MLSS 2010 Boosting Tutorial
Recent Advances in Boosting

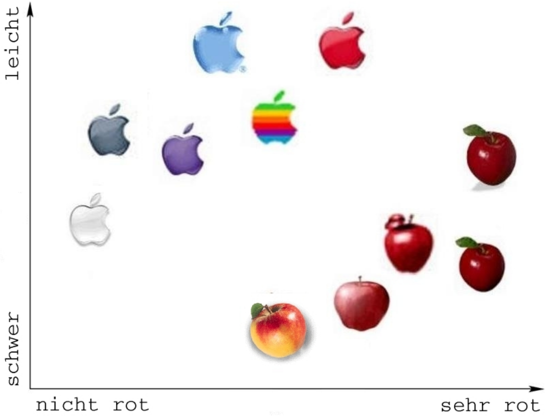
Part I: Entropy Regularized LPBoost
Part II: Boosting from an Optimization Perspective

Manfred K. Warmuth - UCSC
prepared jointly with S.V.N. Vishwanathan - Purdue
adapted from ICML 2009 tutorial

Updated October 4, 2010

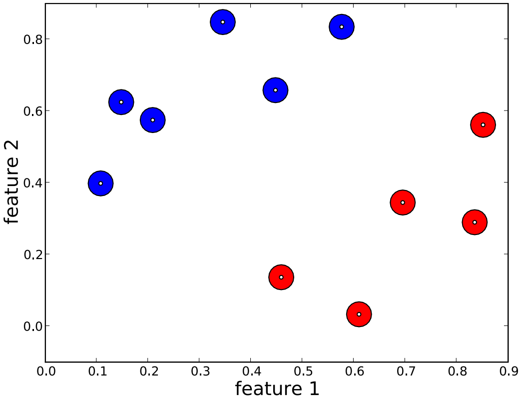
Outline

Setup for Boosting
[Giants of field: Schapire,Freund]



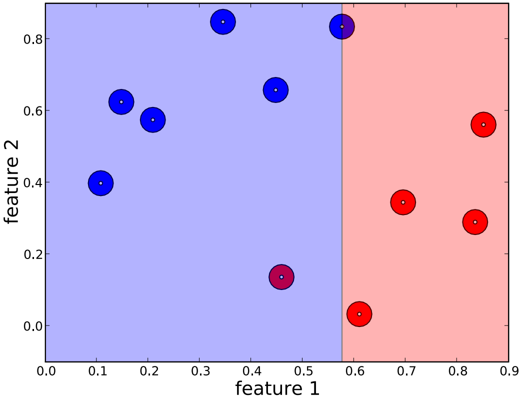
- examples: 11 apples
- +1 if artificial
  - 1 if natural
- goal: classification

Setup for Boosting



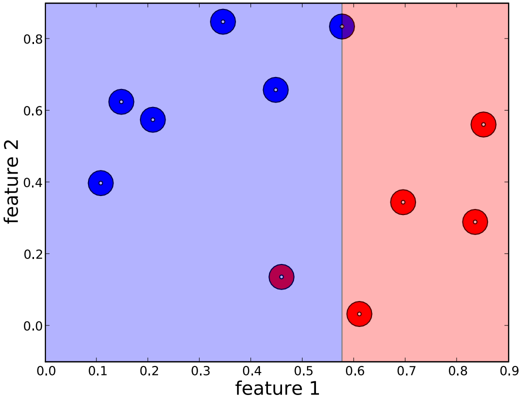
- $+1$/-1 examples
- weight $d_n \approx$ size
- separable

Weak hypotheses



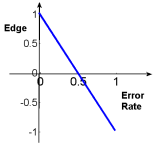
- weak hypotheses:
  decision stumps on two features
  one can't do it
- goal:
  find convex combination of weak hypotheses that classifies all

Boosting: 1st iteration



First hypothesis:

- error: $\frac{1}{11}$
- edge: $\frac{9}{11}$
  low error = high edge



edge = $1 - 2$ error

Accuracy on example / edge

	$y_n h(x_n) \quad := u_n$
perfect	$+1$
wrong	-1
neutral	0

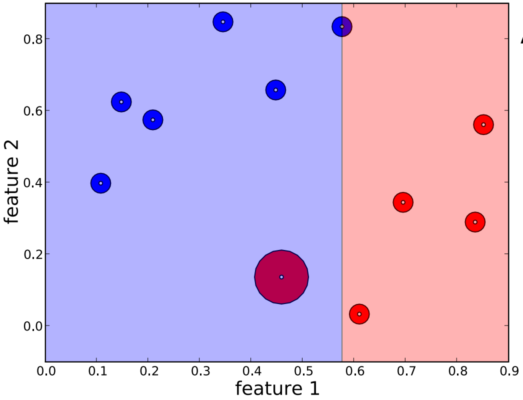
u_n is accuracy of h_n on example (x_n, y_n)
edge is accuracy on all examples weighted by distribution

$$\sum_n d_n u_n = \mathbf{d} \cdot \mathbf{u}$$

Weak hypothesis - column vector of accuracies

examples x_n	labels y_n	1st stump $h^1(x_n)$	accuracies u_n^1
	-1	-1	$\mathbf{1}$
	-1	-1	$\mathbf{1}$
	-1	-1	$\mathbf{1}$
	-1	1	$\mathbf{1}$
	1	1	$\mathbf{1}$
	1	1	$\mathbf{1}$
	1	1	$\mathbf{1}$
	1	-1	$-\mathbf{1}$

Update after 1st



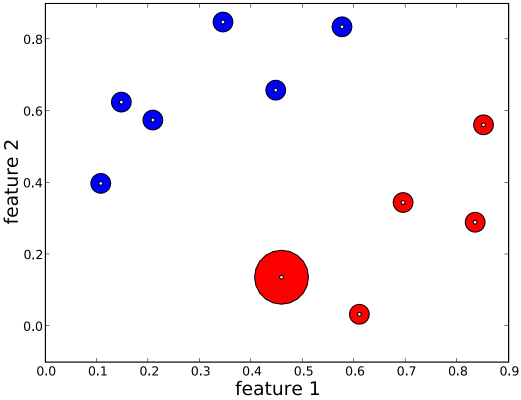
Misclassified examples

- increased weights

After update

- edge of hypothesis decreased

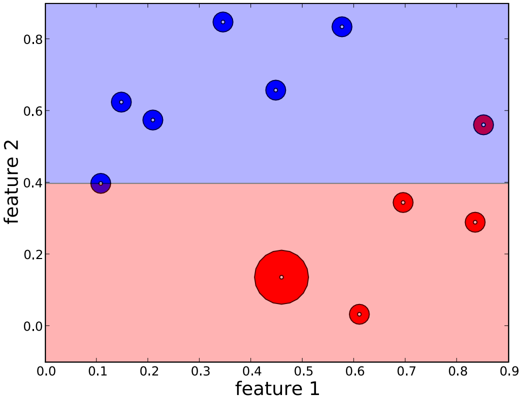
Before 2nd iteration



Hard examples

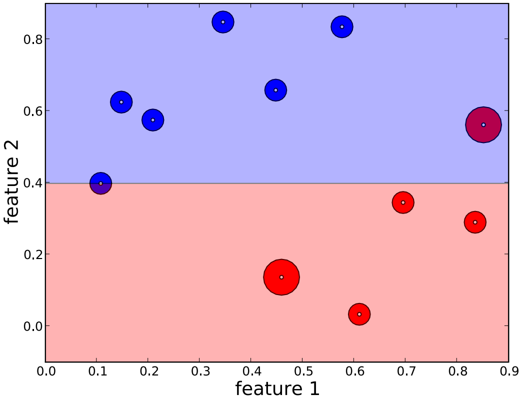
- high weight

Boosting: 2nd hypothesis



Pick hypotheses
with high (weighted) edge

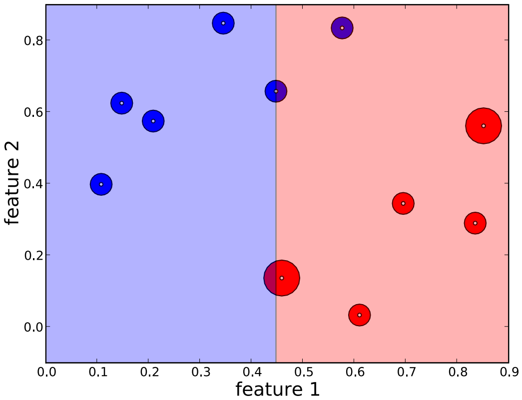
Update after 2nd



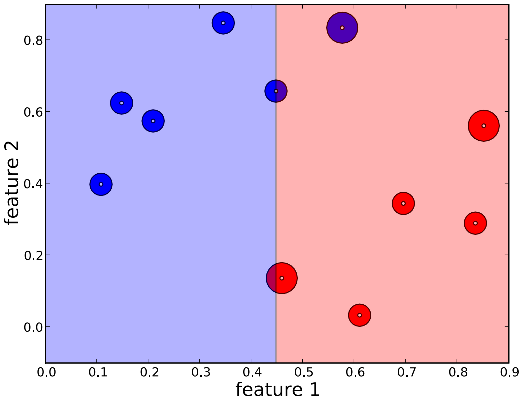
After update

- edges of all past hypotheses should be small

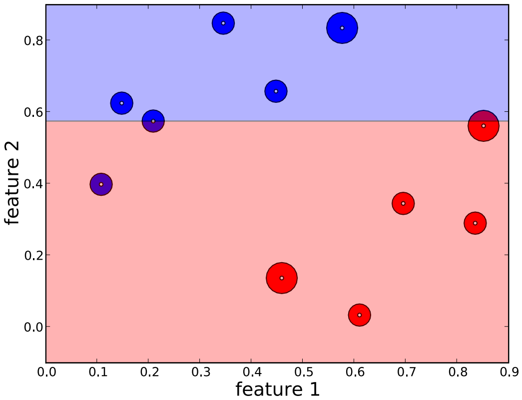
3rd hypothesis



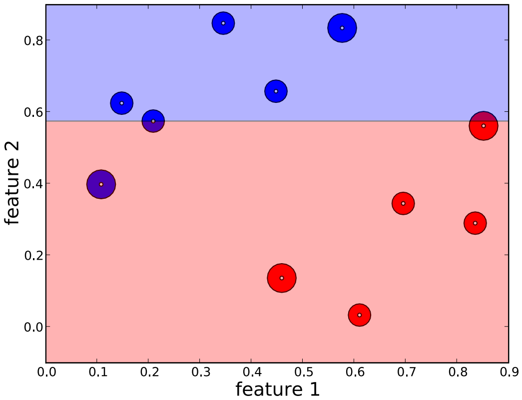
Update after 3rd



4th hypothesis

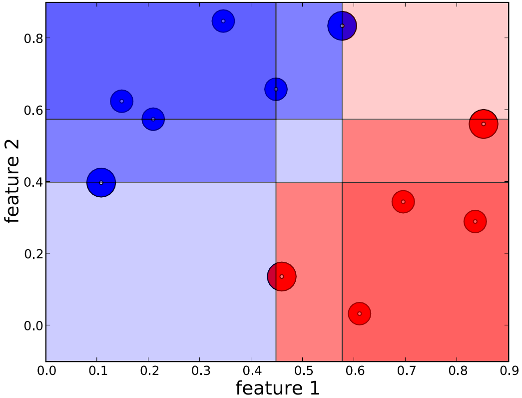


Update after 4th



Final convex combination of all hypotheses

Decision: $\sum_{t=1}^{T} w^t h^t(\mathbf{x}) \geq 0$?



Positive total weight - Negative total weight

Protocol of Boosting [FS97]

- Maintain distribution on N ± 1 labeled examples
- At iteration $t = 1, \ldots, T$:
  - Receive "weak" hypothesis h^t of high edge
  - Update $\mathbf{d}^{t-1}$ to $\mathbf{d}^t$ more weights on "hard" examples
- Output convex combination of the weak hypotheses $\sum_{t=1}^{T} w^t\, h^t(x)$

Two sets of weights:
- distribution on $\mathbf{d}$ on examples
- distribution on $\mathbf{w}$ on hypotheses

Recall data representation

	$y_n h^t(x_n) \quad := u_n^t$
perfect	$+1$
opposite	-1
neutral	0

examples x_n	labels y_n	$h^t(x_n)$	$\mathbf{u}^t$
	-1	-1	$\mathbf{1}$
	-1	-1	$\mathbf{1}$
	-1	-1	$\mathbf{1}$
	-1	1	$-\mathbf{1}$
	1	1	$\mathbf{1}$
	1	1	$\mathbf{1}$
	1	1	$\mathbf{1}$
	1	-1	$-\mathbf{1}$

Reweighing on hard examples?

Example AdaBoost:

$$d_n^t \sim d_n^{t-1} \, \exp(-w^t \, u_n^t)$$

where

- d_n^{t-1} are old weights
- u_n^t are accuracies
- w^t is "learning rate" / coefficient of hypothesis h^t

$$w^t = \frac{1}{2} \ln \frac{1 + \mathbf{d}^{t-1} \cdot \mathbf{u}^t}{1 - \mathbf{d}^{t-1} \cdot \mathbf{u}^t}$$

Outline

Edge vs. margin [Br99]

Edge of a hypothesis h^t for a distribution $\mathbf{d}$ on the examples

$$\underbrace{\sum_{n=1}^{N} \overbrace{u_n^t}^{\text{accuracy on example}} d_n}_{\text{weighted accuracy of hypothesis}} \qquad \mathbf{d} \in \mathcal{P}^N$$

Margin of example n for current hypothesis weighting $\mathbf{w}$

$$\underbrace{\sum_{t=1}^{T} \overbrace{u_n^t}^{\text{accuracy of example}} w_t}_{\text{weighted accuracy of example}} \qquad \mathbf{w} \in \mathcal{P}^T$$

Edge vs. margin [Br99]

Edge of a hypothesis h^t for a distribution **d** on the examples

$$\underbrace{\sum_{n=1}^{N} \overbrace{u_n^t}^{\text{accuracy on example}} d_n}_{\text{weighted accuracy of hypothesis}} \qquad \mathbf{d} \in \mathcal{P}^N$$

Margin of example n for current hypothesis weighting **w**

$$\underbrace{\sum_{t=1}^{T} \overbrace{u_n^t}^{\text{accuracy of example}} w_t}_{\text{weighted accuracy of example}} \qquad \mathbf{w} \in \mathcal{P}^T$$

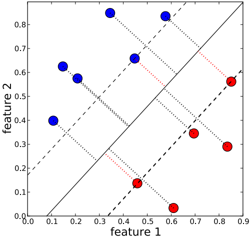
Objectives

Edge

- Edges of past hypotheses should be small after update
- Minimize maximum edge of past hypotheses

Margin

- Choose convex combination of weak hypotheses
  that maximizes the minimum margin



	Which margin?
SVM	2-norm (weights on examples)
Boosting	1-norm (weights on base hypotheses)

Connection between objectives?

Edge vs. margin

$$\text{min max edge} \quad = \quad \text{max min margin}$$

$$\min_{\mathbf{d}\in\mathcal{S}^N} \max_{q=1,2,\ldots,t-1} \underbrace{\mathbf{u}^q \cdot \mathbf{d}}_{\text{edge of hypothesis q}} \quad = \quad \max_{\mathbf{w}\in\mathcal{S}^{t-1}} \min_{n=1,2,\ldots,N} \underbrace{\sum_{q=1}^{t-1} u_n^q \, w^q}_{\text{margin of example } n}$$

Linear Programming duality

Boosting as zero-sum-game [FS97]

Rock, Paper, Scissors game

column player

$$
\begin{array}{cccccc}
 & & & R & P & S \\
 & & & w_1 & w_2 & w_3 \\
 & R & d_1 & 0 & 1 & -1 \\
\text{row player} & P & d_2 & -1 & 0 & 1 \\
 & S & d_3 & 1 & -1 & 0
\end{array}
$$

gain matrix

Single row is pure strategy of row player and $\mathbf{d}$ is mixed strategy

Single column is pure strategy of column player and $\mathbf{w}$ is mixed strategy

Row player minimizes
Column player maximizes

$$
\begin{aligned}
\text{payoff} &= \mathbf{d}^T \mathbf{U}\, \mathbf{w} \\
&= \sum_{i,j} d_i U_{i,j} \mathbf{w}_j
\end{aligned}
$$

Optimum strategy

$$\begin{array}{ccc|ccc}
 & & & R & P & S \\
 & & & w_1 & w_2 & w_3 \\
 & & & .33 & .33 & .33 \\
R & d_1 & .33 & 0 & 1 & -1 \\
P & d_2 & .33 & -1 & 0 & 1 \\
S & d_3 & .33 & 1 & -1 & 0
\end{array}$$

- Min-max theorem:

$$\begin{aligned}
& \min_{d} \max_{\mathbf{w}} \mathbf{d}^T \mathbf{U}\mathbf{w} = \min_{d} \max_{j} \mathbf{d}^T \, \mathbf{U} e_j \\
= \quad & \max_{\mathbf{w}} \min_{d} \; \mathbf{d}^T \mathbf{U}\mathbf{w} = \max_{\mathbf{w}} \min_{i} \; e_i \mathbf{U}\mathbf{w} \\
= \quad & \text{value of the game (0 in example)}
\end{aligned}$$

e_j is pure strategy

Connection to Boosting?

