Boosting, Margins and Beyond

Gunnar Rätsch

Friedrich Miescher Laboratory, Max Planck Society,
Tübingen, Germany

September 12, 2007

29^{th} Annual Symposium of the
German Association for Pattern Recognition

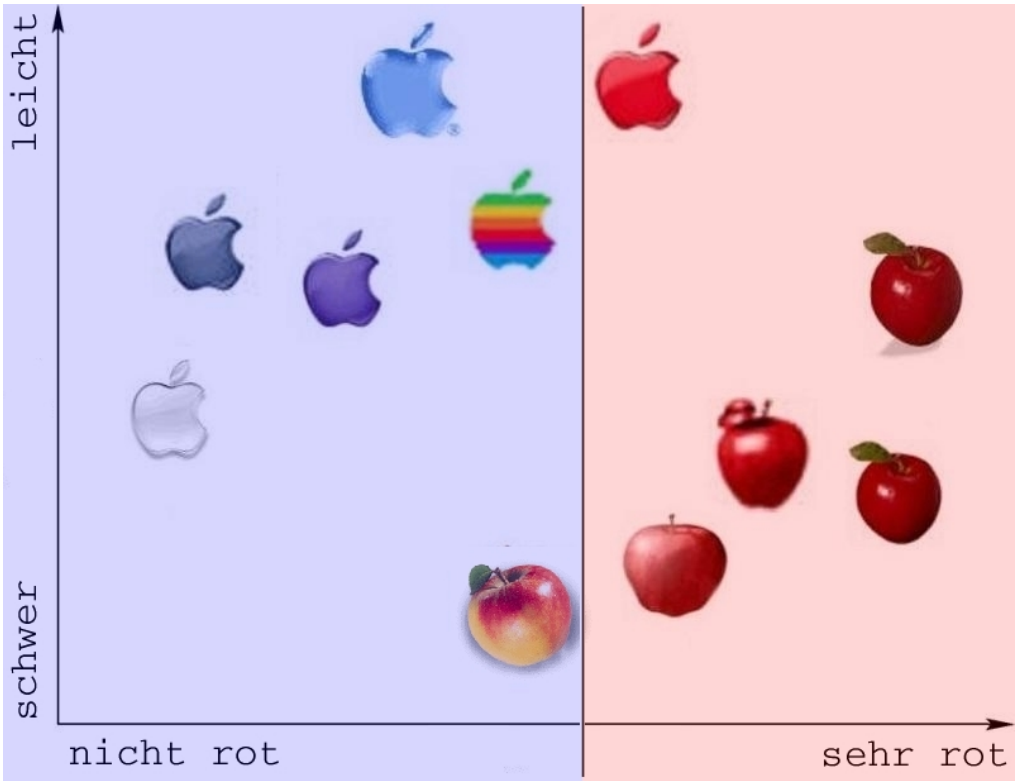
The AdaBoost Algorithm

- Given N examples $(\mathbf{x}_n, y_n)$, $n = 1, \ldots, N$
- Maintain a distribution $\mathbf{d}^t = [d_1^t, \ldots, d_N^t]$ on the examples
- At iteration $t = 1, \ldots, T$:
  1. Receive a "weak" hypothesis $h_t : \mathcal{X} \to [-1, 1]$
  2. Update $\mathbf{d}^t$ to $\mathbf{d}^{t+1}$, put more weights on "hard" examples
- Output a convex combination of the weak hypotheses

$$f(\mathbf{x}) = \sum_{t=1}^{T} \alpha_t h_t(\mathbf{x})$$

[Freund & Schapire, 1995]

Boosting: 1st Iteration

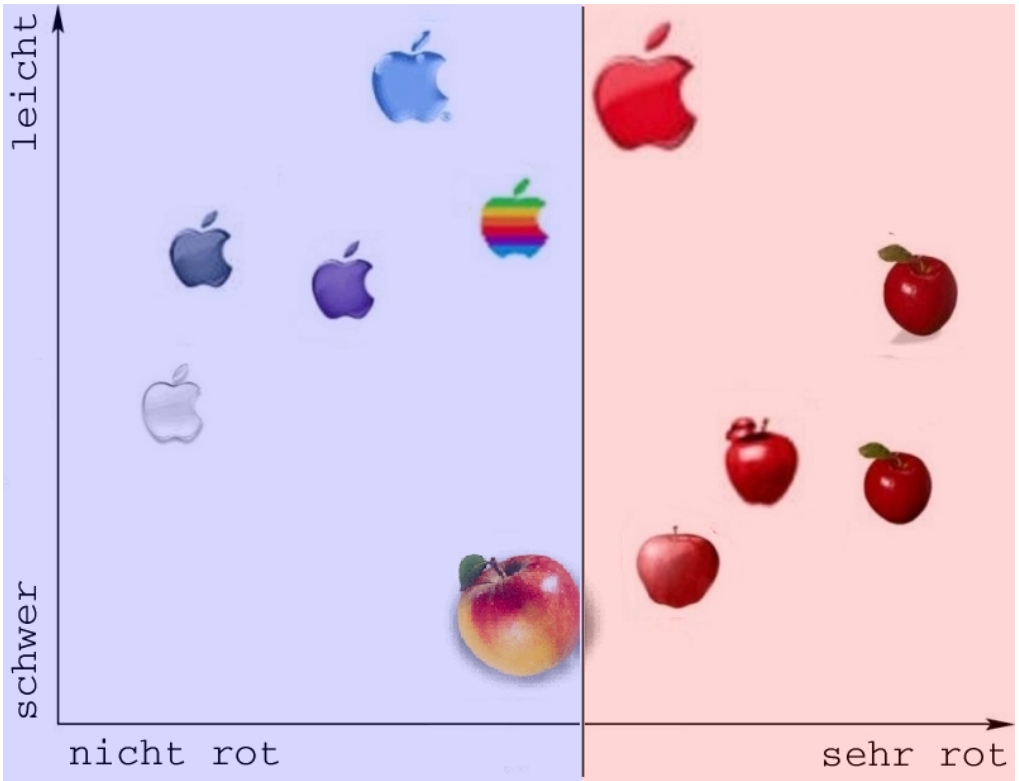


Error rate of 1^{st} hypothesis:

$$\epsilon = \sum_{n=1}^{N} d_n^t \, \mathbf{I}(h_t(\mathbf{x}_n) \neq y_n)$$

$$= \frac{2}{11}$$

Update Distribution



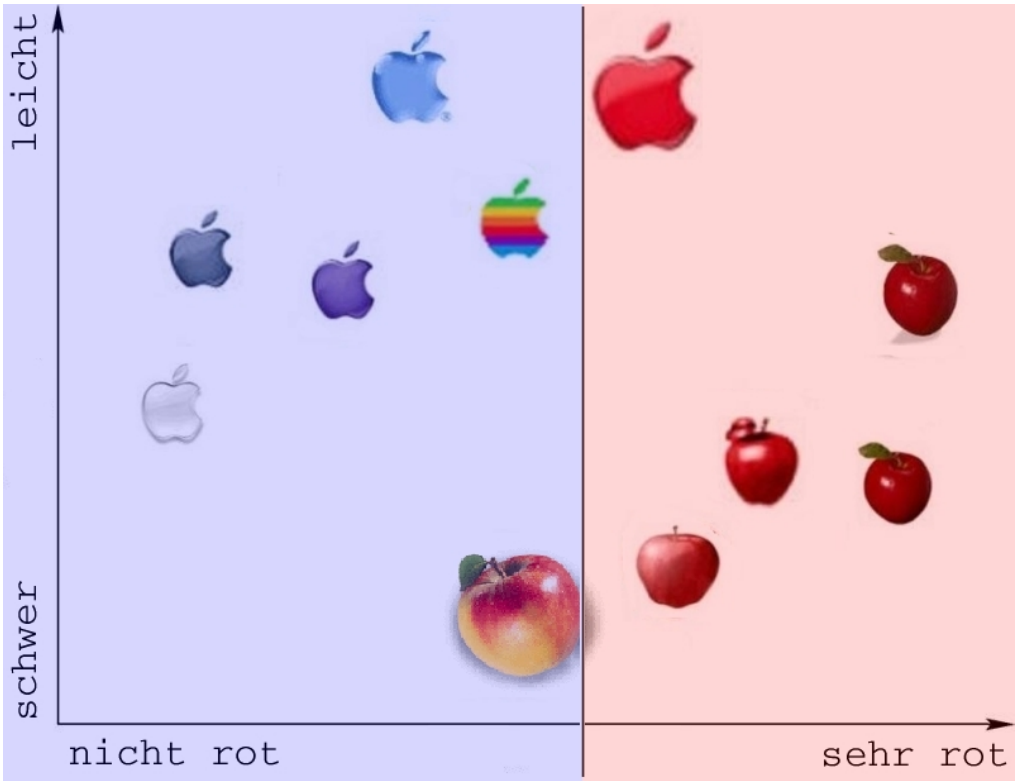
Misclassified examples
$\Rightarrow$ Increase weights

Error after update:

$$\epsilon' = \sum_{n=1}^{N} d_n^{t+1} \, \mathbf{I}(h_t(\mathbf{x}_n) \neq y_n)$$

