ELD411: Graph based Optimization, Feature selection and Learning

Pranjal Rai (2018EE10484)
Shauryasikt Jena (2018EE10500)
Tanvir Singh Bal (2018EE10508)

under the guidance of

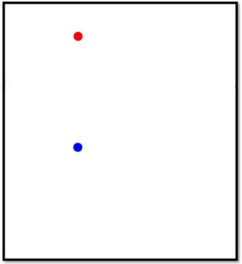
Prof. Sandeep Kumar **Prof. Jayadeva**

Contents

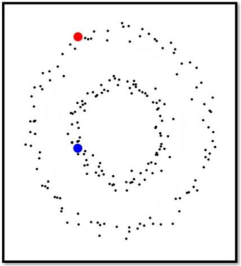
Motivation

Data : $\frac{l}{2}$ positive samples , $\frac{l}{2}$ negative samples , u unlabelled samples

What should be the prior ?



Supervised

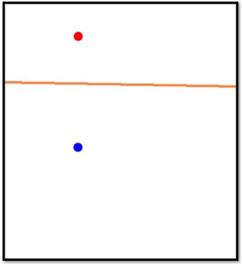


Semi Supervised

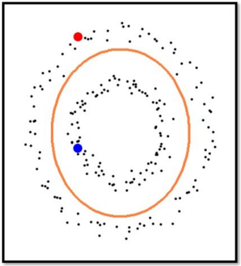
Motivation

Data : $\frac{l}{2}$ positive samples , $\frac{l}{2}$ negative samples , u unlabelled samples

What should be the prior ?



Supervised



Semi Supervised

Intrinsic geometry of the marginal, P_X $\Longrightarrow$ Prior belief

How to incorporate the information related to intrinsic geometry of P_X

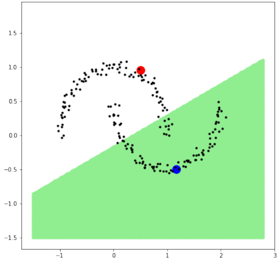
$\Longrightarrow$ Using Manifold Regularization [BNS06]

What we achieved ?

1. Geometric semi-supervised classifier, by minimizing the VC dimension - Laplacian Minimal Complexity Machines (LapMCM)

2. Graph Trend Filtering based geometric frame work for semi-supervised learning
   - Trend Filtered Minimal Complexity Machines (TFMCM)

3. Applications of our minimal algorithms,
   - Feature selection through unlabelled samples
   - Regression and Manifold learning



Manifold Regularization

Conventional Learning framework:

$$f^* = \arg\min_{f \in \mathcal{H}_K} \frac{1}{l} \sum_{i=1}^{l} V(x_i, y_i, f) + \gamma_A \| f \|_A^2 \quad (1)$$

Intrinsic norm using Manifold Regularization [BNS06] :

$$f^* = \arg\min_{f \in \mathcal{H}_K} \frac{1}{l} \sum_{i=1}^{l} V(x_i, y_i, f) + \gamma_A \| f \|_A^2 + \gamma_I \| f \|_I^2 \quad (2)$$

Graph-Laplacian based intrinsic norm [BNS06] :

$$f^* = \arg\min_{f \in \mathcal{H}_K} \frac{1}{l} \sum_{i=1}^{l} V(x_i, y_i, f) + \gamma_A \| f \|_A^2 + \frac{\gamma_I}{(u+l)^2} f^T L f \quad (3)$$

Laplacian Minimal Complexity Machines - LapMCM

We propose the following optimization problem that incorporates unlabeled examples in to the classifier along with minimizing the VC dimension, inspired by MCM [Jay15]

$$\min_{h,\lambda,b,q} \quad \frac{h^2}{2} + C\frac{1}{2l}\sum_{i=1}^{l} q_i^2 + \frac{\gamma_A}{2}\lambda^T K\lambda + \frac{\gamma_I}{l+u}\lambda^T KLK\lambda \tag{4}$$

such that,

$$h \geq y_i\Big(\sum_{j=1}^{l+u}\lambda_j K(x_i, x_j) + b\Big)$$

$$y_i\Big(\sum_{j=1}^{l+u}\lambda_j K(x_i, x_j) + b\Big) + q \geq 1$$

$$\forall i = 1, 2, 3, ..., l$$

On solving the dual reduces to the following problem:

$$\alpha^* = \arg\max \quad e^T\alpha - \frac{1}{2}\alpha^T Q\alpha \tag{5}$$

such that,

$$A\alpha = 0, \;\; \alpha \geq 0$$

$\implies$ Unconstrained Problem $\implies$ use **SUMT**[Jos+12]