- Rows are the examples
- Columns $\mathbf{u}^q$ encode weak hypothesis h^q
- Row sum: margin of example
- Column sum: edge of weak hypothesis
- Value of game:

$$\text{min max edge} = \text{max min margin}$$

Van Neumann's Minimax Theorem

Edges/margins

			R	P	S	
			w_1	w_2	w_3	margin
			.33	.33	.33	
R	d_1	.33	0	1	1	0
P	d_2	.33	-1	0	1	0 min
S	d_3	.33	1	-1	-1	0
	edge		0	0	0	
				max		

value of game 0

New column added: boosting

			R	P	S		
			w_1	w_2	w_3	w_4	margin
			.44	0	.22	.33	
R	d_1	.22	0	1	-1	1	.11
P	d_2	.33	-1	0	1	1	.11 min
S	d_3	.44	1	-1	0	-1	.11
	edge		.11	-.22	.11	.11	
				max			

Value of game **increases** from 0 to .11

Row added: on-line learning

			R	P	S	
			w_1	w_2	w_3	margin	
			.33	.44	.22		
R	d_1	0	0	1	-1	.22	
P	d_2	.22	-1	0	1	-.11	min
S	d_3	.44	1	-1	0	-.11	
	d_4	.33	-1	1	-1	-.11	
	edge		-.11	-.11	-.11		
				max			

Value of game **decreases** from 0 to -.11

Boosting: maximize margin incrementally

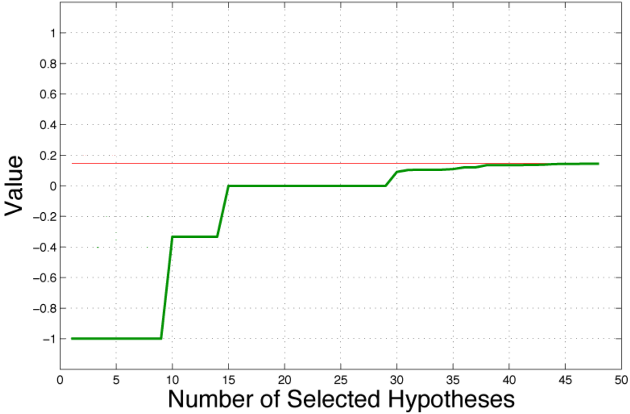
$$\begin{array}{c|c} & w_1^1 \\ \hline d_1^1 & 0 \\ d_2^1 & 1 \\ d_3^1 & -1 \end{array} \qquad \begin{array}{c|cc} & w_1^2 & w_2^2 \\ \hline d_1^2 & 0 & -1 \\ d_2^2 & 1 & 0 \\ d_3^2 & -1 & 1 \end{array} \qquad \begin{array}{c|ccc} & w_1^3 & w_2^3 & w_3^3 \\ \hline d_1^3 & 0 & -1 & 1 \\ d_2^3 & 1 & 0 & -1 \\ d_3^3 & -1 & 1 & 0 \end{array}$$

iteration 1 — iteration 2 — iteration 3

- In each iteration solve optimization problem to update d
- Column player / oracle provides new hypothesis
- Boosting is column generation method in **d** domain and coordinate descent in **w** domain

Boosting = greedy method for increasing margin

Converges to optimum margin w.r.t. all hypotheses



Want small number of iterations

Assumption on next weak hypothesis

For current weighting of examples,
oracle returns hypothesis of edge $\geq g$

Goal

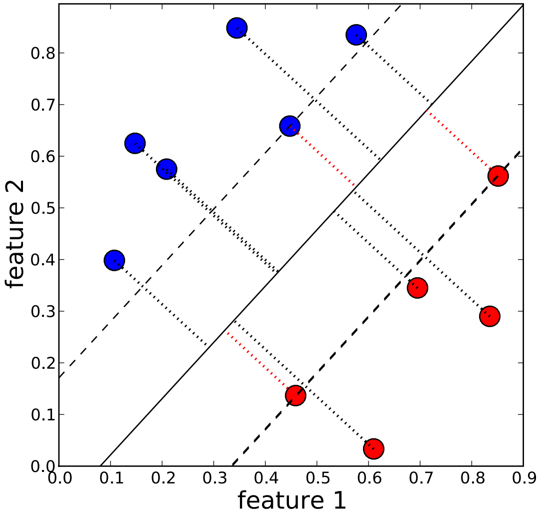
- For given ϵ, produce convex combination of weak hypotheses with soft margin $\geq g - \epsilon$
- Number of iterations $O(\frac{\log N}{\epsilon^2})$

Recall min max thm

$$\min_{\mathbf{d}\in\mathcal{S}^N} \max_{q=1,2,\ldots,t} \underbrace{\mathbf{u}^q \cdot \mathbf{d}}_{\text{edge of hypothesis q}}$$

$$= \max_{\mathbf{w}\in\mathcal{S}^t} \min_{n=1,2,\ldots,N} \underbrace{\left(\sum_{q=1}^{t} u_n^q \, w^q\right)}_{\text{margin of example } n}$$

Visualizing the margin



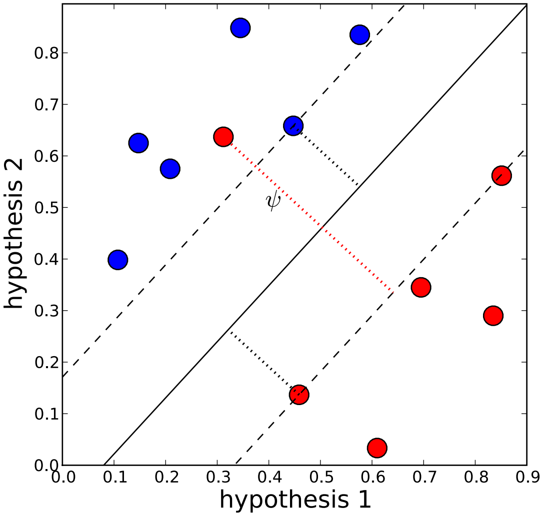
Min max thm - inseparable case

Slack variables in **w** domain = capping in **d** domain

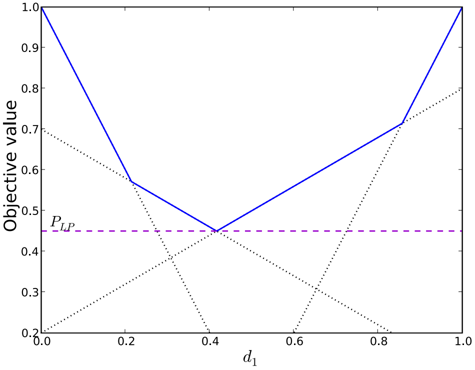
$$\min_{\mathbf{d}\in\mathcal{S}^N,\mathbf{d}\le\frac{1}{\nu}\mathbf{1}} \; \max_{q=1,2,\ldots,t} \; \underbrace{\mathbf{u}^q\cdot\mathbf{d}}_{\text{edge of hypothesis q}}$$

$$= \max_{\mathbf{w}\in\mathcal{S}^t,\boldsymbol{\psi}\ge\mathbf{0}} \; \min_{n=1,2,\ldots,N} \underbrace{\left(\sum_{q=1}^{t} u_n^q\, w^q + \psi_n\right)}_{\text{soft margin of example } n} - \frac{1}{\nu}\sum_{n=1}^{N}\psi_n$$

Visualizing the soft margin



LPBoost



Choose distribution that minimizes the maximum edge of current hypotheses by solving:

$$\underbrace{\min_{\sum_n d_n = 1, \mathbf{d} \leq \frac{1}{\nu}\mathbf{1}} \; \max_{q=1,2,\ldots,t} \; \mathbf{u}^q \cdot \mathbf{d}}_{P_{LP}^t}$$

All weight is put on examples with minimum soft margin

Outline

Entropy Regularized LPBoost

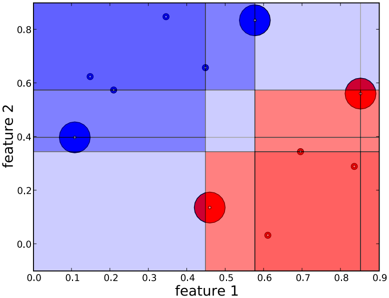
$$\min_{\sum_n d_n = 1, \mathbf{d} \leq \frac{1}{\nu}\mathbf{1}} \max_{q=1,2,\dots,t} \mathbf{u}^q \cdot \mathbf{d} + \frac{1}{\eta}\Delta(\mathbf{d}, \mathbf{d}^0)$$

- $$\mathbf{d}_n = \frac{\exp^{-\eta \text{ soft margin of example } n}}{Z}$$ "soft min"

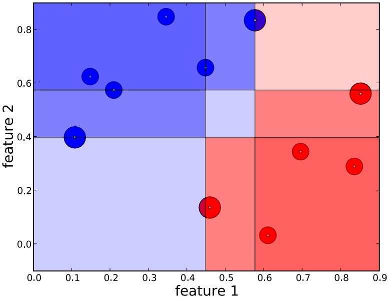
- Form of weights first in ν-Arc algorithm [RSS+00]
- Regularization in **d** domain makes problem strongly convex
- Gradient of dual Lipschitz continuous in **w** [e.g. HL93,RW97]

The effect of entropy regularization

Different distribution on the examples



LPBoost: lots of zeros / brittle



ERLPBoost: smoother

Outline

AdaBoost revisited [FS97]

$$d_n^t := \frac{d_n^{t-1}\exp(-w^t u_n^t)}{\sum_{n'} d_{n'}^{t-1}\exp(-w^t u_{n'}^t)},$$

where w^t s.t. $\underbrace{-\ln \sum_n d_n^{t-1}\exp(-w\ u_n^t)}_{\text{dual objective}}$ is minimized

$$\text{Choose } w \text{ s.t. } \frac{\partial - \ln \sum_{n'} d_{n'}^{t-1}\exp(-w\ u_{n'}^t)}{\partial w} = \mathbf{u}^t \cdot \mathbf{d}^t(w) = 0$$

- Easy to implement
- Adjusts distribution so that edge of last hypothesis is zero
- When $h_n^t \in \{-1, +1\}$ then $w^t = \frac{1}{2}\ln\frac{1+\mathbf{d}^{t-1}\cdot\mathbf{u}^t}{1-\mathbf{d}^{t-1}\cdot\mathbf{u}^t}$
  when $h_n^t \in [-1, +1]$ then line search
- Gets within half of the optimal hard margin [RSD07]
  but only in the limit

Corrective versus totally corrective

Processing last hypothesis versus all past hypotheses

Corrective	Totally Corrective
AdaBoost	LPBoost
LogitBoost	TotalBoost
AdaBoost*	SoftBoost
SS,Colt08	ERLPBoost

From AdaBoost to ERLPBoost

AdaBoost (as interpreted in [KW99,La99])

Primal:

$$\min_{\mathbf{d}} \quad \Delta(\mathbf{d}, \mathbf{d}^{t-1})$$
$$\text{s.t.} \quad \mathbf{d} \cdot \mathbf{u}^t = 0, \ \|\mathbf{d}\|_1 = 1$$

Dual:

$$\max_{\mathbf{w}} \quad -ln \sum_n d_n^{t-1} \exp(-u_n^t w)$$

Achieves half of optimum hard margin in the limit

AdaBoost* [RW05]

Primal:

$$\min_{\mathbf{d}} \quad \Delta(\mathbf{d}, \mathbf{d}^{t-1})$$
$$\text{s.t.} \quad \mathbf{d} \cdot \mathbf{u}^t \leq \gamma_t,$$
$$\|\mathbf{d}\|_1 = 1$$

Dual:

$$\max_{\mathbf{w}} \ -ln \sum_n d_n^{t-1} \exp(-u_n^t w) - \gamma_t w$$
$$\text{s.t.} \quad w \geq 0$$

where edge bound γ_t is adjusted downward by a heuristic

Good iteration bound for reaching optimum hard margin

SoftBoost [WGR07]

Primal:

$$\begin{aligned} \min_{\mathbf{d}} \quad & \Delta(\mathbf{d}, \mathbf{d}^0) \\ \text{s.t.} \quad & \|\mathbf{d}\|_1 = 1,\ \mathbf{d} \leq \tfrac{1}{\nu}\mathbf{1} \\ & \mathbf{d} \cdot \mathbf{u}^q \leq \gamma_t, \\ & 1 \leq q \leq t \end{aligned}$$

Dual:

$$\begin{aligned} \min_{\mathbf{w},\boldsymbol{\psi}} \quad & -\ln \sum_n \mathbf{d}^0_n \exp(-\eta \sum_{q=1}^t u^q_n w^q \\ & -\eta\psi_n) - \tfrac{1}{\nu}\|\boldsymbol{\psi}\|_1 - \gamma_t \|\mathbf{w}\|_1 \\ \text{s.t.} \quad & \mathbf{w} \geq 0,\ \boldsymbol{\psi} \geq 0 \end{aligned}$$

where edge bound γ_t is adjusted downward by a heuristic

Good iteration bound for reaching soft margin

ERLPBoost [WGV08]

Primal:

$$\begin{aligned} \min_{\mathbf{d},\gamma} \quad & \gamma + \tfrac{1}{\eta}\Delta(\mathbf{d}, \mathbf{d}^0) \\ \text{s.t.} \quad & \|\mathbf{d}\|_1 = 1,\ \mathbf{d} \leq \tfrac{1}{\nu}\mathbf{1} \\ & \mathbf{d} \cdot \mathbf{u}^q \leq \gamma, \\ & 1 \leq q \leq t \end{aligned}$$

Dual:

$$\begin{aligned} \min_{\mathbf{w},\boldsymbol{\psi}} \quad & -\tfrac{1}{\eta} \ln \sum_n \mathbf{d}^0_n \exp(-\eta \sum_{q=1}^t u^q_n w^q \\ & -\eta\psi_n) - \tfrac{1}{\nu}\|\boldsymbol{\psi}\|_1 \\ \text{s.t.} \quad & \mathbf{w} \geq 0,\ \|\mathbf{w}\|_1 = 1,\ \boldsymbol{\psi} \geq 0 \end{aligned}$$

where for the iteration bound η is fixed to $\max(\frac{2}{\epsilon} \ln \frac{N}{\nu}, \frac{1}{2})$

Good iteration bound for reaching soft margin

Corrective ERLPBoost [SS08]

Primal:

$$\begin{array}{ll} \min\limits_{\mathbf{d}} & \sum_{q=1}^{t} w^q(\mathbf{u}^q \cdot \mathbf{d}) + \frac{1}{\eta}\Delta(\mathbf{d}, \mathbf{d}^0) \\ \text{s.t.} & \|\mathbf{d}\|_1 = 1,\ \mathbf{d} \le \frac{1}{\nu}\mathbf{1} \end{array}$$

Dual:

$$\begin{array}{ll} \min\limits_{\boldsymbol{\psi}} & -\frac{1}{\eta}\ln\sum\limits_{n} \mathbf{d}_n^0 \exp(-\eta \sum\limits_{q=1}^{t} u_n^q w^q - \eta\psi_n) - \frac{1}{\nu}\|\boldsymbol{\psi}\|_1 \\ \text{s.t.} & \boldsymbol{\psi} \ge 0 \end{array}$$

where for the iteration bound η is fixed to $\max(\frac{2}{\epsilon}\ln\frac{N}{\nu}, \frac{1}{2})$

Good iteration bound for reaching soft margin

Iteration bounds

Corrective	Totally Corrective
AdaBoost	LPBoost
LogitBoost	TotalBoost
AdaBoost*	SoftBoost
SS,Colt08	ERLPBoost

- Strong oracle: returns hypothesis with maximum edge
- Weak oracle: returns hypothesis with edge $\geq g$

- In $O(\frac{\log \frac{N}{\nu}}{\epsilon^2})$ iterations within ϵ of maximum soft margin for strong oracle or within ϵ of g for weak oracle
- Ditto for hard margin case
- When $g > 0$, in $O(\frac{\log N}{g^2})$ iterations consistency with weak oracle

LPBoost may require $\Omega(N)$ iterations

		w_1	w_2	w_3	w_4	w_5	margin
		0	0	0	0	0	
d_1	.125	+1	-.95	-.93	-.91	-.99	–
d_2	.125	+1	-.95	-.93	-.91	-.99	–
d_3	.125	+1	-.95	-.93	-.91	-.99	–
d_4	.125	+1	-.95	-.93	-.91	-.99	–
d_5	.125	-.98	+1	-.93	-.91	+.99	–
d_6	.125	-.97	-.96	+1	-.91	+.99	–
d_7	.125	-.97	-.95	-.94	+1	+.99	–
d_8	.125	-.97	-.95	-.93	-.92	+.99	–
edge		.0137	-.7075	-.6900	-.6725	.0000	
value	**-1**						

LPBoost may require $\Omega(N)$ iterations

		w_1	w_2	w_3	w_4	w_5	margin
		1	0	0	0	0	
d_1	0	+1	-.95	-.93	-.91	-.99	1
d_2	0	+1	-.95	-.93	-.91	-.99	1
d_3	0	+1	-.95	-.93	-.91	-.99	1
d_4	0	+1	-.95	-.93	-.91	-.99	1
d_5	1	-.98	+1	-.93	-.91	+.99	-.98
d_6	0	-.97	-.96	+1	-.91	+.99	-.97
d_7	0	-.97	-.95	-.94	+1	+.99	-.97
d_8	0	-.97	-.95	-.93	-.92	+.99	-.97
edge		-.98	1	-.93	-.91	.99	
value	-1	-.98					

LPBoost may require $\Omega(N)$ iterations

		w_1	w_2	w_3	w_4	w_5	margin
		0	1	0	0	0	
d_1	0	+1	-.95	-.93	-.91	-.99	-.95
d_2	0	+1	-.95	-.93	-.91	-.99	-.95
d_3	0	+1	-.95	-.93	-.91	-.99	-.95
d_4	0	+1	-.95	-.93	-.91	-.99	-.95
d_5	0	-.98	+1	-.93	-.91	+.99	1
d_6	1	-.97	-.96	+1	-.91	+.99	-.96
d_7	0	-.97	-.95	-.94	+1	+.99	-.95
d_8	0	-.97	-.95	-.93	-.92	+.99	-.95
edge		-.97	-.96	1	-.91	.99	
value	-1	-.98	-.96				

LPBoost may require $\Omega(N)$ iterations

		w_1	w_2	w_3	w_4	w_5	margin
		0	0	1	0	0	
d_1	0	+1	-.95	-.93	-.91	-.99	-.93
d_2	0	+1	-.95	-.93	-.91	-.99	-.93
d_3	0	+1	-.95	-.93	-.91	-.99	-.93
d_4	0	+1	-.95	-.93	-.91	-.99	-.93
d_5	0	-.98	+1	-.93	-.91	+.99	-.93
d_6	0	-.97	-.96	+1	-.91	+.99	1
d_7	1	-.97	-.95	-.94	+1	+.99	-.94
d_8	0	-.97	-.95	-.93	-.92	+.99	-.93
edge		-.97	-.95	-.94	1	.99	
value	-1	-.98	-.96	-.94			

LPBoost may require $\Omega(N)$ iterations

		w_1	w_2	w_3	w_4	w_5	margin
		0	0	0	1	0	
d_1	0	+1	-.95	-.93	-.91	-.99	-.91
d_2	0	+1	-.95	-.93	-.91	-.99	-.91
d_3	0	+1	-.95	-.93	-.91	-.99	-.91
d_4	0	+1	-.95	-.93	-.91	-.99	-.91
d_5	0	-.98	+1	-.93	-.91	+.99	-.91
d_6	0	-.97	-.96	+1	-.91	+.99	-.91
d_7	0	-.97	-.95	-.94	+1	+.99	1
d_8	1	-.97	-.95	-.93	-.92	+.99	-.92
edge		-.97	-.95	-.94	-.92	.99	
value	-1	-.98	-.96	-.94	-.92		

LPBoost may require $\Omega(N)$ iterations

		w_1	w_2	w_3	w_4	w_5	margin
		.5	.0026	0	0	.4975	
d_1	.497	+1	-.95	-.93	-.91	-.99	.0051
d_2	0	+1	-.95	-.93	-.91	-.99	.0051
d_3	0	+1	-.95	-.93	-.91	-.99	.0051
d_4	0	+1	-.95	-.93	-.91	-.99	.0051
d_5	0	-.98	+1	-.93	-.91	+.99	.0051
d_6	.490	-.97	-.96	+1	-.91	+.99	.0051
d_7	0	-.97	-.95	-.94	+1	+.99	.0051
d_8	.013	-.97	-.95	-.93	-.92	+.99	.0051
edge		.0051	.0051	.9055	.9100	.0051	
value	-1	-.98	-.96	-.94	-.92	.0051	

No ties!

LPBoost may return bad final hypothesis

How good is the master hypothesis returned by LPBoost compared to the best possible convex combination of hypotheses?

Any linearly separable dataset can be reduced to a dataset on which LPBoost misclassifies all examples by

- adding a bad hypothesis
- adding a bad example

Before adding a bad example

		w_1	w_2	w_3	w_4	w_5	margin
		.5	.0026	0	0	.4975	
d_1	.497	+1	-.95	-.93	-.91	-.99	.0051
d_2	0	+1	-.95	-.93	-.91	-.99	.0051
d_3	0	+1	-.95	-.93	-.91	-.99	.0051
d_4	0	+1	-.95	-.93	-.91	-.99	.0051
d_5	0	-.98	+1	-.93	-.91	+.99	.0051
d_6	.490	-.97	-.96	+1	-.91	+.99	.0051
d_7	0	-.97	-.95	-.94	+1	+.99	.0051
d_8	.013	-.97	-.95	-.93	-.92	+.99	.0051
d_9		-.03	-.03	-.03	-.03	-.03	
edge		.0051	.0051	.9055	.9100	.0051	
value							.0051

After adding a bad example

		w_1 .4445	w_2 .0534	w_3 .0543	w_4 .0561	w_5 .3917	margin
d_1	0	+1	-.95	-.93	-.91	-.99	−.0956
d_2	0	+1	-.95	-.93	-.91	-.99	−.0956
d_3	0	+1	-.95	-.93	-.91	-.99	−.0956
d_4	0	+1	-.95	-.93	-.91	-.99	−.0956
d_5	0	-.98	+1	-.93	-.91	+.99	−.0959
d_6	0	-.97	-.96	+1	-.91	+.99	−.0913
d_7	0	-.97	-.95	-.94	+1	+.99	−.0891
d_8	0	-.97	-.95	-.93	-.92	+.99	−.1962
d_9	1	**-.03**	**-.03**	**-.03**	**-.03**	**-.03**	−.03
edge		−.03	−.03	−.03	−.03	−.03	
value							-.03

Synopsis

- LPBoost often unstable
- For safety, add relative entropy regularization
- Corrective algs
  - Sometimes easy to code
  - Fast per iteration
- Totally corrective algs
  - Smaller number of iterations
  - Faster overall time when ϵ small
- Weak versus strong oracle makes a big difference in practice

$O(\frac{\log N}{\epsilon^2})$ iteration bounds

Good

- Bound is major design tool
- Any reasonable Boosting algorithm should have this bound

Bad

- Bound is weak

	$\frac{\ln N}{\epsilon^2} \geq N$
$\epsilon = .01$	$N \leq 1.2\ 10^5$
$\epsilon = .001$	$N \leq 1.7\ 10^7$

- Why are totally corrective algorithms much better in practice?

