Weak Learnability = Linear Separability

New Relaxations and Efficient Boosting Algorithms

Shai Shalev-Shwartz

Yoram Singer

COLT, July 2008

Outline

- Weak Learnability = Linear separability
  - Follows directly from Von-Neumann's minimax theorem
- Relaxations
  - The equivalence yields a family of relaxations to the separability assumption
  - Proof technique: Fenchel duality & Infimal Convolution
- Boosting Algorithms
  - A primal-dual algorithm
  - Applicable to entire family of relaxations
  - Rate of convergence analysis

Boosting

Input:

- m training examples $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_m, y_m)$
- n base hypotheses $h_1, \ldots, h_n$

Boosting

Input:

- m training examples $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_m, y_m)$
- n base hypotheses $h_1, \ldots, h_n$

$$A = \begin{pmatrix} y_1\, h_1(\mathbf{x}_1) & \ldots & y_1\, h_n(\mathbf{x}_1) \\ \vdots & \ddots & \vdots \\ y_m\, h_1(\mathbf{x}_m) & \ldots & y_m\, h_n(\mathbf{x}_m) \end{pmatrix}$$

Boosting

Input:

- m training examples $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_m, y_m)$
- n base hypotheses $h_1, \ldots, h_n$

$$A = \begin{pmatrix} y_1\, h_1(\mathbf{x}_1) & \ldots & y_1\, h_n(\mathbf{x}_1) \\ \vdots & \ddots & \vdots \\ y_m\, h_1(\mathbf{x}_m) & \ldots & y_m\, h_n(\mathbf{x}_m) \end{pmatrix}$$

Output:

- 'strong' hypothesis $H_{\mathbf{w}}(\cdot) \;=\; \sum_{i=1}^{n} w_i h_i(\cdot)$

Weak Learnability

Definition: γ-weak-learnability

A matrix A is γ-weak-learnable if

$$\gamma \;=\; \min_{\mathbf{d}\in\mathbb{S}^m} \; \max_{j\in[n]} \left|(d^\dagger A)_j\right| \, .$$

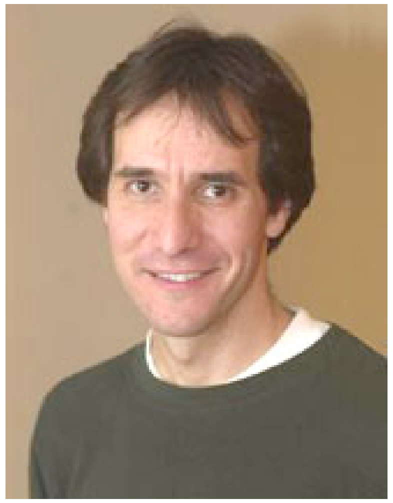
Probability simplex
(distributions over examples)

'edge' of j'th hypothesis

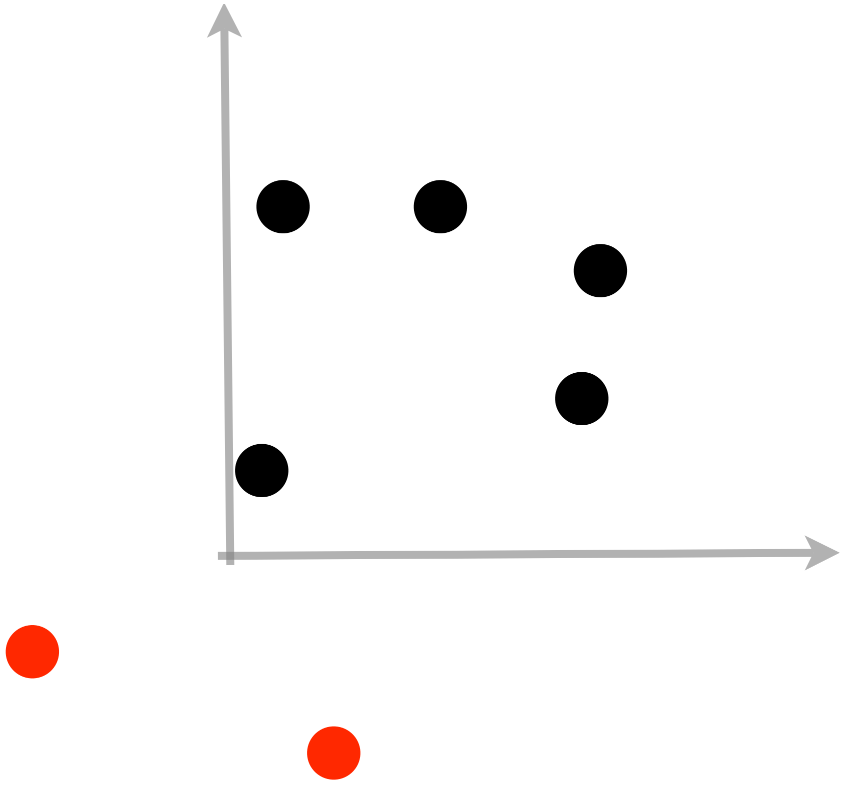
$$(d^\dagger A)_j = \sum_i d_i y_i h_j(\mathbf{x}_i)$$

(dagger for transpose)

