10-810: Computational Molecular Biology: a machine learning approach

Classification and time series analysis
Types of classifiers

- We can divide the large variety of classification approaches into roughly two main types

1. Generative:
   - build a generative statistical model
   - e.g., mixture model

2. Discriminative
   - directly estimate a decision rule/boundary
   - e.g., logistic regression
Why discriminative?

• One reason for using discriminative approaches is robustness:

• For generative models we have to pick the model and determine the parameters (class conditional models, equal covariances etc.)

• If we estimate a linear decision boundary directly we are less dependent on what the true class conditional distributions are

• Examples of discriminative classifiers
  - Linear discriminant analysis
  - Logistic regression (generalized linear models, generalized additive models)
  - Support vector machines
Linear decision boundary

- Simple example: linear regression.

\[
f(x, A) = a_0 + a_1x_1 + \ldots + a_nx_n
\]

\[
f(x, A) > 0 \Rightarrow \text{class} = 1
\]

\[
f(x, A) \leq 0 \Rightarrow \text{class} = 0
\]

- The A vector contains the parameters and X is a vector of expression levels.
- Similarly to the generative model case, we have to solve the
  - estimation problem
  - variable selection problem
Support vector machines

- Optimal hyperplane
- Finding the optimal hyperplane
- Kernel function
- Complexity
Optimal hyperplane

- Let's assume for simplicity that the classification problem is linearly separable

- Maximum margin hyperplane is a hyperplane maximally removed from all the training examples

- This hyperplane can be defined on the basis of only a few training examples called support vectors
Optimal hyperplane (cont.)

- We are estimating a linear classifier:
  \[ f(x, A) = a_0 + a_1 x_1 + \ldots + a_n x_n = a_0 + A^T x \]

- We can try to find the "optimal" hyperplane by requiring that the sign of the decision boundary (clearly) agrees with all the training labels
  \[ y^t (a_0 + A^T x) > 1, t = 1 \ldots n \]

- where the labels y are +1 or -1.
SVM and linear regression

BUT...

- this is actually an alternative definition of linear separability
- there are multiple answers: larger values of \( a_0, A \) would yield larger separation.
SVM

• We find the smallest parameter values that still satisfy the classification constraints
• We find \( \min \| A \|^2 = \sum a_i^2 \)
• subject to the classification constraints \( y^t(a_0 + A^T x) > 1, t = 1...n \)
• Only a few of the classification constraints are relevant: the R support vectors

\[ \sum a_i \geq 1 \]

\[ a_i = 0 \rightarrow x_i \text{ is not an R support vector } \]

\[ a_i = 1 \rightarrow x_i \text{ is an R support vector } \]
Determining the parameters

• We can use Lagrange multipliers to arrive at the following minimization problem:

\[ J(\alpha) = \sum_i \alpha_i + \sum_i \alpha_i y_i (Ax + a_0) - \frac{1}{2} \| A \|^2 \]
Dual formation

• This minimization leads to the following dual formation:

\[ L_D = \sum_i \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j x_i x_j \]

• subject to the constraints

\[ \alpha_i \geq 0, \forall i \]

\[ \sum_i \alpha_i y_i = 0 \]

• For non-separable problems we simply limit for some positive constant C

\[ \alpha_i \leq C \]

• This is leads to a quadratic programming problem
Interpretation of SVMs

- Before:
  - example vectors $x^t$ of dimension $m$ (the number of genes)
  - parameters $a_0…a_n$ which multiply each component of $x$ (genes)

- After:
  - real valued inner products $x^t x^t$ measuring how similar the training examples are
  - weights $\alpha_i$ on the examples indicating how important each training example is to the classification task
Using SVMs

- To use support vector machines we need to:
  - specify similarities between the examples (i.e., \( x^t x^{t'} \))
  - set the example weights \( \alpha_i \) by enforcing the classification constraints.

- We make decisions by comparing each new sample \( x \) with only the \( k \) support vectors

\[
\hat{y} = \text{sign}(Wx + a_0)
\]

\[
W = \sum_i \alpha_i y_i x_i
\]
Non linear classifier

- So far the SVM classifier is able to separate our sample populations only linearly.

- We can easily obtain a non-linear classifier by mapping our samples $x = [x_1; x_2]$ into longer feature vectors:

$$\Theta(x) = [x_1^2, x_2^2, \sqrt{x_1 x_2}, 1]$$

and applying the linear classifier to $\Theta$ instead.

- This way we can for example take into account dependencies among the genes to better classify tissue samples.
Example

Linear separator in the feature space

Non linear separator
Examples

Linear

2$^{\text{nd}}$ order polynomial
$4^{\text{th}}$ order polynomial

$8^{\text{th}}$ order polynomial
Golub data

- Golub et al. leukemia classification problem
- 7130 ORFs
- 38 labeled training examples,
- 34 test examples
- Let's blindly apply SVMs to this problem using polynomial kernels of degree $p=1,2,4,8$. 
- We get 1 test error for all classifiers regardless of their complexity
- There is only a slight overfitting...
Golub-Results

- The figure shows the discriminant function values for the test samples resulting from polynomial kernels of degree $p=1,2,4,8$. 