Lower bounds on the number of iterations

- Majority of $\Omega(\frac{\log N}{g^2})$ hypotheses for achieving consistency with weak oracle of guarantee g [Fr95]

- Easy: $\Omega(\frac{1}{\epsilon^2})$ iteration bound for getting within ϵ of hard margin with strong oracle
- Harder: $\Omega(\frac{\log N}{\epsilon^2})$ iteration bound
  Only for weak oracle [Ne83]

Outline

Conclusion

- Adding relative entropy regularization of LPBoost leads to good boosting alg.
- Boosting is instantiation of MaxEnt and MinxEnt principles [Jaines 57,Kullback 59]
- Relative entropy regularization smoothes one-norm regularization

Open

- When hypotheses have one-sided error then $O(\frac{\log N}{\epsilon})$ iterations suffice [As00,HW03]
- Does ERLPBoost have $O(\frac{\log N}{\epsilon})$ bound when hypotheses one-sided?
- Replace geometric optimizers by entropic ones
- Compare ours with Freund's algorithms that don't just cap, but forget examples

Manfred's acknowledgments

- Rob Schapire and Yoav Freund for pioneering Boosting
- Gunnar Rätsch for bringing in optimization
- Karen Glocer for helping with figures and plots

MLSS 2010 Boosting Tutorial
Recent Advances in Boosting

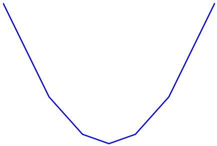
Part I: Entropy Regularized LPBoost
Part II: Boosting from an Optimization Perspective

Manfred K. Warmuth - UCSC
prepared jointly with S.V.N. Vishwanathan - Purdue
adapted from ICML 2009 tutorial

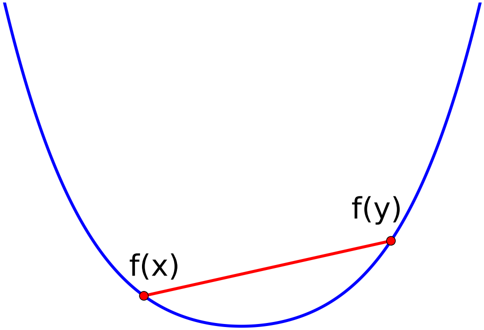
Updated October 4, 2010

Minimizing the Max Edge is a Convex Optimization Problem

$$\min_{\mathbf{d}\in\mathcal{S}^N} \underbrace{\max_{\mathbf{u}\in\mathcal{U}} \langle \mathbf{u}, \mathbf{d}\rangle}_{f(\mathbf{d})}$$



Convex Function - Common Definition

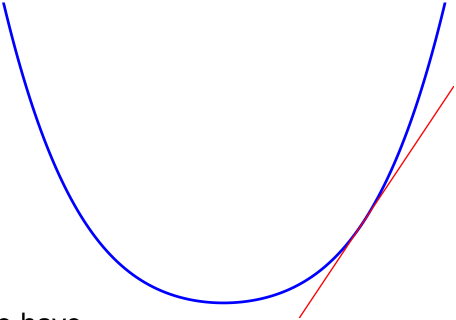


A function f is convex if, and only if, for all x, y and $\alpha \in (0, 1)$ we have

$$f(\alpha x + (1-\alpha)y) \leq \alpha f(x) + (1-\alpha)f(y)$$

A Key Property of Convex Functions

The First Order Taylor approximation always lower bounds the function

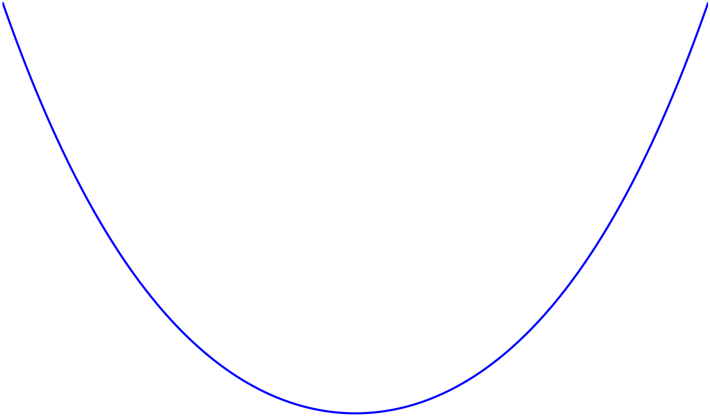


For any x and y we have

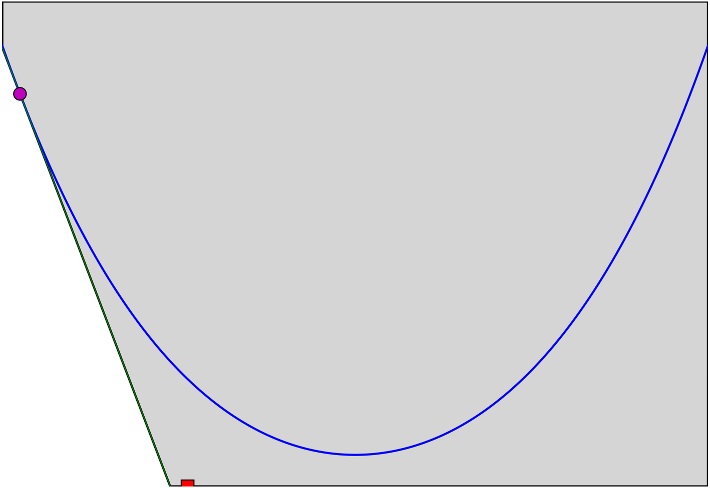
$$f(x) \geq f(y) + \langle x - y, \nabla f(y) \rangle$$

$$\Leftrightarrow \underbrace{f(x) - f(y) + \langle x - y, \nabla f(y) \rangle}_{\text{Bregman divergence } \Delta_f(x,y)} \geq 0$$

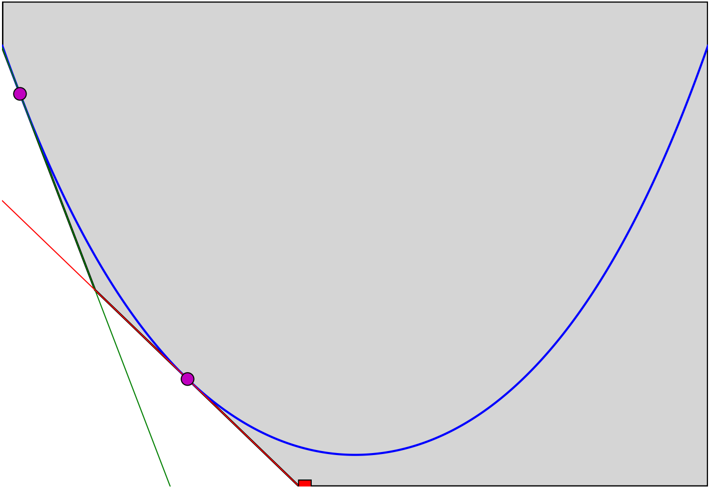
Cutting Plane Methods [Kelly 60]



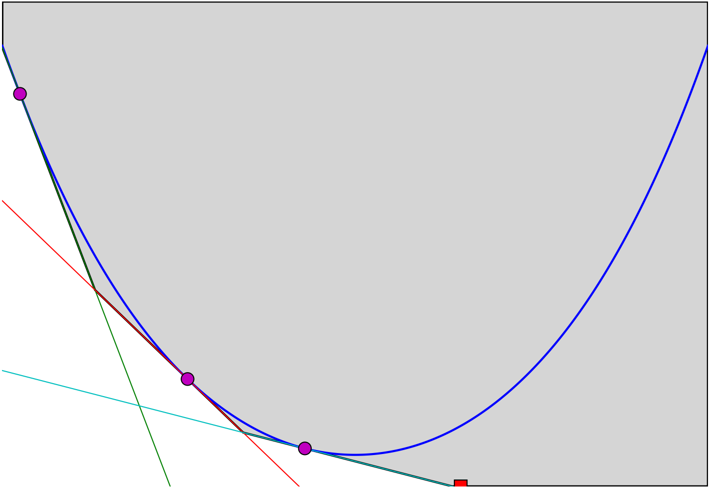
Cutting Plane Methods [Kelly 60]



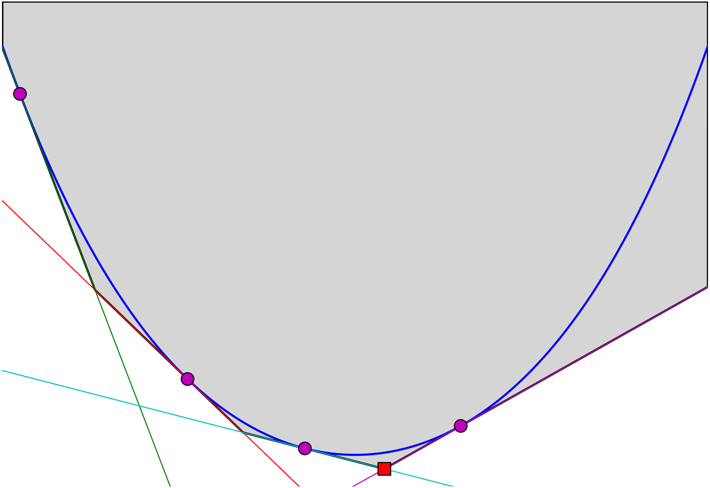
Cutting Plane Methods [Kelly 60]



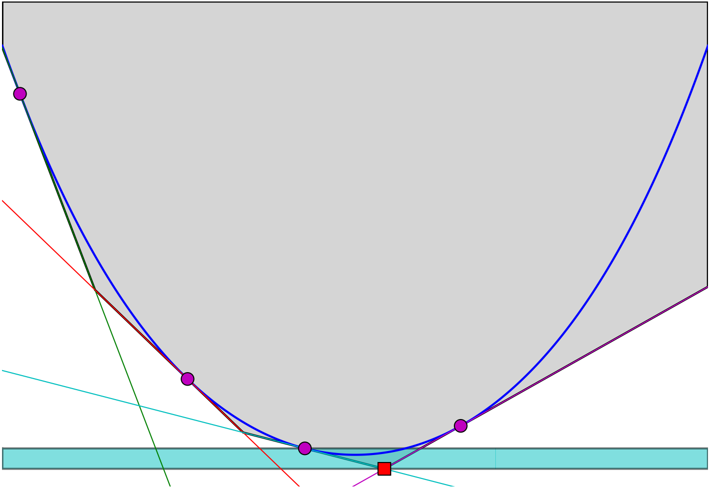
Cutting Plane Methods [Kelly 60]



Cutting Plane Methods [Kelly 60]



Monitoring Convergence



In a Nutshell

- Cutting Plane Methods work by forming the piecewise linear lower bound

$$f(x) \geq f_t^{\mathrm{CP}}(x) := \max_{1 \leq i \leq t} \{ f(x_{i-1}) + \langle x - x_{i-1}, s_i \rangle \}.$$

  where s_i denote gradients $\nabla f(x_{i-1})$.

- At iteration t the set $\{x_i\}_{i=0}^{t-1}$ is augmented by

$$x_t := \operatorname*{argmin}_x f_t^{\mathrm{CP}}(x).$$

- Stop when the duality gap

$$\epsilon_t := \min_{0 \leq i \leq t} f(x_i) - f_t^{\mathrm{CP}}(x_t)$$

  falls below a pre-specified threshold ϵ.

In a Nutshell

- Cutting Plane Methods work by forming the piecewise linear lower bound

$$f(x) \geq f_t^{\mathrm{CP}}(x) := \max_{1 \leq i \leq t} \{ f(x_{i-1}) + \langle x - x_{i-1}, s_i \rangle \}.$$

  where s_i denote gradients $\nabla f(x_{i-1})$.
- At iteration t the set $\{x_i\}_{i=0}^{t-1}$ is augmented by

$$x_t := \operatorname*{argmin}_x f_t^{\mathrm{CP}}(x).$$

- Stop when the duality gap

$$\epsilon_t := \min_{0 \leq i \leq t} f(x_i) - f_t^{\mathrm{CP}}(x_t)$$

  falls below a pre-specified threshold ϵ.

In a Nutshell

- Cutting Plane Methods work by forming the piecewise linear lower bound

$$f(x) \geq f_t^{\mathrm{CP}}(x) := \max_{1 \leq i \leq t} \{ f(x_{i-1}) + \langle x - x_{i-1}, s_i \rangle \}.$$

  where s_i denote gradients $\nabla f(x_{i-1})$.
- At iteration t the set $\{x_i\}_{i=0}^{t-1}$ is augmented by

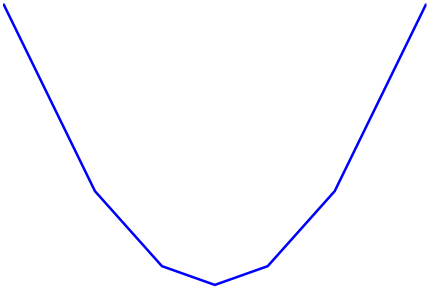
$$x_t := \operatorname*{argmin}_x f_t^{\mathrm{CP}}(x).$$

- Stop when the duality gap

$$\epsilon_t := \min_{0 \leq i \leq t} f(x_i) - f_t^{\mathrm{CP}}(x_t)$$

  falls below a pre-specified threshold ϵ.

What if the Function is NonSmooth?

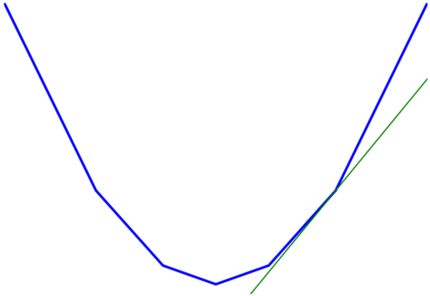


The piecewise linear function

$$f(x) := \max_i \langle u_i, x \rangle$$

is convex but not differentiable at the kinks!

Subgradients to the Rescue

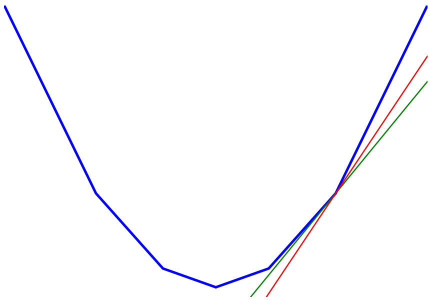


A subgradient at y is any vector μ which satisfies

$$f(x) \geq f(y) + \langle x - y, \mu \rangle \text{ for all } x$$

Set of all subgradients is denoted as $\partial f(y)$

Subgradients to the Rescue

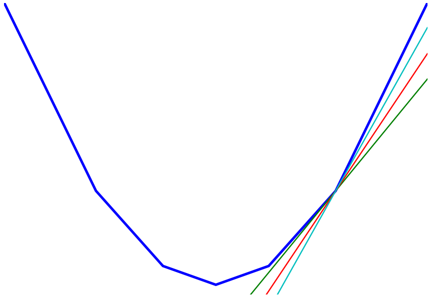


A subgradient at y is any vector μ which satisfies

$$f(x) \geq f(y) + \langle x - y, \mu \rangle \text{ for all } x$$

Set of all subgradients is denoted as $\partial f(y)$

Subgradients to the Rescue



A subgradient at y is any vector μ which satisfies

$$f(x) \geq f(y) + \langle x - y, \mu \rangle \text{ for all } x$$

Set of all subgradients is denoted as $\partial f(y)$

Good News!

Cutting Plane Methods work with subgradients

Just choose an arbitrary one

Boosting as an Optimization Problem

- Minimizing the maximum edge

$$\min_{\mathbf{d}\in\mathcal{S}^N} \underbrace{\max_{\mathbf{u}\in\mathcal{U}} \langle \mathbf{u}, \mathbf{d} \rangle}_{f(\mathbf{d})}$$

  is a convex optimization problem.

- Subgradient: $\partial f(\mathbf{d}) = \mathrm{argmax}_{\mathbf{u}\in\mathcal{U}} \langle \mathbf{u}, \mathbf{d} \rangle$

Computing Subgradient of $f(\mathbf{d})$ = Strong Oracle!

Boosting as an Optimization Problem

- Minimizing the maximum edge

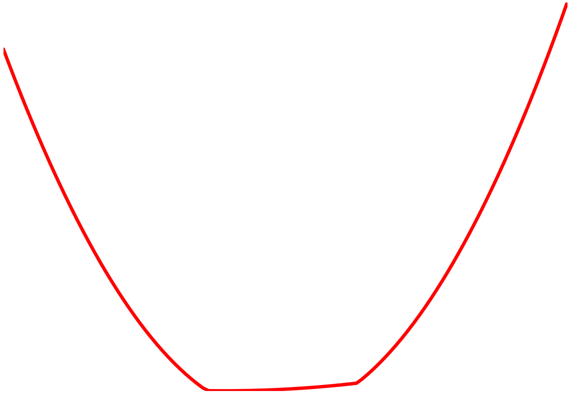
$$\min_{\mathbf{d}\in\mathcal{S}^N} \underbrace{\max_{\mathbf{u}\in\mathcal{U}} \langle \mathbf{u}, \mathbf{d} \rangle}_{f(\mathbf{d})}$$

  is a convex optimization problem.

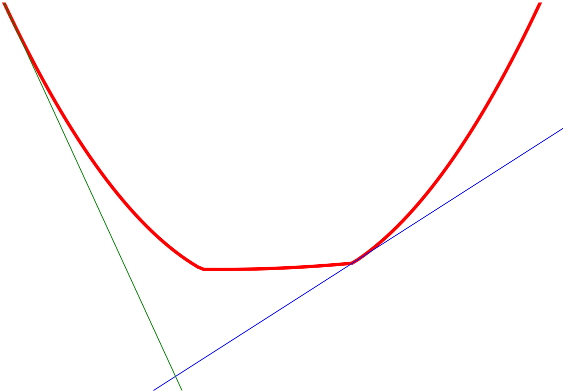
- Subgradient: $\partial f(\mathbf{d}) = \mathrm{argmax}_{\mathbf{u}\in\mathcal{U}} \langle \mathbf{u}, \mathbf{d} \rangle$

Computing Subgradient of $f(\mathbf{d})$ = Strong Oracle!

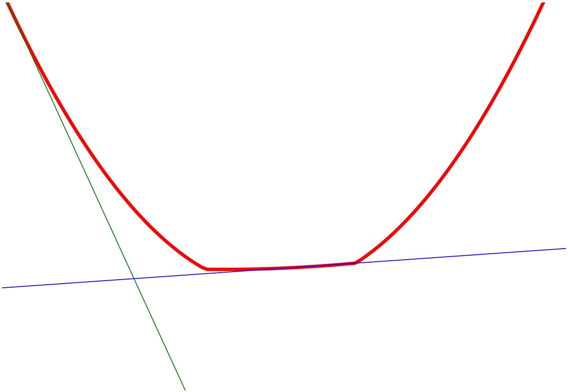
Subgradients and Stability



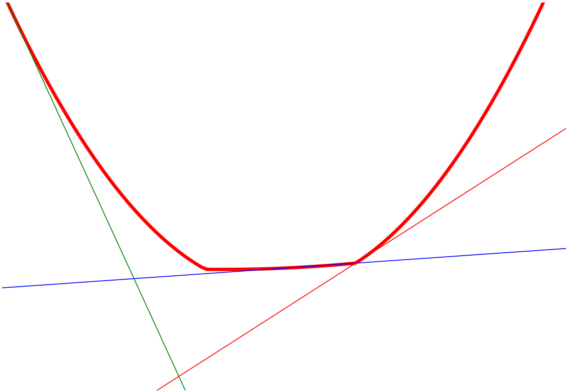
Subgradients and Stability



Subgradients and Stability

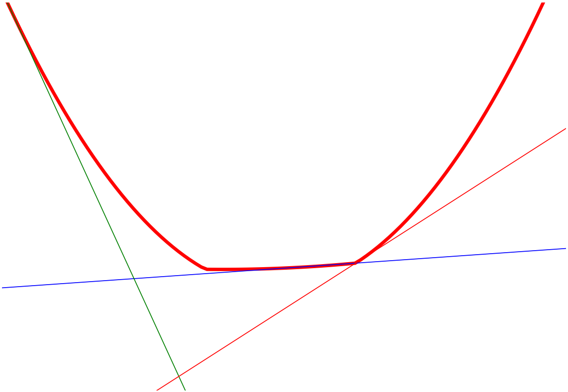


Subgradients and Stability



- Picking an arbitrary subgradient can cause stability issues

Subgradients and Stability



- Picking an arbitrary subgradient can cause stability issues

Back to Convex Analysis

Strong Convexity

A convex function *f* is strongly convex if, and only if, there exists a constant $\sigma > 0$ such that the function $f(x) - \frac{\sigma}{2}\|x\|^2$ is convex.

