Boosting, Margins and Beyond

Gunnar Rätsch
Friedrich Miescher Laboratory, Max Planck Society, Tübingen, Germany

September 12, 2007

29th Annual Symposium of the German Association for Pattern Recognition
The AdaBoost Algorithm

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

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

[Freund & Schapire, 1995]
Boosting: 1st Iteration

Error rate of 1\textsuperscript{st} hypothesis:
\[ \epsilon = \sum_{n=1}^{N} d_{n}^{t} \mathbb{I}(h_{t}(x_{n}) \neq y_{n}) \]
\[ = \frac{2}{11} \]
Misclassified examples ⇒ Increase weights

Error after update:

\[ \epsilon' = \sum_{n=1}^{N} d_n^{t+1} \mathbb{I}(h_t(x_n) \neq y_n) \]

\[ = \frac{1}{2} \]
Update Distribution

Misclassified examples ⇒ Increase weights

Error after update:

\[
\epsilon' = \sum_{n=1}^{N} d_{n}^{t+1} I(h_t(x_n) \neq y_n)
\]

\[
= \frac{1}{2}
\]
Before 2nd Iteration
Boosting: 2nd Hypothesis ($\epsilon < \frac{1}{2}$)
Update Distribution \( (\epsilon' = \frac{1}{2}) \)
Boosting: 3rd Hypothesis

[Diagram with axes labeled "leicht" and "schwer" on the y-axis and "nicht rot" and "sehr rot" on the x-axis, with various apple images scattered across the chart.]
Boosting: 4th Hypothesis

The diagram illustrates a classification task with two classes: "leicht" (light) and "schwer" (heavy), and two attributes: "nicht rot" (not red) and "sehr rot" (very red). The diagram shows a range of samples with varying degrees of lightness and redness, indicating a transition from non-red to very red.
All Hypotheses
Decision: $f_\alpha(x) = \sum_{t=1}^{T} \alpha_t h_t(x) > 0$?
Linear Separation and Large Margins

Input space $\mathcal{X}$

Feature space $\mathcal{F}$

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

Margin of example $(x, y)$:

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

[Mangasarian, 1999; G.R., Mika, Schölkopf & Müller, 2002]
Linear Separation and Large Margins

Input space $\mathcal{X}$

Feature space $\mathcal{F}$

$\Phi(x) = \begin{pmatrix} h_1(x) \\ h_2(x) \\ \vdots \end{pmatrix}$

Margin of example $(x, y)$:

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

[Mangasarian, 1999; G.R., Mika, Schölkopf & Müller, 2002]
Linear Separation and Large Margins

Input space $\mathcal{X}$

Feature space $\mathcal{F}$

Margin of example $(x, y)$:

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

[Ma"{n}gasarian, 1999; G.R., Mika, Sch"{o}lkopf & M"{u}ller, 2002]
# Margin vs. Edge

## Margin

- Measure for "confidence" in prediction of example $n$: $y_n f_\alpha(x_n)$
- Depends on hypothesis weighting $\alpha$

## Edge

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

## What is the connection?

von Neumann's Minimax-Theorem

- minimum edge = maximum margin

[von Neumann, 1928, Breiman, 1999]
Margin vs. Edge

**Margin**
- Measure for “confidence” in prediction of example $n$: $y_n f_\alpha(x_n)$
- Depends on hypothesis weighting $\alpha$

**Edge**
- “Goodness” of hypothesis returned by learner: $\gamma = 1 - 2\epsilon$
- Depends on distribution $d$ on examples

What is the connection?

von Neumann’s Minimax-Theorem

\[
\text{minimum edge} = \text{maximum margin}
\]

[von Neumann, 1928, Breiman, 1999]
Margin vs. Edge

**Margin**
- Measure for “confidence” in prediction of example $n$: $y_n f_{\alpha}(x_n)$
- Depends on hypothesis weighting $\alpha$

**Edge**
- “Goodness” of hypothesis returned by learner: $\gamma = 1 - 2\epsilon$
- Depends on distribution $d$ on examples