$$= \frac{1}{2}$$

Update Distribution

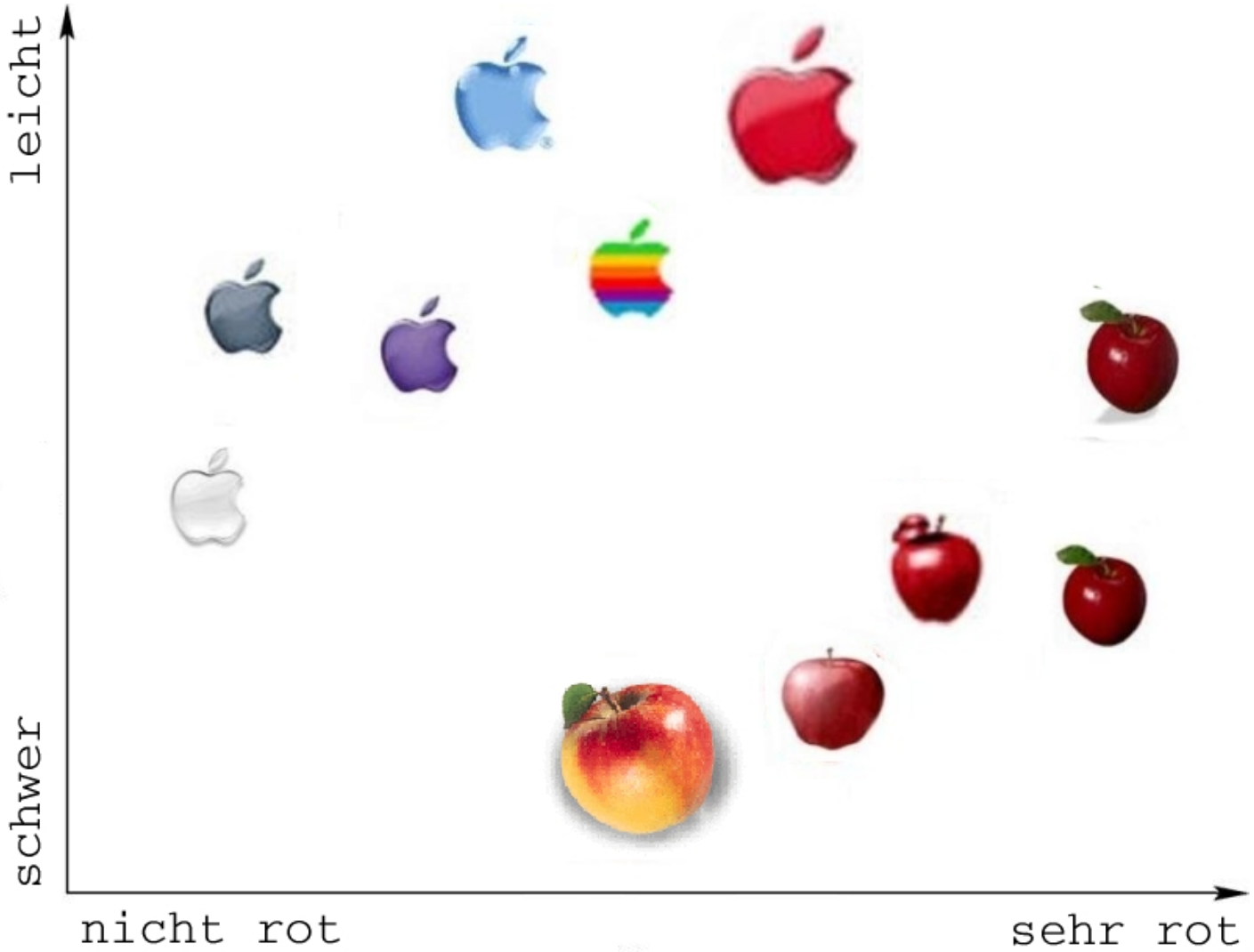


Misclassified examples
$\Rightarrow$ Increase weights

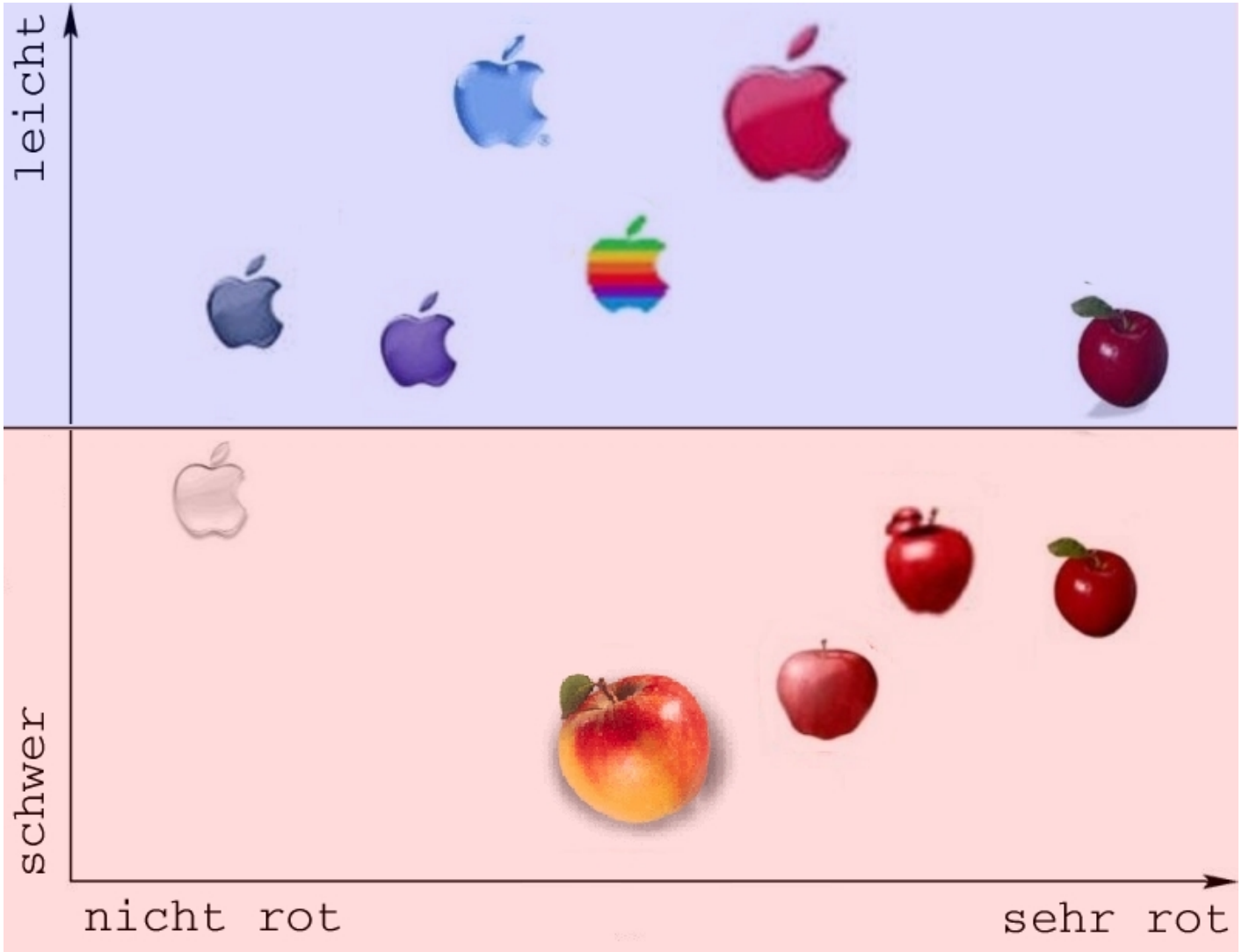
Error after update:

$$\epsilon' = \sum_{n=1}^{N} d_n^{t+1} \, \mathbf{I}(h_t(\mathbf{x}_n) \neq y_n)$$
$$= \frac{1}{2}$$

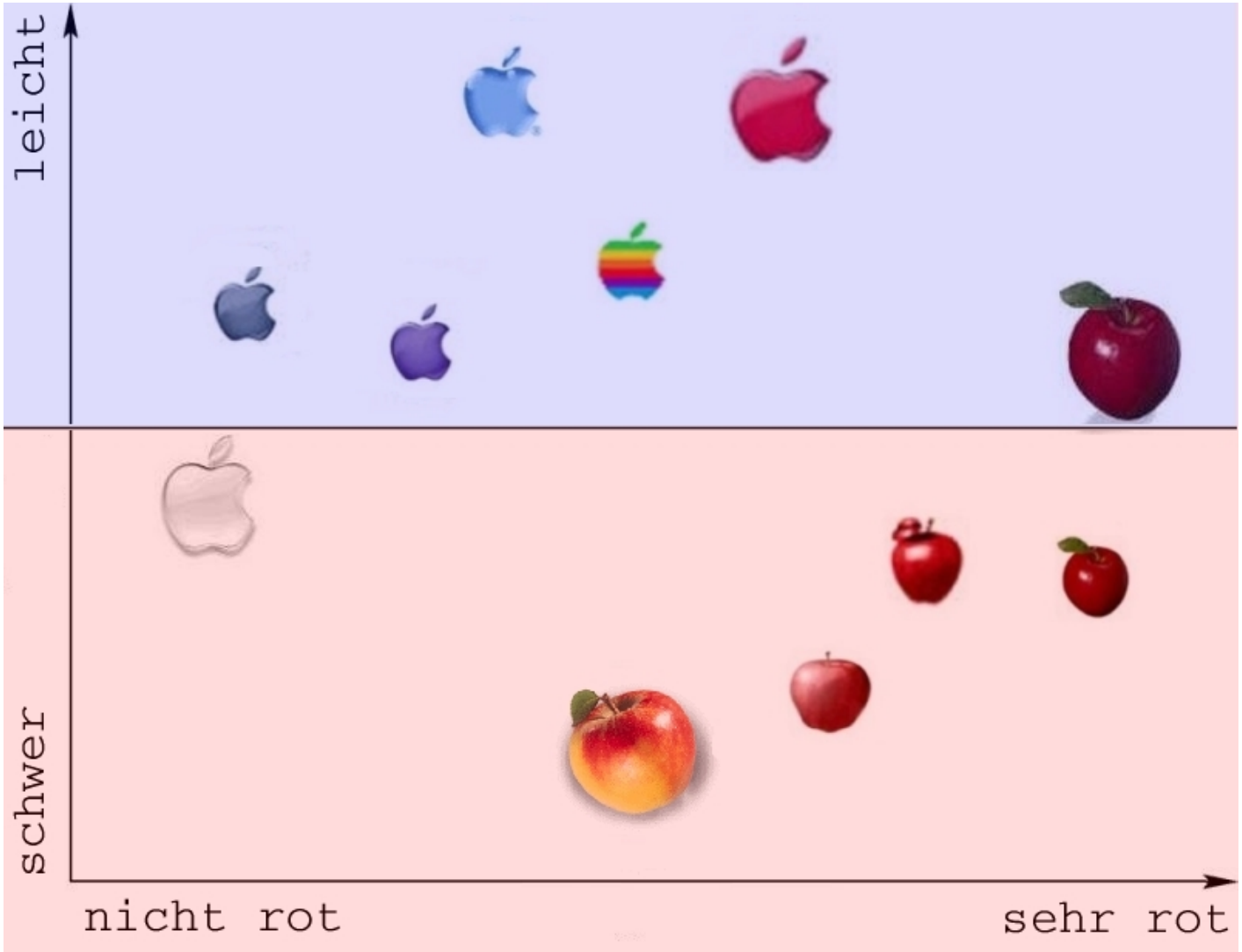
Before 2nd Iteration



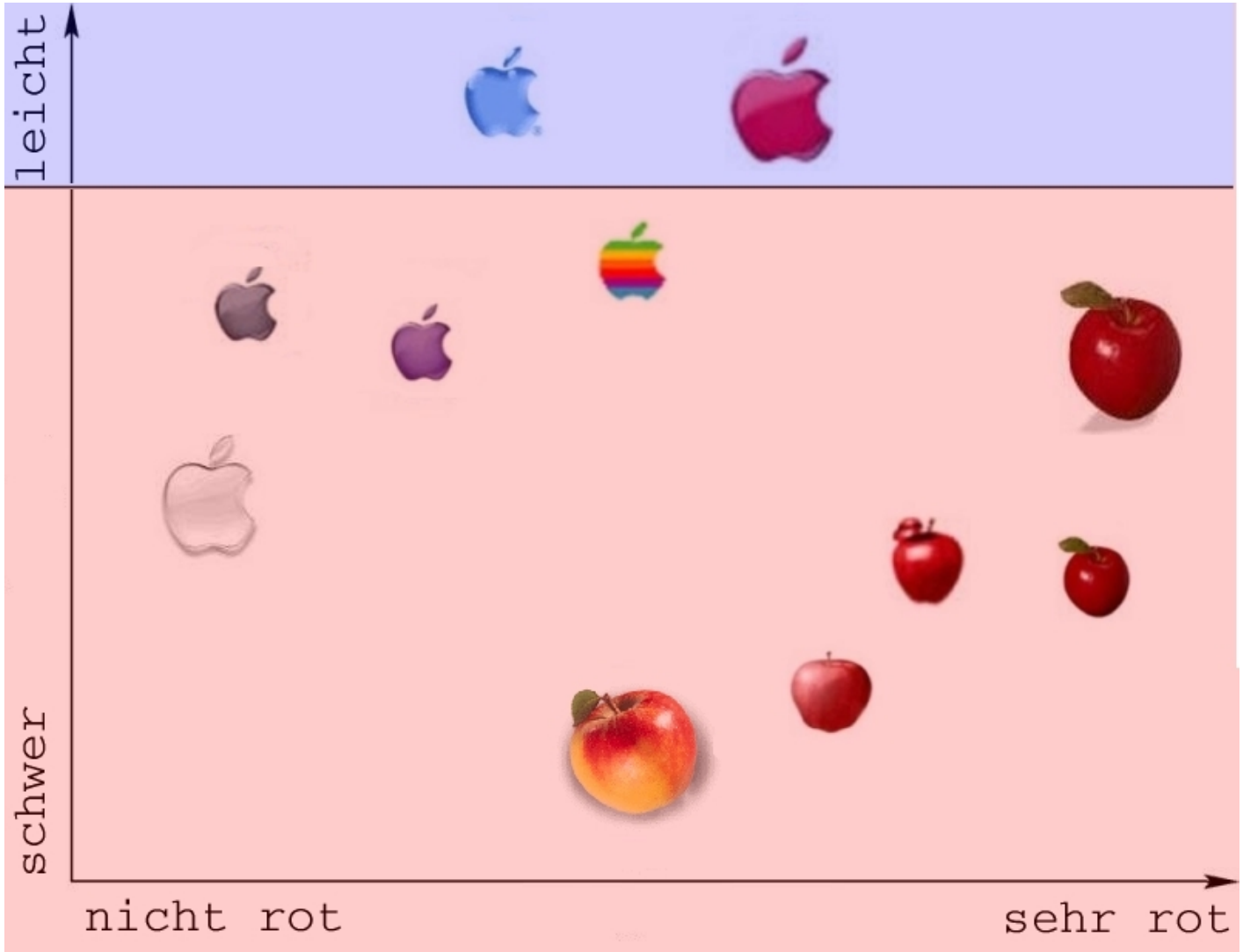
Boosting: 2nd Hypothesis ($\epsilon < \frac{1}{2}$)