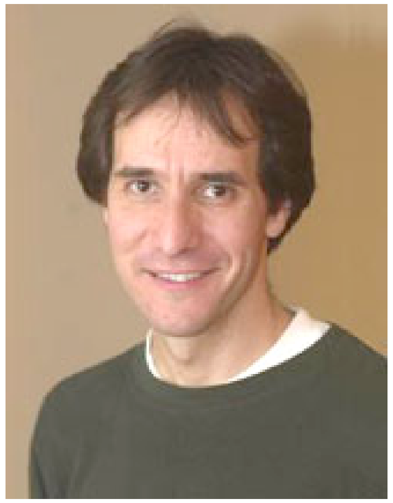
Weak-to-strong learnablity



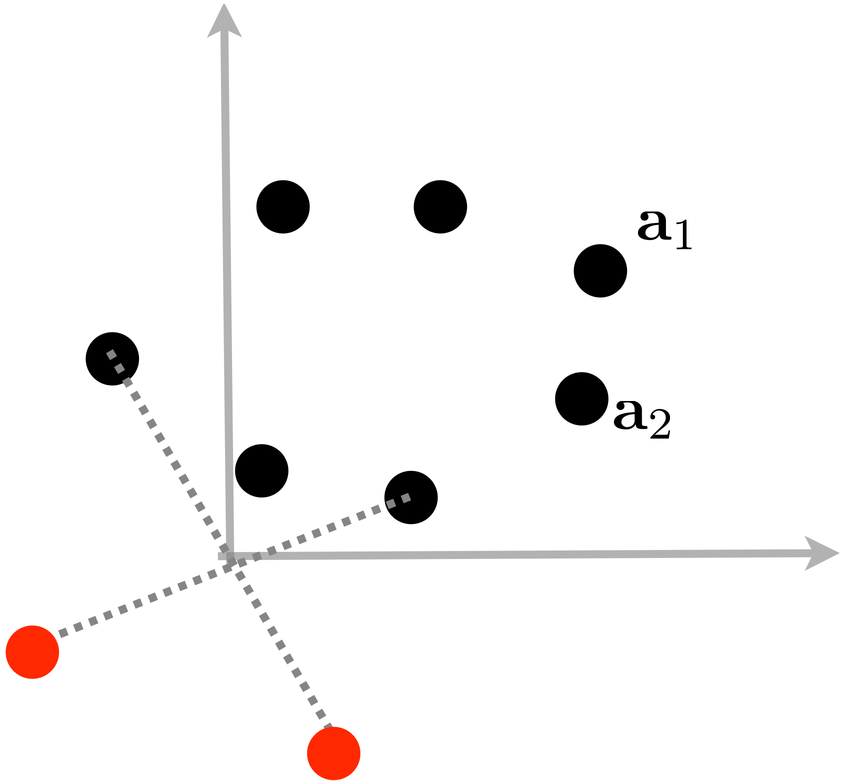
Schapire: Weak learnablity implies separability



