Online Passive-Aggressive Algorithms

Jean-Baptiste Behuet
Tutor: Eneldo Loza Mencía

28/11/2007

Seminar aus Maschinellem Lernen

WS07/08

Overview

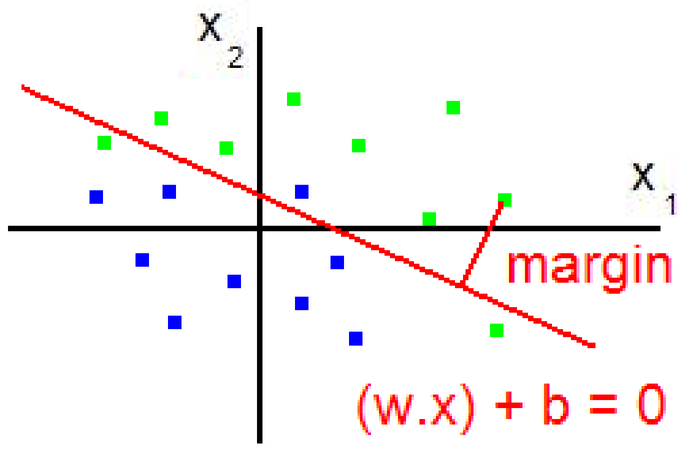
- Online algorithms
- Online Binary Classification Problem
  - Perceptron Algorithm
  - 3 versions of the Passive-Aggressive Algorithm
  - Loss bounds, Comparison with the Perceptron
- Other learning problems
- Experiments
- Conclusion

Online Algorithms

- Sequence of rounds t:
  - Instance x_t as input
  - Predicts $\hat{y}_t$ as output
  - Receives correct output y_t
  - Updates prediction mechanism

Online Binary Classification: Perceptron Algorithm

- Round t:
  - instance $x_t \in \mathbb{R}^n$ with label $y_t \in \{-1,1\}$
  - classification fonction based on weight vector $w_t \in \mathbb{R}^n \rightarrow$ defines hyperplane separating the 2 classes
  - prediction: $\hat{y}_t = sign((w_t . x_t) + b)$
  - signed margin: $y_t((w_t . x_t) + b)$
  - correct if $margin > 0$



- Goal: incrementally learn w_t (incrementally modify hyperplane)

Online Binary Classification: Perceptron Algorithm (2)

- Start with random hyperplane (random w_0)
- At each round t of the algorithm:
  - receives x_t and predicts $\hat{y}_t = sign(w_t.x_t + b_t)$
  - receives correct y_t and updates the hyperplane
- Update minimizes the distance of misclassified exemples to the boundary

  $w_{t+1} = w_t + \rho(y_t - \hat{y}_t)x_t$ with $\rho > 0$ learning rate

  (the hyperplane is updated when an error occurs)

Passive-Aggressive Algorithm for binary classification

- Want: $margin \geq 1$ as often as possible (not only correctly classified exemples)
- Hinge-loss fonction:

$$\ell(\mathbf{w};(\mathbf{x},y)) = \begin{cases} 0 & y(\mathbf{w}\cdot\mathbf{x}) \geq 1 \\ 1-y(\mathbf{w}\cdot\mathbf{x}) & \text{otherwise} \end{cases}$$

- Loss suffered at round t: $\ell_t = \ell(\mathbf{w}_t;(\mathbf{x}_t,y_t))$
- Number of prediction mistakes $\leq \sum l_t^2$

Passive-Aggressive Algorithm for binary classification (2)

- Initialization: $w_1 = (0, \ldots, 0)$
- Update:
  - w_{t+1} solution of constrained optimization problem:

$$\mathbf{w}_{t+1} = \operatorname*{argmin}_{\mathbf{w} \in \mathbb{R}^n} \frac{1}{2} \|\mathbf{w} - \mathbf{w}_t\|^2 \quad \text{s.t.} \quad \ell(\mathbf{w}; (\mathbf{x}_t, y_t)) = 0$$

  - w_{t+1} has the form: $\mathbf{w}_{t+1} = \mathbf{w}_t + \tau_t y_t \mathbf{x}_t$ where $\tau_t = \dfrac{\ell_t}{\|\mathbf{x}_t\|^2}$

Passive-Aggressive Algorithm for binary classification (3)

