The Dynamics of AdaBoost: Cyclic Behavior and Convergence of Margins

Authors:
Cynthia Rudin
Ingrid Daubechies
Robert E. Schapire

Speaker:
Bernardo Dal Seno
dalseno@elet.polimi.it

6 June 2007

What's Awaiting You

- AdaBoost
  - The algorithm and some of its features
- How and why AdaBoost works
  - Loss function and margins
- The dynamics of AdaBoost
  - Some examples
  - Asymptotic behavior
- Your questions, any time

Boosting Performance

- PAC (Probably Approximately Correct) theory
  - Some problems admit “strong” learners
  - Some problems admit “weak” learners
  - They are the very same problems!
- Boosting: make weak algorithms strong
  - Several attempt to build boosting algorithms
  - AdaBoost (Freund & Schapire, 1995) is the first “good” boosting algorithm

Boosting Classifiers

- The boosting approach to classification:
  - Devise an algorithm for building weak classifier, i.e., with poor performance (that's easy!)
  - Obtain a weak classifier for a subset of samples
  - Repeat the previous step T times
  - Combine the classifiers together
- "Weak" means "slightly better than chance"
- Two difficulties:
  - Choice of subsets
  - Combination of classifiers
- AdaBoost addresses these difficulties

Enters AdaBoost

- **Input:**
  training set: (x_i, y_i), $x_i \in X$, $y_i \in \{-1, +1\}$, $i = 1, \ldots, m$
  weak classifiers: $\{h_j : X \rightarrow \{-1, +1\}\}$
  number of iterations: T
- **Algorithm:**
  $\boldsymbol{d}_1 = \left[\frac{1}{m} \cdots \frac{1}{m}\right]^{\mathrm{T}}$ (distribution over samples)
  for $t = 1, \ldots, T$
  $h_t = \mathrm{argmin}_h\, \epsilon_t$, $\epsilon_t = \sum_{h(x_i) y_i = -1} d_{ti}$
  $$\alpha_t = \frac{1}{2} \ln\left(\frac{1-\epsilon_t}{\epsilon_t}\right)$$
  $$d_{ti} = \frac{d_{ti}}{Z_t} \times \begin{cases} e^{-\alpha_t} & \text{if } y_i = h_t(x_i) \\ e^{\alpha_t} & \text{if } y_i \neq h_t(x_i) \end{cases}$$ Z_t normalizing factor

Output:
$H_{\text{final}}(x_i) = sign\left(\sum_{t=1}^{T} \alpha_t h_t(x_i)\right)$ (weighted majority)

An Example

- See Schapire’s presentation

Loss Function

- Why AdaBoost works at all?

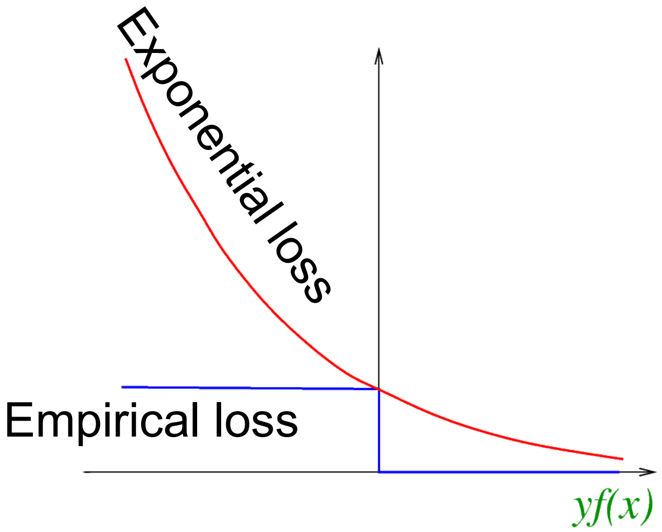
$$d_{ti} = \frac{d_{ti}}{Z_t} \exp\left(-\alpha_t y_i h_t(x_i)\right)$$

$$Z_t = \sum_i d_{ti} \exp\left(-\alpha_t y_i h_t(x_i)\right)$$

- It happens that minimizes

$$\prod Z_t = \frac{1}{m} \sum_i \exp\left(-y_i f(x_i)\right)$$

$$\text{where } f(x_i) = \sum_t \alpha_t h_t(x_i)$$



- It performs a greedy coordinate descent

Boundaries

- Training error is bounded:

  Let $\epsilon_t = 1/2 - \gamma_t$

  $$\text{Tr. err.}(H_{\text{final}}) \le \exp\left(-2\sum_t \gamma_t^2\right)$$

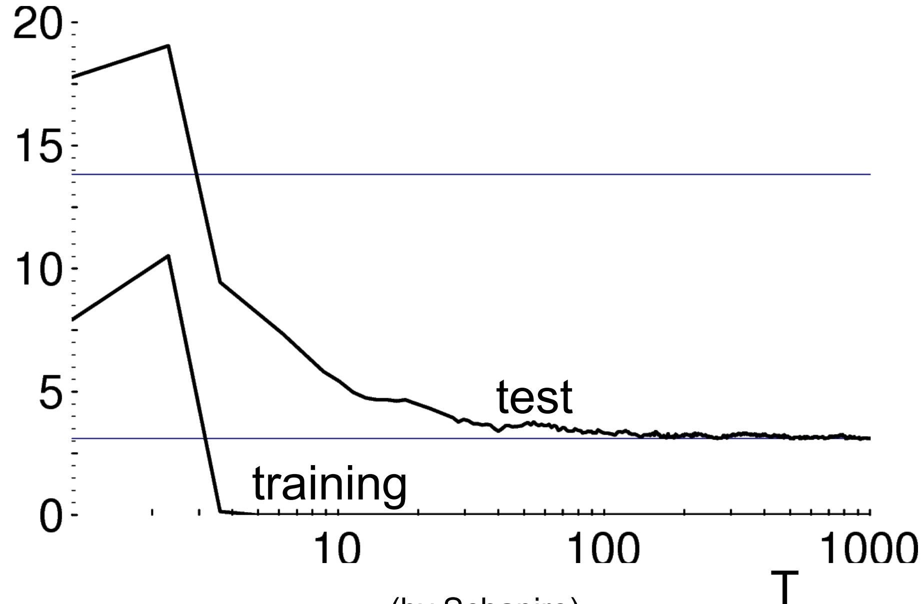
- Exponentially fast!

  if $\gamma_t \ge \gamma > 0$

  $$\text{Tr. err.}(H_{\text{final}}) \le e^{-2\gamma^2 T}$$

Minimizing Ain't Enough

- AdaBoost is an exponential-loss minimizer
- But there is more than that:
  - Often, it does not overfit



(by Schapire)

Another Story: Margins

- Margin for sample *i*:

$$\mu_i = \frac{y_i f(x_i)}{\sum_t |\alpha_t|}$$

$$\text{where } f(x_i) = \sum_t \alpha_t h_t(x_i)$$

- Margin tells how far we are from uncertainty
- We care mostly about the worst sample

$$\mu = \min_i \mu_i$$

Who Cares About Margins?

- High margins are better:

  $$\text{Generalization error} < \Pr[\mu < \theta] + \tilde{O}\left(\frac{\sqrt{d/m}}{\theta}\right)$$

  with high probability

  - Independent from *T*, *d* is the VC-dimension

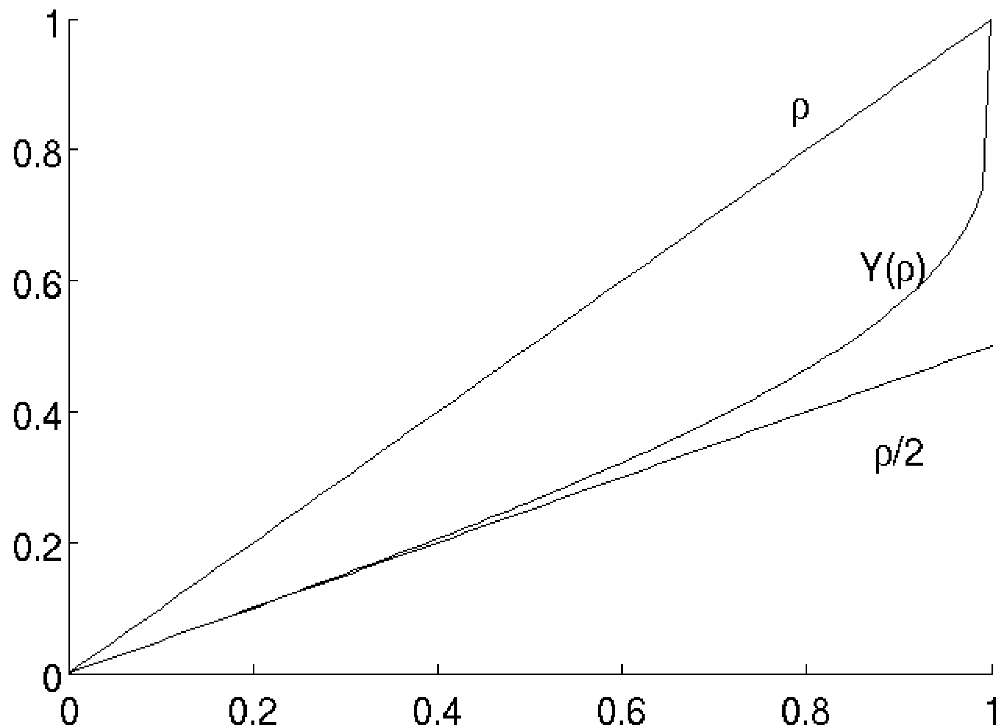
- AdaBoost is aggressive

  $\Pr[\mu < \theta] \to 0$ exponetially fast in T
  provided that $\gamma_t > \theta$

- Maybe AdaBoost maximizes the margin and hence avoids overfitting

Does AdaBoost Care About Margins?