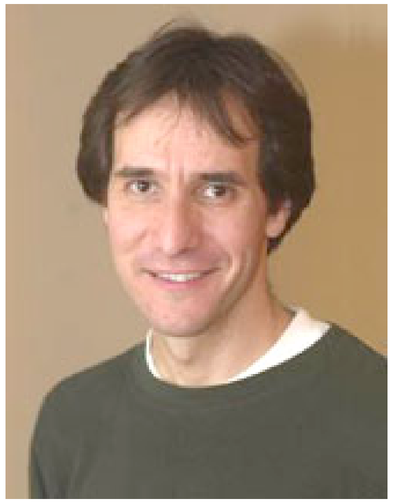
Weak-to-strong learnablity



Schapire: Weak learnablity implies separability

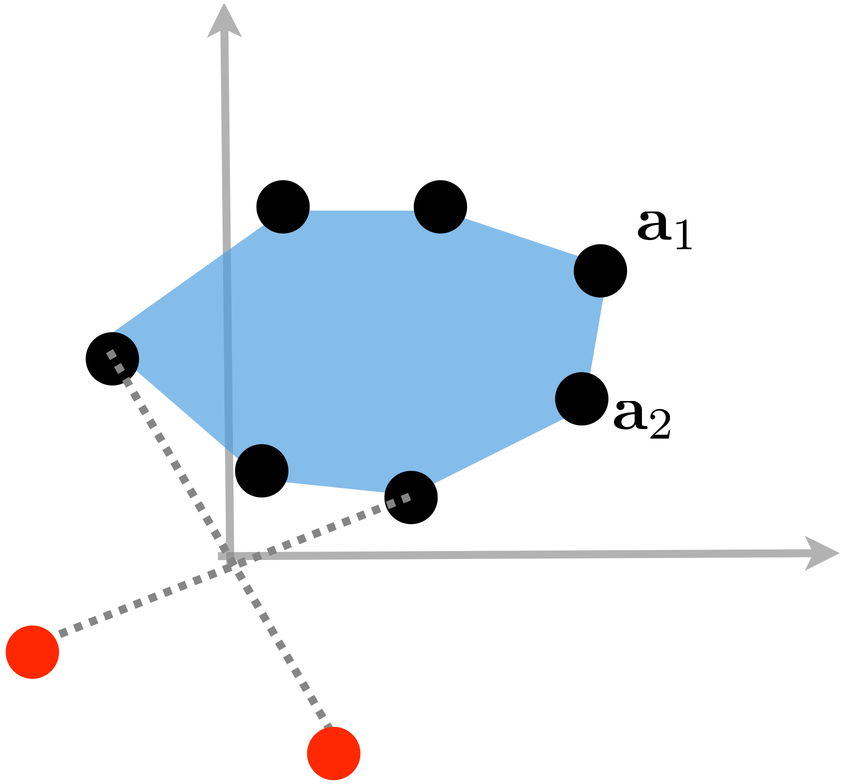


Weak-to-strong learnablity



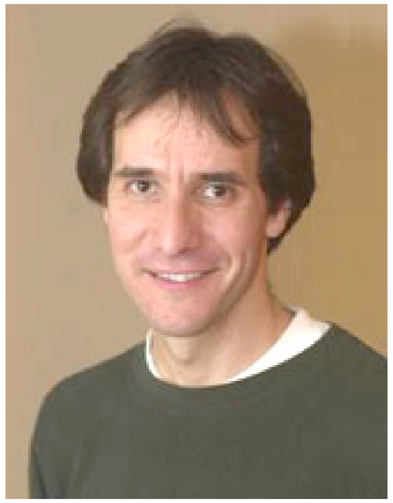
Schapire: Weak learnablity implies separability

- **Weak learnability**: Convex hull of rows of A does not contain the origin

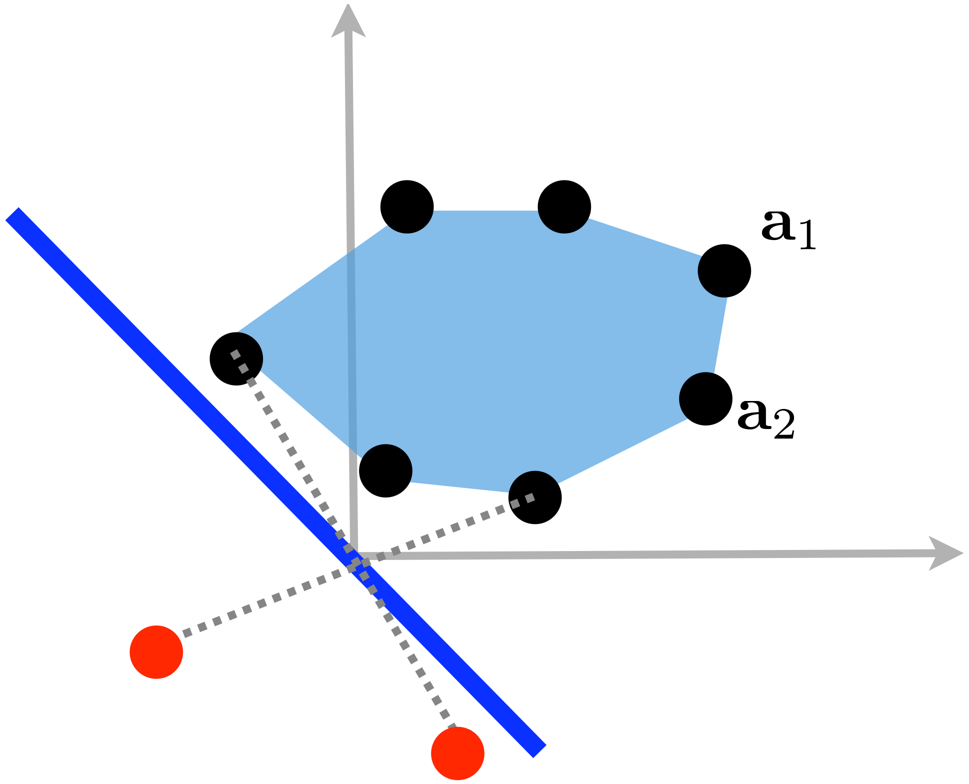


Weak-to-strong learnablity

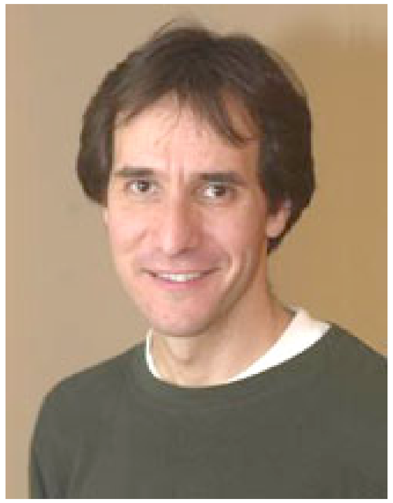
Schapire: Weak learnablity implies separability



- **Weak learnability**: Convex hull of rows of A does not contain the origin
- **Separability**: Exists hyperplane that goes through origin s.t. all rows of A resides in one side