- If *f* is twice differentiable then the eigenvalues of the Hessian of *f* are lower bounded

$$\nabla^2 f(x) \succeq \sigma I.$$

- If *f* is strongly convex then

$$f(x') - f(x) - \langle x' - x, \mu \rangle \geq \frac{\sigma}{2}\|x' - x\|^2 \quad \forall x, x' \text{ and } \mu \in \partial f(x).$$

Back to Convex Analysis

Strong Convexity

A convex function f is strongly convex if, and only if, there exists a constant $\sigma > 0$ such that the function $f(x) - \frac{\sigma}{2}\|x\|^2$ is convex.

- If f is twice differentiable then the eigenvalues of the Hessian of f are lower bounded

$$\nabla^2 f(x) \succeq \sigma I.$$

- If f is strongly convex then

$$f(x') - f(x) - \langle x' - x, \mu \rangle \geq \frac{\sigma}{2}\|x' - x\|^2 \quad \forall x, x' \text{ and } \mu \in \partial f(x).$$

Back to Convex Analysis

Strong Convexity

A convex function f is strongly convex if, and only if, there exists a constant $\sigma > 0$ such that the function $f(x) - \frac{\sigma}{2}\|x\|^2$ is convex.

- If f is twice differentiable then the eigenvalues of the Hessian of f are lower bounded

$$\nabla^2 f(x) \succeq \sigma I.$$

- If f is strongly convex then

$$f(x') - f(x) - \langle x' - x, \mu \rangle \geq \frac{\sigma}{2}\|x' - x\|^2 \quad \forall x, x' \text{ and } \mu \in \partial f(x).$$

Bundle Methods [HL93]

- Are stabilized cutting plane methods
- At every iteration they form the model

$$f_t(x) := \Omega(x) + \max_{1 \leq i \leq t} \{ f(x_{i-1}) + \langle x - x_{i-1}, s_i \rangle \}.$$

  where Ω is an appropriately chosen strongly convex function, and s_i denotes a (sub)gradient of f evaluated at x_{i-1}.
- At iteration t the set $\{x_i\}_{i=0}^{t-1}$ is augmented by

$$x_t := \operatorname{argmin}_x f_t(x).$$

- Stop when

$$\epsilon_t := \min_{0 \leq i \leq t} \Omega(x_i) + f(x_i) - f_t(x_t)$$

  falls below a pre-specified threshold ϵ.

Bundle Methods [HL93]

- Are stabilized cutting plane methods
- At every iteration they form the model

$$f_t(x) := \Omega(x) + \max_{1 \le i \le t} \{ f(x_{i-1}) + \langle x - x_{i-1}, s_i \rangle \}.$$

where Ω is an appropriately chosen strongly convex function, and s_i denotes a (sub)gradient of f evaluated at x_{i-1}.

- At iteration t the set $\{x_i\}_{i=0}^{t-1}$ is augmented by

$$x_t := \operatorname*{argmin}_x f_t(x).$$

- Stop when

$$\epsilon_t := \min_{0 \le i \le t} \Omega(x_i) + f(x_i) - f_t(x_t)$$

falls below a pre-specified threshold ϵ.

Bundle Methods [HL93]

- Are stabilized cutting plane methods
- At every iteration they form the model

$$f_t(x) := \Omega(x) + \max_{1 \le i \le t} \{ f(x_{i-1}) + \langle x - x_{i-1}, s_i \rangle \}.$$

  where Ω is an appropriately chosen strongly convex function, and s_i denotes a (sub)gradient of f evaluated at x_{i-1}.
- At iteration t the set $\{x_i\}_{i=0}^{t-1}$ is augmented by

$$x_t := \operatorname*{argmin}_x f_t(x).$$

- Stop when

$$\epsilon_t := \min_{0 \le i \le t} \Omega(x_i) + f(x_i) - f_t(x_t)$$

  falls below a pre-specified threshold ϵ.

Bundle Methods [HL93]

- Are stabilized cutting plane methods
- At every iteration they form the model

$$f_t(x) := \Omega(x) + \max_{1 \leq i \leq t} \{ f(x_{i-1}) + \langle x - x_{i-1}, s_i \rangle \}.$$

  where Ω is an appropriately chosen strongly convex function, and s_i denotes a (sub)gradient of f evaluated at x_{i-1}.
- At iteration t the set $\{x_i\}_{i=0}^{t-1}$ is augmented by

$$x_t := \operatorname*{argmin}_x f_t(x).$$

- Stop when

$$\epsilon_t := \min_{0 \leq i \leq t} \Omega(x_i) + f(x_i) - f_t(x_t)$$

  falls below a pre-specified threshold ϵ.

Rates of Convergence [SVL08]

Nonsmooth Functions

- The number of iterations to reach ϵ precision is bounded by

$$t \leq \frac{c_1}{\epsilon \cdot \sigma} + c_2$$

where c_1 and c_2 are problem dependent constants, and σ is the modulus of strong convexity of Ω.

Proving Iteration Bounds for Boosting

Lemma

Suppose $0 \leq \Omega(\mathbf{d}) \leq \epsilon/2$ *for all* $\mathbf{d}$*, and let*

$$\min_{0 \leq i \leq t} \Omega(\mathbf{d}_i) + f(\mathbf{d}_i) - f_t(\mathbf{d}_t) \leq \epsilon/2.$$

Then

$$f(\mathbf{d}_t) \leq f(\mathbf{d}^*) + \epsilon$$

Proving Iteration Bounds Contd.

Entropy as a Regularizer

Suppose we set $\Omega(\mathbf{d}) = \frac{\epsilon}{2 \log N} \sum_{i=1}^{N} d_i \log d_i$ then

- $\Omega(\mathbf{d}) \leq \epsilon/2$ for all $\mathbf{d}$
- Ω is strongly convex with modulus $\frac{\epsilon}{2 \log N}$.

- Plugging this into rates of convergence of bundle methods shows that we need

$$t \leq 2 \frac{c_1 \log N}{\epsilon^2} + c_2$$

iterations to obtain an ϵ accurate solution [SSS08].

- Similar iteration bounds for
  - strong oracle: returns $\operatorname{argmax}_{\mathbf{u} \in \mathcal{U}} \langle \mathbf{u}, \mathbf{d} \rangle$
  - weak oracle: returns an $\mathbf{u}$ such that $\langle \mathbf{u}, \mathbf{d} \rangle \geq g$ [WGV08]

Proving Iteration Bounds Contd.

Entropy as a Regularizer

Suppose we set $\Omega(\mathbf{d}) = \frac{\epsilon}{2 \log N} \sum_{i=1}^{N} d_i \log d_i$ then

- $\Omega(\mathbf{d}) \leq \epsilon/2$ for all $\mathbf{d}$
- Ω is strongly convex with modulus $\frac{\epsilon}{2 \log N}$.

- Plugging this into rates of convergence of bundle methods shows that we need

$$t \leq 2\frac{c_1 \log N}{\epsilon^2} + c_2$$

iterations to obtain an ϵ accurate solution [SSS08].

- Similar iteration bounds for
  - strong oracle: returns $\mathrm{argmax}_{\mathbf{u} \in \mathcal{U}} \langle \mathbf{u}, \mathbf{d} \rangle$
  - weak oracle: returns an $\mathbf{u}$ such that $\langle \mathbf{u}, \mathbf{d} \rangle \geq g$ [WGV08]

Proving Iteration Bounds Contd.

Entropy as a Regularizer

Suppose we set $\Omega(\mathbf{d}) = \frac{\epsilon}{2 \log N} \sum_{i=1}^{N} d_i \log d_i$ then

- $\Omega(\mathbf{d}) \leq \epsilon/2$ for all $\mathbf{d}$
- Ω is strongly convex with modulus $\frac{\epsilon}{2 \log N}$.

- Plugging this into rates of convergence of bundle methods shows that we need

$$t \leq 2\frac{c_1 \log N}{\epsilon^2} + c_2$$

iterations to obtain an ϵ accurate solution [SSS08].

- Similar iteration bounds for
  - strong oracle: returns $\operatorname{argmax}_{\mathbf{u} \in \mathcal{U}} \langle \mathbf{u}, \mathbf{d} \rangle$
  - weak oracle: returns an $\mathbf{u}$ such that $\langle \mathbf{u}, \mathbf{d} \rangle \geq g$ [WGV08]

$\Omega(1/\epsilon^2)$ Iteration Bounds [NY78,BMN01]

Setting

- Let B be any black box boosting algorithm
- $\mathbf{d}^1, \ldots, \mathbf{d}^{t-1}$ are intermediate distributions
- $\mathbf{d}^t$ is the distribution produced by B after t iterations

The Adversary

- Produces a set of hypothesis $\mathbf{u}_1, \ldots, \mathbf{u}_t$ which defines a function

$$f(\mathbf{d}) := \max_{q=1,\ldots,t} \langle \mathbf{u}^q, \mathbf{d} \rangle$$

- We will show that $f(\mathbf{d}^t) - \min_{\mathbf{d}} f(\mathbf{d}) \geq \frac{1}{\sqrt{t}}$
- To get ϵ accuracy B needs $O(1/\epsilon^2)$ iterations.

$\Omega(1/\epsilon^2)$ Iteration Bounds [NY78,BMN01]

Setting

- Let B be any black box boosting algorithm
- $\mathbf{d}^1, \ldots, \mathbf{d}^{t-1}$ are intermediate distributions
- $\mathbf{d}^t$ is the distribution produced by B after t iterations

The Adversary

- Produces a set of hypothesis $\mathbf{u}_1, \ldots, \mathbf{u}_t$ which defines a function

$$f(\mathbf{d}) := \max_{q=1,\ldots,t} \langle \mathbf{u}^q, \mathbf{d} \rangle$$

- We will show that $f(\mathbf{d}^t) - \min_{\mathbf{d}} f(\mathbf{d}) \geq \frac{1}{\sqrt{t}}$
- To get ϵ accuracy B needs $O(1/\epsilon^2)$ iterations.

$\Omega(1/\epsilon^2)$ Iteration Bounds [NY78,BMN01]

Setting

- Let B be any black box boosting algorithm
- $\mathbf{d}^1, \dots, \mathbf{d}^{t-1}$ are intermediate distributions
- $\mathbf{d}^t$ is the distribution produced by B after t iterations

The Adversary

- Produces a set of hypothesis $\mathbf{u}_1, \dots, \mathbf{u}_t$ which defines a function

$$f(\mathbf{d}) := \max_{q=1,\dots,t} \langle \mathbf{u}^q, \mathbf{d} \rangle$$

- We will show that $f(\mathbf{d}^t) - \min_{\mathbf{d}} f(\mathbf{d}) \geq \frac{1}{\sqrt{t}}$
- To get ϵ accuracy B needs $O(1/\epsilon^2)$ iterations.

$\Omega(1/\epsilon^2)$ Iteration Bounds [NY78,BMN01]

Setting

- Let B be any black box boosting algorithm
- $\mathbf{d}^1, \ldots, \mathbf{d}^{t-1}$ are intermediate distributions
- $\mathbf{d}^t$ is the distribution produced by B after t iterations

The Adversary

- Produces a set of hypothesis $\mathbf{u}_1, \ldots, \mathbf{u}_t$ which defines a function

$$f(\mathbf{d}) := \max_{q=1,\ldots,t} \langle \mathbf{u}^q, \mathbf{d} \rangle$$

- We will show that $f(\mathbf{d}^t) - \min_{\mathbf{d}} f(\mathbf{d}) \geq \frac{1}{\sqrt{t}}$
- To get ϵ accuracy B needs $O(1/\epsilon^2)$ iterations.

$\Omega(1/\epsilon^2)$ Iteration Bounds [NY78,BMN01]

Setting

- Let B be any black box boosting algorithm
- $\mathbf{d}^1, \ldots, \mathbf{d}^{t-1}$ are intermediate distributions
- $\mathbf{d}^t$ is the distribution produced by B after t iterations

The Adversary

- Produces a set of hypothesis $\mathbf{u}_1, \ldots, \mathbf{u}_t$ which defines a function

$$f(\mathbf{d}) := \max_{q=1,\ldots,t} \langle \mathbf{u}^q, \mathbf{d} \rangle$$

- We will show that $f(\mathbf{d}^t) - \min_{\mathbf{d}} f(\mathbf{d}) \geq \frac{1}{\sqrt{t}}$
- To get ϵ accuracy B needs $O(1/\epsilon^2)$ iterations.

$\Omega(1/\epsilon^2)$ Iteration Bounds [NY78,BMN01]

Setting

- Let B be any black box boosting algorithm
- $\mathbf{d}^1, \ldots, \mathbf{d}^{t-1}$ are intermediate distributions
- $\mathbf{d}^t$ is the distribution produced by B after t iterations

The Adversary

- Produces a set of hypothesis $\mathbf{u}_1, \ldots, \mathbf{u}_t$ which defines a function

$$f(\mathbf{d}) := \max_{q=1,\ldots,t} \langle \mathbf{u}^q, \mathbf{d} \rangle$$

- We will show that $f(\mathbf{d}^t) - \min_{\mathbf{d}} f(\mathbf{d}) \geq \frac{1}{\sqrt{t}}$
- To get ϵ accuracy B needs $O(1/\epsilon^2)$ iterations.

The Resisting Oracle

Hadamard Matrix

The Hadamard matrix is defined recursively:

$$\mathbf{H}_2 = \begin{pmatrix} +1 & +1 \\ +1 & -1 \end{pmatrix} \quad \mathbf{H}_{2N} = \begin{pmatrix} \mathbf{H}_N & \mathbf{H}_N \\ \mathbf{H}_N & -\mathbf{H}_N \end{pmatrix}$$

Hypothesis Space

- The oracle chooses $\mathbf{u}^q$ from the columns of the following matrix:

$$\mathbf{U} = \begin{pmatrix} \mathbf{H}_{N/2} & -\mathbf{H}_{N/2} \\ -\mathbf{H}_{N/2} & \mathbf{H}_{N/2} \end{pmatrix}$$

- Because of symmetry, for any $t < N/2$ we have

$$f(\mathbf{d}^t) = \max_{\mathbf{u} \in \mathbf{U}} \langle \mathbf{u}, \mathbf{d}^t \rangle \geq 0.$$

The Resisting Oracle

Hadamard Matrix

The Hadamard matrix is defined recursively:

$$\mathbf{H}_2 = \begin{pmatrix} +1 & +1 \\ +1 & -1 \end{pmatrix} \quad \mathbf{H}_{2N} = \begin{pmatrix} \mathbf{H}_N & \mathbf{H}_N \\ \mathbf{H}_N & -\mathbf{H}_N \end{pmatrix}$$

Hypothesis Space

- The oracle chooses $\mathbf{u}^q$ from the columns of the following matrix:

$$\mathbf{U} = \begin{pmatrix} \mathbf{H}_{N/2} & -\mathbf{H}_{N/2} \\ -\mathbf{H}_{N/2} & \mathbf{H}_{N/2} \end{pmatrix}$$

- Because of symmetry, for any $t < N/2$ we have

$$f(\mathbf{d}^t) = \max_{\mathbf{u} \in \mathbf{U}} \langle \mathbf{u}, \mathbf{d}^t \rangle \geq 0.$$

The Resisting Oracle

Hadamard Matrix

The Hadamard matrix is defined recursively:

$$\mathbf{H}_2 = \begin{pmatrix} +1 & +1 \\ +1 & -1 \end{pmatrix} \quad \mathbf{H}_{2N} = \begin{pmatrix} \mathbf{H}_N & \mathbf{H}_N \\ \mathbf{H}_N & -\mathbf{H}_N \end{pmatrix}$$

Hypothesis Space

- The oracle chooses $\mathbf{u}^q$ from the columns of the following matrix:

$$\mathbf{U} = \begin{pmatrix} \mathbf{H}_{N/2} & -\mathbf{H}_{N/2} \\ -\mathbf{H}_{N/2} & \mathbf{H}_{N/2} \end{pmatrix}$$

- Because of symmetry, for any $t < N/2$ we have

$$f(\mathbf{d}^t) = \max_{\mathbf{u} \in \mathbf{U}} \langle \mathbf{u}, \mathbf{d}^t \rangle \geq 0.$$

Upper Bound on $\min_{\mathbf{d}} f(\mathbf{d})$

$$
\begin{aligned}
\min_{\mathbf{d}\in\mathcal{S}^N} f(\mathbf{d}) &= \min_{\mathbf{d}\in S^n}\max_{q=1,\ldots,t}\underbrace{\langle \mathbf{u}^q, \mathbf{d}\rangle}_{\text{edge}} \\
&\overset{\text{minimax thm}}{=} \max_{\mathbf{w}\in\mathcal{S}^t}\min_{j=1,\ldots,n}\underbrace{[\mathbf{U}_t\,\mathbf{w}]_j}_{\text{margin}} \\
&\overset{*}{=} \max_{\mathbf{w}\in\mathcal{S}^t} -\left(\max_{j=1,\ldots,n}[\widehat{\mathbf{U}}_t\,\mathbf{w}]_j\right) \\
&= -\min_{\mathbf{w}\in\mathcal{S}^t}\|\widehat{\mathbf{U}}_t\,\mathbf{w}\,\|_\infty
\end{aligned}
$$

* because $\mathbf{U}$ has the form $\mathbf{U} = \begin{pmatrix} \widehat{\mathbf{U}} \\ -\widehat{\mathbf{U}} \end{pmatrix}$

The Resisting Oracle Contd.

$$-\|\cdot\|_\infty \leq -\frac{1}{\sqrt{n}}\|\cdot\|_2$$

$$\begin{aligned}
\min_{\mathbf{d}\in\mathcal{S}^N} f(\mathbf{d}) &= -\min_{\mathbf{w}\in\mathcal{S}^t} \|\widehat{\mathbf{U}}_t\,\mathbf{w}\,\|_\infty \\
&\leq -\frac{1}{\sqrt{n/2}} \min_{\mathbf{w}\in S^t} \|\widehat{\mathbf{U}}_t\,\mathbf{w}\,\|_2 \\
&= -\frac{1}{\sqrt{n/2}} \min_{\mathbf{w}\in S^t} \sqrt{\mathbf{w}^\top \underbrace{\widehat{\mathbf{U}}_t^\top \widehat{\mathbf{U}}_t}_{n/2\cdot\mathbf{I}_t}\mathbf{w}} \\
&= -\min_{\mathbf{w}\in S^t} \sqrt{\mathbf{w}^\top\,\mathbf{w}} \\
&\leq -\frac{1}{\sqrt{t}}
\end{aligned}$$

Towards practical algorithms for our optimization Problem

Primal Problem

$$\min_{\substack{\mathbf{d} \cdot \boldsymbol{I} = 1 \\ \mathbf{d} \leq \frac{1}{\nu} \boldsymbol{I}}} \quad \underbrace{\frac{1}{\eta}\Delta(\mathbf{d}, \mathbf{d}^0) + \max_{q=1,2,\ldots,t} \langle \mathbf{u}^q, \mathbf{d} \rangle}_{:= P^t(\mathbf{d})} .$$

Dual Problem

$$\max_{\substack{\mathbf{w} \geq \mathbf{0} \\ \langle \mathbf{w}, \mathbf{e} \rangle = 1 \\ \psi \geq 0}} \quad -\frac{1}{\eta} \log \sum_{n=1}^{N} d_n^0 \exp(-\eta \sum_{q=1}^{t} u_n^q w_q - \eta \psi_n) - \frac{1}{\nu} \sum_{n=1}^{N} \psi_n$$

Towards practical algorithms for our optimization Problem

Primal Problem

$$\min_{\substack{\mathbf{d}\cdot\boldsymbol{I}=1\\ \mathbf{d}\le\frac{1}{\nu}\boldsymbol{I}}} \quad \underbrace{\frac{1}{\eta}\Delta(\mathbf{d},\mathbf{d}^0)+\max_{q=1,2,\ldots,t}\langle\mathbf{u}^q,\mathbf{d}\rangle}_{:=P^t(\mathbf{d})}.$$

Dual Problem

$$\max_{\substack{\mathbf{w}\ge\mathbf{0}\\ \langle\mathbf{w},\mathbf{e}\rangle=1\\ \boldsymbol{\psi}\ge 0}} \quad -\frac{1}{\eta}\log\sum_{n=1}^{N} d_n^0\exp(-\eta\sum_{q=1}^{t}u_n^q w_q-\eta\psi_n)-\frac{1}{\nu}\sum_{n=1}^{N}\psi_n$$

Gradient Descent

Unconstrained

- Suppose you want to minimize

$$\min_{\mathbf{w} \in \mathbb{R}^n} f(\mathbf{w})$$

- Gradient descent produces a sequence of iterates

$$\mathbf{w}_{t+1} = \mathbf{w}_t - \eta \nabla f(\mathbf{w}_t)$$

- The step-size η found by
  - Solving $\min_\eta f(\mathbf{w}_t - \eta \nabla f(\mathbf{w}_t))$ or
  - Decay schedule such as $\eta_t = \frac{1}{t}$

Gradient Descent

Unconstrained

- Suppose you want to minimize

$$\min_{\mathbf{w} \in \mathbb{R}^n} f(\mathbf{w})$$

- Gradient descent produces a sequence of iterates

$$\mathbf{w}_{t+1} = \mathbf{w}_t - \eta \nabla f(\mathbf{w}_t)$$

- The step-size η found by
  - Solving $\min_\eta f(\mathbf{w}_t - \eta \nabla f(\mathbf{w}_t))$ or
  - Decay schedule such as $\eta_t = \frac{1}{t}$

Gradient Descent

Unconstrained

- Suppose you want to minimize

$$\min_{\mathbf{w} \in \mathbb{R}^n} f(\mathbf{w})$$

- Gradient descent produces a sequence of iterates

$$\mathbf{w}_{t+1} = \mathbf{w}_t - \eta \nabla f(\mathbf{w}_t)$$

- The step-size η found by
  - Solving $\min_\eta f(\mathbf{w}_t - \eta \nabla f(\mathbf{w}_t))$ or
  - Decay schedule such as $\eta_t = \frac{1}{t}$

Projected Gradient Descent

Constrained

- Suppose you want to minimize

$$\min_{\mathbf{w}\in\Gamma} f(\mathbf{w})$$

- Projected Gradient descent produces a sequence of iterates

$$\mathbf{w}_{t+1} = P_\Gamma(\mathbf{w}_t - \eta\nabla f(\mathbf{w}))$$

where $P_\Gamma(\mathbf{z}) := \mathrm{argmin}_{x\in\Omega} \|\mathbf{z} - x\|^2$.

