Wrocław University of Technology

Boosting Algorithm with Sequence-loss Cost Function for Structured Prediction

Tomasz Kajdanowicz, Przemysław Kazienko, Jan Kraszewski
Wroclaw University of Technology, Poland

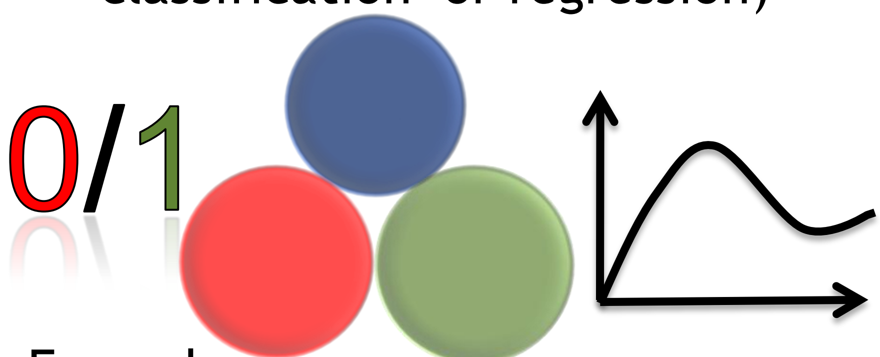
Outline

1. Introduction to Structured Prediction
2. Problem Description
3. The concept of AdaBoost^{Seq}
4. Experiments

Structured prediction

Single value prediction

- function *f* maps an input to an simple output (binary classification , multiclass classification or regression)

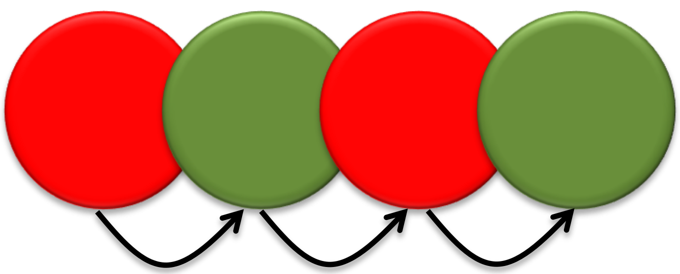


Example :

problem of predicting whether the next day will or will not be rainy on the basis of historical weather data.

Structured prediction

- prediction problems with more complex outputs (structured prediction)

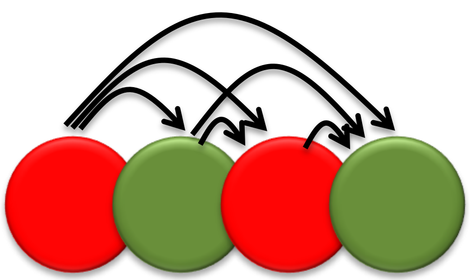


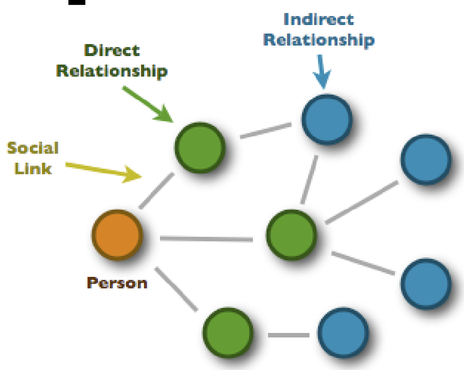
Example :

problem of predicting weather for next few days.

Structured prediction

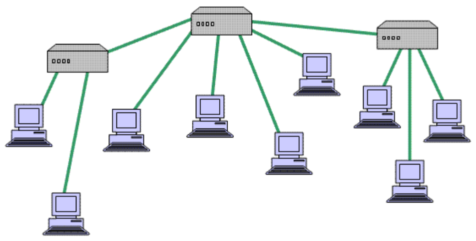
- **<u>Structured prediction</u>** is a cost-sensitive prediction problem, where output has structure of elements decomposing into variable-length vectors. [Daume]





Vector notation is treated as useful encoding not only for sequence labeling problems.