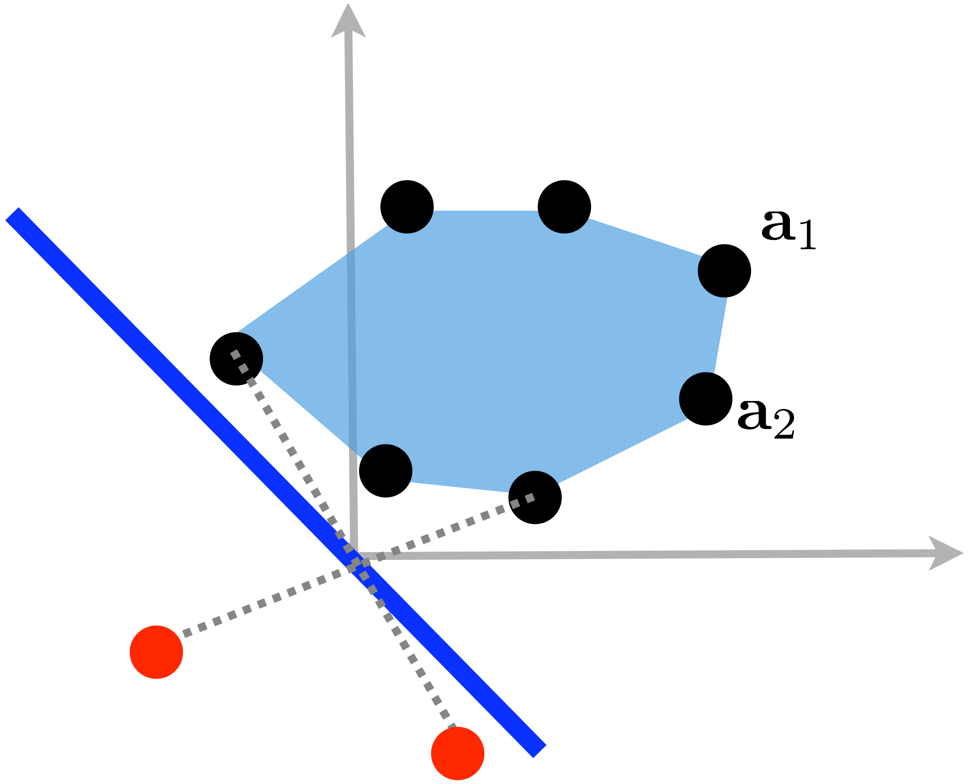
Weak-to-strong learnablity



Schapire: Weak learnablity implies separability

- **Weak learnability**: Convex hull of rows of A does not contain the origin
- **Separability**: Exists hyperplane that goes through origin s.t. all rows of A resides in one side

Quantification ?



Linear Separability

Definition: separability with ℓ_1 margin γ

A matrix A is linearly separable with ℓ_1 margin γ if

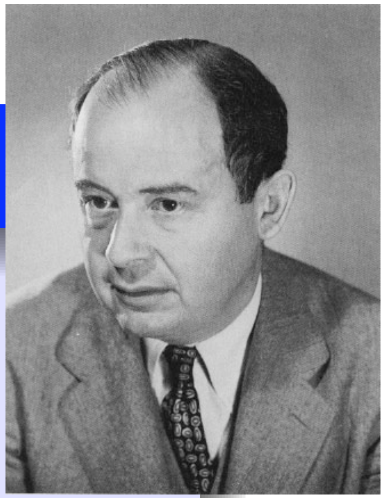
$$\gamma = \max_{\mathbf{w} \in \mathbb{B}_1^n} \min_{i \in [m]} (A\mathbf{w})_i$$

unit ℓ_1 ball
(weights over features)

'margin' of i^{th} example

$$(Aw)_i = y_i \sum_j w_j h_j(\mathbf{x}_i)$$

Weak Learnability = Linear Separability



Theorem

The following properties are equivalent:

- matrix A is γ-weak-learnable
- matrix A is linearly separable with ℓ_1 margin of γ

In other words,

$$\max_{\mathbf{w}\in\mathbb{B}_1^n} \min_{i\in[m]} (A\mathbf{w})_i = \min_{\mathbf{d}\in\mathbb{S}^m} \max_{j\in[n]} |(d^\dagger A)_j|$$

Proof: Equivalence follows from Von-Neumann's minimax theorem

Relaxations -- Main Idea

$$\max_{\mathbf{w}\in\mathbb{B}_1^n} \min_{i\in[m]} (A\mathbf{w})_i \;=\; \min_{\mathbf{d}\in\mathbb{S}^m} \max_{j\in[n]} |(d^\dagger A)_j|$$

Relaxations -- Main Idea

hard margin

$$\max_{\mathbf{w}\in\mathbb{B}_1^n} \boxed{\min_{i\in[m]} (A\mathbf{w})_i} = \min_{\mathbf{d}\in\mathbb{S}^m} \max_{j\in[n]} |(d^\dagger A)_j|$$

Relaxations -- Main Idea

hard margin

$$\max_{\mathbf{w}\in\mathbb{B}_1^n} \min_{i\in[m]} (A\mathbf{w})_i = \min_{\mathbf{d}\in\mathbb{S}^m} \max_{j\in[n]} |(d^\dagger A)_j|$$

$$\max_{\mathbf{w}\in\mathbb{B}_1^n} \boxed{?} = \min_{\mathbf{d}\in\mathbb{S}^m\cap C} \max_{j\in[n]} |(d^\dagger A)_j|$$

Relaxed weak-learnability

Relaxations -- Theorem

hard margin

$$\max_{\mathbf{w}\in\mathbb{B}_1^n} \min_{i\in[m]} (A\mathbf{w})_i = \min_{\mathbf{d}\in\mathbb{S}^m} \max_{j\in[n]} |(d^\dagger A)_j|$$

$$\max_{\mathbf{w}\in\mathbb{B}_1^n} \text{ soft margin } = \min_{\mathbf{d}\in\mathbb{S}^m \cap C} \max_{j\in[n]} |(d^\dagger A)_j|$$

$$\max_{\gamma\in\mathbb{R}} \gamma - \nu \, \|[\gamma - A\mathbf{w}]_+\|_\star$$

$$C = \{\mathbf{w} : \|\mathbf{w}\| \leq \nu\}$$

Relaxations -- Theorem

hard margin

$$\max_{\mathbf{w}\in\mathbb{B}_1^n} \min_{i\in[m]} (A\mathbf{w})_i = \min_{\mathbf{d}\in\mathbb{S}^m} \max_{j\in[n]} |(d^\dagger A)_j|$$

$$\max_{\mathbf{w}\in\mathbb{B}_1^n} \boxed{\text{soft margin}} = \min_{\mathbf{d}\in\mathbb{S}^m\cap C} \max_{j\in[n]} |(d^\dagger A)_j|$$

