Lecture 7

Gene expression analysis: Clustering and Classification
Module II: Gene expression analysis and networks

- **Computational foundations:**
  - Unsupervised Learning: Expectation Maximization
  - Supervised learning: generative/discriminative models
  - Read mapping, significance testing, splice graphs
  - Folding: DP self-alignment, Context Free grammars

- **Biological frontiers:**
  - L6: RNA-Seq analysis, quantifying transcripts, isoforms
  - L7: Gene expression analysis: cluster genes/conditions
  - L8: Networks I: Bayesian Inference, deep learning
  - L9: Networks II: Network structure, spectral methods
Today: Gene Expression Clustering & Classification

1. Introduction to gene expression analysis
   - Technology: microarrays vs. RNAseq. Resulting data matrices
   - Supervised (Clustering) vs. unsupervised (classification) learning

2. K-means clustering (clustering by partitioning)

3. Hierarchical Clustering (clustering by agglomeration)
   - Basic algorithm, Distance measures. Evaluating clustering results

4. Naïve Bayes classification (generative approach to classification)
   - Discriminant function: class priors, and class-conditional distributions
   - Training and testing, Combine mult features, Classification in practice

5. (optional) Support Vector Machines (discriminative approach)
   - SVM formulation, Margin maximization, Finding the support vectors
   - Non-linear discrimination, Kernel functions, SVMs in practice
RNA-Seq: De novo tx reconstruction / quantification

**Microarray technology**
- Synthesize DNA probe array, complementary hybridization
- Variations:
  - One long probe per gene
  - Many short probes per gene
  - Tiled k-mers across genome
- Advantage:
  - Can focus on small regions, even if few molecules / cell

**RNA-Seq technology:**
- Sequence short reads from mRNA, map to genome
- Variations:
  - Count reads mapping to each known gene
  - Reconstruct transcriptome *de novo* in each experiment
- Advantage:
  - Digital measurements, *de novo*
Expression Analysis Data Matrix

- Measure 20,000 genes in 100s of conditions

- Study resulting matrix

Each experiment measures expression of thousands of ‘spots’, typically genes

Expression profile of a gene

Experiment similarity questions

Gene similarity questions

m genes

n experiments

5
Clustering vs. Classification

**Goal of Clustering:** Group similar items that likely come from the same category, and in doing so **reveal hidden structure**
- **Unsupervised learning**

**Goal of Classification:** Extract features from the data that best **assign new elements** to ≥1 of **well-defined classes**
- **Supervised learning**

---

Clustering vs Classification

- **Objects** characterized by one or more features

- **Classification (supervised learning)**
  - Have **labels** for some points
  - Want a “rule” that will accurately assign labels to new points
  - Sub-problem: Feature selection
  - Metric: Classification accuracy

- **Clustering (unsupervised learning)**
  - No labels
  - Group points into clusters based on how “near” they are to one another
  - Identify **structure** in data
  - Metric: independent validation features
Two approaches to clustering

• Partitioning (e.g. k-means)
  – Divides objects into non-overlapping clusters such that each data object is in exactly one subset

• Agglomerative (e.g. hierarchical clustering)
  – A set of nested clusters organized as a hierarchy
1. **Introduction to gene expression analysis**
   - Technology: microarrays vs. RNAseq. Resulting data matrices
   - Supervised (Clustering) vs. unsupervised (classification) learning

2. **K-means clustering (clustering by partitioning)**

3. **Hierarchical Clustering (clustering by agglomerating)**
   - Basic algorithm, Distance measures. Evaluating clustering results

4. **Naïve Bayes classification (generative approach to classification)**
   - Discriminant function: class priors, and class-conditional distributions
   - Training and testing, Combine mult features, Classification in practice

5. **(optional) Support Vector Machines (discriminative approach)**
   - SVM formulation, Margin maximization, Finding the support vectors
   - Non-linear discrimination, Kernel functions, SVMs in practice
K-Means Clustering

The Basic Idea

• Assume a fixed number $K$ of clusters
• Partition points into $K$ compact clusters

The Algorithm

• Initialize $K$ cluster centers randomly
• Repeatedly:
  – Assign points to nearest center
  – Move centers to center of gravity of their points
• Stop at convergence (no more reassignments)
K-Means Algorithm Example

- Randomly Initialize Clusters
- Assign data points to nearest clusters
- Recalculate cluster centers
- Repeat… until convergence
K-Means Algorithm Example

- Randomly Initialize Clusters
- Assign data points to nearest clusters
- Recalculate cluster centers
- Repeat… until convergence
K-Means Algorithm Example

- Randomly Initialize Clusters
- Assign data points to nearest clusters
- Recalculate cluster centers
- Repeat... until convergence
K-Means Algorithm Example

• Randomly Initialize Clusters
• Assign data points to nearest clusters
• Recalculate cluster centers
• Repeat… until convergence
K-Means Algorithm Example

- Randomly Initialize Clusters
- Assign data points to nearest clusters
- Recalculate cluster centers
- Repeat… until convergence
K-Means Algorithm Example

- Randomly Initialize Clusters
- Assign data points to nearest clusters
- Recalculate cluster centers
- Repeat… until convergence
K-means update rules

Re-assign each point $x_i$ to nearest center $k$

$\Rightarrow$ Minimize distance from $x_i$ to $\mu_k$:

$$d_{i,k} = (x_i - \mu_k)^2$$

(“M’’)