Sequential Unconstrained Minimization Technique (SUMT)

SUMT[Jos+12] technique was used for SVM solvers, we incorporated it in our problem,

$$\min \quad f(x) \tag{6}$$
$$\text{such that,} \;\; \forall\, j = 1, 2, 3, ..n$$
$$h_j(x) = 0$$

$$\min \quad E_p(x) = f(x) + \sum_{j=1}^{n} \alpha_p \; h_j^2(x) \tag{7}$$

1. Set $p = 0$. Choose the coefficient α_0, and an initial state x_0
2. Find the minimum of $E_p(x)$. Denote the solution as x_p^*
3. If all the constraints in the original problem are satisfied, stop
4. If not, choose x_p^* as the new initial state, and choose α_{p+1} such that $\alpha_{p+1} > \alpha_p$. Set $p = p + 1$. Go to step 2
5. In the limit, as $p \to \infty$, the sequence of minimas $x_1^*, x_2^*, ...x_p^*, ...$ will converge to the solution of the original problem

Unconstrained LapMCM

The primal objective function for LapMCM is as follows:

$$\min_{h,\lambda,b,q} \quad \frac{h^2}{2} + \frac{C}{2l}\sum_{i=1}^{l} q_i^2 + \frac{\gamma_A}{2}\lambda^T K\lambda + \frac{\gamma_I}{l+u}\lambda^T KLK\lambda + \frac{p}{2}b^2 \tag{8}$$

such that,

$$h \geq y_i\Big(\sum_{j=1}^{l+u}\lambda_j K(x_i, x_j) + b\Big)$$

$$y_i\Big(\sum_{j=1}^{l+u}\lambda_j K(x_i, x_j) + b\Big) + q \geq 1$$

$$\forall i = 1, 2, 3, ..., l$$

On solving the dual reduces to the following problem:

$$\alpha^* = \arg\max \quad e^T\alpha - \frac{1}{2}\alpha^T Q\alpha \tag{9}$$

such that,

$$\alpha \geq 0$$

implies Unconstrained problem $\implies$ solve using **Newton's method**

Training a LapMCM model

Algorithm 1: Laplacian Minimal Complexity Machines

Input: l labelled samples $\{(x_i, y_i)\}_{i=1}^{l}$ and, u un-labelled samples $\{x_j\}_{j=l+1}^{l+u}$

Output: $f(x) = \sum_{j=1}^{l+u} \lambda_j K(x, x_j) + b : \mathcal{R}^n \rightarrow \mathcal{R}$

Data Distance or connectivity graph Graph-Laplacian

Choose hyper-parameters and compute the Gram matrix K_{ij} such that $K_{ij} = K(x_i, x_j)$ (K: Kernel function)

Compute various helper matrices and Q

Minimize $\frac{1}{2}\alpha^T Q \alpha - e^T \alpha$ by calling the *optimize* function

Function `optimize`(Q, *numItr*):

    Randomly Initialize the vector α
    for *k in length(α)* **do**
        for *itr in range(numItr)* **do**
            $\alpha_k(itr+1) \longleftarrow \alpha_k(itr) + \frac{(1 - Q[k,:]^T \alpha)}{Q[k,k]}$ **if** $\alpha_k(itr+1) < 0$ **then**
                $\alpha_k(itr+1) \longleftarrow 0$
                break
            end
        end
    end
    return α

