Game Theory and Boosting

Manfred K. Warmuth

University of California, Santa Cruz

Game Theory Class, UCSC, March 3, 2009

Joint work and help from
Gunnar Rätsch, Vishy Vishwanathan, Karen Glocer

Outline

Rock, Paper, Scissors game

		column player		
		R	P	S
		α_1	α_2	α_3
row player R	d_1	0	1	-1
P	d_2	-1	0	1
S	d_3	1	-1	0

payoff matrix

$$\begin{aligned} \text{payoff} &= \mathbf{d}^T M \boldsymbol{\alpha} \\ &= \sum_{i,j} d_i M_{i,j} \alpha_j \end{aligned}$$

Two-player Zero Sum Game

- Gains of row player = losses of column player
  row player minimizes, column player maximizes
- Single row is pure strategy of row player and
  $\mathbf{d}$ is mixed strategy
- Single column is pure strategy of column player and
  $\boldsymbol{\alpha}$ is mixed strategy

Optimum Strategy

			R	P	S
			α_1	α_2	α_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

- Min-max theorem: [Van Neumann 1928]

$$\min_{\mathbf{d}} \max_{\boldsymbol{\alpha}} \mathbf{d}^T M \boldsymbol{\alpha} = \max_{\boldsymbol{\alpha}} \min_{\mathbf{d}} \mathbf{d}^T M \boldsymbol{\alpha}$$

$$= \text{value of the game (0 in example)}$$

Pure strategies

- $\mathbf{e}_i$ pure strategy of row player
  $\mathbf{e}_j$ pure strategies of column player
- Inner strategy can be pure

$$\min_{\mathbf{d}} \max_{\boldsymbol{\alpha}} \mathbf{d}^T M \boldsymbol{\alpha} = \min_{\mathbf{d}} \max_{j} \mathbf{d}^T M \mathbf{e}_j$$

$$\max_{\boldsymbol{\alpha}} \min_{\mathbf{d}} \mathbf{d}^T M \boldsymbol{\alpha} = \max_{\boldsymbol{\alpha}} \min_{i} \mathbf{e}_i{}^\top M \boldsymbol{\alpha}$$

All equal value of game

New column added

			R	P	S		
			α_1	α_2	α_3	α_4	$\min_i$
			.44	0	.22	.33	
R	d_1	.22	0	1	-1	1	.11
P	d_2	.33	-1	0	1	1	.11
S	d_3	.44	1	-1	0	-1	.11
	$\max_j$		.11	-.22	.11	.11	

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

Row added

			R	P	S	
			α_1	α_2	α_3	$\min_i$
			.33	.44	.22	
R	d_1	0	0	1	-1	.22
P	d_2	.22	-1	0	1	-.11
S	d_3	.44	1	-1	0	-.11
	d_4	.33	-1	1	-1	-.11
	$\max_j$		-.11	-.11	-.11	

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

Incremental games

Column adding game (Boosting) [FS]

- Column player has large pool of columns available
- In each iteration one is added

	α_1^1
d_1^1	0
d_2^1	1
d_3^1	-1

iteration 1

	α_1^2	α_2^2
d_1^2	0	-1
d_2^2	1	0
d_3^2	-1	1

iteration 2

	α_1^3	α_2^3	α_3^3
d_1^3	0	-1	1
d_2^3	1	0	-1
d_3^3	-1	1	0

iteration 3

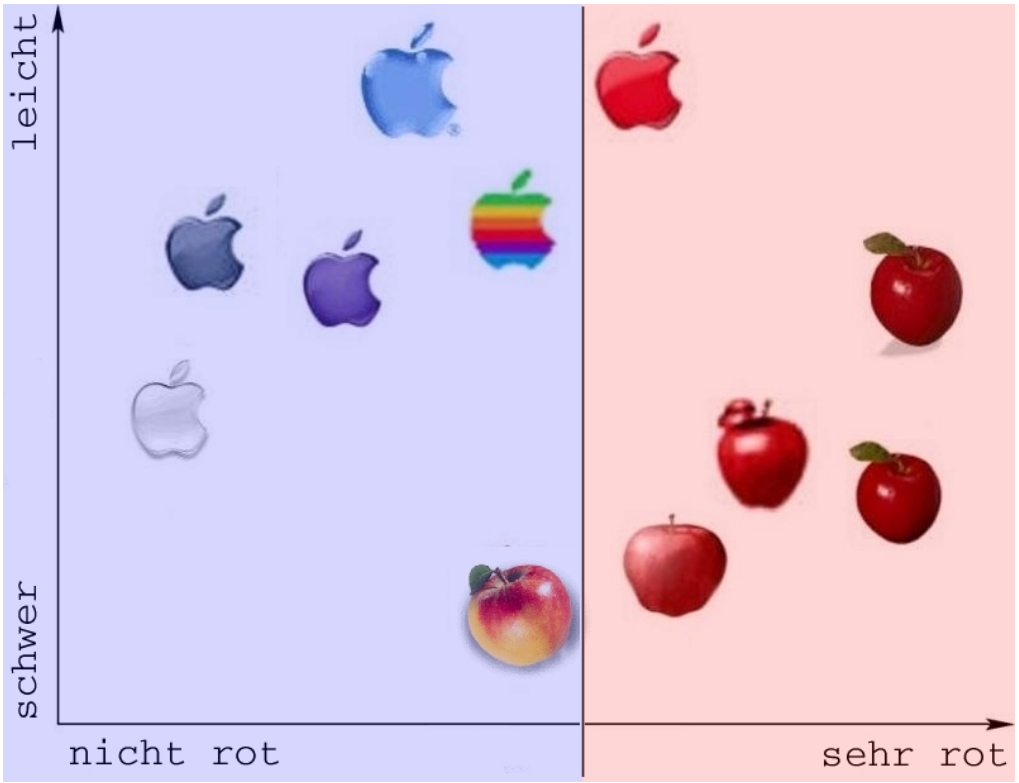
- In each iteration solve optimization problem to update $\mathbf{d}$
- Column player always picks column which has largest edge wrt the current $\mathbf{d}$

Desired Properties of Algorithm

- Want an algorithm that makes the value increase as quickly as possible
- Final game matrix should have value not too much smaller than optimum
- Number of columns needs should be as small as possible

