Goal of Lecture

**Machine Learning** can alleviate the burden of solving many biological problems,

- saving the time and cost required for experiments
- providing predictions that guide new experiments.
Goal of Lecture

Machine Learning can alleviate the burden of solving many biological problems,
- saving the time and cost required for experiments
- providing predictions that guide new experiments.

The goal of this tutorial is to raise awareness and comprehension of machine learning
so that biologists can properly match the task at hand to the corresponding analytical approach
Abstract

We explore representative models, from traditional statistical models to recent machine learning models, presenting several up-to-date research projects in bioinformatics to exemplify how biological questions can benefit from a machine learning approach.
Content

1. Basics
2. Tasks
3. Learning
4. Models with Examples
5. Evaluation and Statistical Tests
Content

1. Basics
2. Tasks
3. Learning
4. Models with Examples
Basics
**Data Representation**

**Data Table (Data Base)**

<table>
<thead>
<tr>
<th></th>
<th>A1</th>
<th>A2</th>
<th>A3</th>
<th>...</th>
<th>A10</th>
<th>y</th>
<th>y</th>
</tr>
</thead>
<tbody>
<tr>
<td>X1</td>
<td>10</td>
<td>5</td>
<td>red</td>
<td>...</td>
<td>1000</td>
<td>class1</td>
<td>1</td>
</tr>
<tr>
<td>X2</td>
<td>6</td>
<td>6</td>
<td>blue</td>
<td>...</td>
<td>3500</td>
<td>class2</td>
<td>20</td>
</tr>
<tr>
<td>X3</td>
<td>7</td>
<td>7</td>
<td>yellow</td>
<td>...</td>
<td>400</td>
<td>class1</td>
<td>45</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>X18</td>
<td>3</td>
<td>56</td>
<td>red</td>
<td>...</td>
<td>0</td>
<td>class2</td>
<td>30</td>
</tr>
<tr>
<td>X19</td>
<td>15</td>
<td>62</td>
<td>red</td>
<td>...</td>
<td>500</td>
<td>class1</td>
<td>100</td>
</tr>
<tr>
<td>X20</td>
<td>3</td>
<td>88</td>
<td>blue</td>
<td>...</td>
<td>700</td>
<td>class2</td>
<td>3</td>
</tr>
</tbody>
</table>

20 records,  
20 samples,  
20 observations,  
20 objects,  
20 data points,  
20 individuals,  
20 experimental units, etc.
### Data Representation

A\textsubscript{j}  
attribute,  
feature,  
descriptor,  
input variable,  
predictor variable,  
independent variable,  
exogeneous variable,  
etc,

<table>
<thead>
<tr>
<th>( \mathbf{X}_1 )</th>
<th>( A_1 )</th>
<th>( A_2 )</th>
<th>( A_3 )</th>
<th>\ldots</th>
<th>( A_{10} )</th>
<th>( y )</th>
<th>( y )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \mathbf{X}_2 )</td>
<td>6</td>
<td>6</td>
<td>blue</td>
<td>\ldots</td>
<td>3500</td>
<td>class1</td>
<td>20</td>
</tr>
<tr>
<td>( \mathbf{X}_3 )</td>
<td>7</td>
<td>7</td>
<td>yellow</td>
<td>\ldots</td>
<td>400</td>
<td>class1</td>
<td>45</td>
</tr>
<tr>
<td>\ldots</td>
<td>\ldots</td>
<td>\ldots</td>
<td>\ldots</td>
<td>\ldots</td>
<td>\ldots</td>
<td>\ldots</td>
<td>\ldots</td>
</tr>
<tr>
<td>( \mathbf{X}_{18} )</td>
<td>3</td>
<td>56</td>
<td>red</td>
<td>\ldots</td>
<td>0</td>
<td>class2</td>
<td>30</td>
</tr>
<tr>
<td>( \mathbf{X}_{19} )</td>
<td>15</td>
<td>62</td>
<td>red</td>
<td>\ldots</td>
<td>500</td>
<td>class1</td>
<td>100</td>
</tr>
<tr>
<td>( \mathbf{X}_{20} )</td>
<td>3</td>
<td>88</td>
<td>blue</td>
<td>\ldots</td>
<td>700</td>
<td>class2</td>
<td>3</td>
</tr>
</tbody>
</table>
### Data Representation

Given a set of inputs $X_i$ (input, predictor, etc.), we have a table that represents the data in two forms:

- **1st Form:**
  
<table>
<thead>
<tr>
<th>$A_1$</th>
<th>$A_2$</th>
<th>$A_3$</th>
<th>...</th>
<th>$A_{10}$</th>
<th>$y$</th>
<th>$y$</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>5</td>
<td>red</td>
<td>...</td>
<td>1000</td>
<td>class1</td>
<td>1</td>
</tr>
<tr>
<td>6</td>
<td>6</td>
<td>blue</td>
<td>...</td>
<td>3500</td>
<td>class2</td>
<td>20</td>
</tr>
<tr>
<td>7</td>
<td>7</td>
<td>yellow</td>
<td>...</td>
<td>400</td>
<td>class1</td>
<td>45</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>$X_{18}$</td>
<td>3</td>
<td>56</td>
<td>red</td>
<td>...</td>
<td>0</td>
<td>class2</td>
</tr>
<tr>
<td>$X_{19}$</td>
<td>15</td>
<td>62</td>
<td>red</td>
<td>...</td>
<td>500</td>
<td>class1</td>
</tr>
<tr>
<td>$X_{20}$</td>
<td>3</td>
<td>88</td>
<td>blue</td>
<td>...</td>
<td>700</td>
<td>class2</td>
</tr>
</tbody>
</table>

- **2nd Form:**

*Input set:* $X = \{X_1, X_2, ..., X_{20}\}$
**Data Representation**

\[ yi \]
output variable, response, target variable, endogeneous variable, label, etc.

*Output(Target) Set:*  \[ Y = \{y_1, y_2, \ldots, y_{20}\} \]
1. Basics
2. Tasks
3. Learning
4. Models with Examples
Tasks
Tasks

- Prediction
  - Classification
  - Regression

- Description
  - Clustering
  - Feature Description

- Dimensionality Reduction
  - Feature Selection
  - Feature Extraction

- Data Reduction (Sample Selection)

- Data Integration
Classification is concerned with the problem of **separating** distinct sets of data points and **allocating** new (test or unknown) data points to previously defined group (class)
## Classification

### Input

<table>
<thead>
<tr>
<th></th>
<th>A1</th>
<th>A2</th>
<th>A3</th>
<th>...</th>
<th>A10</th>
</tr>
</thead>
<tbody>
<tr>
<td>X1</td>
<td>10</td>
<td>5</td>
<td>red</td>
<td>...</td>
<td>1000</td>
</tr>
<tr>
<td>X2</td>
<td>6</td>
<td>6</td>
<td>blue</td>
<td>...</td>
<td>3500</td>
</tr>
<tr>
<td>X3</td>
<td>7</td>
<td>7</td>
<td>yellow</td>
<td>...</td>
<td>400</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>X20</td>
<td>3</td>
<td>88</td>
<td>blue</td>
<td>...</td>
<td>700</td>
</tr>
</tbody>
</table>

### Target

<table>
<thead>
<tr>
<th></th>
<th>y</th>
<th>y</th>
</tr>
</thead>
<tbody>
<tr>
<td>class1</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>class2</td>
<td>20</td>
<td></td>
</tr>
<tr>
<td>class1</td>
<td>45</td>
<td></td>
</tr>
<tr>
<td>class2</td>
<td>3</td>
<td></td>
</tr>
</tbody>
</table>

**Target variable (y) is categorical**

- Nominal: (ex) yes or no, 1 or -1
- (ex) blue, red, yellow...
- Ordinal: (ex) age groups
  - (10-20, 20-30, 30-40,...)
* Predicted Value

$f \{ \text{model, function} \}$

$f(x) \{ \text{predicted value (output), output, score, etc} \}$
Classification

Class 1: \( y = 1 \)
Class 2: \( y = -1 \)

\[
\begin{align*}
\text{if } f(x) &< -\infty \quad \text{(class 2)} \\
\text{if } f(x) &\geq 0 \\
\text{if } f(x) &> 0
\end{align*}
\]

\[
f(x) = \begin{cases} 
+1 \text{ (class1) } & \text{if } f(x) \geq 0 \\
-1 \text{ (class2) } & \text{if } f(x) < 0
\end{cases}
\]
Regression

Regression is concerned with the problem of predicting the value of continuous target variable.
Regression

<table>
<thead>
<tr>
<th>Input</th>
<th>A1</th>
<th>A2</th>
<th>A3</th>
<th>…</th>
<th>A10</th>
<th>Target</th>
<th>y</th>
<th>y</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\mathbf{x}_1$</td>
<td>10</td>
<td>5</td>
<td>red</td>
<td>…</td>
<td>1000</td>
<td>class1</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>$\mathbf{x}_2$</td>
<td>6</td>
<td>6</td>
<td>blue</td>
<td>…</td>
<td>3500</td>
<td>class2</td>
<td>20</td>
<td></td>
</tr>
<tr>
<td>$\mathbf{x}_3$</td>
<td>7</td>
<td>7</td>
<td>yellow</td>
<td>…</td>
<td>400</td>
<td>class1</td>
<td>45</td>
<td></td>
</tr>
<tr>
<td>…</td>
<td>…</td>
<td>…</td>
<td>…</td>
<td>…</td>
<td>…</td>
<td>…</td>
<td>…</td>
<td>…</td>
</tr>
<tr>
<td>$\mathbf{x}_{20}$</td>
<td>3</td>
<td>88</td>
<td>blue</td>
<td>…</td>
<td>700</td>
<td>class2</td>
<td>3</td>
<td></td>
</tr>
</tbody>
</table>

Target variable ($y$) is continuous
Regression

\[ y_{f(x)} = \hat{y} \]

\[ f(x) = y \]

\[ f(x) \approx y \]
* Scales of Variable

** Classification can be regarded as a subset of Regression in viewpoint of modeling (not task).
Clustering

Clustering is concerned with the identification of groups of similar data points based on similarity measures.
Clustering

Clustering is concerned with the identification of groups of similar data points based on similarity measures.

Clustering is distinct from classification in that

• Classification pertains to a known number of groups and its operational objective is to assign new data points to one of these groups

• Clustering makes no assumption concerning the number of groups
## Clustering

### Input

<table>
<thead>
<tr>
<th></th>
<th>$A_1$</th>
<th>$A_2$</th>
<th>$A_3$</th>
<th>...</th>
<th>$A_{10}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$X_1$</td>
<td>10</td>
<td>5</td>
<td>red</td>
<td>...</td>
<td>1000</td>
</tr>
<tr>
<td>$X_2$</td>
<td>6</td>
<td>6</td>
<td>blue</td>
<td>...</td>
<td>3500</td>
</tr>
<tr>
<td>$X_3$</td>
<td>7</td>
<td>7</td>
<td>yellow</td>
<td>...</td>
<td>400</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>$X_{20}$</td>
<td>3</td>
<td>88</td>
<td>blue</td>
<td>...</td>
<td>700</td>
</tr>
</tbody>
</table>

**No target variable ($y$)**
Clustering
Clustering

2 clusters

3 clusters

4 clusters

HyunJung (Helen) Shin, Max Planck Society, European School of Genetic Medicine, 03. 2006
Dimensionality reduction is concerned with the process which removes irrelevant or redundant features (attributes) from the original feature set, in order to avoid “curse of dimensionality”—complication of learning process, erroneous results, computational burden.

* note: irrelevant or redundant for learning or modeling
**Dimensionality Reduction**

- **Feature Selection**
  A process of finding a subset of relevant features (attributes) from the original set of features.

(Ex) **Selected Features: A1, A1000**

<table>
<thead>
<tr>
<th></th>
<th>A1</th>
<th>A2</th>
<th>A3</th>
<th>...</th>
<th>A1000</th>
<th>y</th>
</tr>
</thead>
<tbody>
<tr>
<td>X1</td>
<td>10</td>
<td>5</td>
<td>red</td>
<td>...</td>
<td>1000</td>
<td>1</td>
</tr>
<tr>
<td>X2</td>
<td>6</td>
<td>6</td>
<td>blue</td>
<td>...</td>
<td>3500</td>
<td>20</td>
</tr>
<tr>
<td>X3</td>
<td>7</td>
<td>7</td>
<td>yellow</td>
<td>...</td>
<td>400</td>
<td>45</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>X20</td>
<td>3</td>
<td>88</td>
<td>blue</td>
<td>...</td>
<td>700</td>
<td>3</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>A1</th>
<th>A1000</th>
<th>y</th>
</tr>
</thead>
<tbody>
<tr>
<td>X1</td>
<td>10</td>
<td>1000</td>
<td>1</td>
</tr>
<tr>
<td>X2</td>
<td>6</td>
<td>3500</td>
<td>20</td>
</tr>
<tr>
<td>X3</td>
<td>7</td>
<td>400</td>
<td>45</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>X20</td>
<td>3</td>
<td>700</td>
<td>3</td>
</tr>
</tbody>
</table>
Dimensionality Reduction

• Feature Extraction
A process of defining new descriptors (features) condensed via transformations of the raw features. The descriptors are represented as the features in the new feature space.