Update center $\mu_k$ to the mean of the points assigned to it:

$$\mu_k(n+1) = \sum_{x_i \text{ with label } j} \frac{x_i}{|X^k|}$$

where: $|X^k| = \#x_i$ with label k
K-means Optimality Criterion

We can think of K-means as trying to create clusters that minimize a cost criterion associated with the size of the cluster

$$\text{COST}(x_1, x_2, x_3, \ldots, x_n) = \sum_{\mu_k} \sum_{x_i \text{ with label } k} (x_i - \mu_k)^2$$

To achieve this, minimize each cluster term separately:

$$\sum_{x_i \text{ with label } k} (x_i - \mu_k)^2 = \sum_{x_i \text{ with label } k} x_i^2 - 2x_i \cdot u_k + u_k^2 = \sum x_i^2 - u_k \sum 2x_i + |x^k| \cdot u_k^2$$

Optimum $u_k = \sum_{x_i \text{ with label } k} \frac{x_i}{|x^k|}$, the centroid

However: Some points can be almost halfway between two centers $\Rightarrow$ Assign partial weights

Fuzzy K-means
**Fuzzy K-means update rule**

Re-assign each point $x_i$ to **all** centers, **weighted by distance**

For each point calculate the probability of membership for each category $K$:

$$P(label \ K \mid x_i, \mu_k)$$

Update center $\mu_k$ to the **weighted mean** of the points assigned to it:

$$\mu_k(n + 1) = \frac{\sum_{x_i \text{ with label } j} x_i \cdot P(\mu_k \mid x_i)^b}{\sum_{x_i \text{ with label } j} P(\mu_k \mid x_i)^b}$$

Regular K-Means is a special case of fuzzy $k$-means where:

$$P(label \ K \mid x_i, \mu_k) = \begin{cases} 
1 & \text{if } x_i \text{ is closest to } \mu_k \\
0 & \text{otherwise}
\end{cases}$$
Today: Gene Expression Clustering & Classification

1. Introduction to gene expression analysis
   – Technology: microarrays vs. RNAseq. Resulting data matrices
   – Supervised (Clustering) vs. unsupervised (classification) learning

2. K-means clustering (clustering by partitioning)

3. Hierarchical Clustering (clustering by agglomeration)
   – Basic algorithm, Distance measures. Evaluating clustering results

4. Naïve Bayes classification (generative approach to classification)
   – Discriminant function: class priors, and class-conditional distributions
   – Training and testing, Combine mult features, Classification in practice

5. (optional) Support Vector Machines (discriminative approach)
   – SVM formulation, Margin maximization, Finding the support vectors
   – Non-linear discrimination, Kernel functions, SVMs in practice
K-Means as a Generative Model

Model of $P(X,\text{Labels})$

Observations

Samples drawn from normal distributions with unit variance - a \textit{Gaussian Mixture Model}

$$P(x_i | u_j) = \frac{1}{\sqrt{2\pi}} \exp \left\{ -\frac{(x_i - u_j)^2}{2} \right\}$$

Given only samples, how do we estimate max lik model params: (1) centroid definitions, (2) point assignments?
EM solution: iteratively estimate one from the other

E step: If centers are known ➔ Estimate memberships
M step: If assignments known ➔ Compute centroids

Choose $\mu_k$ and labels that maximize $P(\text{data}|\text{model})$

Solution is exactly the k-means algorithm!
M step: assignments known \( \implies \) compute centroids

Choose \( \mu_k \) and labels that maximize \( P(\text{data}|\text{model}) \)

\[
\arg\max_{\mu_k} \left\{ \log \prod_{i} P \left( x_i \mid \mu \right) \right\} = \arg\max_{\mu} \sum_{i} \left\{ -\frac{1}{2} (x_i - u)^2 + \log \left( \frac{1}{\sqrt{2\pi}} \right) \right\}
\]

Seeking the max likelihood estimate of the cluster mean

\[
= \arg\min_{\mu} \sum_{i} (x_i - u)^2
\]

Solution is the centroid of the \( x_i \)

Equivalent

EM solution \( \iff \) K-means solution
E step: centers known $\Rightarrow$ Estimate memberships

Choose $\mu_k$ and labels that maximize $P(\text{data}|\text{model})$

$$\arg\max_k P_k(x_i | \mu_i) = \arg\max_k \frac{1}{\sqrt{2\pi}} \exp\left\{ -\frac{(x_i - u_k)^2}{2} \right\} = \arg\min_k \left( x_i - u_k \right)^2$$

