Advanced Methods for Sequence Analysis

Cheng Soon Ong[1,2]

[1] Friedrich Miescher Laboratory, Tübingen
[2] Max Planck Institute for Biological Cybernetics, Tübingen

Vorlesung WS 2006/2007
Eberhard Karls Universität Tübingen

18 October 2006

http://www.fml.mpg.de/raetsch/lectures/amsa

Official Information

Advanced Methods for Sequence Analysis, WS 06/07 (Masterstudiengang Bioinformatik)

People

Gunnar Rätsch[1], Cheng Soon Ong[1,2] and Petra Philips[1]
[1] Friedrich Miescher Laboratory, Tübingen
[2] MPI for Biological Cybernetics, Tübingen

Contact: amsa06@tuebingen.mpg.de

Course Website

http://www.fml.mpg.de/raetsch/lectures/amsa

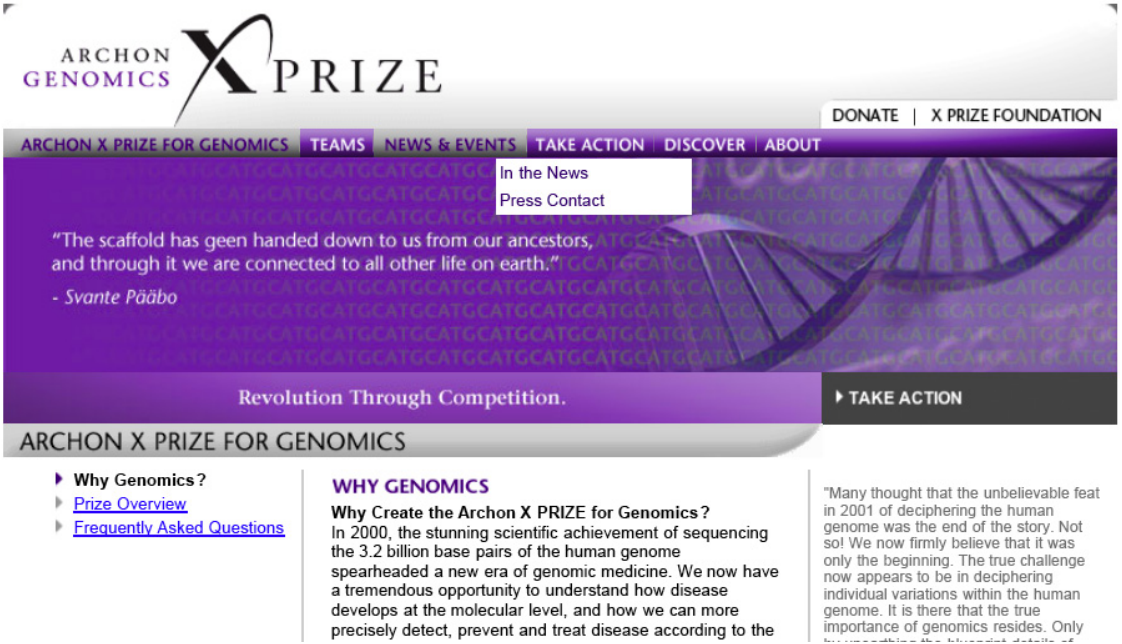
Exam

Oral examination in mid Feb 2007, register by 31 Jan 2007.

Why bioinformatics?