0	1	0	1	1	1



Input = original input + partially produced output (extended notion for feature input space)

Structured prediction algorithms

- Most algorithms are based on the well know binary classification adapted in the specific way [Nguyen et al.]
- Structured perceptron [Collins]
  - minimal requirements on output space shape
  - easy to implement
  - poor generalization
- Max-margin Markov Nets [Taskar et al.]
  - very useful
  - perform very slow
  - limited to Hamming loss function

Structured prediction algorithms

- Conditional Random Fields [Lafferty et al.]
  - extention of logistic regression to the structured outputs
  - probabilistic outputs
  - good generalization
  - relatively slow
- Support Vector Machine for Interdependent and Structured Outputs (SVM^{STRUCT}) [Tsochantaridis et al.]
  - more loss functions

Ensembles

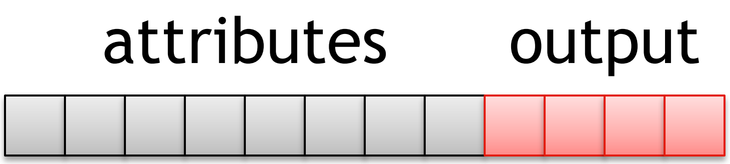
- Combined may be better
  - the goal is to select the right component for building a good hybrid system
  - Lotfi Zadeh is reputed to have said:

Good combined system is like	Bad combined system is like
British Police	British Cuisine
German Mechanics	German Police
French Cuisine	French Mechanics
Swiss Banking	Italian Banking
Italian Love	Swiss Love

Problem Description

prediction of sequential values

- for single case a sequence of output values



Problem Statement

- Binary sequence classification problem

$$f : X \rightarrow Y$$

where:

X – vector input,

Y - variable-length vector $(y_1, y_2, \ldots, y_T)$

$$y^{\mu}_{i} \in \{-1, 1\}$$

- where

$i=1,2,\ldots,N$ – number of observations

$\mu=1,2,\ldots,T$ – length of sequence

Problem Statement

- **Goal:** *T* classifiers combined:
  - optimally designed linear combination
  - *K* base classifiers of the form