- Trade-off:
  - w_{t+1} required to have no loss on current exemple
  - w_{t+1} as close as possible to w_t
- " Passive-Aggressive " :
  - " passive " when $l_t = 0$
  - " aggressive " otherwise: w_{t+1} forced to satisfy the constraint $l_t = 0$ on the current exemple

Two variations of the PA algorithm

- Problem of aggressiveness in case of noise
- PA-I: $$\mathbf{w}_{t+1} = \underset{\mathbf{w}\in\mathbb{R}^n}{\operatorname{argmin}} \frac{1}{2}\|\mathbf{w}-\mathbf{w}_t\|^2 + C\xi \quad \text{s.t.} \quad \ell(\mathbf{w};(\mathbf{x}_t,y_t)) \le \xi \text{ and } \xi \ge 0$$
- PA-II: $$\mathbf{w}_{t+1} = \underset{\mathbf{w}\in\mathbb{R}^n}{\operatorname{argmin}} \frac{1}{2}\|\mathbf{w}-\mathbf{w}_t\|^2 + C\xi^2 \quad \text{s.t.} \quad \ell(\mathbf{w};(\mathbf{x}_t,y_t)) \le \xi$$

  with C aggressiveness parameter
- same update form as for PA: $\mathbf{w}_{t+1} = \mathbf{w}_t + \tau_t y_t \mathbf{x}_t$

$\tau_t = \min\left\{ C\ ,\ \frac{\ell_t}{\|\mathbf{x}_t\|^2} \right\}$ for PA-I $\quad \tau_t = \frac{\ell_t}{\|\mathbf{x}_t\|^2 + \frac{1}{2C}}$ for PA-II

Relative loss bounds

- Number of prediction mistakes $\leq \sum l_t^2$
- Comparison of the loss attained by PA with the loss attained by a fixed classifier *sign(u.x)*

$$\ell_t = \ell\left(\mathbf{w}_t; (\mathbf{x}_t, y_t)\right) \qquad \ell_t^\star = \ell\left(\mathbf{u}; (\mathbf{x}_t, y_t)\right)$$

- For the original PA algorithm:

$$\sum_{t=1}^{T} \ell_t^2 \leq \|\mathbf{u}\|^2 R^2 \qquad \forall t \quad \|x_t\| \leqslant R \quad and \quad l_t^* = 0$$

Relative loss bounds (2)

- For the original PA algorithm:

$$\sum_{t=1}^{T} \ell_t^2 \;\leq\; \left(\|\mathbf{u}\| + 2\sqrt{\textstyle\sum_{t=1}^{T} (\ell_t^\star)^2} \right)^2 \qquad \forall t \;\; \|x_t\| = 1 \;\;,\;\; \forall u \in \mathbb{R}^n$$

- For PA-I:

$$M \leq \max\{R^2, 1/C\} \left(\|\mathbf{u}\|^2 + 2C \sum_{t=1}^{T} \ell_t^\star \right) \qquad \forall t \;\; \|x_t\| \leqslant R \;\;,\;\; \forall u \in \mathbb{R}^n$$

- For PA-II:

$$\sum_{t=1}^{T} \ell_t^2 \leq \left(R^2 + \frac{1}{2C} \right) \left(\|\mathbf{u}\|^2 + 2C \sum_{t=1}^{T} (\ell_t^\star)^2 \right) \qquad \forall t \;\; \|x_t^2\| \leqslant R^2 \;\;,\;\; \forall u \in \mathbb{R}^n$$