- Schapire, Freund, Bartlett, and Lee say "Almost" AdaBoost achieves half of the maximum margin
- Rätsch and Warmuth say it can do even better
- The gap is still wide



The Matrix Is Everything

- We don’t care about weak classifier answers, just correctness

$$\epsilon_t = \sum_{h(x_i) y_i = -1} d_{ti}$$

- Suppose we can enumerate weak classifiers, all we need is a classification matrix:

$$\boldsymbol{M}, \quad M_{ij} = y_i h_j(x_i) \qquad (m \times n \text{ matrix})$$

$$\epsilon_t = \sum_{M_{ij} = -1} d_{ti}$$

- Final classifier:

$$H_{\text{final}}(x_i) = sign\left(\sum_j \lambda_j h_j(x_i)\right) \quad \lambda_j = \frac{\sum_t \mathbf{1}_{h_t = j} \alpha_t}{\sum_t |\alpha_t|}$$

AdaBoost Revisited

- Input: training set: (x_i, y_i), $x_i \in X$, $y_i \in \{-1, +1\}$, $i = 1, \ldots, m$
  classification matrix: $\boldsymbol{M}$, $M_{ij} = y_i h_j$
  number of iterations: T
- Algorithm: $\boldsymbol{\lambda}_1 = \boldsymbol{0}$ (weights of classifiers)
  for $t = 1, \ldots, T$

$$d_{ti} = \frac{e^{-(\boldsymbol{M}\boldsymbol{\lambda}_t)_i}}{\sum_{i'} e^{-(\boldsymbol{M}\boldsymbol{\lambda}_t)_{i'}}}$$ (distribution over samples)

$$j_t \in \operatorname{argmax}_j (\boldsymbol{d}_t^{\mathrm{T}} \boldsymbol{M})_j$$

$$r_t = (\boldsymbol{d}_t^{\mathrm{T}} \boldsymbol{M})_{j_t}$$ (edge of classifier j_t)

$$\alpha_t = \frac{1}{2} \ln\left(\frac{1 + r_t}{1 - r_t}\right)$$

$$\boldsymbol{\lambda}_{t+1} = \boldsymbol{\lambda}_t + \alpha_t \boldsymbol{e}_{j_t} \quad \boldsymbol{e}_{j_t} = [0 \ldots 0 1 0 \ldots 0]^{\mathrm{T}}$$

Output:

$$H_{\text{final}}(x_i) = sign\left(\sum_{j=1}^{n} \lambda_{Tj} h_j(x_i)\right)$$ (weighted majority)

Compare The "Old" AdaBoost

- **Input:**
  training set: $(x_i, y_i),\ \ x_i \in X,\ \ y_i \in \{-1,+1\},\ \ i=1,\dots,m$
  weak classifiers: $\{h_j : X \rightarrow \{-1,+1\}\}$
  number of iterations: T
- **Algorithm:**
  $$\boldsymbol{d}_1 = \left[\frac{1}{m} \cdots \frac{1}{m}\right]^{\mathrm{T}} \quad \text{(distribution over samples)}$$
  for $t=1,\dots,T$
  $$h_t = \mathrm{argmin}_h\, \epsilon_t\ ,\ \ \epsilon_t = \sum_{h(x_i) y_i = -1} d_{ti}$$
  $$\alpha_t = \frac{1}{2} \ln\left(\frac{1-\epsilon_t}{\epsilon_t}\right)$$
  $$d_{ti} = \frac{d_{ti}}{Z_t} \times \begin{cases} e^{-\alpha_t} & \text{if } y_i = h_t(x_i) \\ e^{\alpha_t} & \text{if } y_i \neq h_t(x_i) \end{cases} \quad Z_t \text{ normalizing factor}$$

Output:
$$H_{\mathrm{final}}(x_i) = sign\left(\sum_{t=1}^{T} \alpha_t h_t(x_i)\right) \quad \text{(weighted majority)}$$

Margins, Again

- Redefine the margin:

$$f(x_i) = \sum_j \bar{\lambda}_{Tj} h_j(x_i), \quad \text{where } \bar{\boldsymbol{\lambda}}_t = \frac{\boldsymbol{\lambda}_t}{\sum_j |\lambda_{tj}|}$$

$$\mu_i = y_i f(x_i) = y_i \sum_j \bar{\lambda}_{Tj} h_j(x_i) = (\boldsymbol{M}\bar{\boldsymbol{\lambda}})_i$$

- Classifier margin:

$$\mu(\bar{\boldsymbol{\lambda}}) = \min_i (\boldsymbol{M}\bar{\boldsymbol{\lambda}})_i$$

- We want high margin:

$$\rho = \max_{\bar{\lambda}} \min_i (\boldsymbol{M}\bar{\boldsymbol{\lambda}})_i \quad (Theoretical\, margin)$$

- AdaBoost minimizes:

$$F(\boldsymbol{\lambda}) = \sum_i e^{-(\boldsymbol{M}\boldsymbol{\lambda})_i}$$

AdaBoost Dynamics

- In the non-separable case AdaBoost converges

$$\text{if } \rho = 0 \quad \boldsymbol{\lambda}_t \rightarrow \boldsymbol{\lambda}^*$$
$$\boldsymbol{\lambda}^* = \operatorname{argmin}_{\lambda} \sum_i e^{-(\boldsymbol{M}\boldsymbol{\lambda})_i}$$

