Laplacian-based semi-supervised learning

Mario.Guarracino@cnr.it
Laboratory of Computational & Data Science
High Performance Computing and Networking Institute
National Research Council of Italy

Lion 11 – June 19-21, 2017 – Nizhny Novgorod

Introduction

- Machine learning powers many aspects of modern society: from web searches to recommendation systems, from transcribing speech to text to the identification of genetic mutations leading to diseases.
- The ability to take advantage of increases in available computation and data will considerably impact in its success.
- In this tutorial we will see how to extract knowledge from information structured from available data.

Summary

- Supervised learning
- Semi-supervised
- Laplacian based regularization
- GEPSVM & SVM
- Algorithms, test cases and numerical examples.

Supervised learning

- The most common form of machine learning is ***supervised learning***.
- Imagine we want to build a system that can classify patients affected by cancer using the abundance of specific molecules present in blood samples.
- We first collect a large dataset of blood samples, each labeled with presence/absence of cancer.
- During training, the machine is shown the data and produces a model minimizing the error in assigning data to their class.

Supervised learning

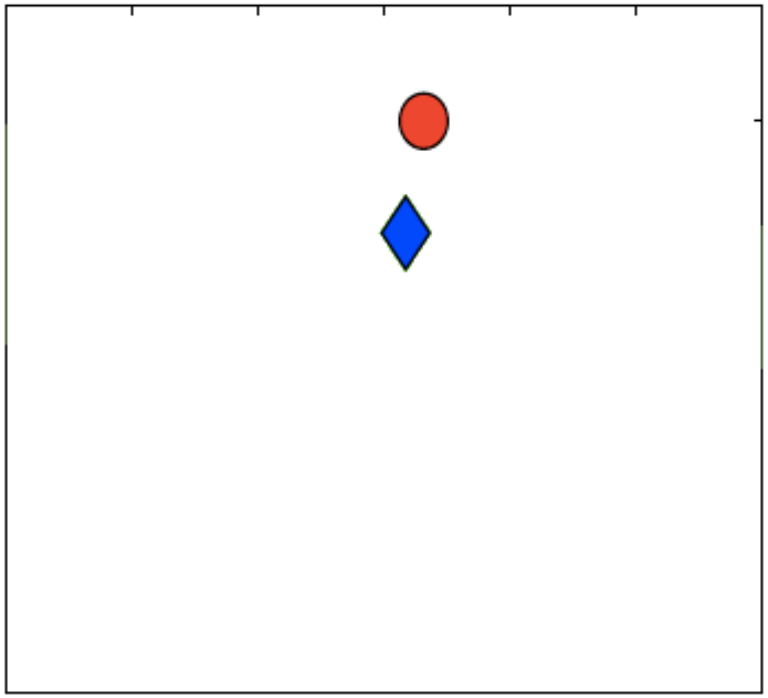
- ***Supervised learning*** refers to the capability of a system to learn from examples (*training set*)
- The trained system is able to provide an answer (*output*) for each new question (*input*)
- ***Supervised*** means that the desired output for the training set is provided by an external teacher
- ***Binary classification*** is among the most successful methods for supervised learning

Semi-supervised learning

- Digital technologies produce a large quantity of un-annotated data, and few labeled data useful for training.
- Exploiting info from the unlabeled data in the learning phase is of great practical importance.
- The significance of semi-supervised learning might find its foundations in human learning.
  - Human learning occurs in a semi supervised regime (eg. acoustic to phonetic association)

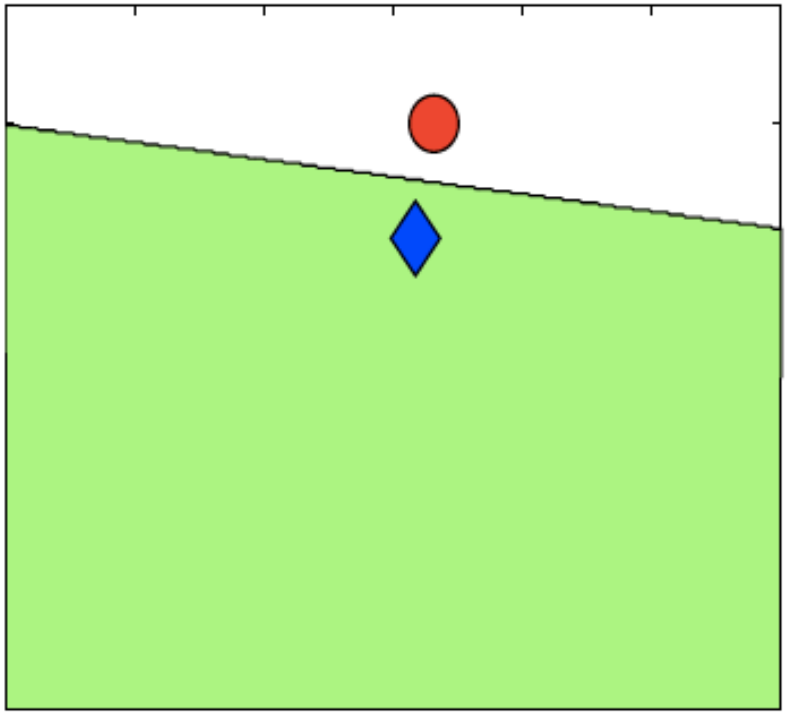
From natural choices to probability distributions

- How can we build a classifier on the basis of these two points?



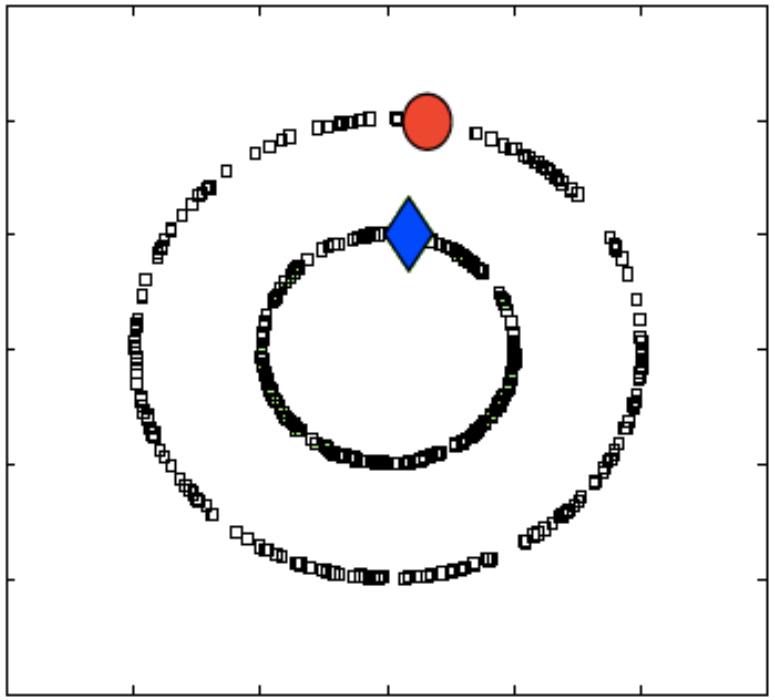
From natural choices to probability distributions

- How can we build a classifier on the basis of these two points?
- A natural choice would be a linear separator.
- Many formalisms (Bayesian, regularization, minimum description length or structural risk minimization ...) would provide such a model.



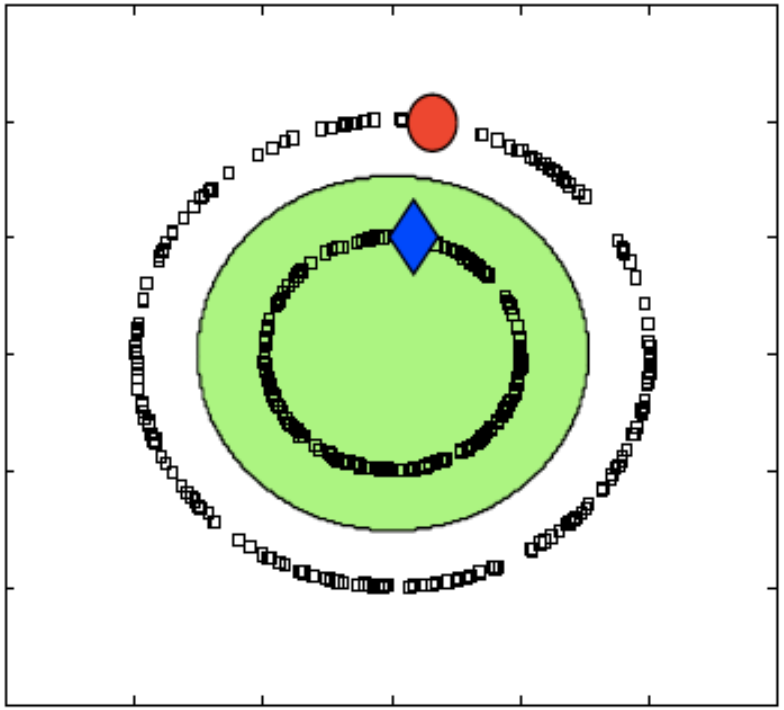
From natural choices to probability distributions

- Suppose we have some more unlabeled data available.
- It looks self evident it would be better to reevaluate the previous choice.