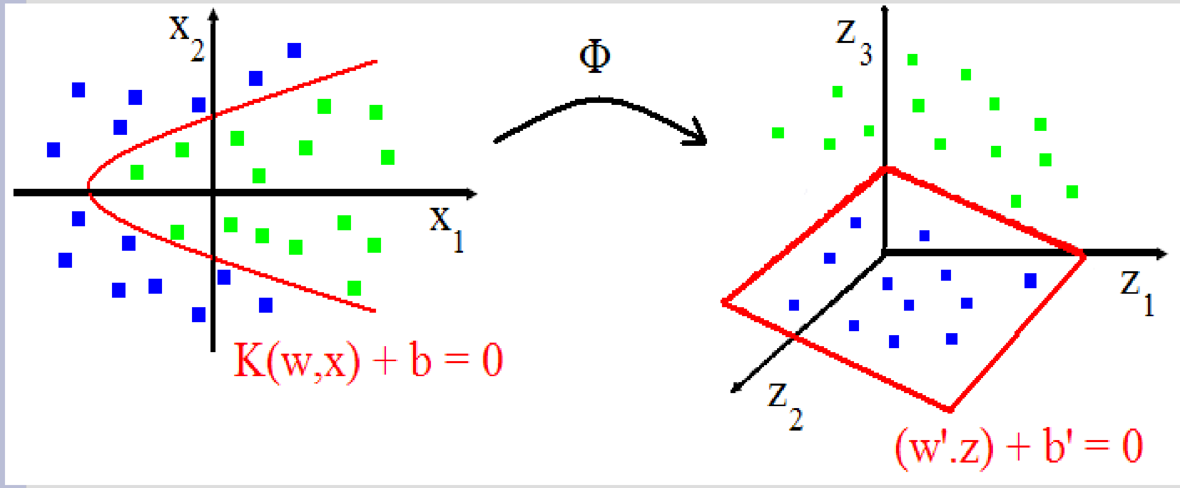
Comparison with the Perceptron Algorithm

Passive-Aggressive Algorithms	**Perceptron Algorithm**
INPUT: aggressiveness parameter $C > 0$	INPUT: learning rate $\rho > 0$
INITIALIZE: $\mathbf{w}_1 = (0,\ldots,0)$	INITIALIZE: $\mathbf{w}_1$ random
For $t = 1,2,\ldots$	For $t = 1,2,\ldots$
• receive instance: $\mathbf{x}_t \in \mathbb{R}^n$	• receive instance: $\mathbf{x}_t \in \mathbb{R}^n$
• predict: $\hat{y}_t = \text{sign}(\mathbf{w}_t \cdot \mathbf{x}_t)$	• predict: $\hat{y}_t = \text{sign}(\mathbf{w}_t \cdot \mathbf{x}_t)$
• receive correct label: $y_t \in \{-1,+1\}$	• receive correct label: $y_t \in \{-1,+1\}$
• suffer loss: $\ell_t = \max\{0\ ,\ 1 - y_t(\mathbf{w}_t \cdot \mathbf{x}_t)\}$	• update:
• update:	$\mathbf{w}_{t+1} = \mathbf{w}_t + \rho\,(y_t - \hat{y}_t)\,\mathbf{x}_t$
1. set: $\tau_t = \frac{\ell_t}{\|\mathbf{x}_t\|^2}$ (PA); $\tau_t = \min\left\{C\ ,\ \frac{\ell_t}{\|\mathbf{x}_t\|^2}\right\}$ (PA-I); $\tau_t = \frac{\ell_t}{\|\mathbf{x}_t\|^2 + \frac{1}{2C}}$ (PA-II)	
2. update: $\mathbf{w}_{t+1} = \mathbf{w}_t + \tau_t y_t \mathbf{x}_t$	

- Bounds are comparable both in separable (PA) and non-separable (PA-I, PA-II) cases

Generalization to the non-linear case: Principle

- map the data space into a feature space where the data is now linearly separable



- feature map $\Phi: \chi \rightarrow H$
- replace $(w.x)$ by Mercer Kernel $K(w,x)$ (non-linear function)

- $K(w,x)$ is the inner product of the vectors $\Phi(w)$ and $\Phi(x)$
- algorithm learns w'_t (weight vector in feature space H) and predicts $\hat{y}_t = sign(w'^T_t . \Phi(x_t))$

Other problems

- Regression
- Uniclass prediction
- Multiclass problems

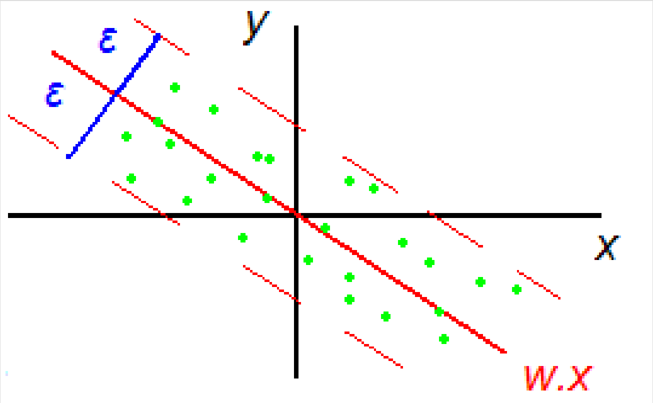
Regression

- main difference with the binary problem:

  $y \notin \{-1, 1\}\ ,\ y \in \mathbb{R}$

- instance $x_t \in \mathbb{R}^n$

  $\rightarrow$ prediction $\hat{y}_t = (w_t . x_t)$



- ε-sensitive hinge loss function:

$$\ell_\varepsilon(\mathbf{w}; (\mathbf{x}, y)) = \begin{cases} 0 & |\mathbf{w} \cdot \mathbf{x} - y| \le \varepsilon \\ |\mathbf{w} \cdot \mathbf{x} - y| - \varepsilon & \text{otherwise} \end{cases}$$

Regression: PA algorithms

- Initialization: $w_1 = (0, \ldots, 0)$
- Update:
  - w_{t+1} solution of constrained optimization problem:

$$\mathbf{w}_{t+1} = \underset{\mathbf{w} \in \mathbb{R}^n}{\operatorname{argmin}} \frac{1}{2} \|\mathbf{w} - \mathbf{w}_t\|^2 \quad \text{s.t.} \quad \ell_\varepsilon\big(\mathbf{w}; (\mathbf{x}_t, y_t)\big) = 0$$

  - w_{t+1} has the form: $\mathbf{w}_{t+1} = \mathbf{w}_t + \operatorname{sign}(y_t - \hat{y}_t)\tau_t \mathbf{x}_t$

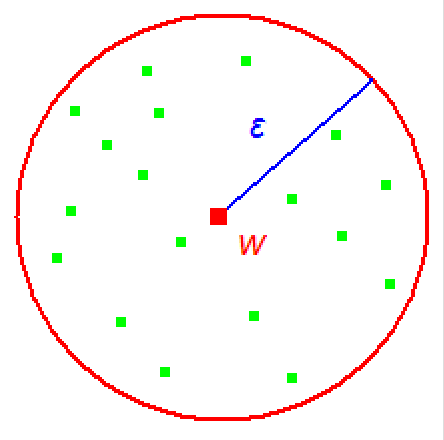
$$\tau_t = \ell_t / \|\mathbf{x}_t\|^2 \quad \textit{(PA)} \qquad \tau_t = \min\left\{ C, \frac{\ell_t}{\|\mathbf{x}_t\|^2} \right\} \quad \textit{(PA-I)} \qquad \tau_t = \frac{\ell_t}{\|\mathbf{x}_t\|^2 + \frac{1}{2C}} \quad \textit{(PA-II)}$$

- Same loss bounds as for binary classification

Uniclass prediction

- Principle of a round:
  - no input x_t
  - predicts the next element of the sequence to be w_t
  - receives y_t and suffers loss:

$$\ell_\varepsilon(\mathbf{w};\mathbf{y}) = \begin{cases} 0 & \|\mathbf{w}-\mathbf{y}\| \leq \varepsilon \\ \|\mathbf{w}-\mathbf{y}\| - \varepsilon & \text{otherwise} \end{cases}$$