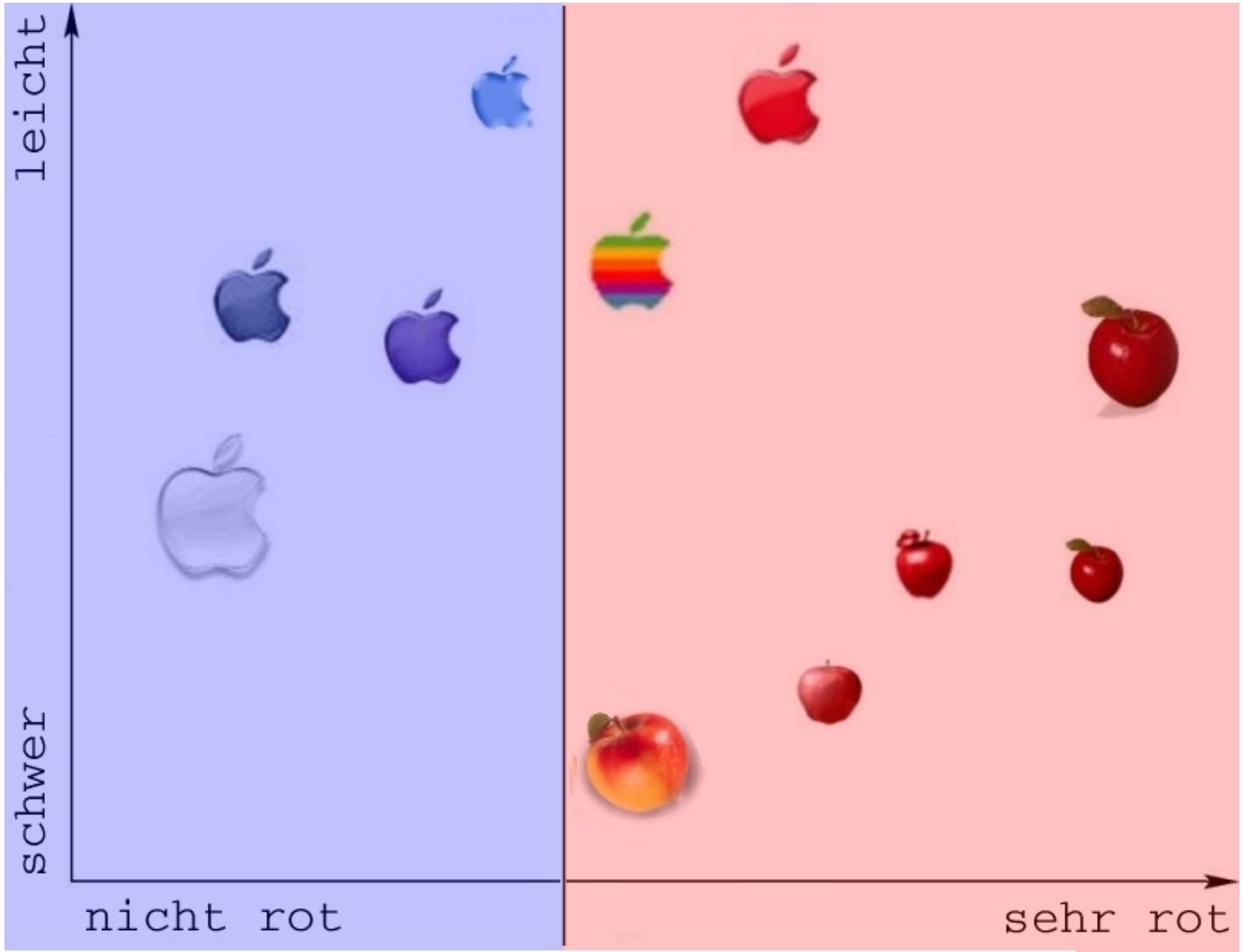
Update Distribution ($\epsilon' = \frac{1}{2}$)