End Function

Compute EFS vector, λ from α

$f(x) = \sum_{j=1}^{l+u} \lambda_j K(x, x_j) + b$ and the predicted class, $y = sgn(f(x))$

LapMCM Vs LapSVM

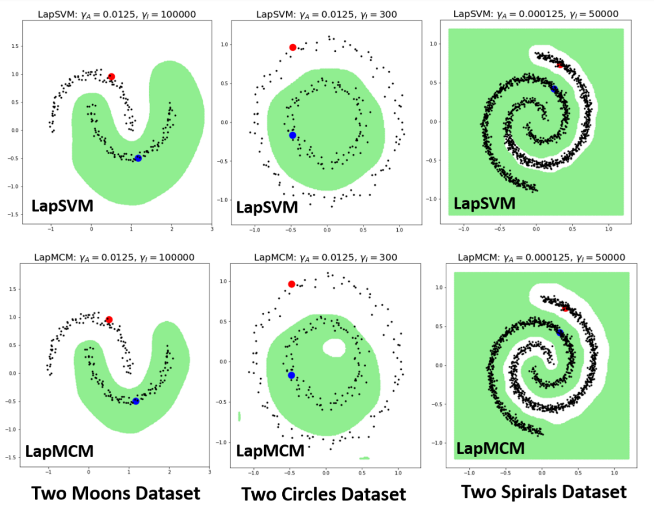


Figure: Performance of LapMCM on artificial datasets

Graph Trend Filtering

[Wan+16] proposed Graph Trend Filtering (GTF) as the following problem,

$$\hat{\beta} = \arg\min \ \frac{1}{2} \| \ y - \beta \ \|_2^2 + \lambda \| \ D^{k+1} \beta \ \|_1$$

GTF have various advantages over Graph-Laplacian based l-2 norm and the matrix, D called the Graph Difference Operator (GDO) plays an important role

$$D^{(k+1)T} D^{(k+1)} = L \tag{10}$$

For 1st, order GDO consider the l^{th} edge joining the i^{th} and j^{th} node i.e, $e_{l_{ij}}$. Then the l^{th} row of the GDO becomes,

$$D_l = (0, ..., \underset{\substack{\downarrow \\ i}}{1}, ..., \underset{\substack{\downarrow \\ j}}{-1}, ..., 0)$$

GDO also satisfies,

$$\| \ D\beta \ \|_1 = \sum_{\{i,j\} \in E} |\beta_i - \beta_j| \tag{11}$$

Weighted Graph Difference Operator (GDO)

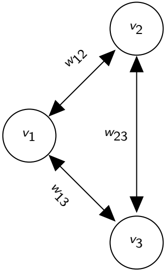
For 1st, order GDO consider the l^{th} edge joining the i^{th} and j^{th} node i.e, $e_{l_{ij}}$ with weight w_{ij}. Then we define, weighted GDO, Δ such that it's l^{th} row of the GDO becomes,

$$\Delta_l = (0, \ldots, \underset{\substack{\downarrow \\ i}}{w_{ij}^{1/2}}, \ldots, \underset{\substack{\downarrow \\ j}}{-w_{ij}^{1/2}}, \ldots, 0)$$

Δ is also simply $L^{1/2}$,

$$\Delta^T \Delta = L \qquad (12)$$

An illustration showing the calculation of Δ,



$$\Longrightarrow \Delta = \begin{bmatrix} w_{13}^{1/2} & 0 & -w_{13}^{1/2} \\ w_{12}^{1/2} & -w_{12}^{1/2} & 0 \\ 0 & w_{23}^{1/2} & -w_{23}^{1/2} \end{bmatrix}$$

Trend Filtering based semi-supervised learning framework

Conventional Learning framework:

$$f^* = \arg\min_{f \in \mathcal{H}_K} \frac{1}{l} \sum_{i=1}^{l} V(x_i, y_i, f) + \gamma_A \| f \|_A^2 \tag{13}$$

Intrinsic norm for Manifold Regularization [BNS06] :

$$f^* = \arg\min_{f \in \mathcal{H}_K} \frac{1}{l} \sum_{i=1}^{l} V(x_i, y_i, f) + \gamma_A \| f \|_A^2 + \gamma_I \| f \|_I^2 \tag{14}$$