Projected Gradient Descent

Constrained

- Suppose you want to minimize

$$\min_{\mathbf{w} \in \Gamma} f(\mathbf{w})$$

- Projected Gradient descent produces a sequence of iterates

$$\mathbf{w}_{t+1} = P_\Gamma(\mathbf{w}_t - \eta \nabla f(\mathbf{w}))$$

where $P_\Gamma(\mathbf{z}) := \mathrm{argmin}_{x \in \Omega} \| \mathbf{z} - x \|^2$.

Spectral Projected Gradient Method [BMR00]

- **Step 1:** Detect if current point is stationary
- **Step 2:** Backtracking
  - **Step 2.1:** Compute $\mathbf{d}_t = P_\Omega(\mathbf{w}_t - \alpha_t \nabla f(\mathbf{w})) - \mathbf{w}_t$. Set $\lambda = 1$
  - **Step 2.2:** Set $\mathbf{w}_+ = \mathbf{w}_t + \lambda \mathbf{d}_t$
  - **Step 2.3:** If $f(\mathbf{w}_+) \leq \max_{0 \leq j \leq M} f(\mathbf{w}_{t-j}) + \gamma \lambda \langle \mathbf{d}_t, \nabla f(\mathbf{w}_{t-j}) \rangle$
    $\lambda_t = \lambda$, $\mathbf{w}_{t+1} = \mathbf{w}_+$, $\mathbf{s}_k = \mathbf{w}_{t+1} - \mathbf{w}_t$, $\mathbf{y}_t = \nabla f(\mathbf{w}_{t+1}) - \nabla f(\mathbf{w}_t)$
  - Else define $\lambda \in [\sigma_1 \lambda, \sigma_2 \lambda]$ and go to Step 2.2
- **Step 3:** Compute $b_t = \langle \mathbf{s}_t, \mathbf{y}_t \rangle$.
  - If $b_t \leq 0$ set $\alpha_{t+1} = \alpha_{\max}$
  - Else compute $a_t = \langle \mathbf{y}_t, \mathbf{y}_t \rangle$ and set
    $\alpha_{t+1} = \min\{\alpha_{\max}, \max\{\alpha_{\min}, a_t / b_t\}\}$

Spectral Projected Gradient Method [BMR00]

- **Step 1:** Detect if current point is stationary
- **Step 2:** Backtracking
  - **Step 2.1:** Compute $\mathbf{d}_t = P_\Omega(\mathbf{w}_t - \alpha_t \nabla f(\mathbf{w})) - \mathbf{w}_t$. Set $\lambda = 1$
  - **Step 2.2:** Set $\mathbf{w}_+ = \mathbf{w}_t + \lambda\, \mathbf{d}_t$
  - **Step 2.3:** If $f(\mathbf{w}_+) \leq \max_{0 \leq j \leq M} f(\mathbf{w}_{t-j}) + \gamma\lambda \langle \mathbf{d}_t, \nabla f(\mathbf{w}_{t-j}) \rangle$
    $\lambda_t = \lambda$, $\mathbf{w}_{t+1} = \mathbf{w}_+$, $\mathbf{s}_k = \mathbf{w}_{t+1} - \mathbf{w}_t$, $\mathbf{y}_t = \nabla f(\mathbf{w}_{t+1}) - \nabla f(\mathbf{w}_t)$
  - Else define $\lambda \in [\sigma_1 \lambda, \sigma_2 \lambda]$ and go to Step 2.2
- **Step 3:** Compute $b_t = \langle \mathbf{s}_t, \mathbf{y}_t \rangle$.
  - If $b_t \leq 0$ set $\alpha_{t+1} = \alpha_{\max}$
  - Else compute $a_t = \langle \mathbf{y}_t, \mathbf{y}_t \rangle$ and set
    $\alpha_{t+1} = \min\{\alpha_{\max}, \max\{\alpha_{\min}, a_t/b_t\}\}$

Spectral Projected Gradient Method [BMR00]

- **Step 1:** Detect if current point is stationary
- **Step 2:** Backtracking
  - **Step 2.1:** Compute $\mathbf{d}_t = P_\Omega(\mathbf{w}_t - \alpha_t \nabla f(\mathbf{w})) - \mathbf{w}_t$. Set $\lambda = 1$
  - **Step 2.2:** Set $\mathbf{w}_+ = \mathbf{w}_t + \lambda\, \mathbf{d}_t$
  - **Step 2.3:** If $f(\mathbf{w}_+) \leq \max_{0 \leq j \leq M} f(\mathbf{w}_{t-j}) + \gamma\lambda \langle \mathbf{d}_t, \nabla f(\mathbf{w}_{t-j}) \rangle$ $\lambda_t = \lambda$, $\mathbf{w}_{t+1} = \mathbf{w}_+$, $\mathbf{s}_k = \mathbf{w}_{t+1} - \mathbf{w}_t$, $\mathbf{y}_t = \nabla f(\mathbf{w}_{t+1}) - \nabla f(\mathbf{w}_t)$
  - Else define $\lambda \in [\sigma_1\lambda, \sigma_2\lambda]$ and go to Step 2.2
- **Step 3:** Compute $b_t = \langle \mathbf{s}_t, \mathbf{y}_t \rangle$.
  - If $b_t \leq 0$ set $\alpha_{t+1} = \alpha_{\max}$
  - Else compute $a_t = \langle \mathbf{y}_t, \mathbf{y}_t \rangle$ and set $\alpha_{t+1} = \min\{\alpha_{\max}, \max\{\alpha_{\min}, a_t / b_t\}\}$

Spectral Projected Gradient Method [BMR00]

- **Step 1:** Detect if current point is stationary
- **Step 2:** Backtracking
  - **Step 2.1:** Compute $\mathbf{d}_t = P_\Omega(\mathbf{w}_t - \alpha_t \nabla f(\mathbf{w})) - \mathbf{w}_t$. Set $\lambda = 1$
  - **Step 2.2:** Set $\mathbf{w}_+ = \mathbf{w}_t + \lambda\, \mathbf{d}_t$
  - **Step 2.3:** If $f(\mathbf{w}_+) \leq \max_{0 \leq j \leq M} f(\mathbf{w}_{t-j}) + \gamma\lambda \langle \mathbf{d}_t, \nabla f(\mathbf{w}_{t-j}) \rangle$ $\lambda_t = \lambda$, $\mathbf{w}_{t+1} = \mathbf{w}_+$, $\mathbf{s}_k = \mathbf{w}_{t+1} - \mathbf{w}_t$, $\mathbf{y}_t = \nabla f(\mathbf{w}_{t+1}) - \nabla f(\mathbf{w}_t)$
  - Else define $\lambda \in [\sigma_1 \lambda, \sigma_2 \lambda]$ and go to Step 2.2
- **Step 3:** Compute $b_t = \langle \mathbf{s}_t, \mathbf{y}_t \rangle$.
  - If $b_t \leq 0$ set $\alpha_{t+1} = \alpha_{\max}$
  - Else compute $a_t = \langle \mathbf{y}_t, \mathbf{y}_t \rangle$ and set $\alpha_{t+1} = \min\{\alpha_{\max}, \max\{\alpha_{\min}, a_t/b_t\}\}$

Spectral Projected Gradient Method [BMR00]

- **Step 1:** Detect if current point is stationary
- **Step 2:** Backtracking
  - **Step 2.1:** Compute $\mathbf{d}_t = P_\Omega(\mathbf{w}_t - \alpha_t \nabla f(\mathbf{w})) - \mathbf{w}_t$. Set $\lambda = 1$
  - **Step 2.2:** Set $\mathbf{w}_+ = \mathbf{w}_t + \lambda \, \mathbf{d}_t$
  - **Step 2.3:** If $f(\mathbf{w}_+) \leq \max_{0 \leq j \leq M} f(\mathbf{w}_{t-j}) + \gamma\lambda \langle \mathbf{d}_t, \nabla f(\mathbf{w}_{t-j}) \rangle$
    $\lambda_t = \lambda$, $\mathbf{w}_{t+1} = \mathbf{w}_+$, $\mathbf{s}_k = \mathbf{w}_{t+1} - \mathbf{w}_t$, $\mathbf{y}_t = \nabla f(\mathbf{w}_{t+1}) - \nabla f(\mathbf{w}_t)$
  - Else define $\lambda \in [\sigma_1\lambda, \sigma_2\lambda]$ and go to Step 2.2
- **Step 3:** Compute $b_t = \langle \mathbf{s}_t, \mathbf{y}_t \rangle$.
  - If $b_t \leq 0$ set $\alpha_{t+1} = \alpha_{\max}$
  - Else compute $a_t = \langle \mathbf{y}_t, \mathbf{y}_t \rangle$ and set
    $\alpha_{t+1} = \min\{\alpha_{\max}, \max\{\alpha_{\min}, a_t/b_t\}\}$

Spectral Projected Gradient Method [BMR00]

- **Step 1:** Detect if current point is stationary
- **Step 2:** Backtracking
  - **Step 2.1:** Compute $\mathbf{d}_t = P_\Omega(\mathbf{w}_t - \alpha_t \nabla f(\mathbf{w})) - \mathbf{w}_t$. Set $\lambda = 1$
  - **Step 2.2:** Set $\mathbf{w}_+ = \mathbf{w}_t + \lambda\, \mathbf{d}_t$
  - **Step 2.3:** If $f(\mathbf{w}_+) \leq \max_{0 \leq j \leq M} f(\mathbf{w}_{t-j}) + \gamma\lambda \langle \mathbf{d}_t, \nabla f(\mathbf{w}_{t-j}) \rangle$
    $\lambda_t = \lambda$, $\mathbf{w}_{t+1} = \mathbf{w}_+$, $\mathbf{s}_k = \mathbf{w}_{t+1} - \mathbf{w}_t$, $\mathbf{y}_t = \nabla f(\mathbf{w}_{t+1}) - \nabla f(\mathbf{w}_t)$
  - Else define $\lambda \in [\sigma_1\lambda, \sigma_2\lambda]$ and go to Step 2.2
- **Step 3:** Compute $b_t = \langle \mathbf{s}_t, \mathbf{y}_t \rangle$.
  - If $b_t \leq 0$ set $\alpha_{t+1} = \alpha_{\max}$
  - Else compute $a_t = \langle \mathbf{y}_t, \mathbf{y}_t \rangle$ and set
    $\alpha_{t+1} = \min\{\alpha_{\max}, \max\{\alpha_{\min}, a_t/b_t\}\}$

Spectral Projected Gradient Method [BMR00]

- **Step 1:** Detect if current point is stationary
- **Step 2:** Backtracking
  - **Step 2.1:** Compute $\mathbf{d}_t = P_\Omega(\mathbf{w}_t - \alpha_t \nabla f(\mathbf{w})) - \mathbf{w}_t$. Set $\lambda = 1$
  - **Step 2.2:** Set $\mathbf{w}_+ = \mathbf{w}_t + \lambda\, \mathbf{d}_t$
  - **Step 2.3:** If $f(\mathbf{w}_+) \leq \max_{0 \leq j \leq M} f(\mathbf{w}_{t-j}) + \gamma\lambda \langle \mathbf{d}_t, \nabla f(\mathbf{w}_{t-j}) \rangle$ $\lambda_t = \lambda$, $\mathbf{w}_{t+1} = \mathbf{w}_+$, $\mathbf{s}_k = \mathbf{w}_{t+1} - \mathbf{w}_t$, $\mathbf{y}_t = \nabla f(\mathbf{w}_{t+1}) - \nabla f(\mathbf{w}_t)$
  - Else define $\lambda \in [\sigma_1\lambda, \sigma_2\lambda]$ and go to Step 2.2
- **Step 3:** Compute $b_t = \langle \mathbf{s}_t, \mathbf{y}_t \rangle$.
  - If $b_t \leq 0$ set $\alpha_{t+1} = \alpha_{\max}$
  - Else compute $a_t = \langle \mathbf{y}_t, \mathbf{y}_t \rangle$ and set $\alpha_{t+1} = \min\{\alpha_{\max}, \max\{\alpha_{\min}, a_t/b_t\}\}$

Spectral Projected Gradient Method [BMR00]

- **Step 1:** Detect if current point is stationary
- **Step 2:** Backtracking
  - **Step 2.1:** Compute $\mathbf{d}_t = P_\Omega(\mathbf{w}_t - \alpha_t \nabla f(\mathbf{w})) - \mathbf{w}_t$. Set $\lambda = 1$
  - **Step 2.2:** Set $\mathbf{w}_+ = \mathbf{w}_t + \lambda\, \mathbf{d}_t$
  - **Step 2.3:** If $f(\mathbf{w}_+) \leq \max_{0 \leq j \leq M} f(\mathbf{w}_{t-j}) + \gamma\lambda \langle \mathbf{d}_t, \nabla f(\mathbf{w}_{t-j}) \rangle$
    $\lambda_t = \lambda$, $\mathbf{w}_{t+1} = \mathbf{w}_+$, $\mathbf{s}_k = \mathbf{w}_{t+1} - \mathbf{w}_t$, $\mathbf{y}_t = \nabla f(\mathbf{w}_{t+1}) - \nabla f(\mathbf{w}_t)$
  - Else define $\lambda \in [\sigma_1\lambda, \sigma_2\lambda]$ and go to Step 2.2
- **Step 3:** Compute $b_t = \langle \mathbf{s}_t, \mathbf{y}_t \rangle$.
  - If $b_t \leq 0$ set $\alpha_{t+1} = \alpha_{\max}$
  - Else compute $a_t = \langle \mathbf{y}_t, \mathbf{y}_t \rangle$ and set
    $\alpha_{t+1} = \min\{\alpha_{\max}, \max\{\alpha_{\min}, a_t/b_t\}\}$

Spectral Projected Gradient Method [BMR00]

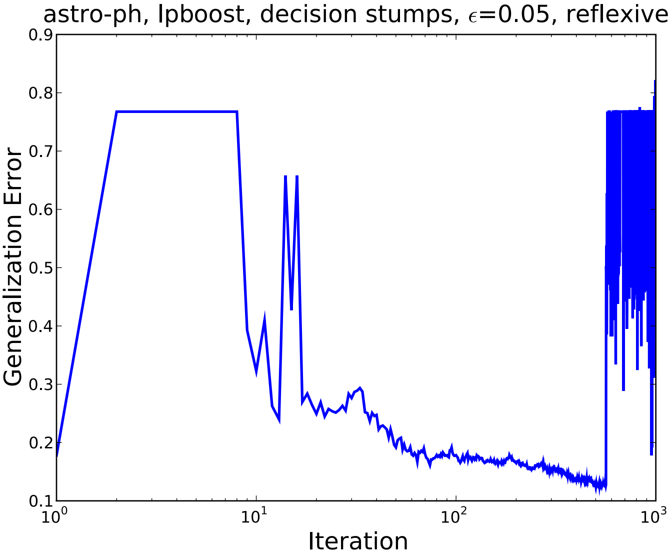
- **Step 1:** Detect if current point is stationary
- **Step 2:** Backtracking
  - **Step 2.1:** Compute $\mathbf{d}_t = P_\Omega(\mathbf{w}_t - \alpha_t \nabla f(\mathbf{w})) - \mathbf{w}_t$. Set $\lambda = 1$
  - **Step 2.2:** Set $\mathbf{w}_+ = \mathbf{w}_t + \lambda\, \mathbf{d}_t$
  - **Step 2.3:** If $f(\mathbf{w}_+) \leq \max_{0 \leq j \leq M} f(\mathbf{w}_{t-j}) + \gamma\lambda \langle \mathbf{d}_t, \nabla f(\mathbf{w}_{t-j}) \rangle$
    $\lambda_t = \lambda$, $\mathbf{w}_{t+1} = \mathbf{w}_+$, $\mathbf{s}_k = \mathbf{w}_{t+1} - \mathbf{w}_t$, $\mathbf{y}_t = \nabla f(\mathbf{w}_{t+1}) - \nabla f(\mathbf{w}_t)$
  - Else define $\lambda \in [\sigma_1\lambda, \sigma_2\lambda]$ and go to Step 2.2
- **Step 3:** Compute $b_t = \langle \mathbf{s}_t, \mathbf{y}_t \rangle$.
  - If $b_t \leq 0$ set $\alpha_{t+1} = \alpha_{\max}$
  - Else compute $a_t = \langle \mathbf{y}_t, \mathbf{y}_t \rangle$ and set
    $\alpha_{t+1} = \min\{\alpha_{\max}, \max\{\alpha_{\min}, a_t/b_t\}\}$

Dataset Properties

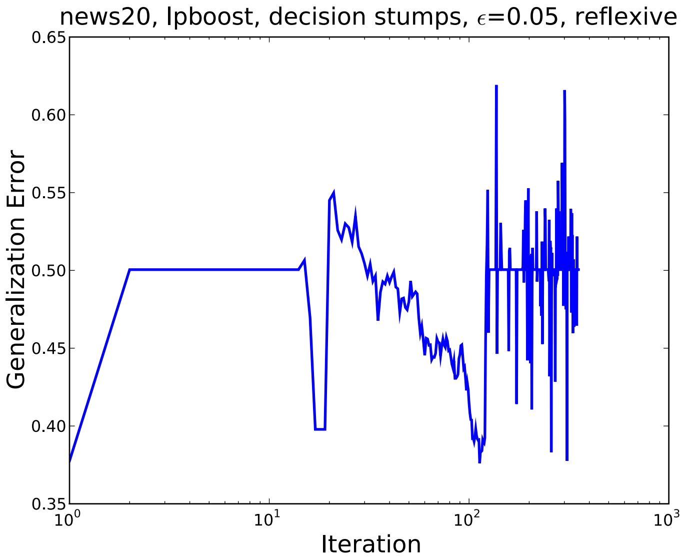
Name	Size	Dimension	Density
Astro-ph	94,856	99,757	0.008
News20	19,954	1,355,191	0.03
Real-Sim	72,201	20,958	0.25
rcv1	677,399	47,236	0.15

- Combined test and train. 60% randomly chosen for train, 20% for test and 20% for validation.
- Plot for generating the splits available.

