\mathbf{L}^\top + \mathbf{x}^k{\mathbf{x}^k}^\top \ . \tag{4}$$

The modified Cholesky factor is

$$\begin{aligned} \tilde{\mathbf{L}}\tilde{\mathbf{L}}^\top &= \mathbf{L}\mathbf{L}^\top + \mathbf{u}\mathbf{u}^\top \\ &= \mathbf{L}(\mathbf{I} + \mathbf{q}\mathbf{q}^\top)\mathbf{L}^\top \\ &= \mathbf{L}\bar{\mathbf{L}}\bar{\mathbf{L}}^\top\mathbf{L}^\top \ , \end{aligned} \tag{5}$$

$\tilde{\mathbf{L}}$ can be directly computed from $\mathbf{L}$. Updated model parameters become:

$$\tilde{b} = \mathbf{1}^\top\tilde{\boldsymbol{\chi}}(\mathbf{1}^\top\tilde{\boldsymbol{\nu}})^{-1} \ , \qquad \tilde{\boldsymbol{\alpha}} = \tilde{\boldsymbol{\chi}} - \tilde{b}\tilde{\boldsymbol{\nu}} \ . \tag{6}$$

where $\tilde{\mathbf{L}}\tilde{\mathbf{L}}^\top\tilde{\boldsymbol{\chi}} = \mathbf{y}$ and $\tilde{\mathbf{L}}\tilde{\mathbf{L}}^\top\tilde{\boldsymbol{\nu}} = \mathbf{1}$.

Experiments

Data

Gene expression data.

- Leukemia. $n = 72$, $d = 7129$.
- Colon cancer $n = 60$, $d = 2000$.

Algorithms

- SVM-RFE with and without retraining
- Naive LS-SVM with forward selection.
- LS-SVM with fast LOO and rank-one modifications.

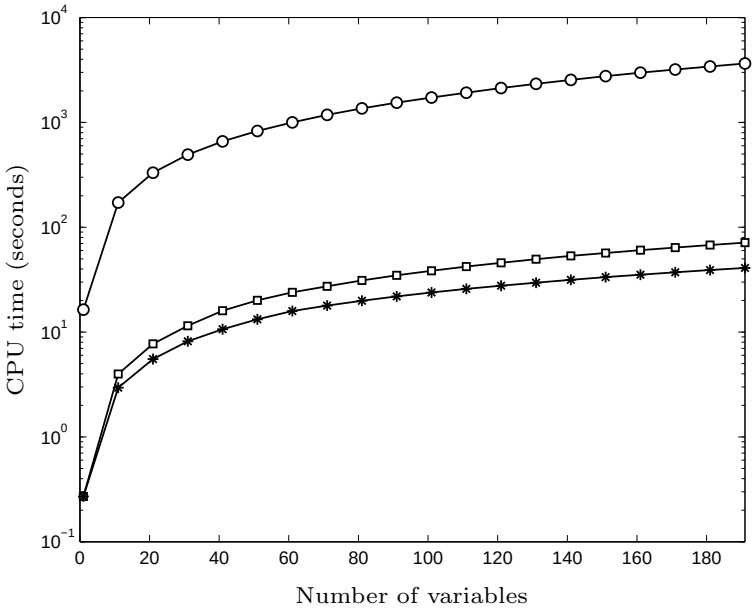
Validation

- Computational complexity.
- 10-fold cross-validation.

Computational time. Colon data set.

Goal: Select 200 and 500 genes.

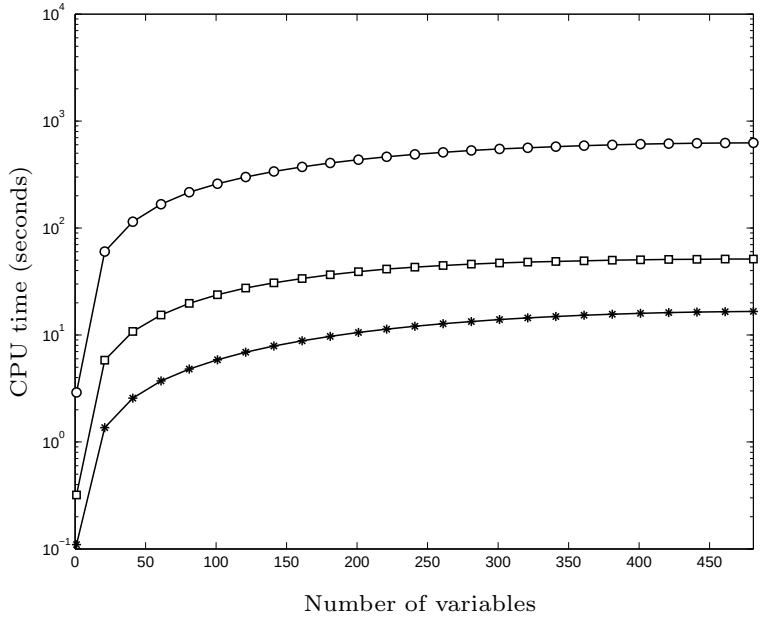
Forward algorithms



Baseline LS-SVM (∘), LOO bound (□), Low rank *updated* LS-SVM (∗)

Improvement by two orders.

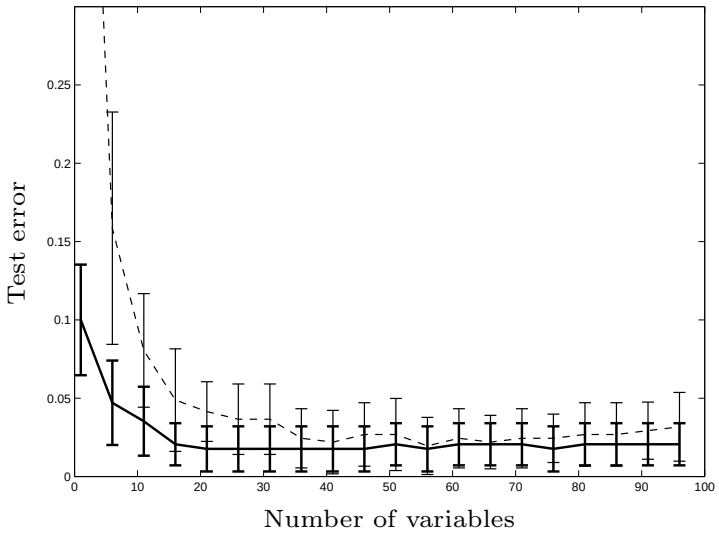
Backward algorithms



SVM-RFE1 (∘), SVM-RFE2 (□), Low rank *downdated* LS-SVM(∗).