Trend-Filtering based intrinsic norm (Ours):

$$f^* = \arg\min_{f \in \mathcal{H}_K} \frac{1}{l} \sum_{i=1}^{l} V(x_i, y_i, f) + \gamma_A \| f \|_A^2 + \frac{\gamma_I}{(u+l)} \| \Delta f \|_1 \tag{15}$$

Trend filtering based framework $\sim L^{1/2}$ regularization as $\Delta^T \Delta = L$

Trend Filtered MCM - TFMCM

$$\min_{h,q,b,\alpha} \quad h + \frac{1}{l}\sum_{i=1}^{l} q_i + \frac{\gamma_l}{u+l} \| \Delta K\lambda \|_1 \tag{16}$$

such that ,

$$h \geq y_i\Big(\sum_{j=1}^{l+u} \lambda_j \times K_{ij} + b\Big)$$

$$y_i\Big(\sum_{j=1}^{l+u} \lambda_j \times K_{ij} + b\Big) + q_i \geq 0$$

$$q_i \geq 0, \quad h \geq 1$$

$$\forall i = 1, 2, 3, , ..., l$$

Advantages: Same as that of GTF [Wan+16]

1. Computational efficiency
2. Local adaptivity
3. Complex extensions

TFMCM Vs LapMCM Vs LapSVM

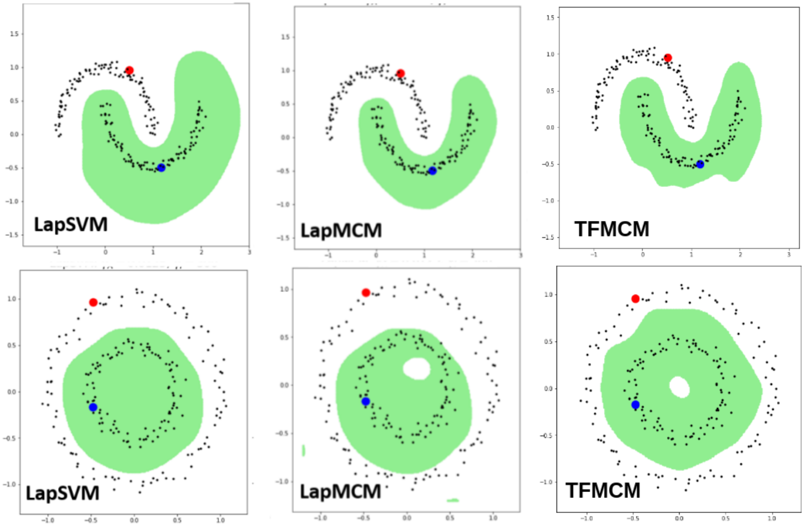


Figure: Performance of TFMCM on artificial datasets

Significance of "C" in LapMCM

$$\min_{h,\lambda,b,q} \frac{h^2}{2} + \frac{C}{2l}\sum_{i=1}^{l} q_i^2 + \frac{\gamma_A}{2}\lambda^T K\lambda + \frac{\gamma_I}{l+u}\lambda^T KLK\lambda + \frac{p}{2}b^2 \tag{17}$$

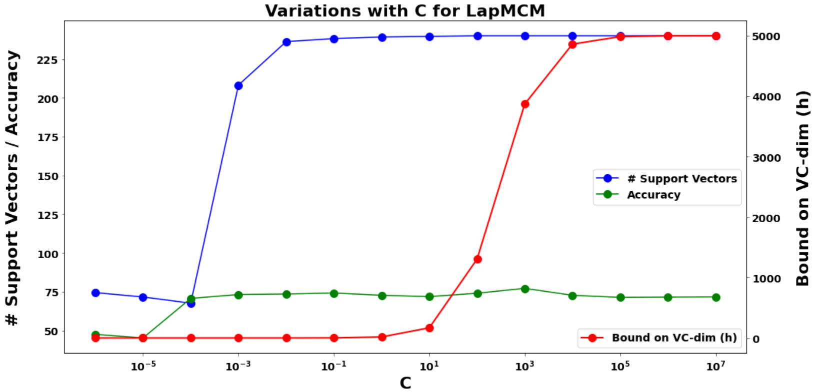


Figure: Role of the hyper-parameter "C" in LapMCM

Minimal complexity of LapMCM

Increasing data points $\implies$ 40% labelled samples $\implies$ Tuned using Grid search