$$F^{\mu}(x) = \sum_{k=1}^{K} \alpha_k \Phi(x; \Theta_k)$$

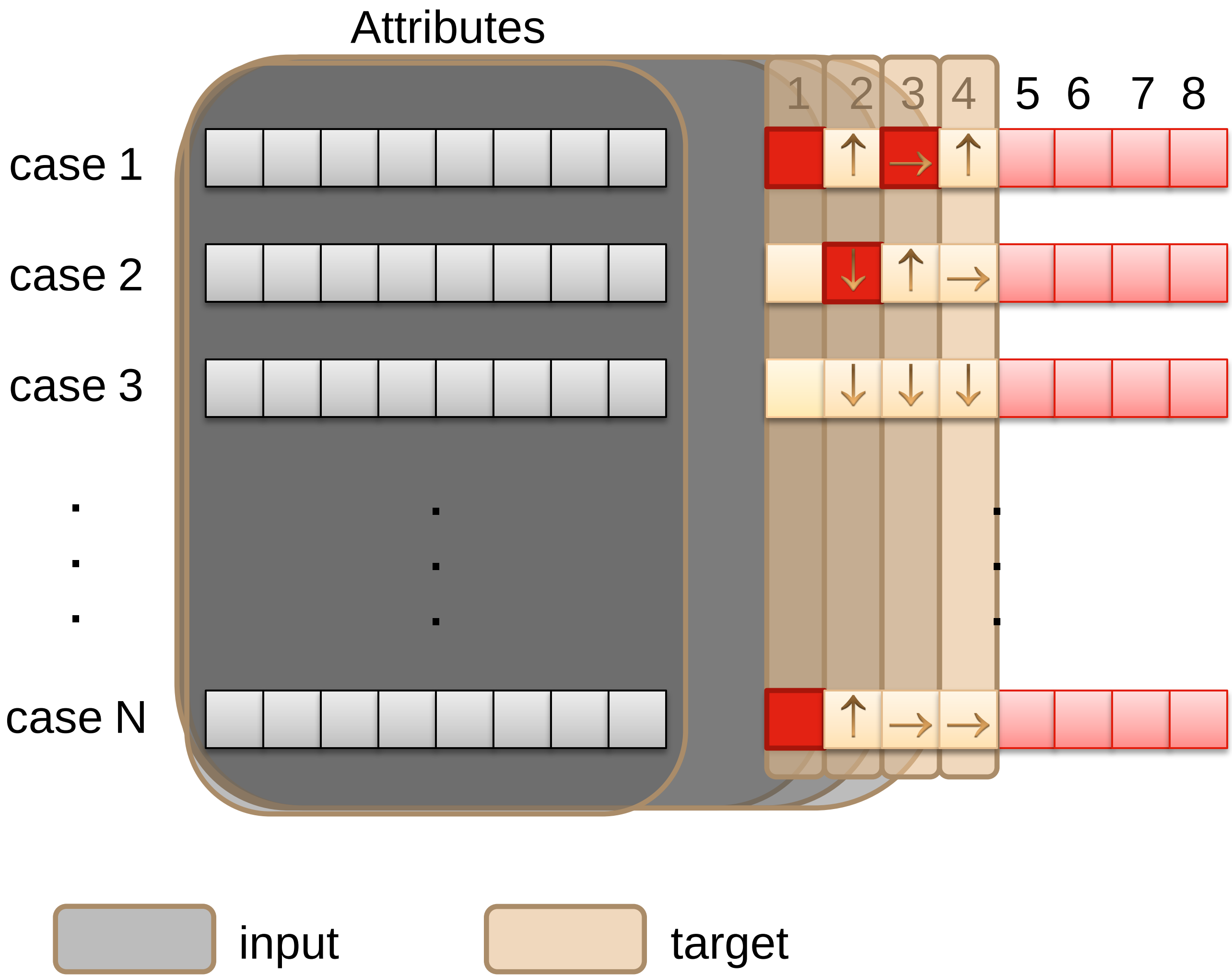
where

$\Phi(x, \Theta_k)$ - *k*th base classifier

Θ_k - parameters of *k*th classifier

α_k - weight associated to the *k*th classifier

General Idea of AdaBoost^{Seq}



And so on..

AdaBoost^{Seq}

- A novel algorithm for sequence prediction
- Optimization for each sequence item:

$$\arg \min_{\alpha_k;\Theta_k;k:1,K} \sum_{i=1}^{N} \exp\left(-y_i F^{\mu}(x_i)\right)$$

- Equation is highly complex => a stage-wise suboptimal method is performed

AdaBoost^{Seq}

- By definition of the *m*th partial sum:

$$F_m^{\mu}(x) = \sum_{k=1}^{m} \alpha_k \Phi(x;\Theta_k), m = 1,2,\ldots,K$$

- The recurence is obvious:

$$F_m^{\mu}(x) = F_{m-1}^{\mu}(x) + \alpha_m \Phi(x;\Theta_m)$$

- Stagewise optimization
  - *m*th step, $F_{m-1}(x)$ is part of the previous step
  - the new target is: $(\alpha_m, \Theta_m) = \arg\min_{\alpha,\Theta} J(\alpha,\Theta)$