(Ex) Extracted Features:

\[
\begin{align*}
P_1 &= \beta_1 A_1 + \beta_2 A_2 + \beta_3 A_1 A_350 \\
P_2 &= \Phi (A_1, A_2, \ldots, A_{1000})
\end{align*}
\]

<table>
<thead>
<tr>
<th>( A_1 )</th>
<th>( A_2 )</th>
<th>( A_3 )</th>
<th>( \ldots )</th>
<th>( A_{1000} )</th>
<th>( y )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( x_1 )</td>
<td>10</td>
<td>5</td>
<td>red</td>
<td>( \ldots )</td>
<td>1000</td>
</tr>
<tr>
<td>( x_2 )</td>
<td>6</td>
<td>6</td>
<td>blue</td>
<td>( \ldots )</td>
<td>3500</td>
</tr>
<tr>
<td>( x_3 )</td>
<td>7</td>
<td>7</td>
<td>yellow</td>
<td>( \ldots )</td>
<td>400</td>
</tr>
<tr>
<td>( \ldots )</td>
<td>( \ldots )</td>
<td>( \ldots )</td>
<td>( \ldots )</td>
<td>( \ldots )</td>
<td>( \ldots )</td>
</tr>
<tr>
<td>( x_{20} )</td>
<td>3</td>
<td>88</td>
<td>blue</td>
<td>( \ldots )</td>
<td>700</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>( P_1 )</th>
<th>( P_2 )</th>
<th>( y )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( x_1 )</td>
<td>-2</td>
<td>-0.5</td>
</tr>
<tr>
<td>( x_2 )</td>
<td>0</td>
<td>0.01</td>
</tr>
<tr>
<td>( x_3 )</td>
<td>25</td>
<td>0.9</td>
</tr>
<tr>
<td>( \ldots )</td>
<td>( \ldots )</td>
<td>( \ldots )</td>
</tr>
<tr>
<td>( x_{20} )</td>
<td>10</td>
<td>-0.7</td>
</tr>
</tbody>
</table>
Data Reduction (Sample Selection)

Data Reduction is concerned with the process which removes irrelevant or redundant “data points” from the original data set,

in order to avoid complication of learning process or computational burden.

* note: irrelevant or redundant for learning or modeling
# Data Reduction (Sample Selection)

(Ex) Selected Data Points: $x_2$, $x_3$, $x_{100}$, $x_{9999}$

<table>
<thead>
<tr>
<th></th>
<th>$A_1$</th>
<th>$A_2$</th>
<th>$A_3$</th>
<th>...</th>
<th>$A_{10}$</th>
<th>$y$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$X_1$</td>
<td>10</td>
<td>5</td>
<td>red</td>
<td>...</td>
<td>1000</td>
<td>class1</td>
</tr>
<tr>
<td>$X_2$</td>
<td>6</td>
<td>6</td>
<td>blue</td>
<td>...</td>
<td>3500</td>
<td>class2</td>
</tr>
<tr>
<td>$X_3$</td>
<td>7</td>
<td>7</td>
<td>yellow</td>
<td>...</td>
<td>400</td>
<td>class1</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>$X_{100}$</td>
<td>3</td>
<td>88</td>
<td>blue</td>
<td>...</td>
<td>700</td>
<td>class1</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>$X_{500}$</td>
<td>60</td>
<td>68</td>
<td>red</td>
<td>...</td>
<td>1700</td>
<td>class2</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>$X_{9999}$</td>
<td>3</td>
<td>85</td>
<td>green</td>
<td>...</td>
<td>2500</td>
<td>class2</td>
</tr>
<tr>
<td>$X_{100000}$</td>
<td>3</td>
<td>1</td>
<td>blue</td>
<td>...</td>
<td>5700</td>
<td>class1</td>
</tr>
</tbody>
</table>

$f$

<table>
<thead>
<tr>
<th></th>
<th>$A_1$</th>
<th>$A_2$</th>
<th>$A_3$</th>
<th>...</th>
<th>$A_{10}$</th>
<th>$y$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$X_2$</td>
<td>6</td>
<td>6</td>
<td>blue</td>
<td>...</td>
<td>3500</td>
<td>class2</td>
</tr>
<tr>
<td>$X_3$</td>
<td>7</td>
<td>7</td>
<td>yellow</td>
<td>...</td>
<td>400</td>
<td>class1</td>
</tr>
<tr>
<td>$X_{100}$</td>
<td>3</td>
<td>88</td>
<td>blue</td>
<td>...</td>
<td>700</td>
<td>class1</td>
</tr>
<tr>
<td>$X_{9999}$</td>
<td>3</td>
<td>85</td>
<td>green</td>
<td>...</td>
<td>2500</td>
<td>class2</td>
</tr>
</tbody>
</table>

HyunJung (Helen) Shin, Max Planck Society, European School of Genetic Medicine, 03. 2006
Data Reduction (Sample Selection)

(Ex) Classification

Random Sampling

Informative Sampling

Class 1

Class 2

Class 1

Class 2

HyunJung (Helen) Shin, Max Planck Society, European School of Genetic Medicine, 03. 2006
Data Integration

Data Integration is concerned with the integration of different or heterogeneous data sources (sets) in order to enhance the total information about the problem at hand.

Each data source contains partly independent and partly complementary pieces of information about the problem.
Data Integration

Ex) Heterogeneous Representation of Multiple Data Sources

Vectorial Data

<table>
<thead>
<tr>
<th></th>
<th>A1</th>
<th>A2</th>
<th>...</th>
<th>A10</th>
<th>y</th>
</tr>
</thead>
<tbody>
<tr>
<td>x1</td>
<td>10</td>
<td>5</td>
<td>...</td>
<td>1000</td>
<td>1</td>
</tr>
<tr>
<td>x2</td>
<td>6</td>
<td>6</td>
<td>...</td>
<td>3500</td>
<td>-1</td>
</tr>
<tr>
<td>x3</td>
<td>7</td>
<td>7</td>
<td>...</td>
<td>400</td>
<td>1</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>x7</td>
<td>3</td>
<td>88</td>
<td>...</td>
<td>700</td>
<td>-1</td>
</tr>
</tbody>
</table>

Graph (network)

Sequence (string)

<table>
<thead>
<tr>
<th></th>
<th>x1</th>
<th>x2</th>
<th>...</th>
<th>x7</th>
</tr>
</thead>
<tbody>
<tr>
<td>a1</td>
<td>aegcttttagctatatgcgtatagggct</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>a2</td>
<td>cagtgtcgaatagccgctcgaaaaa</td>
<td>-1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>a7</td>
<td>catgttatcctgatccgttagcttcatgc</td>
<td>1</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

HyunJung (Helen) Shin, Max Planck Society, European School of Genetic Medicine, 03. 2006
Content

1. Basics
2. Tasks
3. Learning
4. Models with Examples
Learning
Learning

Building a model $f$ given dataset $\{X,Y\}$ is called “Learning” or “Training”
Learning

Building a model \( f \) given dataset \( \{X,Y\} \) is called “Learning” or “Training”

**Regression**

Ex) Regression Model

\[
f(x) = \beta_1 x^2 + \beta_2 x + c
\]

In other words, given data \( \{X, Y\} \), finding the values of parameters, \( \beta_1, \beta_2, \) and \( c \) is “Learning”
## Data Set Split

```
<table>
<thead>
<tr>
<th></th>
<th>A1</th>
<th>A2</th>
<th>A3</th>
<th>...</th>
<th>A10</th>
<th>y</th>
</tr>
</thead>
<tbody>
<tr>
<td>X1</td>
<td>10</td>
<td>5</td>
<td>red</td>
<td></td>
<td>1000</td>
<td>1</td>
</tr>
<tr>
<td>X2</td>
<td>6</td>
<td>6</td>
<td>blue</td>
<td></td>
<td>3500</td>
<td>20</td>
</tr>
<tr>
<td>X3</td>
<td>7</td>
<td>7</td>
<td>yellow</td>
<td></td>
<td>400</td>
<td>45</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>X18</td>
<td>3</td>
<td>56</td>
<td>red</td>
<td></td>
<td>0</td>
<td>30</td>
</tr>
<tr>
<td>X19</td>
<td>15</td>
<td>62</td>
<td>red</td>
<td></td>
<td>500</td>
<td>100</td>
</tr>
<tr>
<td>X20</td>
<td>3</td>
<td>88</td>
<td>blue</td>
<td></td>
<td>700</td>
<td>3</td>
</tr>
<tr>
<td>X21</td>
<td>5</td>
<td>42</td>
<td>red</td>
<td></td>
<td>560</td>
<td>?</td>
</tr>
<tr>
<td></td>
<td>...</td>
<td>...</td>
<td></td>
<td>...</td>
<td>...</td>
<td>?</td>
</tr>
<tr>
<td>X50</td>
<td>25</td>
<td>56</td>
<td>blue</td>
<td></td>
<td>600</td>
<td>?</td>
</tr>
</tbody>
</table>
```

“Known” data points

“Unknown” data points

*Known* data points

*Unknown* data points
Data Set Split

“Known” data points

Training set
Training (or learning or building) a model $f$

* Model: $f(x) = \beta_1 x^2 + \beta_2 x + c$

Validation set
Model selection (or model parameter selection)
* Best parameters ($\beta_1, \beta_2, c$) ?

“Unknown” data points

Test set
Prediction with a trained model
Data Set Split & Learning

Training set

Build a model “f” minimizing the errors

\[ \sum_{i=1}^{n} \varepsilon^2 = \sum_{i=1}^{n} (f(x_i) - y_i)^2 \]

Prediction Target

Error (SSE: sum of squared error)
Train a model "f" minimizing the errors

\[
\text{min. } \sum_{i=1}^{n} \varepsilon^2 = \sum_{i=1}^{n} (f(x_i) - y_i)^2
\]
**Data Set Split & Learning**

**Training set**

Build a model “f” minimizing the errors

\[
\text{min. } \sum_{i=1}^{n} \varepsilon^2 = \sum_{i=1}^{n} (f(x_i) - y_i)^2
\]
Data Set Split & Learning

Training Error

\[ \sum_{i=1}^{n} \varepsilon^2 = \sum_{i=1}^{n} (f(x_i) - y_i)^2 \]

Learning (Training) Error Curve

HyunJung (Helen) Shin, Max Planck Society, European School of Genetic Medicine, 03. 2006
Data Set Split & Learning

Learning (Training) Error Curve

HyunJung (Helen) Shin, Max Planck Society, European School of Genetic Medicine, 03. 2006
Data Set Split & Learning

Training Error Curve

Learning (Training) Error Curve

HyunJung (Helen) Shin, Max Planck Society, European School of Genetic Medicine, 03. 2006
Data Set Split & Learning

Learning (Training) Error Curve

HyunJung (Helen) Shin, Max Planck Society, European School of Genetic Medicine, 03. 2006
Data Set Split & Learning

Error of $f_A$ > Error of $f_B$ > Error of $f_C$

Learning (Training) Error Curve

HyunJung (Helen) Shin, Max Planck Society, European School of Genetic Medicine, 03. 2006
Data Set Split & Learning

*Training set*

\[ f_C \text{ is the best model?} \]

\[
\begin{array}{c}
  y \\
  \uparrow \\
  0 \\
\end{array}
\]

\[
\begin{array}{c}
  x \\
  \downarrow \\
  x \\
\end{array}
\]

\[ f_C \]
Data Set Split & Learning

Training set

fc is the best model? No

Why not?

fc

HyunJung (Helen) Shin, Max Planck Society, European School of Genetic Medicine, 03. 2006
Data Set Split & Learning

Training set

Test set

$f_c$
The data points in Test set are assumed to be drawn from the same distribution as those in Training set.
However, when the fully trained model $f_C$ is applied to the test data points, it does not fit them well any more.
Data Set Split & Learning

Training set

However,
when the fully trained model $f_c$ is applied to the
test data points, it does not fit them well any more

Test set

$y$

$0$

$x$

$y$

$0$

$x$

Poor Generalization Ability
Data Set Split & Learning

On the contrary, a “properly” trained model $f_B$ has more generalization ability.
Then, how can we find a “proper” model with absence of Test set?
Then, how can we find a “proper” model with absence of Test set?

Use Validation set (say, a Pseudo Test Set)!

: Temporarily assume that the data set is “Unknown”
Data Set Split & Learning

- Training set
- Validation set

Error

Learning Progress

Training Error

Validation Error
Data Set Split & Learning

Training Error

Validation Error

HyunJung (Helen) Shin, Max Planck Society, European School of Genetic Medicine, 03. 2006
Data Set Split & Learning

Early Stopping

Training Error

Validation Error

Learning Progress

Error

HyunJung (Helen) Shin, Max Planck Society, European School of Genetic Medicine, 03. 2006
Data Set Split & Learning

- **Underfitting**: Model is too simple, does not fit the data well.
- **Overfitting**: Model is too complex, fits the noise in the data.
- **Early Stopping**: Technique to avoid overfitting by stopping training before the error on a validation set begins to increase.

**Graphical Representation**

- **Training Error**: Decreases during the learning process.
- **Validation Error**: Increases when the model starts to overfit.