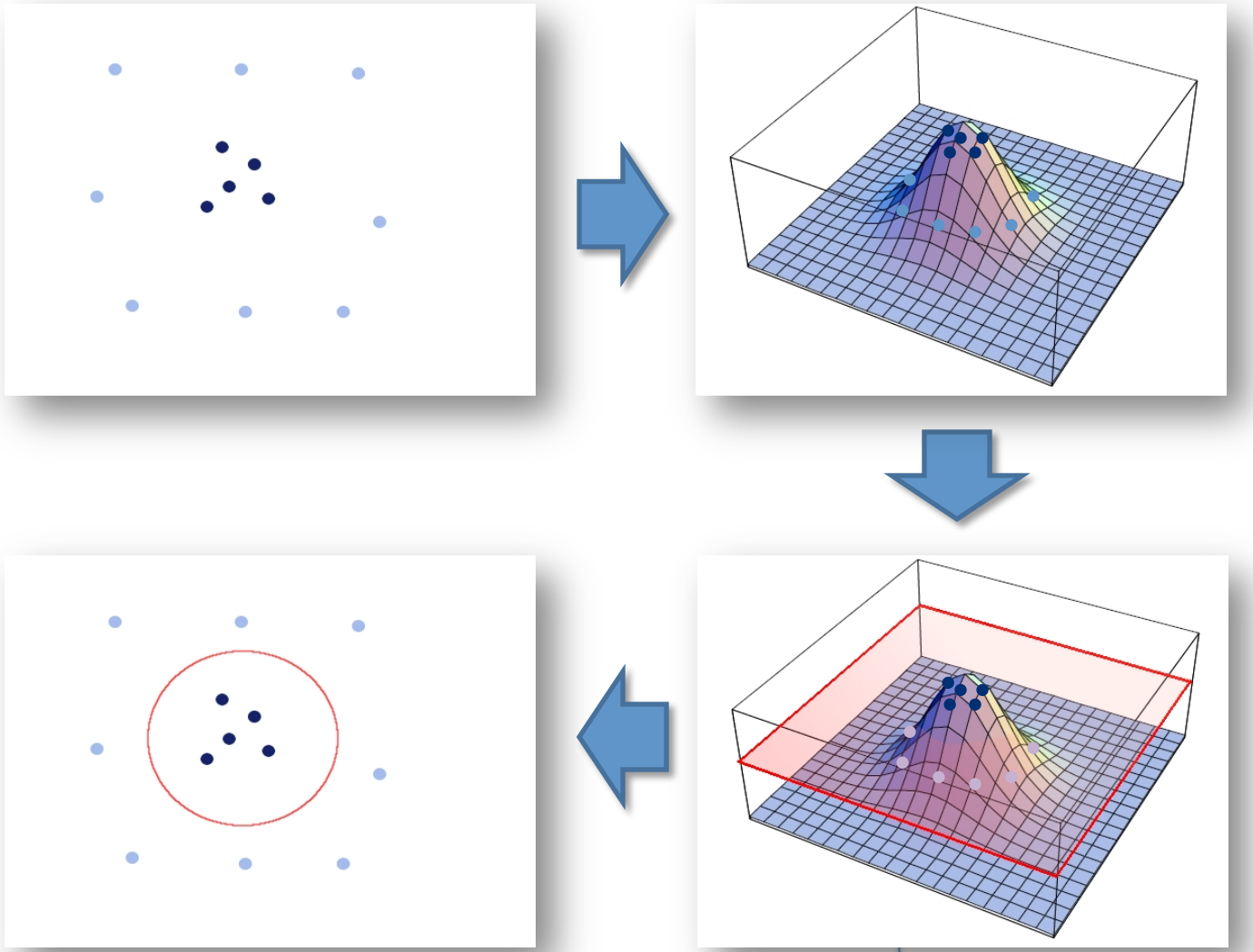
From natural choices to probability distributions

- Suppose we have some more unlabeled data available.
- It looks self evident it would be better to reevaluate the previous choice.
- The particular geometric structure of the marginal distribution suggests a circular classifier.



The kernel trick

- To obtain greater separation between classes, nonlinearly embed points into a higher dimensional space



Manifold regularization

Journal of Machine Learning Research 7 (2006) 2399-2434 Submitted 4/05; Revised 5/06; Published 11/06

Manifold Regularization: A Geometric Framework for Learning from Labeled and Unlabeled Examples

Mikhail Belkin MBELKIN@CSE.OHIO-STATE.EDU
Department of Computer Science and Engineering
The Ohio State University
2015 Neil Avenue, Dreese Labs 597
Columbus, OH 43210, USA

Partha Niyogi NIYOGI@CS.UCHICAGO.EDU
Departments of Computer Science and Statistics
University of Chicago
1100 E. 58th Street
Chicago, IL 60637, USA

Vikas Sindhwani VIKASS@CS.UCHICAGO.EDU
Department of Computer Science
University of Chicago
1100 E. 58th Street
Chicago, IL 60637, USA

Manifold regularization

- A technique for using the shape of a dataset to constrain the functions that should be learned on that dataset.
- The data to be learned do not cover the entire input space.
  - For example, a genetic mutation detection system may not need to classify all possible genetic variations, but only a subset that contain mutations.
- The technique assumes the relevant subset of data comes from a manifold and the function to be learned is smooth:
  - data with different labels are not likely to be close together, and so the labeling function should not change quickly in areas where there are likely to be many data points.
- These algorithms can use unlabeled data to inform where the learned function is allowed to change quickly and where it is not (extension of the technique of Tikhonov regularization).

Learning from examples

- Suppose labeled examples (x,y) are generated according to a probability distribution $\mathcal{P}$ on $X \times \mathbb{R}$. Unlabeled examples are $x \in X$ drawn according to the marginal distribution $\mathcal{P}_X$ of $\mathcal{P}$.
- **Can the knowledge of $\mathcal{P}_X$ be exploited for better function learning** (e.g., in classification or regression tasks)**?**
- If there is no relation between $\mathcal{P}_X$ and the conditional $\mathcal{P}(y|x)$, the knowledge of $\mathcal{P}_X$ is likely useless.
- Therefore, we will assume: if two points $x_1, x_2 \in X$ are close in the intrinsic geometry of $\mathcal{P}_X$, then the conditional distributions $\mathcal{P}(y|x_1)$ and $\mathcal{P}(y|x_2)$ are similar.
  - The conditional probability distribution $\mathcal{P}(y|x)$ varies smoothly along the geodesics in the intrinsic geometry of $\mathcal{P}_X$.

Learning from examples

- We will make use of these geometric intuitions to extend an established framework for function learning.
- A number of popular algorithms such as SVM, Ridge regression, splines, Radial Basis Functions may be broadly interpreted as regularization algorithms with different empirical cost functions and complexity measures.
- These features are chosen in an appropriate Reproducing Kernel Hilbert Space (RKHS).

Learning from examples

- For a Mercer kernel $K: X \times X \rightarrow \mathbb{R}$, there is an associated RKHS $\mathcal{H}_K$ of functions $X \rightarrow \mathbb{R}$ with norm $\|.\|_K$.
- Given a set of labeled examples $\{(x_i, y_i),\ i = 1,\ldots,l\}$, the standard framework estimates an unknown function by minimizing

$$f^* = \operatorname*{argmin}_{f \in \mathcal{H}_K} \frac{1}{l} \sum_{i=1}^{l} V(x_i, y_i, f) + \gamma \|f\|_K^2,$$

- where V is some loss function
  - i.e. squared loss $(y_i - f(x_i))^2$ for RLS, hinge loss function $\max[0; 1 - y_i f(x_i)]$ for SVM.
- Penalizing the RKHS norm imposes smoothness conditions on possible solutions.

Learning from examples

- Every function in $\mathcal{H}_K$ that minimises an empirical risk function can be written as a linear combination of the kernel function evaluated at the training points. (Representer Theorem)

$$f^*(x) = \sum_{i=1}^{l} \alpha_i K(x_i, x).$$

- Therefore, the problem is reduced to optimizing over the finite dimensional space of coefficients α_i, which is the algorithmic basis for SVM, regularized least squares and other regression and classification schemes.