  - AdaBoost should achieve the maximum margin

- What about the separable case?
  - No unique solution
  - We know nothing about margins

$$\text{if } (\boldsymbol{M}\bar{\lambda})_i > 0 \quad \lim_{a \rightarrow \infty} F(a\bar{\lambda}) = 0$$

Yet Another View Of AdaBoost

- AdaBoost in three simple steps (iterated map):

$$j_t \in \operatorname{argmax}_j (\boldsymbol{d}_t^{\top} \boldsymbol{M})_j$$
$$r_t = (\boldsymbol{d}_t^{\top} \boldsymbol{M})_{j_t}$$
$$d_{t+1,i} = \frac{d_{ti}}{1 + M_{ij_t} r_t}$$

- The separable case does not converge to a point:
  - From the *min-max* theorem:

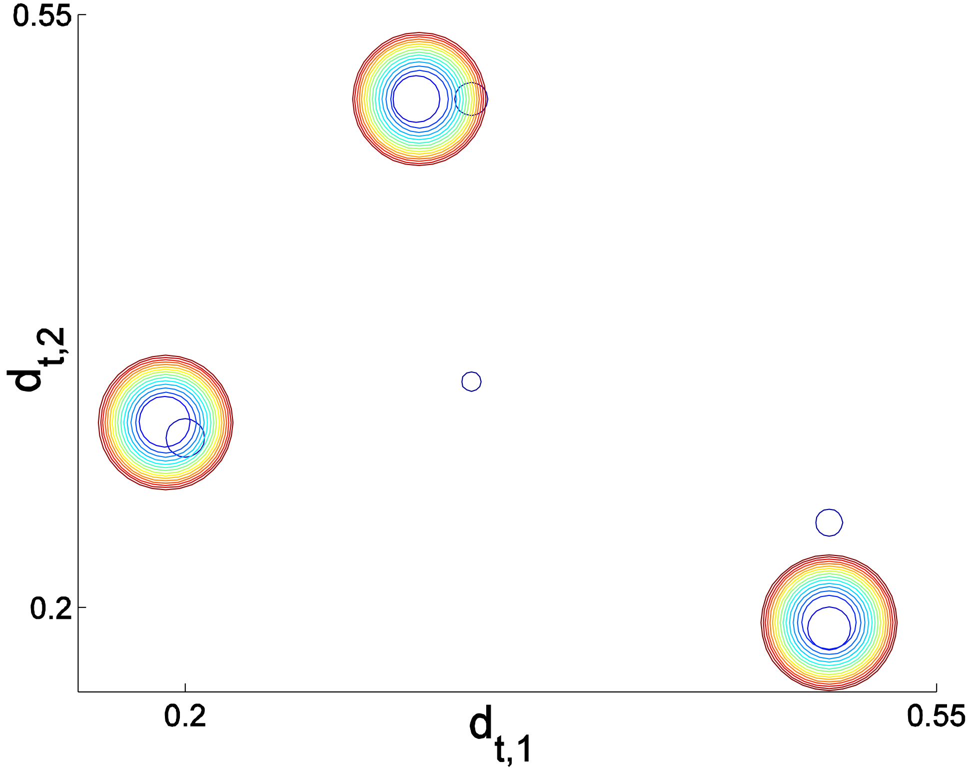
$$\rho = \max_{\bar{\lambda}} \min_i (\boldsymbol{M} \bar{\lambda})_i = \min_{\boldsymbol{d}} \max_j (\boldsymbol{d}^{\top} \boldsymbol{M})_i$$

  - Then:

$$r_t \geq \rho > 0$$
$$\text{if } d_{ti} > 0 \quad d_{t+1,i} \neq d_{ti}$$

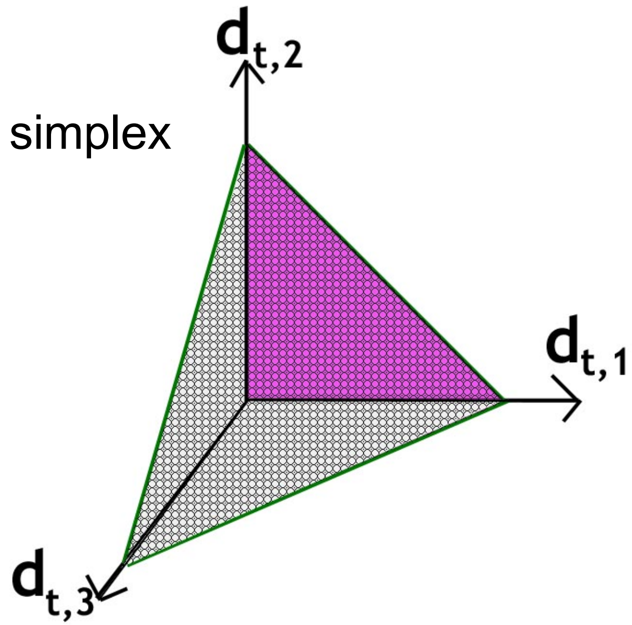
A Detailed Example

$$\boldsymbol{M} = \begin{pmatrix} -1 & 1 & 1 \\ 1 & -1 & 1 \\ 1 & 1 & -1 \end{pmatrix}$$



