

Learning structured outputs efficiently via maximum margins

How to teach the Support Vector Machine to learn arbitrary outputs

Sandor Szedmak

ISIS, Electronics and Computer Science
University of Southampton

Oxford 09/2009

Contributors

- Katja Astikainen
- Esther Galbrun
- Juho Rousu
 - ▶ University of Helsinki
- Yizhao Ni
- Craig J. Saunders
 - ▶ University of Southampton
- John Shawe-Taylor
- Zhuoran Wang
 - ▶ University College London
- Tijl de Bie
 - ▶ University of Bristol
- others ...

Outline

- 1 Looking back
- 2 Learning strategy
- 3 Optimization model
- 4 Multiclass learning
- 5 Experiments
- 6 One step forward, beyond the positive definiteness
- 7 Extending the scope, other kind of regularizations and loss
- 8 Learning game, modelling the uncertainty
- 9 Prediction via density functions

A very short CV I.

- MS degree in Mathematics, Kossuth Lajos University, Debrecen, Hungary, 1980;
- PhD degree in Operations Research, RUTCOR, Rutgers Center For Operations Research, Rutgers, The State University of New Jersey, USA, 2002;

PhD (Operations Research, Stochastic Programming)

Title Methods for solving ill-conditioned and large scale probability bounding and discrete moment problems

Institute RUTCOR, Rutgers Center for Operations Research, Rutgers, The State University of New Jersey, USA

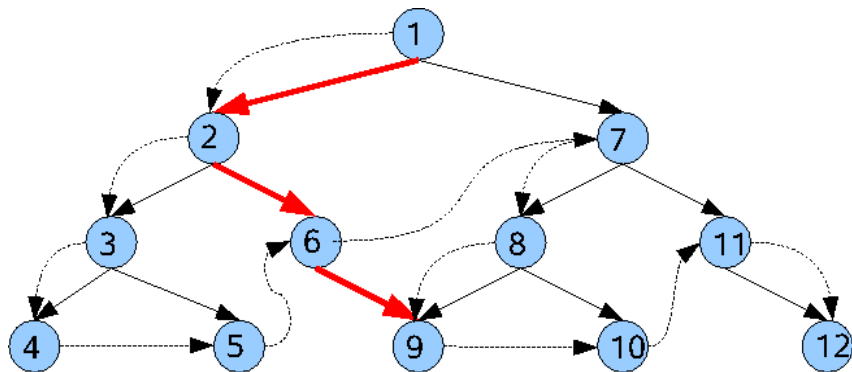
Adviser András Prékopa

A very short CV II.

- **LAVA (Learning for Adaptive Visual Assistants), EU Project** , *Royal Holloway, University of London and University of Southampton* , John Shawe-Taylor
- **Learning with Labeled and Unlabeled Data, PASCAL Network** , *University of Southampton* , John Shawe-Taylor
- **Modelling Functional Shifts in Enzyme evolution, Academy of Finland**, *University of Helsinki* , Juho Rousu
- **SMART(Statistical Multilingual Analysis for Retrieval and Translation) Eu Project** , *University of Southampton*, Craig J. Saunders, Xerox XRCE and Mahesan Niranjan, University of Southampton

How one can predict something like this ...

What about SVM?



Examples of known approaches:

- Cut the problem into several parts,
 - ▶ Apply plenty of binary classifiers ...
- Max-Margin Markov Networks,
 - ▶ Taskar(2003)
- Least-square approaches,
 - ▶ Cortes(2005)

Main problems are the high computational complexity and not too much satisfactory results.

WIPO-alpha dataset

WIPO-alpha	$l_{0/1}$	l_{Δ}	P	R	F1	Time
SVM	87.2	1.84	93.1	58.2	71.6	~ 0.5h
Hier-SVM	76.2	1.74	90.3	63.3	74.4	~ 2h
Hier-RLS	72.1	1.69	88.5	66.4	75.9	~ 1day

Table: Prediction losses $l_{0/1}$ and l_{Δ} , precision, recall and F1 values obtained using different learning algorithms. All figures, except l_{Δ} , are given as percentages. Precision and recall are computed in terms of totals of microlabel predictions in the test set.

WIPO-alpha dataset

WIPO-alpha	$l_{0/1}$	l_{Δ}	P	R	F1	Time
SVM	87.2	1.84	93.1	58.2	71.6	~ 0.5h
Hier-SVM	76.2	1.74	90.3	63.3	74.4	~ 2h
Hier-RLS	72.1	1.69	88.5	66.4	75.9	~ 1day
Hier- M^3 - l_{Δ}	70.9	1.67	90.3	65.3	75.8	40 min
Hier- M^3 - $l_{\bar{H}}$	65.0	1.73	84.1	70.6	76.7	40 min

Table: Prediction losses $l_{0/1}$ and l_{Δ} , precision, recall and F1 values obtained using different learning algorithms. All figures, except l_{Δ} , are given as percentages. Precision and recall are computed in terms of totals of microlabel predictions in the test set.

WIPO-alpha dataset

WIPO-alpha	$l_{0/1}$	l_{Δ}	P	R	F1	Time
SVM	87.2	1.84	93.1	58.2	71.6	~ 0.5h
Hier-SVM	76.2	1.74	90.3	63.3	74.4	~ 2h
Hier-RLS	72.1	1.69	88.5	66.4	75.9	~ 1day
Hier- M^3 - l_{Δ}	70.9	1.67	90.3	65.3	75.8	40 min
Hier- M^3 - $l_{\bar{H}}$	65.0	1.73	84.1	70.6	76.7	40 min
MMR _{Gauss}	46.9	1.77	77.9	77.9	77.9	1s!

Table: Prediction losses $l_{0/1}$ and l_{Δ} , precision, recall and F1 values obtained using different learning algorithms. All figures, except l_{Δ} , are given as percentages. Precision and recall are computed in terms of totals of microlabel predictions in the test set.

Learning strategy

- Embedding** where the structures of the input and output objects are represented in properly chosen spaces(Hilbert, Banach, ...).
- Optimization** has to find the similarity based matching between the input and the output representations.
- Inversion(Pre-image problem)** has to recover the best fitting output structure of its representation.

Embedding

Embedding

$$\begin{aligned}\phi &: \overbrace{\text{input space}}^{\mathcal{X}} \rightarrow \overbrace{\text{feature space}}^{\mathcal{H}_\phi} \\ \psi &: \overbrace{\text{output space}}^{\mathcal{Y}} \rightarrow \overbrace{\text{label space}}^{\mathcal{H}_\psi}\end{aligned}$$

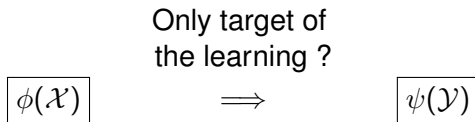
Similarity transformation

$$\tilde{\mathbf{W}} = (\mathbf{W}, \mathbf{b}) \Rightarrow \psi(\mathbf{y}) \sim \tilde{\mathbf{W}}\phi(\mathbf{x})$$

Inversion

$$\psi^{-1}(\mathbf{Y})$$

Learning as task to find “Probably, approximately isomorph” relations



Engineering ?



Engineering ?

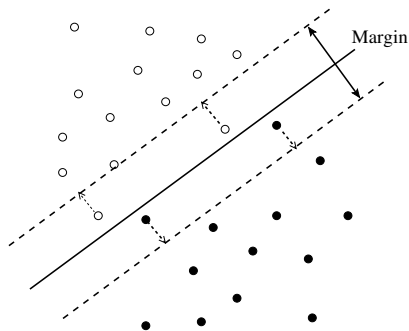
\mathcal{X}



\mathcal{Y}

The "real" question!

The “Classical” Support Vector Machine(SVM)



$$\min \frac{1}{2} \mathbf{w}^T \mathbf{w} + C \mathbf{1}^T \boldsymbol{\xi}$$

w.r.t. $\mathbf{w} : \mathcal{H}_\phi \rightarrow \mathbb{R}$, normal vec.