Seeking the label $k$ that maximizes likelihood of point

Solution is the nearest center

EM solution $\leftrightarrow$ K-means solution

Equivalent
## Algorithmic vs. Machine Learning Formulations

<table>
<thead>
<tr>
<th></th>
<th>K-means</th>
<th>Fuzzy K-means</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><strong>Algorithmic formulation</strong></td>
<td><strong>Probabilistic interpretation</strong></td>
</tr>
<tr>
<td><strong>Initialization</strong></td>
<td>Initialize K centers $\mu_k$</td>
<td>Initialize model parameters</td>
</tr>
<tr>
<td></td>
<td>$\mu_k$</td>
<td>$\mu_k$</td>
</tr>
<tr>
<td><strong>E-step:</strong></td>
<td>Assign $x_i$ label of nearest center distance</td>
<td>Estimate most likely missing label given</td>
</tr>
<tr>
<td></td>
<td>$d_{i,k} = (x_i - \mu_k)^2$</td>
<td>previous parameters</td>
</tr>
<tr>
<td></td>
<td>$P(x_i</td>
<td>label K, \mu_k)$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>each point to each class $P(label K</td>
</tr>
<tr>
<td><strong>M-step:</strong></td>
<td>Move $\mu_k$ to centroid of all points with</td>
<td>Choose new max likelihood params given points</td>
</tr>
<tr>
<td></td>
<td>that label</td>
<td>in label</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Move $\mu_k$ to weighted centroid of all</td>
</tr>
<tr>
<td></td>
<td></td>
<td>points, each weighted by $P(label)$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Choose new params to maximize expected</td>
</tr>
<tr>
<td></td>
<td></td>
<td>expected likelihood given label estimates</td>
</tr>
<tr>
<td><strong>Iteration</strong></td>
<td>Iterate</td>
<td>Iterate</td>
</tr>
<tr>
<td></td>
<td>Iterate</td>
<td>Iterate</td>
</tr>
<tr>
<td></td>
<td>Iterate</td>
<td>Iterate</td>
</tr>
<tr>
<td></td>
<td>$P(x</td>
<td>\text{Model})$ <strong>guaranteed</strong> to increase</td>
</tr>
<tr>
<td></td>
<td>each iteration of EM algo</td>
<td></td>
</tr>
</tbody>
</table>

**Initialization:** Initialize $K$ centers $\mu_k$ and model parameters.

**E-step:** Assign $x_i$ label of nearest center distance $d_{i,k} = (x_i - \mu_k)^2$.

**M-step:** Move $\mu_k$ to centroid of all points with that label.

**Iteration:** Iterate until convergence.
EM is much more general than fuzzy K-means

<table>
<thead>
<tr>
<th>K-means solution</th>
<th>EM generalization</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cluster sizes</td>
<td>Uniform priors</td>
</tr>
<tr>
<td>Spread of points</td>
<td>Unit distance function</td>
</tr>
<tr>
<td>Cluster shape</td>
<td>Symmetric, x-y indpt</td>
</tr>
<tr>
<td>Label assignment</td>
<td>K-means: Pick max</td>
</tr>
<tr>
<td></td>
<td>Fuzzy: Full density</td>
</tr>
<tr>
<td></td>
<td>EM: Full density</td>
</tr>
<tr>
<td></td>
<td>Gibbs: sample posterior</td>
</tr>
</tbody>
</table>

Cluster priors: $P(class_i)$

Gaussian distribution: $\mu_i, \sigma_i$

Co-variance matrix:

$$q_{jk} = \frac{1}{N} \sum_{i=1}^{N} (x_{ij} - \bar{x}_j)(x_{ik} - \bar{x}_k)$$
Three options for assigning points, and their parallels across K-means, HMMs, Motifs

<table>
<thead>
<tr>
<th>Update rule</th>
<th>Update assignments (E step) ➔ Estimate hidden labels</th>
<th>Algorithm implementing E step in each of the three settings</th>
<th>Update model parameters (M step) ➔ max likelihood</th>
</tr>
</thead>
<tbody>
<tr>
<td>The hidden label is:</td>
<td>Expression clustering</td>
<td>HMM learning</td>
<td>Motif discovery</td>
</tr>
<tr>
<td>Pick a best</td>
<td>Assign each point to best label</td>
<td>K-means: Assign each point to nearest cluster</td>
<td>Viterbi training: label sequence with best path</td>
</tr>
<tr>
<td>Average all</td>
<td>Assign each point to all labels, probabilistically</td>
<td>Fuzzy K-means: Assign to all clusters, weighted by proximity</td>
<td>Baum-Welch training: label sequence w all paths (posterior decoding)</td>
</tr>
<tr>
<td>Sample one</td>
<td>Pick one label at random, based on their relative probability</td>
<td>N/A: Assign to a random cluster, sample by proximity</td>
<td>N/A: Sample a single label for each position, according to posterior prob.</td>
</tr>
</tbody>
</table>
1. Introduction to gene expression analysis
   – Technology: microarrays vs. RNAseq. Resulting data matrices
   – Supervised (Clustering) vs. unsupervised (classification) learning

2. K-means clustering (clustering by partitioning)

3. Hierarchical Clustering (clustering by agglomerative clustering)
   – Basic algorithm, Distance measures. Evaluating clustering results

4. Naïve Bayes classification (generative approach to classification)
   – Discriminant function: class priors, and class-conditional distributions
   – Training and testing, Combine mult features, Classification in practice

5. (optional) Support Vector Machines (discriminative approach)
   – SVM formulation, Margin maximization, Finding the support vectors
   – Non-linear discrimination, Kernel functions, SVMs in practice
Challenge of K-means: picking K

• How do we select K?
  – We can always make clusters “more compact” by increasing K
  – e.g. What happens is if K=number of data points?
  – What is a meaningful improvement?

