Support Vector Machine Learning for Interdependent and Structured Output Spaces

Ioannis Tsochantaridis Thomas Hofmann Thorsten Joachims Yasemin Altun

presenter: Meng Tian

Problem:

Mapping from input $\mathbf{x} \in \mathcal{X}$ to $\mathbf{y} \in \mathcal{Y}$:

Recall:

multiclass classification –

arbitrary label set or identifiers $\mathcal{Y} = \{1, ..., K\}$

regression –

$\mathcal{Y} = \mathbb{R}$ and the response variable is a scalar...

Structural Problem:

Input-output pairs $(\mathbf{x}_1, \mathbf{y}_1), \ldots, (\mathbf{x}_n, \mathbf{y}_n) \in \mathcal{X} \times \mathcal{Y}$

structured output spaces $\mathcal{Y}$ e.g. sequences, strings, labeled trees...

How to improve:

Generalizing large margin methods, specifically multi-class SVMs to the broader problem of learning structured

Specifying discriminant functions that exploit the structure and dependencies within $\mathcal{Y}$

Improving loss function that may be appropriate for specific applications

Not primarily to define more complex functions, but to deal with more complex output spaces by extracting combined features over inputs and outputs

Discriminant function:

$$F : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$$

the general form of hypotheses *f*

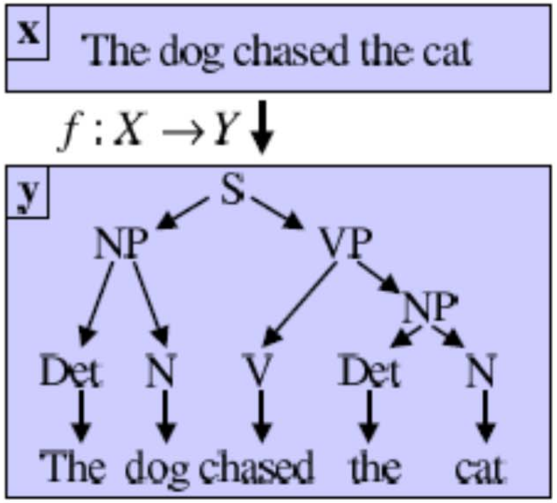
$$f(\mathbf{x};\mathbf{w}) = \underset{\mathbf{y}\in\mathcal{Y}}{\operatorname{argmax}}\, F(\mathbf{x},\mathbf{y};\mathbf{w})$$

(***w*** *– a parameter vector , F is a w-parameterized family cost function*)

combined feature function : $\Psi(\mathbf{x},\mathbf{y})$

$$F(\mathbf{x},\mathbf{y};\mathbf{w}) = \langle \mathbf{w}, \Psi(\mathbf{x},\mathbf{y}) \rangle$$

Example: natural language parsing model

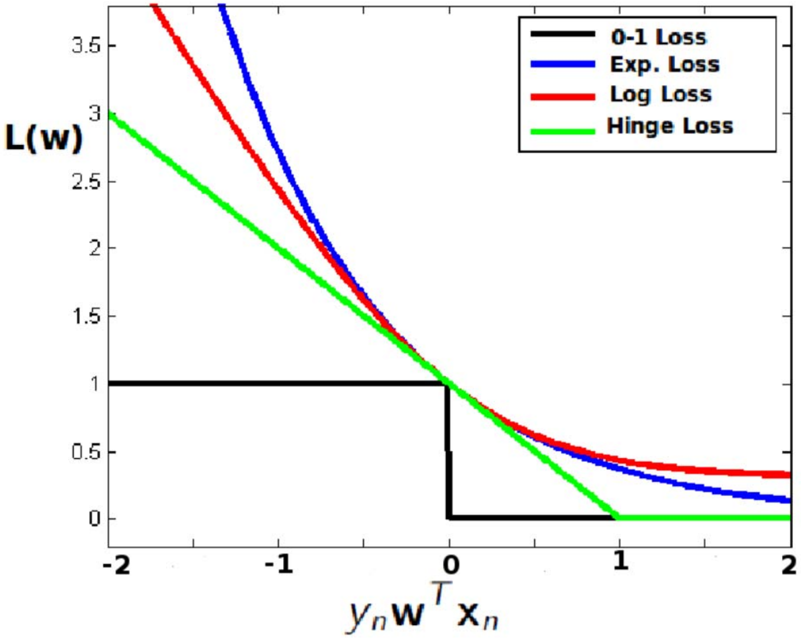


$$\Psi(\mathbf{x},\mathbf{y}) = \left(\begin{array}{c} 1 \\ 0 \\ 2 \\ 1 \\ \vdots \\ 0 \\ 2 \\ 1 \\ 1 \\ 1 \end{array}\right) \begin{array}{c} S \to NP\,VP \\ S \to NP \\ NP \to Det\ N \\ VP \to V\ NP \\ \\ Det \to dog \\ Det \to the \\ N \to dog \\ V \to chased \\ N \to cat \end{array}$$