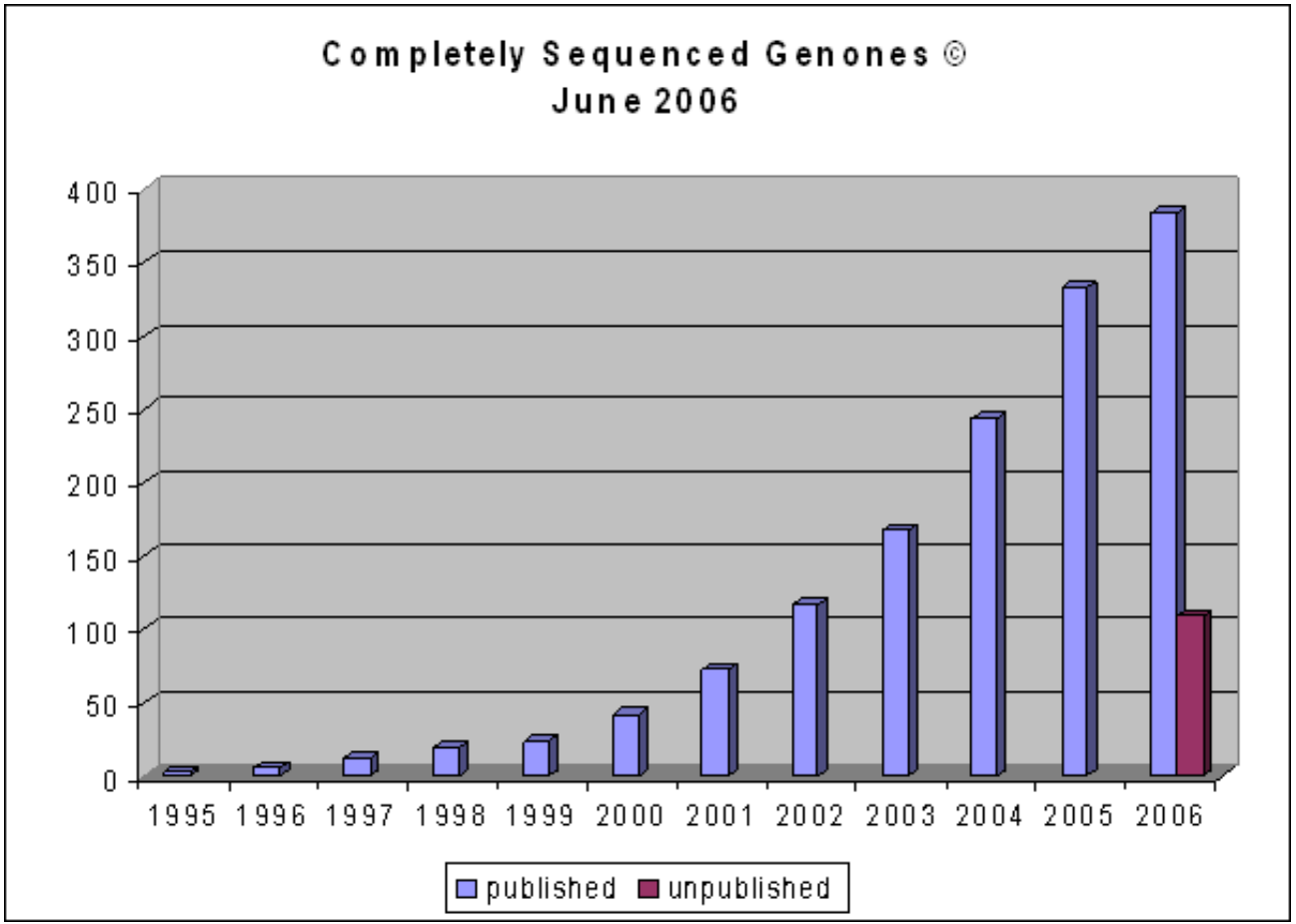


USD$10 Million

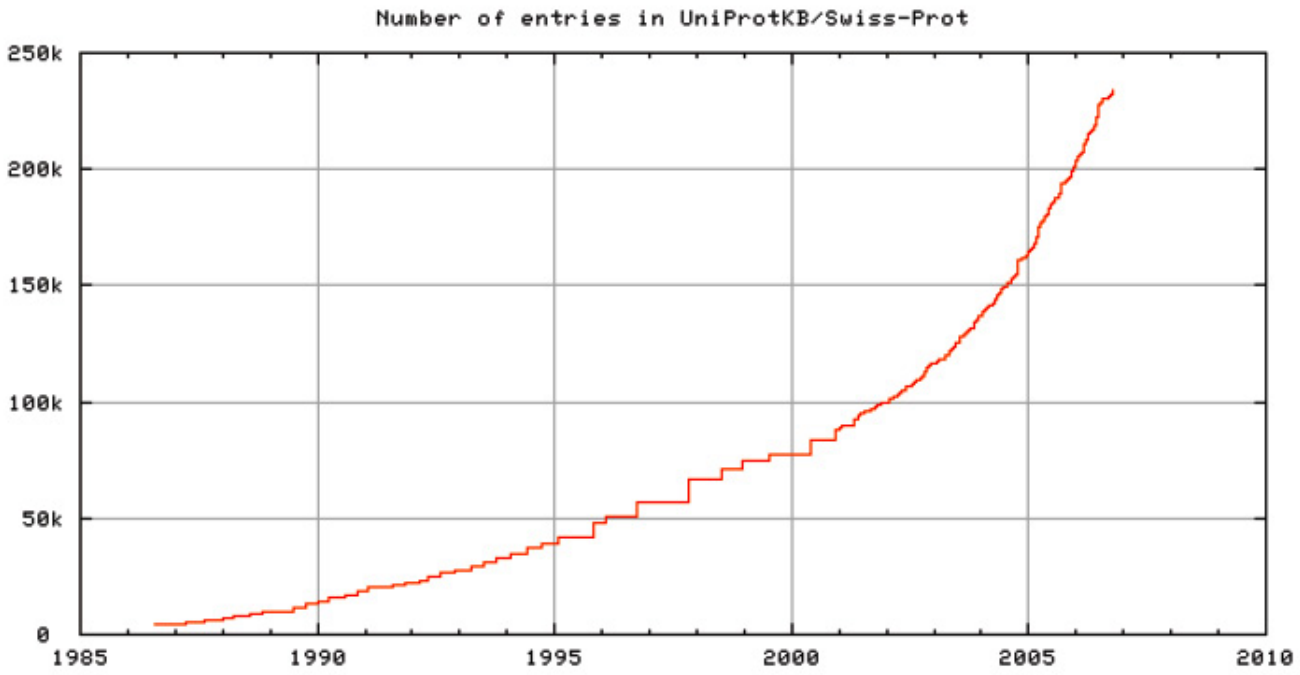


- Sequence 100 human genomes in 10 days.
- 3 billion base pairs per genome.
- 60GB of data.

Current sequencing efforts



Protein Sequences



- Protein structures difficult to obtain
- Proteins interact with the cell

Why machine learning?



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

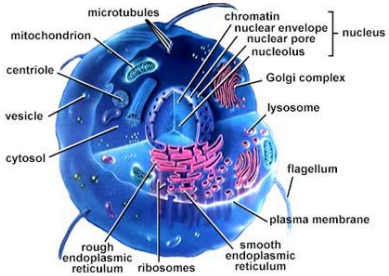
Let the data do the talking!

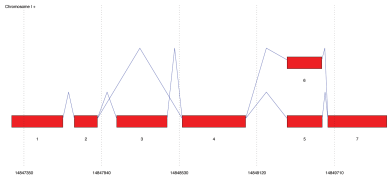
Course Outline

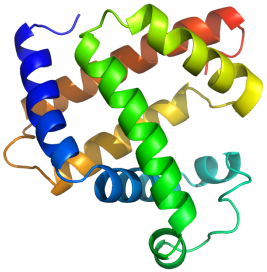
- Machine Learning
- Support Vector Machine
- Kernels
- Sequence and Graph Kernels
- Interpreting Classifiers
- Structured Output Learning