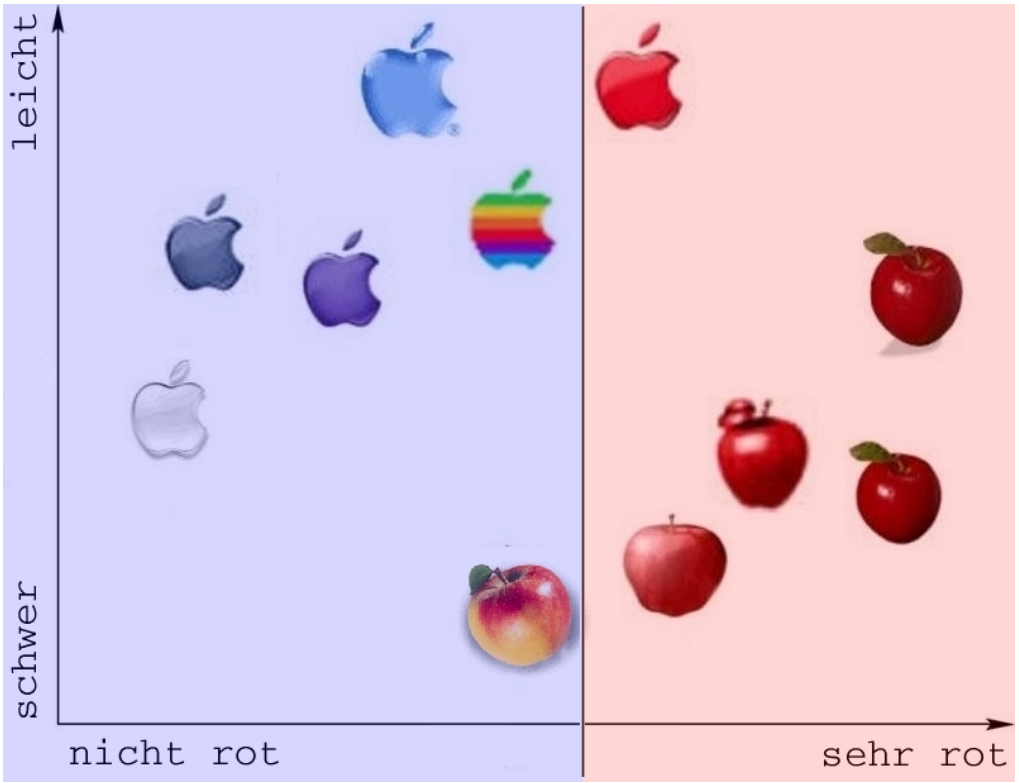
Outline

Setup for Boosting



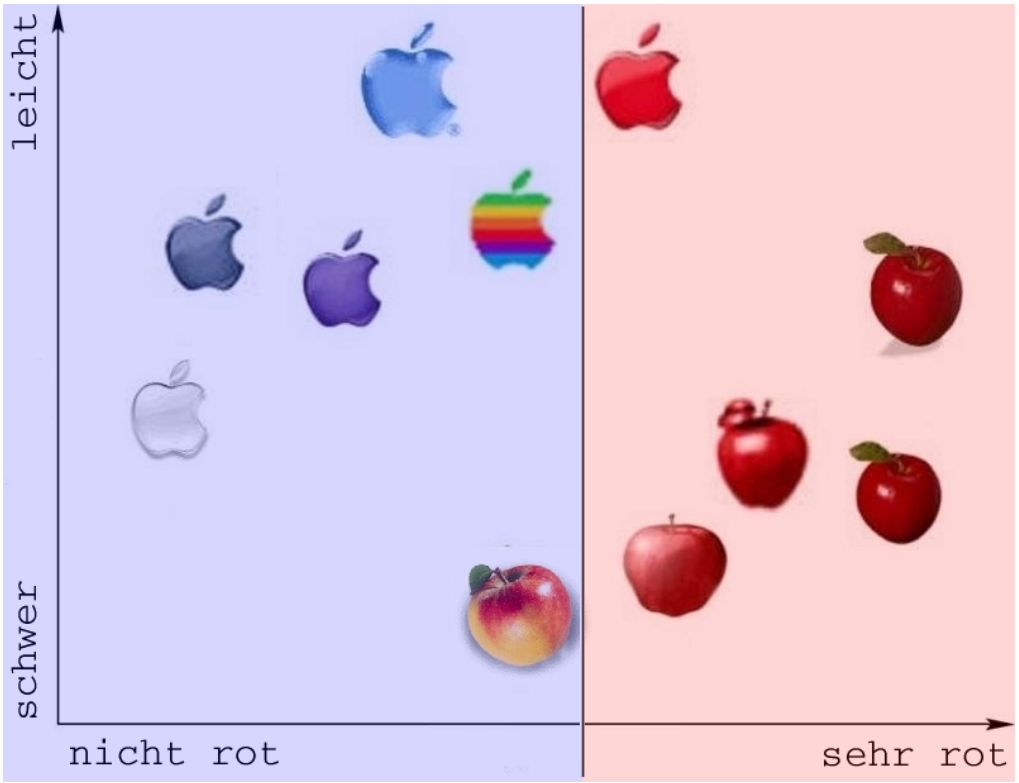
- 11 apples (examples)
- labeled +1 if natural and -1 if artificial
- want to classify the apples
- given weak hypotheses decision stumps along the two features
- examples = rows
- weak hypotheses = possible columns

Setup for Boosting



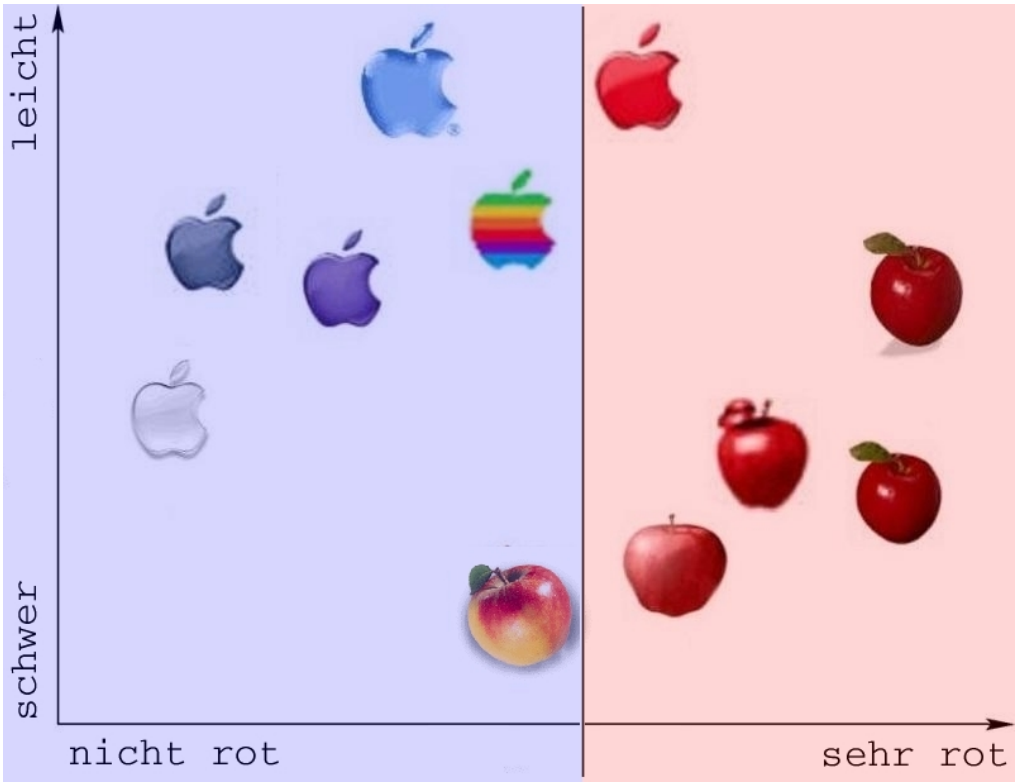
- 11 apples (examples)
- labeled +1 if natural and -1 if artificial
- want to classify the apples
- given weak hypotheses decision stumps along the two features
- examples = rows
- weak hypotheses = possible columns

Setup for Boosting



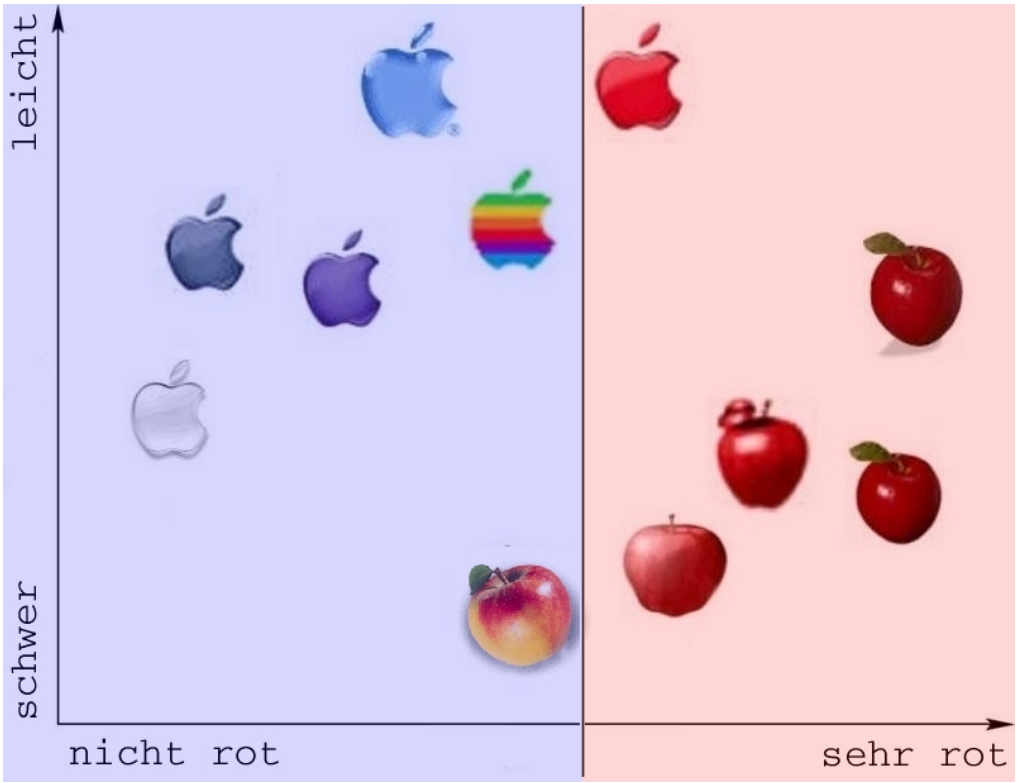
- 11 apples (examples)
- labeled +1 if natural and -1 if artificial
- want to classify the apples
- given weak hypotheses decision stumps along the two features
- examples = rows
- weak hypotheses = possible columns

Setup for Boosting



- 11 apples (examples)
- labeled +1 if natural and -1 if artificial
- want to classify the apples
- given weak hypotheses decision stumps along the two features
- examples = rows
- weak hypotheses = possible columns