The Representer theorem

- Assume that the penalty term $\|f\|_I$ is sufficiently smooth with respect to the RKHS norm $\|f\|_K$.
- Then the solution f^* to the optimization problem

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

- exists and admits the following representation

$$f^*(x) = \sum_{i=1}^{l} \alpha_i K(x_i, x) + \int_{\mathcal{M}} \alpha(z) K(x, z) \, d\mathcal{P}_X(z)$$

- where $\mathcal{M} = \operatorname{supp}\{\mathcal{P}_X\}$ is the support of the marginal $\mathcal{P}_X$.
- We can express the solution f directly in terms of the labeled data, the (ambient) kernel K, and the marginal $\mathcal{P}_X$.
- If $\mathcal{P}_X$ is unknown, the solution may be expressed in terms of an empirical estimate of $\mathcal{P}_X$.

Unknown marginal distribution

- In most applications $\mathcal{P}_X$ is unknown and we must attempt to get empirical estimates of $\mathcal{P}_X$ and $\|.\|_I$.
- To get such empirical estimates it is sufficient to have unlabeled examples.
- In case $\mathcal{P}_X$ is a compact submanifold $\mathcal{M}$ in $\mathbb{R}^n$, one natural choice for $\|f\|_I$ is $\int_{x\in\mathcal{M}} \|\nabla_{\mathcal{M}} f\|^2 d\mathcal{P}_X(x)$, where $\nabla_{\mathcal{M}}$ is the gradient of f along the manifold $\mathcal{M}$ and the integral is taken over the marginal distribution.
- The optimization problem becomes

$$f^* = \underset{f\in\mathcal{H}_K}{\operatorname{argmin}} \frac{1}{l}\sum_{i=1}^{l} V(x_i, y_i, f) + \gamma_A \|f\|_K^2 + \gamma_I \int_{x\in\mathcal{M}} \|\nabla_{\mathcal{M}} f\|^2 d\mathcal{P}_X(x).$$

- The regularization term can be approximated on the basis of labeled and unlabeled data using the **graph Laplacian**.

Unknown marginal distribution

- Given a set of l labeled examples $\{(x_i, y_i)\ i=1,\ldots,l\}$ and a set of u unlabeled examples $\{x_j, j=l+1,\ldots,l+u\}$, we consider the following optimization problem:

$$f^* = \operatorname*{argmin}_{f\in\mathcal{H}_K} \frac{1}{l}\sum_{i=1}^{l} V(x_i, y_i, f) + \gamma_A \|f\|_K^2 + \frac{\gamma_I}{(u+l)^2}\sum_{i,j=1}^{l+u}(f(x_i)-f(x_j))^2 W_{ij},$$

$$= \operatorname*{argmin}_{f\in\mathcal{H}_K} \frac{1}{l}\sum_{i=1}^{l} V(x_i, y_i, f) + \gamma_A \|f\|_K^2 + \frac{\gamma_I}{(u+l)^2}\mathbf{f}^T L\mathbf{f}.$$

- where W_{ij} are edge weights in the data adjacency graph, $f = [f(x_1),\ldots, f(x_{l+u})]^T$, and L= D-W is the graph Laplacian, with diagonal matrix D elements $D_{ii} = \sum_{j=1}^{l+u} W_{ij}$.

A remark

- The solution can be found in a finite dimensional space.
- It is possible to show that the minimizer has an expansion in terms of both labeled and unlabeled examples:

$$f^*(x) = \sum_{i=1}^{l+u} \alpha_i K(x_i, x)$$

Support Vector Machine

- For SVMs, the following problem is solved:

$$\min_{f \in \mathcal{H}_K} \frac{1}{l} \sum_{i=1}^{l} (1 - y_i f(x_i))_+ + \gamma \|f\|_K^2,$$

- where the hinge loss is defined as: $(1 - y f(x))_+ = \max(0, 1 - y f(x))$ and the labels $y_i \in \{-1, +1\}$.
- Again, the solution is given by:

$$f^\star(x) = \sum_{i=1}^{l} \alpha_i^\star K(x, x_i).$$

Support Vector Machine

- SVM problem can be equivalently written as:

$$\min_{f \in \mathcal{H}_K, \xi_i \in \mathbb{R}} \frac{1}{l} \sum_{i=1}^{l} \xi_i + \gamma \|f\|_K^2$$

$$\text{subject to: } y_i f(x_i) \geq 1 - \xi_i \quad i = 1, \ldots, l$$

$$\xi_i \geq 0 \quad i = 1, \ldots, l.$$

- We can take advantage of Lagrangian multiplier technique.

Support Vector Machine

- The SVM problem has a simpler quadratic dual program in the Lagrange multipliers $\beta = [\beta_1,...,\beta_l]^T \in \mathbb{R}^l$:

$$\beta^\star = \max_{\beta \in \mathbb{R}^l} \sum_{i=1}^{l} \beta_i - \frac{1}{2}\beta^T Q \beta$$

$$\text{subject to:} \quad \sum_{i=1}^{l} y_i \beta_i = 0$$

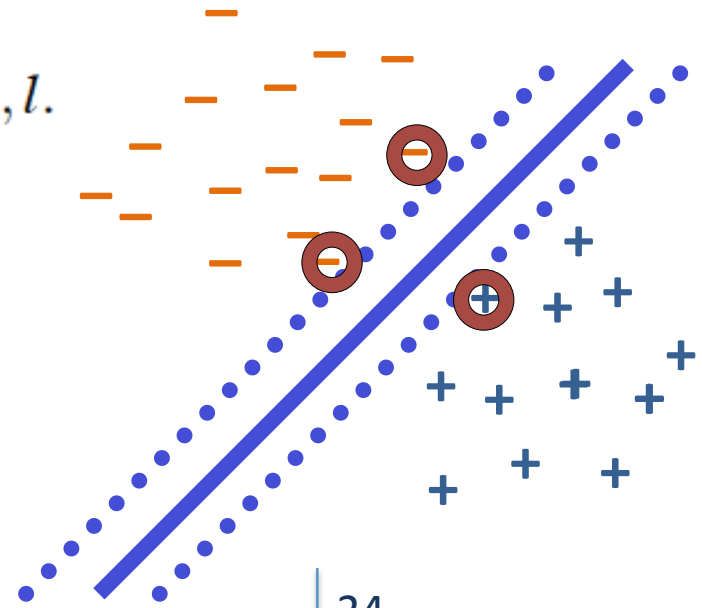
$$0 \le \beta_i \le \frac{1}{l} \quad i = 1, \ldots, l.$$

- Where:

$$Y = \text{diag}(y_1, y_2, ..., y_l),$$

$$Q = Y\left(\frac{K}{2\gamma}\right)Y,$$

$$\alpha^\star = \frac{Y\beta^\star}{2\gamma}.$$



Support Vector Machine

- The robustness of SVM relies in the strong fundamentals of statistical learning theory
- The training relies on optimization of a quadratic convex cost function, for which many methods are available
  - Packages for R, Matlab, Weka include SVM-Lite and LIBSVM

O. Vapnik The nature of statistical learning theory. Springer, 1995.

Laplacian SVM

- By including the intrinsic smoothness penalty term, we can extend SVMs by solving the

$$\min_{f \in \mathcal{H}_K} \frac{1}{l} \sum_{i=1}^{l} (1 - y_i f(x_i))_+ + \gamma_A \|f\|_K^2 + \frac{\gamma_I}{(u+l)^2} \mathbf{f}^T L \mathbf{f}.$$

- By the Representer theorem, as before, the solution to the problem above is given by:

$$f^\star(x) = \sum_{i=1}^{l+u} \alpha_i^\star K(x, x_i).$$

Laplacian SVM

- Laplacian SVMs can be implemented by solving a standard standard SVM, and using the solution to obtain the coefficients of the linear system

$$\alpha = (2\gamma_A I + 2\frac{\gamma_I}{(u+l)^2}LK)^{-1}J^T Y\beta^\star.$$

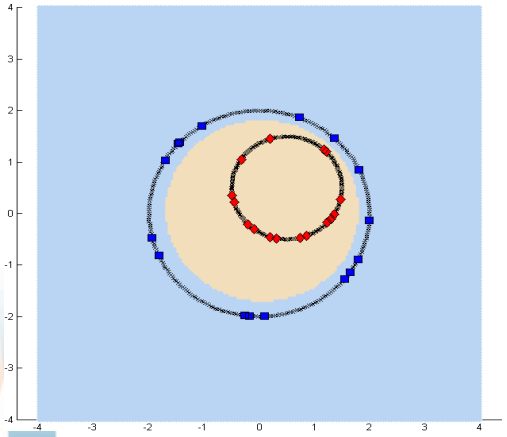
- When $\gamma_I = 0$, the SVM QP gives zero expansion coefficients over the unlabeled data, whereas the coefficients over the labeled data and the Q matrix are as in standard SVM.