$b \in \mathbb{R}$, bias

$\boldsymbol{\xi} \in \mathbb{R}^m$, error vector

$$\text{s.t. } \left| y_i (\mathbf{w}^T \phi(\mathbf{x}_i) + b) \right| \geq 1 - \xi_i$$
$$\boldsymbol{\xi} \geq \mathbf{0}, i = 1, \dots, m$$

Dual problem

$$\min \quad \sum_{i,j=1}^m \alpha_i \alpha_j \underbrace{\left[\underbrace{\kappa_{ij}^Y}_{y_i y_j} \underbrace{\kappa_{ij}^\phi}_{\langle \phi(\mathbf{x}_i), \phi(\mathbf{x}_j) \rangle} \right]}_{K_{YX}} - \sum_{i=1}^m \alpha_i,$$

w.r.t. $\alpha_i \in \mathbb{R}$,

s.t. $\sum_{i=1}^m y_i \alpha_i = 0$,

$0 \leq \alpha_i \leq C, i = 1, \dots, m$.

- κ_{ij}^ϕ input kernel,
- κ_{ij}^Y **output kernel!**
- $K_{YX} = K_Y \bullet K_X$ joined kernel by element-wise product
- The objective function is a symmetric function of the input and the output.

The extended primal

$$\min \quad \frac{1}{2} \|\mathbf{W}\|_2^2 + \mathbf{C}\mathbf{1}^T \boldsymbol{\xi}$$

$$\text{w.r.t. } \begin{cases} \mathbf{W} \\ b \in \mathbb{R}, \text{ bias} \\ \boldsymbol{\xi} \in \mathbb{R}^m, \text{ error vector} \end{cases}$$

$$\text{s.t. } \begin{cases} F(\mathbf{W}; \phi(\mathbf{x}_i, \mathbf{y}_i)) + b \geq 1 - \xi_i \\ \boldsymbol{\xi} \geq \mathbf{0}, i = 1, \dots, m \end{cases}$$

- $F(\mathbf{W}; \phi(\mathbf{x}_i, \mathbf{y}_i))$ linear function of \mathbf{W} , parametrized by a function of the input and the output. It has to be monotonic, increasing function of $\|\mathbf{W}\|_2$.

Reinterpretation of the normal vector \mathbf{w}

- Original
- $y_i \in \{-1, +1\}$ binary outputs
 - \mathbf{w} is the normal vector of the separating hyperplane.
- New
- $y_i \in \mathcal{Y}$ arbitrary outputs
 - ▶ $\psi(y_i) \in \mathcal{H}_\psi$ embedded labels in a linear vector space
 - \mathbf{w}^T is a linear operator projecting the input space into the output space.
 - ▶ The aim to find the highest similarity between the output and the projected input.

The output space is a one dimensional subspace in the SVM.

Primal problems

Binary class learning

Support Vector Machine(SVM)

$$\min \frac{1}{2} \underbrace{\mathbf{w}^T \mathbf{w}}_{\|\mathbf{w}\|_2^2} + C \mathbf{1}^T \boldsymbol{\xi}$$

w.r.t. $\mathbf{w} : \mathcal{H}_\phi \rightarrow \mathbb{R}$, normal vec.

$b \in \mathbb{R}$, bias

$\boldsymbol{\xi} \in \mathbb{R}^m$, error vector

$$\text{s.t. } y_i(\mathbf{w}^T \phi(\mathbf{x}_i) + b) \geq 1 - \xi_i$$

$$\boldsymbol{\xi} \geq \mathbf{0}, i = 1, \dots, m$$

Vector label learning

Maximum Margin Robot(MMR)

$$\min \frac{1}{2} \underbrace{\text{tr}(\mathbf{W}^T \mathbf{W})}_{\|\mathbf{W}\|_{\text{Frobenius}}^2} + C \mathbf{1}^T \boldsymbol{\xi}$$

$\mathbf{W} : \mathcal{H}_\phi \rightarrow \mathcal{H}_\psi$, linear operator

$\mathbf{b} \in \mathcal{H}_\psi$, translation(bias)

$\boldsymbol{\xi} \in \mathbb{R}^m$, error vector

$$\text{s.t. } \langle \psi(\mathbf{y}_i), \mathbf{W} \phi(\mathbf{x}_i) + \mathbf{b} \rangle_{\mathcal{H}_\psi} \geq 1 - \xi_i$$

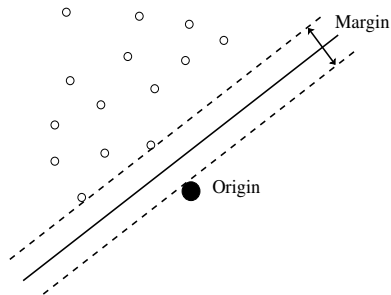
$$\boldsymbol{\xi} \geq \mathbf{0}, i = 1, \dots, m$$

One-class SVM interpretation

No bias

Let us reformulate the inner-product occurring in the constraints

$$\begin{aligned} & \langle \psi(\mathbf{y}_i), \mathbf{W}\phi(\mathbf{x}_i) \rangle_{\mathcal{H}_\psi} \\ &= \text{tr}(\psi(\mathbf{y}_i)^T \mathbf{W}\phi(\mathbf{x}_i)) \\ &= \text{tr}(\mathbf{W}\phi(\mathbf{x}_i)\psi(\mathbf{y}_i)^T) \\ &= \left\langle \mathbf{W}, [\psi(\mathbf{y}_i) \otimes \phi(\mathbf{x}_i)] \right\rangle_{\mathcal{H}_\psi \otimes \mathcal{H}_\phi} \end{aligned}$$



thus, we have a one-class SVM problem living in the tensor product space of the output and the input.

(\otimes denotes the tensor product)

One-class SVM interpretation

One step further ...

One can extend the range of applications by using not only tensor product but more general relationship between the output and input, i.e.,

$$\left\langle \mathbf{W}, \Psi(\mathbf{y}_i, \mathbf{x}_i) \right\rangle_{\mathcal{H}_W}, \quad \Psi : \mathcal{H}_\psi \times \mathcal{H}_\phi \rightarrow \mathcal{H}_W.$$

If $\mathbf{dim}(\mathcal{H}_W) > \mathbf{dim}(\mathcal{H}_\psi) + \mathbf{dim}(\mathcal{H}_\phi)$ then the support of the distribution of one-class sample items is restricted on a manifold in \mathcal{H}_W .

Alternative linear functions of \mathbf{W}

They are subversions of the general case, but they can better express some kind of relationship between the input and output.

- $\mathbf{x}_i, \mathbf{y}_i$ matrices, Ψ covers the operation of matrix product. It allows to use sample items with different dimensionality.
- $\mathbf{x}_i, \mathbf{y}_i$ matrices of same size, Ψ covers the operation of point-wise product.
- $\mathbf{x}_i, \mathbf{y}_i$ are taken from an algebra with special properties, e.g. Clifford, Jordan, Ψ expresses the product operation of the algebra. They can represent complex structures.

Advantage of the tensor product

- The identity

$$\langle \mathbf{x}_i \otimes \mathbf{y}_i, \mathbf{x}_j \otimes \mathbf{y}_j \rangle = \langle \mathbf{x}_i, \mathbf{x}_j \rangle \langle \mathbf{y}_i, \mathbf{y}_j \rangle$$

allows us

- ▶ to separate the input and output kernels,
- ▶ to work with vectors which may have infinite number of components, they can be functions, e.g. probability densities, generalized functions - Dirac δ -, etc..