AdaBoost^{Seq}

$$J(\alpha,\Theta)=\sum_{i=1}^{N}\exp\left(-y_i\left(\xi F_{m-1}^{\mu}(x_i)+(1-\xi)y_i\widehat{R}_m^{\mu}(x_i)+\alpha\Phi(x_i;\Theta)\right)\right)$$

where

$\widehat{R}_m^{\mu}$ - impact function denoting the influence of the quality of preceding sequence labels prediction

$$\widehat{R}_m^{\mu}(x_i)=\sum_{i=1}^{m-1}\alpha_i R^{\mu}(x)$$

$$R^{\mu}(x)=\frac{\sum_{i=1}^{\mu-1} y\frac{F_i(x)}{\sum_{j=1}^{K}\alpha_j}}{\mu-1}$$

AdaBoost^{Seq}

- For given α :

$$\Theta = \arg\min_{\Theta} \sum_{i=1}^{N} w_i^{(m)} \exp\left(-y_i \alpha \Phi\left(x_i ; \Theta\right)\right)$$

$$w_i^{(m)} \equiv \exp\left(-y_i \left(\xi F_{m-1}(x_i) + (1-\xi) y_i \hat{R}^{\mu}(x)\right)\right)$$

- Because $w_i^{(m)}$ does not depend neighter on α nor $\Phi(x_i;\Theta)$, it can be threated as a weigth of x_i
- Binary nature of base classifier:

$$\Theta_m = \arg\min_{\Theta} \left\{ P_m = \sum_{i=1}^{N} w_i^{(m)} I\left(1 - y_i \Phi\left(x_i ; \Theta\right)\right) \right\}$$

P_m – weighted empirical error

$$I(x) = \begin{cases} 0, if \ x = 0 \\ 1, if \ x > 0 \end{cases}$$

AdaBoost^{Seq}

- Computing base classifier at step *m*:

$$\sum_{y_i \Phi(x_i;\Theta_m)<0}^{N} w_i^{(m)} = P_m$$

$$\sum_{y_i \Phi(x_i;\Theta_m)>0}^{N} w_i^{(m)} = 1 - P_m$$

AdaBoost^{Seq}

- Getting equations together:

$$\alpha_m = \arg\min_{P_m} \left\{ \exp(-\alpha)(1 - P_m) + \exp(\alpha) P_m \right\}$$

- derivative:

$$\alpha_m = \frac{1}{2} \ln \frac{1 - P_m}{P_m}$$

AdaBoost^{Seq}

- Weight of the *i*th case:

$$w_i^{(m+1)} = \frac{w_i^{(m)} \exp\left(-y_i \xi \alpha_m \Phi\left(x_i;\Theta_m\right) - \left(1-\xi\right)\alpha_m R^{\mu}(x)\right)}{Z_m}$$

- Z_m - normalizator:

$$Z_m = \sum_{i=1}^{N} w_i^{(m)} \exp\left(-y_i \xi \alpha_m \Phi\left(x_i;\Theta_m\right) - \left(1-\xi\right)\alpha_m R^{\mu}(x)\right)$$

Algorithm AdaBoost^{Seq}

- For each sequence position (μ=1 to T)
  - Initialization: $w_i^{(1)}=1/N,\ i=1,2,\dots,N;\ m=1$
  - While termination criterion is not met:
    - obtain optimal Θ_m and $\Phi(\cdot;\Theta_m)$ (min. P_m)
    - obtain optimal Pm
    - $a_m=1/2\ln((1-P_m)/P_m)$
    - $Z_m = 0.0$
    - For $i = 1$ do N
      - $w_i^{(m+1)}= w_i^{(m)}exp(-y_i\ \xi\ a_m\Phi(x_i;\ \Theta_m)-(1-\ \xi)\ a_m\ R^{\mu}(x))$
      - $Z_m=Z_m+w_i^{(m+1)}$
    - End For
    - For $i = 1$ do N
      - $w_i^{(m+1)}= w_i^{(m)}/Z_m$
    - End For
    - $K = m;\ m = m+1$
  - End while
  - $f^{\mu}(\cdot)=\text{sign}(\Sigma^{K}_{k=1}a_k\Phi(\cdot;\Theta_k))$
- End for