A case study

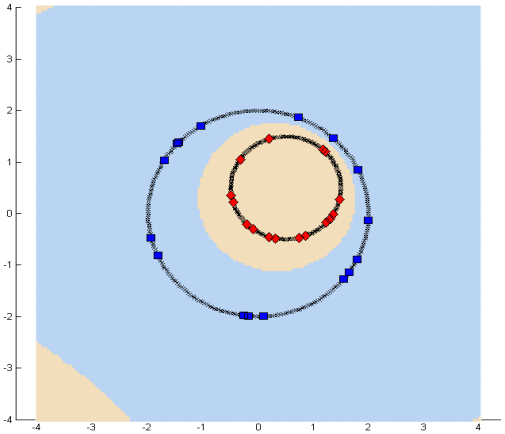
- Two circumferences
  1. Well separated
  2. Poorly separated
  3. Crossing

- Algorithms: SVM vs LapSVM

- No labeled points nearby the intersection!

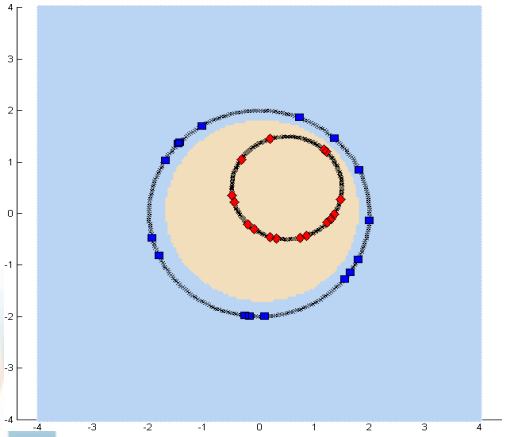


SVM
100%

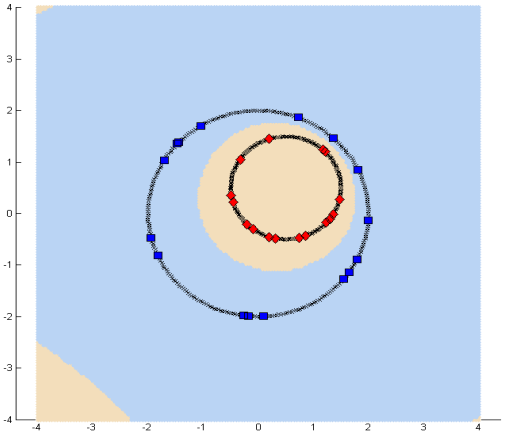


LapSVM
100%

Well separated

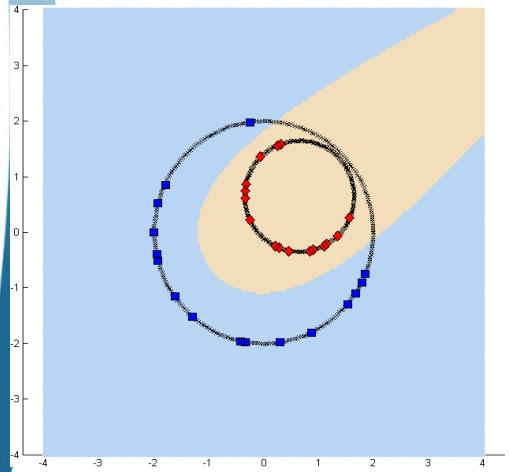


SVM
100%

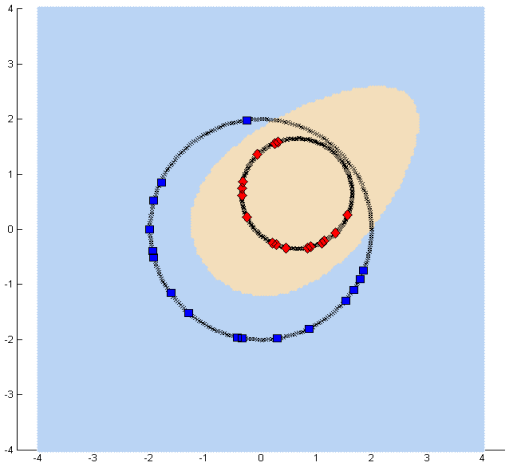


LapSVM
100%

Well separated

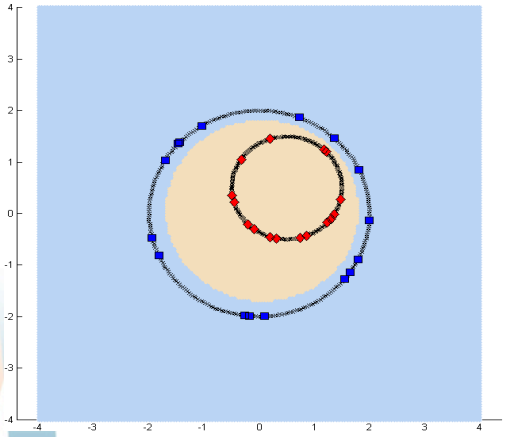


SVM
87.5%

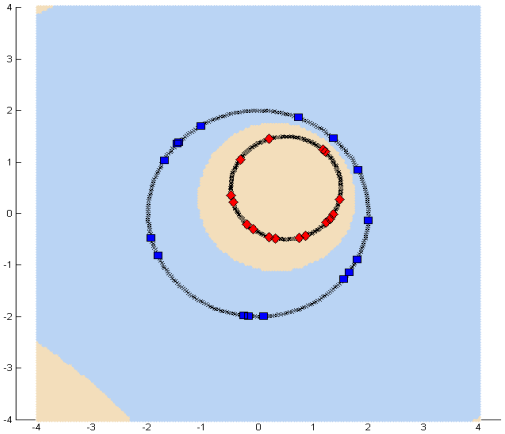


LapSVM
88.1%

Poorly separated

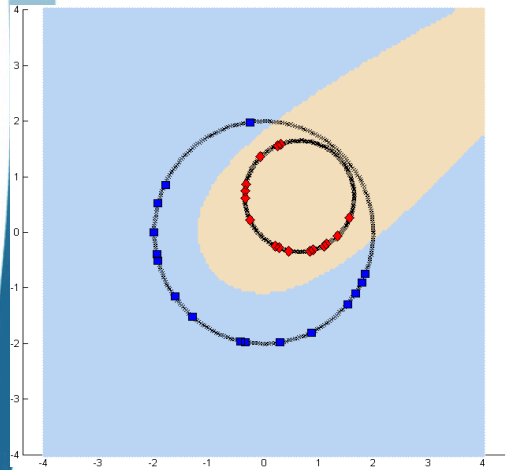


SVM
100%

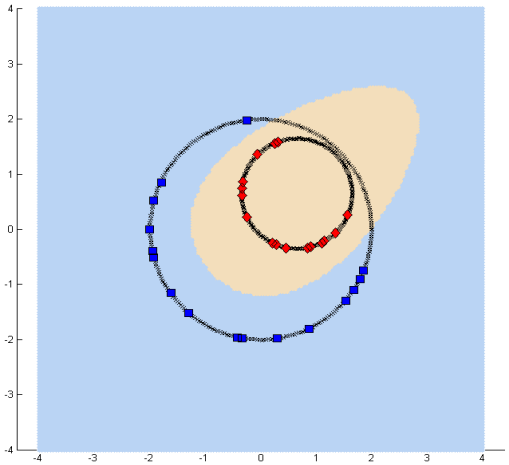


LapSVM
100%

Well separated

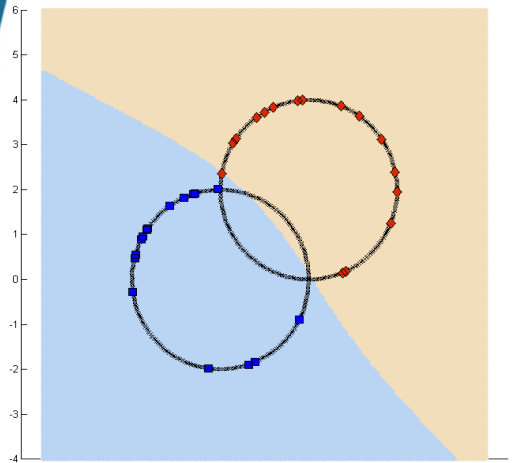


SVM
87.5%

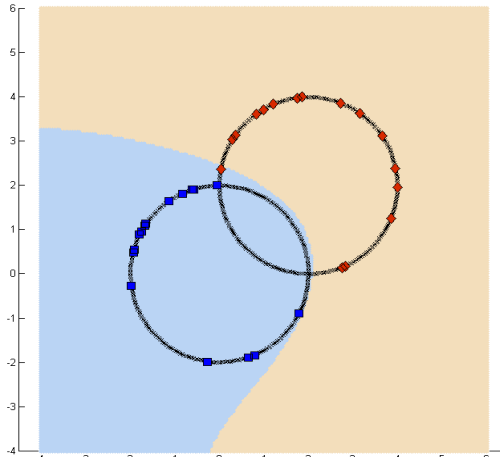


LapSVM
88.1%

Poorly separated



SVM
75.7%

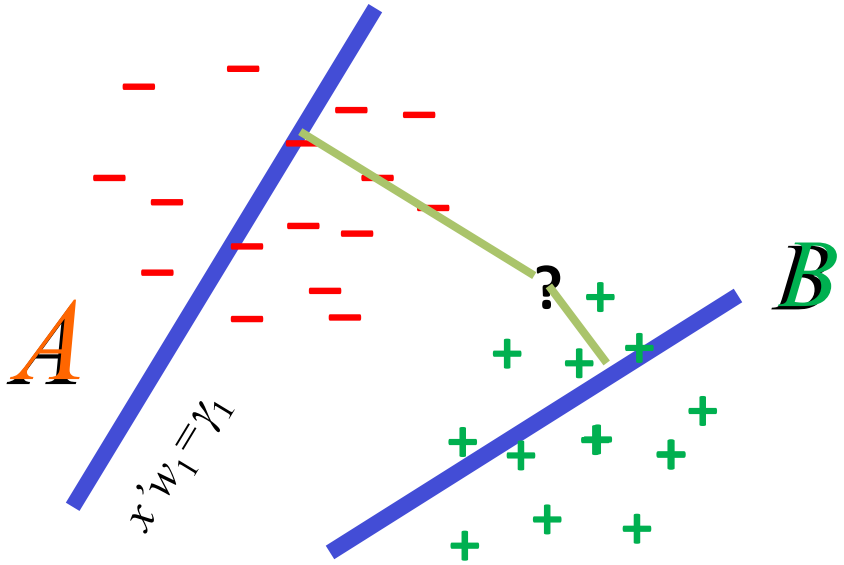


LapSVM
85.4%

Crossing

GEPSVM

- A binary classification problem can be formulated as a generalized eigenvalue problem (ReGEC)
  - Find $x'w_1=\gamma_1$ the closest to A and the farthest from B:

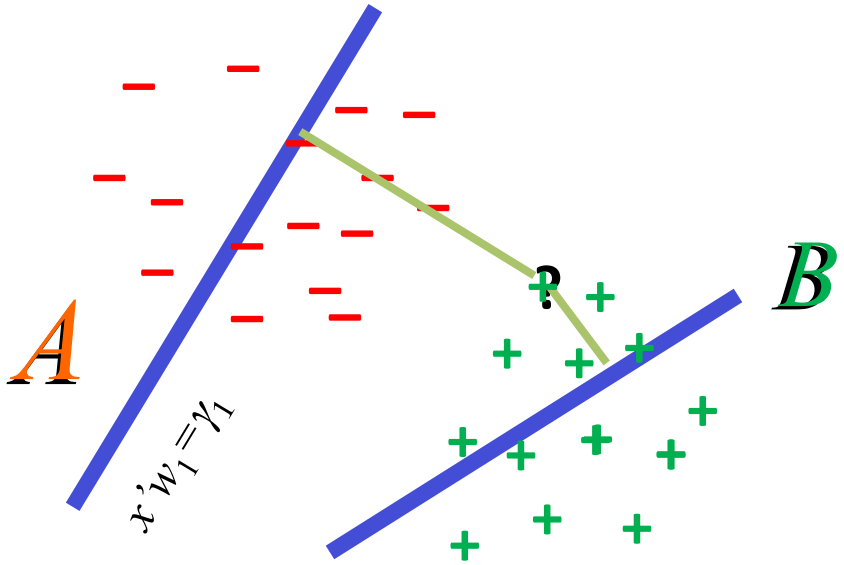


$$\min_{\omega,\gamma\neq 0} \frac{\| A\omega - e\gamma \|^2}{\| B\omega - e\gamma \|^2}$$

O. Mangasarian, E. Wild Multisurface Proximal Support Vector Classification via Generalized Eigenvalues. IEEE PAMI 2006.

GEPSVM

- A binary classification problem can be formulated as a generalized eigenvalue problem (ReGEC)
  - Find $x'w_1=\gamma_1$ the closest to A and the farthest from B:



$$\min_{\omega,\gamma\neq 0} \frac{\| A\omega - e\gamma \|^2}{\| B\omega - e\gamma \|^2}$$