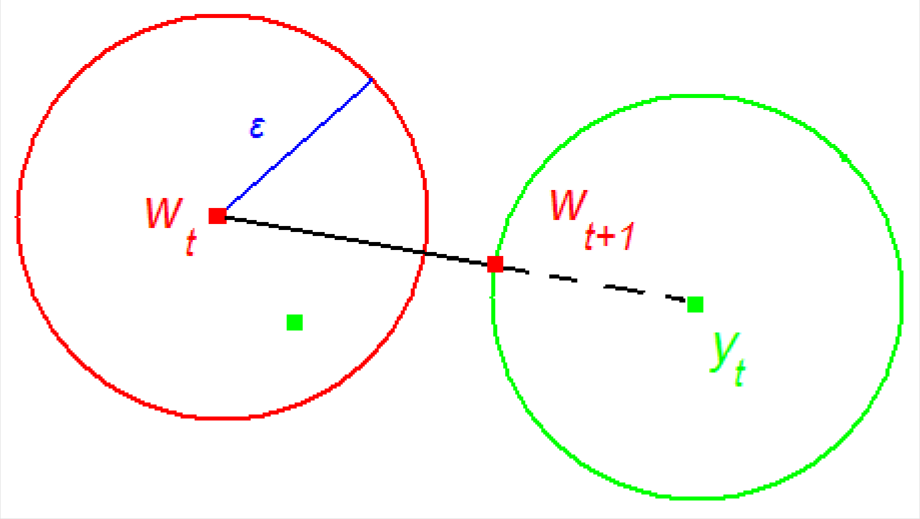


- Equivalent:
  - find the center → elements are within a radius of ε

Uniclass prediction: PA algorithms

- Update: w_{t+1} solution of optimization problem:

$$\mathbf{w}_{t+1} = \underset{\mathbf{w} \in \mathbb{R}^n}{\operatorname{argmin}} \frac{1}{2} \|\mathbf{w} - \mathbf{w}_t\|^2 \quad \text{s.t.} \quad \ell_\varepsilon(\mathbf{w}; \mathbf{y}_t) = 0$$



w_{t+1} has the form:

$$\mathbf{w}_{t+1} = \mathbf{w}_t + \tau_t \frac{\mathbf{y}_t - \mathbf{w}_t}{\|\mathbf{y}_t - \mathbf{w}_t\|}$$

$\tau_t = \ell_t$ (PA)

$\tau_t = \min\{ C , \ell_t \}$ (PA-I)

$\tau_t = \dfrac{\ell_t}{1 + \frac{1}{2C}}$ (PA-II)

Multiclass multilabel classification

- Principle:
  - set of all possible labels $Y = \{1, \ldots, k\}$
  - receives instance x_t (associated with relevant labels)
  - outputs a score for each of the k labels
    - $prediction\,vector \in \mathbb{R}^k$
  - receives the set of " relevant " labels Y_{t} for x_t
    - " relevant " must be ranked higher than " irrelevant "
  - updates the prediction mechanism

Multiclass multilabel: Problem settings

- feature vector: $\Phi(x,y) = (\Phi_1(x,y), ..., \Phi_d(x,y))$

  (set of features: $\Phi_1, ..., \Phi_d$)

- Prediction vector:

$$\left(\left(\mathbf{w}_t \cdot \Phi\left(\mathbf{x}_t, 1\right)\right), \ldots, \left(\mathbf{w}_t \cdot \Phi\left(\mathbf{x}_t, k\right)\right)\right) \qquad w_t \in \mathbb{R}^d$$

- Margin of the exemple (x_t, Y_t):

$$\gamma\left(\mathbf{w}_t; \left(\mathbf{x}_t, Y_t\right)\right) = \min_{r \in Y_t} \mathbf{w}_t \cdot \Phi\left(\mathbf{x}_t, r\right) - \max_{s \notin Y_t} \mathbf{w}_t \cdot \Phi\left(\mathbf{x}_t, s\right)$$

Multiclass multilabel: Problem settings (2)

- Margin: difference between
  - score of the lowest ranked relevant label
  - score of the highest ranked irrelevant label

$$r_t = \operatorname*{argmin}_{r \in Y_t} \mathbf{w}_t \cdot \Phi(\mathbf{x}_t, r) \quad \text{and} \quad s_t = \operatorname*{argmax}_{s \notin Y_t} \mathbf{w}_t \cdot \Phi(\mathbf{x}_t, s)$$

- Hinge-loss function:

$$\ell_{\mathrm{MC}}(\mathbf{w}; (\mathbf{x}, Y)) = \begin{cases} 0 & \gamma(\mathbf{w}; (\mathbf{x}, Y)) \geq 1 \\ 1 - \gamma(\mathbf{w}; (\mathbf{x}, Y)) & \text{otherwise} \end{cases}$$