Dual problem

$$\begin{aligned} \min \quad & \sum_{i,j=1}^m \alpha_i \alpha_j \overbrace{\langle \phi(\mathbf{x}_i), \phi(\mathbf{x}_j) \rangle}^{\kappa_{ij}^\phi} \overbrace{\langle \psi(\mathbf{y}_i), \psi(\mathbf{y}_j) \rangle}^{\kappa_{ij}^\psi} - \sum_{i=1}^m \alpha_i, \\ \text{w.r.t.} \quad & \alpha_i \in \mathbb{R}, \\ \text{s.t.} \quad & \boxed{\sum_{i=1}^m (\psi(\mathbf{y}_i))_t \alpha_i = 0, \quad t = 1, \dots, \dim(\mathcal{H}_\psi)}, \quad \text{Only if bias is used} \\ & 0 \leq \alpha_i \leq C, \quad i = 1, \dots, m. \end{aligned}$$

- κ_{ij}^ϕ input kernel,
- κ_{ij}^ψ output kernel
- The objective function is a symmetric function of the input and the output.

To get rid of occurrences of explicit labels ...

The explicit occurrences of the label vectors can be transformed into implicit ones¹:

$$\begin{aligned} \sum_{i=1}^m (\psi(\mathbf{y}_i))_t \alpha_i &= 0, \quad t = 1, \dots, \dim(\mathcal{H}_\psi), \\ \Downarrow \\ \sum_{i=1}^m \kappa_{ij}^\psi \alpha_i &= 0, \quad j = 1, \dots, m \end{aligned}$$

This transformation preserves the feasibility domain!

¹Tijl De Bie, Private conversation

Prediction

No bias

The linear operator:

$$\mathbf{W} = \sum_{i=1}^m \alpha_i \boldsymbol{\psi}(\mathbf{y}_i) \boldsymbol{\phi}(\mathbf{x}_i)^T$$

Prediction in the label space:

$$\begin{aligned} \boldsymbol{\psi}(\mathbf{y}) &= \mathbf{W} \boldsymbol{\phi}(\mathbf{x}) \\ &= \sum_{i=1}^m \alpha_i \boldsymbol{\psi}(\mathbf{y}_i) \underbrace{\langle \boldsymbol{\phi}(\mathbf{x}_i), \boldsymbol{\phi}(\mathbf{x}) \rangle}_{\kappa^{\boldsymbol{\phi}}(\mathbf{x}_i, \mathbf{x})} \end{aligned}$$

Prediction when the labels are implicit

An approach

Assume the set of outcomes is known

$\mathbf{y} \in \tilde{\mathcal{Y}} \iff$ Set of the possible outputs

$$\mathbf{y}^* = \arg \max_{\mathbf{y} \in \tilde{\mathcal{Y}}} \boldsymbol{\psi}(\mathbf{y})^T \mathbf{W} \boldsymbol{\phi}(\mathbf{x})$$

$$= \arg \max_{\mathbf{y} \in \tilde{\mathcal{Y}}} \sum_{i=1}^m \alpha_i \overbrace{\langle \boldsymbol{\psi}(\mathbf{y}), \boldsymbol{\psi}(\mathbf{y}_i) \rangle}^{\kappa^{\boldsymbol{\psi}}(\mathbf{y}, \mathbf{y}_i)} \overbrace{\langle \boldsymbol{\phi}(\mathbf{x}_i), \boldsymbol{\phi}(\mathbf{x}) \rangle}^{\kappa^{\boldsymbol{\phi}}(\mathbf{x}_i, \mathbf{x})}$$

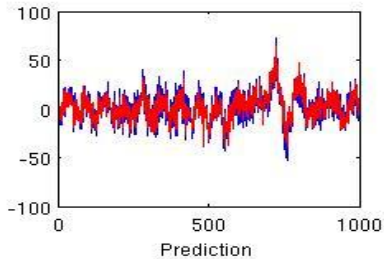
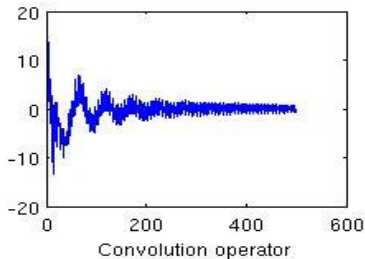
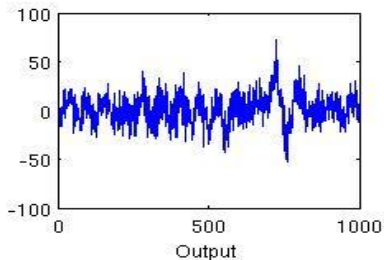
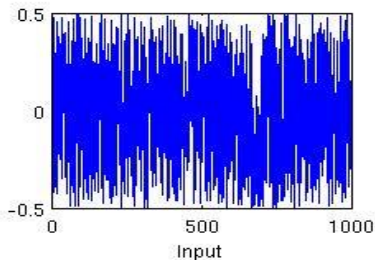
Finite outcome

$$\mathbf{y} \in \tilde{\mathcal{Y}} = \{\mathbf{y}_1, \dots, \mathbf{y}_K\}, \quad K \ll \infty$$

The best candidate for $\tilde{\mathcal{Y}}$ could be the training set!

Learning convolution operator

input windows \Rightarrow output windows



Representation of multiclass output

- **Indicators**, e.g.: 3 classes $\Rightarrow \{(1, 0, 0), (0, 1, 0), (0, 0, 1)\}$,
- **Vectors pointing into the class centers**,
Class centers can be means or medians,
- **Vertices of hyper-tetrahedron** Vectors with unit length and with minimum pair-wise correlation.

The experiments favour the hyper-tetrahedron, it is the most “symmetric” structure.

Vertices of hyper-tetrahedron

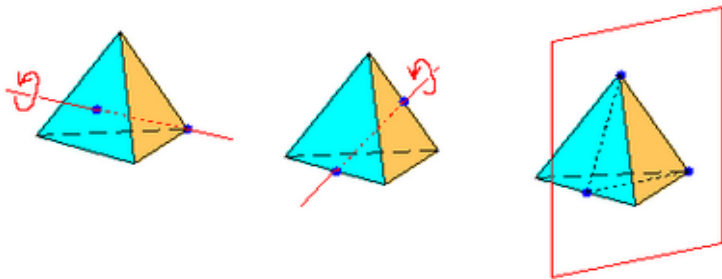
n-class case:

Consider the matrix \mathbf{V} with elements:

$$V_{ij} = \begin{cases} 1 & \text{if } i = j, \\ -\frac{1}{n-1} & \text{otherwise.} \end{cases}$$

The labels are rows of the matrix \mathbf{A} which satisfies $\mathbf{V} = \mathbf{A}\mathbf{A}^T$.

One eigenvalue of \mathbf{V} is zero, thus \mathbf{A} has n rows but $n - 1$ columns only.



* www.wikipedia.org/wiki/tetrahedron

Experiments

Multiclass classification

Name	Test error rate (%)							
	SVM		MMR					
	all vs. all	one	hyper-tetrahedron			indicator		
			—	item	variable	—	item	variable
abalone *	72.3	79.7	73.0	73.0	73.4	73.9	73.0	74.1
glass	30.4	30.8	27.3	27.6	29.2	26.4	29.0	29.0
optdigits *	3.8	2.7	2.0	1.6	3.3	2.1	1.9	3.3
page-blocks	3.4	3.4	4.4	3.4	3.7	4.5	3.6	3.3
satimage *	8.2	7.8	8.2	17.5	8.6	8.7	17.7	9.1
spectrometer	42.8	53.7	99.5	37.5	53.9	99.6	38.4	53.3
yeast	41.0	40.3	41.6	40.6	40.3	42.6	41.6	40.9

Table: Test error rates (%). If the data set has dedicated training and test subsets, marked with *, then the table shows the accuracy computed on the given test subset otherwise the presented accuracies are averages computed via 5-fold cross-validation.