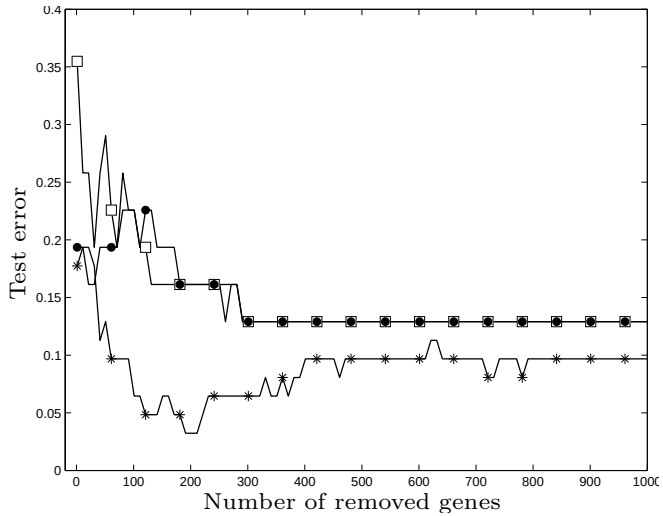
Prediction performance. Test error.

Colon data set



LOO-Bound (Dashed), Low rank *updated* LS-SVM (Solid).

Leukemia data set



SVM-RFE1 (•) and SVM-RFE2 (□), Low rank *downdated* LS-SVM (∗).

Extension to polynomial kernels

Explicit feature map – yield **low rank matrices.**

Explicit feature map $\varphi_p(\cdot)$ for polynomial kernel of degree p

$$\varphi_p(z) = \left[1, \sqrt{\binom{p}{1}} z, \ldots, \sqrt{\binom{p}{p-1}} z^{p-1}, z^p\right]^\top \quad , \tag{7}$$

with $\varphi_p(\cdot) : \mathbb{R} \rightarrow \mathbb{R}^{p+1}$. Hence, Gram matrix becomes

$$\boldsymbol{\Omega}_p^d = \sum_{k=1}^{d} \boldsymbol{\Omega}_p^k \qquad \text{with } \boldsymbol{\Omega}_p^k = \sum_{l=0}^{p} (\varphi_l \circ \mathbf{x}^k)(\varphi_l \circ \mathbf{x}^k)^\top \quad , \tag{8}$$

Matrix notation

$$\boldsymbol{\Omega}_p^k = \Phi_p^k {\Phi_p^k}^\top \quad , \tag{9}$$

$\Phi_p^k = \left[(\varphi_0 \circ \mathbf{x}^k), \ldots, (\varphi_p \circ \mathbf{x}^k)\right]$ is a $n \times (p+1)$ matrix.

Polynomial updates

For a Gram matrix the following holds

$$\operatorname{rank}\left(\Phi_p^k\right) = \operatorname{rank}\left(\Phi_p^k {\Phi_p^k}^\top\right) = \operatorname{rank}\left(\mathbf{\Omega}_p^k\right) \quad , \tag{10}$$

that is $\operatorname{rank}\left(\mathbf{\Omega}_p^k\right) \leq p+1$.
For all inputs, $k = 1, \ldots, d$, $\mathbf{\Omega}_p^d$ is a sum of d rank-$(p+1)$ matrices

$$\operatorname{rank}\left(\mathbf{\Omega}_p^d\right) = \operatorname{rank}\left(\sum_{k=1}^{d} \mathbf{\Omega}_p^k\right) \leq \sum_{k=1}^{d} \operatorname{rank}\left(\mathbf{\Omega}_p^k\right) \quad . \tag{11}$$

Note: For linear kernel $\mathbf{\Omega}^d = \mathbf{\Omega}$, outer product definition.