**Equations**

- **$f_A$**
- **$f_B$**
- **$f_C$**
Data Set Split & Learning

If the known data points are large enough for training after separating the validation set off....
Data Set Split & Learning

If the known data points are insufficient for training after taking the validation set out?
Data Set Split & Learning

(Ex) 5 Cross-Validation (5CV)

Validation

Training

Cross-Validation Error = Validation Error (fold 1) + Validation Error (fold 2) + \ldots + Validation Error (fold 5)

\[ 5 \]
Learning Schemes

• Supervised
• Unsupervised
• Semi-Supervised
Supervised Learning

Learning or Training

\[ f(x) \approx y : \text{Supervised} \]

Input: \( x \)
Output: \( y \)

<table>
<thead>
<tr>
<th>( x_1 )</th>
<th>( x_2 )</th>
<th>( x_3 )</th>
<th>( \ldots )</th>
<th>( x_{19} )</th>
<th>( y )</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>5</td>
<td>red</td>
<td>\ldots</td>
<td>1000</td>
<td>1</td>
</tr>
<tr>
<td>6</td>
<td>6</td>
<td>blue</td>
<td>\ldots</td>
<td>3500</td>
<td>20</td>
</tr>
<tr>
<td>7</td>
<td>7</td>
<td>yellow</td>
<td>\ldots</td>
<td>400</td>
<td>45</td>
</tr>
<tr>
<td>\ldots</td>
<td>\ldots</td>
<td>\ldots</td>
<td>\ldots</td>
<td>\ldots</td>
<td>\ldots</td>
</tr>
<tr>
<td>3</td>
<td>56</td>
<td>red</td>
<td>\ldots</td>
<td>0</td>
<td>30</td>
</tr>
<tr>
<td>15</td>
<td>62</td>
<td>red</td>
<td>\ldots</td>
<td>500</td>
<td>100</td>
</tr>
<tr>
<td>3</td>
<td>88</td>
<td>blue</td>
<td>\ldots</td>
<td>700</td>
<td>3</td>
</tr>
<tr>
<td>5</td>
<td>42</td>
<td>red</td>
<td>\ldots</td>
<td>560</td>
<td>?</td>
</tr>
<tr>
<td>\ldots</td>
<td>\ldots</td>
<td>\ldots</td>
<td>\ldots</td>
<td>\ldots</td>
<td>\ldots</td>
</tr>
<tr>
<td>25</td>
<td>56</td>
<td>blue</td>
<td>\ldots</td>
<td>600</td>
<td>?</td>
</tr>
</tbody>
</table>
Supervised Learning

Prediction or Test

Model $f$

Input: $x$

<table>
<thead>
<tr>
<th>$x_1$</th>
<th>$A_1$</th>
<th>$A_2$</th>
<th>$A_3$</th>
<th>...</th>
<th>$A_n$</th>
<th>$y$</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>5</td>
<td>red</td>
<td>...</td>
<td>1000</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>6</td>
<td>blue</td>
<td>...</td>
<td>3500</td>
<td>20</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>7</td>
<td>yellow</td>
<td>...</td>
<td>400</td>
<td>45</td>
<td></td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td></td>
</tr>
<tr>
<td>$x_{20}$</td>
<td>3</td>
<td>56</td>
<td>red</td>
<td>...</td>
<td>0</td>
<td>30</td>
</tr>
<tr>
<td>$x_{21}$</td>
<td>15</td>
<td>62</td>
<td>red</td>
<td>...</td>
<td>500</td>
<td>100</td>
</tr>
<tr>
<td>$x_{22}$</td>
<td>3</td>
<td>88</td>
<td>blue</td>
<td>...</td>
<td>700</td>
<td>3</td>
</tr>
<tr>
<td>$x_{23}$</td>
<td>5</td>
<td>42</td>
<td>red</td>
<td>...</td>
<td>560</td>
<td>?</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>?</td>
</tr>
<tr>
<td>$x_{25}$</td>
<td>25</td>
<td>56</td>
<td>blue</td>
<td>...</td>
<td>600</td>
<td>?</td>
</tr>
</tbody>
</table>
Supervised Learning

Learning or Training

\[ f(x) \approx y : \text{Supervised} \]

Input: \( x \)
Output: \( y \)

\[
\begin{array}{cccccc}
A_1 & A_2 & A_3 & \ldots & A_{10} & y \\
X_1 & 10 & 5 & \text{red} & \ldots & 1000 & 1 \\
X_2 & 6 & 6 & \text{blue} & \ldots & 3500 & 20 \\
X_3 & 7 & 7 & \text{yellow} & \ldots & 400 & 45 \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
X_{10} & 3 & 56 & \text{red} & \ldots & 0 & 30 \\
X_{11} & 15 & 62 & \text{red} & \ldots & 500 & 100 \\
X_{12} & 3 & 88 & \text{blue} & \ldots & 700 & 3 \\
X_{13} & 5 & 42 & \text{red} & \ldots & 560 & ? \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
X_{20} & 25 & 56 & \text{blue} & \ldots & 600 & ? \\
\end{array}
\]

Train
Val
Test

\[
\begin{array}{cccccc}
A_1 & A_2 & A_3 & \ldots & A_{10} & y \\
X_1 & 10 & 5 & \text{red} & \ldots & 1000 & 1 \\
X_2 & 6 & 6 & \text{blue} & \ldots & 3500 & 20 \\
X_3 & 7 & 7 & \text{yellow} & \ldots & 400 & 45 \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
X_{10} & 3 & 56 & \text{red} & \ldots & 0 & 30 \\
X_{11} & 15 & 62 & \text{red} & \ldots & 500 & 100 \\
X_{12} & 3 & 88 & \text{blue} & \ldots & 700 & 3 \\
X_{13} & 5 & 42 & \text{red} & \ldots & 560 & ? \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
X_{20} & 25 & 56 & \text{blue} & \ldots & 600 & ? \\
\end{array}
\]

Train
Val
Test

Prediction or Test

Predicted Output: \( f(x) \)

Input: \( x \)

\[
\begin{array}{cccccc}
A_1 & A_2 & A_3 & \ldots & A_{10} & y \\
X_1 & 10 & 5 & \text{red} & \ldots & 1000 & 1 \\
X_2 & 6 & 6 & \text{blue} & \ldots & 3500 & 20 \\
X_3 & 7 & 7 & \text{yellow} & \ldots & 400 & 45 \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
X_{10} & 3 & 56 & \text{red} & \ldots & 0 & 30 \\
X_{11} & 15 & 62 & \text{red} & \ldots & 500 & 100 \\
X_{12} & 3 & 88 & \text{blue} & \ldots & 700 & 3 \\
X_{13} & 5 & 42 & \text{red} & \ldots & 560 & ? \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
X_{20} & 25 & 56 & \text{blue} & \ldots & 600 & ? \\
\end{array}
\]
### Unsupervised Learning

**Learning or Training**

- **Input:** $x$
- **Model:** $f$
- **Grouping:** $x$

<table>
<thead>
<tr>
<th>Train</th>
<th>Train</th>
<th>Test</th>
<th>Test</th>
</tr>
</thead>
<tbody>
<tr>
<td>$X_1$</td>
<td>$X_2$</td>
<td>$X_3$</td>
<td>$X_4$</td>
</tr>
<tr>
<td>10</td>
<td>5</td>
<td>red</td>
<td>...</td>
</tr>
<tr>
<td>6</td>
<td>6</td>
<td>blue</td>
<td>...</td>
</tr>
<tr>
<td>7</td>
<td>7</td>
<td>yellow</td>
<td>...</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>3</td>
<td>56</td>
<td>red</td>
<td>...</td>
</tr>
<tr>
<td>15</td>
<td>62</td>
<td>red</td>
<td>...</td>
</tr>
<tr>
<td>3</td>
<td>88</td>
<td>blue</td>
<td>...</td>
</tr>
<tr>
<td>5</td>
<td>42</td>
<td>red</td>
<td>...</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>25</td>
<td>56</td>
<td>blue</td>
<td>...</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Val</th>
<th>Val</th>
<th>Test</th>
<th>Test</th>
</tr>
</thead>
<tbody>
<tr>
<td>$X_5$</td>
<td>$X_6$</td>
<td>$X_7$</td>
<td>$X_8$</td>
</tr>
<tr>
<td>10</td>
<td>5</td>
<td>red</td>
<td>...</td>
</tr>
<tr>
<td>6</td>
<td>6</td>
<td>blue</td>
<td>...</td>
</tr>
<tr>
<td>7</td>
<td>7</td>
<td>yellow</td>
<td>...</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>3</td>
<td>56</td>
<td>red</td>
<td>...</td>
</tr>
<tr>
<td>15</td>
<td>62</td>
<td>red</td>
<td>...</td>
</tr>
<tr>
<td>3</td>
<td>88</td>
<td>blue</td>
<td>...</td>
</tr>
<tr>
<td>5</td>
<td>42</td>
<td>red</td>
<td>...</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>25</td>
<td>56</td>
<td>blue</td>
<td>...</td>
</tr>
</tbody>
</table>

**Prediction or Test**

**Input:** $x$

<table>
<thead>
<tr>
<th>Train</th>
<th>Train</th>
<th>Test</th>
<th>Test</th>
</tr>
</thead>
<tbody>
<tr>
<td>$X_9$</td>
<td>$X_{10}$</td>
<td>$X_{11}$</td>
<td>$X_{12}$</td>
</tr>
<tr>
<td>10</td>
<td>5</td>
<td>red</td>
<td>...</td>
</tr>
<tr>
<td>6</td>
<td>6</td>
<td>blue</td>
<td>...</td>
</tr>
<tr>
<td>7</td>
<td>7</td>
<td>yellow</td>
<td>...</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>3</td>
<td>56</td>
<td>red</td>
<td>...</td>
</tr>
<tr>
<td>15</td>
<td>62</td>
<td>red</td>
<td>...</td>
</tr>
<tr>
<td>3</td>
<td>88</td>
<td>blue</td>
<td>...</td>
</tr>
<tr>
<td>5</td>
<td>42</td>
<td>red</td>
<td>...</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>25</td>
<td>56</td>
<td>blue</td>
<td>...</td>
</tr>
</tbody>
</table>
Semi-Supervised Learning

<table>
<thead>
<tr>
<th>Known Data Input: x</th>
<th>Unknown Data Input: x</th>
<th>Known Data Output: y</th>
</tr>
</thead>
</table>

Learning or Training

Prediction or Test

Why “Semi-”? : In learning, supervised for known data, \( f(x) \approx y \), but unsupervised for known data, \( f(x) \approx ?? \)

Model \( f \):

```
<table>
<thead>
<tr>
<th>( x_1 )</th>
<th>( a_1 )</th>
<th>( a_2 )</th>
<th>( a_3 )</th>
<th>...</th>
<th>( a_{10} )</th>
<th>y</th>
</tr>
</thead>
<tbody>
<tr>
<td>( x_1 )</td>
<td>10</td>
<td>5</td>
<td>red</td>
<td>...</td>
<td>1000</td>
<td>1</td>
</tr>
<tr>
<td>( x_2 )</td>
<td>6</td>
<td>6</td>
<td>blue</td>
<td>...</td>
<td>3500</td>
<td>20</td>
</tr>
<tr>
<td>( x_3 )</td>
<td>7</td>
<td>7</td>
<td>yellow</td>
<td>...</td>
<td>400</td>
<td>45</td>
</tr>
<tr>
<td>( x_{10} )</td>
<td>3</td>
<td>56</td>
<td>red</td>
<td>...</td>
<td>0</td>
<td>30</td>
</tr>
<tr>
<td>( x_{11} )</td>
<td>15</td>
<td>62</td>
<td>red</td>
<td>...</td>
<td>500</td>
<td>100</td>
</tr>
<tr>
<td>( x_{12} )</td>
<td>3</td>
<td>88</td>
<td>blue</td>
<td>...</td>
<td>700</td>
<td>3</td>
</tr>
<tr>
<td>( x_{13} )</td>
<td>5</td>
<td>42</td>
<td>red</td>
<td>...</td>
<td>560</td>
<td>??</td>
</tr>
<tr>
<td>( x_{14} )</td>
<td>25</td>
<td>56</td>
<td>blue</td>
<td>...</td>
<td>600</td>
<td>??</td>
</tr>
</tbody>
</table>
```

Train | Val | Test

```
<table>
<thead>
<tr>
<th>( x_1 )</th>
<th>( a_1 )</th>
<th>( a_2 )</th>
<th>( a_3 )</th>
<th>...</th>
<th>( a_{10} )</th>
<th>y</th>
</tr>
</thead>
<tbody>
<tr>
<td>( x_1 )</td>
<td>10</td>
<td>5</td>
<td>red</td>
<td>...</td>
<td>1000</td>
<td>1</td>
</tr>
<tr>
<td>( x_2 )</td>
<td>6</td>
<td>6</td>
<td>blue</td>
<td>...</td>
<td>3500</td>
<td>20</td>
</tr>
<tr>
<td>( x_3 )</td>
<td>7</td>
<td>7</td>
<td>yellow</td>
<td>...</td>
<td>400</td>
<td>45</td>
</tr>
<tr>
<td>( x_{10} )</td>
<td>3</td>
<td>56</td>
<td>red</td>
<td>...</td>
<td>0</td>
<td>30</td>
</tr>
<tr>
<td>( x_{11} )</td>
<td>15</td>
<td>62</td>
<td>red</td>
<td>...</td>
<td>500</td>
<td>100</td>
</tr>
<tr>
<td>( x_{12} )</td>
<td>3</td>
<td>88</td>
<td>blue</td>
<td>...</td>
<td>700</td>
<td>3</td>
</tr>
<tr>
<td>( x_{13} )</td>
<td>5</td>
<td>42</td>
<td>red</td>
<td>...</td>
<td>560</td>
<td>??</td>
</tr>
<tr>
<td>( x_{14} )</td>
<td>25</td>
<td>56</td>
<td>blue</td>
<td>...</td>
<td>600</td>
<td>??</td>
</tr>
</tbody>
</table>
```