From image to annotation

Images: 695, Words: 132

Method	Properties	Precision	Recall	F1	Comp. Time(s)
KCCA	Linear, 10 factor	44.9(1.4)	36.4(1.5)	40.2(1.3)	0.59
KCCA	Poly(2,0), 10 factor	43.2(2.5)	34.9(2.3)	38.6(2.3)	0.59
KCCA	Gauss(0.8), 10 factor	25.3(10.4)	20.2(8.2)	21.8(7.7)	0.58
MMR	Linear, unnorm.	24.3(1.8)	32.1(3.2)	27.6(2.2)	0.03
MMR	Linear, norm.	45.7(1.1)	37.9(2.1)	41.4(1.4)	0.39
MMR	Poly(5,0), norm.	51.1(1.8)	50.1(2.2)	50.6(1.9)	0.18
MMR	Gauss(0.25), norm.	50.0(2.6)	49.4(1.6)	49.7(1.4)	0.11
Random baseline		3.63			

Table: Precision, Recall and F1 measures provided by the different methods are in percentages and the computational times in seconds. () contains the corresponding Standard Deviation. Properties show the types of the kernels, the number of the factors considered in KCCA, and when the data vectors were or were not normalised to the length of one. The time data is received by MATLAB code on a Pentium machine of 2.2 GHz.

Performance measures

Correctness of the path

$l_{0/1}$ Zero-one loss

l_{Δ} Symmetric difference loss

P Precision

R Recall

F1 Combination of the Precision and Recall

$$\Rightarrow \frac{2PR}{P+R}$$

Methods

SVM Flat SVM

H-SVM Node-wise SVM

H-RLS Hierarchical least square (Cesa-Bianchi)

H-M³ - I_{Δ} H-M³ trained on ℓ_{Δ} (Rousu)

H-M³ - $I_{\bar{H}}$ H-M³ trained on subtree loss (Rousu)

MMR_{lin} Proposed method with linear input kernel

MMR_{poly} Proposed method with polynomial kernel

Enzyme EC-feature dataset

Enzyme: 5934, nodes: 1376+258

Enzyme-EC	$l_{0/1}$	F1
3-levels, 236 nodes		
SVM	99.7	58.2
h-SVM	98.5	58.7
h-rls	95.6	53.3
h-m ³ - l_{Δ}	95.7	63.3
h-m ³ - $l_{\bar{H}}$	85.5	53.4
5-levels, 1376+258 nodes		
h-m ³ - <i>poly</i>	14.2	91.7
MMR _{lin}	16.9	90.0
MMR _{poly}	14.2	91.7

Table: Prediction losses $l_{0/1}$ and l_{Δ} , precision, recall and F1 values obtained using different learning algorithms. All figures, except l_{Δ} , are given as percentages. Precision and recall are computed in terms of totals of microlabel predictions in the test set.

Enzyme prediction with special kernels and methods

Enzymes: 3090, nodes: 493+5

- Accuracies on test:

Measure	Kernels	Nearest neighbour	MMR linear	MMR poly-51	HM ³ linear	HM ³ poly-51
F1	GTG	88.0	81.9	89.3.4	90.2	89.6
0/1-loss	GTG	24.1	36.3	21.4	23.3	21.6

- Kernels are

GTG protein 3D structure,

- Presented on

Machine Learning in Systems Biology (MLSB-2007) Evry, France, [1], The full version published in BMC.

Reformulation of the primal problem

The equality

$$\langle \psi(\mathbf{y}), \mathbf{W}\phi(\mathbf{x}) \rangle_{\psi} = \langle \mathbf{W}, \phi(\mathbf{x})\psi(\mathbf{y})^T \rangle_F$$

and

$$\mathbf{W} = \sum_{k=1}^{m_k} \alpha_k \psi(\mathbf{y}_k) \phi(\mathbf{x}_k)^T$$

give the constraints, where α the dual variable replaced with \mathbf{u} to distinct the primal.

$$\sum_{k=1}^{m_k} u_k \overbrace{\langle \psi(\mathbf{y}_i), \psi(\mathbf{y}_k) \rangle}^{\kappa^{\psi}(\mathbf{y}_i, \mathbf{y}_k)} \overbrace{\langle \phi(\mathbf{x}_k), \phi(\mathbf{x}_i) \rangle}^{\kappa^{\phi}(\mathbf{x}_k, \mathbf{x}_i)} \geq 1 - \xi_i, \quad i = 1, \dots, m$$
$$\mathbf{C} \geq u_k \geq 0, \quad k = 1, \dots, m_k$$

Reparametrization of the primal problem

Unbiased case

$$\min \frac{1}{2} \langle \mathbf{W}'\mathbf{W} \rangle_F + \mathbf{C}\mathbf{1}'\xi$$

$$\left| \frac{1}{2} \mathbf{u}'\mathbf{Q}\mathbf{u} + \mathbf{C}\mathbf{1}'\xi \right.$$

$$\text{w.r.t. } \begin{cases} \mathbf{W} : \mathcal{H}_\phi \rightarrow \mathcal{H}_\psi, \\ \xi \in \mathbb{R}^m, \end{cases}$$

$$\left| \begin{cases} \mathbf{u} \in \mathbb{R}^{n_r} \\ \xi \in \mathbb{R}^m, \end{cases} \right.$$

$$\text{s.t. } \langle \psi(\mathbf{y}_i), \mathbf{W}\phi(\mathbf{x}_i) \rangle_{\mathcal{H}_\psi} \geq 1 - \xi_i$$

$$\xi \geq \mathbf{0}, i = 1, \dots, m$$

$$\left| \begin{cases} \sum_{r=1}^{n_r} u_r \kappa_{ir}^\psi \kappa_{ri}^\phi \geq 1 - \xi_i \\ \xi \geq \mathbf{0}, i = 1, \dots, m \end{cases} \right.$$

$$\kappa_{ir}^{H_\psi} = \langle \psi(y_i), \psi(y_r) \rangle_{\mathbf{H}_\psi}$$

$$\kappa_{ri}^{H_\phi} = \langle \phi(x_r), \phi(x_i) \rangle_{\mathbf{H}_\phi}$$

$$\left[\mathbf{K}_{H_\psi} \right]_{pq} = \kappa_{pq}^{H_\psi}$$

$$\left[\mathbf{K}_{H_\phi} \right]_{rs} = \kappa_{rs}^{H_\phi}$$

$$\mathbf{Q} = \mathbf{K}_{H_\psi} \bullet \mathbf{K}_{H_\phi} \text{ or } \mathbf{I}$$

Similar reparametrization proposed by Mangasarian for the binary SVM [4].

One class form

- Let \mathbf{G} be a matrix such that

$$G_{ij} = (\Psi(x_i, y_i))_j = \kappa_{ij}^\psi \kappa_{ij}^\phi = \langle \psi(y_i), \psi(y_j) \rangle \langle \phi(x_i), \phi(x_j) \rangle$$

and we have the base problem:

	Primal		Dual
min	$\frac{1}{2} \ \mathbf{u}\ _2^2 + \mathbf{C}\mathbf{1}'\xi$	min	$\frac{1}{2} \alpha' \mathbf{G}\mathbf{G}' \alpha - \mathbf{1}' \alpha$
w.r.t.	\mathbf{u}, ξ	w.r.t.	$\alpha \in \mathbb{R}^m,$
s.t.	$\mathbf{G}\mathbf{u} \geq \mathbf{1} - \xi,$	s.t.	$\mathbf{0} \leq \alpha \leq \mathbf{C}\mathbf{1}$
	$\xi \geq \mathbf{0}$		

!A consequence!

- **All the information known about the data incorporated into the matrix G**
 - ⇒ **similar to the kernel trick!**
- But what kind of matrices are the proper ones to choose them as data descriptors?

!A consequence!

- **All the information known about the data incorporated into the matrix G**
 - ⇒ **similar to the kernel trick!**
- **But what kind of matrices are the proper ones to choose them as data descriptors?**

Properties of \mathbf{G}

\mathbf{G} is not restricted to be

Positive (Semi)Definite It can contain non-definite inner products, e.g. Minkowski or Hyperbolic geometry,

Symmetric It can contain anti-symmetric inner products, i.e.
 $\langle a, b \rangle = -\langle b, a \rangle$

Square matrix

The structures processed in a learning task might have very irregular geometrical properties²

- they are not vectors of a Hilbert space, or
- they can not be approximated by this kind of objects.