![Graph showing discriminant function values for test samples with polynomial kernels of different degrees.](image)
Time series expression
Expression Experiments

Static: Snapshot of the activity in the cell

Time series: Multiple arrays at various temporal intervals
Time Series Examples: Development

Development of fruit flies [Arbeitman, Science 02]
Time Series Examples: Systems

The cell cycle system in yeast [Simon et al, Cell 01]
Distribution of Number of Time Points in Time Series Data Available in the Stanford Microarray Database
Unique features of time series expression experiments

- Autocorrelation between successive points.
- Can identify complete set of acting genes.
- Allows to infer causality.
Time Series Expression Analysis

- **Networks**
  - Computational: information fusion
  - Biological: dynamic regulatory networks

- **Pattern Recognition**
  - Computational: clustering, classification
  - Biological: function, response programs

- **Individual Gene**
  - Computational: normalization, miss. values, interpolation
  - Biological: alignment, diff. expressed genes

- **Experimental Design**
  - Computational: sampling rates, duration
  - Biological: transcription, decay rates
Networks

Pattern Recognition

Individual Gene

Experimental Design
Sampling Rates

- Non uniform
- Differ between experiments
Issues to address

• Continuous representation
• Alignment
• Identifying differentially expressed genes
• Synchronization
# Yeast Cell Cycle Datasets

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Method of arrest</th>
<th>Duration</th>
<th>Cell cycle length</th>
<th>Sampling</th>
<th>Repeats</th>
</tr>
</thead>
<tbody>
<tr>
<td>alpha (Spellman 98)</td>
<td>alpha mating factor</td>
<td>0-119m</td>
<td>64m</td>
<td>every 7 minutes</td>
<td>1</td>
</tr>
<tr>
<td>cdc15 (Spellman 98)</td>
<td>temp. sensitive cdc15</td>
<td>10-290m</td>
<td>112m</td>
<td>ev. 20m for 1 hr, ev. 10m for 3 hr, ev. 20m for final hr</td>
<td>1</td>
</tr>
<tr>
<td>cdc28 (Cho98)</td>
<td>temp. sensitive cdc28</td>
<td>0-160m</td>
<td>85m</td>
<td>every 10 minutes</td>
<td>1</td>
</tr>
<tr>
<td>fkh1/fkh2 knockout (Zhu00)</td>
<td>alpha mating factor</td>
<td>0-215m</td>
<td>105m</td>
<td>every 15m until 165m then after 45m</td>
<td>2</td>
</tr>
<tr>
<td>yox1/yhp1 knockout (Pramila02)</td>
<td>alpha mating factor</td>
<td>0-120m</td>
<td>60m</td>
<td>every 10 minutes</td>
<td>1</td>
</tr>
</tbody>
</table>
Representing time series expression data

- We are capturing a continuous process with a few samples.
- We need a way to convert our samples for each gene to an expression profile.
- Some simple techniques:
  - Linear interpolation
  - Spline interpolation
  - Functional assignment
Standard interpolation

If we have missing values and noise linear interpolation will fail to reproduce an accurate representation.
Cubic Splines

• Piecewise cubic polynomials satisfying continuity and smoothness constraints.

• B-splines represents the splines as a linear combination of basis functions, where the coefficients are the spline control points.

\[ Y_i(t) = S(t)F \]

• When faced with noise and missing values, splines overfit the data.

Many of the genes are co-expressed. Thus, we use classes of similarly expressed genes to constrain spline assignment, and overcome noise and missing data.
Continuous representation: The power of co-expression

Many of the genes are co-expressed, we can use co-expressed genes to overcome noise in individual gene

Q: How can we identify the set of co-expressed genes?
A: Clustering

Q: How do we use the cluster genes?
A: Instead of average representation extract shape information (co-variance matrix)

Q: Covariance matrix is very big, what about overfitting?
A: Use dimensionality reduction methods (splines)

A mixed effects model

Class covariance matrix

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>3</th>
<th>5</th>
<th>8</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>.3</td>
<td>.5</td>
<td>-.5</td>
<td>.5</td>
<td>-.5</td>
</tr>
<tr>
<td>3</td>
<td>.5</td>
<td>.3</td>
<td>-1</td>
<td>1</td>
<td>-1</td>
</tr>
<tr>
<td>5</td>
<td>-.5</td>
<td>-1</td>
<td>.3</td>
<td>-1</td>
<td>1</td>
</tr>
<tr>
<td>8</td>
<td>.5</td>
<td>1</td>
<td>-1</td>
<td>.3</td>
<td>-1</td>
</tr>
<tr>
<td>10</td>
<td>-.5</td>
<td>-1</td>
<td>1</td>
<td>-1</td>
<td>.3</td>
</tr>
</tbody>
</table>
How Good is the Learned Model?