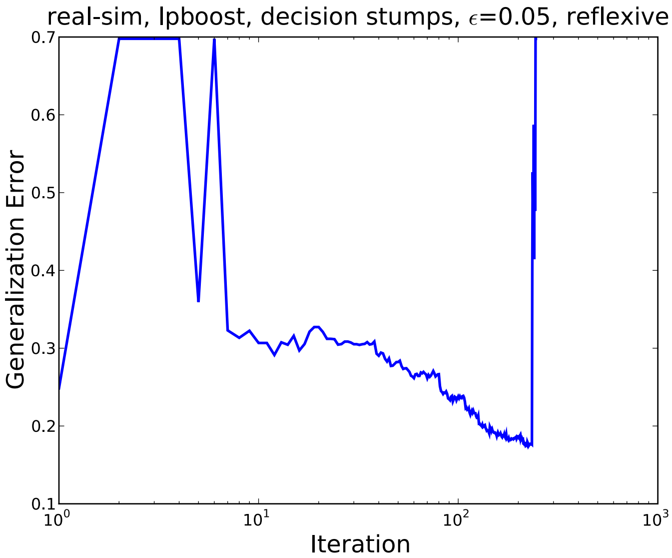
LPBoost is Brittle



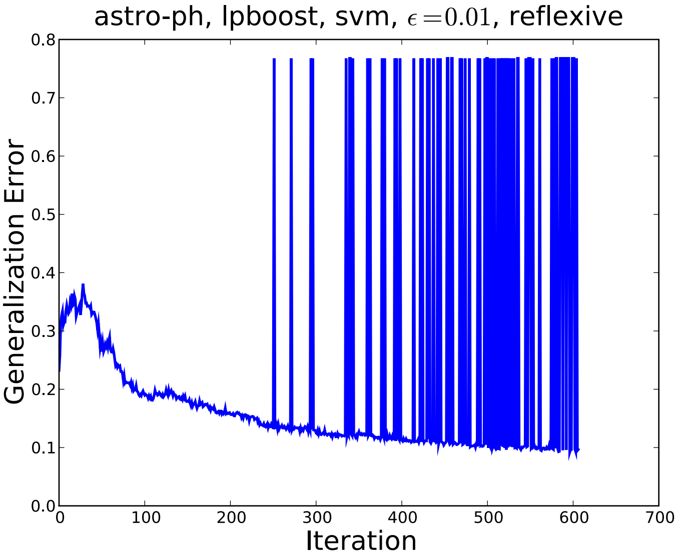
LPBoost is Brittle



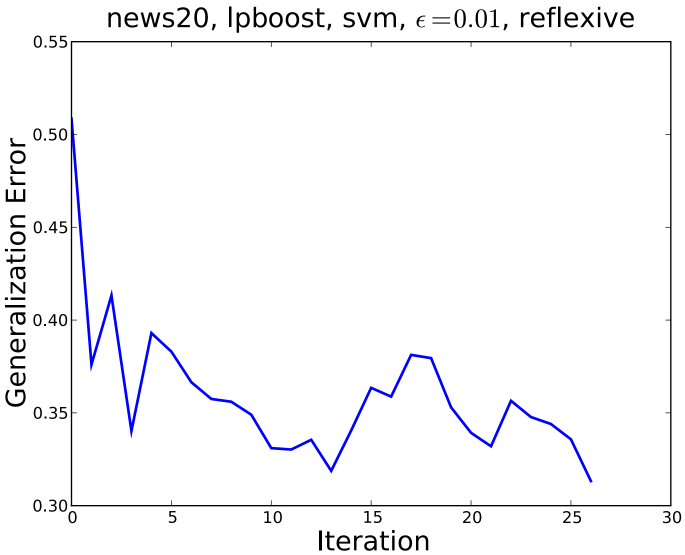
LPBoost is Brittle



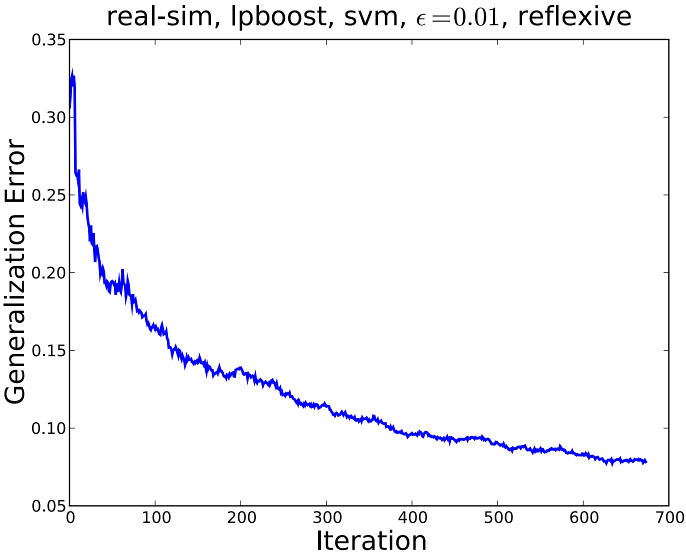
LPBoost is Brittle



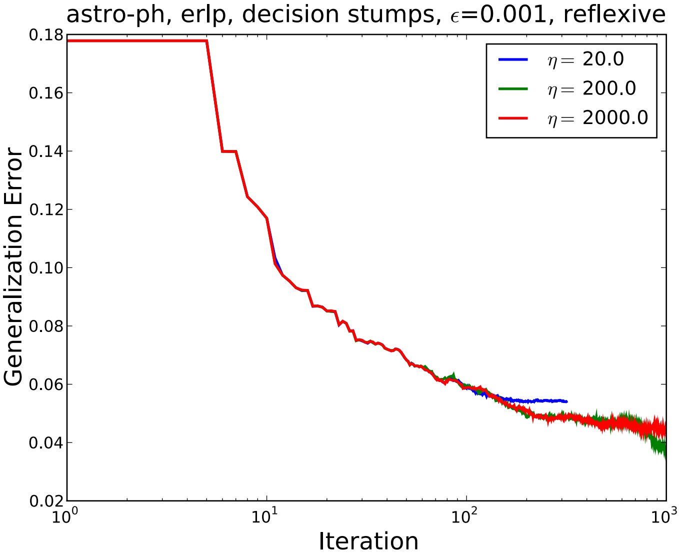
LPBoost is Brittle



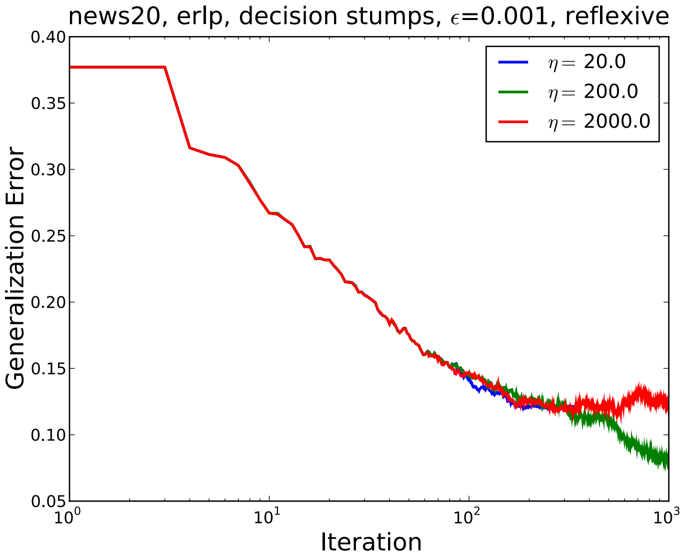
LPBoost is Brittle



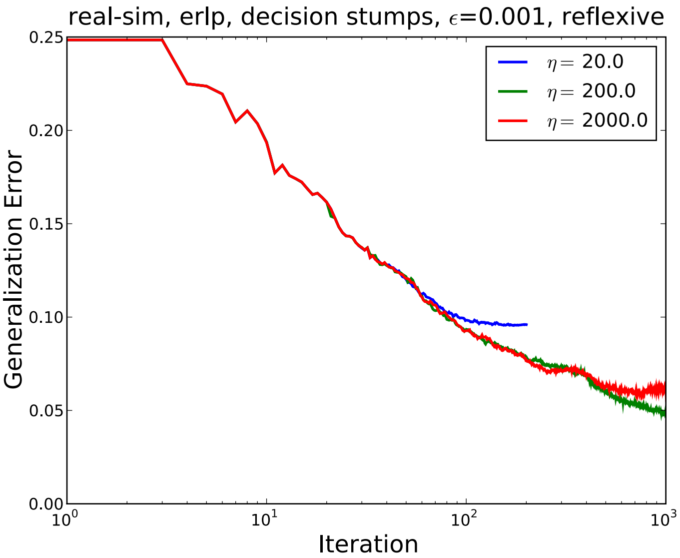
ERLPBoost fixes the problem



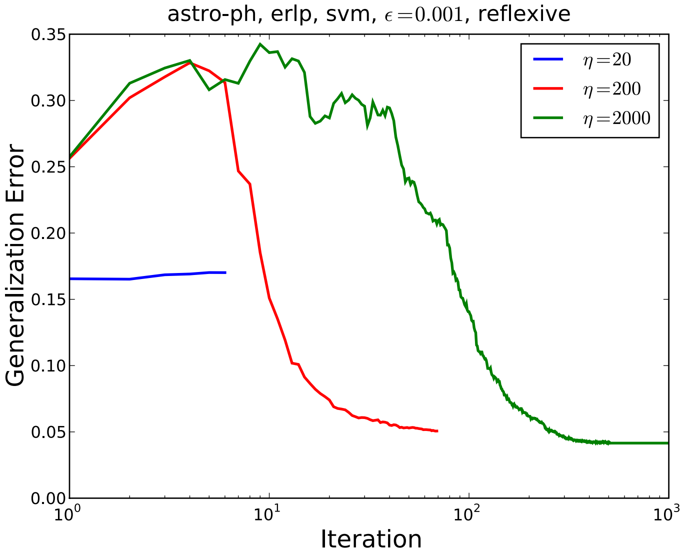
ERLPBoost fixes the problem



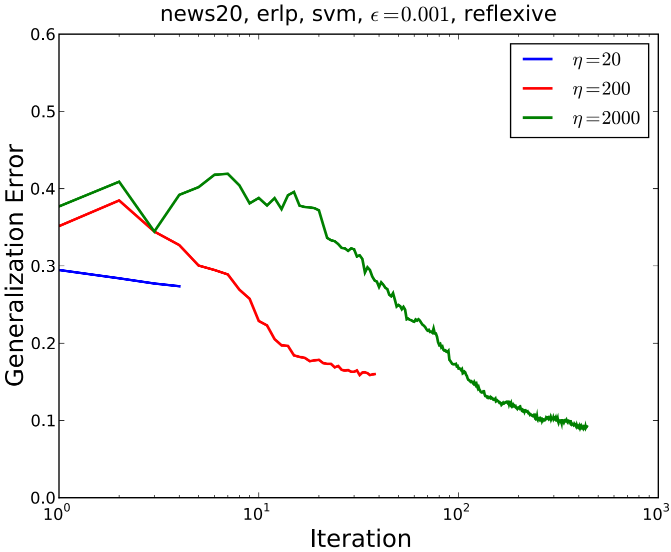
ERLPBoost fixes the problem



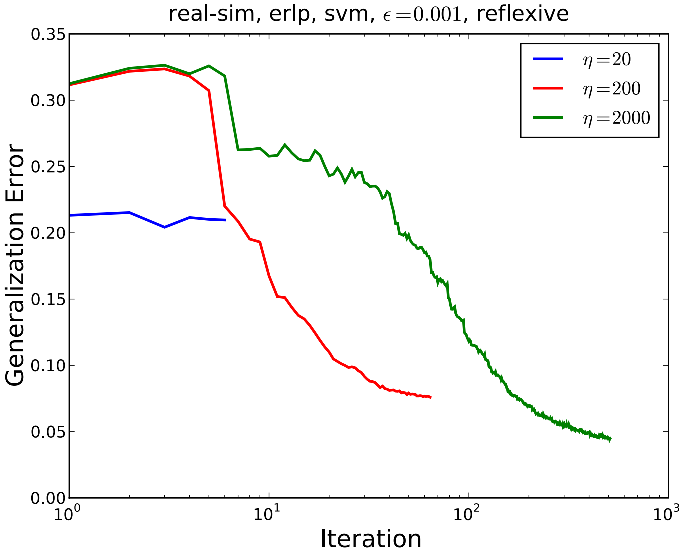
ERLPBoost fixes the problem



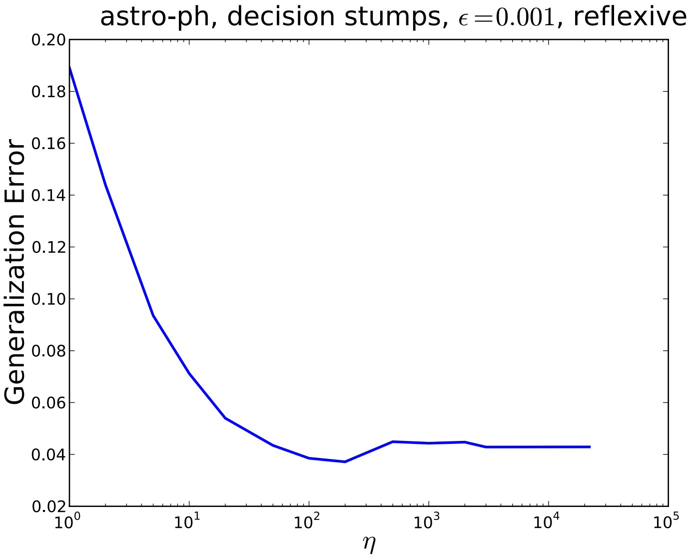
ERLPBoost fixes the problem



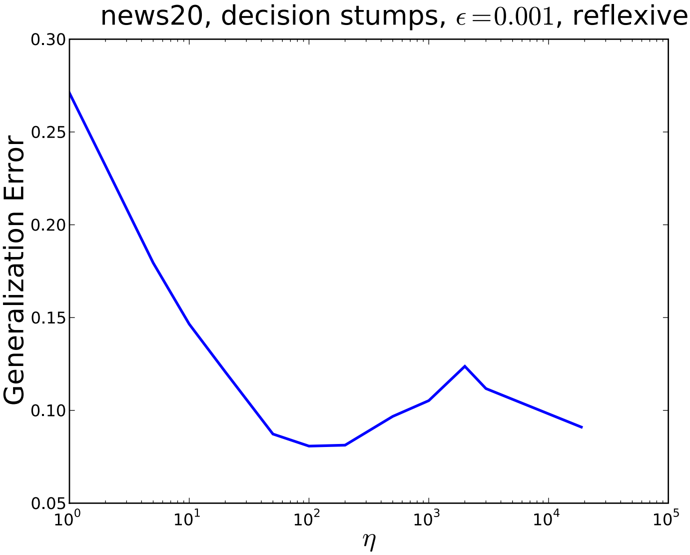
ERLPBoost fixes the problem



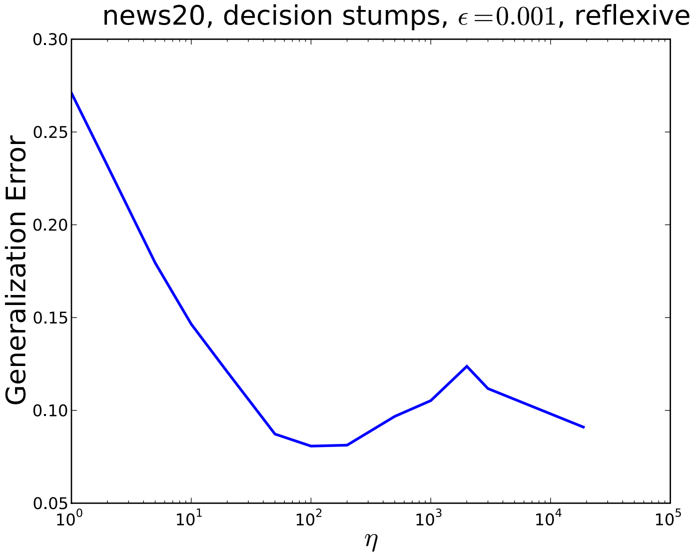
Generalization as a function of η



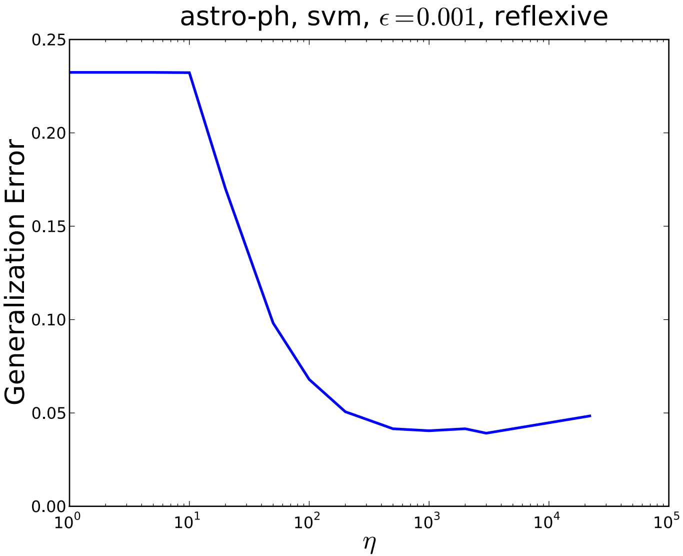
Generalization as a function of η



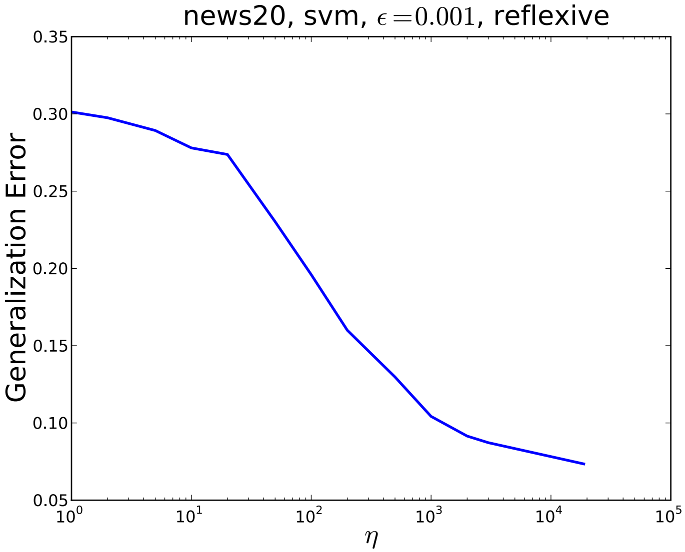
Generalization as a function of η



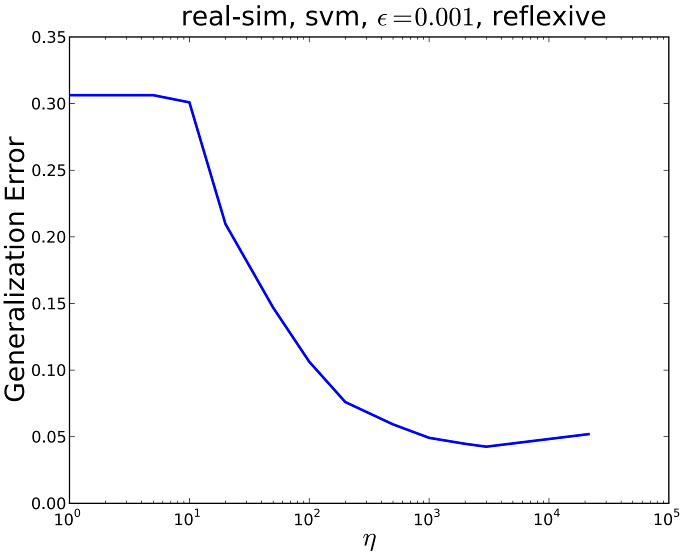
Generalization as a function of η



Generalization as a function of η

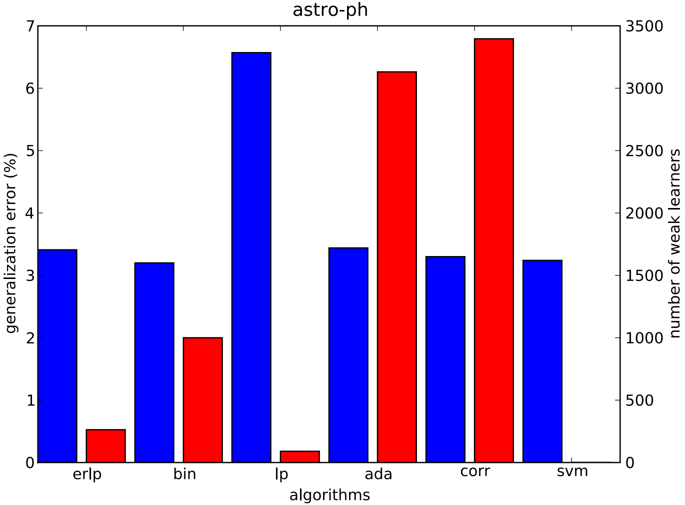


Generalization as a function of η



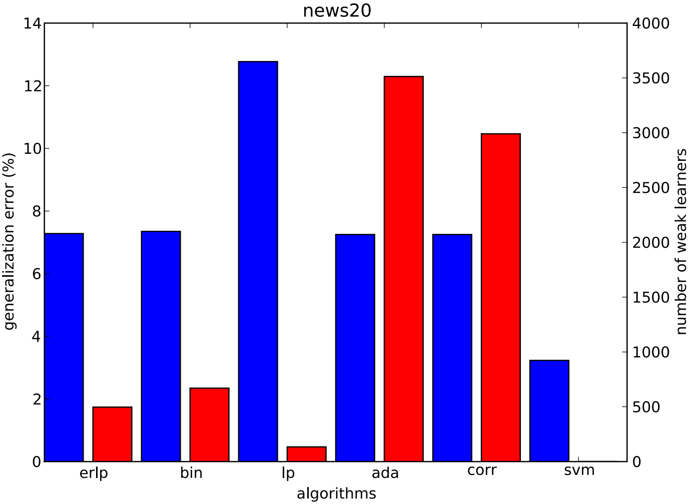
Generalization Error and # of Weak Hypothesis