Setup for Boosting

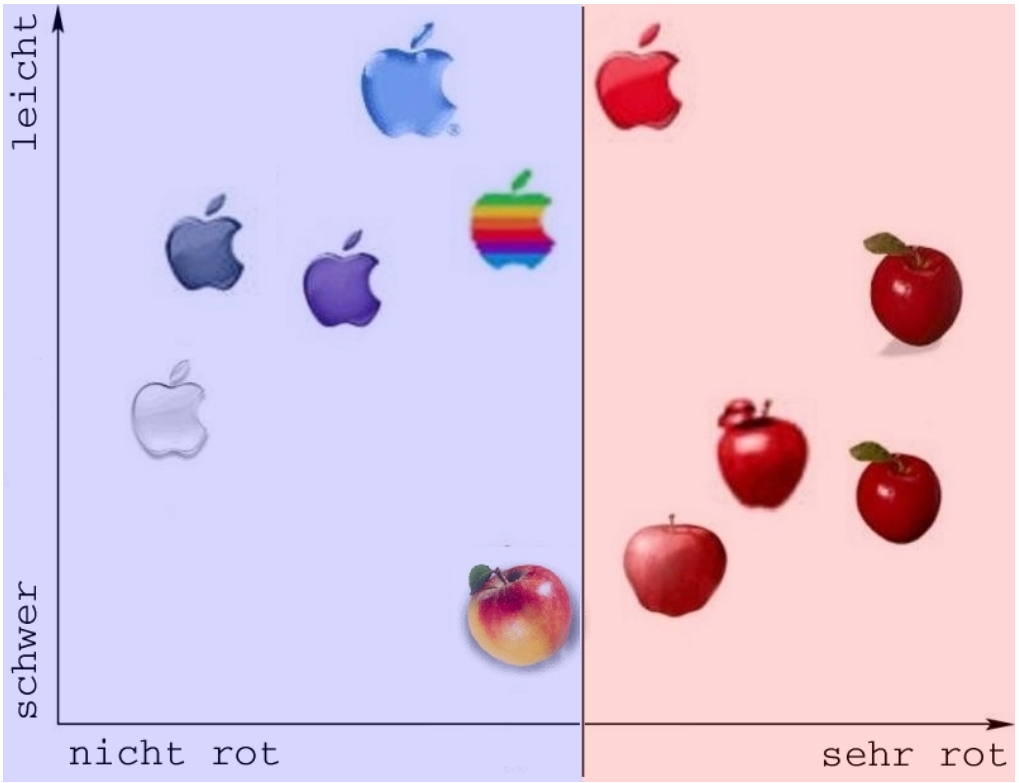


- 11 apples (examples)
- labeled $+1$ if natural and -1 if artificial
- want to classify the apples
- given weak hypotheses decision stumps along the two features
- examples $=$ rows
- weak hypotheses $=$ possible columns

Examples and Hypotheses

Examples	Labels	h^1: redness	
	-1	-1	
	-1	-1	
	-1	-1	
	-1	1	mistake
	1	1	
	1	1	
	1	1	
	1	-1	mistake

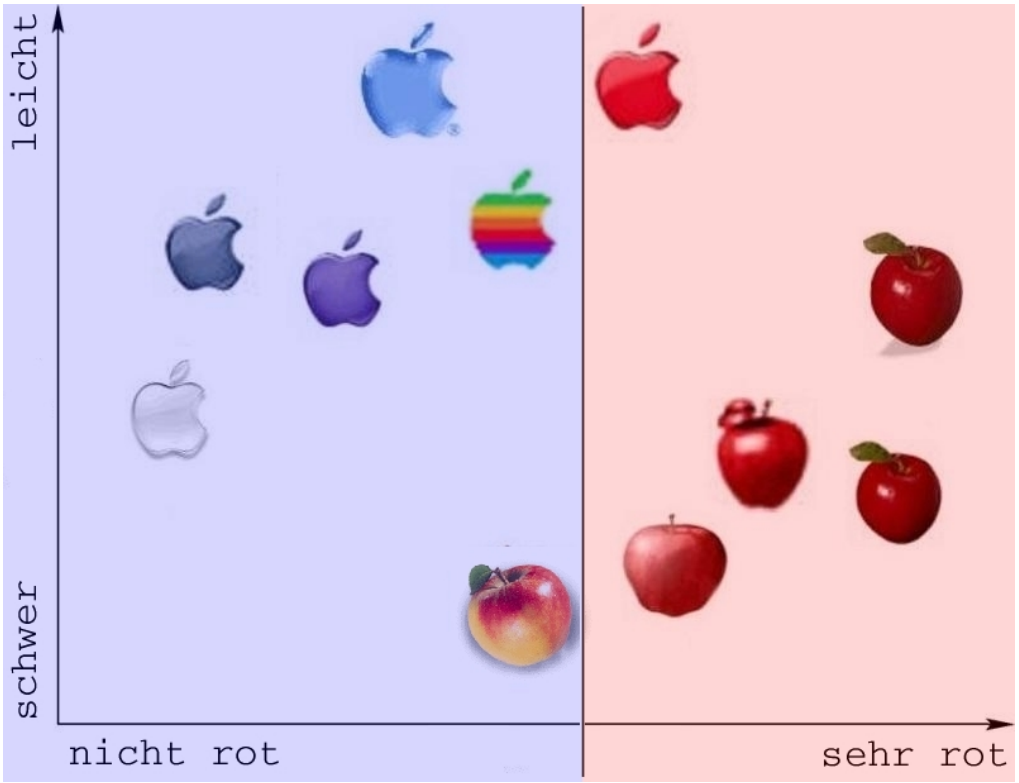
Boosting: 1st Iteration



First hypothesis:

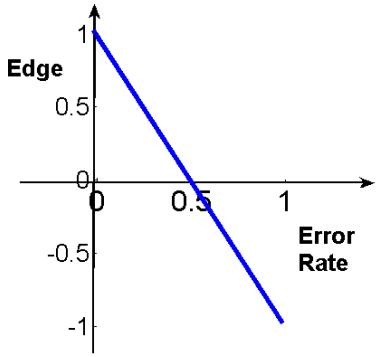
- error rate$_t = \frac{2}{11}$
  $= \sum_{i=1}^{n} d_i^t \, \mathbf{I}(h^t(\mathbf{x}_i) \neq y_i)$
- edge$_t = \frac{9}{22}$
  $= \sum_{i=1}^{n} d_i^t \, y_i h^t(\mathbf{x}_i)$
  $= 1 - 2\epsilon_t$

Boosting: 1st Iteration



First hypothesis:

- error rate$_t = \frac{2}{11}$
  $= \sum_{i=1}^{n} d_i^t \, \mathbf{I}(h^t(\mathbf{x}_i) \neq y_i)$
- edge$_t = \frac{9}{22}$
  $= \sum_{i=1}^{n} d_i^t \, y_i h^t(\mathbf{x}_i)$
  $= 1 - 2\epsilon_t$



Connection to column adding game?

- Rows are the examples (fixed)
- Columns the weak hypotheses
- $M_{i,j} = \underbrace{h^j(\overbrace{\mathbf{x}_i}^{\text{instance}})}_{j\text{-th hypothesis}} \overbrace{y_i}^{\text{label}}$
- Column sum: edge of weak hypothesis
- Row sum: margin of example
- Value of game as large as possible

Game Matrix M

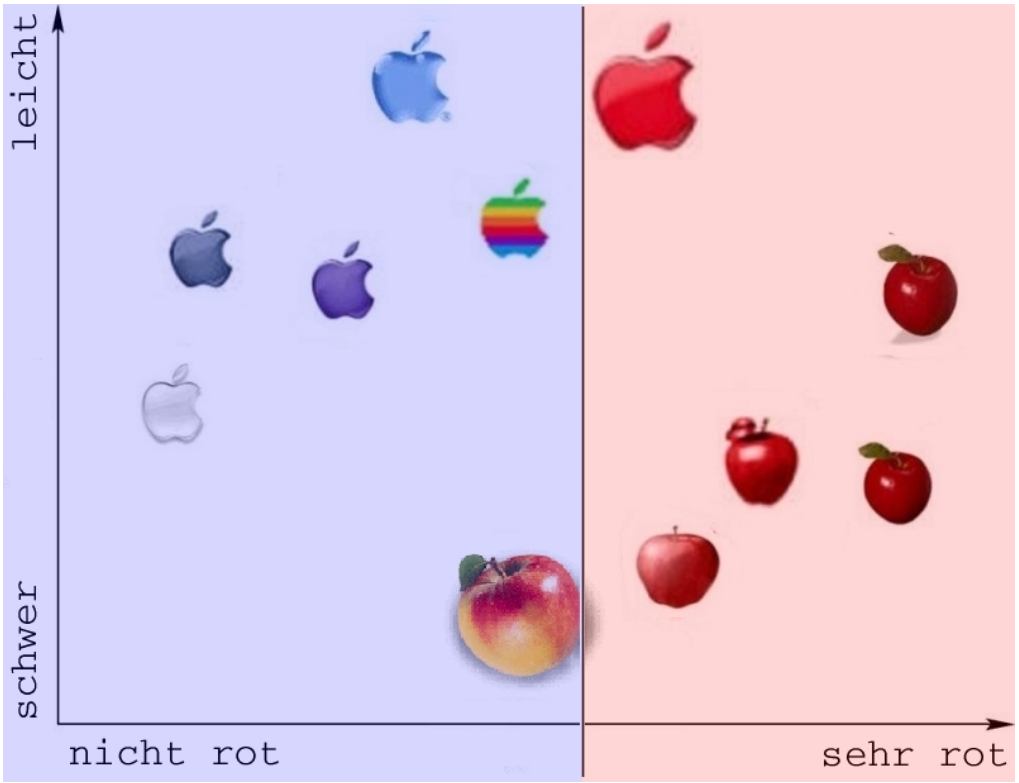
Examples	Labels	h^1: redness	$\mathbf{M_{1,*}}$
	-1	-1	$\mathbf{1}$
	-1	-1	$\mathbf{1}$
	-1	-1	$\mathbf{1}$
	-1	1	$-\mathbf{1}$
	1	1	$\mathbf{1}$
	1	1	$\mathbf{1}$
	1	1	$\mathbf{1}$
	1	-1	$-\mathbf{1}$