Predicted Output: \( f \)
Models
Content

1. Basics
2. Tasks
3. Learning
4. Models with Examples
Models
Models

- Traditional Statistical Methods
- Neural Networks
- Decision Trees
- Kernel Methods
- Semi-Supervised Learning (SSL)
- Ensemble Methods
- Generative (Probabilistic) Methods
# Models: Traditional Statistical Models

<table>
<thead>
<tr>
<th>Task</th>
<th>Model</th>
</tr>
</thead>
<tbody>
<tr>
<td>Classification</td>
<td>• Logistic Regression,</td>
</tr>
<tr>
<td></td>
<td>• Discriminant Analysis, etc</td>
</tr>
<tr>
<td>Regression</td>
<td>• Regression, etc</td>
</tr>
<tr>
<td>Clustering</td>
<td>• k-Means Clustering,</td>
</tr>
<tr>
<td></td>
<td>• Agglomerative (hierarchical) Clustering, etc</td>
</tr>
<tr>
<td>Feature Extraction &amp; Selection</td>
<td>• Principal Component Analysis (PCA),</td>
</tr>
<tr>
<td></td>
<td>• Canonical Correlation Analysis (CCA),</td>
</tr>
<tr>
<td></td>
<td>• Factor Analysis (FA), etc</td>
</tr>
<tr>
<td>Sample Selection</td>
<td>• Random Sampling,</td>
</tr>
<tr>
<td></td>
<td>• Stratified Sampling, etc</td>
</tr>
<tr>
<td>Variance Analysis</td>
<td>• ANalysis Of VAriance (ANOVA)</td>
</tr>
<tr>
<td>Significance Validation</td>
<td>• Hypothesis and Test</td>
</tr>
</tbody>
</table>
Models: Neural Networks

<table>
<thead>
<tr>
<th>Task</th>
<th>Model</th>
</tr>
</thead>
<tbody>
<tr>
<td>Classification</td>
<td>Multi-Layer Perceptron (MLP) Network</td>
</tr>
<tr>
<td>Regression</td>
<td>Self-Organized Map, etc</td>
</tr>
<tr>
<td>Clustering</td>
<td>Radial Basis Function Network</td>
</tr>
<tr>
<td>Feature Extraction &amp; Selection</td>
<td>etc, etc, etc,...</td>
</tr>
</tbody>
</table>
### Models: Decision Trees (or Rule-base)

<table>
<thead>
<tr>
<th>Task</th>
<th>Model</th>
</tr>
</thead>
<tbody>
<tr>
<td>Classification</td>
<td>CART, C4.5, CHAID, MARS, QUEST, FOREST, etc, etc, etc, ...</td>
</tr>
<tr>
<td>Regression</td>
<td>CHAID</td>
</tr>
<tr>
<td>Feature Extraction</td>
<td>C4.5</td>
</tr>
<tr>
<td>&amp; Selection</td>
<td>MARS, QUEST, FOREST, etc, etc, etc, ...</td>
</tr>
</tbody>
</table>
Models: Kernel Methods

<table>
<thead>
<tr>
<th>Task</th>
<th>Model</th>
</tr>
</thead>
<tbody>
<tr>
<td>Classification</td>
<td>Support Vector Machines (SVM)</td>
</tr>
<tr>
<td>Regression</td>
<td>kPCA</td>
</tr>
<tr>
<td>Clustering</td>
<td>kCCA</td>
</tr>
<tr>
<td>Feature Extraction &amp; Selection</td>
<td>Kernel Independent Component Analysis (kICA)</td>
</tr>
<tr>
<td></td>
<td>• etc, etc, etc, ...</td>
</tr>
</tbody>
</table>
Models: Semi-Supervised Learning

<table>
<thead>
<tr>
<th>Task</th>
<th>Model</th>
</tr>
</thead>
<tbody>
<tr>
<td>Classification</td>
<td>Semi-Supervised Learning (SSL)</td>
</tr>
<tr>
<td>Regression</td>
<td>Transductive Inference Methods</td>
</tr>
<tr>
<td></td>
<td>etc, etc, etc,...</td>
</tr>
</tbody>
</table>

*Note that, currently, the term of “Semi-Supervised Learning” has been used as a name of “model” as well as the concept of “learning scheme”*
**Models: Generative (Probabilistic) Methods**

<table>
<thead>
<tr>
<th>Task</th>
<th>Model</th>
</tr>
</thead>
<tbody>
<tr>
<td>Classification</td>
<td>Bayesian Models</td>
</tr>
<tr>
<td>Regression</td>
<td>Naive Bayes</td>
</tr>
<tr>
<td>Clustering</td>
<td>Gaussian Process</td>
</tr>
</tbody>
</table>

etc, etc, etc, etc, ...  •

etc, etc, etc, etc, ...  •
**Models: Ensemble Methods**

<table>
<thead>
<tr>
<th>Task</th>
<th>Model</th>
</tr>
</thead>
<tbody>
<tr>
<td>Better Performance for Various Tasks</td>
<td>Bagging, Boosting, Arcing, etc, etc, etc,...</td>
</tr>
</tbody>
</table>

HyunJung (Helen) Shin, Max Planck Society, European School of Genetic Medicine, 03. 2006
The Most Up-To-Date Models

Kernel Methods

Support Vector Machines (SVM), kPCA, kCCA, kICA, etc

Semi-Supervised Learning Methods

Graph-based SSL, Transductive Inference Methods, etc

* Note that, currently, the term of “Semi-Supervised Learning” has been used as a name of “model” as well as the concept of “learning scheme”
The Most Up-To-Date Models

**Kernel Methods**

- Support Vector Machines (SVM), kPCA, kCCA, kICA, etc

**Semi-Supervised Learning Methods**

- Graph-based SSL, Transductive Inference Methods
Kernel Methods:
Support Vector Machines (SVM)
Kernel Methods

Why KM?

• Kernel methods can operate on very general types of data and can detect very general types of relations

\[
\begin{align*}
\text{PCA,} & \quad \text{CCA,} \\
\text{FA,} & \quad \text{DA,} \\
\text{Clustering} & \quad \text{vectors,} \\
\text{} & \quad \text{sequences,} \\
\text{} & \quad \text{text,} \\
\text{} & \quad \text{images,} \\
\text{} & \quad \text{graphs}
\end{align*}
\]

• Various tasks can be performed on diverse data

• Integration of different types of data is easy and natural
Kernel Methods

**Procedure**

**Data Set**

<table>
<thead>
<tr>
<th>A1</th>
<th>A2</th>
<th>...</th>
<th>A5</th>
</tr>
</thead>
<tbody>
<tr>
<td>x1</td>
<td>10</td>
<td>...</td>
<td>100</td>
</tr>
<tr>
<td>x2</td>
<td>6</td>
<td>...</td>
<td>350</td>
</tr>
<tr>
<td>x3</td>
<td>7</td>
<td>...</td>
<td>400</td>
</tr>
</tbody>
</table>
| ...| ...| ... | ...
| x10| 3  | 88  | 700|

10 data points
5 attributes

**Kernel Function**

\[ K(x_i, x_j) \]

\[ K_{ij} = K(x_i, x_j) = \phi(x_i) \cdot \phi(x_j) \] where \( \Phi(.) \) is a mapping function

**Modeling**

- SVM
- CCA
- PCA
- FA
- DA
- Clustering
- etc

**Model Output**

\[ f(x) \]

**Kernel Matrix: \( K \)**

10 x 10 matrix

*HyunJung (Helen) Shin, Max Planck Society, European School of Genetic Medicine, 03. 2006*
Kernel Methods

"Hyper-Surface"

"Hyper-Plane"

KM operates in Feature Space!

Mapping \( \Phi(\cdot) \)

Feature Space

Input Space \( I \)

Feature Space \( \Phi \)
Kernel Methods

Feature Space

The Mapping from Input to Feature space is...

- Highly Nonlinear
- Dimension Expanding (up to infinite dim.)
- Not unique to a Feature Space, Probably Unknown

Finding the mapping function has been the most difficult barrier in the traditional statistics and early machine learning algorithms.

HyunJung (Helen) Shin, Max Planck Society, European School of Genetic Medicine, 03. 2006
Kernel Methods

Kernel Function

\[ K(x, y) = \phi(x) \cdot \phi(y) \]

In KM, those difficulties could be circumvented by means of “Kernel Trick” which replaces the dot product between mapping functions.
Kernel Methods

**Kernel Function**

Kernel Function: \[ K(x, y) = \phi(x) \cdot \phi(y) \]

**Functions Satisfying Mercer’s Theorem**

- **Polynomial kernels**
  \[ K(x, y) = (x \cdot y)^P \]

- **Radial Basis (Gaussian) kernels**
  \[ K(x, y) = \exp \left( \frac{-\|x - y\|^2}{2\sigma^2} \right) \]

- **Sigmoid Kernels (3-MLP NN)**
  \[ K(x, y) = \tanh \left\{ \kappa (x \cdot y) + \Theta \right\} \]
A Single Kernel Produces Multiple Mappings

Ex) Input Space : $R^2$. Polynomial Kernel

$$K(\bar{x}, \bar{y}) = (\bar{x} \cdot \bar{y})^2 = \left(\begin{pmatrix} x_1 \\ x_2 \end{pmatrix}, \begin{pmatrix} y_1 \\ y_2 \end{pmatrix}\right)^2 = \Phi(\bar{x}) \cdot \Phi(\bar{y})$$

(1) Feature Space : $R^3$

$$\Phi(\bar{x}) = \begin{pmatrix} x_1^2 \\ \sqrt{2}x_1x_2 \\ x_2^2 \end{pmatrix}$$

(2) Feature Space : $R^4$

$$\Phi(\bar{x}) = \frac{1}{\sqrt{2}} \begin{pmatrix} x_1^2 - x_1^2 \\ 2x_1x_2 \\ (x_1^2 + x_2^2) \end{pmatrix}$$

(3) Feature Space : $R^4$

$$\Phi(\bar{x}) = \begin{pmatrix} x_1^2 \\ x_1x_2 \\ x_1y_2 \\ x_2^2 \end{pmatrix}$$

$$(\bar{x} \cdot \bar{y})^2 = \left(\begin{pmatrix} x_1 \\ x_2 \end{pmatrix}, \begin{pmatrix} y_1 \\ y_2 \end{pmatrix}\right)^2 = \left(\begin{pmatrix} x_1^2 \\ \sqrt{2}x_1x_2 \\ x_2^2 \end{pmatrix}, \begin{pmatrix} y_1^2 \\ \sqrt{2}y_1y_2 \\ y_2^2 \end{pmatrix}\right) = \Phi(\bar{x}) \cdot \Phi(\bar{y})$$

HyunJung (Helen) Shin, Max Planck Society, European School of Genetic Medicine, 03. 2006
Kernel Methods

The flexible combination of appropriate kernel design and relevant kernel algorithms has given rise to a powerful class of methods, whose computational and statistical properties are well understood.

Particularly, KM has increasingly been used in Bioinformatics as diverse as biosequences and microarray data analysis, etc.
SVM Classification

Basic Idea of SVM

Properties of SVM  ...Optional

- Margin
- Convexity
- Duality
- Kernels
- Sparseness
Basic Idea of SVM
Basic Idea of SVM

SVM looks for the **Separating Hyperplane** with the Largest Margin.

![Diagram showing SVM classification](image)

**Training data**

\[ \{x_i, y_i\}, \ i = 1, \ldots, l, \quad y_i \in \{-1, 1\} \]

**Separating Hyperplane**

\[ f(x) = w \cdot x + b = 0 \]

\[ \text{sign}(f(x)) = \begin{cases} +1 & \text{if } x \cdot w + b \geq 0 \\ -1 & \text{if } x \cdot w + b < 0 \end{cases} \]
Basic Idea of SVM

SVM looks for the Separating Hyperplane with the Largest Margin.

\[ f(x) = x_i \cdot w + b = 0 \]

Class 1 \((y_i = +1)\)

Class 2 \((y_i = -1)\)

Supporting Hyperplanes

\[ H1: x_i \cdot w + b \geq +1 \quad \text{for } y_i = +1 \]
\[ H2: x_i \cdot w + b \leq -1 \quad \text{for } y_i = -1 \]

Margin

Distance between H1 and H2

\[
\frac{|1 - b|}{||w||} - \frac{|-1 - b|}{||w||} = 2
\]

HyunJung (Helen) Shin, Max Planck Society, European School of Genetic Medicine, 03. 2006
Basic Idea of SVM

Find the Pair of Hyperplanes (Support Vectors)

\[ x_i \cdot w + b \geq +1 \quad \text{for } y_i = +1 \]
\[ x_i \cdot w + b \leq -1 \quad \text{for } y_i = -1 \]

under the constraints which gives Maximum Margin \( \frac{2}{||w||} \)
Basic Idea of SVM

Separable Case

Class 1

Class 2

Margin

f(x)
Basic Idea of SVM

Separable Case

Minimize $||w||^2$ under the constraints !!

$$\min \frac{1}{2} ||w||^2$$

s.t. $y_i(x_i \cdot w + b) - 1 \geq 0 \quad \forall_i$

Quadratic Programming

(convex QP : obj fn is convex, constraints form a convex set)
Basic Idea of SVM

Non-Separable Case?