Sequence Analysis

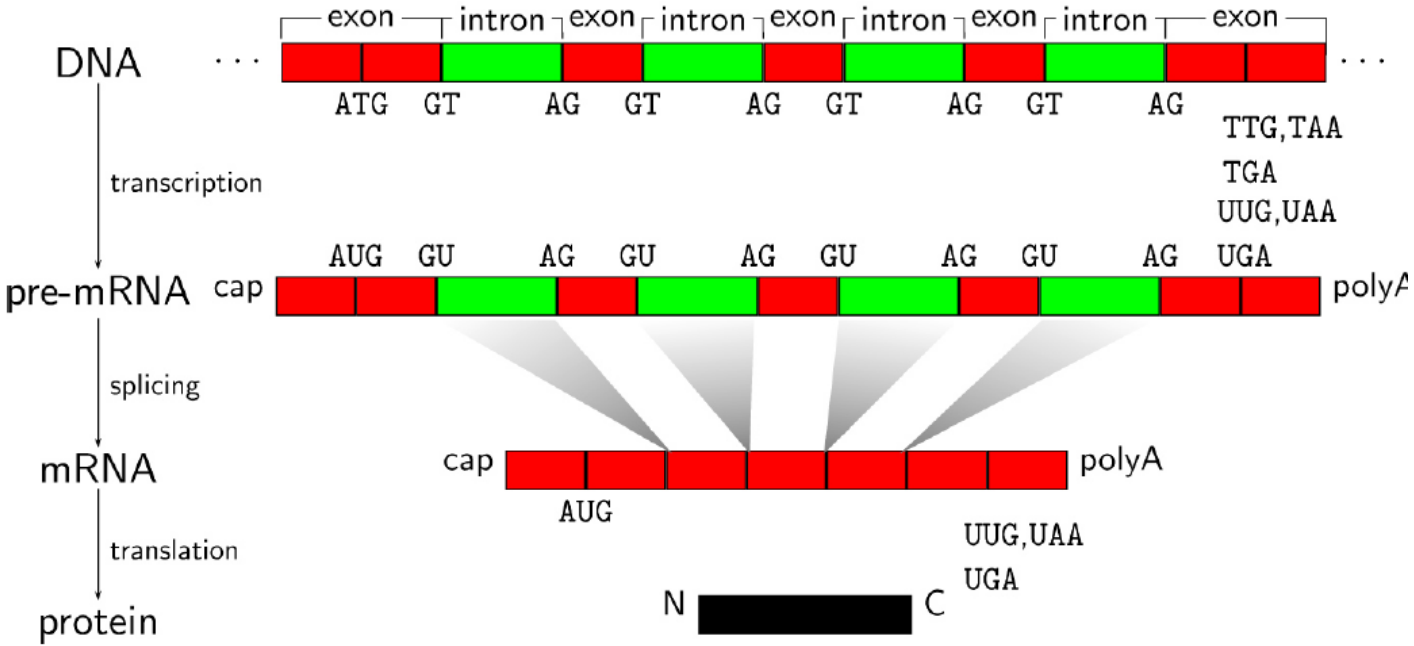
- Annotate whole sequence
  - Protein Subcellular Localization
  - Motif finding
- Annotate positions
  - Gene finding
  - Splice form prediction
- Alignment
  - EST to DNA
  - Remote Homology
- Infer Structure
  - RNA secondary structure
  - Protein secondary structure