What is the connection?

von Neumann’s Minimax-Theorem

minimum edge = maximum margin

[von Neumann, 1928, Breiman, 1999]
Margin vs. Edge

Margin
- Measure for “confidence” in prediction of example $n$: $y_n f_\alpha(x_n)$
- Depends on hypothesis weighting $\alpha$

Edge
- “Goodness” of hypothesis returned by learner: $\gamma = 1 - 2\epsilon$
- Depends on distribution $d$ on examples

What is the connection?

von Neumann’s Minimax-Theorem

\[
\text{minimum edge} = \text{maximum margin}
\]

[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

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

---

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

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

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

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
    - \( \Omega(N) \)
- **AdaBoost** requires \( 2 \log(N) \) 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

Convergence speeds can be truly different!

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

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 ("$\nu$-Arc", "SoftBoost")
   - Change optimization problem to include slacks
   - Convergence proofs did not work $\Rightarrow$ only asymptotic results

## 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$_{reg}$")**
   - Heuristic algorithm, no convergence result . . .
   - ...but works very well in practice

2. **Soft margins à la SVMs ("$\nu$-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)

---

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 ("$\nu$-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)

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 ("$\nu$-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)

**SoftBoost = TotalBoost with $\nu$-Trick**

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

<table>
<thead>
<tr>
<th>Dataset</th>
<th>AdaBoost</th>
<th>SoftBoost</th>
</tr>
</thead>
<tbody>
<tr>
<td>Banana</td>
<td>13.3%</td>
<td>11.1%</td>
</tr>
<tr>
<td>B.Cancer</td>
<td>32.1%</td>
<td>28.0%</td>
</tr>
<tr>
<td>Diabetes</td>
<td>27.9%</td>
<td>24.4%</td>
</tr>
<tr>
<td>German</td>
<td>26.9%</td>
<td>24.7%</td>
</tr>
<tr>
<td>Heart</td>
<td>20.1%</td>
<td>18.2%</td>
</tr>
<tr>
<td>Ringnorm</td>
<td>1.9%</td>
<td>1.8%</td>
</tr>
<tr>
<td>F.Sonar</td>
<td>36.1%</td>
<td>35.5%</td>
</tr>
<tr>
<td>Thyroid</td>
<td>4.4%</td>
<td>4.9%</td>
</tr>
<tr>
<td>Titanic</td>
<td>22.8%</td>
<td>23.0%</td>
</tr>
<tr>
<td>Waveform</td>
<td>10.5%</td>
<td>9.8%</td>
</tr>
</tbody>
</table>

Hard-margin separation
SoftBoost = TotalBoost with $\nu$-Trick

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

<table>
<thead>
<tr>
<th></th>
<th>AdaBoost</th>
<th>SoftBoost</th>
</tr>
</thead>
<tbody>
<tr>
<td>Banana</td>
<td>13.3%</td>
<td>11.1%</td>
</tr>
<tr>
<td>B.Cancer</td>
<td>32.1%</td>
<td>28.0%</td>
</tr>
<tr>
<td>Diabetes</td>
<td>27.9%</td>
<td>24.4%</td>
</tr>
<tr>
<td>German</td>
<td>26.9%</td>
<td>24.7%</td>
</tr>
<tr>
<td>Heart</td>
<td>20.1%</td>
<td>18.2%</td>
</tr>
<tr>
<td>Ringnorm</td>
<td>1.9%</td>
<td>1.8%</td>
</tr>
<tr>
<td>F.Sonar</td>
<td>36.1%</td>
<td>35.5%</td>
</tr>
<tr>
<td>Thyroid</td>
<td>4.4%</td>
<td>4.9%</td>
</tr>
<tr>
<td>Titanic</td>
<td>22.8%</td>
<td>23.0%</td>
</tr>
<tr>
<td>Waveform</td>
<td>10.5%</td>
<td>9.8%</td>
</tr>
</tbody>
</table>