Class 1

Class 2

$f(x)$
Basic Idea of SVM

Use Slack Variables!

\[
\begin{align*}
\min & \quad \frac{1}{2} \| w \|^2 + C \sum_i \xi_i \\
\text{s.t.} & \quad y_i (x_i \cdot w + b) \geq 1 - \xi_i, \quad \forall i
\end{align*}
\]

\textit{C: Error Tolerance Parameter}
Basic Idea of SVM

Nonlinear Case?

Class 1

Class 2

\[ f(x) \]
Basic Idea of SVM

Solve (linear) problem in the Feature Space!

NonLinear Algorithm in Input Space

Linear Algorithm in Feature Space

\[ \Phi: R^n \rightarrow F \]
Basic Idea of SVM

Feature Space

SVMs map the training data nonlinearly into a higher-dimensional feature space via $\phi$ and construct a separating hyperplane with maximum margin there.

This yields a nonlinear decision boundary in input space.
< Example >
Nonlinear & NonSeparable
< Example >
Nonlinear & NonSeparable
Properties of SVM ...optional
SVM looks for the **Separating Hyperplane** with the Largest Margin.

**Training data**

\[ \{x_i, y_i\}, \ i = 1, \ldots, l, \ y_i \in \{-1,1\} \]

**Separating Hyperplane**

\[ f(x) = w \cdot x + b = 0 \]

\[ \text{sign}(f(x)) = \begin{cases} +1 & \text{if } x \cdot w + b \geq 0 \\ -1 & \text{if } x \cdot w + b < 0 \end{cases} \]
SVM looks for the Separating Hyperplane with the Largest Margin.

Class 1 \((y_i = +1)\)

\[ f(x) = \mathbf{x}_i \cdot \mathbf{w} + b = 0 \]

Supporting Hyperplanes

- \(H1: \mathbf{x}_i \cdot \mathbf{w} + b \geq +1 \) for \(y_i = +1\)
- \(H2: \mathbf{x}_i \cdot \mathbf{w} + b \leq -1 \) for \(y_i = -1\)

Margin

Distance between \(H1\) and \(H2\)

\[
\frac{|1-b|}{\|\mathbf{w}\|} - \frac{|-1-b|}{\|\mathbf{w}\|} = 2
\]
SVM looks for the Separating Hyperplane with the Largest Margin.

\[ f(x) = x_i \cdot w + b = 0 \]

**Support Vectors**

\[ H1: x_i \cdot w + b - 1 = 0 \quad \text{for } y_i = +1 \]
\[ H2: x_i \cdot w + b + 1 = 0 \quad \text{for } y_i = -1 \]

\( x_i \)'s are the Closest Data from Separating Hyperplane, \( w \cdot x + b = 0 \)
Find the Pair of Hyperplanes (Support Vectors)

\[ x_i \cdot w + b \geq +1 \quad \text{for} \ y_i = +1 \]
\[ x_i \cdot w + b \leq -1 \quad \text{for} \ y_i = -1 \]

under the constraints which gives Maximum Margin \( \frac{2}{\|w\|} \)!

Minimize \( \|w\|^2 \) under the constraints !!

\[
\min \frac{1}{2} \|w\|^2 \\
\text{s.t.} \quad y_i(x_i \cdot w + b) - 1 \geq 0 \quad \forall_i
\]
Separable Case

Class 1

Class 2

Margin

Convexity  Duality  Kernel  Sparseness

[Margin]
Minimize $||w||^2$ under the constraints!!

$$\min \frac{1}{2} \|w\|^2$$

s.t. $y_i(x_i \cdot w + b) - 1 \geq 0 \quad \forall i$

**Quadratic Programming**

*(convex QP: obj fn is convex, constraints form a convex set)*
Non-Separable Case?
Use Slack Variables!

**Problem**

\[
\begin{align*}
\text{min} \quad & \frac{1}{2} \| w \|^2 \\
\text{s.t.} \quad & y_i(x_i \cdot w + b) - 1 \geq 0 \quad \forall_i
\end{align*}
\]

**NonSeparable Case**

\[
\begin{align*}
\text{min} \quad & \frac{1}{2} \| w \|^2 + C \sum \xi_i \\
\text{s.t.} \quad & y_i(x_i \cdot w + b) \geq 1 - \xi_i, \quad \forall_i
\end{align*}
\]

\text{C: Error Tolerance Parameter}
Primal Problem

\[
\text{Minimize } \|w\|^2 \text{ under the constraints!}
\]

\[
\text{min } \frac{1}{2} \|w\|^2
\]

\[
s.t. \quad y_i(x_i \cdot w + b) - 1 \geq 0 \quad \forall_i
\]

\[\]
Use Lagrange theory!
(Karush-Kuhn-Tucker Condition)
Karush-Kuhn-Tucker Condition

Min: \( f(x) \)

s.t.  \( h(x) = 0 \)  \((m \text{ equality constraints})\)

\( g(x) \leq 0 \)  \((k \text{ inequality constraints})\)

Lagrangian:  
\[ L(x,a,m) = f(x) + a h(x) + \sum u_i (g_i (x) + s_i ) \]

1) Gradient of the Lagrangian = 0

2) Constraints:  \( h(x) = 0 \)  &  \( g(x) \leq 0 \)

3) Complementary Slackness:  \( u \cdot s = 0 \)

4) Feasibility for the inequality constraints:  \( s \geq 0 \)

5) Sign condition on the inequality multipliers:  \( u \geq 0 \)
KKT conditions are satisfied at the solution of any constrained optimization problem.

For convex problem, KKT conditions are necessary and sufficient condition for primal, dual solution.
Margin [Convexity] Duality Kernel Sparseness

Primal Problem

\[
\min \frac{1}{2} \| w \|^2 \\
\text{s.t. } y_i(x_i \cdot w + b) - 1 \geq 0 \quad \forall i
\]

Lagrangian

\[
L(w, b) = \frac{1}{2} \| w \|^2 - \sum_{i}^{l} \alpha_i y_i (x_i \cdot w + b) + \sum_{i}^{l} \alpha_i
\]

HyunJung (Helen) Shin, Max Planck Society, European School of Genetic Medicine, 03. 2006
Margin  [Convexity]  Duality  Kernel  Sparseness

\[ L(w, b) = \frac{1}{2} \| w \|^2 - \sum_{i} \alpha_i y_i (x_i \cdot w + b) + \sum_{i} \alpha_i \]
\[ \ \text{... Lagrangian} \]

\[ \frac{\partial}{\partial w_v} L_P = w_v - \sum_{i} \alpha_i y_i x_i = 0 \]
\[ \frac{\partial}{\partial b} L_P = -\sum_{i} \alpha_i y_i = 0 \]
\[ \text{... Gradient of the Lagrangian} = 0 \]

\[ y_i (x_i \cdot w + b) - 1 \geq 0 \ \forall i \]
\[ \text{... Primal Feasibility} \]

\[ \alpha_i \geq 0 \ \forall i \]
\[ \text{... Dual Feasibility} \]

\[ \alpha_i (y_i (x_i \cdot w + b) - 1) = 0 \ \forall i \]
\[ \text{... Complementarity Conditions} \]
Solving the SVM problem is equivalent to finding a solution KKT conditions.
Lagrangian $L$ has to be minimized w.r.t. the primal variables $w$ and $b$ and maximized w.r.t. the dual variables $\alpha_i$

- **Minimize $L_p$ with respect to $w, b$:**

$$
\min L_p = \frac{1}{2} \|w\|^2 - \sum_{i}^{l} y_i (x_i \cdot w + b) + \sum_{i}^{l} \alpha_i \nabla
\Rightarrow w = \sum_{i}^{l} \alpha_i y_i x_i \quad , \quad \sum_{i}^{l} \alpha_i y_i = 0
$$

- **Maximize $L_D$ with respect to $\alpha_i$:**

$$
\max L_D = \sum_{i}^{l} \alpha_i - \frac{1}{2} \sum_{i,j}^{l} \alpha_i \alpha_j y_i y_j x_i \cdot x_j \\
\text{s.t.} \quad \alpha_i \geq 0, \quad \sum_{i}^{l} \alpha_i y_i = 0, \quad \forall i
$$
Why Dual ?

\[
\begin{align*}
\max \quad & L_D = \sum_{i}^{l} \alpha_i - \frac{1}{2} \sum_{i,j}^{l} \alpha_i \alpha_j y_i y_j x_i \cdot x_j \\
\text{s.t.} \quad & \alpha_i \geq 0, \quad \sum_{i}^{l} \alpha_i y_i = 0, \quad \forall i
\end{align*}
\]
Margin  Convexity  [Duality]  Kernel  Sparseness

Why Dual ?

\[
\begin{align*}
\max_{\alpha_i} L_D &\equiv \sum_{i=1}^{l} \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j \langle x_i, x_j \rangle \\
\text{s.t. } &\alpha_i \geq 0, \quad \sum_{i} \alpha_i y_i = 0, \quad \forall i
\end{align*}
\]

Dot Product between Training Vectors:
We can use Kernel functions!
Margin  Convexity  Duality  [Kernel]  Sparseness

Nonlinear Case?

Class 1

Class 2

\( f(x) \)
Margin  Convexity  Duality  [Kernel]  Sparseness

Nonlinear Mapping from Input space (\(\mathbb{R}^N\)) to Feature Space (F)

(EX) \(\Phi : \mathbb{R}^2 \rightarrow \mathbb{R}^3, (x_1, x_2) \mapsto (z_1, z_2, z_3)\)

**NonLinear Algorithm in Input Space**

**Linear Algorithm in Feature Space**
Feature Space

SVMs map the training data nonlinearly into a higher-dimensional feature space via $\phi$ and construct a separating hyperplane with maximum margin there.

This yields a nonlinear decision boundary in input space.
From Input Space to Feature Space

**Linear Case**

**Problem**

\[
\max L_D = \sum_{i=1}^{l} \alpha_i - \frac{1}{2} \sum_{i,j}^{l} \alpha_i \alpha_j y_i y_j \mathbf{x}_i \cdot \mathbf{x}_j
\]

s.t. \( \alpha_i \geq 0, \sum_{i=1}^{l} \alpha_i y_i = 0, \quad \forall i \)

**Decision Function**

\[
f(x) = \text{sign}(w \cdot \mathbf{x} + b) = \text{sign}(\sum_{i=1}^{l} \alpha_i y_i \mathbf{x}_i \cdot \mathbf{x} + b)
\]

**Nonlinear Case**

**Problem**

\[
\max L_D = \sum_{i=1}^{l} \alpha_i - \frac{1}{2} \sum_{i,j}^{l} \alpha_i \alpha_j y_i y_j \mathbf{\phi}(\mathbf{x}_i) \cdot \mathbf{\phi}(\mathbf{x}_j)
\]

s.t. \( \alpha_i \geq 0, \sum_{i=1}^{l} \alpha_i y_i = 0, \quad \forall i \)

**Decision Function**

\[
f(x) = \text{sign}(w \cdot \mathbf{\phi}(x) + b) = \text{sign}(\sum_{i=1}^{l} \alpha_i y_i \mathbf{\phi}(x_i) \cdot \mathbf{\phi}(x) + b)
\]
Margin  Convexity  Duality  [Kernel]  Sparseness

**Mapping Function (Φ)?**

*However,*

*Mapping function is not unique.*

*Feature Space could be (possibly) infinite dimensional.*

*Computation Demanding*

*difficult to find!*
Margin  Convexity  Duality  [Kernel]  Sparseness

**Mapping Function (\(\Phi\))?**

How can we know the mapping function?

How can we to handle the infinite dimensionality?
Use Kernel Functions!

**SVM depends only on Dot Products between patterns.**

\[
\begin{align*}
\text{max} & \quad L_D \equiv \sum_i \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j \phi(x_i) \cdot \phi(x_j) \\
\text{s.t.} & \quad \alpha_i \geq 0, \quad \sum_i \alpha_i y_i = 0, \quad \forall i
\end{align*}
\]

**Problem**

**Decision Function**

\[
f(x) = \text{sign}(w \cdot \phi(x) + b) = \text{sign} \left( \sum_{i=1}^l \alpha_i y_i \phi(x_i) \cdot \phi(x) + b \right)
\]

*By the use of a kernel function, it is possible to compute the dot product in input space without explicitly carrying out the map into the feature space*
Kernel Function: $k(x, y) = \phi(x) \cdot \phi(y)$

Functions Satisfying Mercers’s Theorem

Polynomial kernels

$k(x, y) = (x \cdot y)^p$

Radial Basis kernels

$k(x, y) = \exp\left(-\frac{\|x - y\|^2}{2\sigma^2}\right)$

Sigmoid Kernels (3-MLP NN)

$k(x, y) = \tanh(\kappa (x \cdot y) + \Theta)$
Margin  Convexity  Duality  [Kernel]  Sparseness

Nonlinear & Nonseparable Case
Only the points nearest to the hyperplane have positive weight!