$\Psi(\mathbf{x},\mathbf{y})$ denotes a histogram vector of counts(how often each grammar rule occurs in the tree y) , $f(\mathbf{x};\mathbf{w})$ computed by finding the structure $\mathbf{y} \in \mathcal{Y}$ that maximizes $F(\mathbf{x},\mathbf{y};\mathbf{w})$ via the CKY algorithm

Loss function:

loss function



Standard zero-one loss function is not appropriate for most kinds of structured responses example: natural language parsing

Loss function:

In order to quantify the accuracy of a prediction, consider learning with arbitrary loss function $\Delta : \mathcal{Y} \times \mathcal{Y} \rightarrow \mathbb{R}$

If the true output is y, prediction is $\hat{\mathbf{y}}$:

$$\triangle(\mathbf{y}, \hat{\mathbf{y}}) \quad \longrightarrow \quad \triangle(\mathbf{y}, f(\mathbf{x}))$$

Empirical Risk minimization:

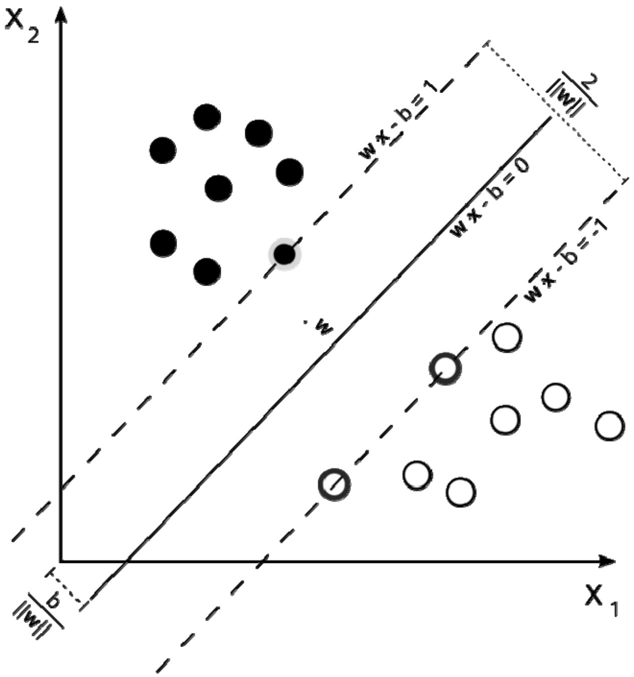
$$\mathcal{R}_P^{\triangle}(f) = \int_{\mathcal{X} \times \mathcal{Y}} \triangle(\mathbf{y}, f(\mathbf{x}))\, dP(\mathbf{x}, \mathbf{y})$$

find a function f in a given hypothesis class such that the risk is minimized

Margins and Margin Maximization

Recall: SVM finds the maximum margin hyperplane

Maximizing the margin(distance) = minimizing $||\mathbf{w}||$ (the norm)



x_2
$\mathbf{w}\cdot\mathbf{x} - b = 1$
$\frac{2}{||\mathbf{w}||}$
$\mathbf{w}\cdot\mathbf{x} - b = 0$
$\mathbf{w}\cdot\mathbf{x} - b = -1$
$\mathbf{w}$
$\frac{b}{||\mathbf{w}||}$
x_1

Margins and Margin Maximization

Hard-margin optimization problem:

$$\mathrm{SVM}_0 : \min_{\mathbf{w}} \quad \frac{1}{2}\|\mathbf{w}\|^2$$

$$\forall i,\ \forall \mathbf{y} \in \mathcal{Y} \setminus \mathbf{y}_i : \ \langle \mathbf{w}, \boxed{\delta\Psi_i(\mathbf{y})} \rangle \geq 1$$

$$\Downarrow$$

$$\Psi(\mathbf{x}_i, \mathbf{y}_i) - \Psi(\mathbf{x}_i, \mathbf{y})$$

Margins and Margin Maximization

Soft-margin:

Allow some training examples to fall within the margin

Add: slack variables

$$\text{SVM}_1 : \min_{\mathbf{w}, \boldsymbol{\xi}} \ \frac{1}{2}\|\mathbf{w}\|^2 + \boxed{\frac{C}{n}} \boxed{\sum_{i=1}^{n} \xi_i}, \ \text{s.t.} \ \forall i, \xi_i \geq 0$$
$$\forall i, \forall \mathbf{y} \in \mathcal{Y} \setminus \mathbf{y}_i : \ \langle \mathbf{w}, \delta\Psi_i(\mathbf{y}) \rangle \geq 1 - \xi_i \, .$$