O. Mangasarian, E. Wild Multisurface Proximal Support Vector Classification via Generalized Eigenvalues. IEEE PAMI 2006.

The kernel trick for GEPSVM

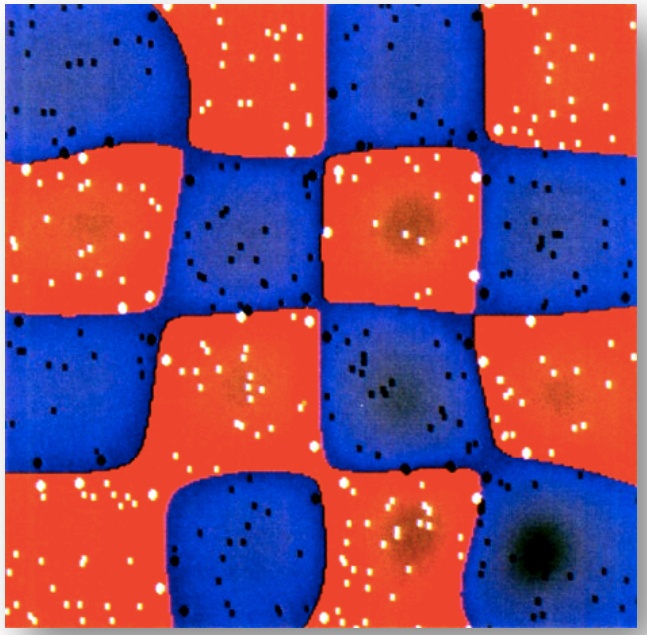
- A nonlinear embedding can be obtained with a *RBF kernel function*:

$$K(x_i, x_j) = e^{-\frac{\|x_i - x_j\|^2}{\sigma}}$$

- Each element of kernel matrix is:

$$K(A, \Gamma)_{ij} = e^{-\frac{\|A_i - \Gamma_j\|^2}{\sigma}}$$

$$\Gamma = [A^T \; B^T]^T$$



- And the model becomes:

$$\min_{\omega,\gamma \neq 0} \frac{\| K(A,\Gamma)\omega - e\gamma \|^2}{\| K(B,\Gamma)\omega - e\gamma \|^2}$$

GEPSVM regularization (ReGEC)

$$\min_{u,\gamma\neq 0}\frac{\|K(A,\Gamma)u-e\gamma\|^2}{\|K(B,\Gamma)u-e\gamma\|^2}=\min_{u,\gamma\neq 0}\frac{\|[K(A,\Gamma)\quad -e]^T[u'\quad \gamma]'\|^2}{\|[K(B,\Gamma)\quad -e]^T[u'\quad \gamma]'\|^2}$$

- Let

$$M=[K(A,\Gamma)\quad -e]^T[K(A,\Gamma)\quad -e],$$
$$N=[K(B,\Gamma)\quad -e]^T[K(B,\Gamma)\quad -e],$$
$$z=[u'\quad \gamma]'.$$

- the equation becomes:

$$\min_{z\in R^{n+1}}\frac{z'Mz}{z'Nz}$$

- Rayleigh quotients of $Mz=\lambda Nz$.

M.R.G., C. Cifarelli, O. Seref, P. Pardalos. A Classification Method Based on Generalized Eigenvalue Problems, OMS, 2007.

GEPSVM regularization (ReGEC)

$$\min_{\mathbf{z}\neq\mathbf{0}} \frac{\mathbf{z}^T (M + \delta\tilde{N})\,\mathbf{z}}{\mathbf{z}^T (N + \delta\tilde{M})\,\mathbf{z}},$$

- $\tilde{M}$ and $\tilde{N}$ main diagonals of $K(A, \Gamma)$ and $K(B, \Gamma)$
- This approach halves the execution time, still providing similar accuracy results
- Regularization is a form of *robustification*

P. Xanthopoulos, M.R.G., PM Pardalos *Robust generalized eigenvalue classifier with ellipsoidal uncertainty* AOR, 2014.

Why another method?