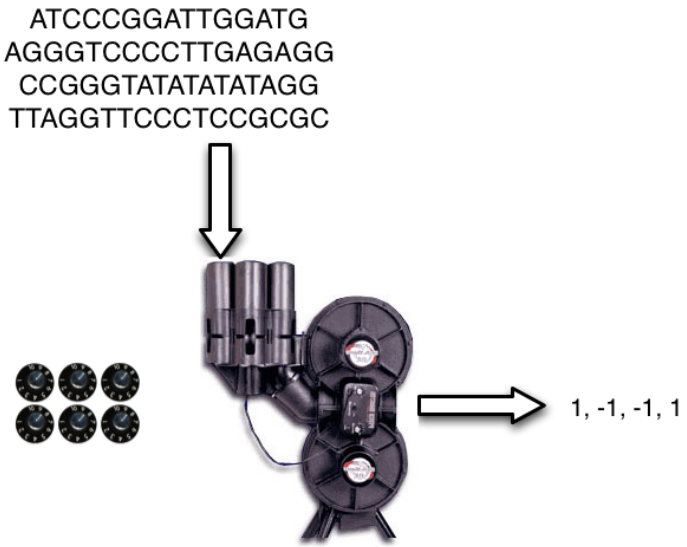
Running Example: Splicing



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



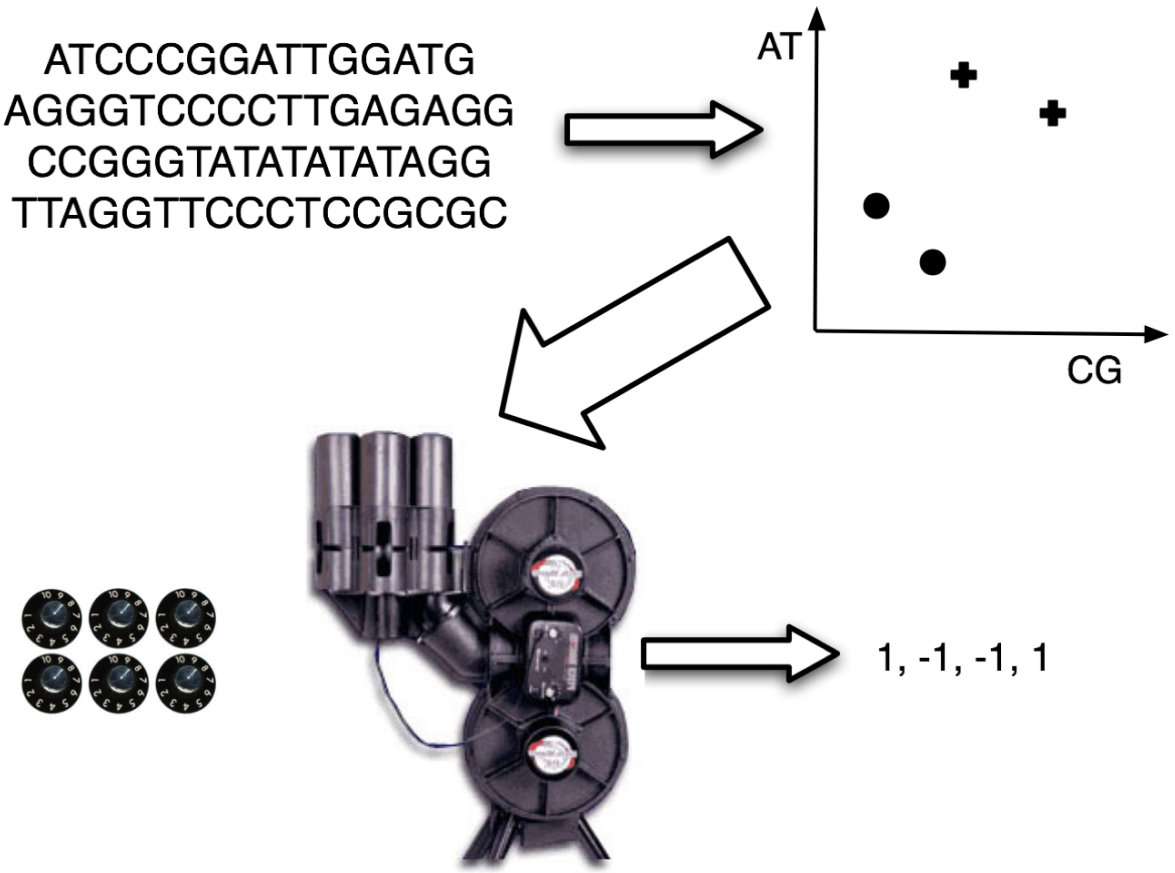
From Sequences to Features

- Many algorithms depend on numerical representations.
  - Each example is a vector of values (features).
- Use background knowledge to design good features.



	$\mathbf{x}_1$	$\mathbf{x}_2$	$\mathbf{x}_3$	$\mathbf{x}_4$	$\mathbf{x}_5$	$\mathbf{x}_6$	$\mathbf{x}_7$	$\mathbf{x}_8$	...
GC before	0.6	0.2	0.4	0.3	0.2	0.4	0.5	0.5	...
GC after	0.7	0.7	0.3	0.6	0.3	0.4	0.7	0.6	...
AG**AG**AAG	0	0	0	1	1	0	0	1	...
TTT**AG**	1	1	1	0	0	1	0	0	...
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋱
Label	+1	+1	+1	−1	−1	+1	−1	−1	...

Numerical Representation



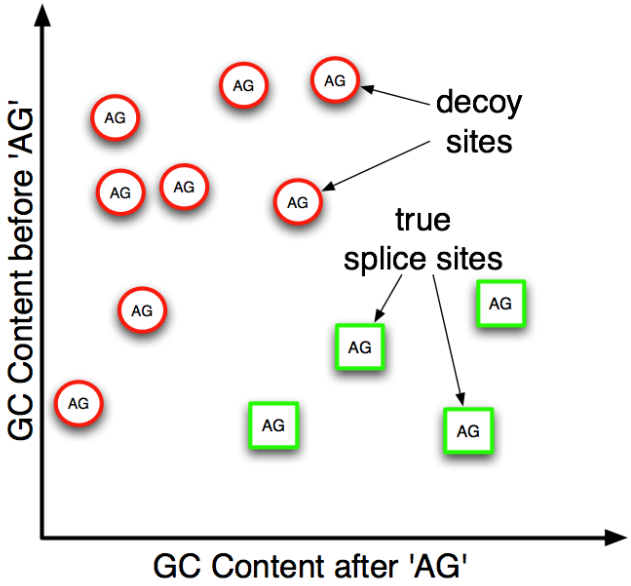
Recognition of Splice Sites

- Given: Potential acceptor splice sites

```
AAACAAATAAGTAACTAATCTTTTAGGAAGAACGTTTCAACCATTTTGAG
AAGATTAAAAAAAAACAAATTTTTAGCATTACAGATATAATAATCTAATT
CACTCCCCAAATCAACGATATTTTAGTTCACTAACACATCCGTCTGTGCC
TTAATTTCACTTCCACATACTTCCAGATCATCAATCTCCAAAACCAACAC
```

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

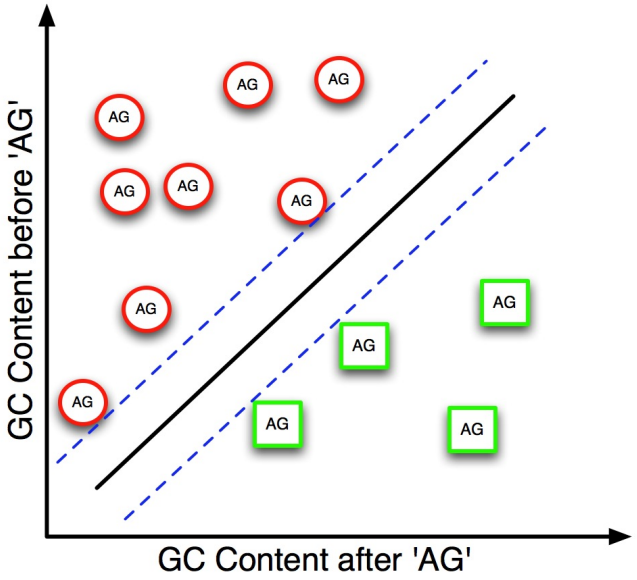
Recognition of Splice Sites

- Given: Potential acceptor splice sites

```
AAACAAATAAGTAACTAATCTTTTAGGAAGAACGTTTCAACCATTTTGAG
AAGATTAAAAAAAAACAAATTTTTAGCATTACAGATATAATAATCTAATT
CACTCCCCAAATCAACGATATTTTAGTTCACTAACACATCCGTCTGTGCC
TTAATTTCACTTCCACATACTTCCAGATCATCAATCTCCAAAACCAACAC
```