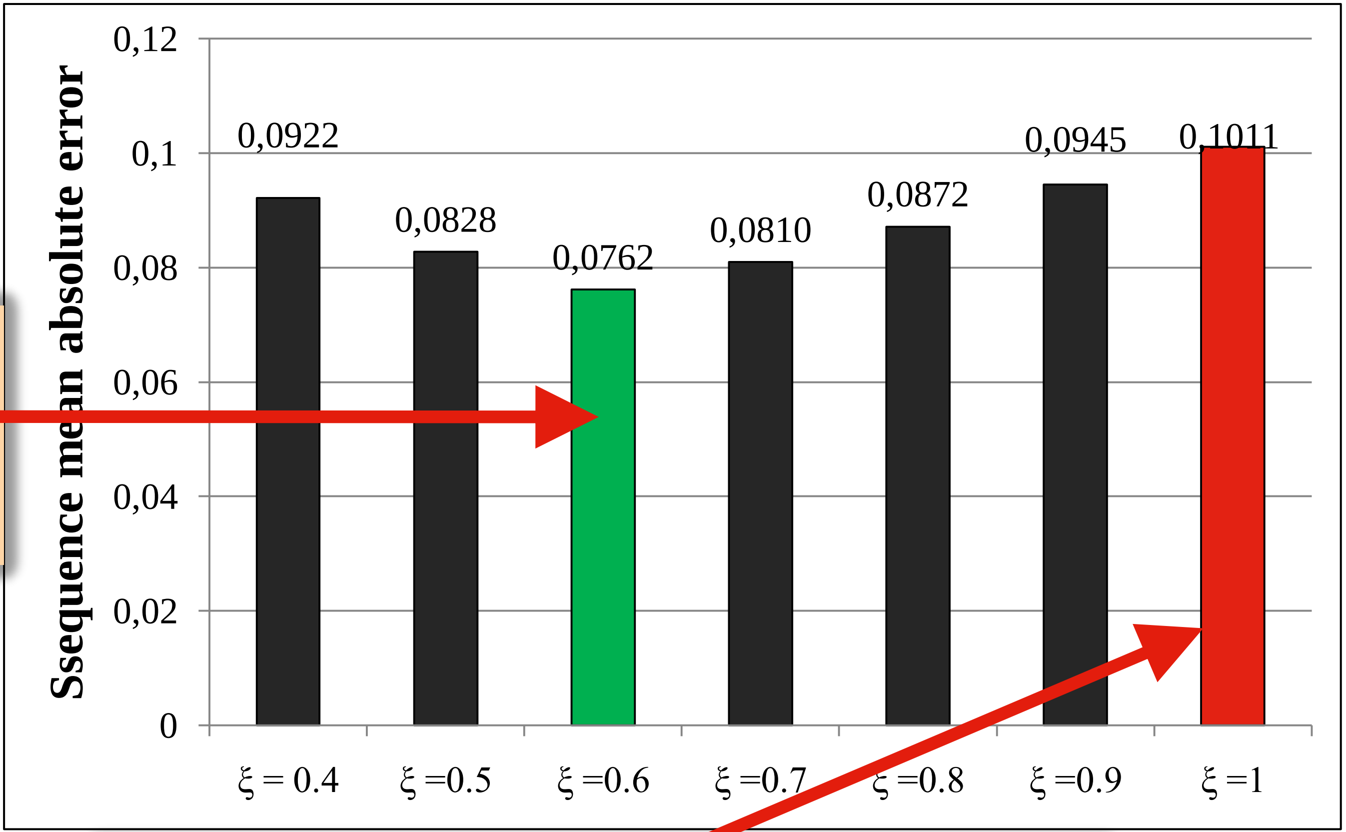
Profile of AdaBoost^{Seq}

- A **new** algorithm for sequence prediction
- For each sequence item
  - AdaBoost^{Seq} considers also prediction errors for all previous items in the sequence **within the boosting algorithm**
  - the more errors on previous sequence items, the stronger focus on bad cases at the recent item
- **Self-adaptive**