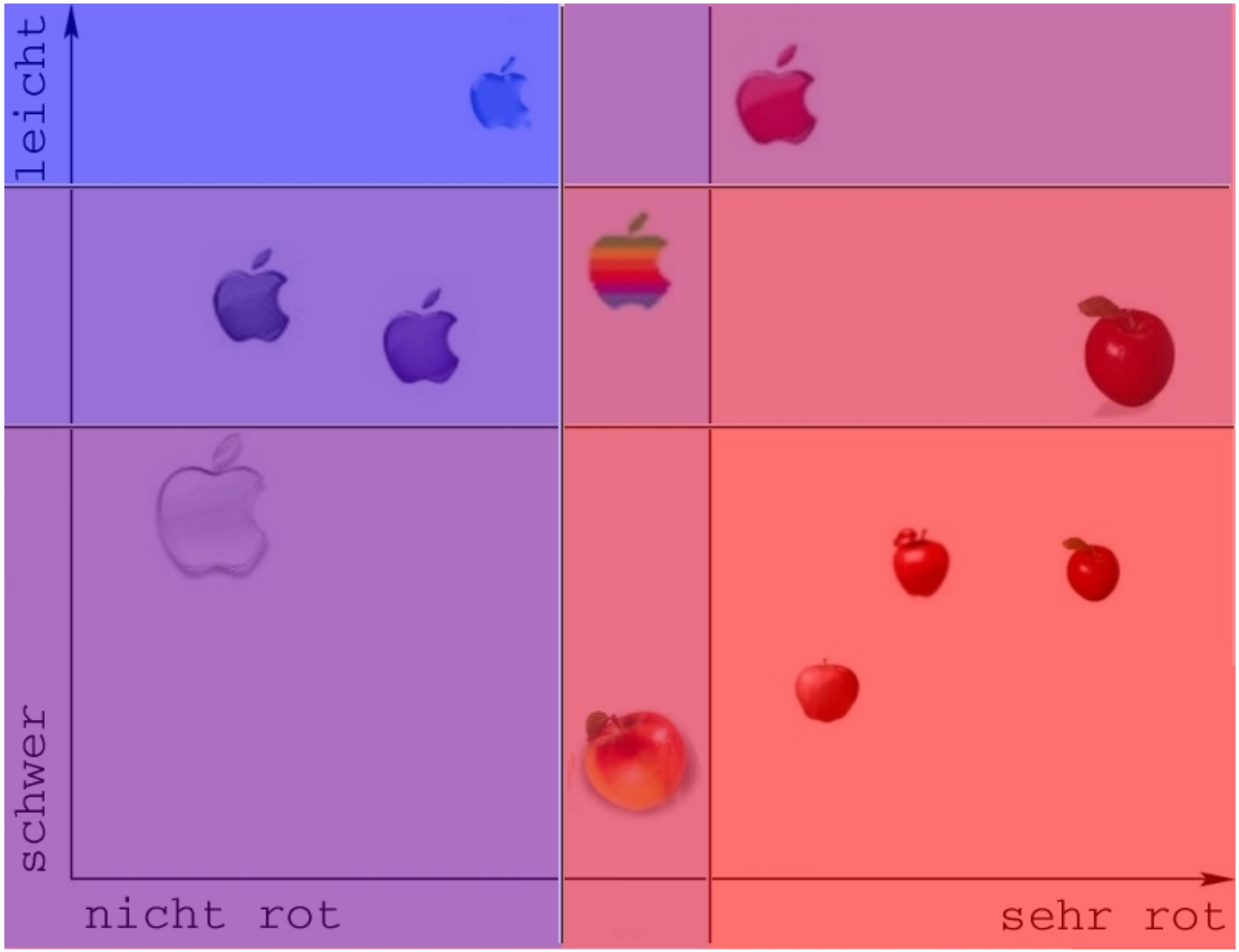
Boosting: 3rd Hypothesis



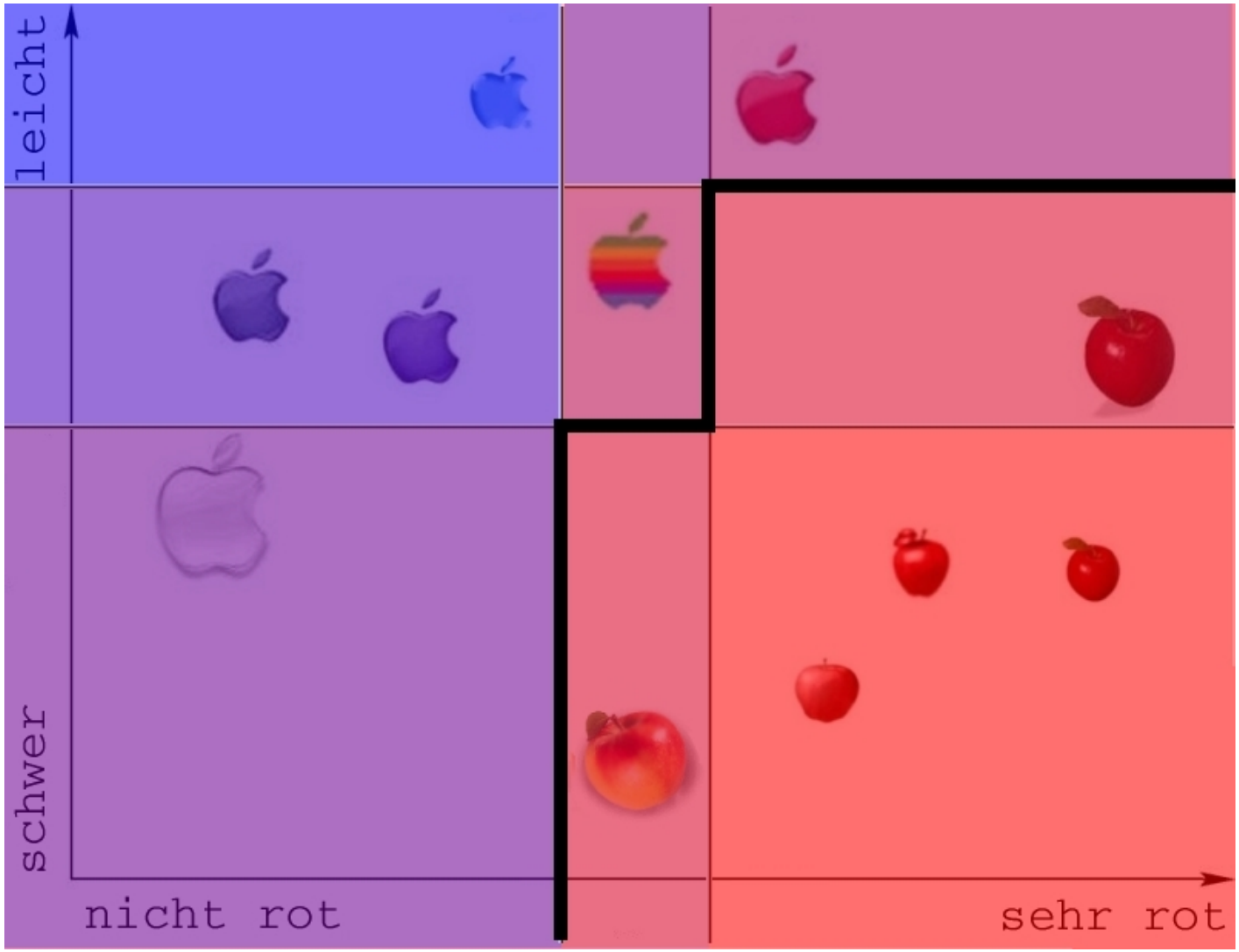
Boosting: 4th Hypothesis



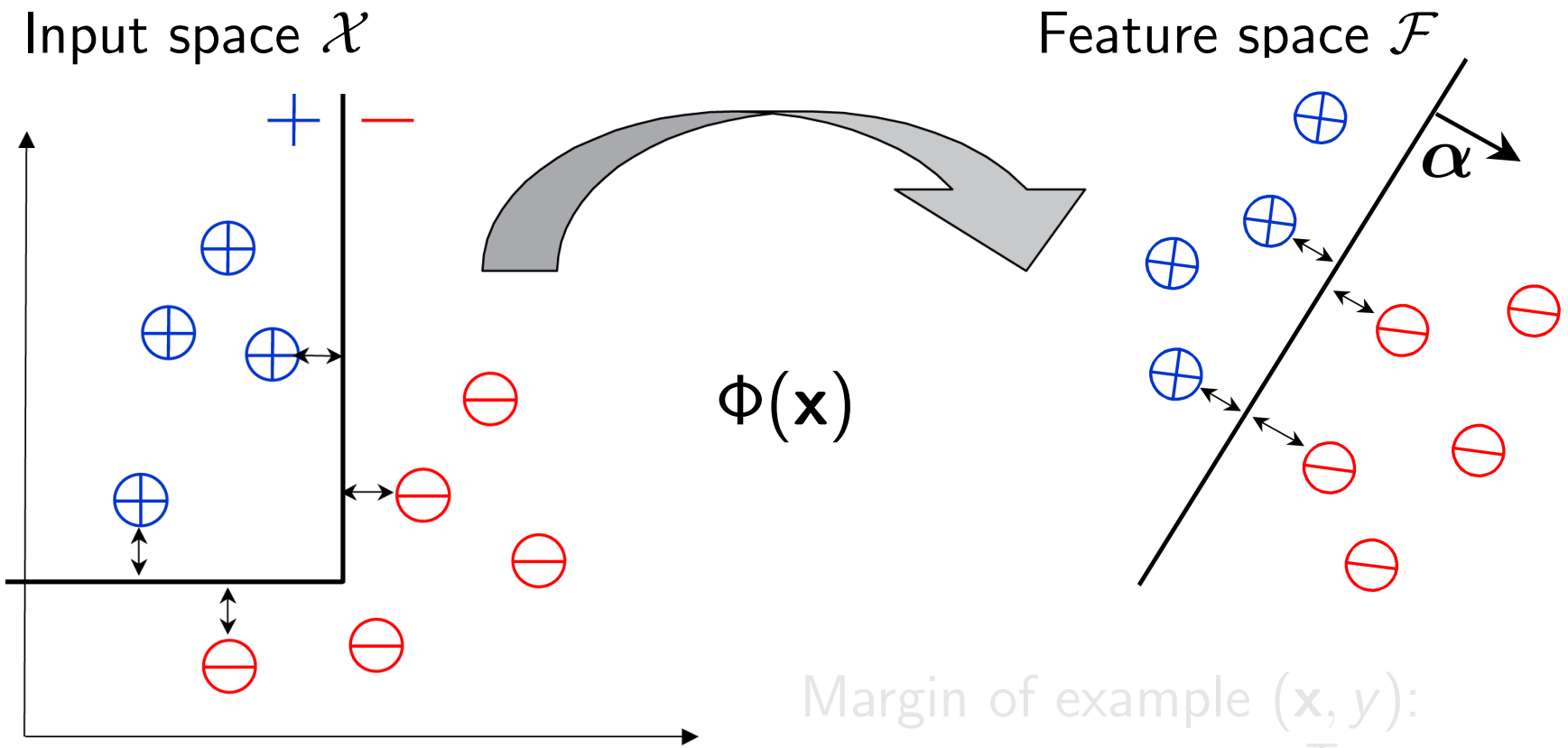
All Hypotheses



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



Linear Separation and Large Margins



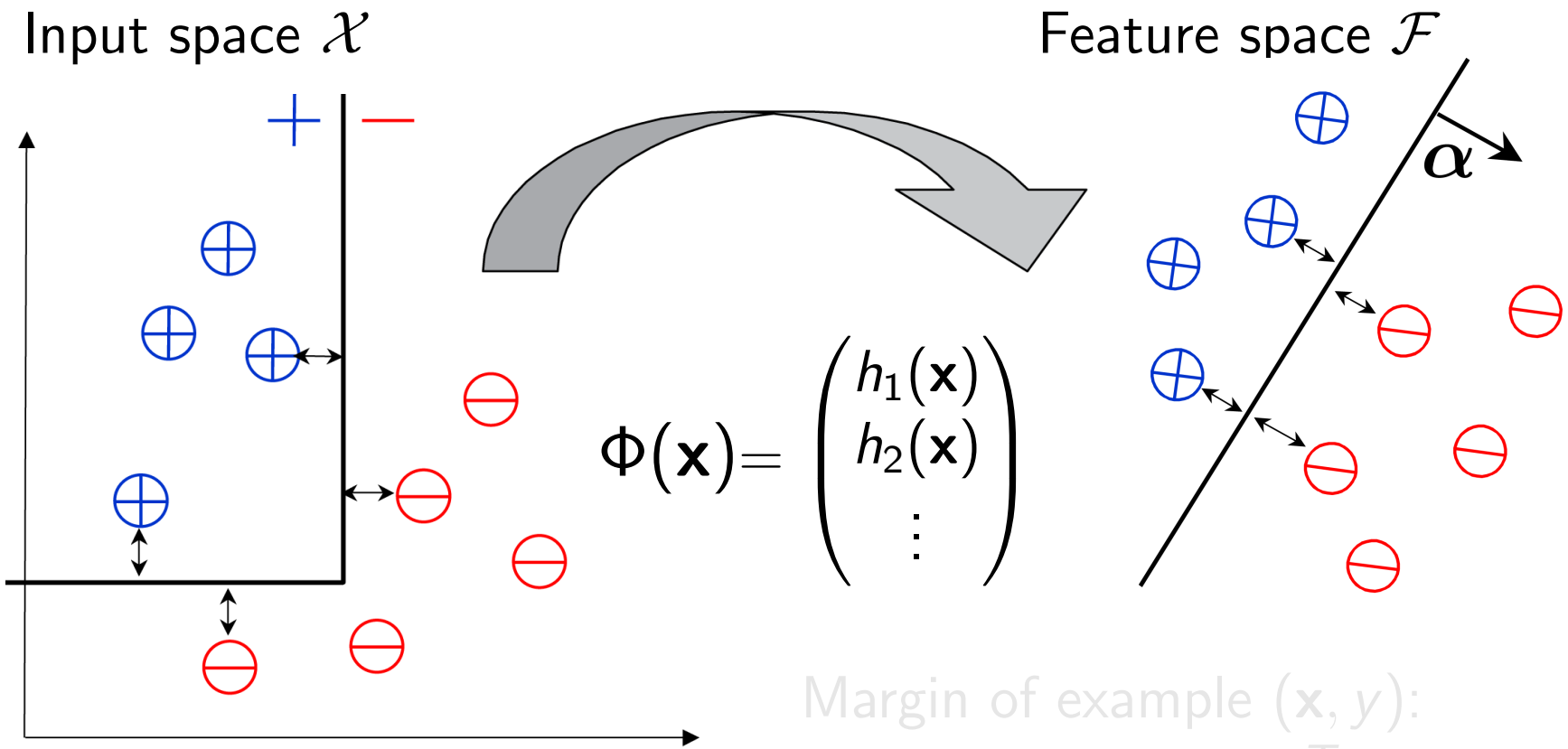
Linear separation in $\mathcal{F}$ is
nonlinear separation in $\mathcal{X}$

Margin of example $(\mathbf{x}, y)$:

$$y\langle \Phi(\mathbf{x}), \alpha \rangle = y \sum_{t=1}^{T} \alpha_t h_t(\mathbf{x})$$

[Mangasarian, 1999; G.R., Mika, Schölkopf & Müller, 2002]

Linear Separation and Large Margins



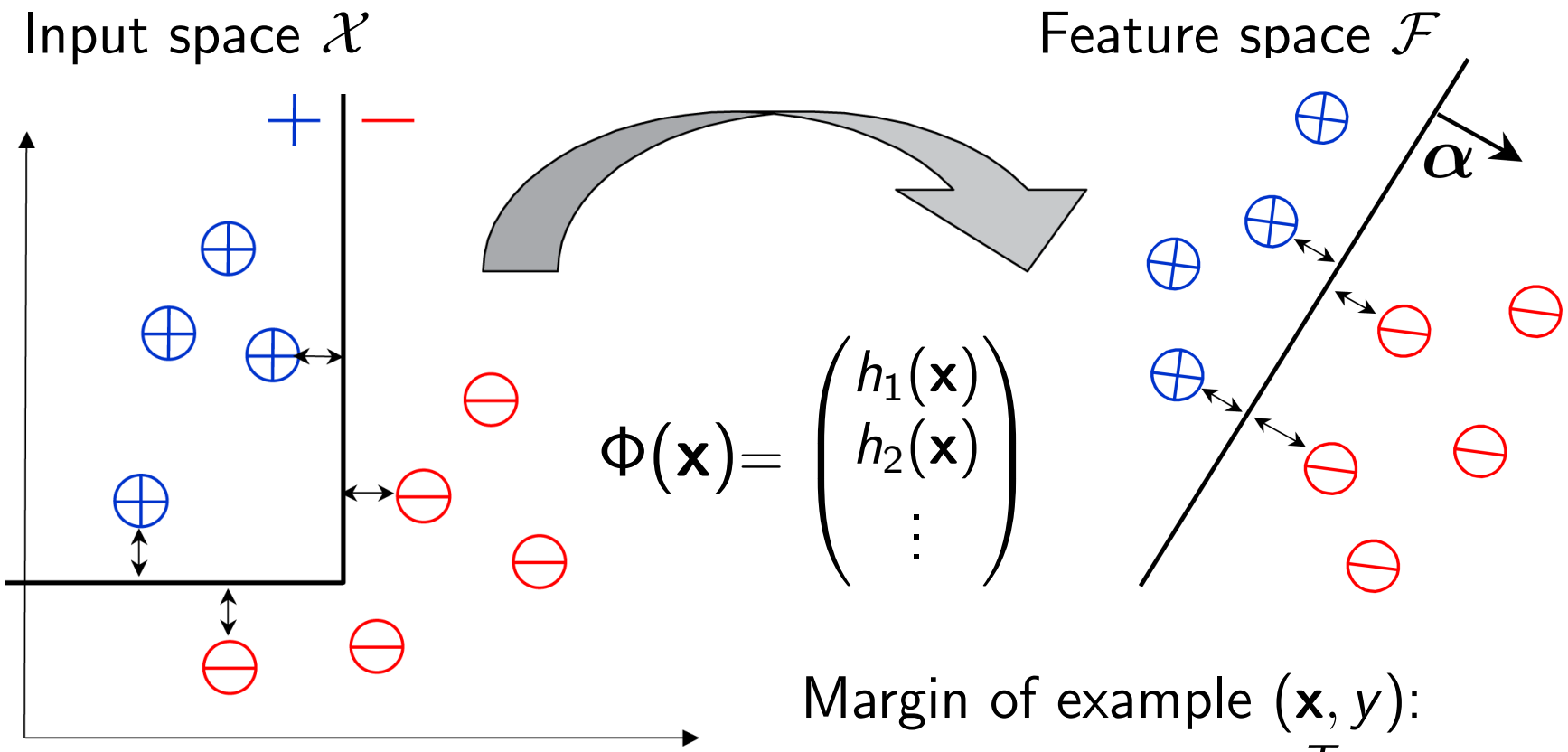
Linear separation in $\mathcal{F}$ is
nonlinear separation in $\mathcal{X}$

Margin of example $(\mathbf{x}, y)$:

$$y\langle \Phi(\mathbf{x}), \alpha \rangle = y \sum_{t=1}^{T} \alpha_t h_t(\mathbf{x})$$

[Mangasarian, 1999; G.R., Mika, Schölkopf & Müller, 2002]

Linear Separation and Large Margins



Linear separation in $\mathcal{F}$ is nonlinear separation in $\mathcal{X}$

Margin of example $(\mathbf{x}, y)$:

$$y\langle \Phi(\mathbf{x}), \boldsymbol{\alpha} \rangle = y \sum_{t=1}^{T} \alpha_t h_t(\mathbf{x})$$