They are called Support Vectors!
Remind the Complementarity Conditions
\[ \alpha_i (y_i (x_i \cdot w + b) - 1) = 0 \quad \forall i \quad \text{also note that} \quad w = \sum_i \alpha_i y_i x_i \]

Class 1

Class 2

HyunJung (Helen) Shin, Max Planck Society, European School of Genetic Medicine, 03. 2006
**Margin**  Convexity  Duality  Kernel  [Sparseness]

**SVs are distributed around decision boundary!**

1. **Patterns OUT OF THE MARGIN**  \( \alpha_i = 0 \)
2. **Patterns ON THE MARGIN (SVs)**  \( 0 < \alpha_i < C \)
3. **Patterns BETWEEN THE MARGINS (SVs)**  \( \alpha_i = C \)
**SVM Decision Function**

$$f(x) = \begin{cases} 
+1 \text{ (class 1)} & \text{if } f(x) \geq 0 \\
-1 \text{ (class 2)} & \text{if } f(x) < 0 
\end{cases}$$

$$f(x) = \text{sign} (w \cdot x + b) = \text{sign} \left( \sum_{i=1}^{l} \alpha_i y_i x_i \cdot x + b \right)$$

Support Vectors

Test Data Point

where $$w = \sum_{i}^{l} \alpha_i y_i x_i$$
Wrap-up

SVM QP Problem: (Non-linear & Non-Separable)

\[
\begin{align*}
\text{min.} & \quad \frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{i=1}^{M} \xi_i \\
\text{s.t.} & \quad y_i (\mathbf{w} \cdot \Phi(\mathbf{x}_i) + b) \geq 1 - \xi_i, \quad i = 1, \ldots, M
\end{align*}
\]

SVM Decision Function:

\[
f(\mathbf{x}) = \text{sign} \left( \sum_{i \in SV} y_i \alpha_i \Phi(\mathbf{x}_i) \cdot \Phi(\mathbf{x}) + b \right)
\]

Kernels:

\[
\Phi(\mathbf{x}) \cdot \Phi(\mathbf{x}') = k(\mathbf{x}, \mathbf{x}') = \begin{cases}
\exp \left( - \frac{\| \mathbf{x} - \mathbf{x}' \|^2}{2 \sigma^2} \right) \\
\tanh(\kappa (\mathbf{x} \cdot \mathbf{x}') + \Theta) \\
(\mathbf{x} \cdot \mathbf{x}' + 1)^p
\end{cases}
\]
Wrap-up

\[ \min_{0 \leq \alpha_i \leq C} W(\alpha_i, b) = \frac{1}{2} \sum_{i,j=1}^{M} \alpha_i \alpha_j y_i y_j K(\tilde{x}_i, \tilde{x}_j) - \sum_{i=1}^{M} \alpha_i + b \sum_{i=1}^{M} y_i \alpha_i \]

**KKT**

\[ \frac{\partial W(\alpha_i, b)}{\partial \alpha_i} = \sum_{j=1}^{M} y_i y_j K(\tilde{x}_i, \tilde{x}_j) \alpha_j + y_i b - 1 = y_i \tilde{f}(x_j) - 1 \]

\[ \frac{\partial W(\alpha_i, b)}{\partial b} = \sum_{j=1}^{M} y_j \alpha_j = 0 \]

where \( \tilde{f}(\tilde{x}) = \sum_{i=1}^{M} y_i \alpha_i K(\tilde{x}_i, \tilde{x}) + b \)
Application I

Recognition of Alternatively Spliced Exons in C.elegans

Task: Classification
Model: Support Vector Machines
Application: C.elegans Genes - Alternative Splicing
Splicing
Splicing

Splice sites are
- the exon/intron boundaries
- recognized by five snRNAs
- assembled in snRNPs
- flanked by regulatory elements

Spliceosomal Proteins
- interact with snRNPs and mRNA
- regulate recognition of splice sites
- can lead to alternative transcripts

One gene may correspond to several transcripts/proteins !!
Alternative Splicing

- Exon skipping
- Alternative 5' splice sites
- Alternative 3' splice sites
- Intron retention
- Mutually exclusive
Alternative Splicing

Alternative Splicing (AS)..

- can produce several mRNA transcript per gene
  (sometimes leading to more than 100 slightly different proteins)

- greatly increases the proteome diversity in eukaryotes
  (about 70% of human genes are alternatively spliced! )
Alternative Splicing

Methods for identifying alternative splicing ...

- usually need many EST sequences or
- exploit conservation between several organisms

Novel AS prediction method only using the pre-mRNA
Alternatively Spliced Exons

Idea: Use Machine Learning to

- understand differences between alternative and constitutive splicing
- exploit and identify regulative elements
- predict unknown alternative splicing events
**Alternatively Spliced Exons**

**Previous work**

*Analysis of conserved alternatively spliced exons*

(Sorek et al., Yeo et al. and others)
- consider conserved alternative spliced exons (ACE)
- exploit that ACE and flanking introns are more conserved between mouse and human

**Problem**

*only works for conserved exons*

*Derive the features from the “pre-mRNA” in order to find “novel” exons !!*
Task Formulation

Two-class Classification Problem

A (or B) is true splice site or not?

Use Support Vector Machines!
Remind the Procedure of Kernel Methods!

Data Set

<table>
<thead>
<tr>
<th>A1</th>
<th>A2</th>
<th>...</th>
<th>A5</th>
</tr>
</thead>
<tbody>
<tr>
<td>x1</td>
<td>10</td>
<td>5</td>
<td>...</td>
</tr>
<tr>
<td>x2</td>
<td>6</td>
<td>6</td>
<td>...</td>
</tr>
<tr>
<td>x3</td>
<td>7</td>
<td>7</td>
<td>...</td>
</tr>
</tbody>
</table>
| ...| ...| ... | ...| ...
| x10| 3  | 88  | ...| 700|

Kernel Function

\[ K(x_i, x_j) \]

Modeling

SVM

Kernel Matrix: \( K \)

Model Output

\( f(x) \)
**Procedure - Data Set**

Window of \( \approx 100 \) nucleotides

True sites \((y=1)\): fixed window around a true splice site

Decoys sites \((y=-1)\): generated by shifting the window

---

HyunJung (Helen) Shin, Max Planck Society, European School of Genetic Medicine, 03. 2006
Procedure - Data Set

Data Set – Strings

Modeling

Model

Kernel Function

Kernel Matrix: \( K \)

\[ K(S_i, S_j) \]

\[ f(x) \]
Procedure - Kernel Function (Matrix)

Kernels measure similarities between sequences
Weighted Degree Kernel (Sonnenburg et al., 2002)

Given two sequences $S_1$ and $S_2$ of equal length, the kernel consists of a weighted sum to which each match in the sequences makes a contribution.

The longer matches contribute more significantly.
Procedure - Kernel Function (Matrix)

Data Set – Strings

Kernel Function

Modeling

Kernel Matrix: $K$

Model Output

$f(x)$
Procedure - Modeling & Output

HyunJung (Helen) Shin, Max Planck Society, European School of Genetic Medicine, 03. 2006
**Procedure - Modeling & Output**

Data Set – Strings

Modeling -- SVM Model

Kernel Function

Kernel Matrix: K
Results

Exons Known

- 21,000 exons and 28,000 introns (single EST confirmed)
Results

280 AS spliced exons (total)
- ~1% of known exons are alternatively spliced (AS)
- ~0.25% of AS exons are yet completely unknown

RT-PCR with primers in flanking exons
(25 random exons & introns from 1-2% top ranks)
- 13 confirmed by RT-PCR

Additional 80 AS exons can be found with less than
200 additional RT-PCRs
The Most Up-To-Date Models

Kernel Methods

Support Vector Machines (SVM), kPCA, kCCA, kICA, etc

Semi-Supervised Learning Methods

Graph-based SSL, Transductive Inference Methods

* Note that, currently, the term of “Semi-Supervised Learning” has been used as a name of “model” as well as the concept of “learning scheme”
Semi-Supervised Learning Methods:
Graph-Based SSL
Semi-Supervised Learning

Semi-Supervised Learning utilizes every possible information in hand (known + unknown), therefore enhances prediction accuracy of a model.

<table>
<thead>
<tr>
<th>Supervised</th>
<th>Semi-Supervised</th>
</tr>
</thead>
<tbody>
<tr>
<td>Known</td>
<td>Known</td>
</tr>
<tr>
<td>Unknown</td>
<td>Unknown</td>
</tr>
</tbody>
</table>

### Supervised

<table>
<thead>
<tr>
<th>$X_1$</th>
<th>$A_1$</th>
<th>$A_2$</th>
<th>$A_3$</th>
<th>...</th>
<th>$A_{10}$</th>
<th>$y$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$X_1$</td>
<td>10</td>
<td>5</td>
<td>red</td>
<td>...</td>
<td>1000</td>
<td>1</td>
</tr>
<tr>
<td>$X_2$</td>
<td>6</td>
<td>6</td>
<td>blue</td>
<td>...</td>
<td>3500</td>
<td>20</td>
</tr>
<tr>
<td>$X_3$</td>
<td>7</td>
<td>7</td>
<td>yellow</td>
<td>...</td>
<td>400</td>
<td>45</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>$X_{10}$</td>
<td>3</td>
<td>56</td>
<td>red</td>
<td>...</td>
<td>0</td>
<td>30</td>
</tr>
<tr>
<td>$X_{20}$</td>
<td>15</td>
<td>62</td>
<td>red</td>
<td>...</td>
<td>500</td>
<td>100</td>
</tr>
<tr>
<td>$X_{30}$</td>
<td>3</td>
<td>88</td>
<td>blue</td>
<td>...</td>
<td>700</td>
<td>3</td>
</tr>
<tr>
<td>$X_{40}$</td>
<td>5</td>
<td>42</td>
<td>red</td>
<td>...</td>
<td>560</td>
<td>?</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>?</td>
</tr>
<tr>
<td>$X_{50}$</td>
<td>25</td>
<td>56</td>
<td>blue</td>
<td>...</td>
<td>600</td>
<td>?</td>
</tr>
</tbody>
</table>

### Semi-Supervised

<table>
<thead>
<tr>
<th>$X_1$</th>
<th>$A_1$</th>
<th>$A_2$</th>
<th>$A_3$</th>
<th>...</th>
<th>$A_{10}$</th>
<th>$y$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$X_1$</td>
<td>10</td>
<td>5</td>
<td>red</td>
<td>...</td>
<td>1000</td>
<td>1</td>
</tr>
<tr>
<td>$X_2$</td>
<td>6</td>
<td>6</td>
<td>blue</td>
<td>...</td>
<td>3500</td>
<td>20</td>
</tr>
<tr>
<td>$X_3$</td>
<td>7</td>
<td>7</td>
<td>yellow</td>
<td>...</td>
<td>400</td>
<td>45</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>$X_{10}$</td>
<td>3</td>
<td>56</td>
<td>red</td>
<td>...</td>
<td>0</td>
<td>30</td>
</tr>
<tr>
<td>$X_{20}$</td>
<td>15</td>
<td>62</td>
<td>red</td>
<td>...</td>
<td>500</td>
<td>100</td>
</tr>
<tr>
<td>$X_{30}$</td>
<td>3</td>
<td>88</td>
<td>blue</td>
<td>...</td>
<td>700</td>
<td>3</td>
</tr>
<tr>
<td>$X_{40}$</td>
<td>5</td>
<td>42</td>
<td>red</td>
<td>...</td>
<td>560</td>
<td>?</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>?</td>
</tr>
<tr>
<td>$X_{50}$</td>
<td>25</td>
<td>56</td>
<td>blue</td>
<td>...</td>
<td>600</td>
<td>?</td>
</tr>
</tbody>
</table>
Semi-Supervised Learning with a Single Graph
Semi-Supervised Learning with a Single Graph

- **Adjacency (similarity) matrix of the network:** $W$
- **Known Labels:** $y_1, ..., y_l \in \{-1, 1\}$
- **Unknown Labels:** $y_{l+1}, ..., y_n \in \{0\}$
- **Predicted outputs:** $f_1, ..., f_n$

  $f_i$ should be close to those of adjacent nodes, $f_j$'s where $i \sim j$.
  $f_i$ should be close to the given label $y_i$ at training nodes
Semi-Supervised Learning with a Single Graph

Learning Problem

$$\min \quad \mu \sum_{i \sim j} w_{ij} (f_i - f_j)^2 + \sum_i (f_i - y_i)^2$$

Equivalent Vector Form

$$\min \quad \mu \mathbf{f}^T \mathbf{L} \mathbf{f} + (\mathbf{f} - \mathbf{y})^T (\mathbf{f} - \mathbf{y})$$

$L$ is called the graph Laplacian matrix where

$$L = D - W, \quad D = \text{diag}(d_i), \quad d_i = \sum_j w_{ij}$$
Semi-Supervised Learning with a Single Graph

**Objective Function**

\[
\min_{f} \quad \mu f^{T} L f + (f - y)^{T} (f - y)
\]

**Solution**

\[
f = \left( I + \mu L \right)^{-1} y
\]
Application II

Functional Class Prediction with Multiple Networks

Task: Classification, Data Integration
Model: Semi-Supervised Learning
Application: Yeast Protein: Protein Function Prediction
Functional Class Prediction on a Protein Network

Proteins : Nodes
**Functional Class Prediction on a Protein Network**

**Functional Classes of Proteins**: Labeled / Unlabeled Nodes

- **unknown**
- **known**
- ?