• Hierarchical clustering side-steps this issue
Hierarchical clustering

Most widely used algorithm for expression data

- Start with each point in a separate cluster
- At each step:
  - Choose the pair of closest clusters
  - Merge

Phylogeny (UPGMA)

Unweighted Pair Group Method with Arithmetic-mean

Select a “cut level” to create disjoint clusters
Distance between clusters

- \( CD(X,Y) = \min_{x \in X, y \in Y} D(x,y) \) 
  *Single-link method*

- \( CD(X,Y) = \max_{x \in X, y \in Y} D(x,y) \) 
  *Complete-link method*

- \( CD(X,Y) = \text{avg}_{x \in X, y \in Y} D(x,y) \) 
  *Average-link method*

- \( CD(X,Y) = D(\text{avg}(X), \text{avg}(Y)) \) 
  *Centroid method*

Cluster distance affects both results and runtime
## Point-to-point (Dis)Similarity Measures

<table>
<thead>
<tr>
<th>Similarity Measure</th>
<th>Formula</th>
</tr>
</thead>
<tbody>
<tr>
<td>Manhattan distance</td>
<td>$d_{fg} = \sum_c</td>
</tr>
<tr>
<td>Euclidean distance</td>
<td>$d_{fg} = \sqrt{\sum_c (e_{fc} - e_{gc})^2}$</td>
</tr>
<tr>
<td>Mahalanobis distance</td>
<td>$d_{fg} = (e_f - e_g)\Sigma^{-1}(e_f - e_g)$, where $\Sigma$ is the (full or within-cluster) covariance matrix of the data</td>
</tr>
<tr>
<td>Pearson correlation (centered)</td>
<td>$d_{fg} = 1 - r_{fg}$ with $r_{fg} = \frac{\sum_c (e_{fc} - \bar{e}<em>f)(e</em>{gc} - \bar{e}<em>g)}{\sqrt{\sum_c (e</em>{fc} - \bar{e}<em>f)^2\sum_c (e</em>{gc} - \bar{e}_g)^2}}$</td>
</tr>
<tr>
<td>Uncentered correlation (angular)</td>
<td>$d_{fg} = 1 - r_{fg}$ with $r_{fg} = \frac{\sum_c e_{fc}e_{gc}}{\sqrt{\sum_c e_{fc}^2\sum_c e_{gc}^2}}$</td>
</tr>
<tr>
<td>Spellman rank correlation</td>
<td>As Pearson correlation, but replace $e_{gc}$ with the rank of $e_{gc}$ within the expression values of gene $g$ across all conditions $c = 1...C$</td>
</tr>
<tr>
<td>Absolute or squared correlation</td>
<td>$d_{fg} = 1 -</td>
</tr>
</tbody>
</table>

$d_{fg}$, distance between expression patterns for genes $f$ and $g$, $e_{gc}$, expression level of gene $g$ under condition $c$. 

D’haeseleer (2005) Nat Biotech

Courtesy of Macmillan Publishers Limited. Used with permission.

Today: Gene Expression Clustering & Classification

1. Introduction to gene expression analysis
   – Technology: microarrays vs. RNA-seq. Resulting data matrices
   – Supervised (Clustering) vs. unsupervised (classification) learning

2. K-means clustering (clustering by partitioning)

3. Hierarchical Clustering (clustering by agglomeration)
   – Basic algorithm, Distance measures. Evaluating clustering results

4. Naïve Bayes classification (generative approach to classification)
   – Discriminant function: class priors, and class-conditional distributions
   – Training and testing, Combine mult features, Classification in practice

5. (optional) Support Vector Machines (discriminative approach)
   – SVM formulation, Margin maximization, Finding the support vectors
   – Non-linear discrimination, Kernel functions, SVMs in practice
Evaluating Cluster Performance

In general, it depends on your goals in clustering

• Robustness
  – Select random samples from data set and cluster
  – Repeat
  – Robust clusters show up in all clusters

• Category Enrichment
  – Look for categories of genes “over-represented” in particular clusters
  – Also used in Motif Discovery
Evaluating clusters – Hypergeometric Distribution

\[
P(\text{pos} \geq r) = \sum_{m \geq r} \frac{\binom{p}{m} \binom{N-p}{k-m}}{\binom{N}{k}}
\]

P-value of uniformity in computed cluster

Prob that a randomly chosen set of k experiments would result in m positive and k-m negative

- N experiments, p labeled +, (N-p) –
- Cluster: k elements, m labeled +, k-m labeled -
- P-value of *single* cluster containing k elements of which at least r are +

Select k elements (at random)

m happen to be + (out of p +’s)
k-m happen to be - (out of N-p -’s)
Evaluation using functional enrichment

Clustered 8600 human genes using expression time course in fibroblasts

(A) Cholesterol biosynthesis
(B) Cell cycle
(C) Immediate early response
(D) Signalling and angiogenesis
(E) Wound healing

Evaluation based on motif content

Expression from 15 time points during yeast cell cycle

Courtesy of Nature Publishing Group. Used with permission.
Today: Gene Expression Clustering & Classification

1. Introduction to gene expression analysis
   – Technology: microarrays vs. RNAseq. Resulting data matrices
   – Supervised (Clustering) vs. unsupervised (classification) learning

2. K-means clustering (clustering by partitioning)

3. Hierarchical Clustering (clustering by agglomeration)
   – Basic algorithm, Distance measures. Evaluating clustering results