Projection of $\boldsymbol{d}_t$ on the violet triangle
Circles grow with t

d lies on a simplex

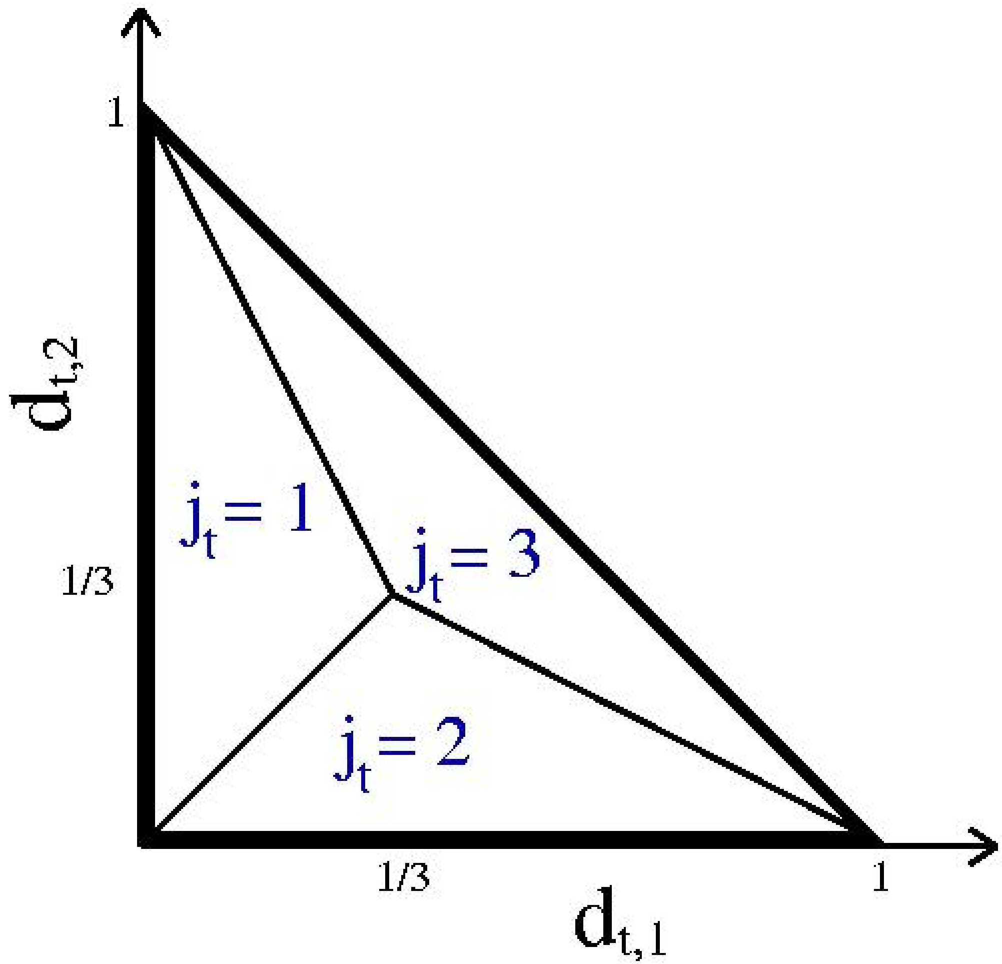


$$j_t \in \operatorname{argmax}_j (\boldsymbol{d}_t^{\mathsf{T}} \boldsymbol{M})_j$$