-1  
known

<table>
<thead>
<tr>
<th>+1</th>
</tr>
</thead>
<tbody>
<tr>
<td>+1</td>
</tr>
</tbody>
</table>

+1/-1 : Labeled proteins with/without a specific function

? : Unlabeled proteins
**Functional Class Prediction on a Protein Network**

**Similarities between Proteins**

- Edges

- Edges in Physical Interaction Network: Two proteins physically interact (e.g., docking)
- Edges in Metabolic Network of Enzymes: Two enzymes catalyzing successive reactions
Functional Class Prediction on a Protein Network

The task is to predict labels of unlabeled proteins using similarities.
Graph Representation on Biological Networks

Example: Metabolic Gene Network
Graph Representation on Biological Networks

The first three reactions of the Glycolysis pathway, together with the catalyzing enzymes in the Yeast *S. cerevisiae*. 
Graph Representation on Biological Networks

Fructose-6P \rightarrow Fructose-1, 6P2

Substrate of PFK1

Enzyme or Protein, PFK1

Product of PFK1
Graph Representation on Biological Networks

Glucose

Glucose-1P

Glucose

Glucose-6P

Fructose-6P

Fructose-1, 6P2

GAL10

HKA

HKB

GLK1

PGM1

PGM2

PGT1

FBP1

PFK1

PFK2

FBA1

HyunJung (Helen) Shin, Max Planck Society, European School of Genetic Medicine, 03. 2006
Graph Representation on Biological Networks

How to Make a Graph?

Glucose

Glucose-1P

Glucose-6P

Fructose-1, 6P2

HyunJung (Helen) Shin, Max Planck Society, European School of Genetic Medicine, 03. 2006
Graph Representation on Biological Networks

Glucose

Glucose-1P

GAL10

HKA

HKB

GLK1

PGM1

PGM2

Glucose-6P

PGT1

Fructose-6P

Fructose-1, 6P2

FBP1

PFK1

PFK2

FBA1

A Node corresponds to a Protein

HyunJung (Helen) Shin, Max Planck Society, European School of Genetic Medicine, 03. 2006
An Edge is made if the Product of the First protein is the Substrate of the Second one.
An Edge is made if the Product of the First protein is the Substrate of the Second one.
Graph Representation on Biological Networks

An **Edge** is made if the **Product of the First** protein is the **Substrate of the Second one**

HyunJung (Helen) Shin, Max Planck Society, European School of Genetic Medicine, 03. 2006
Graph Representation on Biological Networks
**Semi-Supervised Learning with a Single Graph**

**Objective Function**

\[ \min_{\mathbf{f}} \quad \mu \mathbf{f}^T \mathbf{L} \mathbf{f} + (\mathbf{f} - \mathbf{y})^T (\mathbf{f} - \mathbf{y}) \]

**Solution**

\[ \mathbf{f} = \left\{ \mathbf{I} + \mu \mathbf{L} \right\}^{-1} \mathbf{y} \]
If Multiple Graphs are Given?
If Multiple Graphs are Given?

$G_1$

$G_2$

$G_3$

$G_K$
If Multiple Graphs are Given?

Protein-protein interactions

Cell cycle gene expression measurements

Genetic interactions

Co-participation in a protein complex

HyunJung (Helen) Shin, Max Planck Society, European School of Genetic Medicine, 03. 2006
If Multiple Graphs are Given?

Each graph can solely predict the label of the unlabeled nodes depending on its own similarity.
If Multiple Graphs are Given?

Since different graphs contain partly independent and partly complementary pieces of information about the problem at hand, one thus can enhance the total information about the problem by combining those graphs.
If Multiple Graphs are Given?

Example: Multiple Graph Sources on Proteins

Physical interactions of the proteins

Gene regulatory relationships
  [Lee et al., 2002, Ihmels et al., 2002, Segal et al., 2003]

Edges in a metabolic pathway [Kanehisa et al., 2004]

Similarities between protein sequences [Yona et al., 1999]

etc.
If Multiple Graphs are Given?

Lee et. al., 2004.
A Probabilistic Functional Network of Yeast Genes,
Science, vol. 306
"a label of an unlabeled node is more likely to be that of more adjacent or more strongly connected node to it."

"a label of an unlabeled node is more likely to be that of more adjacent or more strongly connected node to it."

Since different graphs contain partly independent and partly complementary pieces of information about the problem at hand,

one thus can enhance the total information about the problem by combining those graphs.
If Multiple Graphs are Given?

**Previous Approach**

**SDP/SVM**: Semi-Definite Programming based Support Vector Machine

[Lanckriet et al., Bioinformatics, 2004]
SDP/SVM (Kernel Method)

Diffusion Kernel

Each graph is converted to a kernel matrix

SDP

Kernel matrices are combined with weights which are automatically learned by Semi-Definite Programming

SVM

Labels are predicted based on the combined kernel matrix

\[
K(\mu) = \mu_1 K_1 + \mu_2 K_2 + \mu_3 K_3 + \cdots + \mu_k K_k
\]
SDP/SVM (Kernel Method)

<table>
<thead>
<tr>
<th>Diffusion Kernel</th>
<th>Each graph is converted to a kernel matrix</th>
</tr>
</thead>
<tbody>
<tr>
<td>SDP</td>
<td>Kernel matrices are combined with weights which are automatically learned by Semi-Definite Programming</td>
</tr>
<tr>
<td>SVM</td>
<td>Labels are predicted based on the combined kernel matrix</td>
</tr>
</tbody>
</table>

**Good accuracy**

*which is much better than Markov Random Field*

**But**

*Very Slow*
**SDP/SVM (Kernel Method)**

**SDP/SVM : Semi-Definite Programming based SVM**

In SDP/SVM, multiple kernel matrices corresponding to each of data sources are combined with weights obtained by solving an SDP.

However, when trying to apply SDP/SVM to large problems, the computational cost can become prohibitive, since both Converting the data to a kernel matrix for the SVM and Solving the SDP are time and memory demanding.
**SDP/SVM (Kernel Method)**

**Diffusion Kernel**

\[
K_\beta = e^{\beta L} = \lim_{s \to \infty} (I + \frac{\beta L}{s})^s = I + \beta L + \frac{\beta^2}{2} L^2 + \frac{\beta^3}{6} L^3 + \ldots
\]

\(L\) : graph Laplacian.

\(\beta\) : diffusion rate

[Kondor and Lafferty, 2002].
**SDP/SVM (Kernel Method)**

**Semi-Definite Programming**

[Vandenberg and Boyd, 1996]
[Boyd and Vandenberg, 2003]

\[
\min_{u} \quad c^T u \\
\text{s.t.} \quad F^j(u) = F_o + \sum_{k=1}^{K} u_k F_k \geq 0, \quad j = 1, \ldots, J.
\]

where \( c \in \mathbb{R}^K \), \( F_k \in \mathbb{R}^{n \times n} \), \( F(u) \in \mathbb{R}^{n \times n} \) : symmetric, positive semi-definite

*Convex optimization problem since its objective and constrains are convex*
**SDP/SVM (Kernel Method)**

**SDP/SVM**

**SVM dual problem**

\[
\begin{align*}
\max_{\alpha} & \quad 2\alpha^T e - \alpha^T (G(K) + \tau I)\alpha \\
\text{s.t.} & \quad 0 \leq \alpha \leq C, \quad \alpha^T y = 0, \\
& \quad \text{trace}(K) = c.
\end{align*}
\]

where \( G_{ij}(K) = k(x_i, x_j) y_i y_j \)

**SVM cast as an SDP**

\[
\begin{align*}
\min_{K, t, \lambda, v, \delta} & \quad t \\
\text{s.t.} & \quad \text{trace}(K) = c, \\
& \quad \left( G(K_{tr}) + \tau I_{ntr} \quad e + v - \delta + \lambda y \right)^T \\
& \quad \left( (e + v - \delta + \lambda y)^T \quad t - 2C\delta^T e \right) \geq 0, \\
& \quad v \geq 0, \\
& \quad \delta \geq 0.
\end{align*}
\]

[Lanckriet et al., 2004]
**SDP/SVM (Kernel Method)**

**[SDP/SVM]**

SVM cast as an SDP

\[
\begin{align*}
\min & \quad t \\
\text{s.t.} & \quad \text{trace}\left(\sum_{i=1}^{m} \mu_i K_i\right) = c, \\
& \quad \sum_{i=1}^{m} \mu_i K_i \geq 0, \\
& \quad \left\{ G\left(\sum_{i=1}^{m} \mu_i K_{i,\text{tr}}\right) + \tau I_{ntr} - e + v - \delta + \lambda y \right\} \geq 0, \\
& \quad (e + v - \delta + \lambda y)^T t - 2C\delta^T e \geq 0, \\
& \quad v \geq 0, \\
& \quad \delta \geq 0.
\end{align*}
\]

**Multiple Kernels**

*HyunJung (Helen) Shin, Max Planck Society, European School of Genetic Medicine, 03. 2006*
SDP/SVM (Kernel Method)

Calculating a Diffusion Kernel from a Graph

\[ O(n^3), \quad A \text{ dense matrix of } n \times n \]

Solving SDP

\[ O((m+n)^2n^{2.5}) \]

\[ m: \text{ the number of kernel matrices} \]
\[ n: \text{ number of nodes (data)} \]

Computationally Expensive both in Time and Memory
Why not use a more direct approach for combining graphs based on significant progress of graph-based semi-supervised learning methods?

- Zhou et al., 2004
- Belkin and Niyogi, 2003
- Zhu et al., 2003
- Chapelle et al., 2003
Semi-Supervised Learning Extension to Multiple Graphs

• Combining weights are automatically assigned to Graphs
• Comparable Accuracy to SDP/SVM
• Very Fast
Extension to Multiple Graphs

Linear Combination of Laplacians

\[ L(\beta) = \sum_{k=1}^{K} \beta_k L_k \]

How to Find Combining Weights?
Extension to Multiple Graphs

Single Graph

$$\min_f (f - y)^T (f - y) + c f^T L f$$

Without loss of generality, the problem is rewritten by penalizing the upper-bound

$$\min_{f, \gamma} (f - y)^T (f - y) + c\gamma, \quad f^T L f \leq \gamma.$$
Extension to Multiple Graphs

**Multiple Graphs**

\[
\min_{f} \quad (f - y)^T (f - y) + c \{ \beta_1 f^T L_1 f + \beta_2 f^T L_2 f + \ldots + \beta_k f^T L_k f \}
\]

*Without loss of generality, the problem is rewritten by penalizing the upper-bound*

\[
\min_{f, \gamma} \quad (f - y)^T (f - y) + c \gamma, \quad f^T L_k f \leq \gamma, \quad k = 1, \ldots, K.
\]
Extension to Multiple Graphs: Optimization

**Primal**

\[
\min_{\mathbf{f}, \gamma} \quad (\mathbf{f} - \mathbf{y})^T (\mathbf{f} - \mathbf{y}) + c\gamma,
\]

s.t. \( \mathbf{f}^T \mathbf{L}_k \mathbf{f} \leq \gamma, \)
\( \gamma \geq 0, \quad k = 1, \ldots, K. \)

**Dual**

\[
\min_{\mathbf{\beta}} \quad d(\mathbf{\beta}) \equiv \mathbf{y}^T (\mathbf{I} + \sum_{k=1}^{K} \mathbf{\beta}_k \mathbf{L}_k)^{-1} \mathbf{y},
\]

\[
\text{s.t.} \quad \sum_{k=1}^{K} \mathbf{\beta}_k \leq c
\]

\( \mathbf{\beta}_k : \text{Weight for Network } k, \quad \text{Lagrange Multiplier.} \)
Extension to Multiple Graphs: Solution

\[ f = \left( I + \sum_{k=1}^{K} \beta_k L_k \right)^{-1} y \]

\( \beta_k \): Weight for Network \( k \), Lagrange Multiplier.

Sparse Linear System

\[ y = \left( I + \sum_{k=1}^{K} \beta_k L_k \right) f \]

Linear Systems
Extension to Multiple Graphs: Meaning of Weights

By KKT complementarity condition, we have the following relationship at the optimal solution,

$$\beta_k (f^T L_k f - \delta) = 0$$

$$\beta_k = 0 \quad \text{iff} \quad f^T L_k f < \delta$$

$$\beta_k > 0 \quad \text{iff} \quad f^T L_k f = \delta$$

The score vector $f$ would not be changed much with those graphs, thus those are considered as redundant.