4. Naïve Bayes classification (generative approach to classification)
   – Discriminant function: class priors, and class-conditional distributions
   – Training and testing, Combine mult features, Classification in practice

5. (optional) Support Vector Machines (discriminative approach)
   – SVM formulation, Margin maximization, Finding the support vectors
   – Non-linear discrimination, Kernel functions, SVMs in practice
Two Approaches to Classification

• **Generative**
  – Bayesian Classification (e.g. Naïve Bayes)
  – Pose classification problem in prob terms
  – Model feature distribution in different classes
  – Use probability calculus for making decisions

• **Discriminative**
  – E.g. Support Vector Machines
  – No modeling of underlying distributions
  – Make decisions using distance from boundary

• Example: Gene finding: HMMs vs. CRFs
Bayesian classification with a single feature

1. If you know both distributions, how to classify a new example
2. If you have many classified examples, how to estimate model params.
   - Parametric vs. non-parametric models. Class-conditional distributions. Priors
3. Bayes’ Rule:
   - $P(C|F)$ from $P(F|C)$
   - Take probability ratios

Ex 1: DNA repair genes show higher expression during stress
Ex 2: Protein-coding regions show higher conservation levels
Ex 3: Regulatory regions show higher GC-content

In general: foreground signal vs. background

$P(\text{Feature} \mid \text{Class})$
Classification problem: Max Probability Class

Select the class that maximizes posterior:

\[
P(Class \mid Feature) = \frac{P(Feature \mid Class) P(Class)}{P(Feature)}
\]

Maximum-A-Posteriori (MAP) estimates

\[
BestClass = \arg\max_C P(Class \mid Feature)
\]

\[
= \arg\max_C P(Feature \mid Class) P(Class)
\]

Scaling the above distribution based on class priors
Likelihood:

\[ P(Class \mid Feature) = \frac{P(Feature \mid Class)P(Class)}{P(Feature)} \]

Features for each class drawn from conditional probability distributions (conditional on the class)

\[ \text{P(X|Class1)} \quad \text{P(X|Class2)} \]

Our first goal will be to model these class-conditional probability distributions (CCPD)
Class Priors: 

We model prior probabilities to quantify the expected a priori chance of seeing a class

\[ P(Class \mid Feature) = \frac{P(Feature \mid Class)P(Class)}{P(Feature)} \]

\[ \text{P(Class2) \ & \ P(Class1)} \]

\[ P(\text{mito}) = \text{how likely is the next protein to be a mitochondrial protein before I see any features to help me decide} \]

We expect \(~1500\) mitochondrial genes out of \(~21000\) total, so

\[ P(\text{mito})=1500/21000 \]
\[ P(\sim\text{mito})=19500/21000 \]
Evidence

If we observe an object with feature $X$, how do we decide if the object is from Class 1?

The Bayes Decision Rule is simply choose Class 1 if:

$$P(Class_1 | X) > P(Class_2 | X)$$

Total evidence is $P(Feature) = \sum_i P(Feature | Class_i)P(Class_i)$

But it does not need to be known for classification

If we observe an object with feature $X$, how do we decide if the object is from Class 1?

The Bayes Decision Rule is simply choose Class 1 if:

$$\frac{P(X | Class_1)P(Class_1)}{P(X)} > \frac{P(X | Class_2)P(Class_2)}{P(X)}$$

$\Rightarrow$ $P(Feature)$ does not need to be computed for classification.
Discriminant Function for selecting Class 1

We can create a convenient representation of the Bayes Decision Rule

\[ P(X \mid \text{Class 1})P(\text{Class 1}) > P(X \mid \text{Class 2})P(\text{Class 2}) \]

\[
\frac{P(X \mid \text{Class 1})P(\text{Class 1})}{P(X \mid \text{Class 2})P(\text{Class 2})} > 1
\]

\[ G(X) = \log \frac{P(X \mid \text{Class 1})}{P(X \mid \text{Class 2})} \frac{P(\text{Class 1})}{P(\text{Class 2})} > 0 \]

*If \( G(X) > 0 \), we classify as Class 1*
Today: Gene Expression Clustering & Classification

1. Introduction to gene expression analysis
   – Technology: microarrays vs. RNAseq. Resulting data matrices
   – Supervised (Clustering) vs. unsupervised (classification) learning

2. K-means clustering (clustering by partitioning)

3. Hierarchical Clustering (clustering by agglomeration)
   – Basic algorithm, Distance measures. Evaluating clustering results

4. Naïve Bayes classification (generative approach to classification)
   – Discriminant function: class priors, and class-conditional distributions
   – Training and testing, Combine mult features, Classification in practice

5. (optional) Support Vector Machines (discriminative approach)
   – SVM formulation, Margin maximization, Finding the support vectors
   – Non-linear discrimination, Kernel functions, SVMs in practice
Training and Testing Datasets

The Rule

We *must* test our classifier on a different set from the training set: the labeled test set

The Task

We will classify each object in the test set and count the number of each type of error
Getting $P(X|\text{Class})$ from Training Set

**One Simple Approach**

Divide $X$ values into bins

And then we simply count frequencies

In general, and especially for continuous distributions, this can be a complicated problem: **Density Estimation**

How do we get this from these?