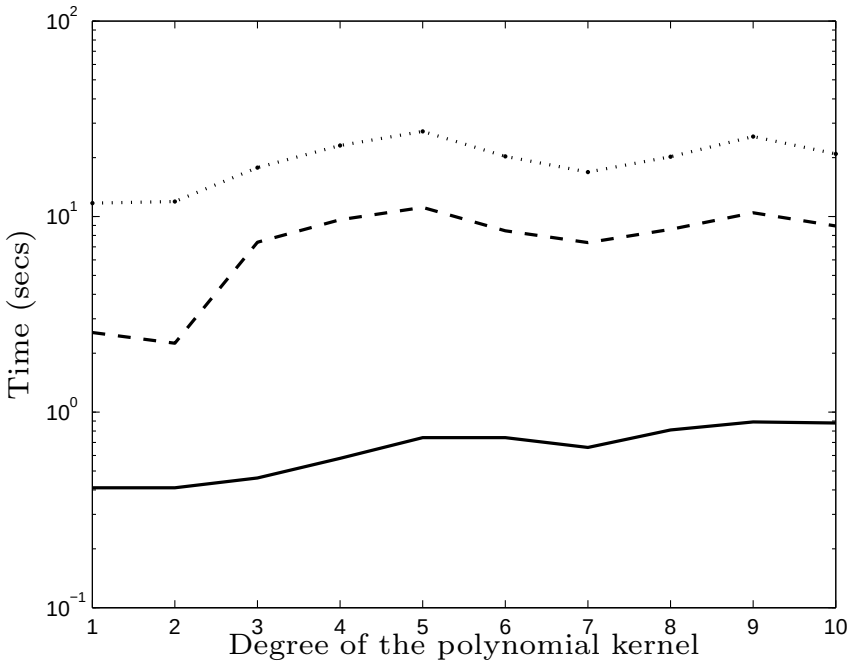
rank-$(p+1)$ updates

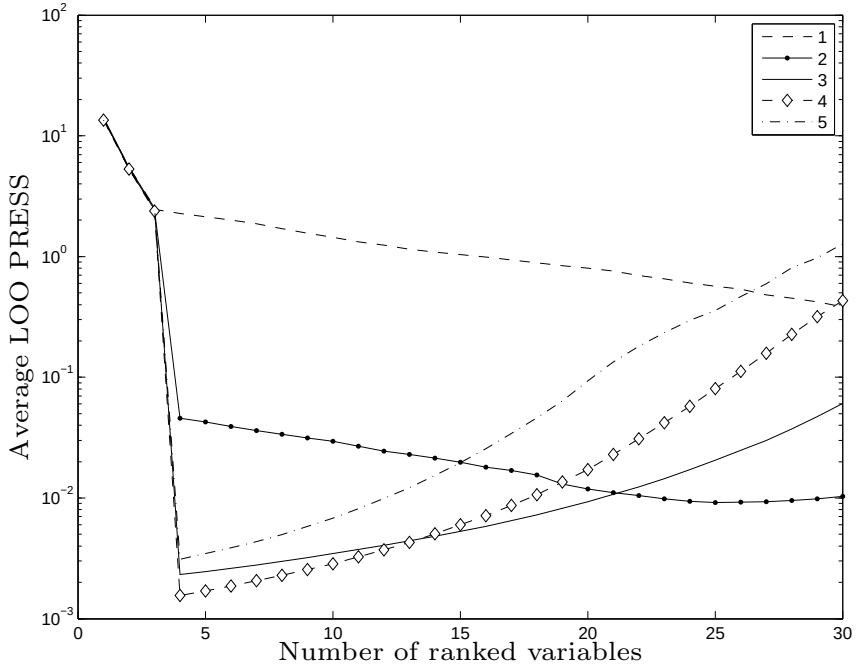
$$\tilde{\mathbf{L}}\tilde{\mathbf{L}}^\top = \mathbf{L}\mathbf{L}^\top + \Phi_p^k {\Phi_p^k}^\top \quad . \tag{12}$$

Apply $(p+1)$ rank-1 updates sequentially over columns of Φ_p^k.

Experimental results

Synthetic data set with $n = 100$ and $d = 500$:
$y_i = 10\ \mathrm{sinc}(x_i^1) + 20(x_i^2 - 0.5)^2 + 10x_i^3 + 5x_i^4 + \epsilon_i,\ \epsilon_i \sim \mathcal{N}(0, 1).$





- Time required to computed 50 updates
- Linear and quadratic model do not retrieve true variables.

Benchmark

Benchmark study: 11 UCI data sets.

	LS-SVM		LowR LS-SVM			SVM		
Kernel	RBF		Linear	Polynomial		RBF		
Algorithm	LS-SVM	LS-SVM ARD	LowR-lin	LowR-poly2	LowR-poly3	$R\|\mathbf{w}\|_2^2$	Span	$\|\mathbf{w}\|_2^2$
BREAST	**26.73 ± 0.47**	29.08 ± 0.41	27.88 ± 5.14	33.34 ± 4.82	34.06 ± 4.90	26.84 ± 4.71	**25.59 ± 4.18**	x
DIABETES	23.34 ± 0.17	24.35 ± 0.19	23.55 ± 1.71	25.46 ± 2.20	25.45 ± 1.93	**23.25 ± 1.70**	**23.19 ± 1.67**	28.50
FLARE	34.22 ± 0.17	34.39 ± 0.19	**33.49 ± 1.70**	33.82 ± 1.65	**33.31 ± 1.78**	x	x	x
GERMAN	**23.55 ± 0.22**	26.10 ± 0.26	**24.97 ± 2.17**	28.29 ± 2.69	28.89 ± 2.59	x	x	x
HEART	16.64 ± 0.36	23.65 ± 0.35	**16.41 ± 3.14**	16.83 ± 3.25	17.85 ± 3.72	**15.92 ± 3.18**	16.13 ± 3.11	27.00
IMAGE	**3.00 ± 0.16**	**1.96 ± 0.11**	18.44 ± 0.72	10.56 ± 1.03	7.99 ± 0.78	x	x	x
RINGNORM	**1.61 ± 0.01**	**2.11 ± 0.04**	25.43 ± 0.55	5.48 ± 0.54	6.18 ± 0.53	x	x	8.40
SPLICE	10.97 ± 0.16	**5.86 ± 0.18**	16.13 ± 0.66	10.55 ± 0.41	**6.91 ± 0.50**	x	x	x
THYROID	4.68 ± 0.23	4.68 ± 0.20	20.24 ± 5.05	18.57 ± 4.61	11.87 ± 4.13	**4.62 ± 2.03**	**4.56 ± 1.97**	x
TWONORM	2.84 ± 0.02	5.18 ± 0.07	**2.53 ± 0.18**	**2.67 ± 0.21**	3.43 ± 0.27	x	x	9.30
WAVEFORM	**9.79 ± 0.04**	13.56 ± 0.14	14.69 ± 0.10	**11.13 ± 0.57**	11.41 ± 0.56	x	x	x

Error rates averaged over 100 test data realizations.

Outline

1 Background

2 Low rank updated LS-SVM

3 Sparse linear models

4 Entropy based spectral clustering

5 Conclusions

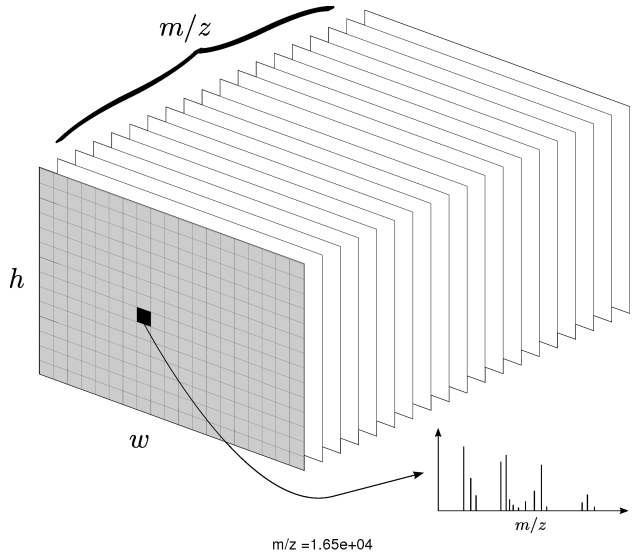
Sparse models for MSI data

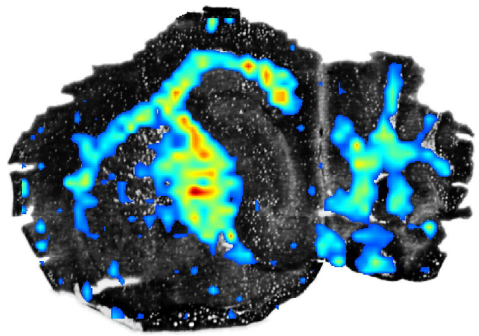
Challenges

- Small fraction of labeled spectra
- Spatial information usually discarded
- Model structural information of MSI data?