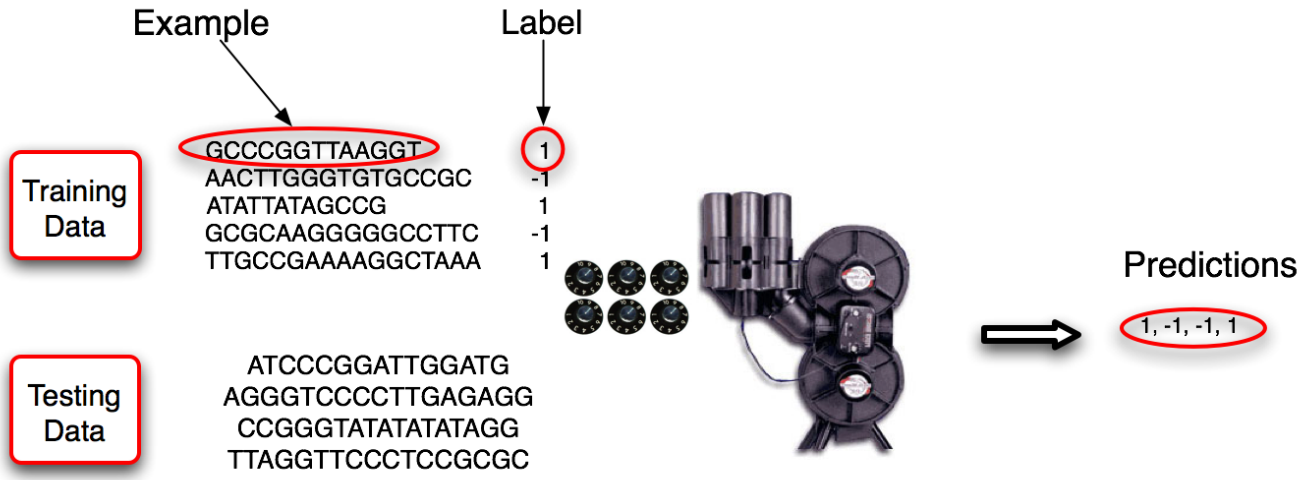
intron exon

- Goal: Rule that distinguishes true from false ones



Linear classifiers with large **margin**

Empirical Inference



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.

Words, words, words...

example $\mathbf{x}_i \in \mathcal{X}$ for example a set of amino acid sequences.

label $y_i \in \mathcal{Y}$ for example whether the sequence is an alpha helix, beta sheet of coil.

Training Data Data consisting of examples and associated labels, which are used for training the machine.

Testing Data Data consisting only of examples, used for generating predictions.

Predictions Output of the trained machine.

Machine Learning: Main Tasks

Supervised Learning

We have both examples and labels for each example. The aim is to learn about the pattern between examples and labels.

Unsupervised Learning

We do not have labels for the examples, and wish to discover the underlying structure of the data.

Reinforcement Learning

How an autonomous agent that senses and acts in its environment can learn to choose optimal actions to achieve its goals.

Estimators

Basic Notion

We want to estimate the relationship between the examples $\mathbf{x}_i$ and the associated label y_i.

Formally

We want to choose an estimator

$$f : \mathcal{X} \to \mathcal{Y}.$$

Intuition

We would like a function f which correctly predicts the label y for a given example $\mathbf{x}$.

Question

How do we measure how well we are doing?

Loss Function

Basic Notion

We characterize the quality of an estimator by a loss function.

Formally

We define a loss function

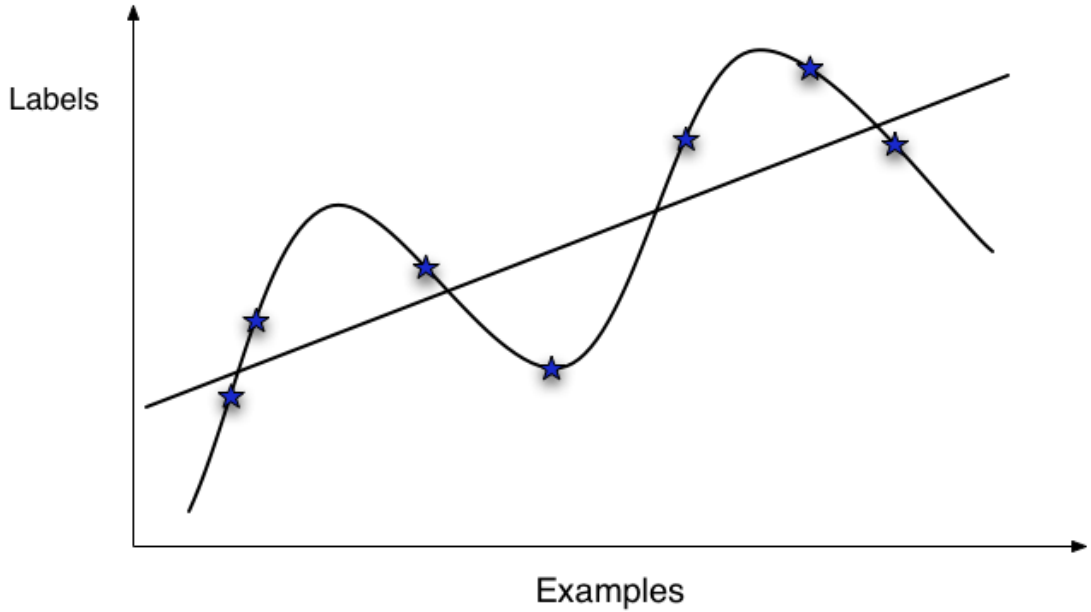
$$\ell(f(\mathbf{x}_i), y_i) : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}_+.$$

Intuition

For a given label y_i and a given prediction $f(\mathbf{x}_i)$, we want a positive value telling us how much error we have made. For example, in the case of classification, we can have the loss:

$$\ell(f(\mathbf{x}_i), y_i) = \begin{cases} 0 & \text{if } f(\mathbf{x}_i) = y_i \\ 1 & \text{if } f(\mathbf{x}_i) \neq y_i \end{cases}$$

Regression



In the case or regression ($\mathcal{Y} = \mathbb{R}$), the loss function is usually the squared distance between the prediction ($f(\mathbf{x}_i)$) and the label (y_i), that is $\ell(f(\mathbf{x}_i), y_i) = (f(\mathbf{x}_i) - y_i)^2$.