Margins and Edges

- The margin of example x_i at iteration t is summing ith row $= \sum_{j=1}^{t} \alpha_j \, y_i h^j(x_i)$
- The edge of hypothesis h^j at iteration t is summing jth column $= \sum_{i=1}^{n} d_i^t \, y_i h^j(x_i)$
- Example:

			h^1	h^2	h^3	
			α_1	α_2	α_3	margin
			.2	.4	.4	
x_1	d_1	.4	1	1	-1	.2
x_2	d_2	.4	-1	0	1	.2
x_3	d_3	.2	1	-1	1	.2
	edge		.2	.2	.2	

Update Distribution

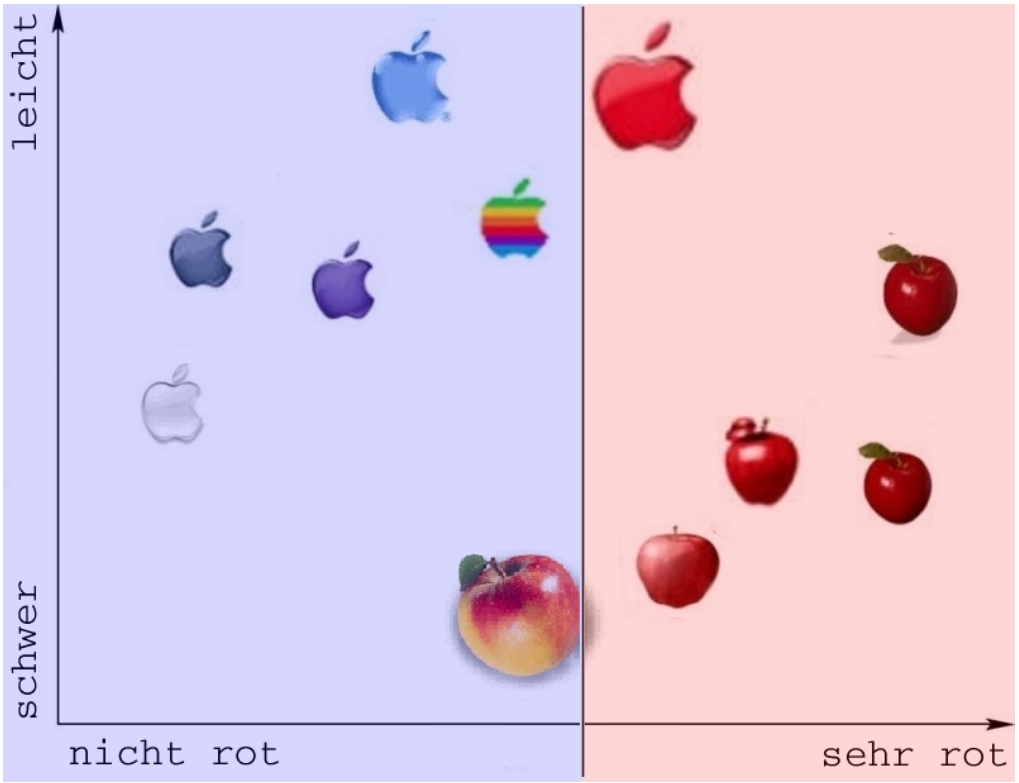


Misclassified examples
$\Rightarrow$ Increased weights

After update:

- Minimum edge small

Update Distribution

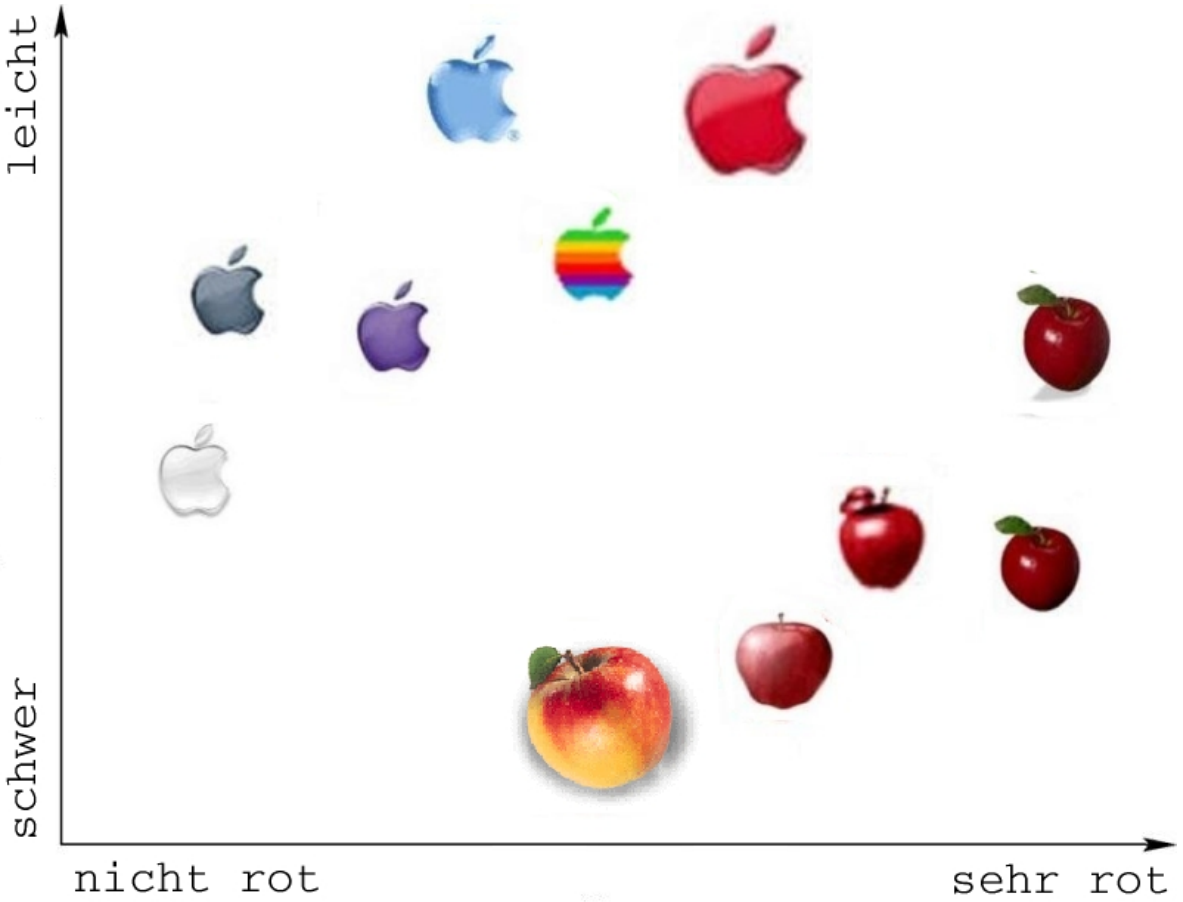


Misclassified examples
$\Rightarrow$ Increased weights

After update:

- Minimum edge small

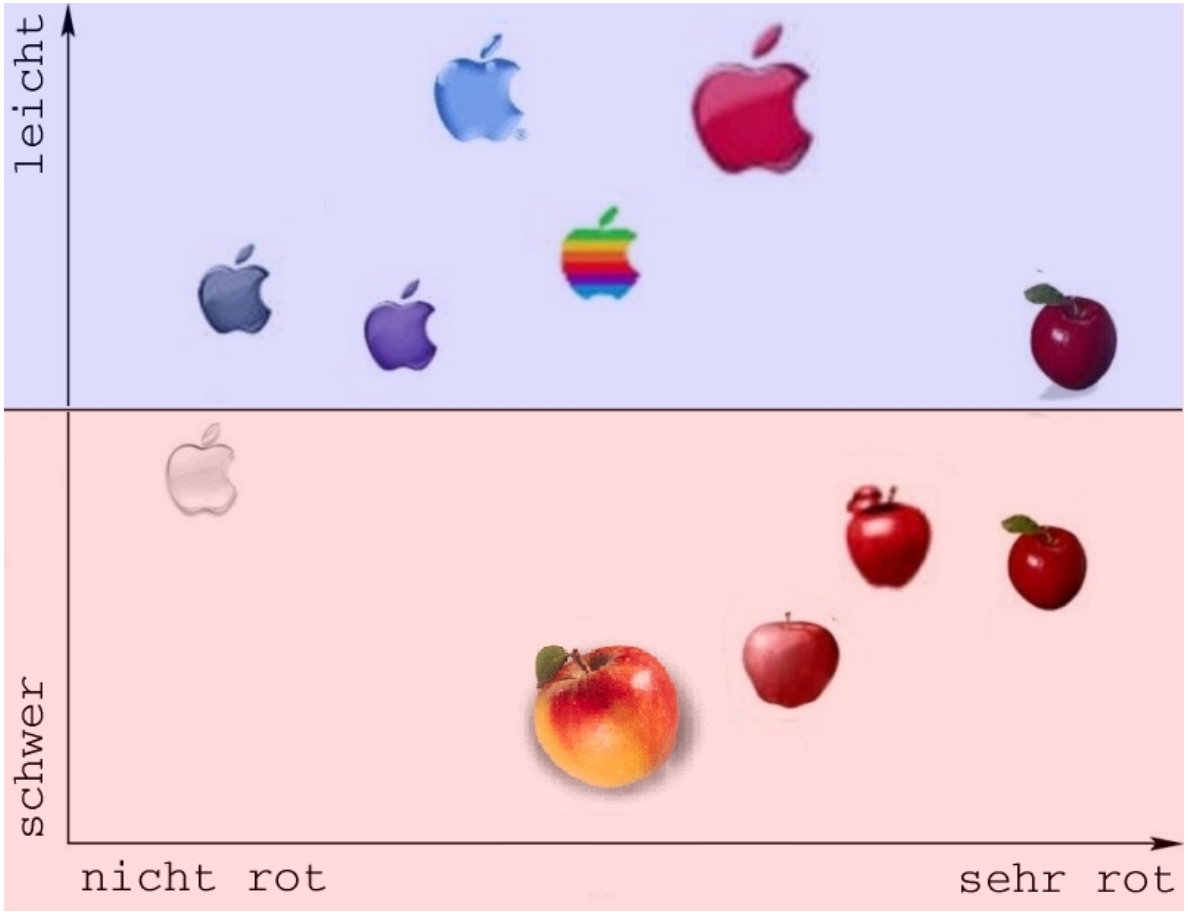
Before 2nd Iteration



Hard examples

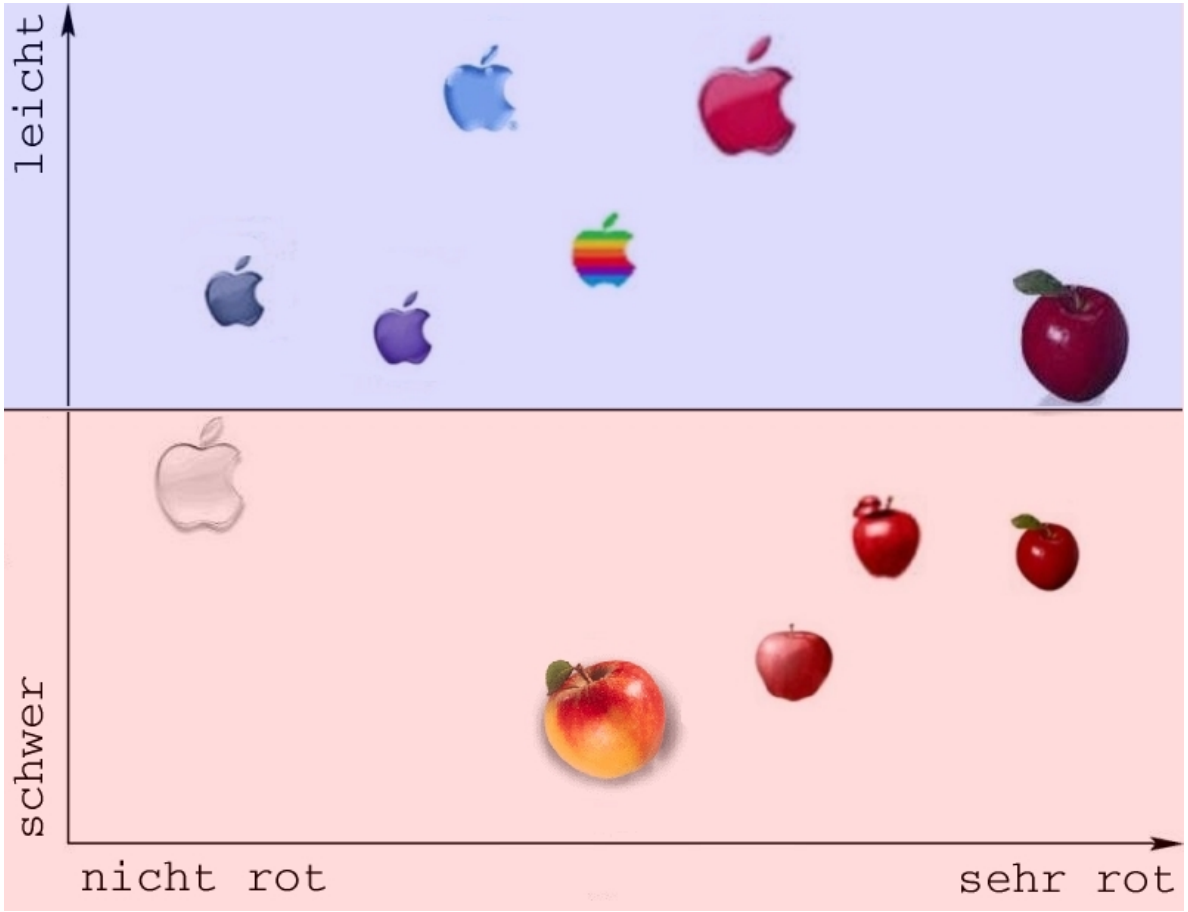
- High weight

Boosting: 2nd Hypothesis



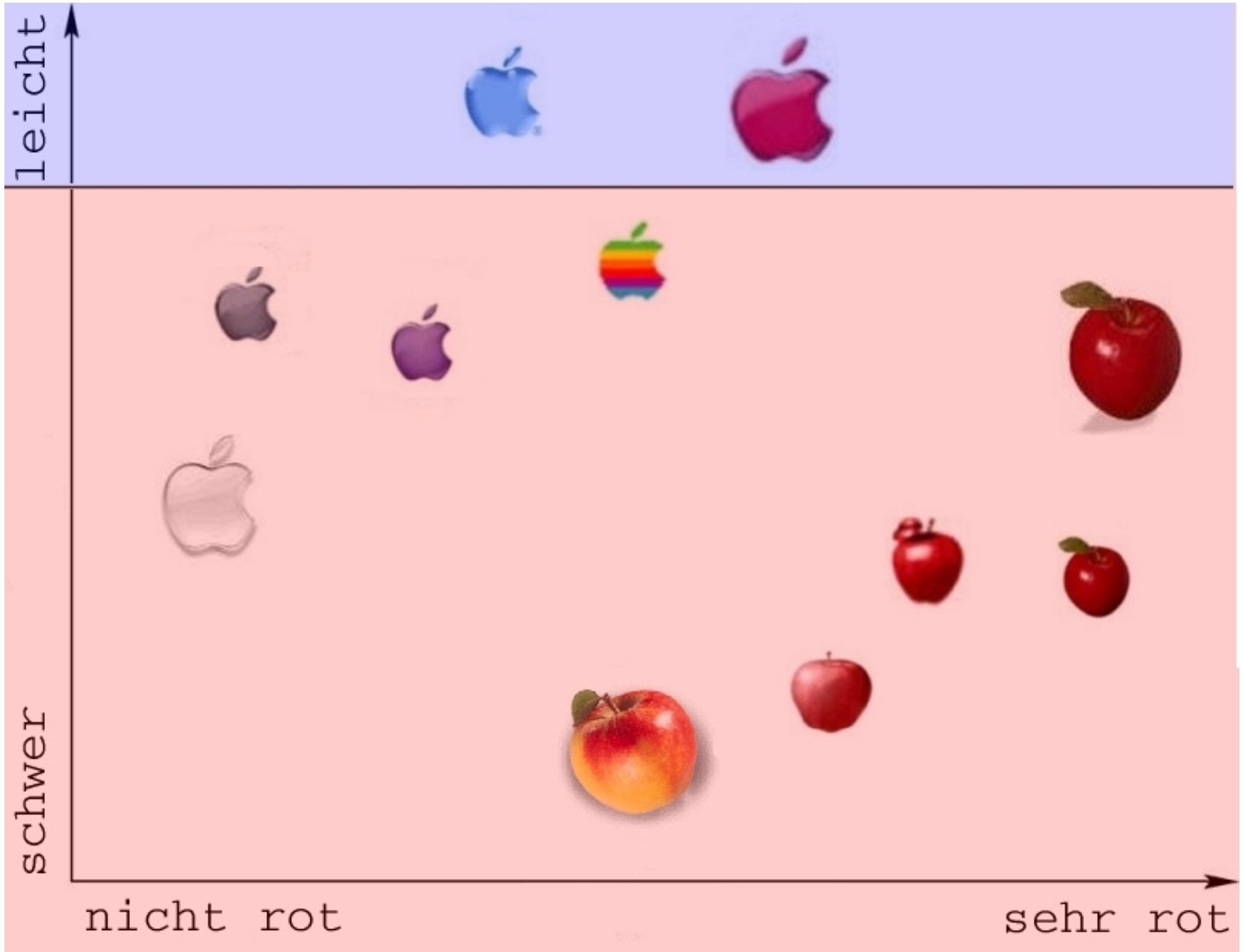
Pick hypotheses with high edge

Update Distribution

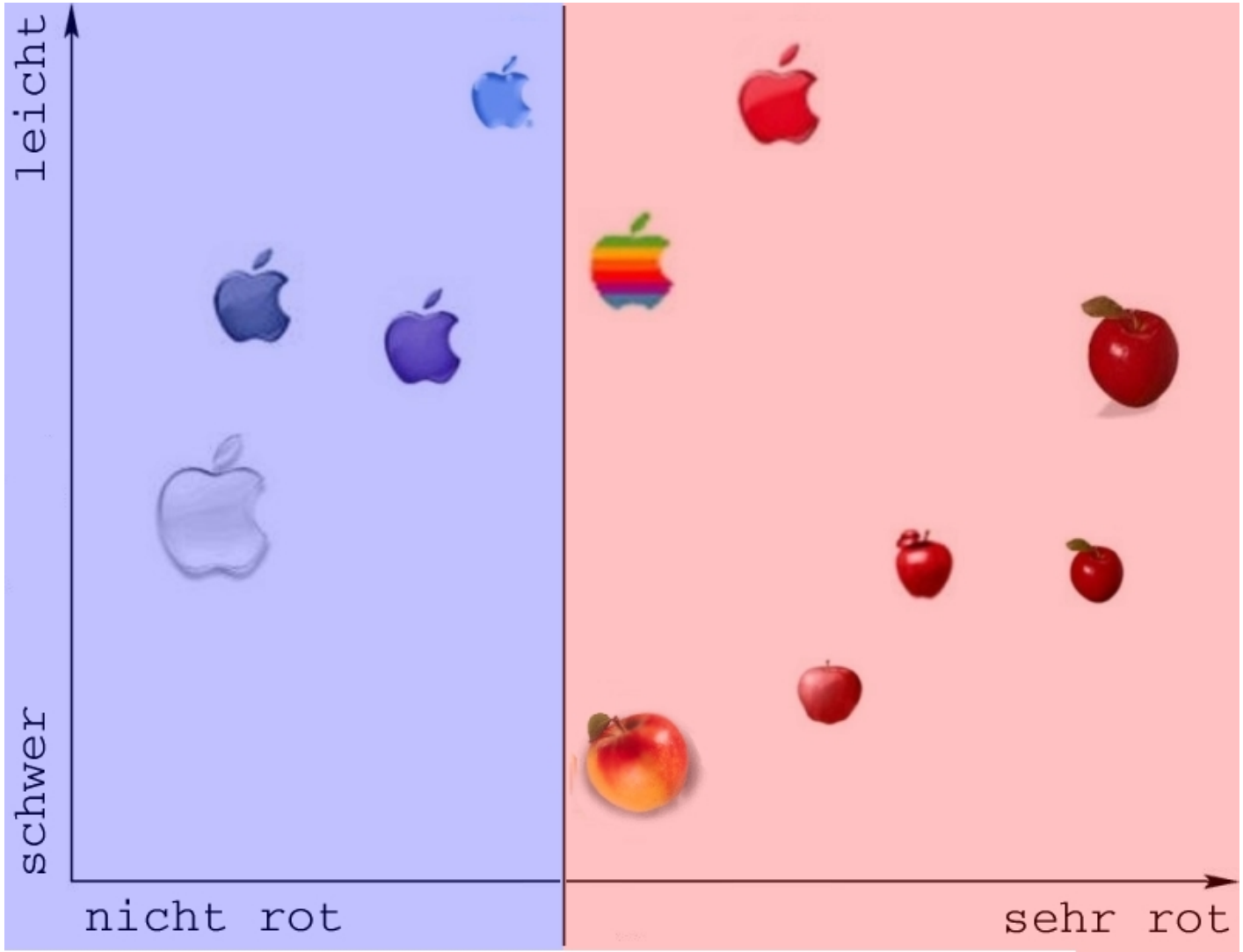


After update:
edge of all
chosen hypotheses
is small

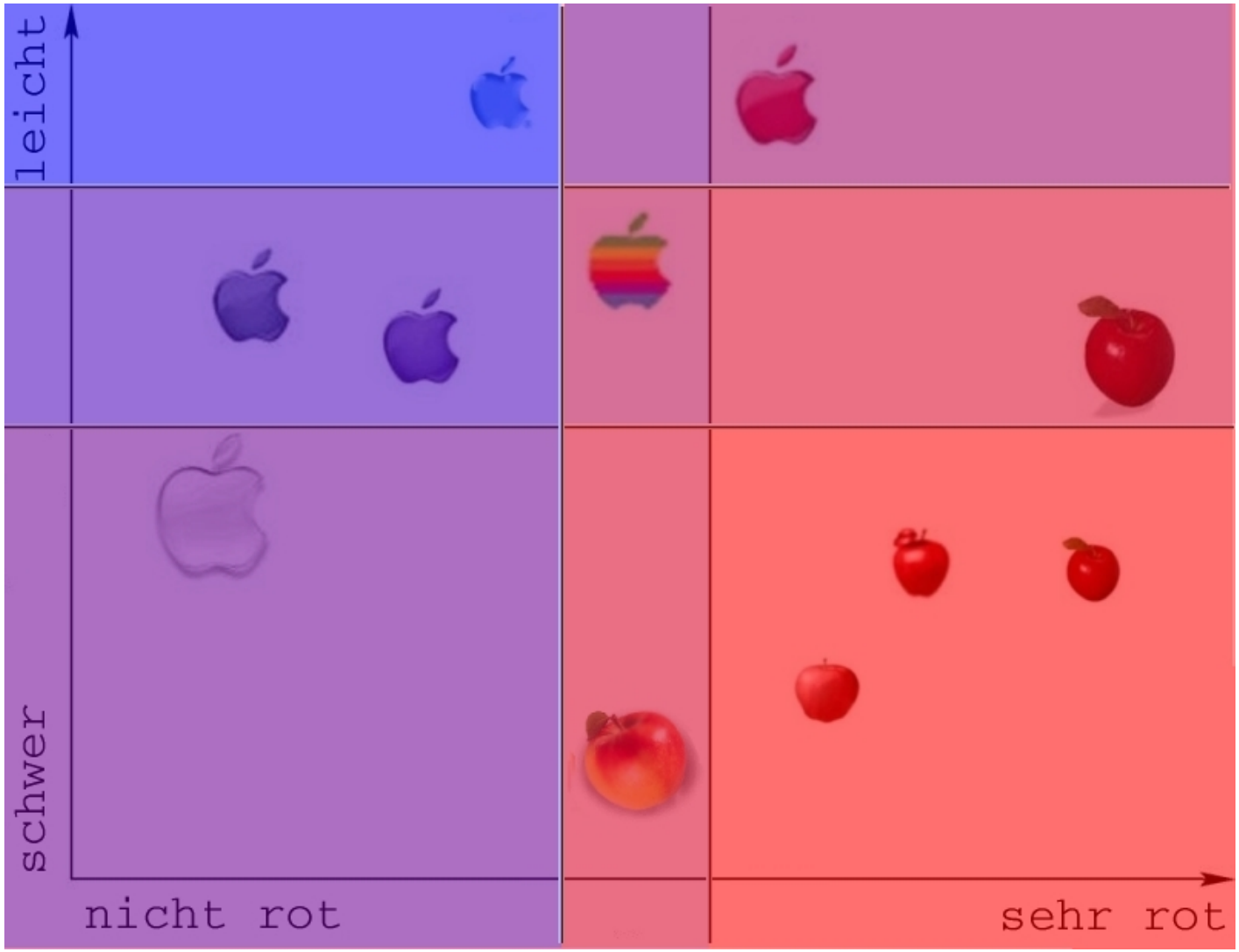
Boosting: 3nd Hypothesis



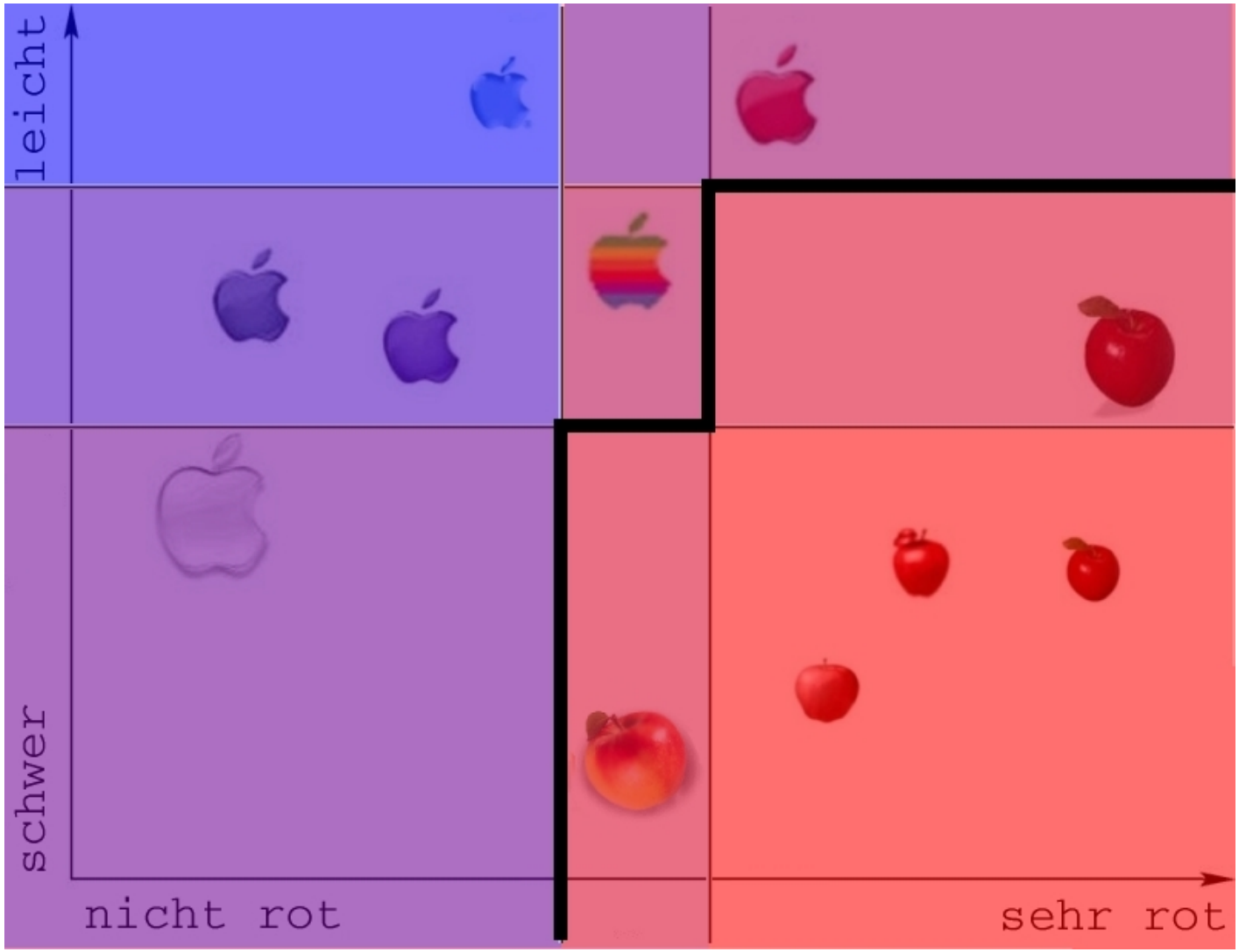
Boosting: 4th Hypothesis



All Hypotheses



Decision: $f_\alpha(\mathbf{x}) = \sum_{t=1}^{T} \alpha_t h^t(\mathbf{x}) > 0$?



Apple Classification Problem in Matrix Form

			α_1	α_2	α_3	α_4	margin