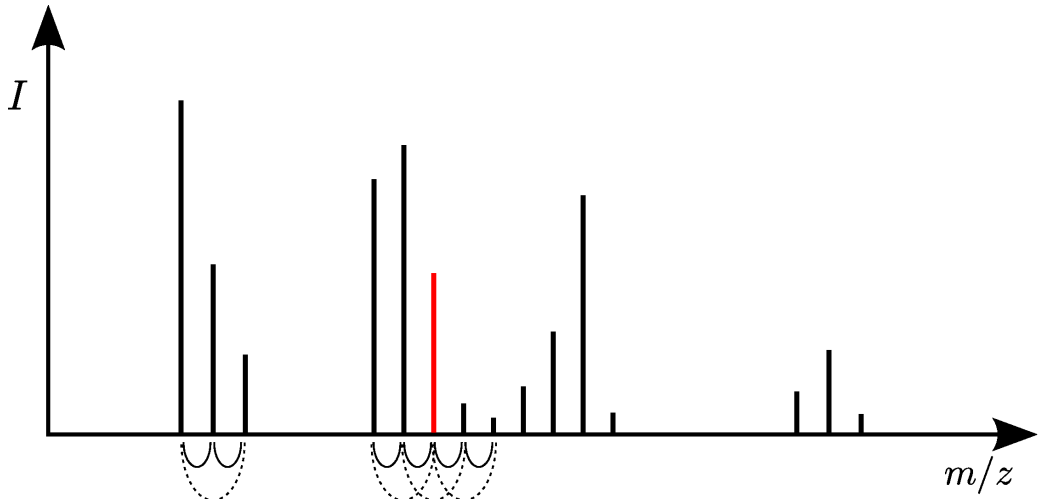
Approach

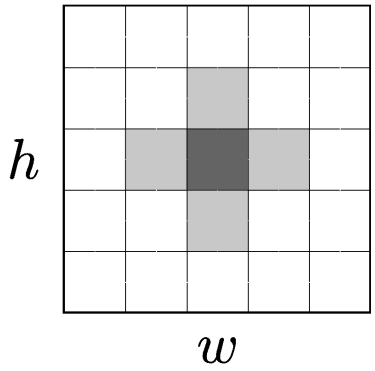
1. Regularized least-squares problem
2. Add extra penalty terms to model
   - ordering of the m/z variables
   - spatial location on tissue of spectra
3. Produce sparse solutions





Structural information in MSI





- *Left*: First order (solid) and second order (dashed) connectivity structure over the set of the variables.
- *Right*: *Cross-like* spatial neighborhood imposed over the set of spectra.

Goal: Encoding *prior* information into the learning framework.

Encoding ordered variables

Every m/z variable $\mathbf{x}^k$ is connected to the preceding $\mathbf{x}^{k-1}$ and the subsequent $\mathbf{x}^{k+1}$.

Graph $\mathcal{G}^p$ over set of variables, a node per $\mathbf{x}^k, k = 1, ..., d$. Thus $L_{\mathbf{w}} \in \mathbb{R}^{d \times d}$ is the Laplacian over the set of variables, and we have

$$\hat{\mathbf{w}} = \arg\min_{\mathbf{w}} \|\mathbf{y} - \mathbf{X}\mathbf{w}\|_2^2 + \lambda_1 \sum_{k=1}^{d} |w_k| + \lambda_2 \mathbf{w}^\top L_{\mathbf{w}} \mathbf{w} \ , \qquad (13)$$

Regularization

- $\lambda_1 > 0$, second term enforces sparsity on $\mathbf{w}$,
- $\lambda_2 > 0$, third term smooths $\mathbf{w}$ on the network.

Encoding prior spatial information

Each spectrum is a node in $\mathcal{G}^s$, $L_s \in \mathbb{R}^{n_s \times n_s}$ is constructed using *cross-like* neighborhood pattern.

Introducing penalty for predicted responses, *i.e.* $\hat{\mathbf{y}}^\top L_s \hat{\mathbf{y}}$, then

$$\hat{\mathbf{w}} = \arg\min_{\mathbf{w}} \|\mathbf{y} - \mathbf{X}\mathbf{w}\|_2^2 + \lambda_1 \sum_{k=1}^{d} |w_k| + \lambda_2 \mathbf{w}^\top L_{\mathbf{w}} \mathbf{w} \tag{14}$$

$$\text{s.t.} \quad \hat{\mathbf{y}}^\top L_s \hat{\mathbf{y}} \leq \xi, \tag{15}$$

$$\hat{y}_j = \sum_{k=1}^{d} w_k x_j^k, \; j = 1, \ldots, n_s \; . \tag{16}$$

where $\xi > 0$ controls the influence of this term.

Encoding prior spatial information (cont.)

Expressing $\mathbf{w}^\top(\mathbf{X}_s)^\top L_s(\mathbf{X}_s)\mathbf{w} = \mathbf{w}^\top G_s \mathbf{w}$ and defining $H_{\lambda_3} = L_{\mathbf{w}} + \frac{\lambda_3}{\lambda_2} G_s$, with $\lambda_3 > 0$

$$\hat{\mathbf{w}} = \arg\min_{\mathbf{w}} \|\mathbf{y} - \mathbf{X}\mathbf{w}\|_2^2 + | + \|\mathbf{0}_{d\times 1} - \sqrt{\lambda_2} H_{\lambda_3}^{1/2}\mathbf{w}\|_2^2 + \lambda_1 \sum_{k=1}^{d} |w_k| \ ,$$