Experiments

- 4019 cases in the dataset
- 20 input features
- Sequence lenght=10
- Decision stump as the base classifier
- 10 fold cross-validation

AdaBoost vs. AdaBoost^{Seq} (with ξ) Mean Absolute Error

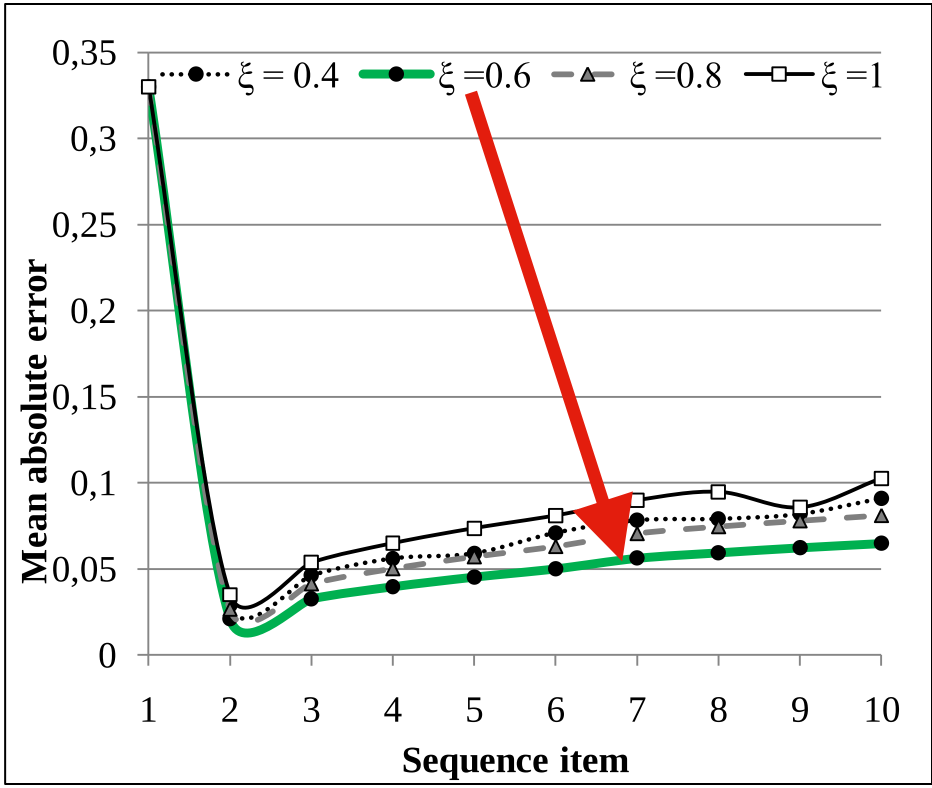


ξ=0.6 the best

For ξ=1 it is a standard AdaBoost (the worst)

Summary of the Experiments

- For item 2+ **error reduced dramatically (6 times!)** since it respects errors on previous items
- ξ influences error
- ξ=0.6 **error decreases by 24%** for the whole sequence compared to the standard approach (ξ=1)



Conclusions and Future Work

- AdaBoost^{Seq} - **a new algorithm for sequence prediction** based on AdaBoost
- While prediction of the following items in sequence, **the errors from the previous items** are utilized
- Much **more accurate than AdaBoost** applied to sequence items independently
- Parametrized, ξ - **how much errors are respected**
- Recent application: prediction for debt valuation
- Future work: new cost functions (on HMM canva)

Wrocław University of Technology

Thank you for attention
Q & A?

Wroclaw, HAIS 2011

Wrocław University of Technology