			-	-	-	-	
	d_1	1/11	1	-1	-1	1	-
	d_2	1/11	1	1	-1	1	-
	d_3	1/11	1	1	-1	1	-
	d_4	1/11	1	1	1	-1	-
	d_5	1/11	1	1	-1	1	-
	d_6	1/11	-1	1	1	-1	-
	d_7	1/11	1	-1	1	1	-
	d_8	1/11	1	1	1	1	-
	d_9	1/11	1	1	1	1	-
	d_{10}	1/11	1	1	1	1	-
	d_{11}	1/11	-1	1	1	1	-
	edge		.64	.64	.27	.64	
	value	**-1**					

Apple Classification Problem in Matrix Form

			α_1	α_2	α_3	α_4	margin
			1	0	0	0	
	d_1	0	1	-1	-1	1	1
	d_2	0	1	1	-1	1	1
	d_3	0	1	1	-1	1	1
	d_4	0	1	1	1	-1	1
	d_5	0	1	1	-1	1	1
	d_6	.5	-1	1	1	-1	-1
	d_7	0	1	-1	1	1	1
	d_8	0	1	1	1	1	1
	d_9	0	1	1	1	1	1
	d_{10}	0	1	1	1	1	1
	d_{11}	.5	-1	1	1	1	-1
	edge		-1	1	1	0	
	value	-1	**-1**				

Apple Classification Problem in Matrix Form

			α_1	α_2	α_3	α_4	margin