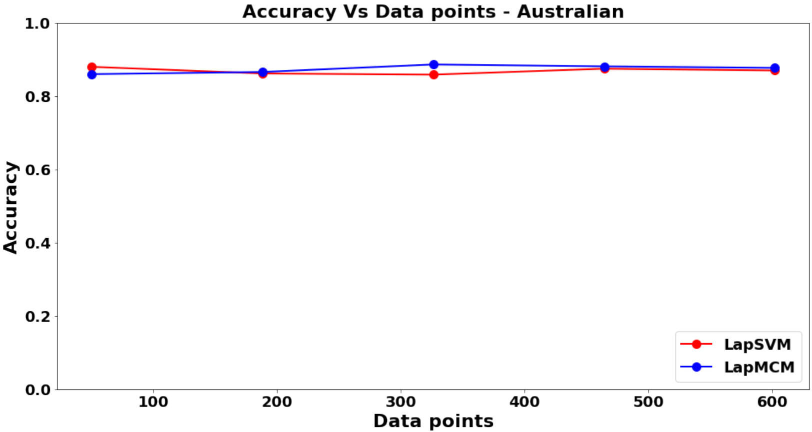


Figure: Accuracy vs datapoints for LapSVM and LapMCM on Australian dataset

$\implies$ similar accuracies, with slight edge for LapMCM

Minimal complexity of LapMCM

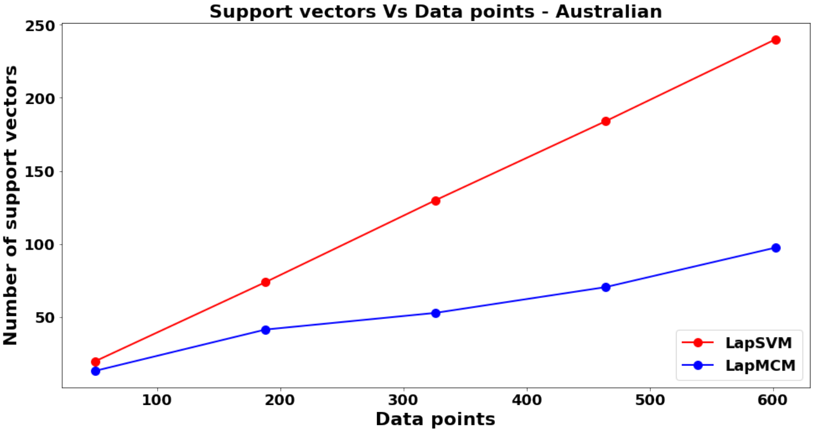


Figure: No. of support vectors vs datapoints - LapSVM & LapMCM - Australian

$\implies$ similar accuracies, but drastically lesser number of support vectors for LapMCM
$\implies$ Minimal complexity

