Lecture 08
Gene expression analysis: Clustering and Classification

Challenges in Computational Biology

Module II: Modeling genes and gene expression

Important: Brainstorming make-up tomorrow at 4pm

Today: Gene Expression Clustering & Classification

1. Probing Gene Regulation: Technologies and Datasets
2. The clustering problem: Finding structure in unlabelled data
   a) K-means clustering: Algorithmic formulation
   b) Hierarchical Clustering: Algorithmic formulation
3. The classification problem: Partition data into known labels
   a) Generative classification: Bayesian classification
   b) Discriminative classification: Support Vector Machines

• Computational Foundations
  – Hidden Markov Models (HMMs): Central tool in CS
  – Decoding, evaluation, parsing, likelihood, scoring
  – Unsupervised Learning: Expectation Maximization
  – Supervised learning: generative/discriminative models
• Biological frontiers:
  – PS2: Modeling conservation, GC content, CpG islands
  – L6/L7: Genome annotation and parsing
  – L8: Gene expression analysis: cluster genes/conditions
  – L9: Regulatory motif discovery: EM, gibbs sampling, info
Central Dogma

DNA → mRNA → Protein → Phenotype

We can measure amounts of mRNA for every gene in a cell

Expression Microarrays

• A way to measure the levels of mRNA in every gene

• Two basic types
  – Affymetrix gene chips
  – Spotted oligonucleotides

• Both work on same principle
  – Put DNA probe on slide
  – Complementary hybridization

Expression Microarrays

• Measure the level of mRNA messages in a cell

RNA 1 → cDNA 1

RNA 4 → cDNA 4

RNA 6 → cDNA 6

• Variations on the theme:
  – Several probes per gene: tiled k-mers, averaged
  – Whole-genome tiling arrays forego gene prediction

mRNA-Sequencing

• Advent of next-generation sequencing changed game

• Sequence short reads from mRNA, map to genome
  – Measure the number of reads overlapping each known gene
  – Reconstruct entire transcriptome de novo in each experiment

Expression Analysis Data Matrix

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

• Measure 6,000 genes in 100s of conditions
• Study resulting matrix

Today: Gene Expression Clustering & Classification

1. Probing Gene Regulation: Technologies and Datasets
2. The clustering problem: Finding structure in unlabelled data
   – Partitioning (k-means) vs. Agglomerative (hierarchical) methods
   a) K-means clustering: Algorithmic formulation
   b) K-means clustering: Machine learning formulation
      – Generative models and learning. Expectation Maximization.
   c) Hierarchical Clustering: Algorithmic formulation
      – Basic algorithm, Distance measures. Evaluating clustering results
3. The classification problem: Partition data into known labels
   – Generative (model) vs. Discriminative (no model) classification
   a) Generative classification: Bayesian classification
      – Discriminant function: class priors, and class-conditional distributions
      – Training and testing. Combine multi-features. Classification in practice
   b) Discriminative classification: Support Vector Machines
      – SVM formulation, Margin maximization. Finding the support vectors
      – Non-linear discrimination, Kernel functions, SVMs in practice
Natural 1st step: group similar rows/columns

Grouping in both dimensions simultaneously

If labels are known: find more of same type

Today: Gene Expression Clustering & Classification

1. Probing Gene Regulation: Technologies and Datasets
2. The clustering problem: Finding structure in unlabelled data
   - Partitioning (k-means) vs. Agglomerative (hierarchical) methods
   a) K-means clustering: Algorithmic formulation
      - The basic algorithm. Update rule, optimality criterion. Fuzzy k-means.
   b) K-means clustering: Machine learning formulation
      - Generative models and learning. Expectation Maximization.
   c) Hierarchical Clustering: Algorithmic formulation
      - Basic algorithm, Distance measures. Evaluating clustering results
3. The classification problem: Partition data into known labels
   - Generative (model) vs. Discriminative (no model) classification
   a) Generative classification: Bayesian classification
      - Discriminant function: class priors, and class-conditional distributions
      - Training and testing, Combine multi features. Classification in practice
   b) Discriminative classification: Support Vector Machines
      - SVM formulation. Margin maximization, Finding the support vectors
      - Non-linear discrimination, Kernel functions, SVMs in practice