$$r_t = (\boldsymbol{d}_t^{\mathsf{T}} \boldsymbol{M})_{j_t}$$

$$d_{t+1,i} = \frac{d_{ti}}{1 + M_{ij_t} r_t}$$

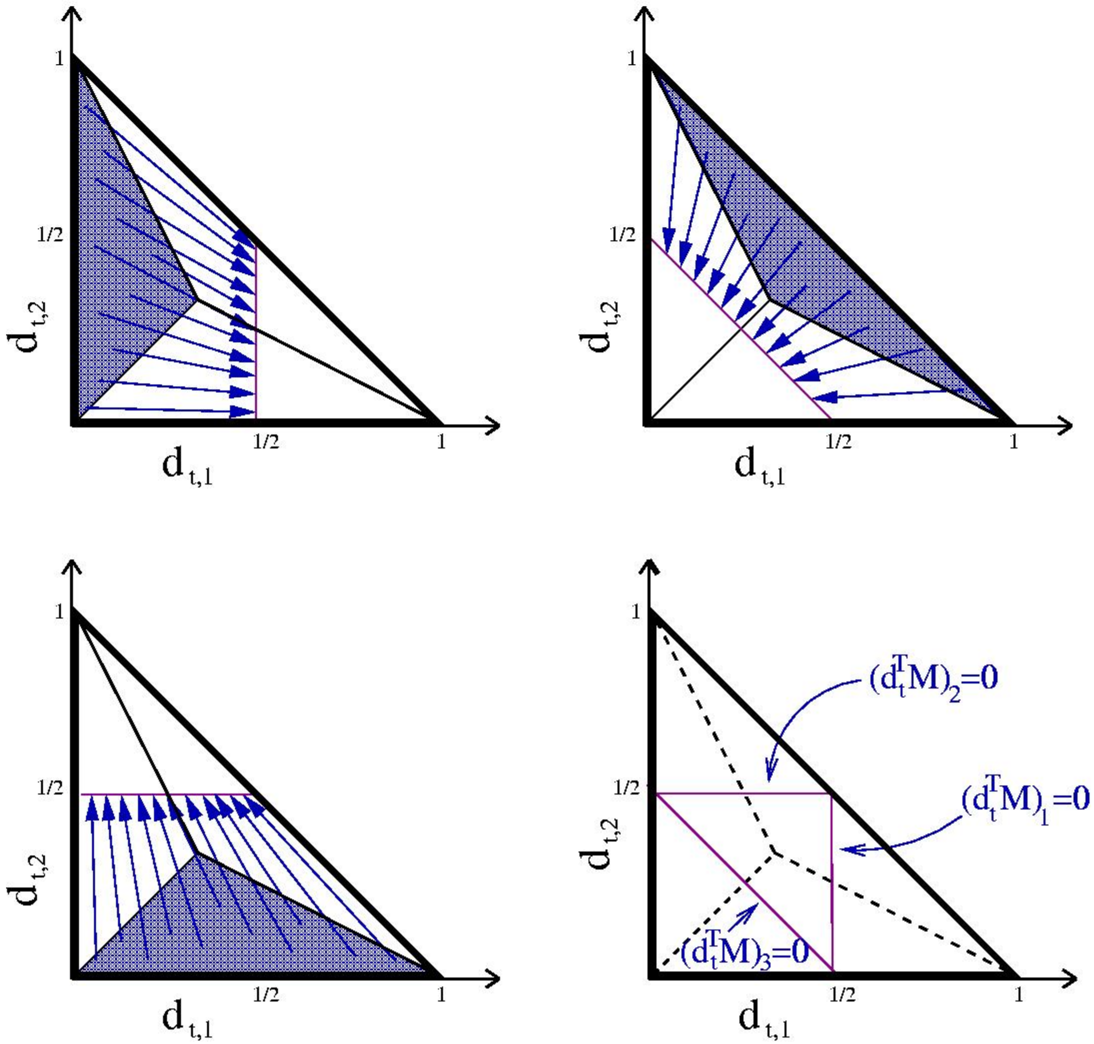
A Detailed Example: Fields



$$\boldsymbol{M} = \begin{pmatrix} -1 & 1 & 1 \\ 1 & -1 & 1 \\ 1 & 1 & -1 \end{pmatrix}$$

$$j_t \in \operatorname{argmax}_j (\boldsymbol{d}_t^{\mathrm{T}} \boldsymbol{M})_j$$
$$r_t = (\boldsymbol{d}_t^{\mathrm{T}} \boldsymbol{M})_{j_t}$$
$$d_{t+1,i} = \frac{d_{ti}}{1 + M_{ij_t} r_t}$$

A Detailed Example: Maps



$$\boldsymbol{M} = \begin{pmatrix} -1 & 1 & 1 \\ 1 & -1 & 1 \\ 1 & 1 & -1 \end{pmatrix}$$

