Advanced Methods for Sequence Analysis in Bioinformatics

Gunnar Rätsch¹, Cheng Soon Ong¹,², Petra Philips¹

¹ Friedrich Miescher Laboratory, Tübingen
² MPI for Biological Cybernetics, Tübingen

Vorlesung WS 2007/2008
Eberhard-Karls-Universität Tübingen

16 October 2007

http://www.fml.mpg.de/raetsch/lectures/amsa07
AMSA, WS 07/08 (Masterstudiengang Bioinformatik)

People
Gunnar Rätsch, Cheng Soon Ong, Petra Philips
Contact: amsa07@tuebingen.mpg.de

Course Website
http://www.fml.mpg.de/raetsch/lectures/amsa07

Practical Sessions
4 times X 3 hours: Irregular times and places!
→ see lecture flyer and webpage

Exam
Sequence 100 human genomes in 10 days.
3 billion base pairs per genome.
60GB of data.
Current sequencing efforts

![Bar chart showing the number of completely sequenced genomes from 1995 to 2007. The chart indicates a significant increase in the number of published genomes between 2003 and 2007.](image)
There’s more to come
Protein structures difficult to obtain

Proteins interact with the cell
Genes

- Fragments of the genome sequence coding for one protein [Alberts et al., 2002]
- Estimated no. of human genes: \( \approx 20,000-25,000 \)
- Protein-coding DNA: 2% of the overall genome

But

- One gene can code for several protein products.
- Genes can overlap.
- Genes can code only for RNA. (80% of DNA expressed - see E. Pennisi, Science, June 2007)

Genetic code is complex and far away from being completely understood!
Why machine learning?

- A lot of data
- Data is noisy
- No clear biological theory
- Large number of features
- Complex relationships

Let the data do the talking!
Sequence Analysis

- Annotate whole sequence
- Protein Subcellular Localization
- Motif finding
- Annotate positions
- Gene finding
- Splice form prediction
- Alignment
- EST to DNA
- Remote Homology
- Infer Structure
- RNA secondary structure
- Protein secondary structure
Course Outline

- Machine Learning
- Support Vector Machines and Kernel Methods
- Sequence and Graph Kernels
- Interpreting Classifiers
- Structured Output Learning
- State-of-the-art Applications
Running Example: Splicing

DNA → transcription → pre-mRNA → splicing → mRNA → translation → protein

- DNA contains exons and introns.
- Exons are transcribed into pre-mRNA.
- Introns are spliced out during processing.
- The resulting mRNA is translated into a protein.

Key terms:
- **Cap**: AUG
- **PolyA**: UUG, UAA
- **Start Codon**: AUG
- **Stop Codons**: TTG, TAA, TGA, UUG, UAA
- **Stop Codon**: UGA
Classification of Sequences

Example: Recognition of splice sites

- Every ‘AG’ is a possible acceptor splice site
- Computer has to learn what splice sites look like
  - given some known genes/splice sites . . .
- Prediction on unknown DNA

ATCCCGGATTGGATG
AGGGTCCCTTGAGAGG
CCGGGTATATATATAGG
TTAGGGTCCCTCCGC

1, -1, -1, 1
Many algorithms depend on numerical representations. Each example is a vector of values (features).

Use background knowledge to design good features.

<table>
<thead>
<tr>
<th></th>
<th>x_1</th>
<th>x_2</th>
<th>x_3</th>
<th>x_4</th>
<th>x_5</th>
<th>x_6</th>
<th>x_7</th>
<th>x_8</th>
</tr>
</thead>
<tbody>
<tr>
<td>GC before</td>
<td>0.6</td>
<td>0.2</td>
<td>0.4</td>
<td>0.3</td>
<td>0.2</td>
<td>0.4</td>
<td>0.5</td>
<td>0.5</td>
</tr>
<tr>
<td>GC after</td>
<td>0.7</td>
<td>0.7</td>
<td>0.3</td>
<td>0.6</td>
<td>0.3</td>
<td>0.4</td>
<td>0.7</td>
<td>0.6</td>
</tr>
<tr>
<td>AGAGAAG</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>TTTAG</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Label</td>
<td>+1</td>
<td>+1</td>
<td>+1</td>
<td>-1</td>
<td>-1</td>
<td>+1</td>
<td>-1</td>
<td>-1</td>
</tr>
</tbody>
</table>

exon

intron
Numerical Representation

ATCCCGGATTGGATG
AGGGTCCCCTTGAGAGG
CCGGGTATATATATTAGG
TTAGGTCCCTCCGCGC

1, -1, -1, 1

AT
CG
Recognition of Splice Sites

- **Given**: Potential acceptor splice sites

```
AAACAAATAAGTAACTAATCTTTTAGGAAGAACGTGGCAACCATTTTGAG
AAGATTAAGAAAAAACAAATTTTTAGCATTACAGATATAATAATCTAATT
CACTCCCCAATCAACGATATTTTAGCTTCACTACACACATCCGCTGCTGCC
TTAATTTCACTTCCATACCTCCAGATCATCAATCTCCAAACCAACAC
```