²See Pekalska (2005) [9]

Extending the scope, other kind of regularizations

- Let us change the regularization and the loss terms

$$\begin{aligned} \min \quad & \frac{1}{2} \mathcal{R}(\mathbf{w}) + \mathcal{L}(\boldsymbol{\xi}) \\ \text{w.r.t.} \quad & \mathbf{w}, \boldsymbol{\xi} \\ \text{s.t.} \quad & \mathbf{G}\mathbf{w} \geq \mathbf{1} - \boldsymbol{\xi} \\ & \boldsymbol{\xi} \geq \mathbf{0}, \end{aligned}$$

where $\mathcal{R}(\cdot)$ and $\mathcal{L}(\cdot)$ might be $\|\cdot\|_1$, $\|\cdot\|_2$, $\|\cdot\|_2^2$, $\|\cdot\|_\infty$ and any reasonable measures of regularisation.

Examples for the matrix \mathbf{G}

Similarity case: large values of G_{ij} mean high similarity

- e.g. inner product³:

$$G_{ij} = \overbrace{\langle \psi(\mathbf{y}_i), \psi(\mathbf{y}_j) \rangle}^{s^\psi(\mathbf{y}_i, \mathbf{y}_j)} \overbrace{\langle \phi(\mathbf{x}_j), \phi(\mathbf{x}_i) \rangle}^{s^\phi(\mathbf{x}_j, \mathbf{x}_i)}$$

- Using inverse distances, potential functions

$$G_{ij} = \frac{1}{1 + d^2(\psi(\mathbf{y}_i), \psi(\mathbf{y}_j))d^2(\phi(\mathbf{x}_j), \phi(\mathbf{x}_i))}$$

³Mangasarian (1998): Generalized SVM [4]

Can we use distances or any dissimilarity measures?

Dissimilarity case: small values of G_{ij} mean high similarity

- e.g. distances:

$$G_{ij} = d(\psi(\mathbf{y}_i), \psi(\mathbf{y}_j))d(\phi(\mathbf{x}_j), \phi(\mathbf{x}_i))$$

We need to change the regularization strategy!

Predictions, a plausible approach

- Conjecture that maximizing (minimizing) the margin gives the best answer.
- Assume that the set of the possible outputs is $\tilde{\mathcal{Y}}$.
- Similarity case:

$$\mathbf{y}_* = \arg \max_{\mathbf{y} \in \tilde{\mathcal{Y}}} \sum_{k=1}^{m_k} w_k \overbrace{\langle \psi(\mathbf{y}), \psi(\mathbf{y}_k) \rangle}^{s^\psi(\mathbf{y}, \mathbf{y}_k)} \overbrace{\langle \phi(\mathbf{x}_k), \phi(\mathbf{x}) \rangle}^{s^\phi(\mathbf{x}_k, \mathbf{x})}$$

- Dissimilarity case:

$$\mathbf{y}_* = \arg \min_{\mathbf{y} \in \tilde{\mathcal{Y}}} \sum_{i=1}^m \alpha_i \overbrace{d(\psi(\mathbf{y}), \psi(\mathbf{y}_i))}^{d^\psi(\mathbf{y}, \mathbf{y}_i)} \overbrace{d(\phi(\mathbf{x}_i), \phi(\mathbf{x}))}^{d^\phi(\mathbf{x}_i, \mathbf{x})}$$

Alternatives

... just a starting collection

Type of Learning	Learning Algorithm	\mathcal{R}	\mathcal{L}	\mathbf{G}	\mathbf{g}	\mathcal{W}
Classification	SVM	$\frac{1}{2} \ \mathbf{w}\ _2^2$	$\ \xi_+\ _1$	$G_{ij} = y_i(\phi(x_i))_j$	$g_i = 1$	
	LS-SVM	$\frac{1}{2} \ \mathbf{w}\ _2^2$	$\ \xi\ _2^2$	$G_{ij} = y_i(\phi(x_i))_j$	$g_i = 1$	
	LPBoost	$\ \mathbf{w}\ _1$	$\ \xi_+\ _1$	$G_{ij} = y_i f_j(x_i)$	$g_i = 1$	$\mathbf{w} \geq \mathbf{0}$
	LP Machine	$\ \mathbf{w}\ _1$	$\ \xi_+\ _1$	$G_{ij} = y_i(\phi(x_i))_j$	$g_i = 1$	
Regression	Ridge Reg.	$\frac{1}{2} \ \mathbf{w}\ _2^2$	$\ \xi\ _2^2$	$G_{ij} = (x_i)_j$	$g_i = y_i$	
	Lasso	$\ \mathbf{w}\ _1$	$\ \xi\ _2^2$	$G_{ij} = (x_i)_j$	$g_i = y_i$	
Structured	MMR_{base}	$\frac{1}{2} \ \mathbf{w}\ _2^2$	$\ \xi_+\ _1$	$G_{ij} = (\phi(x_i) \otimes \psi(y_i))_j$	$g_i = 1$	
	$\text{MMR}_{\text{sim.}}$	$\frac{1}{2} \ \mathbf{w}\ _2^2$	$\ \xi_+\ _1$	$G_{ij} = \mathbf{s}_y(\psi(y_i), \psi(y_j)) \mathbf{s}_x(\phi(x_i), \phi(x_j))$	$g_i = 1$	
	$\text{MMR}_{\text{dissim.}}$	$-\ \mathbf{w}\ _1$	$\ \xi_-\ _1$	$G_{ij} = \mathbf{d}_y(\psi(y_i), \psi(y_j)) \mathbf{d}_x(\phi(x_i), \phi(x_j))$	$g_i = 1$	

Table: Several different algorithms that can be described under our framework for classification, regression and structured output learning.

Let's play!

The game

We are given

- two players,
- a payoff matrix \mathbf{G} .

	Player 1			
Player 2	-2	-1	0	= \mathbf{G}
	-1	0	1	
	0	1	2	

- Player 1 chooses a column index j and
- Player 2 chooses a row index i then
- Player 1 gains G_{ij} and
- Player 2 loses the same.

It is called: two players, zero-sum game.

See von Neumann (1928) [8].

Let's play!

Repeat the game

- Players have to be unpredictable otherwise they can lose
- They change their choice of indices

The strategies:

Choose column or row with a certain probabilities.

- Player 1 chooses j with probability a_j and
- Player 2 chooses i with probability d_i .

They are called mixed strategies.

The learning game

- Choose g_{ij} as $y_i h_j(\mathbf{x}_i)$!
 - ▶ $g_{ij} > 0$ if y_i and $h_j(\mathbf{x}_i)$ agree in sign and
 - ▶ $g_{ij} < 0$ if y_i and $h_j(\mathbf{x}_i)$ are distinct.

\mathbf{G} is a real payoff for Player 1.

- The expected payoff for Player 1 equals to

$$\begin{aligned} & \sum_{ij} G_{ij} \mathbf{Prob}(\text{Player 1} = j, \text{Player 2} = i) \\ & = \sum_{ij} G_{ij} a_j d_i, \end{aligned}$$

since the player choices are independent by definition.

- Player 1 tries to maximize, player 2 tries to minimize this value.

L_1 norm regularization, \Rightarrow linear programming

The learning game

Players

Learner(1)

Nature(2)

Strategies

Find the best weights
for weak learners!

Find the worst distribution,
the weights to the data!

$$\max_w \min_{\alpha} \sum_{ij} \alpha_i \mathbf{G}_{ij} w_j = \min_{\alpha} \max_w \sum_{ij} \alpha_i \mathbf{G}_{ij} w_j$$

$$\sum_j w_j = 1, w_j \geq 0, j = 1, \dots, n,$$

$$\sum_i \alpha_i = 1, \alpha_i \geq 0, i = 1, \dots, m.$$

\mathbf{w} Learner strategy

α Nature strategy

J. Neumann, 1928

Boosting

Given

- a sample $\mathcal{S} = \{y_i, \mathbf{x}_i\}$, $i = 1, \dots, m$, where
 - $y_i \in \{-1, 1\}$ are the labels that we are going to predict,
 - $\mathbf{x}_i \in \mathbb{R}^{n_x}$ are input vectors,
- a set of so called weak learners
 - $\mathcal{H} = \{h_j : \mathbf{x} \rightarrow \{-1, 1\}, j = 1, \dots, n\}$,
 - assume if $h_j \in \mathcal{H}$ then $-h_j \in \mathcal{H}$.