Parameters tuned for best generalization performance



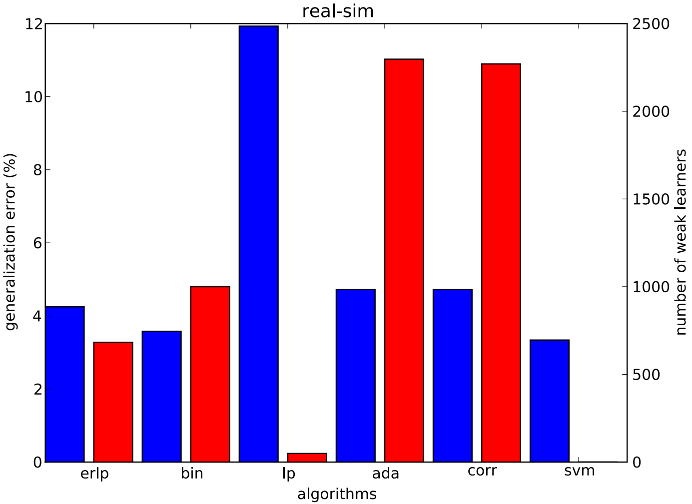
Generalization Error and # of Weak Hypothesis

Parameters tuned for best generalization performance

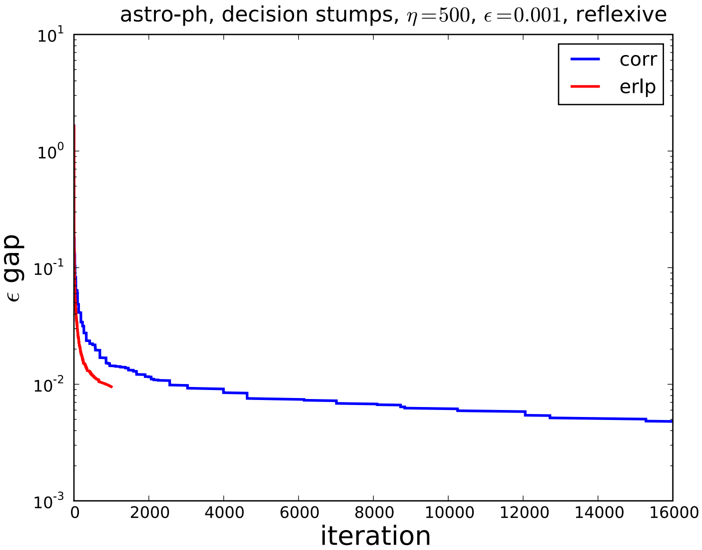


Generalization Error and # of Weak Hypothesis

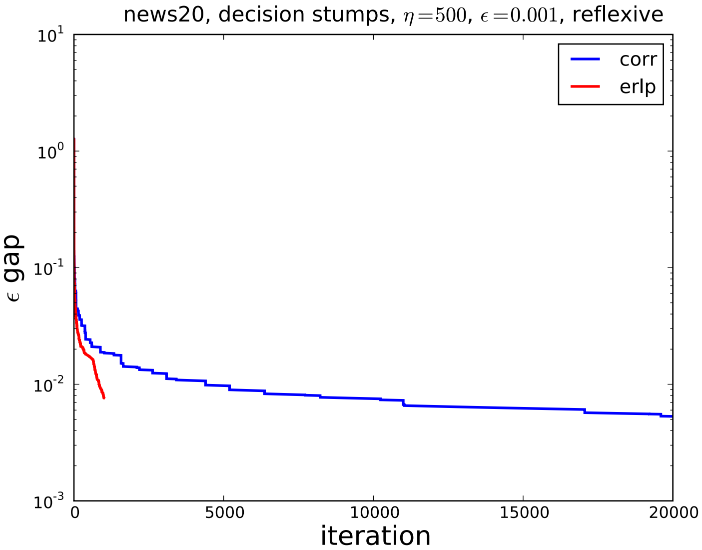
Parameters tuned for best generalization performance



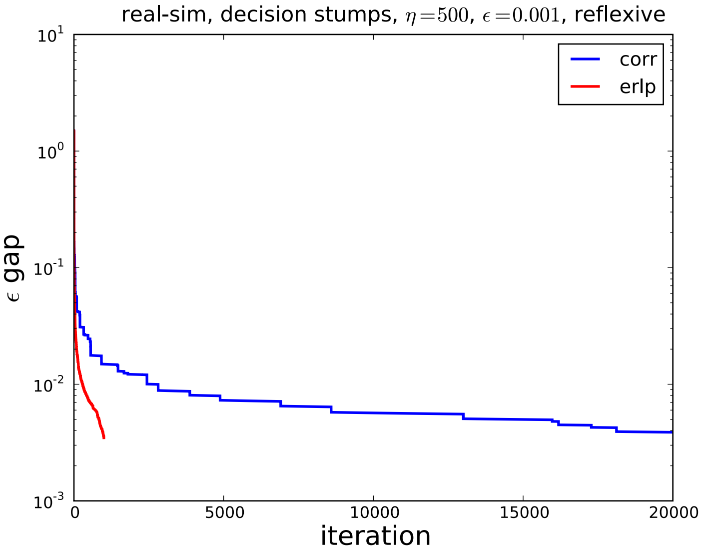
Corrective vs Totally Corrective



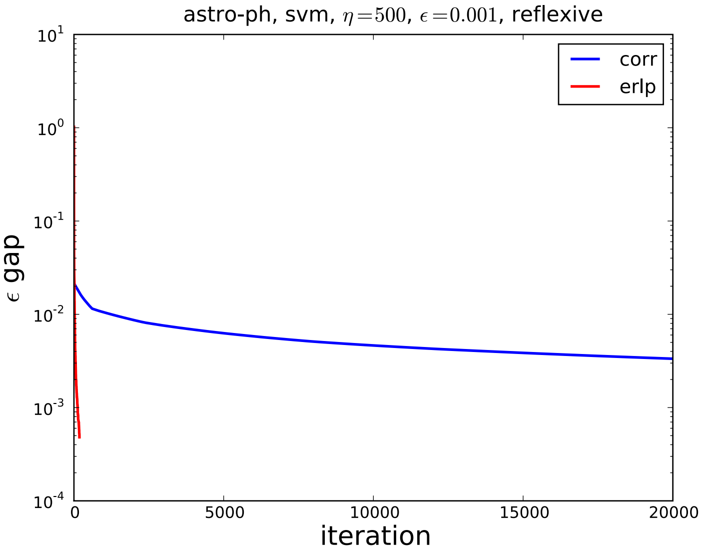
Corrective vs Totally Corrective



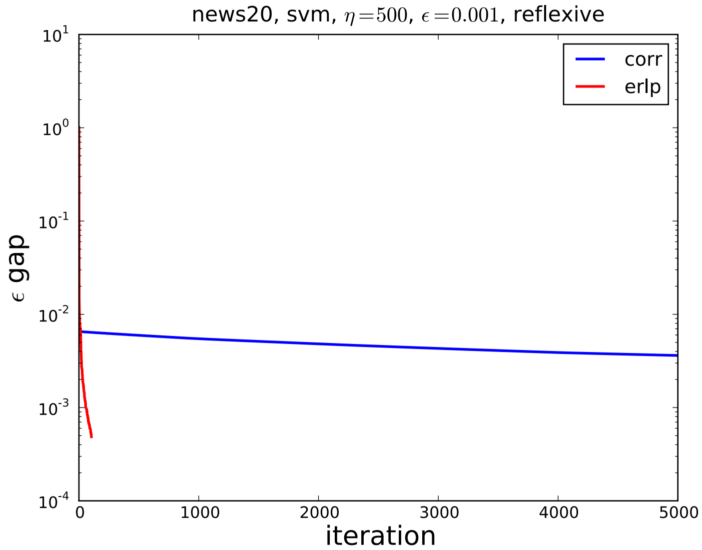
Corrective vs Totally Corrective



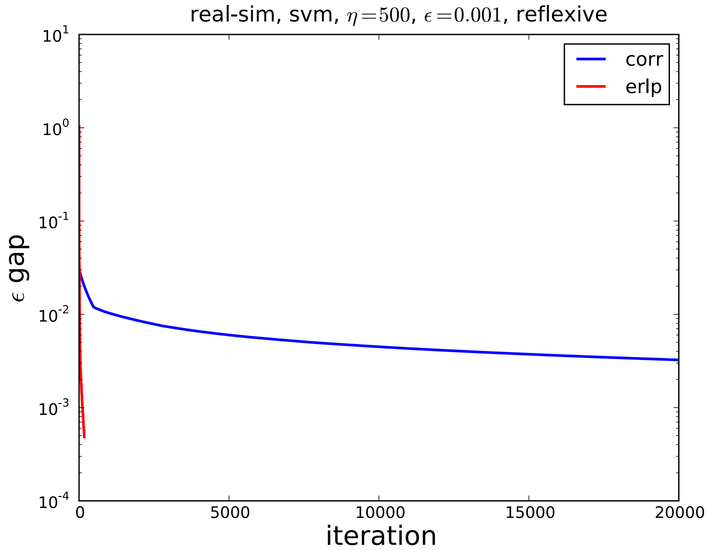
Corrective vs Totally Corrective



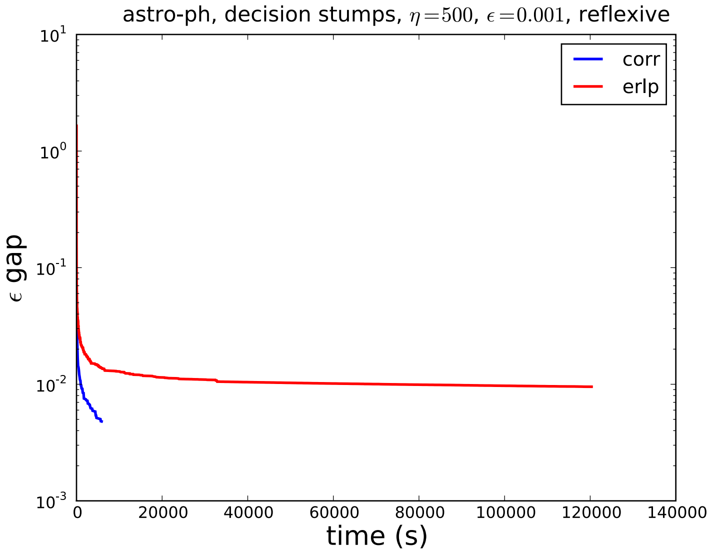
Corrective vs Totally Corrective



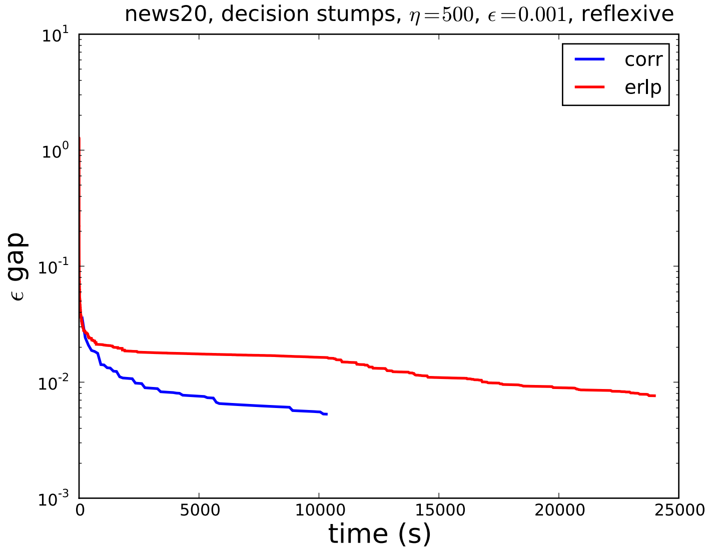
Corrective vs Totally Corrective



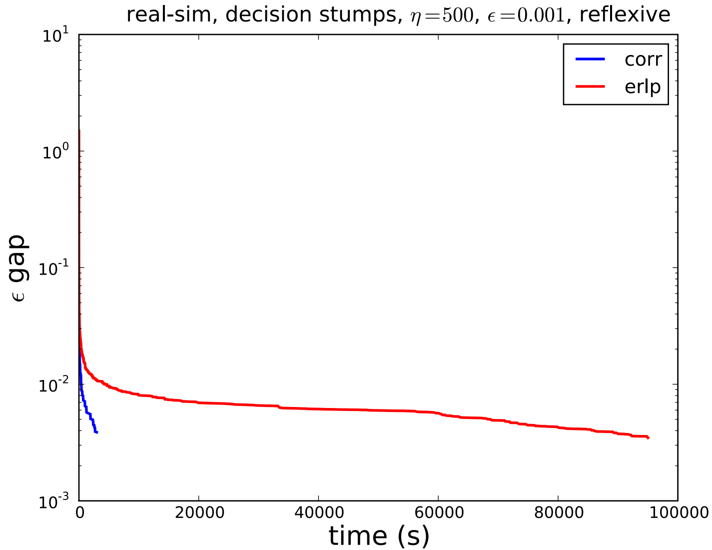
Corrective vs Totally Corrective contd.



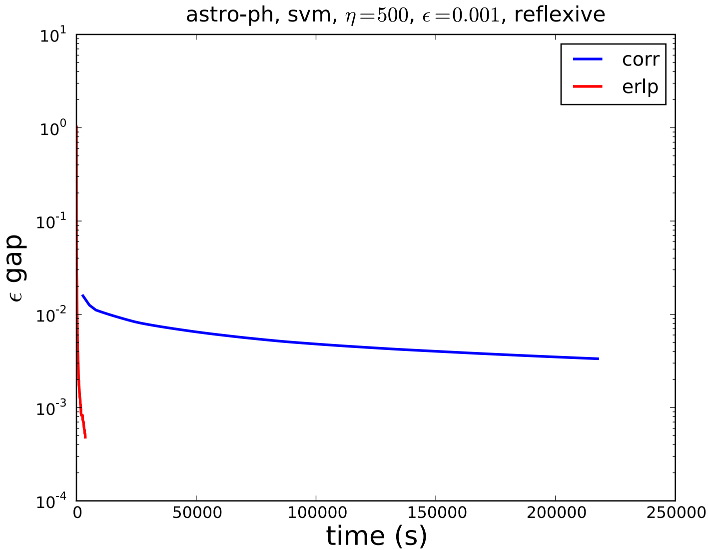
Corrective vs Totally Corrective contd.



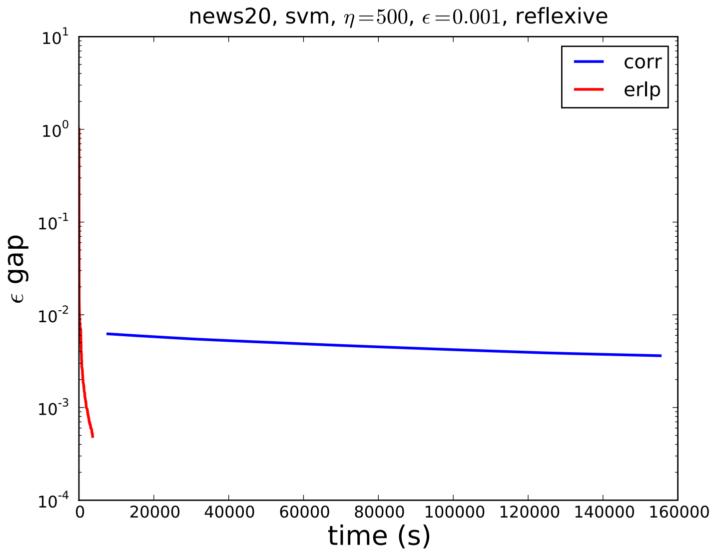
Corrective vs Totally Corrective contd.



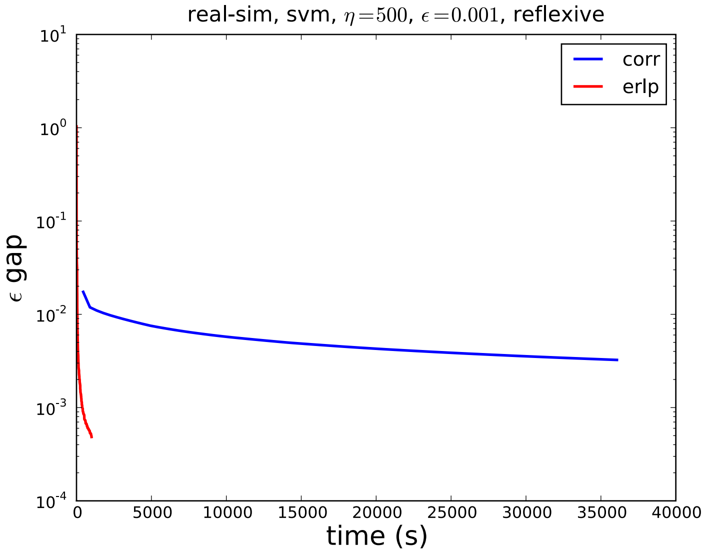
Corrective vs Totally Corrective contd.



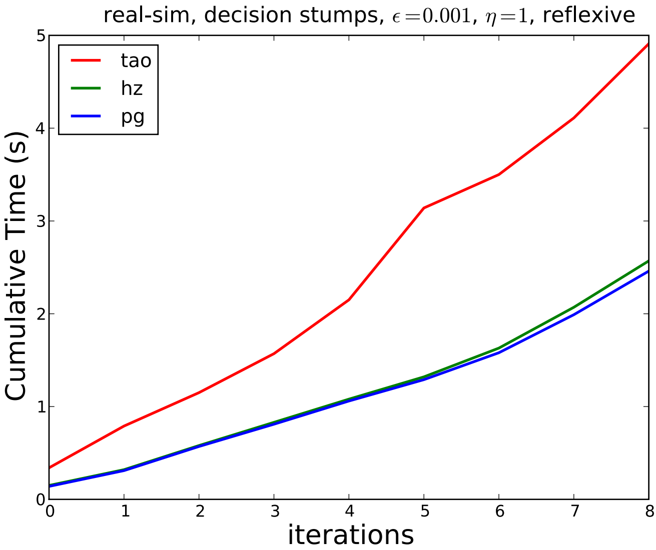
Corrective vs Totally Corrective contd.