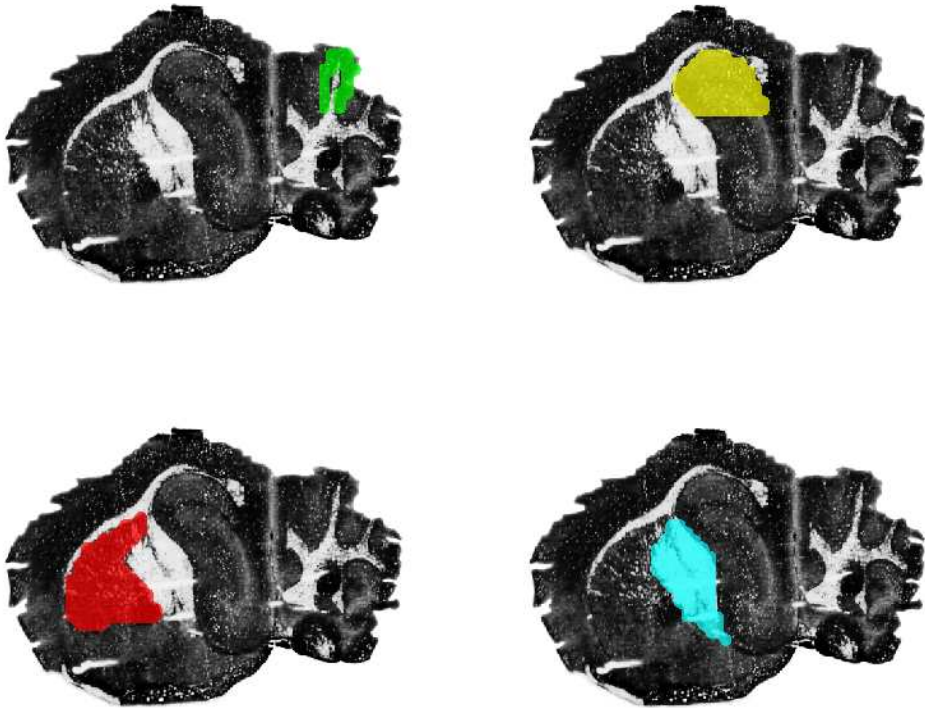
$$= \left\| \begin{bmatrix} \mathbf{y} \\ \mathbf{0}_{d\times 1} \end{bmatrix} - \begin{bmatrix} \mathbf{X} \\ \sqrt{\lambda_2} H_{\lambda_3}^{1/2} \end{bmatrix} \mathbf{w} \right\|_2^2 + \lambda_1 \sum_{k=1}^{d} |w_k| \ . \qquad (17)$$

- Modified problem with dimensions $(n + d) \times d$.
- Solution obtained via LARS (Least Angle Regression) algorithm.
- $\lambda_3 = 0$ and $L_{\mathbf{w}} = \mathbf{I}$ corresponds to Elastic net (ENET) algorithm.

Experiments

- Data set: sagittal section of mouse brain.
- Spatial grid of 51×34 measurements (i.e. 1734 pixels).
- Measurements from 2800 to 25000 Da in 6490 (m/z) bins.
- Data structure: $n = 1734$ mass spectra, $d = 6490$ m/z variables.
- Partial labels for 279 spectra of four anatomical regions

cerebellar cortex	(cc)
Ammon's horn	(ca)
cauda-putamen	(cp)
lateral ventricle	(vl)

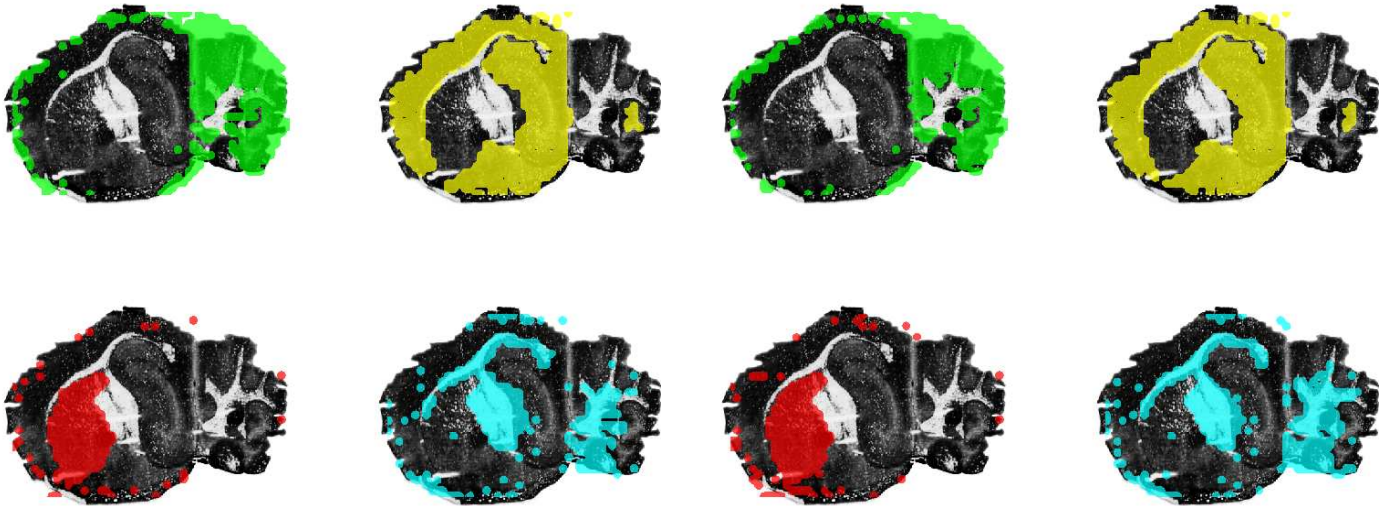


Multi-class ones-vs-one results

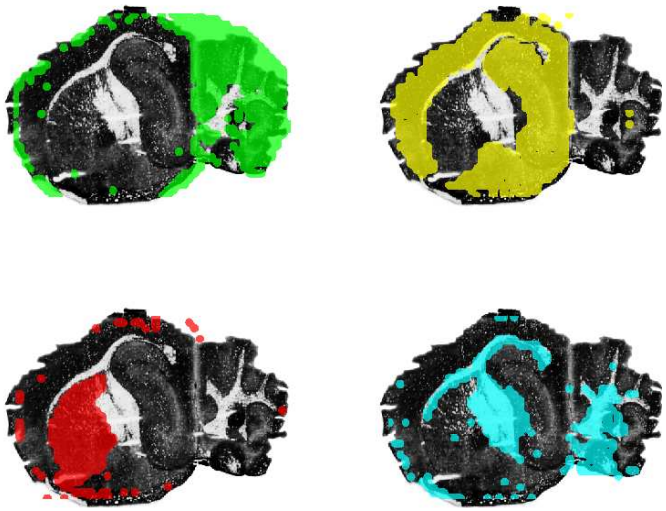
Classes	10-fold accuracy					
	LASSO		ENET		Proposed	
cc vs *ca*	1.0000	(0.0)	1.0000	(0.0)	1	(0)
cc vs *cp*	0.9738	(0.0532)	0.9905	(0.0202)	0.9288	(0.1076)
cc vs *vl*	0.9250	(0.1208)	0.9333	(0.1097)	0.8938	(0.1719)
ca vs *cp*	0.9740	(0.0436)	0.9687	(0.0477)	0.9758	(0.0319)
ca vs *vl*	0.9143	(0.0732)	0.9330	(0.0549)	0.9446	(0.0447)
cp vs *vl*	0.9142	(0.0739)	0.9123	(0.0766)	0.9288	(0.0580)

Table: Multi-class one-vs-one and 10-fold cross-validation results.