[Mangasarian, 1999; G.R., Mika, Schölkopf & Müller, 2002]

Margin vs. Edge

Margin

- Measure for "confidence" in prediction of example n: $y_n f_\alpha(\mathbf{x}_n)$
- Depends on hypothesis weighting α

Edge

- "Goodness" of hypothesis returned by learner: $\gamma = 1 - 2\epsilon$
- Depends on distribution $\mathbf{d}$ on examples

What is the connection?

von Neumann's Minimax-Theorem

$$\textbf{minimum edge} \quad = \quad \textbf{maximum margin}$$

[von Neumann, 1928, Breiman, 1999]

Margin vs. Edge

Margin

- Measure for "confidence" in prediction of example n: $y_n f_{\boldsymbol{\alpha}}(\mathbf{x}_n)$
- Depends on hypothesis weighting $\boldsymbol{\alpha}$

Edge

- "Goodness" of hypothesis returned by learner: $\gamma = 1 - 2\epsilon$
- Depends on distribution $\mathbf{d}$ on examples

What is the connection?

von Neumann's Minimax-Theorem

$$\text{minimum edge} \quad = \quad \text{maximum margin}$$

[von Neumann, 1928, Breiman, 1999]

Margin vs. Edge

Margin

- Measure for "confidence" in prediction of example n: $y_n f_{\boldsymbol{\alpha}}(\mathbf{x}_n)$
- Depends on hypothesis weighting $\boldsymbol{\alpha}$

Edge

- "Goodness" of hypothesis returned by learner: $\gamma = 1 - 2\epsilon$
- Depends on distribution $\mathbf{d}$ on examples

What is the connection?

von Neumann's Minimax-Theorem

$$\text{minimum edge} \quad = \quad \text{maximum margin}$$

[von Neumann, 1928, Breiman, 1999]

Margin vs. Edge

Margin

- Measure for "confidence" in prediction of example n: $y_n f_{\boldsymbol{\alpha}}(\mathbf{x}_n)$
- Depends on hypothesis weighting $\boldsymbol{\alpha}$

Edge

- "Goodness" of hypothesis returned by learner: $\gamma = 1 - 2\epsilon$
- Depends on distribution $\mathbf{d}$ on examples

What is the connection?

von Neumann's Minimax-Theorem

$$\textbf{minimum edge} \quad = \quad \textbf{maximum margin}$$

[von Neumann, 1928, Breiman, 1999]

How to Maximize the Margin?

- AdaBoost does not generally maximize the margin
  - Slow and may cycle
- Arc-GV asymptotically maximizes the margin
  - Relatively slow, asymptotic convergence
- LPBoost uses a linear programming solver
  - Often very fast in practice, no converge rates (new: $\Omega(N)$)
- AdaBoost* requires $\frac{2\log(N)}{\delta^2}$ iterations to get $\rho^t \in [\rho^* - \delta, \rho^*]$
  - Slow in practice, i.e. not faster than theory predicts
- TotalBoost requires $\frac{2\log(N)}{\delta^2}$ iterations to get $\rho^t \in [\rho^* - \delta, \rho^*]$
  - Needs convex optimizer, fast in practice
  - Combination of benefits

[Breiman, 1999; Bennett et al.; G.R. et al., 2001, 2004, 2006; Rudin et al., 2004]

How to Maximize the Margin?

- AdaBoost does not generally maximize the margin
  - Slow and may cycle
- Arc-GV asymptotically maximizes the margin
  - Relatively slow, asymptotic convergence
- LPBoost uses a linear programming solver
  - Often very fast in practice, no converge rates (new: $\Omega(N)$)
- AdaBoost* requires $\frac{2\log(N)}{\delta^2}$ iterations to get $\rho^t \in [\rho^* - \delta, \rho^*]$
  - Slow in practice, i.e. not faster than theory predicts
- TotalBoost requires $\frac{2\log(N)}{\delta^2}$ iterations to get $\rho^t \in [\rho^* - \delta, \rho^*]$
  - Needs convex optimizer, fast in practice
  - Combination of benefits

[Breiman, 1999; Bennett et al.; G.R. et al., 2001, 2004, 2006; Rudin et al., 2004]

How to Maximize the Margin?

- AdaBoost does not generally maximize the margin
  - Slow and may cycle
- Arc-GV asymptotically maximizes the margin
  - Relatively slow, asymptotic convergence
- LPBoost uses a linear programming solver
  - Often very fast in practice, no converge rates (new: $\Omega(N)$)
- AdaBoost* requires $\frac{2\log(N)}{\delta^2}$ iterations to get $\rho^t \in [\rho^* - \delta, \rho^*]$
  - Slow in practice, i.e. not faster than theory predicts
- TotalBoost requires $\frac{2\log(N)}{\delta^2}$ iterations to get $\rho^t \in [\rho^* - \delta, \rho^*]$
  - Needs convex optimizer, fast in practice
  - Combination of benefits

[Breiman, 1999; Bennett et al.; G.R. et al., 2001, 2004, 2006; Rudin et al., 2004]

How to Maximize the Margin?

- AdaBoost does not generally maximize the margin
  - Slow and may cycle
- Arc-GV asymptotically maximizes the margin
  - Relatively slow, asymptotic convergence
- LPBoost uses a linear programming solver
  - Often very fast in practice, no converge rates (new: $\Omega(N)$)
- AdaBoost* requires $\frac{2\log(N)}{\delta^2}$ iterations to get $\rho^t \in [\rho^* - \delta, \rho^*]$
  - Slow in practice, i.e. not faster than theory predicts
- TotalBoost requires $\frac{2\log(N)}{\delta^2}$ iterations to get $\rho^t \in [\rho^* - \delta, \rho^*]$
  - Needs convex optimizer, fast in practice
  - Combination of benefits

[Breiman, 1999; Bennett et al.; G.R. et al., 2001, 2004, 2006; Rudin et al., 2004]

How to Maximize the Margin?

- AdaBoost does not generally maximize the margin
  - Slow and may cycle
- Arc-GV asymptotically maximizes the margin
  - Relatively slow, asymptotic convergence
- LPBoost uses a linear programming solver
  - Often very fast in practice, no converge rates (new: $\Omega(N)$)
- AdaBoost* requires $\frac{2\log(N)}{\delta^2}$ iterations to get $\rho^t \in [\rho^* - \delta, \rho^*]$
  - Slow in practice, i.e. not faster than theory predicts
- TotalBoost requires $\frac{2\log(N)}{\delta^2}$ iterations to get $\rho^t \in [\rho^* - \delta, \rho^*]$
  - Needs convex optimizer, fast in practice
  - Combination of benefits

[Breiman, 1999; Bennett et al.; G.R. et al., 2001, 2004, 2006; Rudin et al., 2004]

How to Maximize the Margin?

- AdaBoost does not generally maximize the margin
  - Slow and may cycle
- Arc-GV
  - Rela
- LPBoost
  - Ofte : $\Omega(N)$)
- AdaBoos $-\delta, \rho^*]$
  - Slow in practice, i.e. not faster than theory predicts
- TotalBoost requires $\frac{2\log(N)}{\delta^2}$ iterations to get $\rho^t \in [\rho^* - \delta, \rho^*]$
  - Needs convex optimizer, fast in practice
  - Combination of benefits

Convergence speeds can be truly different!

- LPBoost/TotalBoost may need 10-50 instead of 10,000 iterations
- LPBoost can be much slower than TotalBoost

[Breiman, 1999; Bennett et al.; G.R. et al., 2001, 2004, 2006; Rudin et al., 2004]

Improvements . . . ?

Does it Work Better?

- **Usually not!** Algorithms just overfit quicker.
- Only slight improvements in the noise free case.

Soft Margins for AdaBoost

1. Limit the *influence* of examples ("AdaBoost$_{\text{reg}}$")
   - Heuristic algorithm, no convergence result . . .
   - . . . but works very well in practice
2. Soft margins à la SVMs ("ν-Arc", "SoftBoost")
   - Change optimization problem to include slacks
   - Convergence proofs did not work $\Rightarrow$ only asymptotic results

[G.R. et al. 2000-2005, Warmuth, Glocer & G.R., 2007]

Improvements . . . ?

Does it Work Better?

- **Usually not!** Algorithms just overfit quicker.
- Only slight improvements in the noise free case.

Soft Margins for AdaBoost

1. Limit the *influence* of examples ("$\text{AdaBoost}_{\text{reg}}$")
   - Heuristic algorithm, no convergence result . . .
   - . . . but works very well in practice
2. Soft margins à la SVMs ("ν-Arc", "SoftBoost")
   - Change optimization problem to include slacks
   - Convergence proofs did not work ⇒ only asymptotic results
   - News: Can prove fast convergence with soft margin (NIPS'07)

[G.R. et al. 2000-2005, Warmuth, Glocer & G.R., 2007]

Improvements . . . ?

Does it Work Better?

- **Usually not!** Algorithms just overfit quicker.
- Only slight improvements in the noise free case.

Soft Margins for AdaBoost

1. Limit the *influence* of examples ("AdaBoost$_{\text{reg}}$")
   - Heuristic algorithm, no convergence result . . .
   - . . . but works very well in practice
2. Soft margins à la SVMs ("ν-Arc", "SoftBoost")
   - Change optimization problem to include slacks
   - Convergence proofs did not work $\Rightarrow$ only asymptotic results
   - **New:** Can prove fast convergence with soft margin (NIPS'07)

[G.R. et al. 2000-2005, Warmuth, Glocer & G.R., 2007]

Improvements . . . ?

Does it Work Better?

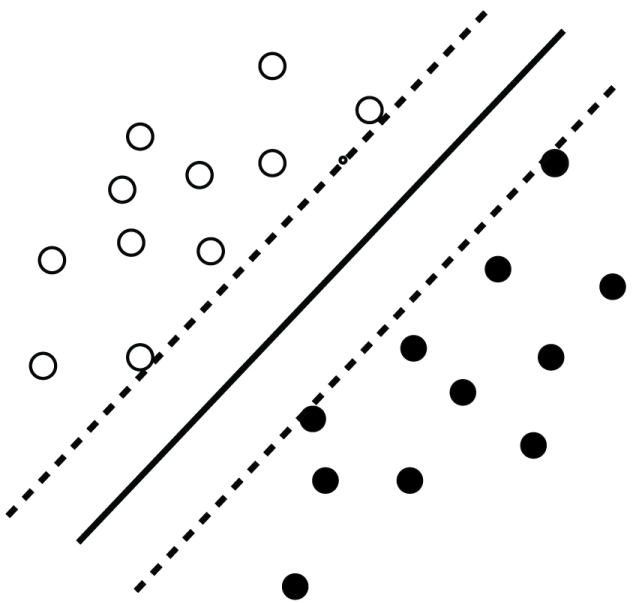
- **Usually not!** Algorithms just overfit quicker.
- Only slight improvements in the noise free case.