			.5	.5	0	0	
	d_1	.25	1	-1	-1	1	0
	d_2	0	1	1	-1	1	1
	d_3	0	1	1	-1	1	1
	d_4	0	1	1	1	-1	1
	d_5	0	1	1	-1	1	1
	d_6	.25	-1	1	1	-1	0
	d_7	.25	1	-1	1	1	0
	d_8	0	1	1	1	1	1
	d_9	0	1	1	1	1	1
	d_{10}	0	1	1	1	1	1
	d_{11}	.25	-1	1	1	1	0
	edge		0	0	.5	.5	
	value	-1	-1	**0**			

Apple Classification Problem in Matrix Form

			α_1	α_2	α_3	α_4	margin
			.5	.18	.32	0	
	d_1	.5	1	-1	-1	1	0
	d_2	0	1	1	-1	1	.36
	d_3	0	1	1	-1	1	.36
	d_4	0	1	1	1	-1	1
	d_5	0	1	1	-1	1	.36
	d_6	.25	-1	1	1	-1	0
	d_7	0	1	-1	1	1	.64
	d_8	0	1	1	1	1	1
	d_9	0	1	1	1	1	1
	d_{10}	0	1	1	1	1	1
	d_{11}	.25	-1	1	1	1	0
	edge		0	0	0	.5	
	value	-1	-1	0	**0**		

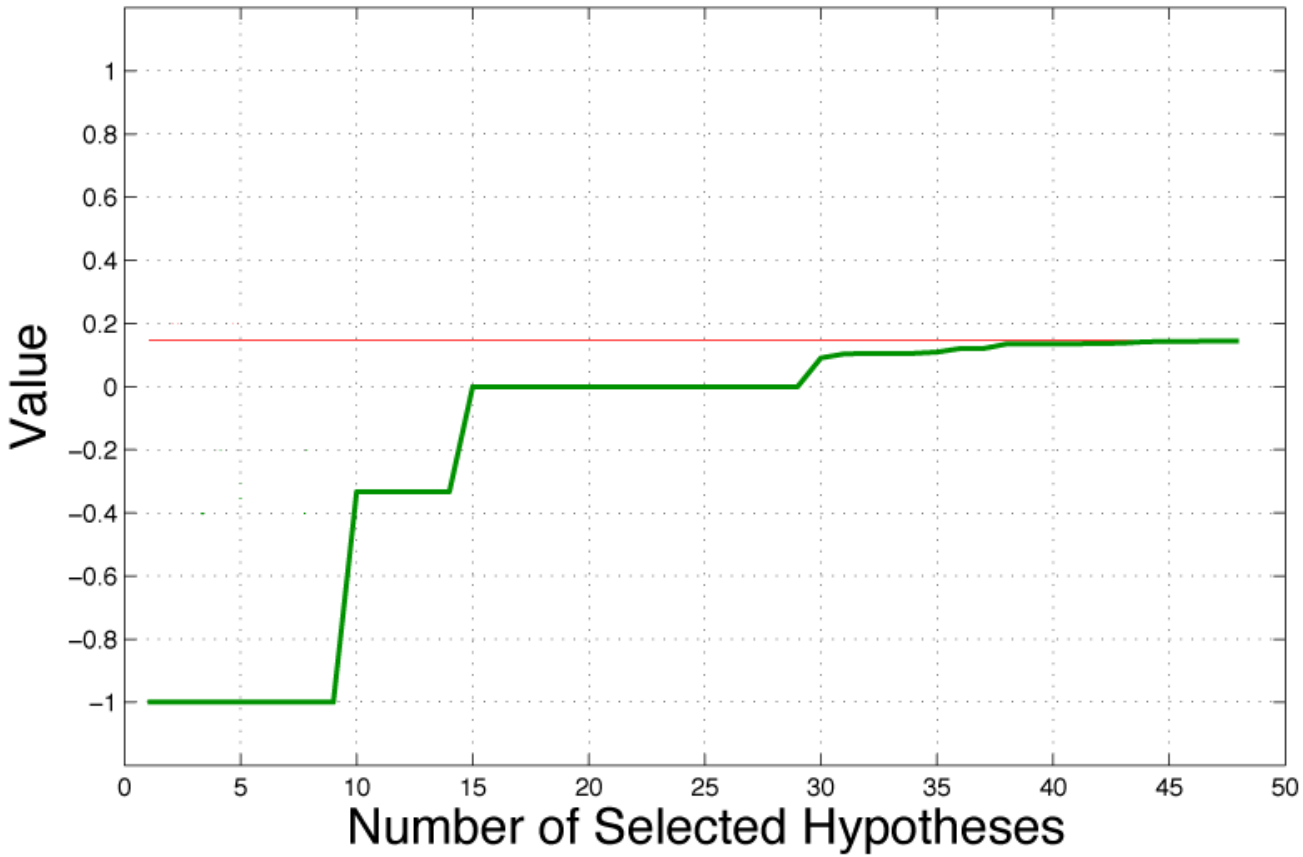
Apple Classification Problem in Matrix Form