Visualization



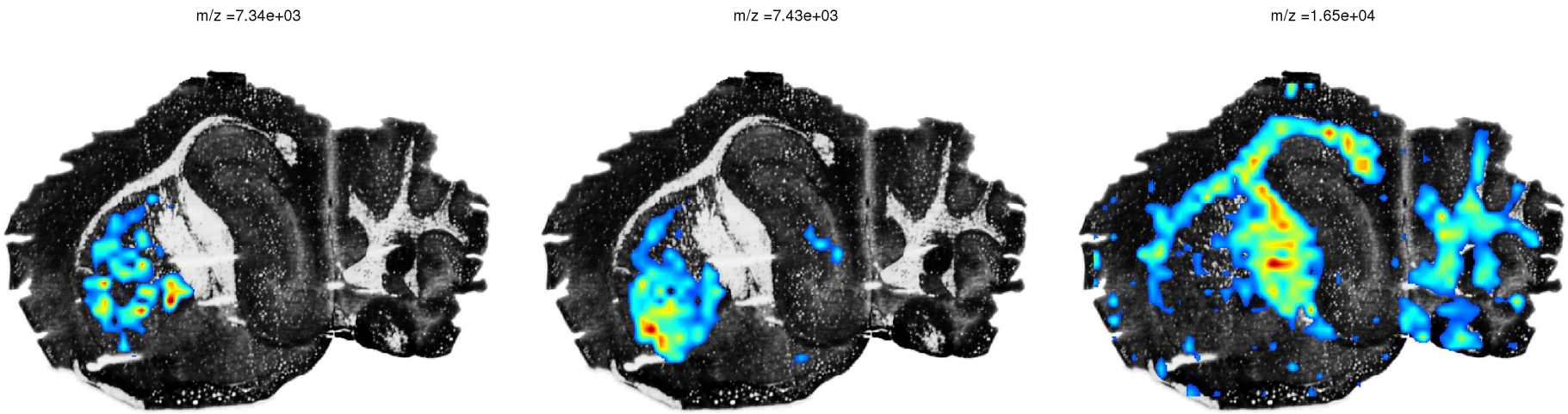
(a) LASSO (b) ENET



(c) Proposed

Visualization (cont.)

- Clear differentiation of (vl) and (cp) from surrounding.
- (vl) delineates to elongated corpus callosum and cerebellar nucleus.
- (cc) exceed intended boundaries, few labeled examples (21)
- (ca) extends to capture complete hippocampus.



(d) $m/z = 7.34 \times 10^3$ (e) $m/z = 7.43 \times 10^3$ (f) $m/z = 1.65 \times 10^4$

Figure: Selected m/z variables discriminating the (cc) and (vl) tissue regions. Left most variable at $m/z = 1.65e+4$ Da only appears in the proposed model.

Outline

1 Background

2 Low rank updated LS-SVM

3 Sparse linear models

4 Entropy based spectral clustering

5 Conclusions

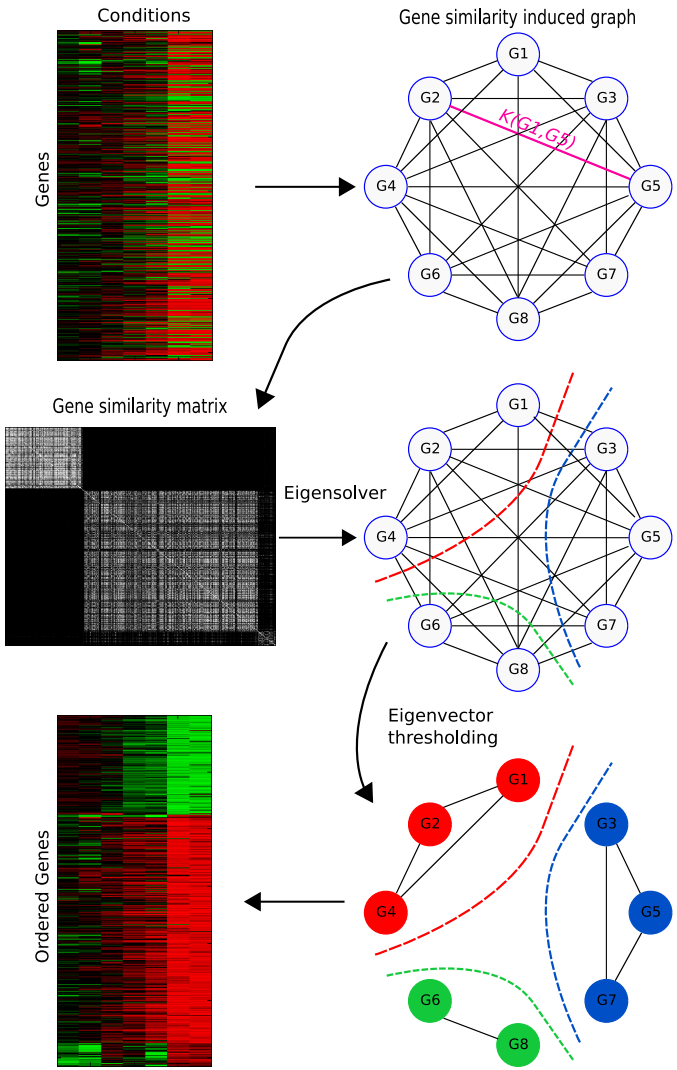
Entropy based selection for spectral clustering

Challenges

- Traditionally clustering uses full set of genes
- How to assign future test genes?
- How to perform training/validation? Prevent overfitting?