- **intron**
- **exon**

- **Goal**: Rule that distinguishes true from false ones
  - e.g. exploit that exons have higher GC content
  - or
  - that certain motifs are located nearby
Recognition of Splice Sites

Given: Potential acceptor splice sites

Goal: Rule that distinguishes true from false ones

Linear classifiers with large margin
Max-Margin Linear Classifiers

Advantage

- Simple.
- We can compute them.
- Good separation of training data.
The machine utilizes information from training data to predict the outputs associated with a particular test example.

- Use **training** data to “train” the machine.
- Use trained machine to perform prediction on **test** data.
Important not just to memorize the training examples!

How do we know that our estimator will perform well on unseen data?
Assumption
Future examples are similar to seen labeled examples.

Max-Margin Linear Classifiers
Good separation of training data implies provably good separation of new data.

All these are in a probabilistic sense...
Given: Potential acceptor splice sites

Goal: Rule that distinguishes true from false ones

More realistic problem!?  
- Not linearly separable!  
- Need nonlinear separation!?  
- Need more features!?
Nonlinear Separation

- Given: Potential acceptor splice sites

<table>
<thead>
<tr>
<th>intron</th>
<th>exon</th>
</tr>
</thead>
<tbody>
<tr>
<td>AAACAAAAATAGTAACTAATCTTTTAGGAAGAACGTTTCAACCATTTTGGAG</td>
<td>AAGATTAAlAAlAACAAlTTTTTAGCATTACAGATATAATAATCTAATT</td>
</tr>
<tr>
<td>CACTCCCAAAATCAAAGATATTTTAGTTTCACTAAACACATCCGTCTGTCGCC</td>
<td>TTAATTTCACTTCCACATACCTCCAGATCATCAATCTCCAAACACACAC</td>
</tr>
</tbody>
</table>

- Goal: Rule that distinguishes true from false ones

More realistic problem!?  
- Not linearly separable!  
- Need nonlinear separation!?  
- Need more features!?
Nonlinear Algorithms in Feature Space

Idea: Map into a higher dimensional feature space where one can separate linearly.

Example: all second order monomials

\[ \Phi : \mathbb{R}^2 \rightarrow \mathbb{R}^3 \]

\[ (x_1, x_2) \mapsto (z_1, z_2, z_3) := (x_1^2, \sqrt{2} x_1 x_2, x_2^2) \]
Kernel “Trick”

- For special maps $\Phi$ (kernels) and special feature spaces (with scalar products) the scalar products in feature space (here $\mathbb{R}^3$) can be computed directly in the input space (here $\mathbb{R}^2$)!

- This also works for higher orders and dimensions
  $\Rightarrow$ relatively low dimensional input spaces
  $\Rightarrow$ very high dimensional feature spaces

We can still think of linear classifiers!

The art consists in designing the ’right’ kernels!
Advantage

- Simple.
- We can compute them.
- Good separation of **training** data.
- Good separation of **unseen** data.
Consider linear classifiers with parameters $w, b$:

$$f(x) = \sum_{j=1}^{d} w_j x_j + b = \langle w, x \rangle + b$$
Margin maximization is equivalent to minimizing $\|w\|$. 
Minimize

\[
\frac{1}{2} \| \mathbf{w} \|^2 + C \sum_{i=1}^{N} \xi_i
\]

Subject to

\[
y_i(\langle \mathbf{w}, \mathbf{x}_i \rangle + b) \geq 1 - \xi_i \\
\xi_i \geq 0 \\
\text{for all } i = 1, \ldots, N.
\]

The examples on the margin are called support vectors [Vapnik, 1995]

Called the soft margin SVM or the \( C \)-SVM [Cortes and Vapnik, 1995]
Minimize

\[ \frac{1}{2} \| w \|^2 + C \sum_{i=1}^{N} \xi_i \]

Subject to

\[ y_i (\langle w, x_i \rangle + b) \geq 1 - \xi_i \]
\[ \xi_i \geq 0 \]
for all \( i = 1, \ldots, N \).