- ReGEC scales up with training set dimension
- ReGEC is based on a generalized eigenvalue problem => straight forward implementation in many programming languages
  - Python, Java, C++, Fortran…
- ReGEC reduces to 1 line of code in many problem solving environments (Matlab, R, Weka,…)
- It can be easily parallelized/distributed with existing software (ScaLAPACK, GridSolve,…)

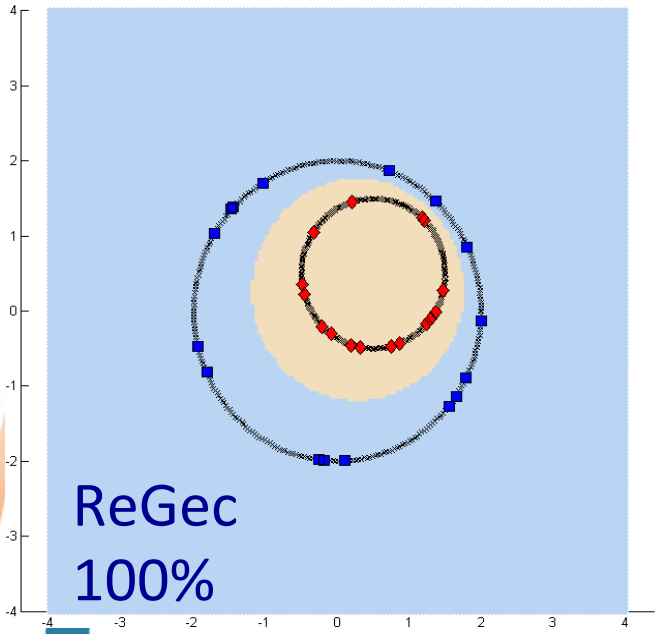
Semi-supervised ReGEC

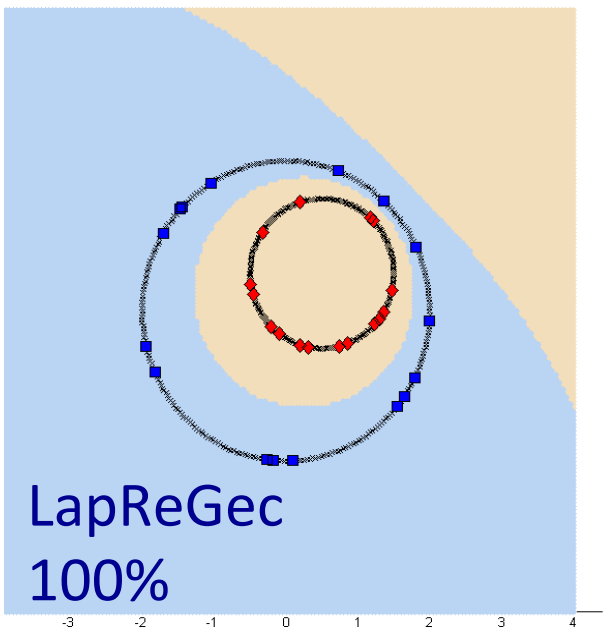
- Let L be the k-NN *graph Laplacian* built for the whole training set (labeled + unlabeled points) Γ.
- L=D-W, where W is the adjacency matrix of the graph and the diagonal matrix D is s.t. $d_{ii} = \sum_{j=1}^{n} w_{ij}$.
- The matrix J is defined as J=[X, **e**].

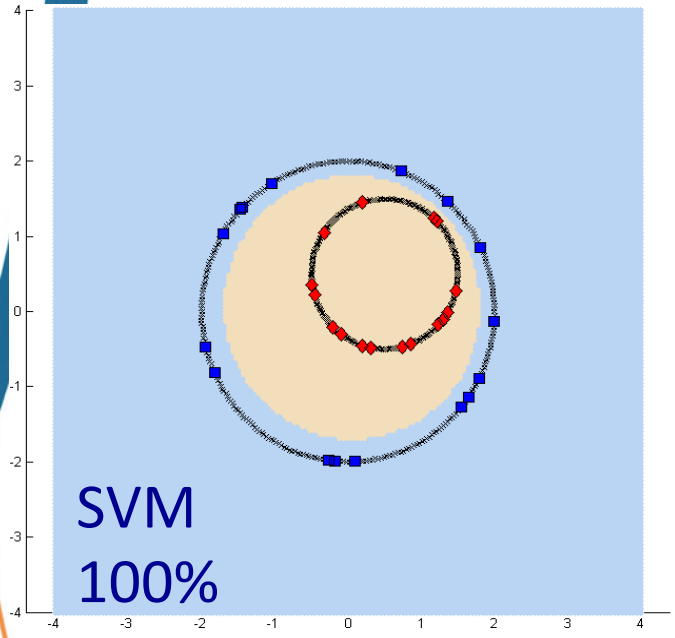
$$M = [K(A,\Gamma),\ -\mathbf{e}]^T\,[K(A,\Gamma),\ -\mathbf{e}]\,,$$
$$N = [K(B,\Gamma),\ -\mathbf{e}]^T\,[K(B,\Gamma),\ -\mathbf{e}]\,, \quad \mathbf{z} = \left[\mathbf{u}^T, b\right]^T$$

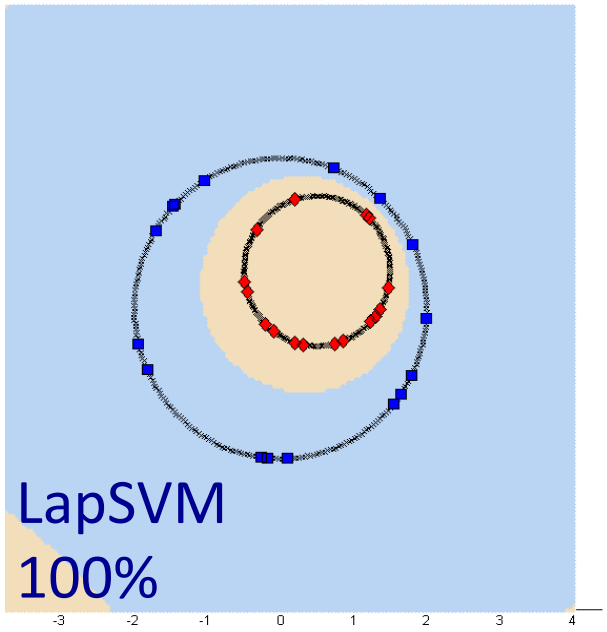
$$\min_{\mathbf{z}\neq\mathbf{0}} \frac{\mathbf{z}^T\,(M+\delta\tilde{N})\,\mathbf{z}}{\mathbf{z}^T\,(N+\delta\tilde{M})\,\mathbf{z}}, \quad \Rightarrow \quad \min_{\mathbf{z}\neq\mathbf{0}} \frac{\mathbf{z}^T\,(M+\delta\tilde{N}+c_{lap}J^T L J)\,\mathbf{z}}{\mathbf{z}^T\,(N+\delta\tilde{M})\,\mathbf{z}},$$