There are 13 data points
Distributions Over Many Features

**Estimating** $P(X_1,X_2,X_3,…,X_8|\text{Class } 1)$ can be difficult

- Assume each feature binned into 5 possible values
- We have $5^8$ combinations of values we need to count the frequency for

- Generally will not have enough data
  - We will have lots of nasty zeros
Getting Priors

Three general approaches

1. Estimate priors by counting fraction of classes in training set
   \[ P(\text{Class1}) = \frac{13}{23} \]
   \[ P(\text{Class2}) = \frac{10}{23} \]

   But sometimes fractions in training set are not representative of world

2. Estimate from “expert” knowledge
   Example
   \[ P(\text{mito}) = \frac{1500}{21000} \]
   \[ P(\neg\text{mito}) = \frac{19500}{21000} \]

3. We have no idea – use equal (uninformative) priors
   \[ P(\text{Class1}) = P(\text{Class2}) \]
1. **Introduction to gene expression analysis**
   - Technology: microarrays vs. RNAseq. Resulting data matrices
   - Supervised (Clustering) vs. unsupervised (classification) learning

2. **K-means clustering (clustering by partitioning)**

3. **Hierarchical Clustering (clustering by agglomerative)**
   - Basic algorithm, Distance measures. Evaluating clustering results

4. **Naïve Bayes classification (generative approach to classification)**
   - Discriminant function: class priors, and class-conditional distributions
   - Training and testing, Combine mult features, Classification in practice

5. **(optional) Support Vector Machines (discriminative approach)**
   - SVM formulation, Margin maximization, Finding the support vectors
   - Non-linear discrimination, Kernel functions, SVMs in practice
Combining Multiple Features

- We have focused on a single feature for an object
- But mitochondrial protein prediction (for example) has 7 features

So $P(X|\text{Class})$ become $P(X_1,X_2,X_3,\ldots,X_8|\text{Class})$ and our discriminant function becomes

$$G(X) = \log \frac{P(X_1, X_2, \ldots, X_7 | \text{Class1}) \cdot P(\text{Class1})}{P(X_1, X_2, \ldots, X_7 | \text{Class2}) \cdot P(\text{Class2})} > 0$$
Naïve Bayes Classifier

We are going to make the following assumption:

**All features are independent given the class**

\[
P(X_1, X_2, ..., X_n \mid \text{Class}) = P(X_1 \mid \text{Class})P(X_2 \mid \text{Class})...P(X_n \mid \text{Class})
\]

\[
= \prod_{i=1}^{n} P(X_i \mid \text{Class})
\]

We can thus estimate **individual distributions** for each feature and just **multiply** them together!
Thus, with the Naïve Bayes assumption, we can now rewrite, this:

\[ G(X_1, \ldots, X_7) = \log \frac{P(X_1, X_2, \ldots, X_7 \mid \text{Class1}) \cdot P(\text{Class1})}{P(X_1, X_2, \ldots, X_7 \mid \text{Class2}) \cdot P(\text{Class2})} > 0 \]

As this:

\[ G(X_1, \ldots, X_7) = \log \prod_{i=1}^{7} \frac{P(X_i \mid \text{Class1}) \cdot P(\text{Class1})}{P(X_i \mid \text{Class2}) \cdot P(\text{Class2})} > 0 \]

Which can be simply computed as the sum of log scores
Binary Classification Errors

<table>
<thead>
<tr>
<th>Predicted</th>
<th>True (Mito)</th>
<th>False (~Mito)</th>
</tr>
</thead>
<tbody>
<tr>
<td>True</td>
<td>TP</td>
<td>FP</td>
</tr>
<tr>
<td>False</td>
<td>FN</td>
<td>TN</td>
</tr>
</tbody>
</table>

Sensitivity = $\frac{TP}{TP+FN}$  Specificity = $\frac{TN}{TN+FP}$

- **Sensitivity**
  - Fraction of all Class 1 (True) that we correctly predicted at Class 1
  - *How good are we at finding what we are looking for*

- **Specificity**
  - Fraction of all Class 2 (False) called Class 2
  - *How many of the Class 2 do we filter out of our Class 1 predictions*
1. **Introduction to gene expression analysis**
   - Technology: microarrays vs. RNAseq. Resulting data matrices
   - Supervised (Clustering) vs. unsupervised (classification) learning

2. **K-means clustering (clustering by partitioning)**

3. **Hierarchical Clustering (clustering by agglomeration)**
   - Basic algorithm, Distance measures. Evaluating clustering results

4. **Naïve Bayes classification (generative approach to classification)**
   - Discriminant function: class priors, and class-conditional distributions
   - Training and testing, Combine mult features, Classification in practice

5. *(optional)* **Support Vector Machines (discriminative approach)**
   - SVM formulation, Margin maximization, Finding the support vectors
   - Non-linear discrimination, Kernel functions, SVMs in practice
Classifying Mitochondrial Proteins

Derive 7 features for all human proteins

- Targeting signal
- Protein domains
- Co-expression
- Mass Spec
- Homology
- Induction
- Motifs

Predict nuclear encoded mitochondrial genes

Maestro

First page of article removed due to copyright restrictions.
Individual Feature Distributions

Instead of a single big distribution, we have a smaller one for each feature (and class)