Approach

1. Select genes via entropy maximization
2. Build clustering model on subsample
3. Tune model parameters
4. Infer cluster for remaining genes



Entropy selection

Underlying density distribution where $\mathbf{\Omega}_h \in \mathbb{R}^{m \times m}$ Parzen estimator.

$$H_S = -\log \int p(x)^2 \mathrm{d}x$$

$$H_S \approx H_R = \int \hat{p}(x)^2 \mathrm{d}x = \frac{1}{m^2} \mathbf{1}^\top \mathbf{\Omega}_h \mathbf{1} \ .$$

Spectral clustering - weighted kernel PCA

Weight matrix $\mathbf{V} = \mathbf{D}^{-1}$ and $\Phi = [\varphi(\mathbf{x}_1)^\top; ...; \varphi(\mathbf{x}_n)^\top]$

$$\max_{\mathbf{w},\mathbf{z}} \quad \gamma \frac{1}{2} \mathbf{z}^\top \mathbf{V} \mathbf{z} - \frac{1}{2} \mathbf{w}^\top \mathbf{w}$$

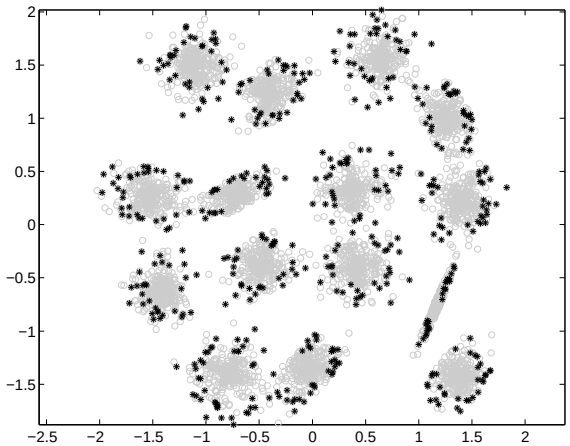
$$\text{s.t.} \quad \mathbf{z} = \Phi \mathbf{w} \ , \ \mathbf{V} = \mathbf{V}^\top > 0 \ ,$$

Eigenvalue problem: $\mathbf{V}\mathbf{\Omega}\boldsymbol{\alpha} = \lambda \boldsymbol{\alpha}$, with $\mathbf{\Omega}_{ij} = K(\mathbf{x}_i, \mathbf{x}_j), \lambda = 1/\gamma$.

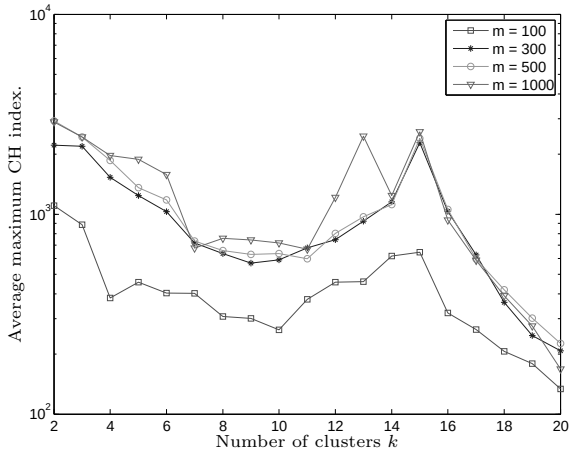
For a new sample $\mathbf{x}$, $z^{(r)}(\mathbf{x}) = \sum_{i=1} \alpha_i^{(r)} K(\mathbf{x}_i, \mathbf{x}), r = 1, \ldots, k$.

Simulated data – proof of concept

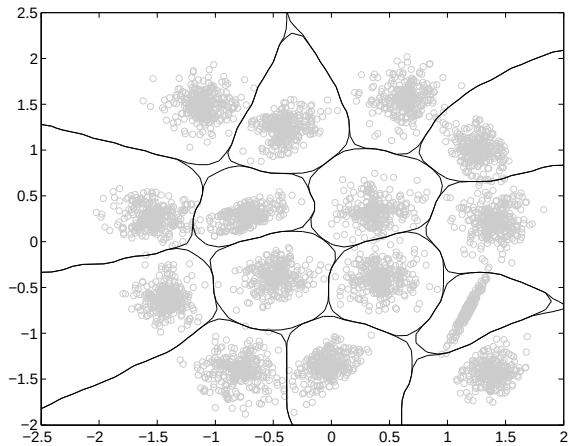
Data set with $k = 15$ clusters, $n = 15000$, $m = 500$.



entropy

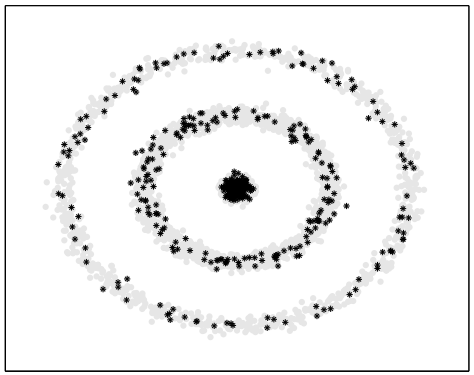


number clusters

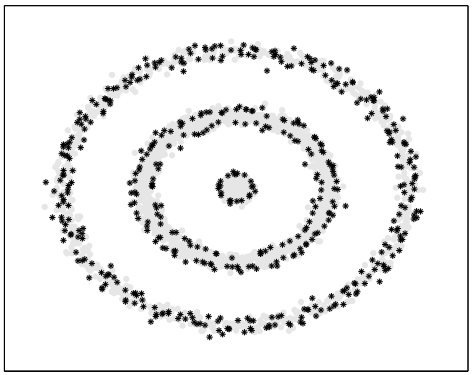


clustering

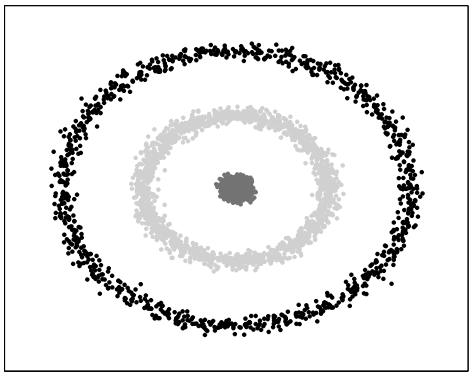
Non-linear example: $n = 7000$, $m = 500$.