Those graphs are considered important.
Extension to Multiple Graphs

Computational Efficiency

1. Repetition of an Identical Form of Inverse Matrix
2. Implicit Calculation of Matrix Inversion
Extension to Multiple Graphs

Computational Efficiency

1. Repetition of an Identical Form of Inverse Matrix:
in the objective function and the derivative, (and the network output).

**Objective Function**

\[
\min_{\beta} \quad d(\beta) = y^T (I + \sum_{k=1}^{K} \beta_k L_k)^{-1} y
\]

**Solution Update**

\[
\frac{\partial d}{\partial \beta_k} = -y^T (I + \sum_{j=1}^{K} \beta_j L_j)^{-1} L_k (I + \sum_{j=1}^{K} \beta_j L_j)^{-1} y
\]

**Network Output**

\[
f = \left\{ I + \sum_{k=1}^{K} \beta_k L_k \right\}^{-1} y
\]
Extension to Multiple Graphs

**Computational Efficiency**

2. Implicit Calculation of Matrix Inversion:
The solution can be obtained by solving the “sparse linear systems.”
Therefore, computational cost is nearly linear in the number of non-zero entries of $\sum_{k=1}^{K} \beta_k L_k$ – (Spielman and Teng, 2004).

**Matrix Inversion**

$$f = \left\{ I + \sum_{k=1}^{K} \beta_k L_k \right\}^{-1} y$$

**Linear Systems**

$$y = \left\{ I + \sum_{k=1}^{K} \beta_k L_k \right\} f$$
Function Prediction Experiments

MIPS Comprehensive Yeast Genome Database (CYGD-mips.gsf.de/proj/yeast).

- **Data**: 3588 yeast proteins
- **Output**: 13 functional categories
  - Binary classification for each category
- **Input**: 5 networks
- **Setting**: 5 fold cross validation
  - 5 times repetition
Protein Functional Categories

MIPS Comprehensive Yeast Genome Database (CYGD-mips.gsf.de/proj/yeast).

13 CYGD functional Classes

1. metabolism
2. energy
3. cell cycle and DNA processing
4. transcription
5. protein synthesis
6. protein fate
7. cellular transportation and transportation mechanism
8. cell rescue, defense and virulence
9. interaction with cell environment
10. cell fate
11. control of cell organization
12. transport facilitation
13. others
Inputs (5 networks)

Network created from **Pfam domain structure**. A protein is represented by a 4950-dimensional binary vector, in which each bit represents the presence or absence of one Pfam domain. An edge is created if the inner product between two vectors exceeds 0.06. The edge weight corresponds to the inner product.

\[ W_1 \]

**Co-participation in a protein complex** (determined by tandem affinity purification, TAP). An edge is created if there is a bait-prey relationship between two proteins.

\[ W_2 \]

**Protein-protein interactions** (MIPS physical interactions)

\[ W_3 \]

**Genetic interactions** (MIPS genetic interactions)

\[ W_4 \]

Network created from the **cell cycle gene expression measurements** [Spellman et al., 1998]. An edge is created if the Pearson coefficient of two profiles exceeds 0.8. The edge weight is set to 1. This is identical with the network used in [Deng et al., 2003]

\[ W_5 \]
Network created from **Pfam domain structure**. A protein is represented by a 4950-dimensional binary vector, in which each bit represents the presence or absence of one Pfam domain. An edge is created if the inner product between two vectors exceeds 0.06. The edge weight corresponds to the inner product.
Co-participation in a protein complex (determined by tandem affinity purification, TAP). An edge is created if there is a bait-prey relationship between two proteins.
$W_3$ **Protein-protein interactions** (MIPS physical interactions)
Inputs (5 networks)

\[ W_4 \textbf{Genetic interactions} \text{ (MIPS genetic interactions)} \]
Network created from the **cell cycle gene expression measurements** [Spellman et al., 1998]. An edge is created if the Pearson coefficient of two profiles exceeds 0.8. The edge weight is set to 1. This is identical with the network used in [Deng et al., 2003]
**Inputs (5 networks)**

<table>
<thead>
<tr>
<th>Density of Laplacians (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>0.7805</strong></td>
</tr>
<tr>
<td><strong>0.0570</strong></td>
</tr>
<tr>
<td><strong>0.0565</strong></td>
</tr>
<tr>
<td><strong>0.0435</strong></td>
</tr>
<tr>
<td><strong>0.0919</strong></td>
</tr>
</tbody>
</table>

Network created from Pfam domain structure. A protein is represented by a 4950-dimensional binary vector, in which each bit represents the presence or absence of one Pfam domain. An edge is created if the inner product between two vectors exceeds 0.06. The edge weight corresponds to the inner product.

Co-participation in a protein complex (determined by tandem affinity purification, TAP). An edge is created if there is a bait-prey relationship between two proteins.

Protein-protein interactions (MIPS physical interactions)

Genetic interactions (MIPS genetic interactions)

Network created from the cell cycle gene expression measurements [Spellman et al., 1998]. An edge is created if the Pearson coefficient of two profiles exceeds 0.8. The edge weight is set to 1. This is identical with the network used in [Deng et al., 2003].
Density of Working Matrices

**Given Graphs**

**SDP/SVM**

Kernel matrix

**Dense**

\[ K(\mu) = \mu_1 K_1 + \mu_2 K_2 + \mu_3 K_3 + \cdots + \mu_k K_k \]

**SSL**

Laplacian matrix \( L \) (or Similarity matrix \( W \))

**Sparse**

\[ L(\beta) = \beta_1 L_1 + \beta_2 L_2 + \beta_3 L_3 + \cdots + \beta_k L_k \]
**Inputs (5 networks)**

- **Network created from Pfam domain structure.** A protein is represented by a 4950-dimensional binary vector, in which each bit represents the presence or absence of one Pfam domain. An edge is created if the inner product between two vectors exceeds 0.06. The edge weight corresponds to the inner product.

  - $W_1$: 0.7805

- **Co-participation in a protein complex (determined by tandem affinity purification, TAP).** An edge is created if there is a bait-prey relationship between two proteins.

  - $W_2$: 0.0570

- **Protein-protein interactions (MIPS physical interactions).**

  - $W_3$: 0.0565

- **Genetic interactions (MIPS genetic interactions).**

  - $W_4$: 0.0435

- **Network created from the cell cycle gene expression measurements [Spellman et al., 1998].** An edge is created if the Pearson coefficient of two profiles exceeds 0.8. The edge weight is set to 1. This is identical with the network used in [Deng et al., 2003].

  - $W_5$: 0.0919

**Density of Laplacians (%)**

- 1/0.0078 = 128

**Memory Saving Ratio (%) against Kernels**

- 1754

- 1770

- 2298

- 1088
Methods in Comparison

\( L_k \)  
Label propagation with an Individual Graphs \((k=1 \ldots 5)\)

\( L_{\text{opt}} \)  
Laplacian of Combined Graph with Optimized Weights

\( L_{\text{fix}} \)  
Label propagation with Equal Weights

\( MRF \)  
Markov Random Field, proposed by Deng et al [2003]

\( SDP/SVM \)  
Semi-definite Programming based Support Vector Machines, proposed by Lanckriet et al [2004]
Measurements

$ROC \ (receiver \ operating \ characteristic) \ score$

$TP1FP, TP10FP$

Computational Time
Measurements

**ROC score**

The area under ROC curve that plots true positive rate as a function of false positive rate for differing classification thresholds.

It measures the **overall quality of the ranking** induced by the classifier, rather than the **quality of a single value of threshold** in that ranking.
**Measurements**

**ROC score**

The closer the curve follows the left-hand border and then the top-border of the ROC space, the more accurate the classifier.

**True Positive Rate (sensitivity)**

**False Positive Rate (1-specificity)**
**Measurements**

**TP10FP**

TP10FP is the rate of true positives at the point that yields 10% false positive rate on the ROC curve.
Results: ROC scores of $L_{\text{opt}}$, $L_{\text{fix}}$ vs. the Best Performing Individual $L_k$

White: the best performing individual network
Blue: $L_{\text{fix}}$
Black: $L_{\text{opt}}$

Across the 13 classes, $L_{\text{fix}}$ or $L_{\text{opt}}$ outperforms the best performing individual.
Results: TP1FP and TP10FP of $L_{opt}, L_{fix}$ vs. Individual $L_k$'s

**TP1FP (%)**

<table>
<thead>
<tr>
<th>$L_1$</th>
<th>$L_2$</th>
<th>$L_3$</th>
<th>$L_4$</th>
<th>$L_5$</th>
<th>$L_{fix}$</th>
<th>$L_{opt}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>30</td>
<td>20</td>
<td>25</td>
<td>35</td>
<td>40</td>
<td>50</td>
<td>60</td>
</tr>
</tbody>
</table>

**TP10FP (%)**

<table>
<thead>
<tr>
<th>$L_1$</th>
<th>$L_2$</th>
<th>$L_3$</th>
<th>$L_4$</th>
<th>$L_5$</th>
<th>$L_{fix}$</th>
<th>$L_{opt}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>70</td>
<td>60</td>
<td>50</td>
<td>40</td>
<td>30</td>
<td>20</td>
<td>10</td>
</tr>
</tbody>
</table>
Results – McNemar’s Test:

A pairwise test for ROC score difference: the combined graph vs. individual graphs

![Graph showing McNemar p-value distribution](image-url)
Results – McNemar’s Test:

A smaller p-value indicates a more statistically significant difference
Results – McNemar’s Test:

In 61% of the total number of trials, there is a statistically significant difference (at a significance level of alpha=0.05).
Results: Obtained Weights

- Others
- Transport Facilitation
- Cell Organization
- Cell Fate
- Interaction with Environment
- Cell Rescue
- Transportation
- Protein Fate
- Protein Synthesis
- Transcription
- Cell Cycle
- Energy
- Metabolism

HyunJung (Helen) Shin, Max Planck Society, European School of Genetic Medicine, 03. 2006
Results: Comparison between Methods

For most classes, the proposed method achieves high scores, which are similar to the SDP/SVM methods. In classes 11 and 13, the proposed method performs poor (but still better than the MRF method). However, taking into account the Simplicity and Efficiency the method shows the promising results.
**Results : Computational Time**

**Average Computation Time**

**Combining Graphs**
- Fixed Weights : 1.41 seconds (std. 0.013)
- Optimized Weights : 49.3 seconds (std. 14.8)

**SDP/SVM** :
- Approx. Several CPU days
  (G. Lanckriet, personal communication)

*Measured in a standard 2.2Ghz PC with 1GByte memory*
Results : Computational Time

Average Computation Time

Combining Graphs: Nearly linearly proportional to the number of non-zero entries of sparse matrices

SDP/SVM: \( O(n^3) + O((m+n)^2n^{2.5}) \)
Results: Summary

Combining Graphs with **Optimized Weights** has **“MORE”**

**Selectivity**

When Compared with Combining Graphs with **Fixed Weights**

Combining Graphs has **“MORE”**

**Simplicity, Computational Efficiency, thus Scalability**

When Compared with **SDP/SVM**
Results : Summary

Combining Graphs with **Optimized Weights** is **“LESS”**

*Simple*

When Compared with Combining Graphs with **Fixed Weights**

Combining Graphs is **“LESS”**

*Accurate*

When Compared with **SDP/SVM**
Results: Summary

Semi-Supervised Learning with Multiple Networks

• ...is Fast and Scalable

• ...provides Selectivity
  (redundant / irrelevant networks can be excluded)
For Further Information...

Multivariate Statistical Methods

R.A. Johnson & D.W. Wichern,
Applied multivariate statistical analysis,

B.F.J. Manly,
Multivariate statistical methods: A primer,

Kernel Methods and SVM

V. Vapnik,
Statistical learning theory,
Wiley, NY, 1998
For Further Information...

Kernel Methods and SVM

C.J.C. Burges,
A tutorial on support vector machines for pattern recognition,
Data Mining and Knowledge Discovery, 1998.

N. Cristianini and J. Shawe-Taylor,
An introduction to support vector machines,

B. Scholkopf and A. J. Smola,
Learning with Kernels,
For Further Information...

**Kernel Methods and Bioinformatics**
B. Scholkopf, K. Tsuda and J-P. Vert,
Kernel Methods in Computational Biology,

**Semi-supervised Learning**
Olivier Chapelle, Bernhard Schoelkopf and Alexander Zien,
Semi-Supervised Learning,
MIT press, 2005
## For Further Information…

### Application I: Alternative Splicing

G. Rätsch, S. Sonnenburg and B. Schölkopf,
A RASE: Recognition of Alternatively Spliced Exons in *C. elegans*,
http://www.fml.tuebingen.mpg.de/raetsch/projects/RASE

### Application II: Protein Function Classification

H. Shin and K. Tsuda,
Prediction of Protein Function from Networks,
http://www.kyb.tuebingen.mpg.de/~shin
http://www.fml.tuebingen.mpg.de/~shin

K. Tsuda, H. Shin, and B. Schölkopf,
Fast Protein Classification with Multiple Networks,
Bioinformatics, 2005.