Soft Margins for AdaBoost

1. Limit the *influence* of examples ("AdaBoost$_{\text{reg}}$")
   - Heuristic algorithm, no convergence result . . .
   - . . . but works very well in practice
2. Soft margins à la SVMs ("ν-Arc", "SoftBoost")
   - Change optimization problem to include slacks
   - Convergence proofs did not work $\Rightarrow$ only asymptotic results
   - **New:** Can prove fast convergence with soft margin (NIPS'07)

[G.R. et al. 2000-2005, Warmuth, Glocer & G.R., 2007]

SoftBoost = TotalBoost with ν-Trick

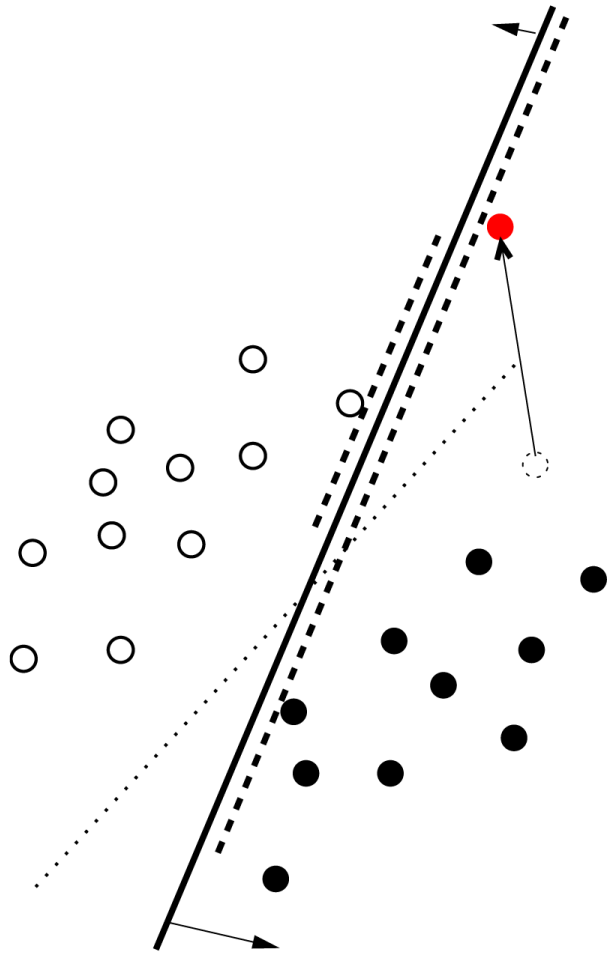


Hard-margin separation

Comparison of AdaBoost/Softboost on UCI data sets (classification error):

	AdaBoost	SoftBoost
Banana	13.3%	**11.1%**
B.Cancer	32.1%	**28.0%**
Diabetes	27.9%	**24.4%**
German	26.9%	**24.7%**
Heart	20.1%	**18.2%**
Ringnorm	1.9%	**1.8%**
F.Sonar	36.1%	**35.5%**
Thyroid	**4.4%**	4.9%
Titanic	**22.8%**	23.0%
Waveform	10.5%	**9.8%**

SoftBoost = TotalBoost with ν-Trick

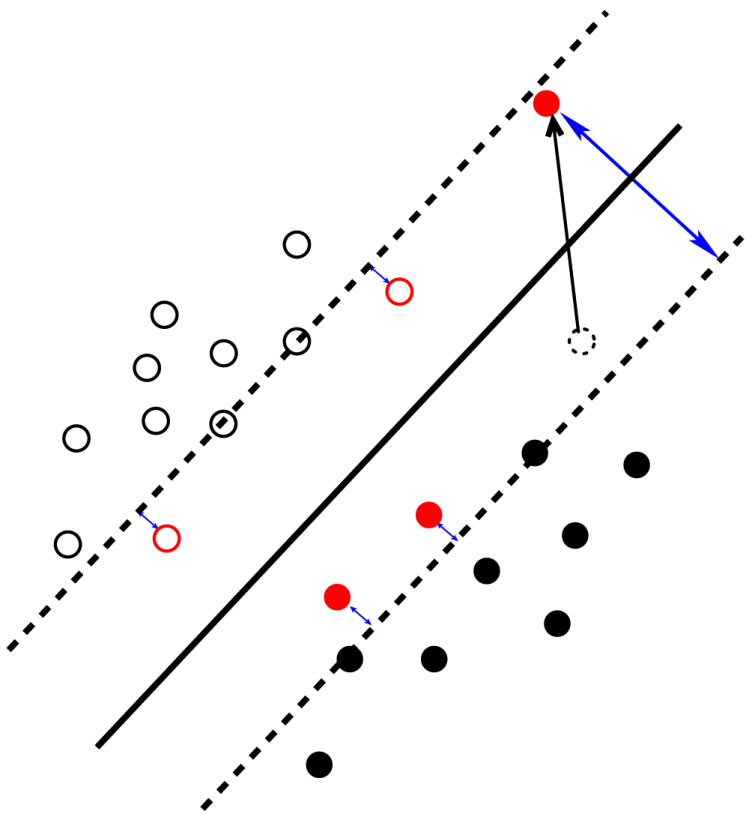


Hard-margin separation with outlier

Comparison of AdaBoost/Softboost on UCI data sets (classification error):

	AdaBoost	SoftBoost
Banana	13.3%	**11.1%**
B.Cancer	32.1%	**28.0%**
Diabetes	27.9%	**24.4%**
German	26.9%	**24.7%**
Heart	20.1%	**18.2%**
Ringnorm	1.9%	**1.8%**
F.Sonar	36.1%	**35.5%**
Thyroid	**4.4%**	4.9%
Titanic	**22.8%**	23.0%
Waveform	10.5%	**9.8%**

SoftBoost = TotalBoost with ν-Trick



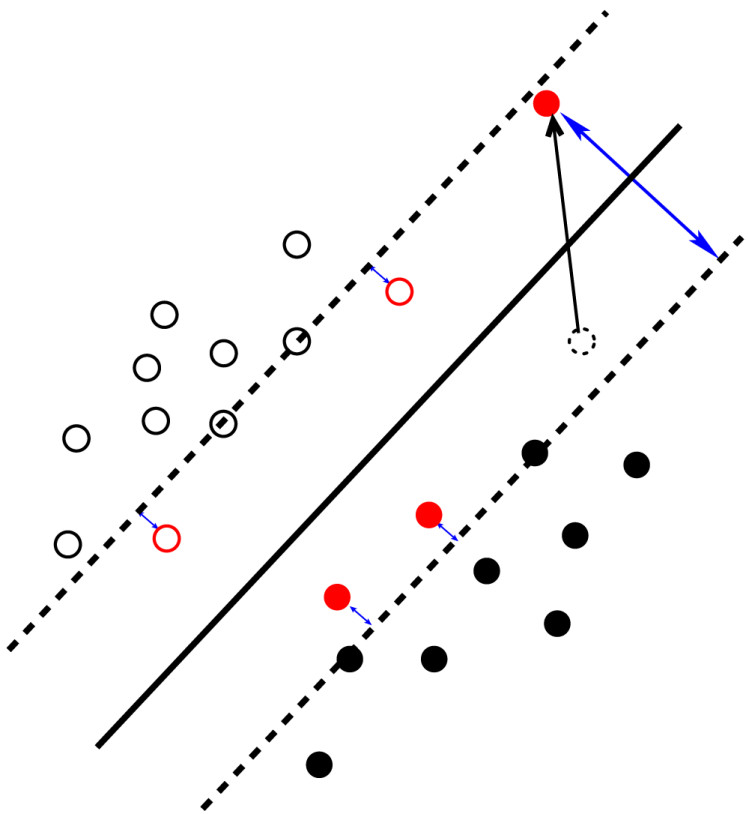
Comparison of AdaBoost/Softboost on UCI data sets (classification error):

	AdaBoost	SoftBoost
Banana	13.3%	**11.1%**
B.Cancer	32.1%	**28.0%**
Diabetes	27.9%	**24.4%**
German	26.9%	**24.7%**
Heart	20.1%	**18.2%**
Ringnorm	1.9%	**1.8%**
F.Sonar	36.1%	**35.5%**
Thyroid	**4.4%**	4.9%
Titanic	**22.8%**	23.0%
Waveform	10.5%	**9.8%**

Soft-margin separation with outlier

Chose margin such that a fraction ν is within the margin area

SoftBoost = TotalBoost with ν-Trick



Comparison of AdaBoost/Softboost on UCI data sets (classification error):

	AdaBoost	SoftBoost
Banana	13.3%	**11.1%**
B.Cancer	32.1%	**28.0%**
Diabetes	27.9%	**24.4%**
German	26.9%	**24.7%**
Heart	20.1%	**18.2%**
Ringnorm	1.9%	**1.8%**
F.Sonar	36.1%	**35.5%**
Thyroid	**4.4%**	4.9%
Titanic	**22.8%**	23.0%
Waveform	10.5%	**9.8%**

Soft-margin separation with outlier

Chose margin such that a fraction ν is within the margin area

So, All Work is Done . . . !?

- SoftBoost combines all benefits [Warmuth, Glocer, G.R., NIPS, 2007]
  - Noise robustness
  - Good rates of convergence
  - Few iterations in practice
- Let's apply it!
  - Benchmarks
  - Drug Discovery (with K. Bennett, K.-R. Müller)
  - Electric appliances monitoring (with T. Onoda)
    - Non-intrusively classify the on/off state of appliances
    - Patented and in use in Japan
  - Biological sequence analysis
    - Classify sequences (e.g. exons vs. introns)
    - Features correspond to subsequences
    - Boosting/LP based methods fail in practical cases
    - Why?

So, All Work is Done . . . !?

- SoftBoost combines all benefits [Warmuth, Glocer, G.R., NIPS, 2007]
  - Noise robustness
  - Good rates of convergence
  - Few iterations in practice
- Let's apply it!
  - \+ Benchmarks
  - \+ Drug Discovery (with K. Bennett, K.-R. Müller)
  - \+ Electric appliances monitoring (with T. Onoda)
    - Non-intrusively classify the on/off state of appliances
    - Patented and in use in Japan
  - − Biological sequence analysis
    - Classify sequences (e.g. exons vs. introns)
    - Features correspond to subsequences
    - Boosting/LP based methods fail in practical cases
    - Why?

So, All Work is Done . . . !?

- SoftBoost combines all benefits [Warmuth, Glocer, G.R., NIPS, 2007]
  - Noise robustness
  - Good rates of convergence
  - Few iterations in practice
- Let's apply it!
  - \+ Benchmarks
  - \+ Drug Discovery (with K. Bennett, K.-R. Müller)
  - \+ Electric appliances monitoring (with T. Onoda)
    - Non-intrusively classify the on/off state of appliances
    - Patented and in use in Japan
  - − Biological sequence analysis
    - Classify sequences (e.g. exons vs. introns)
    - Features correspond to subsequences
    - Boosting/LP based methods fail in practical cases
    - Why?

So, All Work is Done . . . !?

- SoftBoost combines all benefits [Warmuth, Glocer, G.R., NIPS, 2007]
  - Noise robustness
  - Good rates of convergence
  - Few iterations in practice
- Let's apply it!
  - \+ Benchmarks
  - \+ Drug Discovery (with K. Bennett, K.-R. Müller)
  - \+ Electric appliances monitoring (with T. Onoda)
    - Non-intrusively classify the on/off state of appliances
    - Patented and in use in Japan
  - − Biological sequence analysis
    - Classify sequences (e.g. exons vs. introns)
    - Features correspond to subsequences
    - Boosting/LP based methods fail in practical cases
    - Why?

So, All Work is Done . . . !?

- SoftBoost combines all benefits [Warmuth, Glocer, G.R., NIPS, 2007]
  - Noise robustness
  - Good rates of convergence
  - Few iterations in practice
- Let's apply it!
  - \+ Benchmarks
  - \+ Drug Discovery (with K. Bennett, K.-R. Müller)
  - \+ Electric appliances monitoring (with T. Onoda)
    - Non-intrusively classify the on/off state of appliances
    - Patented and in use in Japan
  - – Biological sequence analysis
    - Classify sequences (e.g. exons vs. introns)
    - Features correspond to subsequences
    - Boosting/LP based methods fail in practical cases
    - Why?