random



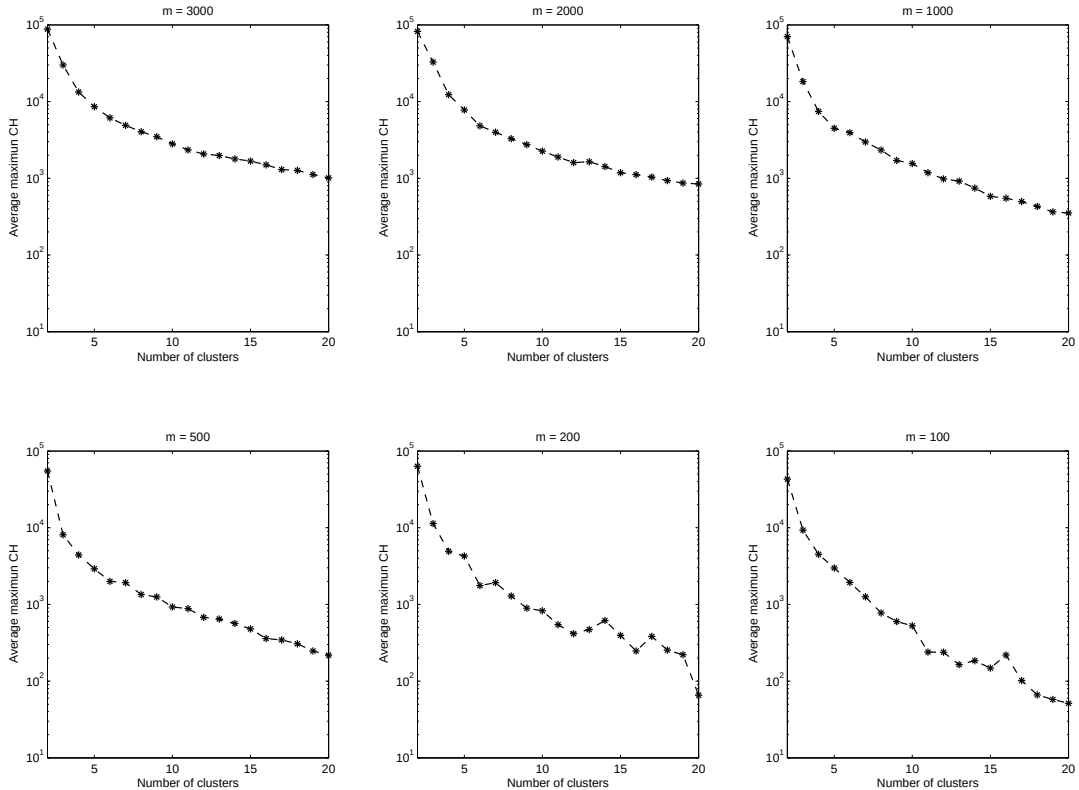
entropy



clustering

NCI60 microarray data set

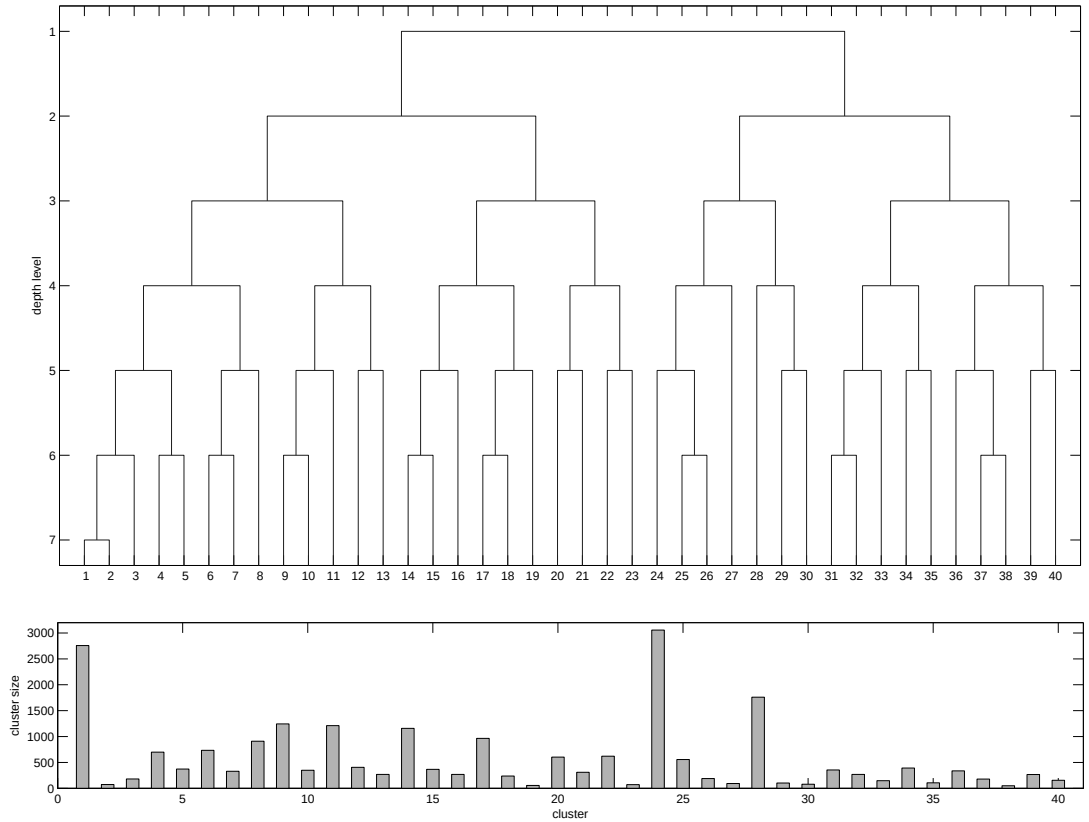
- Genes $n = 22000$, arrays $d = 108$.
- $m \in \{100, ..., 3000\}$, $k \in \{2, ..., 20\}$, $\sigma \in [10^{-3}, ..., 500] \times \sqrt{d}$
- 20 randomizations



Subtle clusters obfuscated by two large groups.

NCI60 – Recursive spectral clustering

- maxdepth $= 10$, $k_{\max} = 5$, $m_{\min} = 100$, $m_{\max} = 1000$
- 20 randomizations. Different resolutions, adjusted σ range.



- Bipartition at each level. Total of 40 clusters on the leaves.
- Three major clusters account for 33% of data set.

Outline

1 Background

2 Low rank updated LS-SVM

3 Sparse linear models

4 Entropy based spectral clustering

5 Conclusions

Conclusions

- Low rank-updated LS-SVM for variable selection
  - Linear kernels
  - Componentwise polynomial kernels
  - Efficient LOO computation
- Sparse models for MSI data
  - Partially labeled data
  - Encoding of structural data properties
  - Additional constraints – modified optimization problem
- Entropy based selection for spectral clustering
  - Model selection
  - Recursive implementation
  - Large scale data

Acknowledgments

- Kernel methods & optimization
  - Carlos Alzate
  - Marco Signoretto
  - Tillmann Falck
  - Kris De Brabanter
- MSI data
  - Raf Van De Plas
  - ProMeTa