Multiclass multilabel: PA algorithms

- Equivalence: $\ell_{\mathrm{MC}}\left(\mathbf{w}_t;(\mathbf{x}_t,Y_t)\right) = \ell\left(\mathbf{w}_t;\left(\Phi(\mathbf{x}_t,r_t)-\Phi(\mathbf{x}_t,s_t),+1\right)\right)$
- w_{t+1} solution of optimization problem:

$$\mathbf{w}_{t+1} = \underset{\mathbf{w}}{\operatorname{argmin}} \frac{1}{2}\|\mathbf{w}-\mathbf{w}_t\|^2 \quad \text{s.t.} \quad \mathbf{w}\cdot\left(\Phi(\mathbf{x}_t,r_t)-\Phi(\mathbf{x}_t,s_t)\right) \geq 1$$

- w_{t+1} has the form: $\mathbf{w}_{t+1} = \mathbf{w}_t + \tau_t\left(\Phi(\mathbf{x}_t,r_t)-\Phi(\mathbf{x}_t,s_t)\right)$

$$\tau_t = \frac{\ell_t}{\|\Phi(\mathbf{x}_t,r_t)-\Phi(\mathbf{x}_t,s_t)\|^2} \quad \text{(PA)} \qquad \tau_t = \min\left\{C\,,\ \frac{\ell_t}{\|\Phi(\mathbf{x}_t,r_t)-\Phi(\mathbf{x}_t,s_t)\|^2}\right\} \quad \text{(PA-I)}$$

$$\tau_t = \frac{\ell_t}{\|\Phi(\mathbf{x}_t,r_t)-\Phi(\mathbf{x}_t,s_t)\|^2+\frac{1}{2C}} \quad \text{(PA-II)}$$

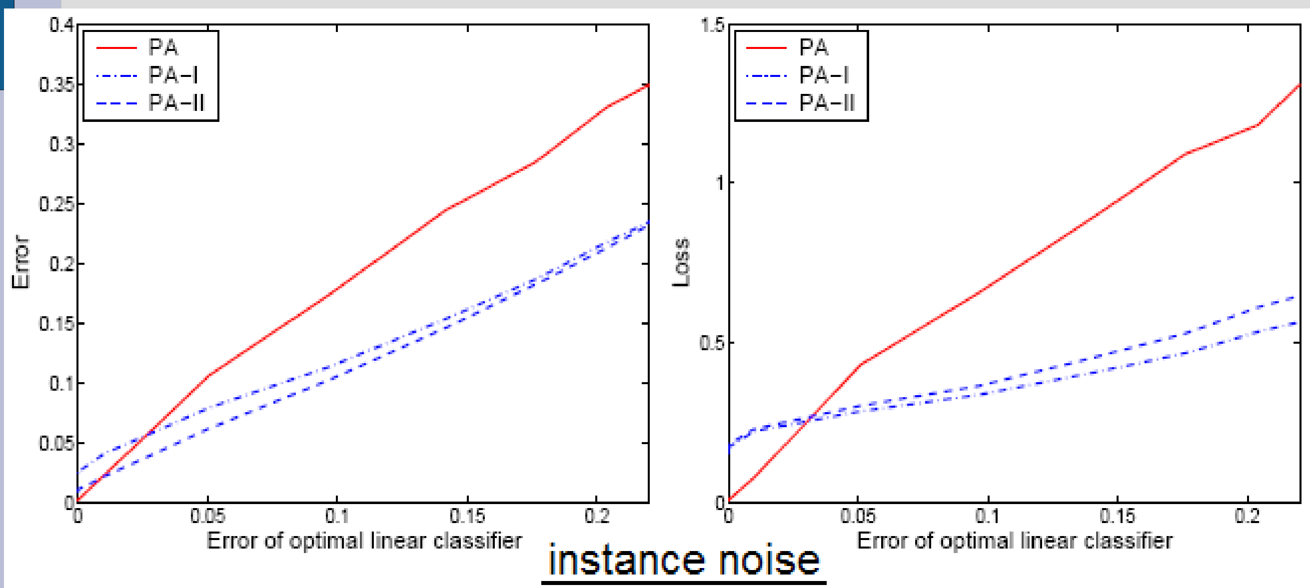
Experiments

1. Robustness to noise
2. Effect of the aggressiveness parameter C
3. Multiclass problems:

   comparison with other online algorithms

Experiment 1: Robustness to noise

- Binary classification, 4000 generated exemples (results averaged on 10 repetitions)