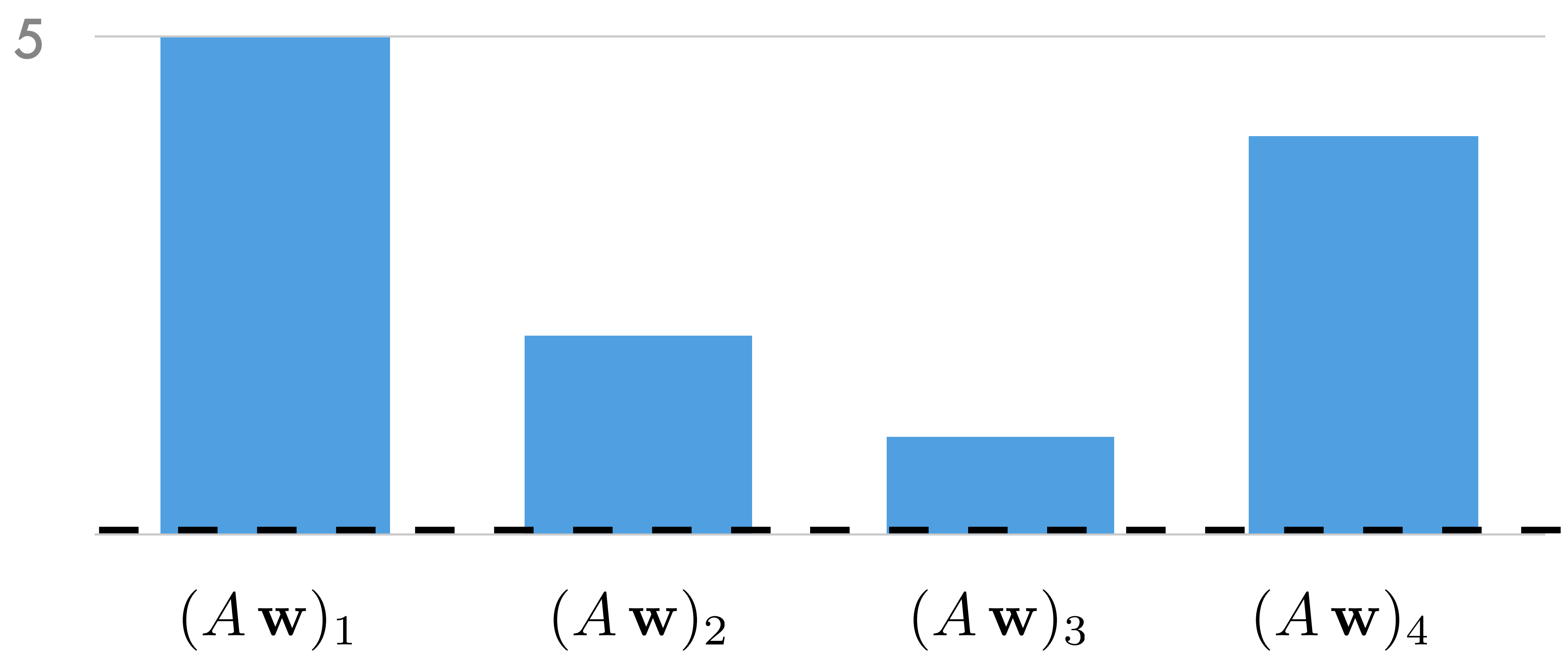
$$\max_{\gamma\in\mathbb{R}} \gamma - \nu \, \|[\gamma - A\mathbf{w}]_+\|_\star$$

$$C = \{\mathbf{w} : \|\mathbf{w}\| \leq \nu\}$$

Relaxations -- Example

- If $C = \{\mathbf{w} : \|\mathbf{w}\|_\infty \leq \frac{1}{k}\}$ soft margin is:

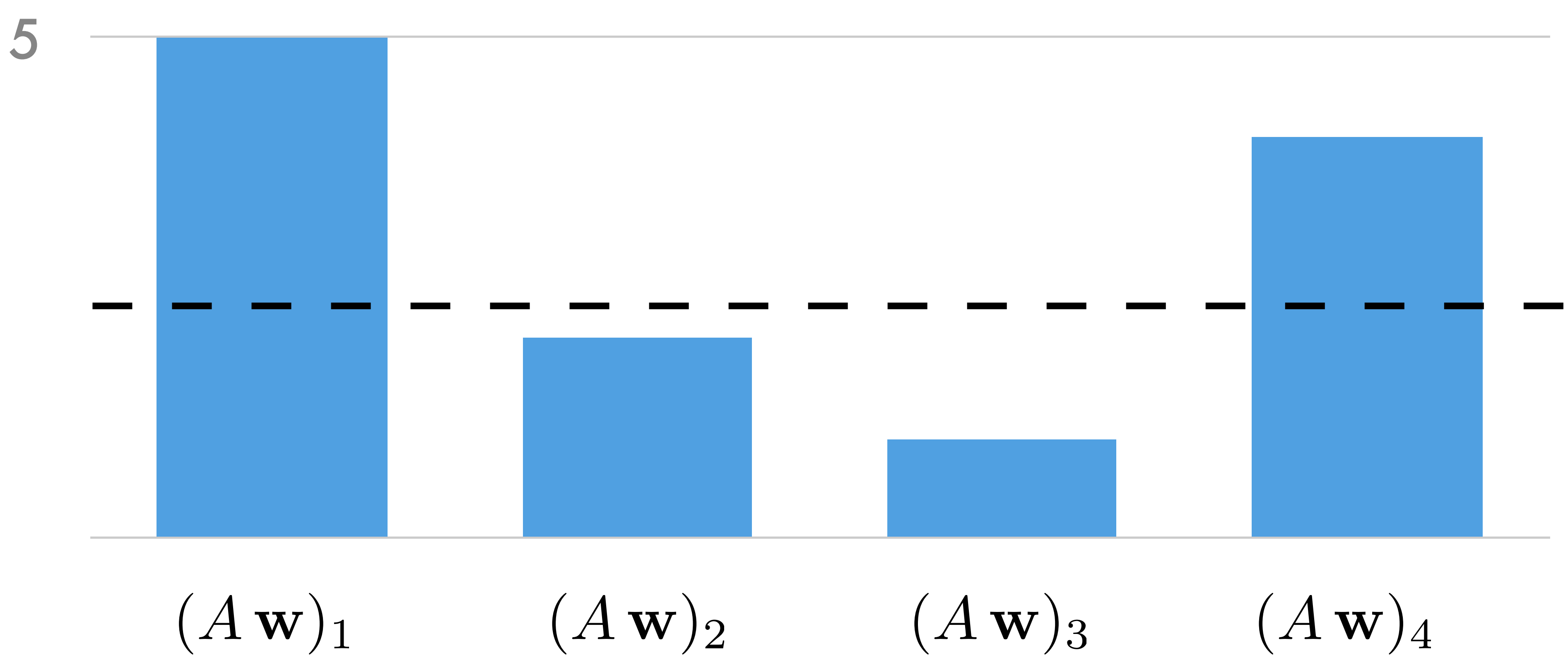
$$\max_{\gamma} \gamma - \frac{1}{k} \|[\gamma - A\mathbf{w}]_+\|_1 \;=\; \mathrm{AvgMin}_k(A\,\mathbf{w})$$



Relaxations -- Example

- If $C = \{\mathbf{w} : \|\mathbf{w}\|_\infty \leq \frac{1}{k}\}$ soft margin is:

$$\max_{\gamma} \gamma - \frac{1}{k} \|[\gamma - A\mathbf{w}]_+\|_1 \;=\; \mathrm{AvgMin}_k(A\,\mathbf{w})$$



Proof Technique

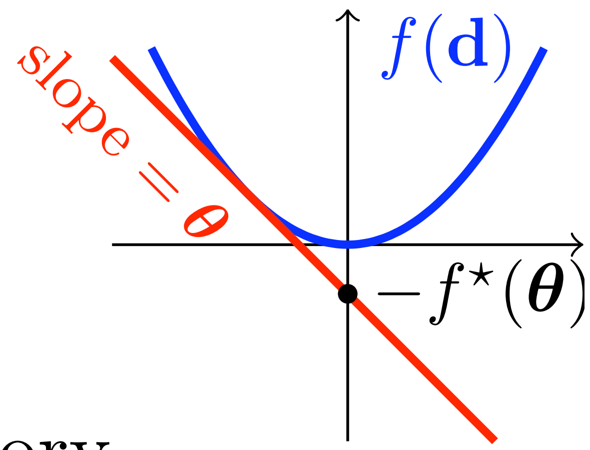
$$\min_{\mathbf{d}\in\mathbb{S}^m\cap C}\max_i |(\mathbf{d}^\dagger A)_i| \;=\; \min_{\mathbf{d}} f(\mathbf{d}) + g(\mathbf{d}^t A) \;=\; \max_{\mathbf{w}} -f^\star(-A\,\mathbf{w}) - g^\star(\mathbf{w})$$

$\mathrm{supp}(\mathbb{S}^m \cap C)$

$\|\cdot\|_\infty$

Fenchel Duality

- In our case:
  - $g^\star(\cdot) = \mathrm{supp}(\mathbb{B}_1^n)$
  - The tricky part is to show that $f^\star(\boldsymbol{\theta}) = -\max_{\gamma\in\mathbb{R}}(\gamma - \nu\|[\gamma+\theta]_+\|_\star)$



- We show that using **infimal convolution** theory

$$f_1^\star + f_2^\star = (f_1 \otimes_{\inf} f_2)^\star$$

A Boosting Algorithm

INITIALIZE: $\mathbf{w}_1 = \mathbf{0}, \quad \beta = \frac{\epsilon}{2\log(m)}$

FOR $t = 1, 2, \ldots, T$

$$\mathbf{d}_t = \operatorname*{argmin}_{\mathbf{d} \in \mathbb{S}^m \cap C} D_{\mathrm{KL}}(\mathbf{d}, \hat{\mathbf{d}}) \text{ where } \hat{d}_{t,i} \propto \exp\left(-\tfrac{1}{\beta}(A\,\mathbf{w}_t)_i\right)$$

$$j_t \in \arg\max_j |(\mathbf{d}_t^\dagger A)_j|$$

$$\eta_t = \max\left\{0, \min\left\{1, \frac{\beta\,\mathbf{d}_t^\dagger A(\mathbf{e}^{j_t} - \mathbf{w}_t)}{\|A(\mathbf{e}^{j_t} - \mathbf{w}_t)\|_\infty^2}\right\}\right\}$$