- \( P(\text{Target} | \text{Mito}) \) vs \( P(\text{Target} | \sim\text{Mito}) \)
- \( P(\text{Domain} | \text{Mito}) \) vs \( P(\text{Domain} | \sim\text{Mito}) \)
- \( P(\text{CE} | \text{Mito}) \) vs \( P(\text{CE} | \sim\text{Mito}) \)
- \( P(\text{Mass} | \text{Mito}) \) vs \( P(\text{Mass} | \sim\text{Mito}) \)
- \( P(\text{Homology} | \text{Mito}) \) vs \( P(\text{Homology} | \sim\text{Mito}) \)
- \( P(\text{Induc} | \text{Mito}) \) vs \( P(\text{Induc} | \sim\text{Mito}) \)
- \( P(\text{Motif} | \text{Mito}) \) vs \( P(\text{Motif} | \sim\text{Mito}) \)

Courtesy of Nature Publishing Group. Used with permission.
Classifying A New Protein

Plug these and priors into the discriminant function

\[
G(X_1, ..., X_7) = \log \frac{\prod P(X_i \mid \text{Mito})}{\prod P(X_i \mid \sim \text{Mito})} \cdot \frac{P(\text{Mito})}{P(\sim \text{Mito})} > 0
\]

**IF G>0, we predict that the protein is from class Mito**
Apply to human proteome: 1,451 predictions (of which 490 are novel predictions)

Problem in genomics: not everything novel is false

Slide Credit: S. Calvo

Naïve Bayes (Maestro) (99%, 71%)
1. Introduction to gene expression analysis
   – Technology: microarrays vs. RNAseq. Resulting data matrices
   – Supervised (Clustering) vs. unsupervised (classification) learning

2. K-means clustering (clustering by partitioning)

3. Hierarchical Clustering (clustering by agglomeration)
   – Basic algorithm, Distance measures. Evaluating clustering results

4. Naïve Bayes classification (generative approach to classification)
   – Discriminant function: class priors, and class-conditional distributions
   – Training and testing, Combine mult features, Classification in practice

5. (optional) Support Vector Machines (discriminative approach)
   – SVM formulation, Margin maximization, Finding the support vectors
   – Non-linear discrimination, Kernel functions, SVMs in practice
Support Vector Machines (SVMs)

Easy to select a line

But many lines will separate these training data

What line should we choose?
A sensible choice is to select a line that maximizes the *margin* between classes.
**SVM Formulation**

We define a vector $w$ normal to the separating line.

Assume all data satisfy the following:

- $x_i \cdot w - b \geq +1$ for $y_i = +1$
- $x_i \cdot w - b \leq -1$ for $y_i = -1$

We want to find the separator with the largest margin.
An Optimization Problem

For full derivation, see Burges (1998)

Minimize \( L_D = \sum_i \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j \langle x_i, x_j \rangle \)

subject to \( \sum_i \alpha_i y_i = 0 \) and \( \alpha_j > 0 \)

\[ \alpha_i \left( y_i \left( x_i \cdot w - b \right) - 1 \right) = 0 \]

\[ w = \sum_i \alpha_i y_i x_i \]

\( x_i \) with \( \alpha_i > 0 \) are the support vectors

\( w \) is determined by these data points!
Using an SVM

Given a new data point we simply assign it the label:

\[ y_i = \text{sign}(w \cdot x_{\text{new}} - b) \]

\[ = \text{sign}\left( \sum_i \alpha_i y_i x_i \cdot x_{\text{new}} - b \right) \]

Again, only dot product of input data!
1. **Introduction to gene expression analysis**
   - Technology: microarrays vs. RNAseq. Resulting data matrices
   - Supervised (Clustering) vs. unsupervised (classification) learning

2. **K-means clustering (clustering by partitioning)**

3. **Hierarchical Clustering (clustering by agglomeration)**
   - Basic algorithm, Distance measures. Evaluating clustering results

4. **Naïve Bayes classification (generative approach to classification)**
   - Discriminant function: class priors, and class-conditional distributions
   - Training and testing, Combine mult features, Classification in practice

5. **(optional) Support Vector Machines (discriminative approach)**
   - SVM formulation, Margin maximization, Finding the support vectors
   - Non-linear discrimination, **Kernel functions**, SVMs in practice
Non-linear Classifier

- Some data not linearly separable in low dimensions
- What if we **transform** it to a higher dimension?

1 dimensional data

2 dimensional data

Kernel function

© sources unknown. All rights reserved. This content is excluded from our Creative Commons license. For more information, see [http://ocw.mit.edu/help/faq-fair-use/](http://ocw.mit.edu/help/faq-fair-use/)

Kernel Mapping

Want a **mapping** from input space, $\mathbb{R}^d$, to other euclidean space, $H$

$$\Phi(x): \mathbb{R}^d \rightarrow H$$

But $\Phi(X)$ can be a mapping to an infinite dimensional space
i.e. $d$ points become an infinite number of points

$$X=(x_1,x_2) \quad \rightarrow \quad \Phi(X)=(\phi_1,\phi_2,\phi_3,\ldots,\phi_\infty)$$

*Rather difficult to work with!*
Kernel Mapping

Want a **mapping** from input space, \( R^d \), to other euclidean space, \( H \)

From previous slide, SVMs *only depend on dot product*

\[
\Phi(x): R^d \rightarrow H
\]

\[
X_i \cdot X_j \quad \text{becomes} \quad \Phi(X_i) \cdot \Phi(X_j)
\]

**Here is trick:** if we have a kernel function such that

\[
K(X_i, X_j) = \Phi(X_i) \cdot \Phi(X_j)
\]

We can just use \( K \) and never know \( \Phi(x) \) explicitly!