Clustering high-dimensional datasets

• Cluster Experiments
  - Group by similar expression profiles

• Cluster Genes
  - Group by similar expression in different conditions
Structure in High-Dimensional Data

- Structure can be used to:
  - reduce dimensionality of data
  - tell us something about underlying phenomena
  - make inferences about new data points

Clustering Algorithms

- 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

Today: Gene Expression Clustering & Classification

1. Probing Gene Regulation: Technologies and Datasets
2. The clustering problem: Finding structure in unlabelled data
   a) K-means clustering: Algorithmic formulation
      - The basic algorithm, Update rule, optimality criterion. Fuzzy k-means.
   b) K-means clustering: Machine learning formulation
      - Generative models and learning: Expectation Maximization.
   c) Hierarchical Clustering: Algorithmic formulation
      - Basic algorithm, Distance measures. Evaluating clustering results
3. The classification problem: Partition data into known labels
   a) Generative classification: Bayesian classification
      - Discriminant function: class priors, and class-conditional distributions
      - Training and testing, Combine mult features, Classification in practice
   b) Discriminative classification: Support Vector Machines
      - 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

Today: Gene Expression Clustering & Classification

1. Probing Gene Regulation: Technologies and Datasets
   - Partitioning (k-means) vs. Agglomerative (Hierarchical) methods
2. The clustering problem: Finding structure in unlabelled data
   - Partitioning (k-means) vs. Agglomerative (Hierarchical) methods
   a) K-means clustering: Algorithmic formulation
      - The basic algorithm, Update rule, optimality criteria, Fuzzy k-means.
   b) K-means clustering: Machine learning formulation
      - Generative models and learning, Expectation Maximization.
   c) Hierarchical Clustering: Algorithmic formulation
      - Basic algorithm, Distance measures, Evaluating clustering results
3. The classification problem: Partition data into known labels
   - Generative (model) vs. Discriminative (no model) classification
   a) Generative classification: Bayesian classification
      - Discriminant function: class priors, and class-conditional distributions
      - Training and testing, Combine multi-features, Classification in practice
   b) Discriminative classification: Support Vector Machines
      - SVM formulation, Margin maximization, Finding the support vectors
      - Non-linear discrimination, Kernel functions, SVMs in practice

K-means update rules

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

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

$$d_{ik} = (x_i - \mu_k)^2$$

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

$$\mu_k(n+1) = \frac{1}{\sum_{i \text{ with label } k} 1} \sum_{i \text{ with label } k} x_i$$

where: $|k| = \#x_i$ with label $k$
Today: Gene Expression Clustering & Classification

1. Probing Gene Regulation: Technologies and Datasets
   - Partitioning (k-means) vs. Agglomerative (Hierarchical) methods
   - The basic algorithm. Update rule, optimality criterion. Fuzzy k-means.
   - K-means clustering: Algorithmic formulation
   - K-means clustering: Machine learning formulation
   - Hierarchical Clustering: Algorithmic formulation
   - Hierarchical Clustering: Machine learning formulation

2. The clustering problem: Finding structure in unlabelled data
   - Discriminant function, class priors, and class-conditional distributions
   - Basic algorithm. Distance measures. Evaluating clustering results
   - Discriminant function: class priors, and class-conditional distributions

3. The classification problem: Partition data into known labels
   - Generative (model) vs. Discriminative (no model) classification
   - SVM formulation, Margin maximization, Finding the support vectors
   - Training and testing, Combine mult features, Classification in practice

K-means as a Generative Model

Model of P(X, Labels)

- Observations
- Generate
- Samples drawn from normal distributions with unit variance - a Gaussian Mixture Model

\[ P(x_i | u_k) = \frac{1}{\sqrt{2\pi}} \exp \left( -\frac{(x_i - u_k)^2}{2} \right) \]

Given only samples, how do we estimate max lik model params: (1) centroid definitions, (2) point assignments?