Hard-margin separation with outlier
**SoftBoost = TotalBoost with \( \nu \)-Trick**

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

<table>
<thead>
<tr>
<th>Dataset</th>
<th>AdaBoost</th>
<th>SoftBoost</th>
</tr>
</thead>
<tbody>
<tr>
<td>Banana</td>
<td>13.3%</td>
<td>11.1%</td>
</tr>
<tr>
<td>B.Cancer</td>
<td>32.1%</td>
<td>28.0%</td>
</tr>
<tr>
<td>Diabetes</td>
<td>27.9%</td>
<td>24.4%</td>
</tr>
<tr>
<td>German</td>
<td>26.9%</td>
<td>24.7%</td>
</tr>
<tr>
<td>Heart</td>
<td>20.1%</td>
<td>18.2%</td>
</tr>
<tr>
<td>Ringnorm</td>
<td>1.9%</td>
<td>1.8%</td>
</tr>
<tr>
<td>F.Sonar</td>
<td>36.1%</td>
<td>35.5%</td>
</tr>
<tr>
<td>Thyroid</td>
<td>4.4%</td>
<td>4.9%</td>
</tr>
<tr>
<td>Titanic</td>
<td>22.8%</td>
<td>23.0%</td>
</tr>
<tr>
<td>Waveform</td>
<td>10.5%</td>
<td>9.8%</td>
</tr>
</tbody>
</table>

**Soft-margin separation with outlier**

Chose margin such that a fraction \( \nu \) is within the margin area.
SoftBoost = TotalBoost with $\nu$-Trick

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

<table>
<thead>
<tr>
<th>Dataset</th>
<th>AdaBoost</th>
<th>SoftBoost</th>
</tr>
</thead>
<tbody>
<tr>
<td>Banana</td>
<td>13.3%</td>
<td>11.1%</td>
</tr>
<tr>
<td>B.Cancer</td>
<td>32.1%</td>
<td>28.0%</td>
</tr>
<tr>
<td>Diabetes</td>
<td>27.9%</td>
<td>24.4%</td>
</tr>
<tr>
<td>German</td>
<td>26.9%</td>
<td>24.7%</td>
</tr>
<tr>
<td>Heart</td>
<td>20.1%</td>
<td>18.2%</td>
</tr>
<tr>
<td>Ringnorm</td>
<td>1.9%</td>
<td>1.8%</td>
</tr>
<tr>
<td>F.Sonar</td>
<td>36.1%</td>
<td>35.5%</td>
</tr>
<tr>
<td>Thyroid</td>
<td>4.4%</td>
<td>4.9%</td>
</tr>
<tr>
<td>Titanic</td>
<td>22.8%</td>
<td>23.0%</td>
</tr>
<tr>
<td>Waveform</td>
<td>10.5%</td>
<td>9.8%</td>
</tr>
</tbody>
</table>

Soft-margin separation with outlier

Chose margin such that a fraction $\nu$ 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
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?
Beyond

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?
Beyond

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]
  - Detection of polymorphisms in genomes [Clark et al., Science, 2007]
  - . . .

Gunnar Rätsch (FML, Tübingen)
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 [G.R. and Sonnenburg, NIPS, 2006]
  - Large Scale Hidden semi-Markov SVMs
- Applications
  - Novel alignment algorithms [Schulze, Hepp, Ong, G.R., Bioinformatics, 2007]
  - Detection of polymorphisms in genomes [Clark et al., Science, 2007]
  - . . .
Beyond

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]
  - Detection of polymorphisms in genomes  [Clark et al., Science, 2007]
  - . . .
Beyond 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]
  - 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

Beyond Understanding Genes and Natural Variation

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

Understanding Genes and Natural Variation

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

Beyond Understanding Genes and Natural Variation

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

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:**

- **Mentors:**
  Klaus-Robert Müller, Bernhard Schölkopf and Manfred Warmuth

- **Thank You!**

Slides with references:
http://www.fml.mpg.de/~raetsch/olympus
References I


References II