Minimal complexity of LapMCM

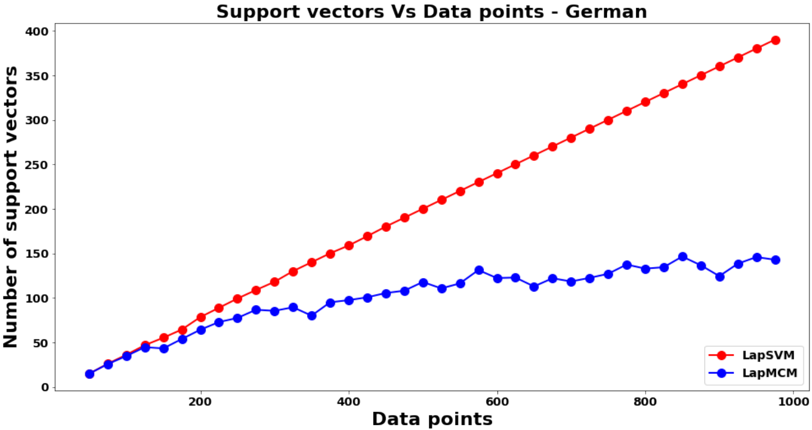


Figure: No. of support vectors vs datapoints - LapSVM & LapMCM - German

$\Longrightarrow$ As data points increase $\Longrightarrow$ No. of support vectors for LapMCM saturates

Accuracies on UCI datasets

UCI datasets with 40% labelled samples distributed equally among two binary classes, with models trained using grid search

Dataset	LapSVM	LapMCM
Australian (690×14×2)	0.869 ± 0.042	$\mathbf{0.875 \pm 0.035}$
German (1000×24×2)	$0.700 \pm 0.04\ 5$	$\mathbf{0.725 \pm 0.015}$
Ionosphere (351×34×2)	0.878 ± 0.077	$\mathbf{0.909 \pm 0.066}$
Heart (270×13×2)	0.811 ± 0.032	$\mathbf{0.833 \pm 0.031}$

Table: Performance of LapMCM on UCI datasets with 40% labelled samples

$\Longrightarrow$ Better performance of LapMCM over LapSVM

$\Longrightarrow$ How do performance vary when with the percentage of labelled samples ?

Performance Vs. LapSVM

Increasing number of labelled samples with fixed total datapoints

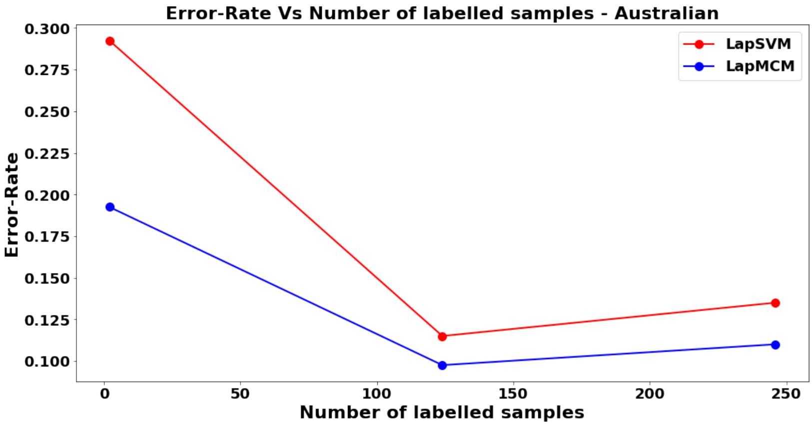


Figure: Error-rate vs No. labelled samples - LapSVM & LapMCM - Australian

$\Longrightarrow$ LapMCM performs better than LapSVM for small number of labelled samples

Performance Vs. LapSVM

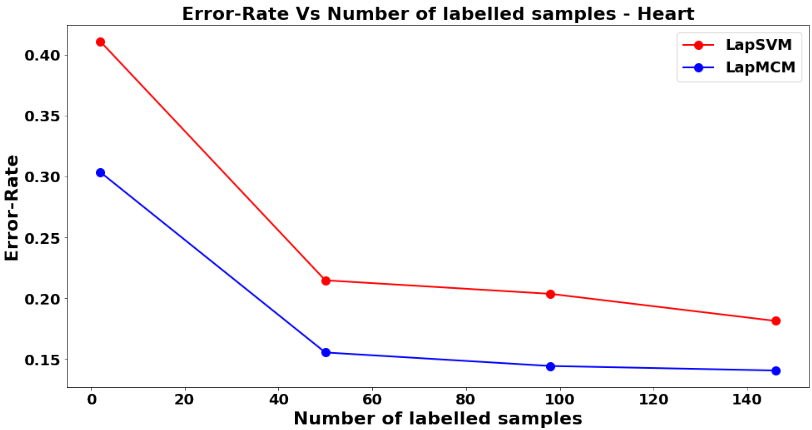


Figure: Error-rate vs No. labelled samples - LapSVM & LapMCM - Heart

$\Longrightarrow$ LapMCM performs better than LapSVM for small number of labelled samples