Illustrative Example: Classify Sequences

Task 1

- Substring discriminates, e.g. `GATTACA`
- Optimal weight vector **sparse**
- Results
  - \+ Boosting/LP
  - − SVM
- Sparseness beneficial
  - Quick identification of discriminating dimensions

Task 2

- Differing substring frequencies
- Optimal weight vector **dense**
- Results
  - − Boosting/LP
  - \+ SVM
- Sparseness detrimental
  - Concentration on too few dimensions

Discrimination of biological signals' sequences

- Needs many different features
- Kernel methods often work better

Illustrative Example: Classify Sequences

Task 1

- Substring discriminates, e.g. `GATTACA`
- Optimal weight vector **sparse**
- Results
  - \+ Boosting/LP
  - − SVM
- Sparseness beneficial
  - Quick identification of discriminating dimensions

Task 2

- Differing substring frequencies
- Optimal weight vector **dense**
- Results
  - − Boosting/LP
  - \+ SVM
- Sparseness detrimental
  - Concentration on too few dimensions

Discrimination of biological signals' sequences

- Needs many different features
- Kernel methods often work better

Illustrative Example: Classify Sequences

Task 1

- Substring discriminates, e.g. `GATTACA`
- Optimal weight vector **sparse**
- Results
  - \+ Boosting/LP
  - − SVM
- Sparseness beneficial
  - Quick identification of discriminating dimensions

Task 2

- Differing substring frequencies
- Optimal weight vector **dense**
- Results
  - − Boosting/LP
  - \+ SVM
- Sparseness detrimental
  - Concentration on too few dimensions

Discrimination of biological signals' sequences

- Needs many different features
- Kernel methods often work better

Illustrative Example: Classify Sequences

Task 1

- Substring discriminates, e.g. `GATTACA`
- Optimal weight vector **sparse**
- Results
  - \+ Boosting/LP
  - − SVM
- Sparseness beneficial
  - Quick identification of discriminating dimensions

Task 2

- Differing substring frequencies
- Optimal weight vector **dense**
- Results
  - − Boosting/LP
  - \+ SVM
- Sparseness detrimental
  - Concentration on too few dimensions

Discrimination of biological signals' sequences

- Needs many different features
- Kernel methods often work better

Illustrative Example: Classify Sequences

Task 1

- Substring discriminates, e.g. `GATTACA`
- Optimal weight vector **sparse**
- Results
  - \+ Boosting/LP
  - − SVM
- Sparseness beneficial
  - Quick identification of discriminating dimensions

Task 2

- Differing substring frequencies
- Optimal weight vector **dense**
- Results
  - − Boosting/LP
  - \+ SVM
- Sparseness detrimental
  - Concentration on too few dimensions

Discrimination of biological signals' sequences

- Needs many different features
- Kernel methods often work better

Biological Sequence Analysis

- New kernels and large scale methods for biological sequences
  - Weighted degree kernels, . . . [G.R. and Sonnenburg, KMCB, 2004]
  - Training with 10,000,000 examples [Sonnenburg et al., BMC Bioinformatics, 2007]
- Combining Boosting & SVMs: Sparseness on feature groups
  - Multiple kernel learning [Sonnenburg, G.R., Schäfer, Schölkopf, JMLR, 2006]
  - Useful for visualization [G.R., Sonnenburg, Schäfer, BMC Bioinformatics, 2006]
- Novel algorithms for structure predictions
  - Apply boosting-like techniques to solve huge LPs
  - Large Scale Hidden semi-Markov SVMs [G.R. and Sonnenburg, NIPS, 2006]
- **Applications**
  - Novel alignment algorithms [Schulze, Hepp, Ong, G.R., Bioinformatics, 2007]
  - Gene finding [G.R. et al, PLoS Computational Biology, 2007]
  - Detection of polymorphisms in genomes [Clark et al., Science, 2007]
  - . . .

Biological Sequence Analysis

- New kernels and large scale methods for biological sequences
  - Weighted degree kernels, . . . [G.R. and Sonnenburg, KMCB, 2004]
  - Training with 10,000,000 examples [Sonnenburg et al., BMC Bioinformatics, 2007]
- Combining Boosting & SVMs: Sparseness on feature groups
  - Multiple kernel learning [Sonnenburg, G.R., Schäfer, Schölkopf, JMLR, 2006]
  - Useful for visualization [G.R., Sonnenburg, Schäfer, BMC Bioinformatics, 2006]
- Novel algorithms for structure predictions
  - Apply boosting-like techniques to solve huge LPs
  - Large Scale Hidden semi-Markov SVMs [G.R. and Sonnenburg, NIPS, 2006]
- **Applications**
  - Novel alignment algorithms [Schulze, Hepp, Ong, G.R., Bioinformatics, 2007]
  - Gene finding [G.R. et al, PLoS Computational Biology, 2007]
  - Detection of polymorphisms in genomes [Clark et al., Science, 2007]
  - . . .

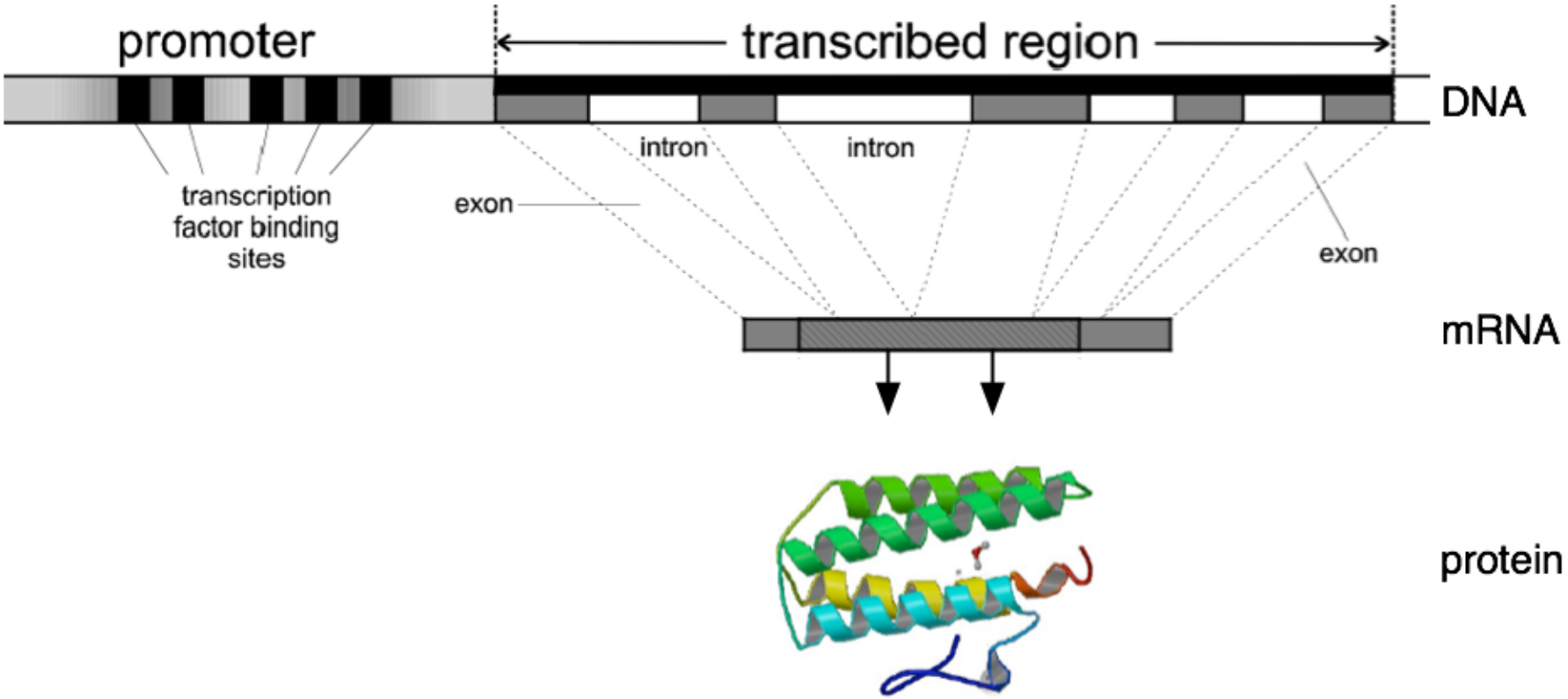
Biological Sequence Analysis

- New kernels and large scale methods for biological sequences
  - Weighted degree kernels, ... [G.R. and Sonnenburg, KMCB, 2004]
  - Training with 10,000,000 examples [Sonnenburg et al., BMC Bioinformatics, 2007]
- Combining Boosting & SVMs: Sparseness on feature groups
  - Multiple kernel learning [Sonnenburg, G.R., Schäfer, Schölkopf, JMLR, 2006]
  - Useful for visualization [G.R., Sonnenburg, Schäfer, BMC Bioinformatics, 2006]
- Novel algorithms for structure predictions
  - Apply boosting-like techniques to solve huge LPs
  - Large Scale Hidden semi-Markov SVMs [G.R. and Sonnenburg, NIPS, 2006]
- **Applications**
  - Novel alignment algorithms [Schulze, Hepp, Ong, G.R., Bioinformatics, 2007]
  - Gene finding [G.R. et al, PLoS Computational Biology, 2007]
  - Detection of polymorphisms in genomes [Clark et al., Science, 2007]
  - ...

Biological Sequence Analysis

- New kernels and large scale methods for biological sequences
  - Weighted degree kernels, ... [G.R. and Sonnenburg, KMCB, 2004]
  - Training with 10,000,000 examples [Sonnenburg et al., BMC Bioinformatics, 2007]
- Combining Boosting & SVMs: Sparseness on feature groups
  - Multiple kernel learning [Sonnenburg, G.R., Schäfer, Schölkopf, JMLR, 2006]
  - Useful for visualization [G.R., Sonnenburg, Schäfer, BMC Bioinformatics, 2006]
- Novel algorithms for structure predictions
  - Apply boosting-like techniques to solve huge LPs
  - Large Scale Hidden semi-Markov SVMs [G.R. and Sonnenburg, NIPS, 2006]
- **Applications**
  - Novel alignment algorithms [Schulze, Hepp, Ong, G.R., Bioinformatics, 2007]
  - Gene finding [G.R. et al, PLoS Computational Biology, 2007]
  - Detection of polymorphisms in genomes [Clark et al., Science, 2007]
  - ...

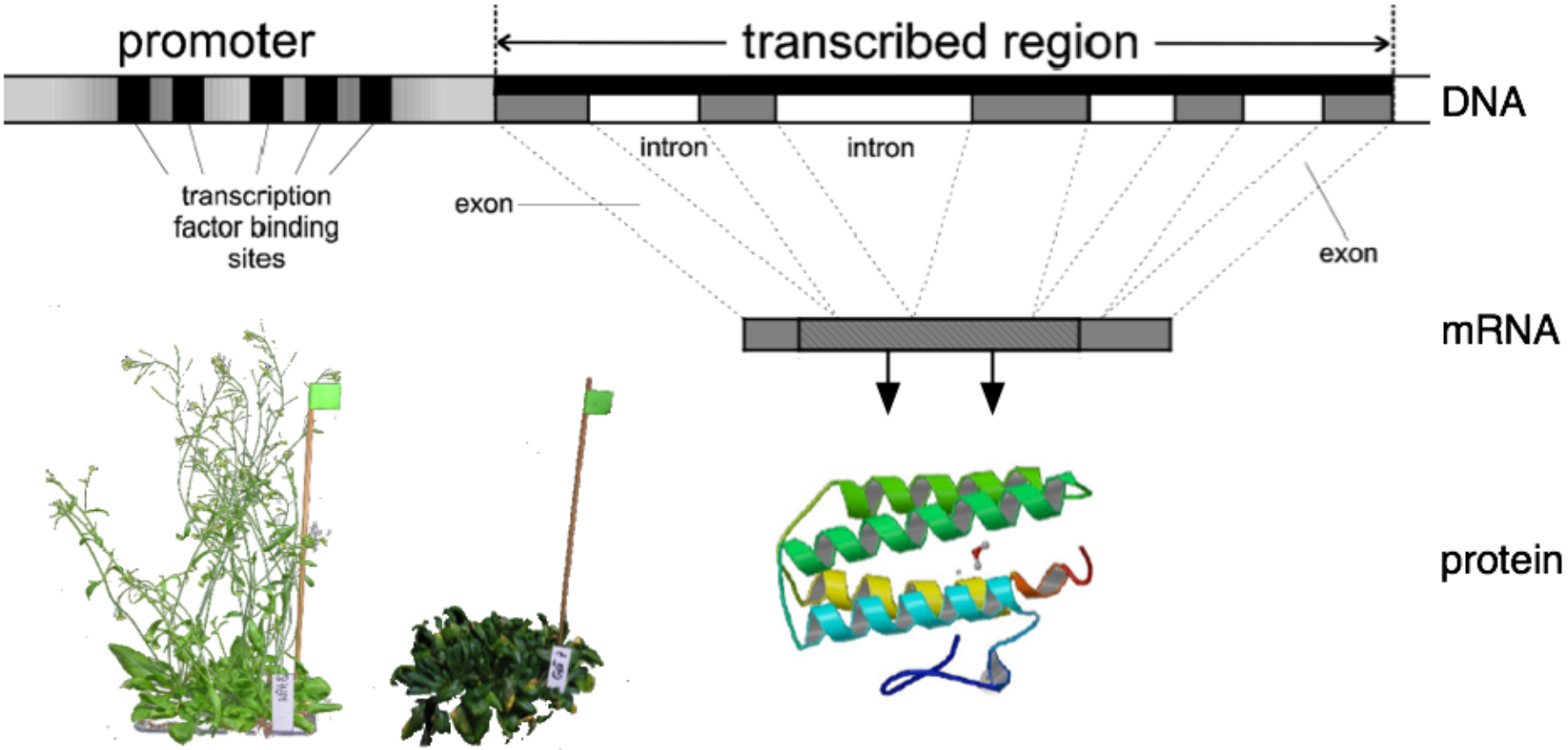
Understanding Genes and Natural Variation