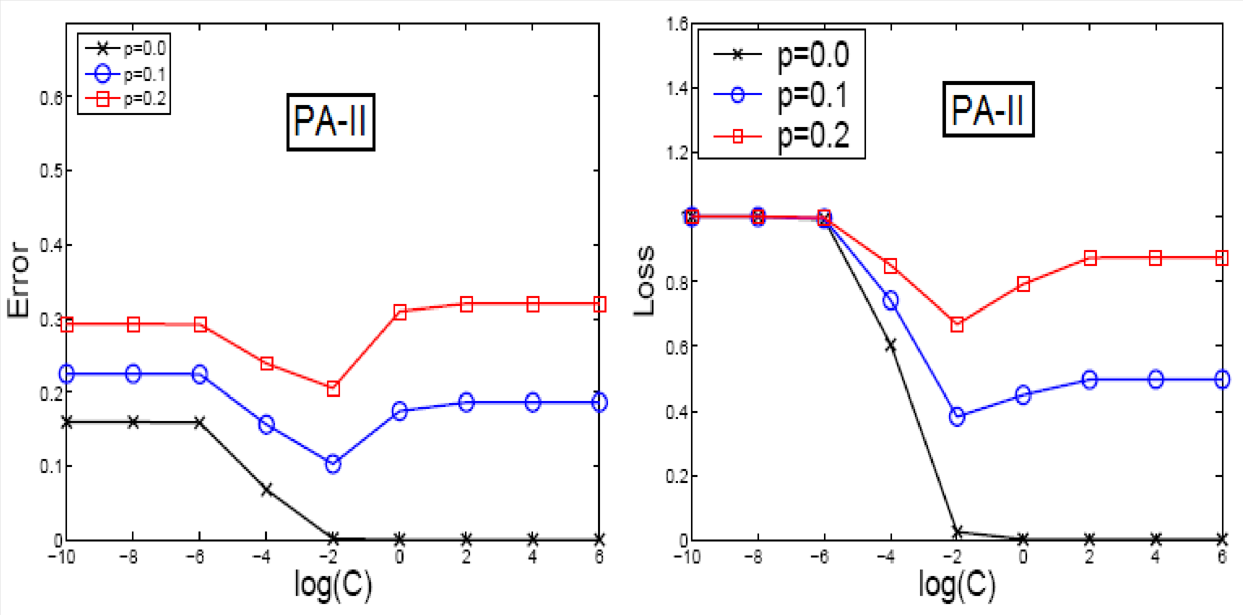


- Instance noise label noise
- Find optimal fixed linear classifier (brute force)
- C = 0.001

→ Low noise level: 3 make similar number of errors

→ High noise level: PA-I and PA-II outperform PA

Experiment 2: Effect of C



C " aggressiveness parameter "

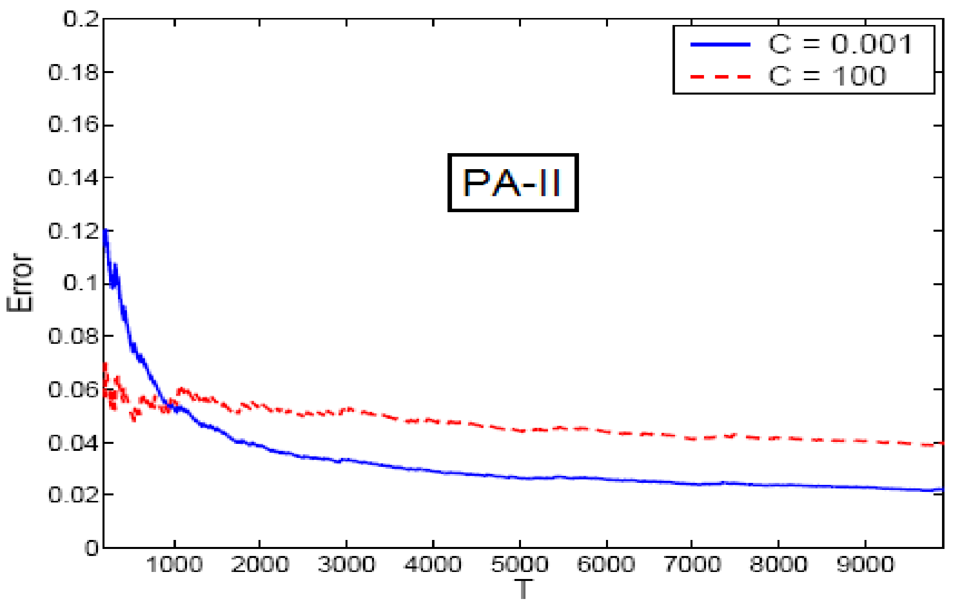
- Results meet the theoretic loss bounds

$$\sum_{t=1}^{T} \ell_t^2 \leq \left(R^2 + \frac{1}{2C}\right)\left(\|\mathbf{u}\|^2 + 2C\sum_{t=1}^{T}(\ell_t^\star)^2\right)$$