			α_1	α_2	α_3	α_4	margin
			.25	.44	.06	.25	
	d_1	.5	1	-1	-1	1	0
	d_2	0	1	1	-1	1	.88
	d_3	0	1	1	-1	1	.88
	d_4	0	1	1	1	-1	.5
	d_5	0	1	1	-1	1	.88
	d_6	.5	-1	1	1	-1	0
	d_7	0	1	-1	1	1	.12
	d_8	0	1	1	1	1	1
	d_9	0	1	1	1	1	1
	d_{10}	0	1	1	1	1	1
	d_{11}	0	-1	1	1	1	.5
	edge		0	0	0	0	
	value	-1	-1	0	0	**0**	

Outline

Boosting = greedy method for increasing margin

Converges to optimum marging 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 margin $\geq g - \epsilon$
- Number of iterations $O(\frac{\ln n}{\epsilon^2})$

Outline

LPBoost [GS98,RSS+00,DBST02]

Choose distribution that minimizes the maximum edge via LP

$$\min_{\sum_n d_n=1,\ d_i\geq 0} \max_{q=1,2,\ldots,t} M_{*,q}\cdot \mathbf{d}$$

$$= \min_{\substack{\sum_n d_n = 1,\ d_i \geq 0 \\ M_{*,q}\cdot \mathbf{d} \leq c}} c$$

- Good practical boosting algorithm
- All weight is put on examples with minimum margin
- Brittle: iteration bound can be linear in number of examples n on malign artificial data sets [WGR07]

LPBoost Counter Example

		α_1	α_2	α_3	α_4	α_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 Counter Example

		α_1	α_2	α_3	α_4	α_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 Counter Example

		α_1	α_2	α_3	α_4	α_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 Counter Example

		α_1	α_2	α_3	α_4	α_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 Counter Example

		α_1	α_2	α_3	α_4	α_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 Counter Example

		α_1	α_2	α_3	α_4	α_5	margin
		.5	.0026	0	0	.4975	
d_1	0.4974	+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	.4898	-.97	-.96	+1	-.91	+.99	.0051
d_7	0	-.97	-.95	-.94	+1	+.99	.0051
d_8	.0127	-.97	-.95	-.93	-.92	+.99	.0051
edge		.0051	.0051	.9055	.9100	.0051	
value	-1	-.98	-.96	-.94	-.92	.0051	

Entropy Regularized LPBoost

$$\min_{\sum_n d_n=1} \max_{q=1,2,\ldots,t} M_{*,q} \cdot \mathbf{d} + \frac{1}{\eta}\Delta(\mathbf{d}, \mathbf{d}^0),$$

where regularizer $\Delta(\mathbf{d}, \mathbf{d}^0)$ is relative entropy $\sum_i d_i \ln \frac{d_i}{d_i^0}$

See visualization in part 2 of Lecture 3:

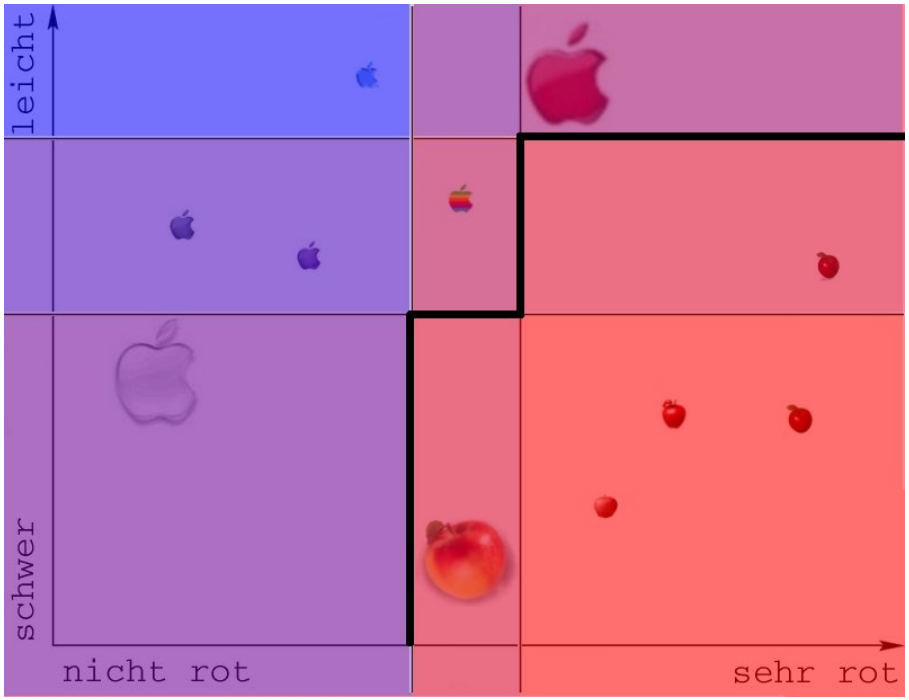
`http://www.soe.ucsc.edu/classes/cmps290c/Spring07/`

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

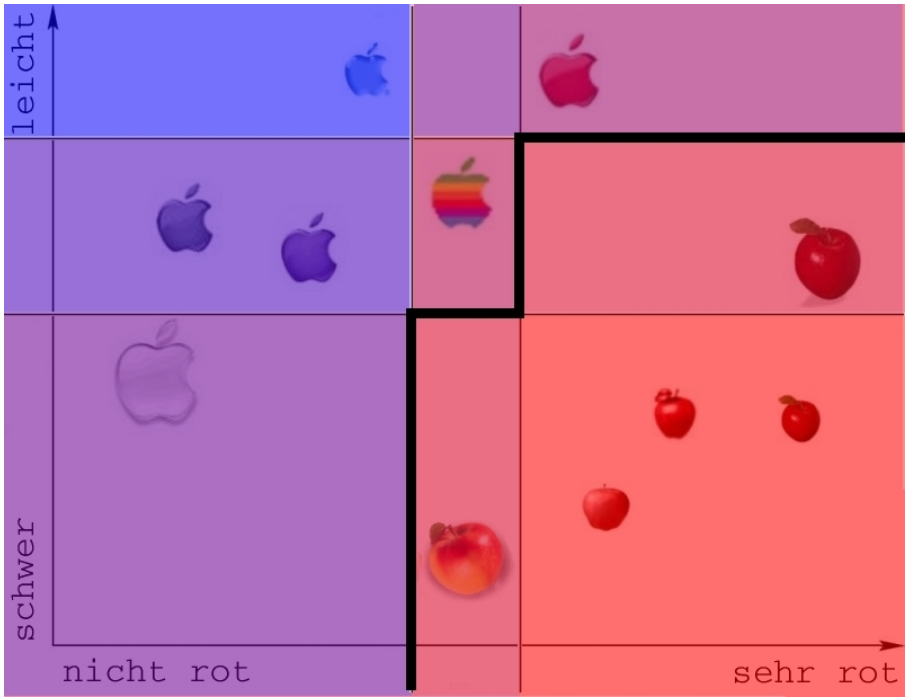
- Within ϵ of maximum margin in $O(\frac{\log n}{\epsilon^2})$ iterations
- Above form of weights first appeared in ν-Arc algorithm [RSS+00]

The effect of entropy regularization

Different distribution on the examples



LPBoost: lots of zeros
sometimes $\Omega(n)$ iterations



ERLPBoost: smoother distribution
always $O(\frac{\log n}{\epsilon^2})$ iterations

Conclusion

- Machine learning problems often modeled as games against nature or adversary
- Often end up with zero-sum games
- Its all about efficiency:
  Even against worst adversary we only need so much resources