Revisiting K-Means

1. Initialize K centers \( u_k \)
2. Assign each \( x \) the label of the nearest center, where the distance between \( x \) and \( u_k \) is

\[ d_{ik} = (x_i - u_k)^2 \]

3. Move the position of each \( u_k \) to the centroid of the points with that label

\[ u_k(n + 1) = \frac{\sum_{i=1}^{N} x_i \text{ with label } k}{\sum_{i=1}^{N} \text{ with label } k} \]

4. Iterate

Fuzzy K-means update rule

Re-assign each point \( x_i \) to all centers weighted by distance

\[ P(\text{label } k | x_i, u_k) = \begin{cases} 1 & \text{if } x_i \text{ is closest to } u_k \\ 0 & \text{otherwise} \end{cases} \]

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

\[ \text{COST} = \sum_{i=1}^{N} \sum_{k=1}^{K} (x_i - u_k)^2 \]

To achieve this, minimize each cluster term separately:

\[ \sum_{i=\text{in cluster } k} (x_i - u_k)^2 = \sum_i x_i^2 - 2 \sum_i x_i u_k + K \sum u_k^2 - 2 \sum_i x_i u_k + K \sum u_k^2 \]

Optimum \( u_k = \sum_{i=\text{in cluster } k} x_i / \text{the centroid} \)

However: Some points can be almost halfway between two centers → Assign probabilistically → Fuzzy K-means

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

\[ \arg \max \left\{ \log \prod_{i=1}^{N} P(x_i | \mu_k) \right\} = \arg \max \sum_{i=1}^{N} \left( -\frac{1}{2} (x_i - u_k)^2 + \log \frac{1}{\sqrt{2\pi}} \right) \]

\[ = \arg \min \sum_{i=1}^{N} (x_i - u_k)^2 \]

Solution is the nearest center

\[ \arg \max P(x_i | \mu_k) = \arg \max \frac{1}{\sqrt{2\pi}} \exp \left( -\frac{(x_i - u_k)^2}{2} \right) = \arg \min (x_i - u_k) \]

Solution is the centroid of the \( x_i \) that maximize data likelihood

If centers are known → Estimate memberships
If assignments known → Compute centroids

Regular K-Means is a special case of fuzzy k-means where: \( (1 - \delta_{ik}) \) if \( x_i \) is closest to \( u_k \), \( \delta_{ik} = 0 \) otherwise

If \( \mu_k \) and labels known → Compute centroids

If \( \mu_k \) and labels unknown → Assign probabilistically

Fuzzy K-means update rule

Re-assign each point \( x_i \) to all centers weighted by distance

\[ P(\text{label } k | x_i, u_k) = \begin{cases} 1 & \text{if } x_i \text{ is closest to } u_k \\ 0 & \text{otherwise} \end{cases} \]
Revisiting Fuzzy K-Means

1. Initialize K centers \( u_k \)
2. For each point calculate the probability of membership for each category
   \[
P(\text{label } K \mid x_i, \mu_k) = \frac{Q_k(x)}{\sum_{k'} Q_{k'}(x)}
\]
3. Move the position of each \( u_k \) to the weighted centroid:
   \[
u_k(n+1) = \frac{\sum_{x_{i \in \	ext{cluster of } k}} x_{i}}{\sum_{x_{i \in \	ext{cluster of } k}} 1}
\]
4. Iterate

**Generative Model Perspective**

- Many generalizations possible. Can vary:
  - Cluster sizes \( \Rightarrow \) Class priors \( P(\text{class}) \)
  - Density/spread of points \( \Rightarrow \) Gaussian \( \mu, \sigma \)
  - Cluster shape \( \Rightarrow \) Co-variance
  - Update rules: Max/Sampling/Density (next slide)

But How Many clusters?