Risk

Basic Notion

We want to measure the average loss over all data points. This is called risk.

Formally

For a given function f, we define the risk to be

$$\mathbf{R} = \int \ell(f(\mathbf{x}), y) d\rho.$$

Question

How is the averaging supposed to be done?

Main Approaches

There is a conceptual difference between frequentist and Bayesian statistics.

Discriminative

Sometimes called frequentist statistics.

- We are interested in finding the best predictor for a particular task.
- Useful when there is a lot of data and not much knowledge about the problem.

Probabilistic

Sometimes called Bayesian methods

- We are interested to find the model that best explains the observed data.
- Useful when we have prior knowledge to design good models.

But we only have some data!

We want to average over all data sets, but we cannot.

Empirical Risk

However, we can find the average on the training data. That is we define the empirical risk to be:

$$\mathbf{R}_{emp}(f, X, Y) = \frac{1}{n} \sum_{i=1}^{n} \ell(f(\mathbf{x}_i), y_i).$$

Basic Notion

Instead of minimizing the risk, we minimize over the empirical risk (the average loss on the training data). This is called the principle of empirical risk minimization.

Question

How do we know that our estimator will perform well on unseen data?

Words, words, words...

Loss $\ell(f(\mathbf{x}_i), y_i)$, How much an error costs us.

Risk $\mathbf{R}(f)$, The expected loss for all data, including the yet unseen.

Empirical Risk $\mathbf{R}_{emp(f)}$. The average loss that we can calculate.

How to measure performance?

Important not just to memorize the training examples!
Use some of the labeled examples for validation.

	$\mathbf{x}_1$	$\mathbf{x}_2$	$\mathbf{x}_3$	$\mathbf{x}_4$	$\mathbf{x}_5$	$\mathbf{x}_6$	$\mathbf{x}_7$	$\mathbf{x}_8$	...
GC before	0.6	0.2	0.4	0.3	0.2	0.4	0.5	0.5	...
GC after	0.7	0.7	0.3	0.6	0.3	0.4	0.7	0.6	...
AG**AG**AAG	0	0	0	1	1	0	0	1	...
TTT**AG**	1	1	1	0	0	1	0	0	...
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋱
Label	+1	+1	+1	−1	−1	+1	−1	−1	...
	Training						Testing		

We assume that the future examples are similar to our labeled examples.

Measuring performance

What to do in practice

We split the data into training and validation sets, and use the error on the validation set to estimate the expected error.

A. Cross validation

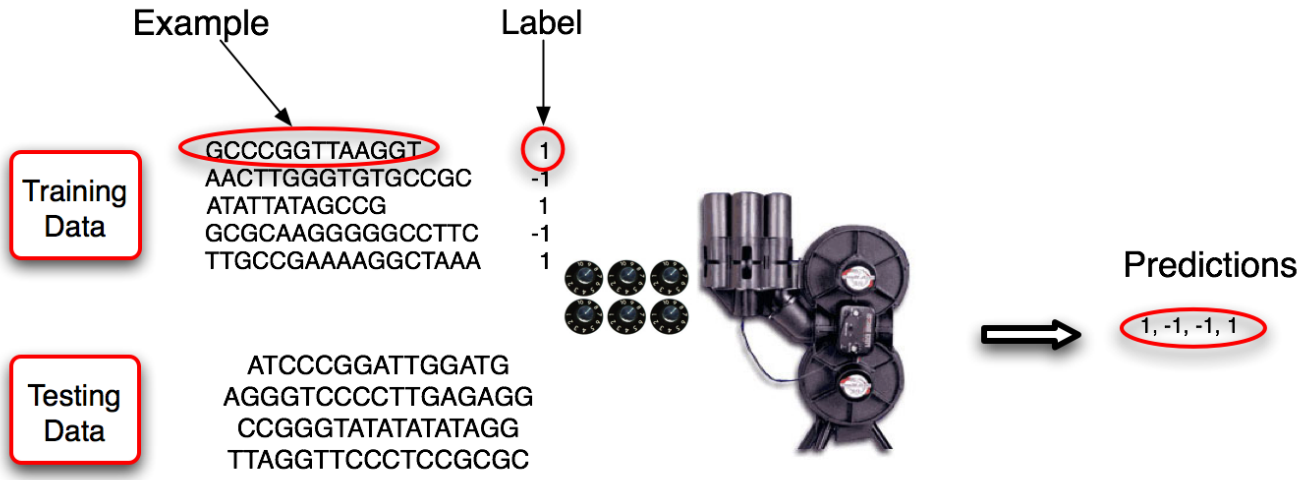
Split data into c disjoint parts, and use each subset as the validation set, while using the rest as the training set.

B. Random splits

Randomly split the data set into two parts, for example 80% of the data for training and 20% for validation. This is usually repeated many times.

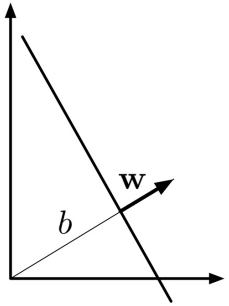
Report mean and standard deviation of performance on the validation set.