\( \Phi(X) \) is high dimensional

\( K \) is a scalar
Kernels

So the key step is to take your input data and transform it into a **kernel matrix**

We have then done two very useful things:

1. Transformed $X$ into a high (possibly infinite) dimensional space (where we hope are data are separable)
2. Taken dot products in this space to create **scalars**
Example Kernels

\[ K(x_i, x_j) = x_i^T x_j \]  \hspace{1cm} \text{Linear}

\[ K(x_i, x_j) = (\gamma x_i^T x_j + r)^d \]  \hspace{1cm} \text{Polynomial}

\[ K(x_i, x_j) = \exp(-\gamma \|x_i - x_j\|^2) \]  \hspace{1cm} \text{Radial Basis Function}

\[ K(x_i, x_j) = \tanh(\gamma x_i^T x_j + r) \]  \hspace{1cm} \text{Sigmoid}

What \( K(X_i, X_j) \) are valid kernels?

Answer given by Mercer’s Condition (see Burgess 1998)
Using (Non-Linear) SVMs

**Step 1 – Transform data to Kernel Matrix K**

![Kernel Matrix Diagram]

**Step 2 – Train SVM on transformed data – get support vectors**

Minimize $L_D = \sum_i \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j x_i \cdot x_j = \sum_i \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j K(x_i, x_j)$

**Step 2 – Test/Classify on new samples**

$y_{\text{new}} = \text{sign} \left( w \cdot x_{\text{new}} \right) = \text{sign} \left( \sum_i \alpha_i y_i x_i \cdot x_{\text{new}} \right) = \text{sign} \left( \sum_i \alpha_i y_i K(x_i, x_{\text{new}}) \right)$
Today: Gene Expression Clustering & Classification

1. Introduction to gene expression analysis
   – Technology: microarrays vs. RNAseq. Resulting data matrices
   – Supervised (Clustering) vs. unsupervised (classification) learning

2. K-means clustering (clustering by partitioning)

3. Hierarchical Clustering (clustering by agglomeration)
   – Basic algorithm, Distance measures. Evaluating clustering results

4. Naïve Bayes classification (generative approach to classification)
   – Discriminant function: class priors, and class-conditional distributions
   – Training and testing, Combine mult features, Classification in practice

5. (optional) Support Vector Machines (discriminative approach)
   – SVM formulation, Margin maximization, Finding the support vectors
   – Non-linear discrimination, Kernel functions, SVMs in practice
Classifying Tumors with Array Data

• **Primary samples:**
  – 38 bone marrow samples
  – 27 ALL, 11 AML
  – obtained from acute leukemia patients at the time of diagnosis;

• **Independent samples:**
  – 34 leukemia samples
  – 24 bone marrow
  – 10 peripheral blood samples

• **Assay ~6800 Genes**

Excerpt of article removed due to copyright restrictions.
Weighted Voting Classification

General approach of Golub et al (1999) paper:

- Choosing a set of informative genes based on their correlation with the class distinction
- Each informative gene casts a weighted vote for one of the classes
- Summing up the votes to determine the winning class and the prediction strength
Results

Initial Samples

• 36 of the 38 samples as either AML or ALL. All 36 samples agree with clinical diagnosis
• 2 not predicted

Independent Samples

• 29 of 34 samples are strongly predicted with 100% accuracy.
• 5 not predicted
Training Set

Figure 3 B and caption removed due to copyright restrictions.
Supplementary Figure 2 and caption removed due to copyright restrictions. Source: Golub, Todd R. et al. "Molecular classification of cancer: Class discovery and class prediction by gene expression monitoring." Science 286, no. 5439 (1999): 531-537.
SVM Approach

Text and table removed due to copyright restrictions.
Methods

• Generate 4 classifiers using different numbers of genes
  – 7129, 999, 99, 49 most informative

• Linear SVM

• Distance from hyperplane (i.e. margin) provides confidence level
Results

Text and table removed removed due to copyright restrictions.
Results

Figure 9.6 removed due to copyright restrictions.
Bringing Clustering and Classification Together

Semi-Supervised Learning

Common Scenario
- Few labeled
- Many unlabeled
- Structured data

What if we cluster first?

Then clusters can help us classify
Today: Gene Expression Clustering & Classification

1. Introduction to gene expression analysis
   – Technology: microarrays vs. RNAseq. Resulting data matrices
   – Supervised (Clustering) vs. unsupervised (classification) learning

2. K-means clustering (clustering by partitioning)

3. Hierarchical Clustering (clustering by agglomeration)
   – Basic algorithm, Distance measures. Evaluating clustering results

4. Naïve Bayes classification (generative approach to classification)
   – Discriminant function: class priors, and class-conditional distributions
   – Training and testing, Combine mult features, Classification in practice

5. (optional) Support Vector Machines (discriminative approach)
   – SVM formulation, Margin maximization, Finding the support vectors
   – Non-linear discrimination, Kernel functions, SVMs in practice
6.047 / 6.878 / HST.507 Computational Biology
Fall 2015

For information about citing these materials or our Terms of Use, visit: http://ocw.mit.edu/terms.