$$j_t \in \mathrm{argmax}_j (\boldsymbol{d}_t{}^T \boldsymbol{M})_j$$

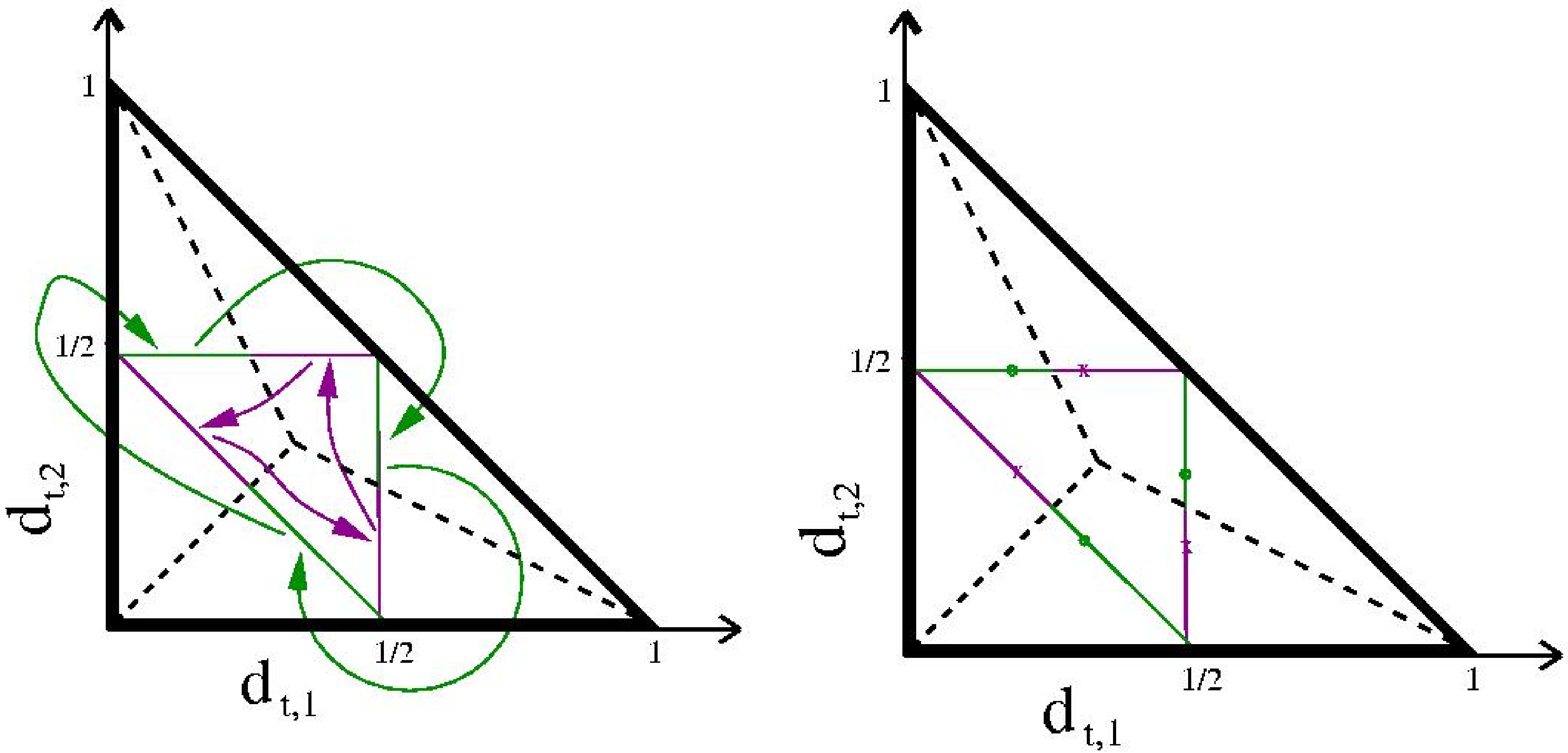
$$r_t = (\boldsymbol{d}_t{}^T \boldsymbol{M})_{j_t}$$

$$d_{t+1,i} = \frac{d_{ti}}{1 + M_{ij_t} r_t}$$

$$(\boldsymbol{d}_{t+1}{}^T \boldsymbol{M})_{j_t} = 0$$

A Detailed Examples: Contractions

- There are two cycles of period 3
- Both achieve the maximum margin 1/3



Many Cycles

- Repeated lines
- There exists a stable manifold of 3-cycles
- Weight can be moved around

$\boldsymbol{d}_t$ and $\boldsymbol{d}_t' = \boldsymbol{d}_t + \boldsymbol{a}$ behave the same
if $\sum_{i \in I_k} a_i = 0$
$(I_k$ are groups indentical rows$)$

$$\boldsymbol{M} = \begin{pmatrix} -1 & 1 & 1 \\ & \dots & \\ -1 & 1 & 1 \\ 1 & -1 & 1 \\ & \dots & \\ 1 & -1 & 1 \\ 1 & 1 & -1 \\ & \dots & \\ 1 & 1 & -1 \end{pmatrix}$$

$$j_t \in \operatorname{argmax}_j (\boldsymbol{d}_t^{\top} \boldsymbol{M})_j$$
$$r_t = (\boldsymbol{d}_t^{\top} \boldsymbol{M})_{j_t}$$
$$d_{t+1,i} = \frac{d_{ti}}{1 + M_{ij_t} r_t}$$

A Bigger Example

$$\boldsymbol{M} = \begin{pmatrix} -1 & 1 & \dots & 1 \\ 1 & -1 & \dots & 1 \\ \vdots & & \ddots & \\ 1 & 1 & \dots & -1 \end{pmatrix}$$

- There exist (at least) (m-1)! stable cycles
- Those cycles achieve the maximum margin

Support Vectors

- Support vectors are training samples *i* such that in cycles $d_{ti} > 0$
- If a cycle is stable, for each *i* either:

$$d_{1,i} = 0$$
$$\prod_{t=1}^{T_c} (1 + M_{ij_t} r_t) > 1 \quad (d_{ti} \to 0)$$
$$\prod_{t=1}^{T_c} (1 + M_{ij_t} r_t) = 1 \quad (d_{ti} = d_{t+T_c,i})$$

- The last condition holds for support vectors
  - They are difficult samples
  - AdaBoost concentrates on them

$$j_t \in \operatorname{argmax}_j (\boldsymbol{d}_t^{\mathrm{T}} \boldsymbol{M})_j$$
$$r_t = (\boldsymbol{d}_t^{\mathrm{T}} \boldsymbol{M})_{j_t}$$
$$d_{t+1,i} = \frac{d_{ti}}{1 + M_{ij_t} r_t}$$