Classifier: depends on training data

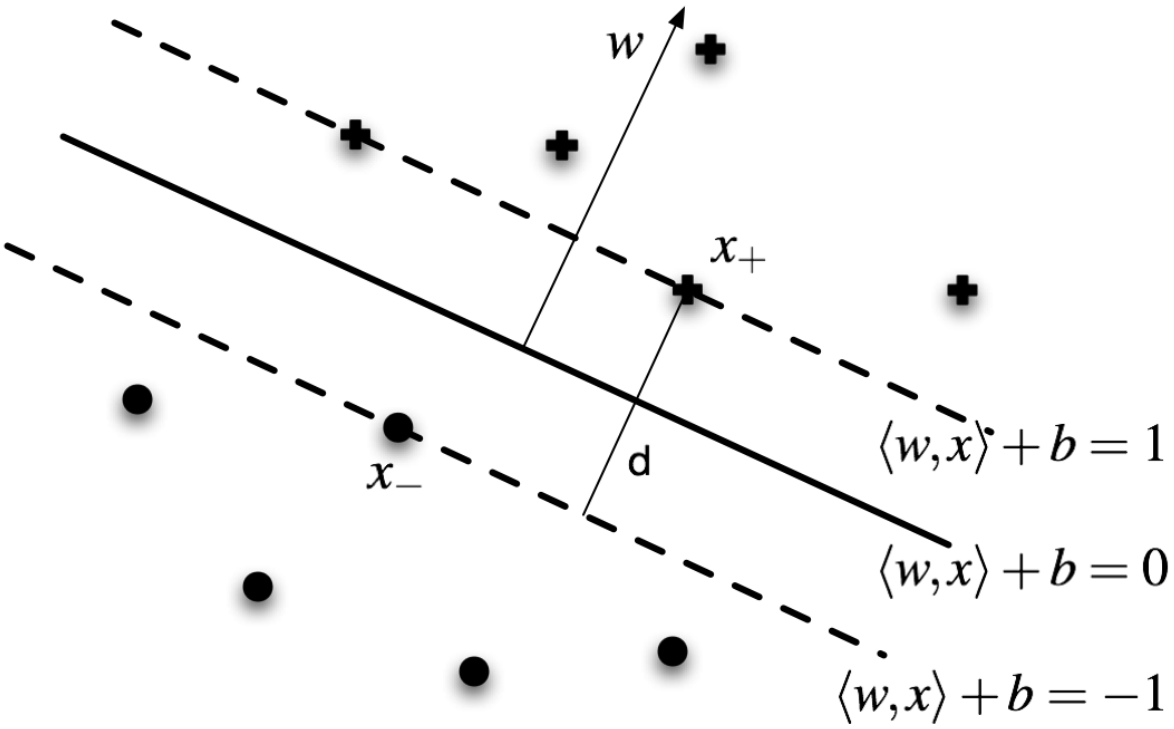


Consider linear classifiers with parameters $\mathbf{w}, b$:

$$f(\mathbf{x}) = \sum_{j=1}^{d} w_j x_j + b = \langle \mathbf{w}, \mathbf{x} \rangle + b$$

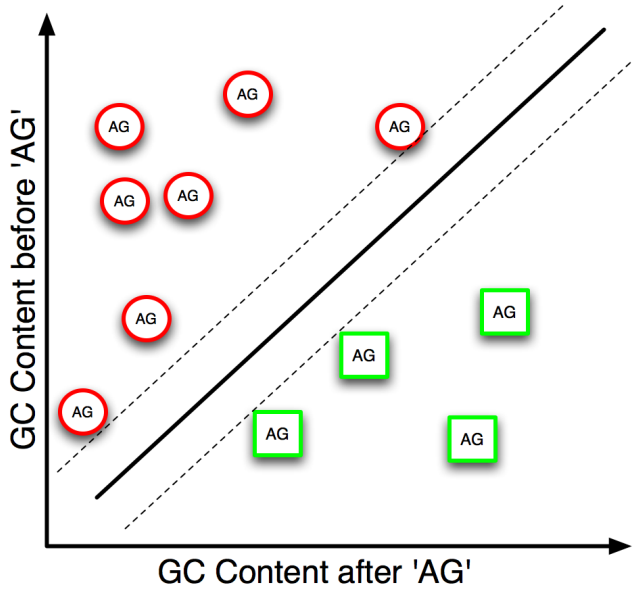


Margin Maximization



Margin maximization is equivalent to minimizing $\|\mathbf{w}\|$.

Classifier: SVM



Minimize

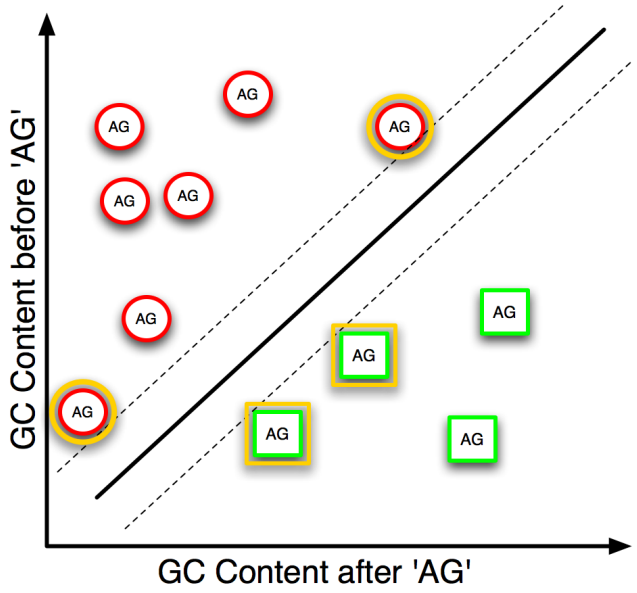
$$\frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{N}\xi_i$$

Subject to

$$y_i(\langle \mathbf{w}, \mathbf{x}_i \rangle + b) \geqslant 1 - \xi_i$$
$$\xi_i \geqslant 0$$
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]