Let $h_{ij} \doteq h_j(\mathbf{x}_i)$.

We are looking for a predictor, a decision function, as a convex combination of the weak learners

$$f(x) = \sum_j a_j h_j(x), \quad \sum_j a_j = 1, \quad a_j \geq 0,$$

which can outperform the prediction capability of the weak learners.
See Schapire (2002) [10].

Linear Programming Boosting

How to solve

The point of view of the learner, Player 1, is:

$$\begin{aligned} \max_a \min_d \quad & \sum_{ij} g_{ij} a_j d_i \\ \text{s.t.} \quad & \sum_j a_j = 1, \quad a_j \geq 0, \quad j = 1, \dots, m, \\ & \sum_i d_i = 1, \quad d_i \geq 0, \quad i = 1, \dots, n. \end{aligned}$$

It boils down into a primal, point of view of the learner, and a dual problem, point view of the nature, where $g_{ij} = y_i h_j(x_i)$

$$\begin{aligned} & \text{Primal} \\ \max_{\rho, a} \quad & \rho \\ \text{s.t.} \quad & \sum_{j=1}^n g_{ij} a_j \geq \rho, \\ & \quad \quad \quad i = 1, \dots, m, \\ & \sum_j a_j = 1, \quad a_j \geq 0, \end{aligned}$$

Learner

$$\begin{aligned} & \text{Dual} \\ \min_{\beta, d} \quad & \beta \\ \text{s.t.} \quad & \sum_{i=1}^m g_{ij} d_i \leq \beta, \\ & \quad \quad \quad j = 1, \dots, n, \\ & \sum_i d_i = 1, \quad d_i \geq 0. \end{aligned}$$

Nature

Handling missing views(values)

	Views			
Source spaces:	\mathcal{Z}_1	\mathcal{Z}_2	\mathcal{Z}_3	\mathcal{Z}_4
	\Downarrow	\Downarrow	\Downarrow	\Downarrow
Complete items:	\mathbf{z}_1^1	\mathbf{z}_1^2	\mathbf{z}_1^3	\mathbf{z}_1^4
(Training)	\vdots	\vdots	\vdots	\vdots
	\mathbf{z}_m^1	\mathbf{z}_m^2	\mathbf{z}_m^3	\mathbf{z}_m^4
Incomplete items:	\mathbf{z}_{m+1}^1	\mathbf{z}_{m+1}^2	.	\mathbf{z}_{m+1}^4
(Test)	.	.	\mathbf{z}_{m+2}^3	\mathbf{z}_{m+2}^4
	.	\mathbf{z}_{m+3}^2	.	.
	\mathbf{z}_{m+4}^1	.	.	\mathbf{z}_{m+4}^4
	.	\mathbf{z}_{m+5}^2	\mathbf{z}_{m+5}^3	.
	.	\mathbf{z}_{m+6}^2	\mathbf{z}_{m+6}^3	\mathbf{z}_{m+6}^4
	\mathbf{z}_{m+7}^1	.	\mathbf{z}_{m+7}^3	.
	.	.	.	\mathbf{z}_{m+8}^4
	\vdots	\vdots	\vdots	\vdots

Multiview learning

Underlying problem

$$\begin{aligned} \min \quad & \frac{1}{2} \|\mathbf{W}\|_F^2 + C \sum_{i=1}^m \xi_i \\ \text{w.r.t.} \quad & \mathbf{W} \text{ tensor} \in \mathcal{Z}^*, \boldsymbol{\xi} \in \mathbb{R}^m, \\ \text{s.t.} \quad & \left\langle \underbrace{\bigotimes_{s \in \mathcal{R}_Y} \mathbf{z}_i^s}_{\text{Outputs}}, \mathbf{W} \underbrace{\bigotimes_{r \in \mathcal{R}_X} \mathbf{z}_i^r}_{\text{Inputs}} \right\rangle_F \geq 1 - \xi_i, \\ & \xi_i \geq 0, i = 1, \dots, m, \end{aligned}$$

\otimes tensor product

Hilbert space as container of the features vectors

- Let \mathcal{H}_ϕ and \mathcal{H}_ψ be Hilbert spaces of the square integrable functions on \mathcal{X} and \mathcal{Y} respectively,

$$\int_{u \in \mathcal{X}} f_\phi(u)^2 du < \infty, f_\phi \in \mathcal{H}_\phi$$

$$\int_{v \in \mathcal{Y}} f_\psi(v)^2 dv < \infty, f_\psi \in \mathcal{H}_\psi$$

Inner product, tensor product

Inner product For two functions of the same space:

$$\langle f, g \rangle = \int_{u \in \mathcal{X}} f(u)g(u)du, \quad f, g \in \mathcal{H}_\phi$$

and similarly in case of \mathcal{Y} and \mathcal{H}_ψ .

Tensor product For two functions of the same space:

$$f(u) \otimes g(v) = h(u, v) = f(u)g(v), \quad u, v \in \mathcal{X}, \quad f, g \in \mathcal{H}_\phi,$$

and we have the same for \mathcal{Y} and \mathcal{H}_ψ

Linear operator

$$\hat{f} = \mathbf{W}f = \int_{u \in \mathcal{X}} W(v, u)f(u)du$$

Norms

L_1

$$\|f\|_1 = \int_{u \in \mathcal{X}} |f(u)| du, f \in \mathcal{H}_\phi$$

and similarly in case of \mathcal{Y} and \mathcal{H}_ψ .

L_2

$$\|f\|_2^2 = \int_{u \in \mathcal{X}} f(u)^2 du, f \in \mathcal{H}_\phi$$

and similarly in case of \mathcal{Y} and \mathcal{H}_ψ .

Operator norm

$$\|W\|_2^2 = \int_{u \in \mathcal{X}} \int_{v \in \mathcal{X}} W(u, v)^2 dv du,$$

Building blocks of the optimization problem

- Linear functional in the constraints reads as:

$$\begin{aligned}\langle \psi(\mathbf{y}_i), \mathbf{W}\phi(\mathbf{x}_i) \rangle_{\mathcal{H}_\psi} &= \int_{v \in \mathcal{Y}} f_\psi(v) \int_{u \in \mathcal{X}} W(v, u) f_\phi(u) du dv \\ &= \int_{v \in \mathcal{Y}} \int_{u \in \mathcal{X}} W(v, u) f_\phi(u) f_\psi(v) du dv \\ &= \left\langle \mathbf{W}, [\psi(\mathbf{y}_i) \otimes \phi(\mathbf{x}_i)] \right\rangle_{\mathcal{H}_\psi \otimes \mathcal{H}_\phi}\end{aligned}$$

- Semi-infinite primal but **finite dual!**
- After fixing a basis in the space $\mathcal{H}_\otimes = \mathcal{H}_\psi \otimes \mathcal{H}_\phi$, the space of \mathbf{W} will be isomorph to \mathcal{H}_\otimes , thus, relative to the subspace of \mathcal{H}_\otimes , spanned by embedded sample items, the linear operator \mathbf{W} can be expressed as a linear combination:

$$\mathbf{W} = \sum_{i=1}^m \beta_i [\psi(\mathbf{y}_i) \otimes \phi(\mathbf{x}_i)]$$

Probability density functions, $\mathcal{D} \subset \mathcal{H}$

- Non-negativity

$$f(u) \geq 0, f \in \mathcal{D}, u \in \mathcal{X}$$

- The integral is equal to 1

$$\|f\|_1 = \int_{u \in \mathcal{X}} |f(u)| du = \int_{u \in \mathcal{X}} f(u) du,$$