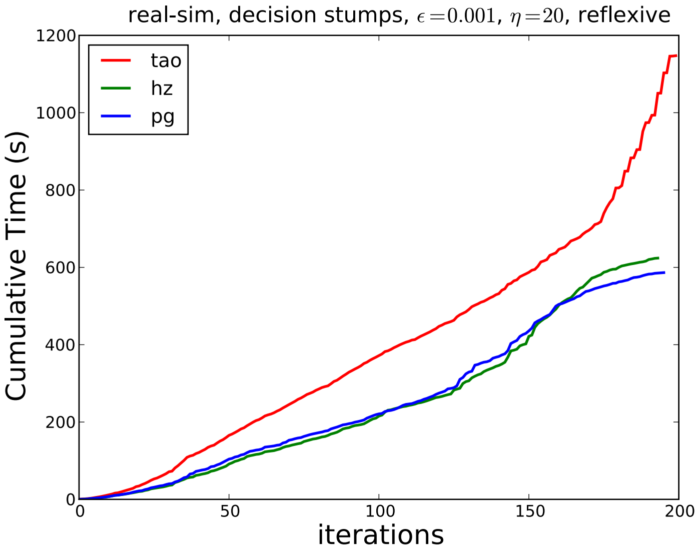
Corrective vs Totally Corrective contd.



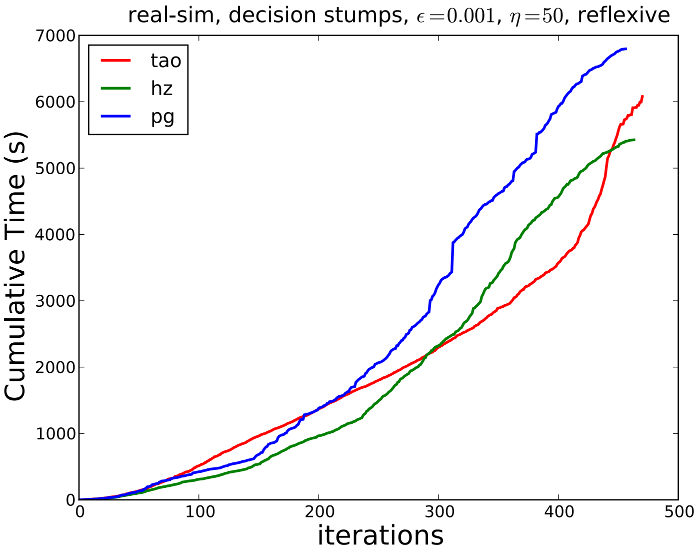
Comparing Different Optimizers



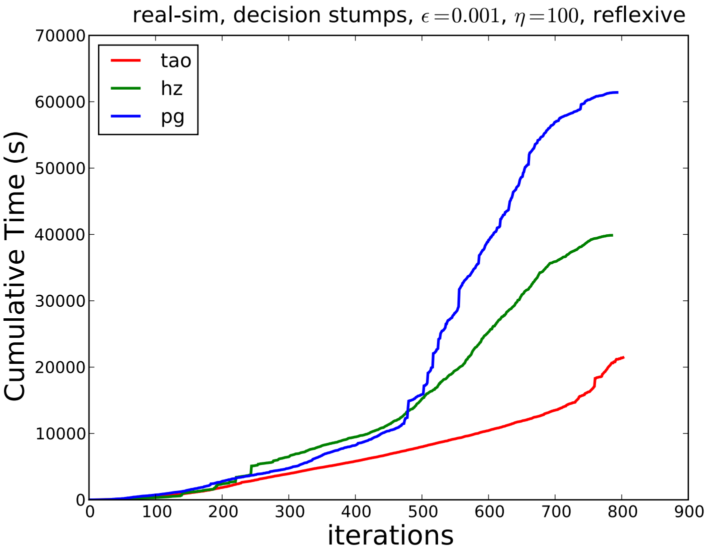
Comparing Different Optimizers



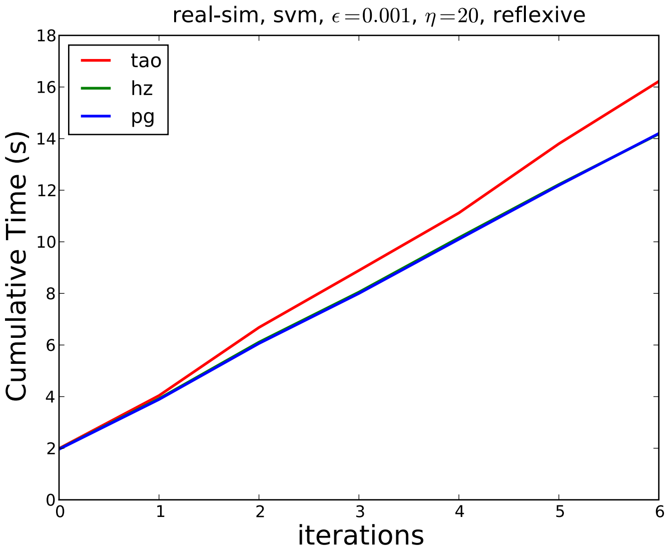
Comparing Different Optimizers



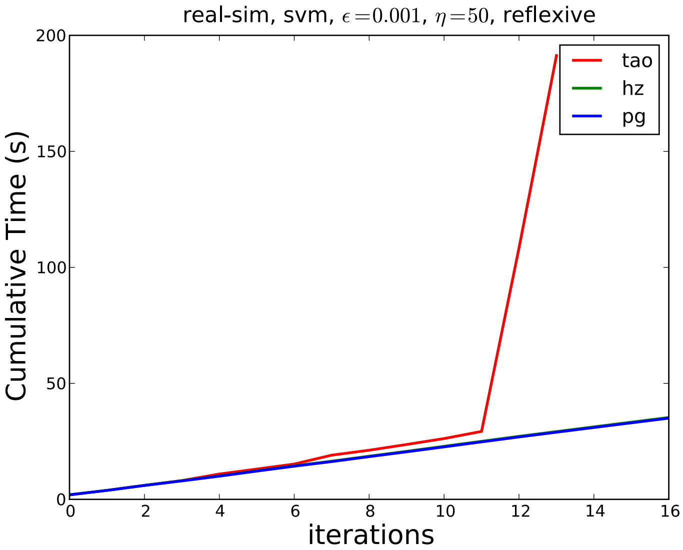
Comparing Different Optimizers



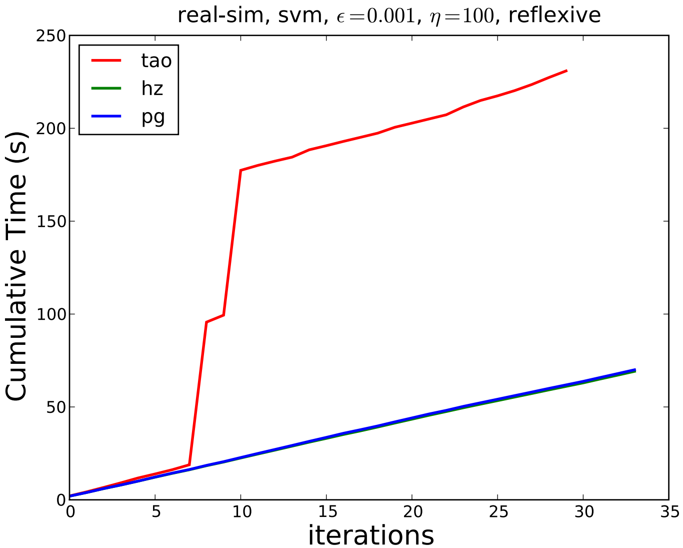
Comparing Different Optimizers



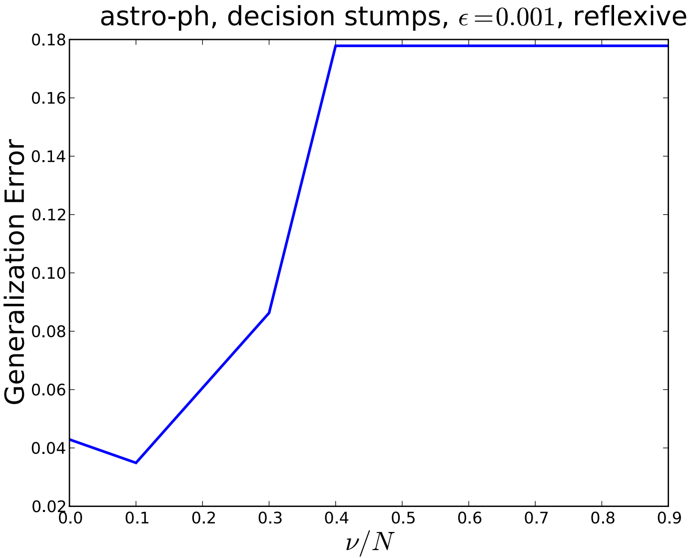
Comparing Different Optimizers



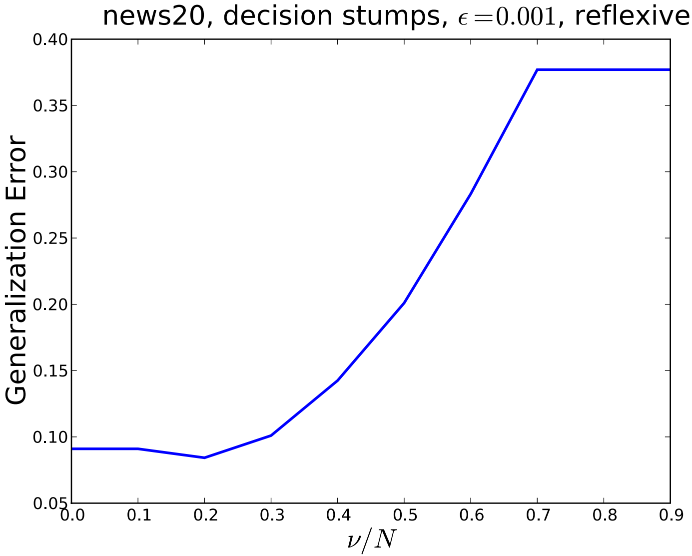
Comparing Different Optimizers



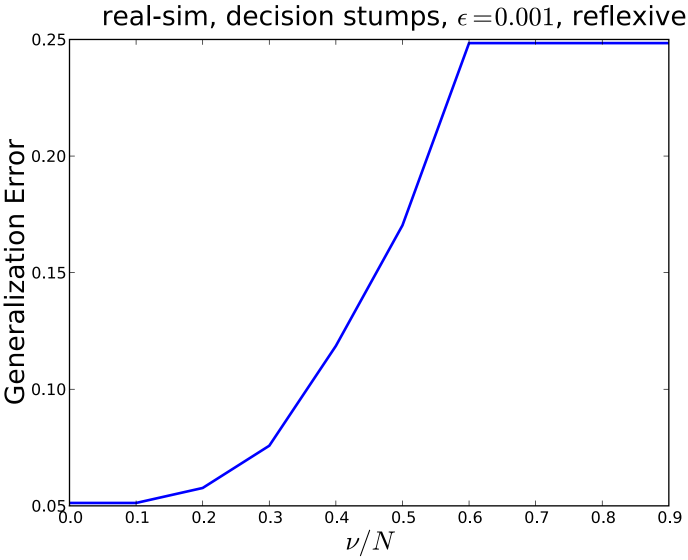
Effect of ν


astro-ph, decision stumps, $\epsilon = 0.001$, reflexive
Generalization Error
0.02
0.04
0.06
0.08
0.10
0.12
0.14
0.16
0.18
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
ν/N

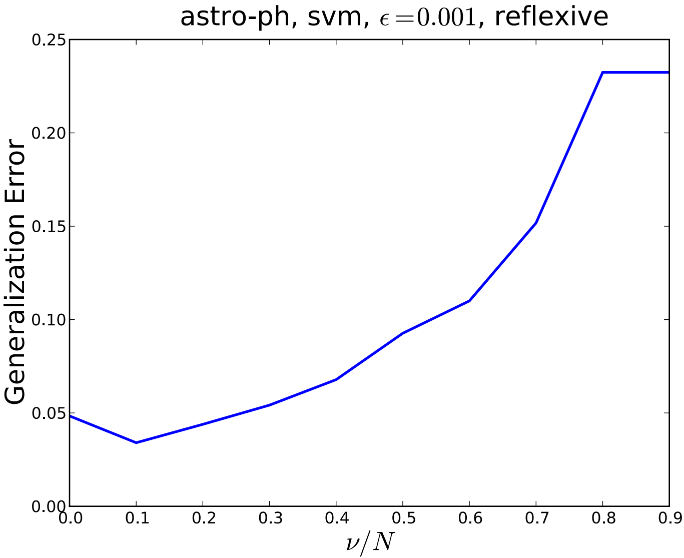
Effect of ν


news20, decision stumps, $\epsilon = 0.001$, reflexive
Generalization Error
0.40
0.35
0.30
0.25
0.20
0.15
0.10
0.05
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
ν/N

Effect of ν

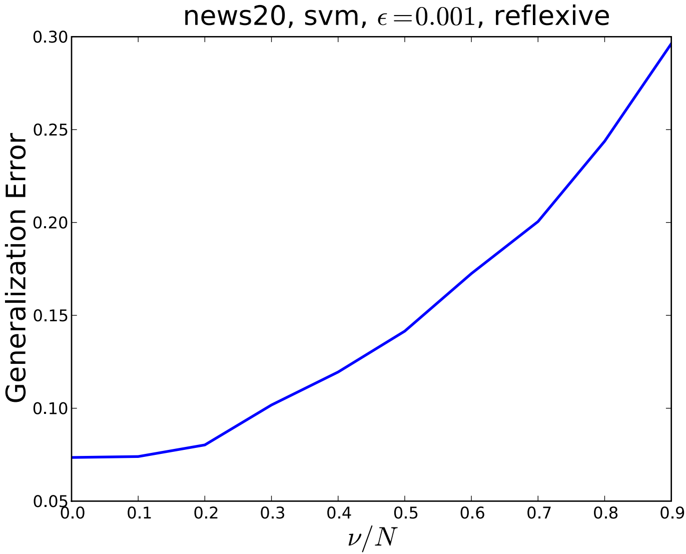

real-sim, decision stumps, $\epsilon = 0.001$, reflexive
Generalization Error
0.25
0.20
0.15
0.10
0.05
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
ν/N

Effect of ν

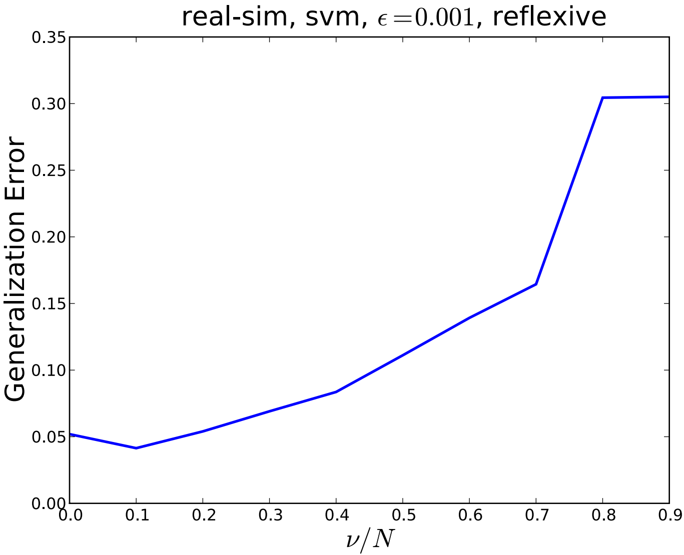


astro-ph, svm, $\epsilon = 0.001$, reflexive
Generalization Error
0.25
0.20
0.15
0.10
0.05
0.00
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
ν/N

Effect of ν


news20, svm, $\epsilon = 0.001$, reflexive
Generalization Error
0.30
0.25
0.20
0.15
0.10
0.05
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
ν/N

Effect of ν


real-sim, svm, $\epsilon = 0.001$, reflexive
Generalization Error
0.00
0.05
0.10
0.15
0.20
0.25
0.30
0.35
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
ν/N

SVMs vs Boosting

SVMs

- Simple Quadratic Optimization Problem
- A decade of experience
- Can handle $\approx 10^6$ points

Boosting

- Softmax problem (log, exp, and simplex constraints)
- Just starting out
- Can handle $\approx 10^4$ points

Our Code

- Freely available under the MPL
- Scaling to large datasets (lots of room for improvement)
- Ideas on how to prune weak learners
- Still looking for large datasets where boosting shines . . .

SVMs vs Boosting

SVMs

- Simple Quadratic Optimization Problem
- A decade of experience
- Can handle $\approx 10^6$ points

Boosting

- Softmax problem (log, exp, and simplex constraints)
- Just starting out
- Can handle $\approx 10^4$ points

Our Code

- Freely available under the MPL
- Scaling to large datasets (lots of room for improvement)
- Ideas on how to prune weak learners
- Still looking for large datasets where boosting shines . . .

SVMs vs Boosting

SVMs

- Simple Quadratic Optimization Problem
- A decade of experience
- Can handle $\approx 10^6$ points

Boosting

- Softmax problem (log, exp, and simplex constraints)
- Just starting out
- Can handle $\approx 10^4$ points

Our Code

- Freely available under the MPL
- Scaling to large datasets (lots of room for improvement)
- Ideas on how to prune weak learners
- Still looking for large datasets where boosting shines ...

SVMs vs Boosting

SVMs

- Simple Quadratic Optimization Problem
- A decade of experience
- Can handle $\approx 10^6$ points

Boosting

- Softmax problem (log, exp, and simplex constraints)
- Just starting out
- Can handle $\approx 10^4$ points

Our Code

- Freely available under the MPL
- Scaling to large datasets (lots of room for improvement)
- Ideas on how to prune weak learners
- Still looking for large datasets where boosting shines ...

SVMs vs Boosting

SVMs

- Simple Quadratic Optimization Problem
- A decade of experience
- Can handle $\approx 10^6$ points

Boosting

- Softmax problem (log, exp, and simplex constraints)
- Just starting out
- Can handle $\approx 10^4$ points

Our Code

- Freely available under the MPL
- Scaling to large datasets (lots of room for improvement)
- Ideas on how to prune weak learners
- Still looking for large datasets where boosting shines . . .

SVMs vs Boosting

SVMs

- Simple Quadratic Optimization Problem
- A decade of experience
- Can handle $\approx 10^6$ points

Boosting

- Softmax problem (log, exp, and simplex constraints)
- Just starting out
- Can handle $\approx 10^4$ points

Our Code

- Freely available under the MPL
- Scaling to large datasets (lots of room for improvement)
- Ideas on how to prune weak learners
- Still looking for large datasets where boosting shines . . .

SVMs vs Boosting

SVMs

- Simple Quadratic Optimization Problem
- A decade of experience
- Can handle $\approx 10^6$ points

Boosting

- Softmax problem (log, exp, and simplex constraints)
- Just starting out
- Can handle $\approx 10^4$ points

Our Code

- Freely available under the MPL
- Scaling to large datasets (lots of room for improvement)
- Ideas on how to prune weak learners
- Still looking for large datasets where boosting shines . . .

SVMs vs Boosting

SVMs

- Simple Quadratic Optimization Problem
- A decade of experience
- Can handle $\approx 10^6$ points

Boosting

- Softmax problem (log, exp, and simplex constraints)
- Just starting out
- Can handle $\approx 10^4$ points

Our Code

- Freely available under the MPL
- Scaling to large datasets (lots of room for improvement)
- Ideas on how to prune weak learners
- Still looking for large datasets where boosting shines . . .

Conclusion

- Lots of exciting connections between boosting and optimization (we are only scratching the surface)
- Bring entropic regularization algorithms up to par with squared Euclidean distance regularization
- Look for datasets that exploit merits of new algorithms
- Find artificial datasets that highlight advantages of different families of algorithms
- Better lower bounds (the case of the missing log n)

Vishy's acknowledgements

- Manfred Warmuth for teaching me about boosting
- Karen Glocer for helping with the plots and code
- Ankan Saha, Choon Hui Teo, Jin Yu, and Xinhua Zhang for technical discussions