Well separated

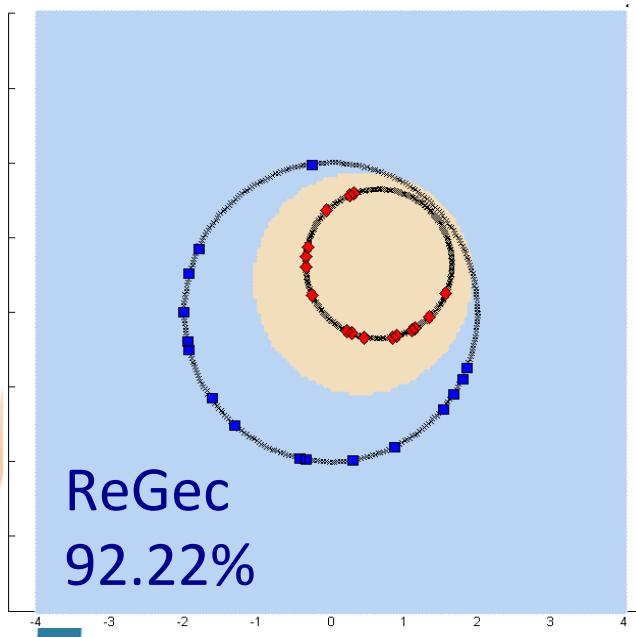


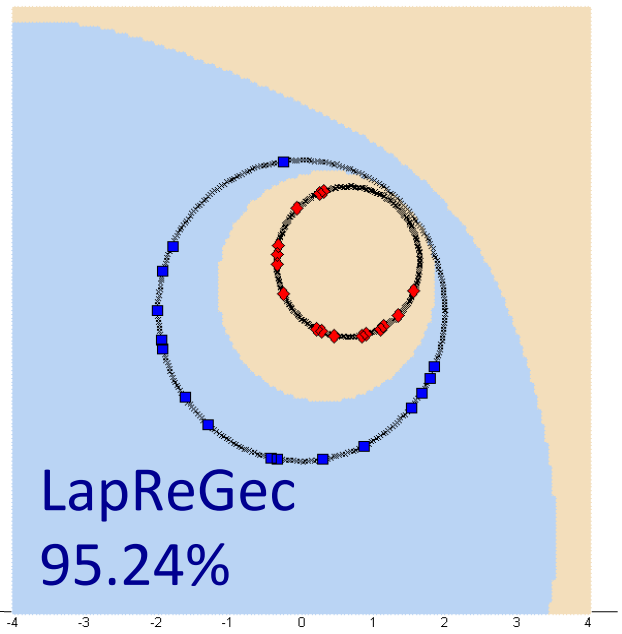


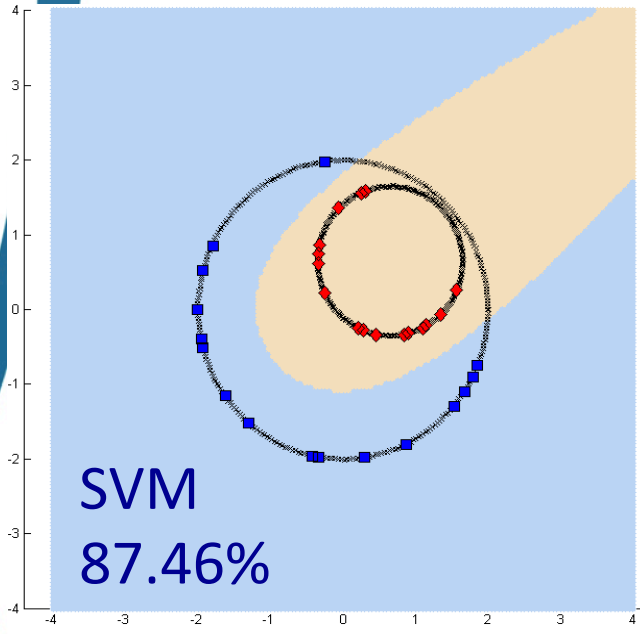


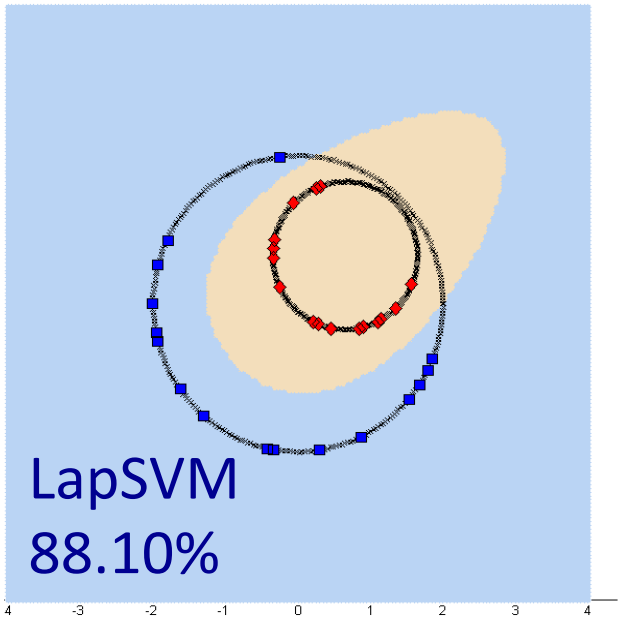


Poorly separated

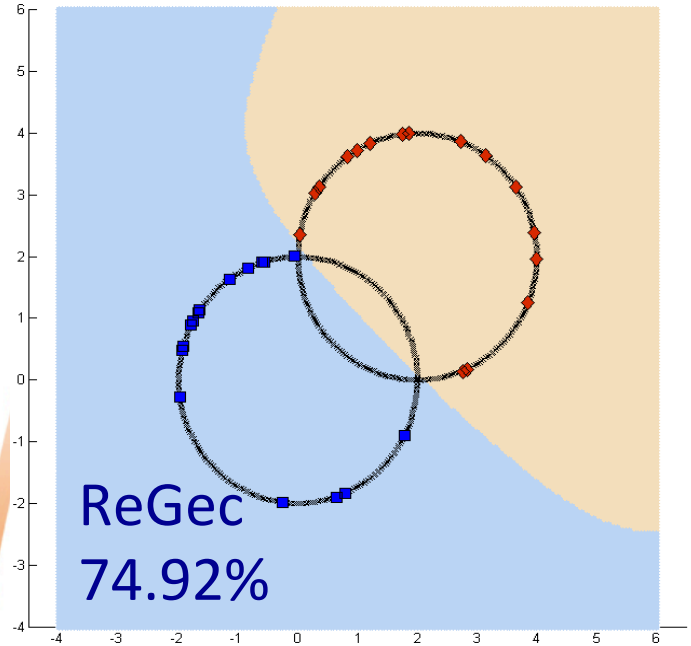


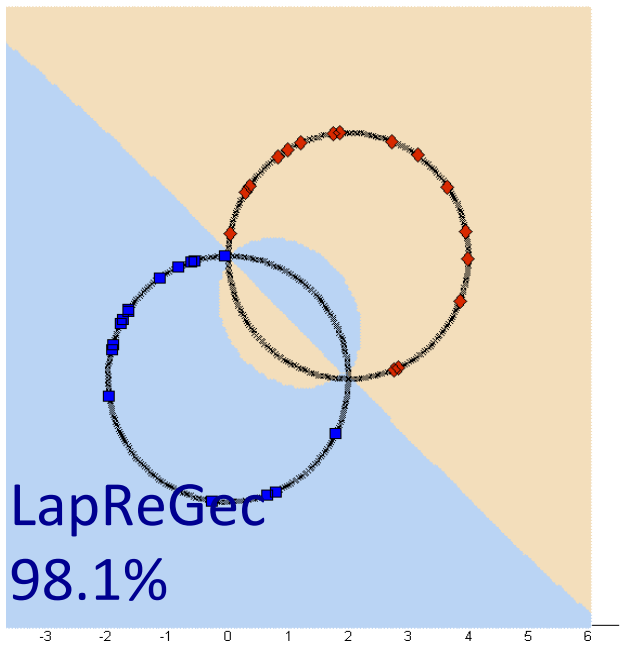


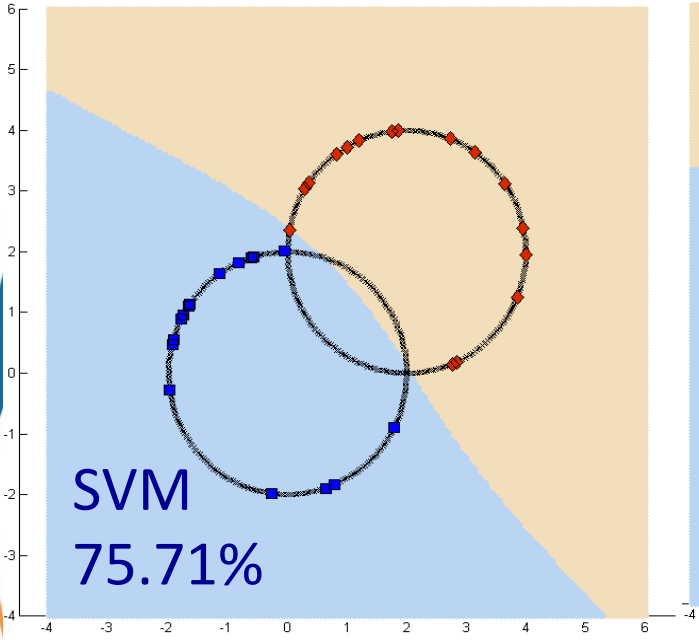


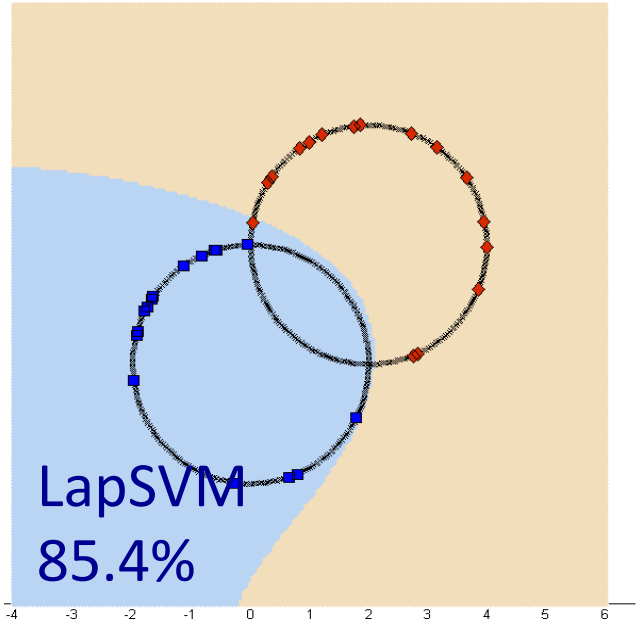


Crossing









The algorithm

Input:	l labeled examples $\{(x_i, y_i)\}_{i=1}^{l}$, u unlabeled examples $\{x_j\}_{j=l+1}^{l+u}$
Output:	Estimated function $f : \mathbb{R}^n \rightarrow \mathbb{R}$
Step 1	► Construct data adjacency graph with $(l+u)$ nodes using, for example, k nearest neighbors or a graph kernel. Choose edge weights W_{ij}, for example, binary weights or heat kernel weights $W_{ij} = e^{-\|x_i - x_j\|^2/4t}$.
Step 2	► Choose a kernel function $K(x, y)$. Compute the Gram matrix $K_{ij} = K(x_i, x_j)$.
Step 3	► Compute graph Laplacian matrix: $L = D - W$ where D is a diagonal matrix given by $D_{ii} = \sum_{j=1}^{l+u} W_{ij}$.
Step 4	► Choose γ_A and γ_I.
Step 5	► Compute α^*
Step 6	► Output function $f^*(x) = \sum_{i=1}^{l+u} \alpha_i^* K(x_i, x)$.

http://www.jmlr.org/papers/volume7/belkin06a/belkin06a.pdf

Efficiency Issues

- Algorithms compute the inverse of a dense Gram matrix which leads to $O((l+u)^3)$ complexity. This may be impractical for large data sets.
- In the case of linear kernels, we can directly write $f^*(x) = w^T x$ and solve for the weight vector w using a primal optimization method. This is much more efficient when the data is low-dimensional.
- For highly sparse data sets (e.g. in text categorization problems), effective conjugate gradient schemes can be used in a large scale implementation.
- For the non-linear case, one may obtain approximate solutions (e.g., using greedy, matching pursuit techniques) where the optimization problem is solved over the span of a small set of basis functions instead of using the full representation.