- If $f, g \in \mathcal{D}$ then $f \otimes g$ is a two-variate joint density of those density functions since:

$$\int_{u \in \mathcal{X}} \int_{v \in \mathcal{X}} f(u)g(v) dv du = 1$$

Functional features, $\mathcal{F} \subset \mathcal{D} \subset \mathcal{H}$

Framework

- Let \mathcal{H}_ϕ and \mathcal{H}_ψ be Hilbert space of the square integrable functions on \mathcal{X} and \mathcal{Y} respectively, and the embedding of the input and output are given by

$$\begin{aligned}\phi : \mathcal{X} &\rightarrow \mathcal{F}_\phi (\subset \mathcal{D}_\phi \subset \mathcal{H}_\phi), \\ \psi : \mathcal{Y} &\rightarrow \mathcal{F}_\psi (\subset \mathcal{D}_\psi \subset \mathcal{H}_\psi),\end{aligned}$$

- The embedding follows the rules

$$\begin{aligned}\phi(x) &= f_\phi(\cdot|x, \Theta_{H_\phi}), \quad f_\phi \in \mathcal{F}_\phi, \quad f_\phi : \mathcal{X} \rightarrow \mathbb{R} \\ \psi(y) &= f_\psi(\cdot|y, \Theta_{H_\psi}), \quad f_\psi \in \mathcal{F}_\psi, \quad f_\psi : \mathcal{Y} \rightarrow \mathbb{R}\end{aligned}$$

where we have for the parameters of the functions

- x and y corresponding to the vectors of the sample items, they provide localization,
- Θ_{H_ϕ} and Θ_{H_ψ} are fixed for all sample items. they give scale, or “shape” to the functions.

Functional features, a concrete case

A possible simple structure of the feature and label spaces

- The sample item related parameters described by translations

$$\begin{aligned}\phi(x) &= f_\phi(\cdot|x, \Theta_{H_\phi}) = f_\phi(u - x|\Theta_{H_\phi}) \\ &= \mathbf{T}(x)f_\phi(u), \quad \text{for all } u \in \mathcal{X}\end{aligned}$$

$$\begin{aligned}\psi(y) &= f_\psi(\cdot|y, \Theta_{H_\psi}) = f_\psi(v - y|\Theta_{H_\psi}) \\ &= \mathbf{T}(y)f_\psi(v) \quad \text{for all } v \in \mathcal{Y}\end{aligned}$$

- Possible choices for the role of the parameters x and y :
 - ▶ They can be expected values: $E[f_\phi(u - x|\Theta_{H_\phi})] = x$ and $E[f_\psi(v - y|\Theta_{H_\psi})] = y$,
 - ▶ They can be medians, if there is no expected value,
 - ▶ or give the maximums.

Functional features

Example

- Let $\mathcal{X} = \mathbb{R}$ and the functions be chosen as Gaussian density functions with fixed variance σ^2 .
- The input items x give the expected value to these functions
-

$$\begin{aligned}\langle \phi(x_i), \phi(x_j) \rangle &= \int_{-\infty}^{\infty} \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(t-x_i)^2}{2\sigma^2}} \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(t-x_j)^2}{2\sigma^2}} dt \\ &= \frac{1}{\sqrt{2\pi}\sqrt{2}\sigma} e^{-\frac{(x_i-x_j)^2}{4\sigma^2}}\end{aligned}$$

gives a Gaussian kernel with width $4\sigma^2$, and multiplied by a constant.

Prediction

Margin maximization

- Margin maximization

$$\begin{aligned}\mathbf{y}^* &= \arg \max_{\{y | \psi(y) \in \mathcal{F}\}} \psi(\mathbf{y})^T \mathbf{W} \phi(\mathbf{x}) \\ &= \arg \max_{\{y | \psi(y) \in \mathcal{F}\}} \sum_{i=1}^m \alpha_i \overbrace{\langle \psi(\mathbf{y}), \psi(\mathbf{y}_i) \rangle}^{\kappa^\psi(\mathbf{y}, \mathbf{y}_i)} \overbrace{\langle \phi(\mathbf{x}_i), \phi(\mathbf{x}) \rangle}^{\kappa^\phi(\mathbf{x}_i, \mathbf{x})}\end{aligned}$$

Prediction

Probabilistic approach

- The prediction is a mixture of the densities of the training outputs, where the mixture weights computed on the dual variables and the inner products of the corresponding inputs:

$$\begin{aligned}\hat{f}(v - y^*) &= \frac{1}{z} \mathbf{W} \phi(\mathbf{x}) \\ &= \frac{1}{z} \sum_{i=1}^m \overbrace{\alpha_i \langle \phi(\mathbf{x}_i), \phi(\mathbf{x}) \rangle}^{\beta_i} f(v - y_i) \\ &= \frac{1}{z} \sum_{i=1}^m \beta_i f(v - y_i),\end{aligned}$$

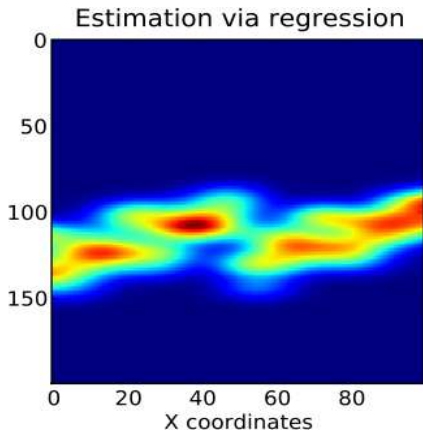
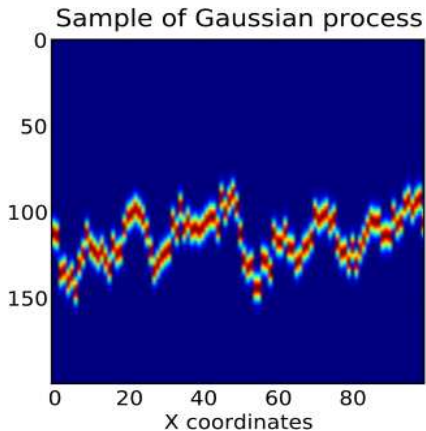
where $z = \sum_{i=1}^m \beta_i$

$\hat{f}(v - y^*)$ may not be of \mathcal{F}_ψ !

- From label to the output; if $E[f_\psi(v - y)] = y$ for all $y \in \mathcal{Y}$, thus the prediction of the outputs can be derived in a simple form:

$$\mathbf{y}^* = E[f(v - y^*)] = \boxed{\frac{1}{z} \sum_{i=1}^m \beta_i y_i}.$$

Example: Gaussian Process



A Gaussian process and its density estimation

Normalization

- **Preprocessing**

$$\begin{aligned}\psi(\mathbf{y}_i) &\Rightarrow \psi(\mathbf{y}_i)/\|\psi(\mathbf{y}_i)\|, \\ \phi(\mathbf{x}_i) &\Rightarrow \phi(\mathbf{x}_i)/\|\phi(\mathbf{x}_i)\|,\end{aligned}$$

- ▶ It can happen within the optimization. (no additional cost!)