The examples on the margin are called **support vectors** [Vapnik, 1995]

Called the soft margin SVM or the \( C \)-SVM [Cortes and Vapnik, 1995]
Minimize
\[ \frac{1}{2}||w||^2 + C \sum_{i=1}^{N} \xi_i \]

Subject to
\[ y_i (\langle w, x_i \rangle + b) \geq 1 - \xi_i \]
\[ \xi_i \geq 0 \]
for all \( i = 1, \ldots, N \).

- The examples on the margin are called **support vectors** [Vapnik, 1995]
- Called the soft margin SVM or the \( C \)-SVM [Cortes and Vapnik, 1995]
SVM is dependent on training data

Minimize
\[
\frac{1}{2} \| w \|^2 + C \sum_{i=1}^{N} \xi_i
\]
Subject to
\[
y_i (\langle w, x_i \rangle + b) \geq 1 - \xi_i \\
\xi_i \geq 0 
\]
for all \( i = 1, \ldots, N \).

Representer Theorem
\[ w = \sum_{i=1}^{N} \alpha_i x_i \]

Minimize
\[
\frac{1}{2} \sum_{i,j}^{N} \alpha_i \alpha_j \langle x_i, x_j \rangle + C \sum_{i=1}^{N} \xi_i
\]
Subject to
\[
y_i (\sum_{j=1}^{N} \alpha_j x_j, x_i) + b) \geq 1 - \xi_i \\
\xi_i \geq 0 
\]
for all \( i = 1, \ldots, N \).

SVM solution only depends on scalar products between examples (kernel trick)
SVM is dependent on training data

Minimize

\[ \frac{1}{2} \| w \| ^2 + C \sum_{i=1}^{N} \xi_i \]

Subject to

\[ y_i (\langle w, x_i \rangle + b) \geq 1 - \xi_i \]
\[ \xi_i \geq 0 \]
for all \( i = 1, \ldots, N \).

Minimize

\[ \frac{1}{2} \sum_{i,j}^{N} \alpha_i \alpha_j \langle x_i, x_j \rangle + C \sum_{i=1}^{N} \xi_i \]

Subject to

\[ y_i (\sum_{j=1}^{N} \alpha_j x_j + b) \geq 1 - \xi_i \]
\[ \xi_i \geq 0 \]
for all \( i = 1, \ldots, N \).

Representer Theorem

\[ w = \sum_{i=1}^{N} \alpha_i x_i \]

SVM solution only depends on scalar products between examples (kernel trick)
SVM is dependent on training data

Minimize

\[ \frac{1}{2} \| \mathbf{w} \|^2 + C \sum_{i=1}^{N} \xi_i \]

Subject to

\[ y_i (\langle \mathbf{w}, \mathbf{x}_i \rangle + b) \geq 1 - \xi_i \]
\[ \xi_i \geq 0 \]
for all \( i = 1, \ldots, N \).

Representer Theorem

\[ \mathbf{w} = \sum_{i=1}^{N} \alpha_i \mathbf{x}_i \]

Minimize

\[ \frac{1}{2} \sum_{i,j}^{N} \alpha_i \alpha_j \langle \mathbf{x}_i, \mathbf{x}_j \rangle + C \sum_{i=1}^{N} \xi_i \]

Subject to

\[ y_i (\sum_{j=1}^{N} \alpha_j \mathbf{x}_j, \mathbf{x}_i) + b) \geq 1 - \xi_i \]
\[ \xi_i \geq 0 \]
for all \( i = 1, \ldots, N \).

SVM solution only depends on scalar products between examples (\( \sim \) kernel trick)
Summary: Prediction

Duda et al. [2001], Schölkopf and Smola [2002], Shawe-Taylor and Cristianini [2004]
Useful to Review

Basics of

- Linear algebra: dot product, metric (euclidean distance)
- Optimization: gradient descent, convex functions
- Probability: joint, conditional, marginal probability
At the end of the semester

you will understand

- what an SVM and kernel methods are
- how they work
- why they work
- how to employ them for sequence analyses
- how they compare to classical sequence analysis algorithms
- and what cutting edge applications they are used for
**Outlook**

**mGene**: An ab initio gene finding system

![Diagram of gene structure with TATAAA, ATG, GT, AG, GT, AG, and TAA regions, and a structured SVM at the center connected to TSS, TIS, DON, ACC, TSTOP, POLYA, and CLEAV nodes with specific DNA sequences like AATCAAC GTTGCCACGAATACGGTACGCGTGTACGACGAGTATCGTCCATTAACCAAACTCTTGCATTTCTGAGAACAAATAA.](image)
Sensitivity and specificity on coding nucleotide, exon and transcript level for \textit{ab initio} gene finding.
References