Performance Vs. LapSVM

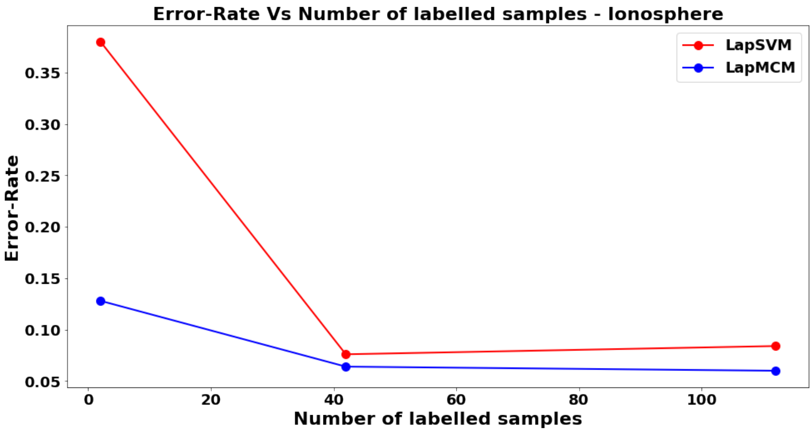


Figure: Error-rate vs No. labelled samples - LapSVM & LapMCM - Ionosphere

$\Longrightarrow$ LapMCM performs better than LapSVM for small number of labelled samples

Feature Selection through unlabelled data

LapMCM minimizes the VC dimension and the VC dimension in a case of spherized data, is determined by the number of features thus LapMCM minimizes features
$\implies$ LapMCM performs feature discrimination

Train a linear LapMCM on data with large features and small number of samples
$\implies$ train with a pair of labelled samples and all other as unlabelled samples
$\implies$ select features with non-zero weights

To check exhaustiveness of selected feature $\implies$
Train and test standard SVM on using the selected features

Datasets	**Features**				**Accuracies**			
(samples X dimensions)	**LapMCM**	**MCM**	**ReliefF**	**FCBF**	**LapMCM**	**MCM**	**ReliefF**	**FCBF**
Alon (62×2000)	25	41	896	1984	87%	83.8%	82.2%	82.1%
Shipp (77×7129)	35	51	3196	7129	97%	96.1%	93.5%	93.5%
Golub (72×7129)	67	47	2271	7129	96%	95.8%	90.3%	95.8%
Singh (102×12600)	66	81	5650	11619	91%	91.2%	89.2%	92.5%
Christensen (198×1413)	198	98	633	1413	99%	99.5%	99.5%	99.5%

Table: LapMCM based feature selection

LapMCM tends to select fewer features than ReliefF, FCBF [Jay+16] and still gives better performance measures which verifies the application of feature selection using unlabelled data

TFMCM based Regressor

Building a regressor from the TFMCM classifier using the method of [BP03]

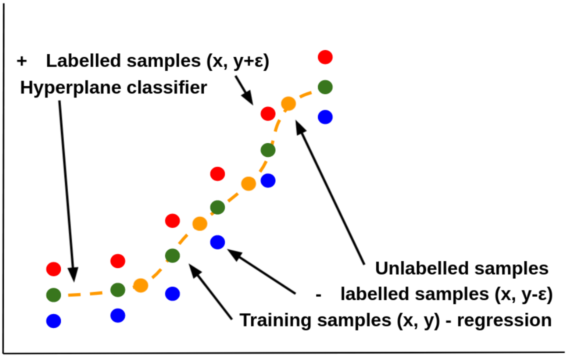


Figure: Regression as a classification problem [BP03]

Thus, the corresponding Kernel TFMCM regressor formulated follwoing the approach of [Jay15] will be,