$$\mathbf{w}_{t+1} = (1 - \eta_t)\mathbf{w}_t + \eta_t\,\mathbf{e}^{j_t}$$

A Boosting Algorithm

desired accuracy

INITIALIZE: $\mathbf{w}_1 = \mathbf{0}, \quad \beta = \frac{\epsilon}{2\log(m)}$

FOR $t = 1, 2, \ldots, T$

$$\mathbf{d}_t = \operatorname*{argmin}_{\mathbf{d} \in \mathbb{S}^m \cap C} D_{\mathrm{KL}}(\mathbf{d}, \hat{\mathbf{d}}) \text{ where } \hat{d}_{t,i} \propto \exp\left(-\tfrac{1}{\beta}(A\,\mathbf{w}_t)_i\right)$$

$$j_t \in \arg\max_j |(\mathbf{d}_t^\dagger A)_j|$$

$$\eta_t = \max\left\{0, \min\left\{1, \frac{\beta\,\mathbf{d}_t^\dagger A(\mathbf{e}^{j_t} - \mathbf{w}_t)}{\|A(\mathbf{e}^{j_t} - \mathbf{w}_t)\|_\infty^2}\right\}\right\}$$

$$\mathbf{w}_{t+1} = (1 - \eta_t)\mathbf{w}_t + \eta_t\,\mathbf{e}^{j_t}$$

A Boosting Algorithm

'algorithmic relaxation'

INITIALIZE: $\mathbf{w}_1 = \mathbf{0}$, $\beta = \frac{\epsilon}{2\log(m)}$

FOR $t = 1, 2, \ldots, T$

$$\mathbf{d}_t = \operatorname*{argmin}_{\mathbf{d} \in \mathbb{S}^m \cap C} D_{\mathrm{KL}}(\mathbf{d}, \hat{\mathbf{d}}) \text{ where } \hat{d}_{t,i} \propto \exp\left(-\tfrac{1}{\beta}(A\,\mathbf{w}_t)_i\right)$$

$$j_t \in \arg\max_j |(\mathbf{d}_t^\dagger A)_j|$$

$$\eta_t = \max\left\{0, \min\left\{1, \frac{\beta\,\mathbf{d}_t^\dagger A(\mathbf{e}^{j_t} - \mathbf{w}_t)}{\|A(\mathbf{e}^{j_t} - \mathbf{w}_t)\|_\infty^2}\right\}\right\}$$

$$\mathbf{w}_{t+1} = (1 - \eta_t)\mathbf{w}_t + \eta_t\,\mathbf{e}^{j_t}$$

A Boosting Algorithm

INITIALIZE: $\mathbf{w}_1 = \mathbf{0}, \quad \beta = \frac{\epsilon}{2\log(m)}$

FOR $t = 1, 2, \ldots, T$

$$\mathbf{d}_t = \underset{\mathbf{d}\in\mathbb{S}^m\cap C}{\operatorname{argmin}} \; D_{\mathrm{KL}}(\mathbf{d}, \hat{\mathbf{d}}) \text{ where } \hat{d}_{t,i} \propto \exp\left(-\tfrac{1}{\beta}(A\,\mathbf{w}_t)_i\right)$$

Similar to 'AdaBoost'

$$j_t \in \arg\max_j |(\mathbf{d}_t^\dagger A)_j|$$

$$\eta_t = \max\left\{0, \min\left\{1, \frac{\beta\,\mathbf{d}_t^\dagger A(\mathbf{e}^{j_t} - \mathbf{w}_t)}{\|A(\mathbf{e}^{j_t} - \mathbf{w}_t)\|_\infty^2}\right\}\right\}$$

$$\mathbf{w}_{t+1} = (1 - \eta_t)\mathbf{w}_t + \eta_t\,\mathbf{e}^{j_t}$$

A Boosting Algorithm

Entropic projection

INITIALIZE: $\mathbf{w}_1 = \mathbf{0}, \quad \beta = \frac{\epsilon}{2\log(m)}$

FOR $t = 1, 2, \ldots, T$

$$\mathbf{d}_t = \operatorname*{argmin}_{\mathbf{d} \in \mathbb{S}^m \cap C} D_{\mathrm{KL}}(\mathbf{d}, \hat{\mathbf{d}}) \text{ where } \hat{d}_{t,i} \propto \exp\left(-\tfrac{1}{\beta}(A\,\mathbf{w}_t)_i\right)$$

$$j_t \in \arg\max_j |(\mathbf{d}_t^\dagger A)_j|$$

$$\eta_t = \max\left\{0, \min\left\{1, \frac{\beta\,\mathbf{d}_t^\dagger A(\mathbf{e}^{j_t} - \mathbf{w}_t)}{\|A(\mathbf{e}^{j_t} - \mathbf{w}_t)\|_\infty^2}\right\}\right\}$$

$$\mathbf{w}_{t+1} = (1 - \eta_t)\mathbf{w}_t + \eta_t\,\mathbf{e}^{j_t}$$

A Boosting Algorithm

INITIALIZE: $\mathbf{w}_1 = \mathbf{0}, \quad \beta = \frac{\epsilon}{2\log(m)}$

FOR $t = 1, 2, \ldots, T$

Weal learner

$$\mathbf{d}_t = \operatorname*{argmin}_{\mathbf{d} \in \mathbb{S}^m \cap C} \; D_{\mathrm{KL}}(\mathbf{d}, \hat{\mathbf{d}}) \text{ where } \hat{d}_{t,i} \propto \exp\left(-\tfrac{1}{\beta}(A\,\mathbf{w}_t)_i\right)$$