- Understand differences between individuals
- New technologies bring plethora of new data/problems

[G.R. et al, PLoS Comp. Biol., 2007, Clark et al., Science, 2007]

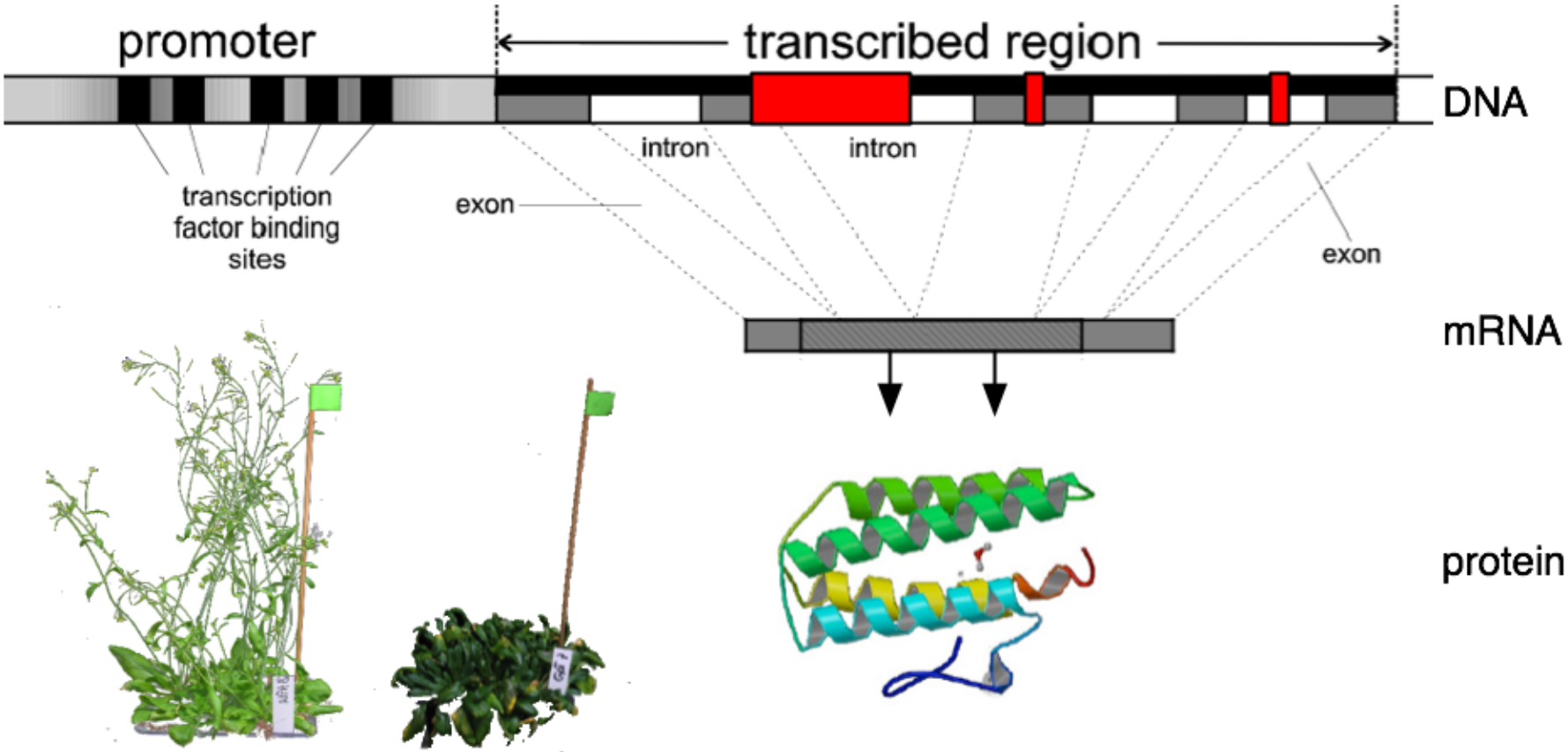
Understanding Genes and Natural Variation



- Understand differences between individuals
- New technologies bring plethora of new data/problems

[G.R. et al, PLoS Comp. Biol., 2007, Clark et al., Science, 2007]

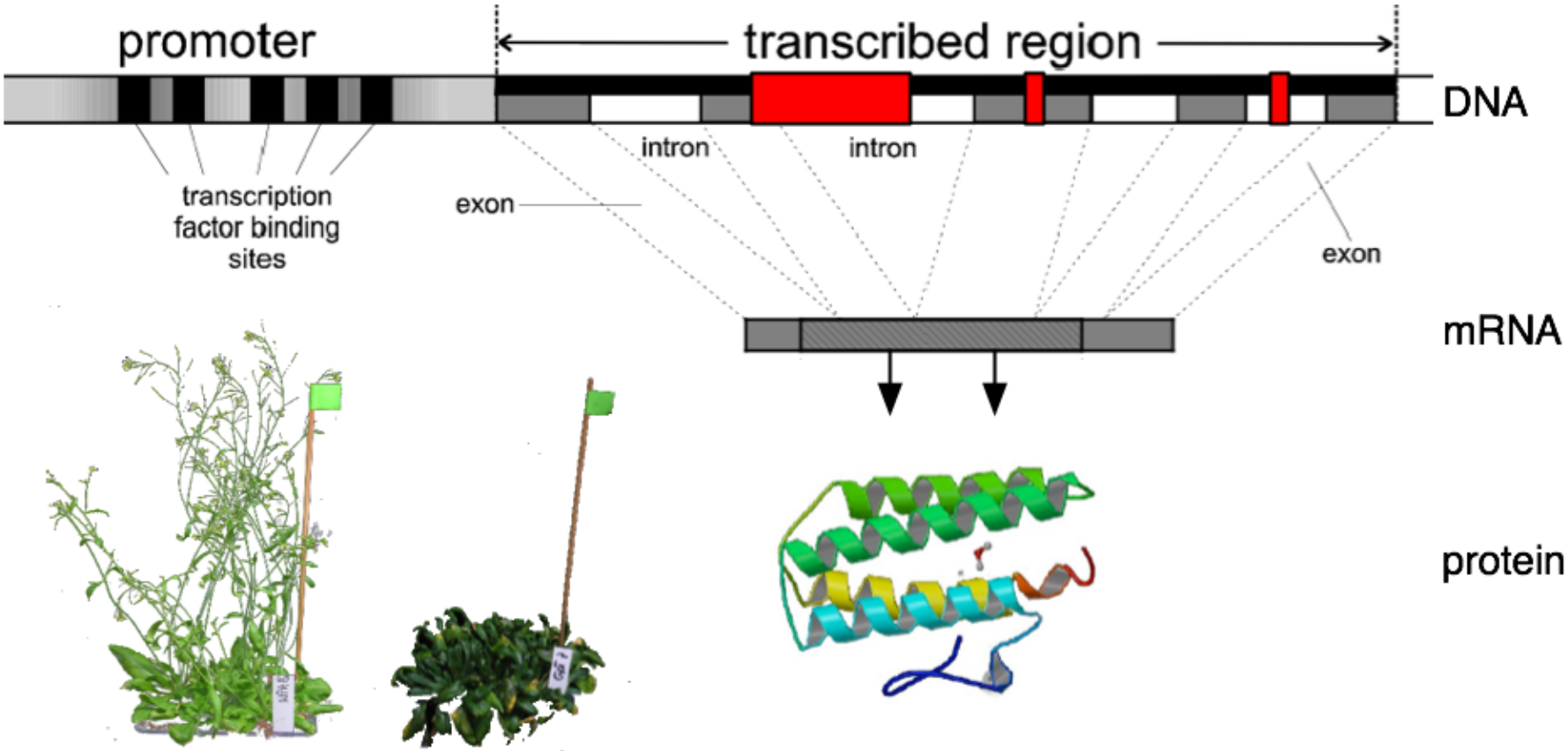
Understanding Genes and Natural Variation



- Understand differences between individuals
- New technologies bring plethora of new data/problems

[G.R. et al, PLoS Comp. Biol., 2007, Clark et al., Science, 2007]

Understanding Genes and Natural Variation



- Understand differences between individuals
- New technologies bring plethora of new data/problems

  [G.R. et al, PLoS Comp. Biol., 2007, Clark et al., Science, 2007]

Acknowledgments

- **Collaborators:**
  Kristin Bennett, Richard Clark, Ayhan Demiriz, Jun Liao, Sebastian Mika, Klaus-Robert Müller, Takashi Onoda, Bernhard Schölkopf, Sören Sonnenburg, Ralf Sommer, Manfred Warmuth, Detlef Weigel, ...
- **Group Members:**
  Fabio de Bona, Elisabeth Georgii, Cheng Soon Ong, Petra Philips, Gabriele Schweikert, Sebastian Schultheiss, Georg Zeller, Alexander Zien, ...
- **Mentors:**
  Klaus-Robert Müller, Bernhard Schölkopf and Manfred Warmuth
- **Thank You!**

Slides with references:

`http://www.fml.mpg.de/~raetsch/olympus`

References I

Y. Altun, I. Tsochantaridis, and T. Hofmann. Hidden Markov support vector machines. In *Proc. ICML'03*, pages 3–10. AAAI Press, 2003. ISBN 1-57735-189-4.

F. Bach, G. Lanckriet, and M. Jordan. Multiple kernel learning, conic duality, and the SMO algorithm. In C. E. Brodley, editor, *Proc. ICML'04*. ACM, 2004.

C. Cortes and V.N. Vapnik. Support vector networks. *Machine Learning*, 20:273–297, 1995.

A. Demiriz, K.P. Bennett, and J. Shawe-Taylor. Linear programming boosting via column generation. *Machine Learning*, 46:225–254, 2002.

Y. Freund and R.E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, 55(1):119–139, 1997.

References II

R. Hettich and K.O. Kortanek. Semi-infinite programming: Theory, methods and applications. *SIAM Review*, 3:380–429, September 1993.

G. Lanckriet, N. Cristianini, L. Ghaoui, P. Bartlett, and M. Jordan. Learning the kernel matrix with semidefinite programming. *Journal of Machine Learning Research*, 5:27–72, 2004.

R. Meir and G. Rätsch. An introduction to boosting and leveraging. In S. Mendelson and A. Smola, editors, *Advanced Lectures on Machine Learning*, LNCS, pages 119–184. Springer, 2003.

G. Rätsch. *Robust Boosting via Convex Optimization*. PhD thesis, University of Potsdam, Neues Palais 10, 14469 Potsdam, Germany, October 2001.

G. Rätsch and M.K. Warmuth. Efficient margin maximization with boosting. *Journal of Machine Learning Research*, 6:2131–2152, 2005.

References III

G. Rätsch, A. Demiriz, and K. Bennett. Sparse regression ensembles in infinite and finite hypothesis spaces. *Machine Learning*, 48(1-3): 193–221, 2002.

S. Sonnenburg, G. Rätsch, C. Schäfer, and B. Schölkopf. Large scale multiple kernel learning. *Journal of Machine Learning Research*, pages 1531–1565, July 2006.

M.K. Warmuth, J. Liao, and G. Rätsch. Totally corrective boosting algorithms that maximize the margin. In W. Cohen and A. Moore, editors, *Proc. ICML'06*, pages 1001–1008. ACM Press, 2006.