- Rule: when there is noise in data, C should be small

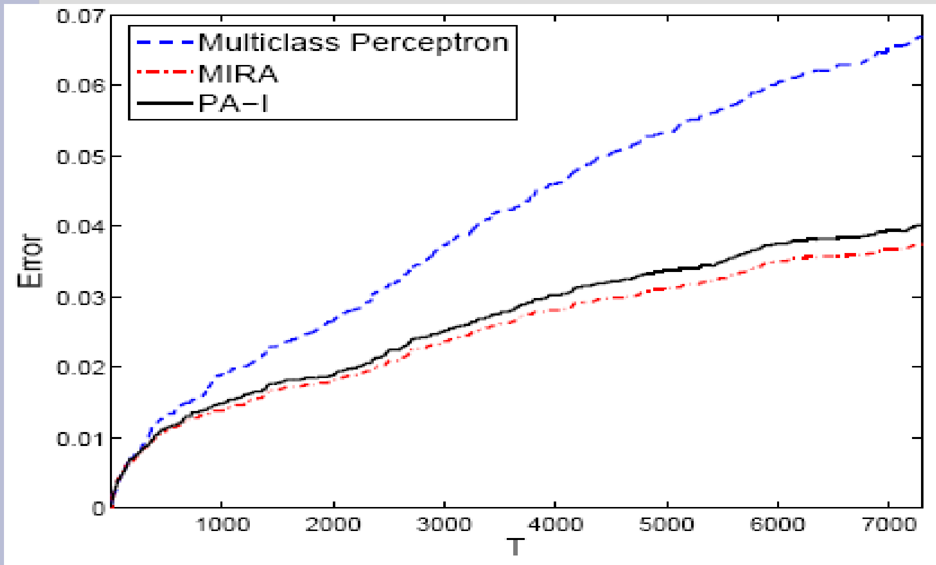
Experiment 2: Effect of C (2)

- Evolution of error rate with the number of exemples



Experiment 3: Multiclass problems

- Use standard multiclass datasets: USPS, MNIST
- Comparison of the multiclass PA algorithms with:
  - multiclass versions of the Perceptron algorithm
  - MIRA (Margin Infused Relaxed Algorithm)



- PA-I and MIRA comparable
- but MIRA solves a complex optimisation problem for each update
  ≠ PA: simple expression

Conclusion

- Further research:
  - extension to other problems
  - conversion to batch algorithms
  - PA with bounded memory constraints

    (memory requirements imposed when using Mercer Kernels)

References

- Crammer, Koby. Dekel, Ofer. Keshet, Joseph. Shalev-Shwartz, Shai. Singer, Yoram. " Online Passive-Aggressive Algorithms ". Jerusalem, 2006 <http://jmlr.csail.mit.edu/papers/volume7/crammer06a/crammer06a.pdf> (20 Oct. 2007)
- Schiele, Bernt. "Maschinelles Lernen - Statistische Verfahren". Darmstadt: Technische Universität Darmstadt, 18. Mai 2oo7 <http://www.mis.informatik.tu-darmstadt.de/Education/Courses/ml/ slides/ ml-2007-0518-svm2-v1.pdf> (16 Nov. 2007)
- Rojas, Raul. "Perceptron Learning". Neural Networks - A Systematic Introduction. Springer-Verlag, Berlin, 1996 <http://page.mi.fu-berlin.de/rojas/neural/chapter/K4.pdf> (22 Nov. 2007)
- Rodriguez, Carlos C. "The Kernel Trick". October 25, 2004 <omega.albany.edu:8008/machine-learning-dir/notes-dir/ker1/ ker1.pdf> (26 Nov. 2007)

Thank you for your attention

Questions?