• Chose randomly 100 genes that do not have any missing values.

• For each one of them we hide 1, 2, 3 and 4 consecutive values, and compute the prediction error of each of the different algorithms.
Comparison of Missing Values Estimation Techniques

- Linear interpolation
- Splines for individual genes
- KNN
- TimeFit w/ Spellman classes
- TimeFit w/ own classes

Error vs. Num of missing values graph.
Comparing Interpolation Methods

Holding out time points and using each method to predict missing data
Issues to address

- Continuous representation
- Alignment
- Identifying differentially expressed genes
- Synchronization
Alignment

- Difference in the timing of similar biological processes
Alignment Solution (1)

• Use a method similar to sequence alignment.

• Let the two time serieses to be aligned be \( a \) and \( b \). Series \( a \) has time points 0, 1, ..., \( n \) at times \( t_0 < t_1 < ... < t_n \), and series \( b \) has time points 0, 1, ..., \( m \) at times \( u_0, < u_1 < ... < u_m \).

We would like to minimize the following sum:

\[
D (a, b) = \sum_{h=1}^{q} d (a _{i(h)}, b _{i(h)})
\]

where \( d \) is the Euclidian distance between the two points

Aach and Church Bioinformatics 2001
Aligning cdc15 and alpha
Cdc15 and alpha – Interpolated Version
TimeWrap - Drawbacks

• Many degrees of freedom.

• Time can ‘stop’ or ‘go backward’

• Alignment score not statistically significant (when compared with random permutations of genes names).
Continuous Alignment

- Using the estimated splines, we can minimize a global error function, and allow for arbitrary start/end points.
Continuous Alignment

• Using the estimated splines, we can minimize a global error function, and allow for arbitrary start/end points.

• Look for two parameters (stretch and translate) by minimizing the area between the two gene expression curves.

$$e_i^2 = \frac{\int_{\alpha}^{\beta} (g_1(s) - g_2(T(s)))^2 \, ds}{\beta - \alpha}$$
Example: S phase cluster

\[ \text{error} = 45.460 \]

avg. expression

![Graph showing the expression of cdc15 and cdc28 genes during S phase.](image-url)
Continuous Alignment

Using the estimated splines, we continuously align two expression datasets by minimizing a global error function.
Issues to address

• Continuous representation
• Alignment
• Identifying differentially expressed genes
• Synchronization
Identifying Differentially Expressed Genes

Problems:  - Not enough repeats
            - Different sampling rates at different segments
            - Value dependent variance
Judging the Significance of the Difference

Hypothesis testing:

H₀: The test curve is a noisy realization of the reference curve.

H₁: The two curves are independent (different)

Problem: Due to lack of repeats and to value dependent noise it is hard to compute a good noise model for these curves.
Identifying Differentially Expressed Genes

Combine individual noise model with global error measurement that captures the temporal difference between the two curves.
Identifying differentially expressed genes

- Hard to perform manual comparison.
- Sampling rates and different timing prevent direct comparison.

Zhu et al, Nature 2000
Results for the Fkh1/2 Knockout
Issues to address

- Continuous representation
- Alignment
- Identifying differentially expressed genes
- Synchronization
Synchronization

Smc3: observed values

A major problem with human data (less than one cycle is synchronized)
Population effects

- Microarray experiments profile population of cells.
- Initially cells are synchronized, but they lose their synchronization over time.
- Need to compensate for synchronization loss in order to recover single cell values.
FACS data

- FACS: Fluorescence-Activated Cell Sorting
Modeling synchronization loss

![Graph showing cell cycle phases G1, S, and G2/M over time. The x-axis represents time, and the y-axis represents the percentage of cells. The graph illustrates the transition of cells through different phases of the cell cycle.](image-url)
Networks

Pattern Recognition

Individual Gene

Experimental Design
Clustering

• Handling non uniform sampling rates.

• Identifying relationships between genes based on expression profiles.

• Determining relationships between clusters.
Qian et al Journal of Molecular Biology 2001
Results

Simultaneous expression profile relationships:

Inverted expression profile relationships:

Time delayed expression profile relationships
Results – Synthetic Data

Hierarchical clustering  Input  Optimal ordering

Hierarchical clustering  Input  Optimal ordering
MIAME, we have a problem

Robert Shields

*Trends in Genetics*, Elsevier, 84 Theobald’s Road, London, UK, WC1X 8RR

Microarrays have captured the imagination of geneticists and molecular biologists like no other technology, with the exception of perhaps PCR. Descended from the humble consistency is improved because hybridizing sequences are then detected forms [3]?
(iv) the demand for data analysts that are adequately trained greatly exceeds supply and this is likely to remain so for the foreseeable future.