Classifier: SVM



Minimize

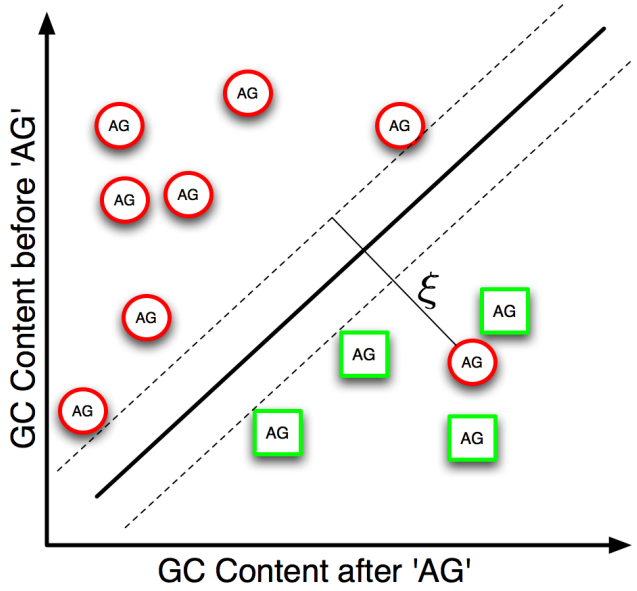
$$\frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{N}\xi_i$$

Subject to

$$y_i(\langle \mathbf{w}, \mathbf{x}_i \rangle + b) \geqslant 1 - \xi_i$$
$$\xi_i \geqslant 0$$
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]

Classifier: SVM



Minimize

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

Subject to

$$y_i(\langle \mathbf{w}, \mathbf{x}_i\rangle + b) \geqslant 1 - \xi_i$$
$$\xi_i \geqslant 0$$
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

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

Subject to

$$y_i(\langle \mathbf{w}, \mathbf{x}_i\rangle + b) \geqslant 1 - \xi_i$$
$$\xi_i \geqslant 0$$
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

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

SVM solution only depends on scalar products between examples ($\rightsquigarrow$ kernel trick)

SVM is dependent on training data

Minimize

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

Subject to

$$y_i(\langle \mathbf{w}, \mathbf{x}_i\rangle + b) \geqslant 1 - \xi_i$$
$$\xi_i \geqslant 0$$
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

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

SVM solution only depends on scalar products between examples ($\rightsquigarrow$ kernel trick)

SVM is dependent on training data

Minimize

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

Subject to

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

Representer Theorem

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

Minimize

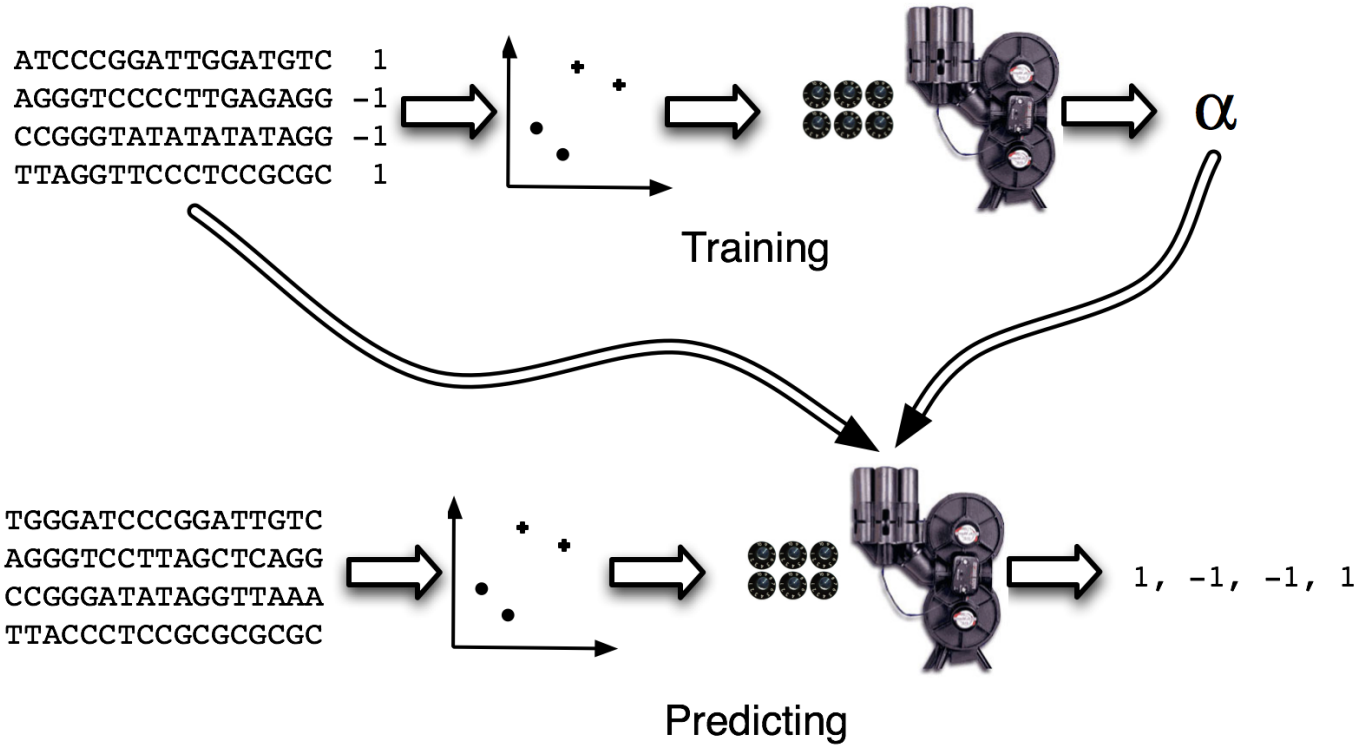
$$\frac{1}{2}\sum_{i,j}^{N}\alpha_i\alpha_j\langle \mathbf{x}_i, \mathbf{x}_j\rangle + C\sum_{i=1}^{N}\xi_i$$

Subject to

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

SVM solution only depends on scalar products between examples ($\rightsquigarrow$ kernel trick)

Summary: Empirical Inference



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

References

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

R.O. Duda, P.E.Hart, and D.G.Stork. *Pattern classification*. John Wiley & Sons, second edition, 2001.

B. Schölkopf and A.J. Smola. *Learning with Kernels*. MIT Press, Cambridge, MA, 2002.

John Shawe-Taylor and Nello Cristianini. *Kernel Methods for Pattern Analysis*. Cambridge University Press, 2004.

V.N. Vapnik. *The nature of statistical learning theory*. Springer Verlag, New York, 1995.