Performance evaluation

- Algorithm tested on IDA Repository datasets.
- For each dataset, we consider 100 hold outs of the data in train and test, and compute the mean and the standard deviation of the classification accuracy.
- For each of the 100 splits we averaged on 4 different choices of a % of the training set used as unlabeled.
- Mean classification accuracy for both the standard ReGEC and for the ReGEC with Laplacian regularization.
- Parameters tuned with a grid search during validation.

Performance evaluation

dataset	unlab (%)	ReGEC	LapReGEC	SVM	LapSVM
breast cancer	0	72,79±4,5	—	71,91±4,8	—
	70	71,09±3,8	71,73±3,8	71,28±4,6	71,59±4,7
	80	70,25±3,6	71,24±3,9	71,51±4,6	**72,21±4,7**
	90	68,18±4,0	70,91±3,8	71,91±4,4	**72,06±4,0**
	95	67,04±4,5	71,14±3,9	70,75±4,1	**71,63±4,2**
diabetis	0	70,37±3,8	—	76,78±1,7	—
	70	70,29±2,7	**73,94±1,8**	65,68±2,5	68,77±2,1
	80	69,12±2,7	**73,47±1,7**	66,17±2,3	69,20±1,8
	90	67,38±2,6	**72,38±1,8**	67,77±2,5	71,36±1,7
	95	66,21±2,4	**71,22±1,9**	69,57±2,3	70,56±2,0
german	0	70,21±2,0	—	76,20±2,0	—
	70	70,82±2,2	**71,99±2,1**	70,18±2,0	70,18±2,0
	80	71,06±2,1	**72,10±2,0**	70,18±2,0	70,18±2,0
	90	70,41±2,4	**71,20±1,9**	70,18±2,0	70,18±2,0
	95	70,02±2,1	**70,67±1,9**	70,18±2,0	70,20±2,0
heart	0	83,53±3,1	—	81,65±3,1	—
	70	**82,17±3,3**	82,05±2,8	80,39±3,4	75,02±5,7
	80	81,40±3,2	**81,55±2,8**	79,65±3,1	77,78±4,3
	90	78,85±3,5	79,77±3,0	76,92±3,6	76,98±3,7
	95	75,26±4,6	77,70±3,4	73,13±4,1	74,21±3,4
thyroid	0	90,09±3,7	—	94,85±2.3	—
	70	90,36±2,6	90,91±2,4	92,30±2,1	92,51±2,1
	80	90,07±2,6	90,80±2,5	91,60±2,3	92,89±2,3
	90	91,68±2,7	91,80±2,3	90,43±2,6	92,85±2,2
	95	91,39±2,9	91,88±2,4	87,44±3,6	92,70±2,3
waveform	0	87,75±1,2	—	88,68±0,6	—
	70	86,34±1,4	**88,05±0,5**	79,59±3,9	82,82±1,6
	80	84,17±1,7	**87,59±0,6**	81,24±3,3	82,88±2,0
	90	82,34±2,0	**86,37±0,9**	81,38±2,6	81,60±2,7
	95	82,24±1,6	**85,09±1,0**	80,54±2,6	80,53±2,6

Blood-platelets RNA-seq data

GEO accession number: GSE68086

Sample Type	Sample number
Healthy Donors (HD)	55
Breast Cancer (BC)	39
Colorectal (CRC)	46
Glioblastoma (GB)	40
Lung (L)	59
TOTAL	**239**

- Raw input data matrix = [239 x 16347] (samples vs transcripts)
- Elements are the spanning exon read counts

Best MG, Sol N, Kooi I, Tannous J et al. *RNA-Seq of Tumor-Educated Platelets Enables Blood-Based Pan-Cancer, Multiclass, and Molecular Pathway Cancer Diagnostics*. Cancer Cell 2015 Nov 9;28(5): 666-76. PMID: 26525104

Test settings

- Three methods: ReGEC, LapReGEC and LapSVM
- Three labeled percentages: 100%, 70% and 30%
- 100-fold cross validation
- Training set= 80%, Test set=20% of the total samples.

LapRegec: Accuracy Results

Healthy vs Tumors

Labeled	ReGEC	LapReGEC	LapSVM
100 %	88 ± 5	**91 ± 4**	90 ± 4
70 %	89 ± 4	**90 ± 4**	87 ± 4
30 %	87 ± 4	**89 ± 4**	81 ± 5

Breast Cancer vs Colorectal cancer

Labeled	ReGEC	LapReGEC	LapSVM
100 %	86.9 ± 8	**88.5 ± 6**	84.6 ± 8
70 %	85.2 ± 4.7	**87.5 ± 4**	84.5 ± 8
30 %	79.6 ± 4.5	**83.8 ± 5**	82.4 ± 8

LapRegec: Accuracy Results

Breast Cancer vs Lung cancer

Labeled	ReGEC	LapReGEC	LapSVM
100 %	86.7 ± 7	**86.9 ± 6**	84.4 ± 8
70 %	86.1 ± 6	**87.1 ± 5**	84.3 ± 8
30 %	82 ± 4.5	**86.0 ± 5**	84.2 ± 8

Glioblastome vs Lung cancer

Labeled	ReGEC	LapReGEC	LapSVM
100 %	90.4 ± 6	90.9 ± 6	**93.6 ± 6**
70 %	90.1 ± 5	91.1 ± 5	**93.5 ± 6**
30 %	89.9 ± 4.5	91.0 ± 4.5	**93.4 ± 6**

Open problems

- Dependence of generalization error on the number of labeled and unlabeled examples.
- Intelligent techniques for model selection involves choosing appropriate values for regularization parameters.
- Scalability issues are critical for large data (cubic complexity).
- Incremental semi-supervised learning.
- Semi-supervised classification of graphs.

Conclusions

- We have seen a framework for data-dependent geometric regularization.
- It is based on a Representer theorem useful for several algorithms for unsupervised, semisupervised and fully supervised learning.
- This framework brings together ideas from the theory of regularization in reproducing kernel Hilbert spaces, manifold learning and spectral methods.
- Its implementation and numerical results show these techniques can provide better generalization results.

INnovative Life sCIence PhD Programme in South ITaly INCIPIT

Let's start

Site Map Accessibility Contact

Search Site

Log in

Home | Key partners | Call for Applicants | Training activities | Supervisors | Fellows | Online Application

You are here: Home

- Key partners
- Call for Applicants
- Training activities
- Supervisors
- Fellows
- Online Application

Consiglio Nazionale delle Ricerche

Commissione europea

Home

INCIPIT is a **novel international PhD programme**, which aims at providing innovative multidisciplinary and intersectoral training in **Life and Biomaterial Sciences**.

It is co-funded by the **COFUND scheme** (Marie Skłodowska - Curie Actions).

This programme pools the different and complementary expertise of 8 research institutes of the Consiglio Nazionale delle Ricerche (CNR), as well as the competencies of 3 Doctoral Schools and several academic, non-profit and industrial Partner Organizations.

INCIPT PhD programme will include:

- laboratory activities to carry out doctoral research projects;
- theoretical courses run by the Doctoral Schools of the Second University of Naples (SUN), the University of Naples Federico II and the University of Catania (UNICT);
- activities related to transferable skills.

INCIPIT will recruit 35 Early Stage Researchers (ESRs) through 2 calls for applicants (the first in January 2016 and the second in January 2017).

Contact us

For any further information please contact the INCIPIT project managers:

Floriana Di Giacomo Monica Autiero

News

Call for Evaluators - Now Open

Jan 25, 2016

More news...

Useful Links

RESEARCH & INNOVATION
Marie Skłodowska-Curie actions

National Research Council of Italy

SECONDA UNIVERSITÀ DEGLI STUDI DI NAPOLI
unina2.it

Università degli Studi di Napoli Federico II

Università degli Studi di Catania

euraxess
RESEARCHERS IN MOTION

2 PhD fellowships