TFMCM based Regressor

$$\min_{h,q,b,\alpha} \quad h + \frac{C}{l}\sum_{i=1}^{l}(q_i^+ + q_i^-) + \frac{\gamma_l}{u+l}||\Delta K\lambda||_1 \tag{18}$$

such that ,

$$h \geq 1 \times \left[(\sum_{j=1}^{l+u} \lambda_j \times K_{ij} + b) + \eta(y_i{+}\epsilon) \right]$$

$$1 \times \left[(\sum_{j=1}^{l+u} \lambda_j \times K_{ij} + b) + \eta(y_i{+}\epsilon) \right] + q_i^+ \geq 1$$

$$h \geq -1 \times \left[(\sum_{j=1}^{l+u} \lambda_j \times K_{ij} + b) + \eta(y_i{-}\epsilon) \right]$$

$$-1 \times \left[(\sum_{j=1}^{l+u} \lambda_j \times K_{ij} + b) + \eta(y_i{-}\epsilon) \right] + q_i^- \geq 1$$

$$q_i^+, q_i^- \geq 0, \quad h \geq 1$$

$$\forall i = 1, 2, ..., l$$

$$y = -\frac{1}{\eta}(\sum_{j=1}^{l+u} \lambda_j \times K_{ij} + b)) \tag{19}$$

TFMCM based Regressor

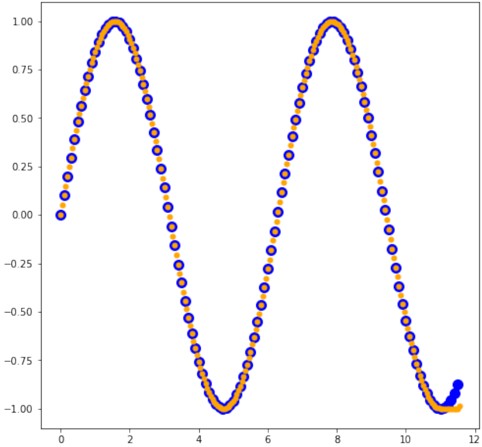


Figure: Results of TFMCM regressor on sine curve

Can learn various functions with complex manifolds, from limited data due to the inherent advantage of **unlimited unlabelled data** for regression

Future extensions of the work

1. Large scale extension of TFMCM: Develop an iterative solution for the TFMCM optimization problem, in the primal form.

2. Exhaustive exploration of the unlabelled samples based feature selection, as an individual research problem.

3. Make SGL framework for learning graphs, adaptive.

References

[1] Mikhail Belkin, Partha Niyogi, and Vikas Sindhwani. "Manifold Regularization: A Geometric Framework for Learning from Labeled and Unlabeled Examples". In: *Journal of Machine Learning Research* 7 (2006), pp. 2399–2434.

[2] Jinbo Bi and Kristin P.Bennett. "A geometric approach to support vector regression". In: *Neurocomputing* 55 (2003), pp. 79–108.

[3] Christopher J.C. Burges. "A Tutorial on Support Vector Machines for Pattern Recognition". In: *Data Mining and Knowledge Discovery* 2 (1998), pp. 121–167.

[4] Jayadeva. "Feature Selection through Minimization of the VC dimension". In: *Preprint submitted to ArXiV* (2014).

[5] Jayadeva. "Learning a hyperplane classifier by minimizing an exact bound on the VC dimension". In: *Neurocomputing* 149 (2015), pp. 683–689.

[6] Jayadeva et al. "Learning a hyperplane regressor through a tight bound on the VC dimension". In: *Neurocomputing* 171 (2016), pp. 1610–1616.

[7] Sachindra Joshi et al. "Using Sequential Unconstrained Minimization Techniques to simplify SVM solvers". In: *Neurocomputing* 77 (1 2012), pp. 253–260.

[8] V Vapnik. *A Tutorial on Support Vector Machines for Pattern Recognition*. Vol. 2. 1998.

[9] Yu-Xiang Wang et al. "Trend Filtering on Graphs". In: *Journal of Machine Learning Research* 17 (2016), pp. 1–41.