But we want the number of misclassified examples to be minimized by minimizing the sum of slack variables

Problem: if output is very large?

generalize the formulation to the case of arbitrary loss function

General Loss Functions:

Re-scale the slack variables:

$$\mathrm{SVM}_1^{\triangle s} : \min_{\mathbf{w},\boldsymbol{\xi}} \ \frac{1}{2}\|\mathbf{w}\|^2 + \frac{C}{n}\sum_{i=1}^{n} \xi_i, \ \text{ s.t. } \forall i, \xi_i \geq 0$$

$$\forall i, \forall \mathbf{y} \in \mathcal{Y} \setminus \mathbf{y}_i : \langle \mathbf{w}, \delta\Psi_i(\mathbf{y})\rangle \geq 1 - \boxed{\frac{\xi_i}{\triangle(\mathbf{y}_i,\mathbf{y})}}$$

High loss $\triangle(\mathbf{y}_i, \mathbf{y})$ should be penalized more severely

$\frac{1}{n}\sum_{i=1}^{n} \xi_i^*$ is the upper bound on the empirical risk

$\triangle(\mathbf{y}_i, \mathbf{y})$ can be replaced by $\sqrt{\triangle(\mathbf{y}_i, \mathbf{y})}$

General Loss Functions:

Re-scale the margin :

$$\forall i,\ \forall \mathbf{y} \in \mathcal{Y} \setminus \mathbf{y}_i :\ \langle \mathbf{w}, \delta\Psi_i(\mathbf{y}) \rangle \geq \triangle(\mathbf{y}_i, \mathbf{y}) - \xi_i$$

special case of the Hamming loss -> general loss function

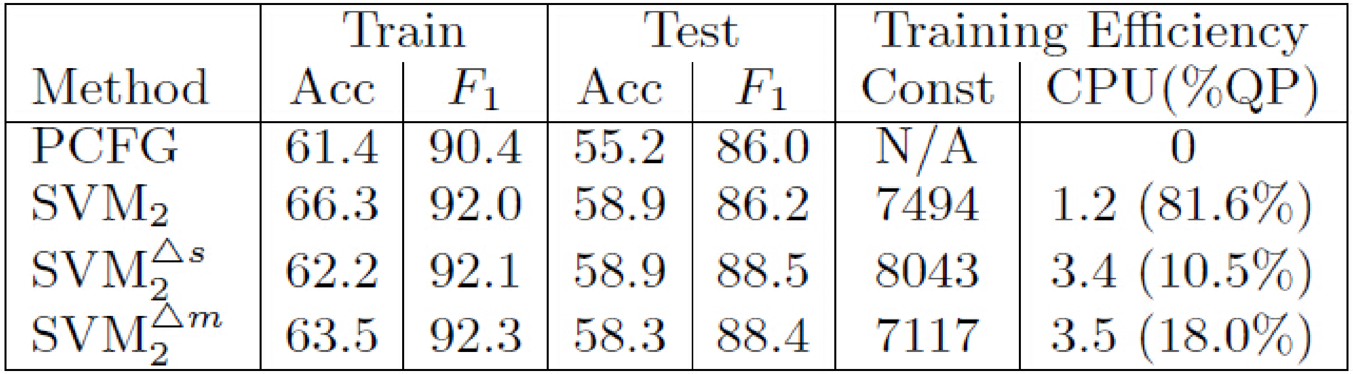
Also results in an upper bound on the empirical risk

Potential disadvantage: give significant weight to output values y that are not even close to the target value y_i ,because every increase in the loss increases the required margin.

Experiment

Natural Language Parsing

4098 sentences for training & 163 sentences for testing



Method	Train		Test		Training Efficiency	
	Acc	F_1	Acc	F_1	Const	CPU(%QP)
PCFG	61.4	90.4	55.2	86.0	N/A	0
SVM_2	66.3	92.0	58.9	86.2	7494	1.2 (81.6%)
$\text{SVM}_2^{\triangle s}$	62.2	92.1	58.9	88.5	8043	3.4 (10.5%)
$\text{SVM}_2^{\triangle m}$	63.5	92.3	58.3	88.4	7117	3.5 (18.0%)

$$F_1 \text{ loss } \triangle(\mathbf{y}_i, \mathbf{y}) = (1 - F_1(\mathbf{y}_i, \mathbf{y}))$$

Comments & Questions

1.Support vector method for supervised learning with structured and interdependent outputs based on a joint feature map over input-output pairs

2.Flexible in its ability to handle loss function

3.Paper year:2004, big contribution for Structured Support Vector Machine, influence for later papers(2007,2008)

1.How to choose the loss function will impact the result deeply.

E.g. hamming distance

2.Which dominates the constraints

Thank you