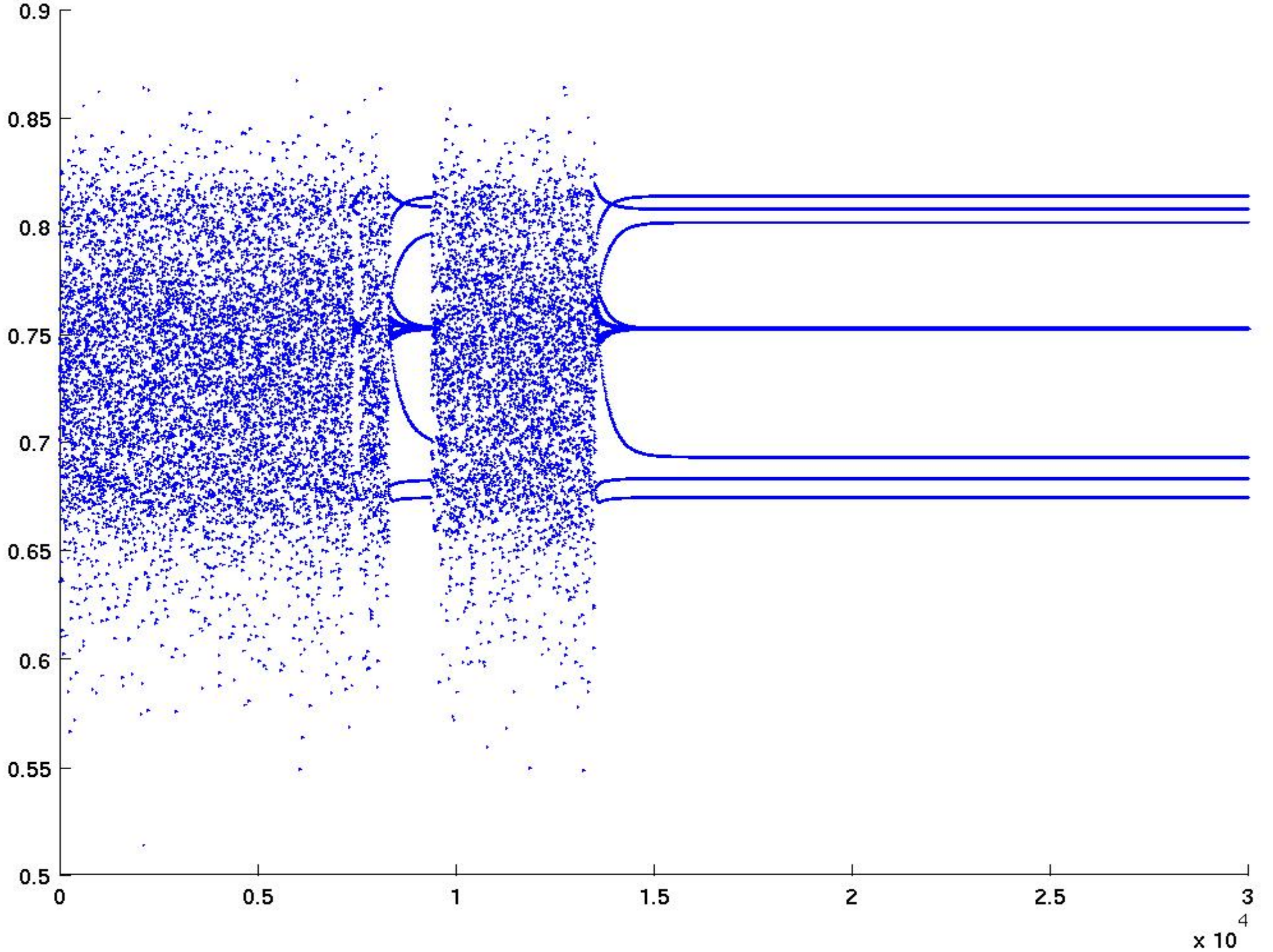
AdaBoost And Margins

- AdaBoost produces the same margin for each support vector and larger margins for other training samples
- This is the margin of H_{final}

- AdaBoost does not always converge to the optimal margin
  - There are counterexamples

More Than Cycles

- Chaotic-like behavior is also possible



The End

That's enough for today

Thank you!

And many thanks to C. Rudin and R. Schapire for their work and their material I shamelessly used in this presentation :-)

Thanks also to www.clipartheaven.com for this image

More To Come

- For other exciting seminars, stay tuned on

http://prlt.elet.polimi.it/mediawiki/index.php/Poli_Interest_Group_for_Machine_Learning

References

- Cynthia Rudin, Ingrid Daubechies, and Robert E. Schapire, “The Dynamics of AdaBoost: Cyclic Behavior and Convergence of Margins”, *The Journal of Machine Learning Research*, Vol. 5, 2004
- Yoav Freund and Robert E. Schapire, “A short introduction to boosting”, *Journal of Japanese Society for Artificial Intelligence*, 14(5):771-780, September, 1999.
- Robert E. Schapire, “The boosting approach to machine learning: An overview”, in *MSRI Workshop on Nonlinear Estimation and Classification*, 2002
- The Web pages of Cynthia Rudin http://www1.cs.columbia.edu/~rudin/main.html and Robert Schapire http://www.cs.princeton.edu/~schapire/ host the above papers and many others
- Cynthia Rudin’s lecture on AdaBoost dynamics at the Chicago Machine Learning Summer School 2005 (incomplete): http://videolectures.net/mlss05us_rudin_da/
- Schapire’s lecture on AdaBoost at the Chicago Machine Learning Summer School 2005: http://videolectures.net/mlss05us_schapire_b/