$$j_t \in \arg\max_j |(\mathbf{d}_t^\dagger A)_j|$$

$$\eta_t = \max\left\{0, \min\left\{1, \frac{\beta\,\mathbf{d}_t^\dagger A(\mathbf{e}^{j_t} - \mathbf{w}_t)}{\|A(\mathbf{e}^{j_t} - \mathbf{w}_t)\|_\infty^2}\right\}\right\}$$

$$\mathbf{w}_{t+1} = (1 - \eta_t)\mathbf{w}_t + \eta_t\,\mathbf{e}^{j_t}$$

A Boosting Algorithm

INITIALIZE: $\mathbf{w}_1 = \mathbf{0}, \quad \beta = \frac{\epsilon}{2\log(m)}$

FOR $t = 1, 2, \ldots, T$

$$\mathbf{d}_t = \operatorname*{argmin}_{\mathbf{d} \in \mathbb{S}^m \cap C} D_{\mathrm{KL}}(\mathbf{d}, \hat{\mathbf{d}}) \text{ where } \hat{d}_{t,i} \propto \exp\left(-\tfrac{1}{\beta}(A\,\mathbf{w}_t)_i\right)$$

$$j_t \in \arg\max_j |(\mathbf{d}_t^\dagger A)_j|$$

$$\eta_t = \max\left\{0, \min\left\{1, \frac{\beta\,\mathbf{d}_t^\dagger A(\mathbf{e}^{j_t} - \mathbf{w}_t)}{\|A(\mathbf{e}^{j_t} - \mathbf{w}_t)\|_\infty^2}\right\}\right\}$$

learning rate

$$\mathbf{w}_{t+1} = (1 - \eta_t)\mathbf{w}_t + \eta_t\,\mathbf{e}^{j_t}$$

A Boosting Algorithm

Initialize: $\mathbf{w}_1 = \mathbf{0}, \quad \beta = \frac{\epsilon}{2\log(m)}$

For $t = 1, 2, \ldots, T$

$$\mathbf{d}_t = \operatorname*{argmin}_{\mathbf{d} \in \mathbb{S}^m \cap C} D_{\mathrm{KL}}(\mathbf{d}, \hat{\mathbf{d}}) \text{ where } \hat{d}_{t,i} \propto \exp\left(-\tfrac{1}{\beta}(A\,\mathbf{w}_t)_i\right)$$

$$j_t \in \arg\max_j |(\mathbf{d}_t^\dagger A)_j|$$

$$\eta_t = \max\left\{0, \min\left\{1, \frac{\beta\,\mathbf{d}_t^\dagger A(\mathbf{e}^{j_t} - \mathbf{w}_t)}{\|A(\mathbf{e}^{j_t} - \mathbf{w}_t)\|_\infty^2}\right\}\right\}$$

$$\mathbf{w}_{t+1} = (1 - \eta_t)\mathbf{w}_t + \eta_t\,\mathbf{e}^{j_t}$$

update

Convergence Rate

Theorem

- For any $m \times n$ matrix A over $[-1, 1]$
- For any relaxation set $C = \{\mathbf{d} : \|\mathbf{d}\| \leq \nu\}$
- The number of iterations required by the algorithm to find an ϵ-accurate solution is

$$T \leq O\left(\frac{\log(m)}{\epsilon^2}\right)$$

Remarks:

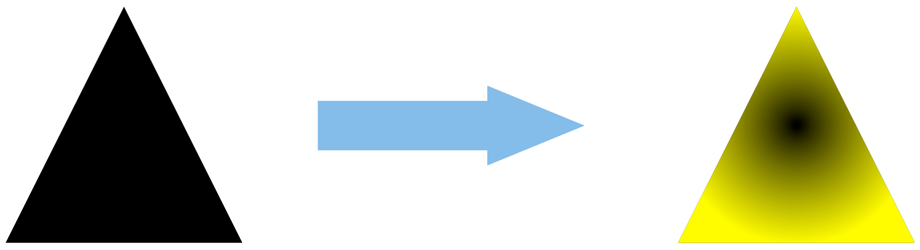
- Matches rate of AdaBoost$_\star$ [RW05] and SoftBoost [WLR06]
- Also bounds the sparseness of solution

Proof Technique

- **Step 1:** If loss function has β Lipschitz continuous derivative:

$$\epsilon_t - \epsilon_{t+1} \;\geq\; \eta\,\epsilon_t - \frac{2\,\eta^2}{\beta} \qquad \Rightarrow \qquad \epsilon_t \leq \frac{8}{\beta(t+1)}$$

- Proof uses duality
- **Step 2:** Approximate any 'soft-margin' loss by 'nicely behaved' loss
  - Domain of conjugate of the loss is a subset of the simplex
  - Add a bit relative entropy
  - Use infimal convolution theorem



Efficient Implementation

- The most expensive operations are the Entropic projection on C and the call to weak learner
- For $C = \{\mathbf{w} : \|\mathbf{w}\|_\infty \leq \nu\}$ projection can be performed in $O(m)$
- The trick: similar to median search
- Proof can be extended to approximated weak learners

Summary

- Weak Learnability = Linear Separability
- Relaxing separability using relaxed weak learnability
- ‘Algorithmic’ relaxations

- Current and Future Work
    - Use equivalence for generalization bounds ?
    - Other intuitive relaxations
    - Other algorithmic relaxations
    - Relation between L_1 and sparsity in a more general setting