- **Kernels with implicit normalization**, e.g. Gaussian,

$$\langle \mathbf{u}, \mathbf{v} \rangle = \exp(-d(\mathbf{u}, \mathbf{v})), \quad d() \geq 0.$$

- **Spherical embedding**

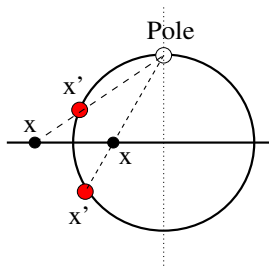
$$\left. \begin{aligned}\psi : \mathcal{Y} &\rightarrow \mathcal{S}_y \subset \mathcal{H}_\psi, \quad \mathcal{S}_y : \\ \phi : \mathcal{X} &\rightarrow \mathcal{S}_x \subset \mathcal{H}_\phi, \quad \mathcal{S}_x : \end{aligned} \right\} \text{Hyper-spheres}$$

Spherical embedding

- Spherical embedding

$$\left. \begin{array}{l} \psi : \mathcal{Y} \rightarrow \mathcal{S}_Y \subset \mathcal{H}_\psi, \mathcal{S}_Y : \\ \phi : \mathcal{X} \rightarrow \mathcal{S}_X \subset \mathcal{H}_\phi, \mathcal{S}_X : \end{array} \right\} \text{Hyper-spheres}$$

- Stereographic projection

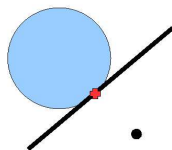


$$\begin{aligned} \Phi : \phi(x) &\rightarrow \phi'(x), \\ K'_{ij} &= \langle \phi'(x)_i, \phi'(x)_j \rangle \\ &= R^2 \left(1 - \frac{2R^2 \|\phi(x_i) - \phi(x_j)\|^2}{(\|\phi(x_i)\|^2 + R^2)(\|\phi(x_j)\|^2 + R^2)} \right) \\ &= R^2 \left(1 - \frac{2R^2 (K_{ii} + K_{jj} - 2K_{ij})}{(K_{ii} + R^2)(K_{jj} + R^2)} \right), \\ R &\text{ Ball radius.} \end{aligned}$$

Effect of the normalization

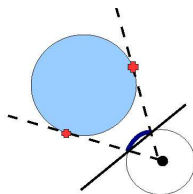
- **Effect of L2 normalization**
Wandering support vectors

$$\mathbf{x} \rightarrow \mathbf{x}$$



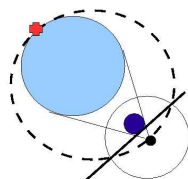
identity

$$\mathbf{x} \rightarrow \frac{\mathbf{x}}{\|\mathbf{x}\|_2}$$



projection onto ball

$$\mathbf{x} \rightarrow \frac{\mathbf{x}}{\|\mathbf{x}\|_2^2}$$



inversion

Solution

Quadratic Augmented Lagrangian Form

$$\begin{aligned} \min \quad & \frac{1}{2} \alpha^T [K_{\psi(y)} \bullet K_{\phi(x)}] \alpha - \mathbf{1}^T \alpha \\ & + \lambda^T K_{\psi(y)} \alpha + \frac{C_{ALP}}{2} \alpha^T K_{\psi(y)}^T K_{\psi(y)} \alpha \quad \Leftarrow \text{biased case} \\ \text{w.r.t.} \quad & \alpha \in \mathbb{R}^m, \text{ primal variables,} \\ & \lambda \in \mathbb{R}^m, \text{ Lagrangian variables,} \\ \text{s.t.} \quad & \mathbf{0} \leq \alpha \leq \mathbf{C}, \Leftarrow \text{Simple box constraint} \end{aligned}$$

- C_{ALP} Augmented Lagrangian Penalty Parameter
- component-wise (Schur) product

Solution schema

Outer loop

- Fix the Lagrangian variables,

Inner loop

- ▶ Solve the problem above the box constraint,
- ▶ The update formula

$$\alpha_i^{k+1} = P_{[0,C]}(-1 - \langle \alpha^k, K_i \rangle / K_{ii})$$
$$i = 1, \dots, m$$

- Update the Lagrangian,
- Increase the penalty constant

If there is no bias only the inner loop has to be processed!!!

Alternative solution approaches

- Extragradient based methods for variational inequalities
 - ▶ Korpelevich [3]
 - ▶ Nesterov [7], [6]
 - ▶ Nemirovski [5]
- Cutting plane methods(e.g. column generation, decomposition)
 - ▶ Joachims [2]
- Active set methods

Multiview learning

Additive case

We have $\{\psi(\mathbf{y})_i, (\phi^1(\mathbf{x}_i^1), \phi^2(\mathbf{x}_i^2), \dots)\}$ several sources of inputs taken out of distinct distributions.

$$\min \quad \frac{1}{2} \left[\sum_{k=1}^{n_k} \text{tr}(\mathbf{W}_k^T \mathbf{W}_k) \right] + \mathbf{C} \mathbf{1}^T \boldsymbol{\xi}$$

$$\text{w.r.t.} \quad \mathbf{W}_k : \mathcal{H}_{\phi^k} \rightarrow \mathcal{H}_{\psi}, \text{ linear op.}$$

$\mathbf{b} \in \mathcal{H}_{\psi}$, translation(bias)

$\boldsymbol{\xi} \in \mathbb{R}^m$, error vector

$$\text{s.t.} \quad \left\langle \psi(\mathbf{y}_i), \sum_{k=1}^{n_k} \mathbf{W}_k \phi^k(\mathbf{x}_i^k) + \mathbf{b} \right\rangle_{\mathcal{H}_{\psi}} \geq 1 - \xi_i$$

$$\boldsymbol{\xi} \geq \mathbf{0}, \quad i = 1, \dots, m$$

Kernel: $\mathbf{K}_y \bullet \sum_{k=1}^{n_k} \mathbf{K}_{x^k}$,
• element-wise product

Multiview learning

Product case

$$\min \quad \frac{1}{2} \text{tr}(\mathbf{W}^T \mathbf{W}) + C \mathbf{1}^T \boldsymbol{\xi}$$

$$\text{w.r.t.} \quad \mathbf{W} : \mathcal{H}_\phi^1 \otimes \mathcal{H}_\phi^2 \rightarrow \mathcal{H}_\psi, \text{ linear op.}$$

$$\mathbf{b} \in \mathcal{H}_\psi, \text{ translation(bias)}$$

$$\boldsymbol{\xi} \in \mathbb{R}^m, \text{ error vector}$$

$$\text{s.t.} \quad \left\langle \boldsymbol{\psi}(\mathbf{y}_i), \mathbf{W}(\phi^1(\mathbf{x}_i^1) \otimes \phi^2(\mathbf{x}_i^2)) + \mathbf{b} \right\rangle_{\mathcal{H}_\psi} \geq 1 - \xi_i$$

$$\boldsymbol{\xi} \geq \mathbf{0}, i = 1, \dots, m,$$

Kernel: $\mathbf{K}_y \bullet \mathbf{K}_{x^1} \bullet \mathbf{K}_{x^2}$,
• element-wise product

Epilogue

“Young man,
in mathematics you don’t understand things.
You just get used to them.”

John von Neumann, one of the greatest mathematician of the Twenty Century.

This is the End

Thanks!



K. Astikainen, J. Rousu, L. Holm, E. Pitkanen, and S. Szedmak.
Towards structured prediction of enzyme function.
In Machine Learning in Systems Biology (MLSB-2007), Evry, France. 2007.



T. Joachims.
Training linear svms in linear time.
In Proceedings of the ACM Conference on Knowledge Discovery and Data Mining (KDD), ACM. 2006.



G. Korpelevich.
The extragradient method for finding saddle points and other problems.
Ekonomika i Matematicheskie Metody, 12:747–756, 1976.
In Russian; English translation in Matekon.



O. L. Mangasarian.
Generalized support vector machines.
In Advances in Large Margin Classifiers, pages 135–146. MIT Press, 2000.



A. Nemirovski.

Prox-method with rate of convergence $o(1/t)$ for variational inequalities with lipschitz continuous monotone operators and smooth convex-concave saddle point problems.

SIAM Journal on Optimization, 15:229–251, 2004.



Y. Nesterov.

Dual extrapolation and its application for solving variational inequalities and related problems.

In *CORE Discussion Paper/68*, September 2003. 2003.



Y. Nesterov.

Smooth minimization of nonsmooth functions.

In *CORE Discussion Paper/12*, February 2003. 2003.



John V. Neumann.

Zur theorie der gesellschaftsspiele.

Mathematische Annalen, 100:295–320, 1928.

English Translation Fin, Tucker, A.W. and R.D. Luce, ed.,
Contributions to the Theory of Games IV, Annals of Mathematics
Studies 40, 1959.



E. Pekalska and R.P.W. Duin.

*The Dissimilarity Representation for Pattern Recognition.
Foundations and Applications.*

World Scientific, Singapore, 2005.



R. Schapire.

The boosting approach to machine learning: an overview.

In MRSI Workshop on Nonlinear Estimation and Control. 2002.