- 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

Expectation Maximization (EM)

1. Initialize parameters
2. \( E \) Step Estimate probability of hidden labels, \( Q \), given parameters and previous parameter
3. \( M \) Step Choose new parameters to maximize expected likelihood of parameters given data and \( Q \)
4. Iterate

### Expectation Maximization (EM)

### Fuzzy K-means only a special case of EM

- Many generalizations possible. Can vary:
  - Cluster sizes \( \Rightarrow \) Class priors \( P(\text{class}) \)
  - Density/spread of points \( \Rightarrow \) Gaussian \( \mu, \sigma \)
  - Cluster shape \( \Rightarrow \) Co-variance
  - Update rules: Max/Sampling/Density (next slide)

### But How Many clusters?

- 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

### Today: Gene Expression Clustering & Classification

1. Probing Gene Regulation: Technologies and Datasets
2. The clustering problem: Finding structure in unlabelled data
   - Partitioning (k-means) vs. Agglomerative (hierarchical) methods
3. k-means clustering: Algorithmic formulation
   - The basic algorithm. Update rule, optimality criterion. Fuzzy k-means.
   - Generative models and learning. Expectation Maximization.
4. Hierarchical Clustering: Algorithmic formulation
   - Basic algorithm. Distance measures. Evaluating clustering results
5. The classification problem: Partition data into known labels
   - Generative (E model) vs. Discriminative (M model) classification
   - Discriminative classification: Support Vector Machines
   - SVM formulation. Margin maximization. Finding the support vectors
   - Non-linear discrimination, Kernel functions, SVMs in practice
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

**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)

**Visualization of results**

Optimal leaf-ordering algorithms

**(Dis)Similarity Measures**

<table>
<thead>
<tr>
<th>Measure</th>
<th>Formula</th>
</tr>
</thead>
<tbody>
<tr>
<td>Euclidean distance</td>
<td>( d_{E} = \sqrt{(x_{1} - y_{1})^2} )</td>
</tr>
<tr>
<td>Manhattan distance</td>
<td>( d_{M} =</td>
</tr>
<tr>
<td>Canberra distance</td>
<td>( d_{C} = \sum_{i=1}^{n} \frac{</td>
</tr>
<tr>
<td>Hellinger distance</td>
<td>( d_{H} = \left( \frac{\sqrt{\sum_{i=1}^{n} (x_{i}^{2} + y_{i}^{2})} - \sqrt{\sum_{i=1}^{n} x_{i}^{2}}}{\sqrt{2}} \right) )</td>
</tr>
<tr>
<td>Pearson correlation</td>
<td>( d_{P} = 1 - \frac{\sum_{i=1}^{n} (x_{i} - \bar{x})(y_{i} - \bar{y})}{\sqrt{\sum_{i=1}^{n} (x_{i} - \bar{x})^2 \sum_{i=1}^{n} (y_{i} - \bar{y})^2}} )</td>
</tr>
<tr>
<td>Spearman rank correlation</td>
<td>( d_{S} = 1 - \frac{6 \sum_{i=1}^{n} (r_{i} - \bar{r})^2}{n(n^2 - 1)} )</td>
</tr>
<tr>
<td>Absolute squared correlation</td>
<td>( d_{A} = 1 - \frac{\sum_{i=1}^{n}</td>
</tr>
</tbody>
</table>

\( r_{i} \): rank of the ith observation
\( \bar{r} \): average rank

\[ d'_{haeseleer} (2005) \text{ Nat Biotech} \]
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

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

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

P-value of uniformity in computed cluster

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

[Eisen (1998) PNAS]

(Tavazoie & Church (1999))

Today: Gene Expression Clustering & Classification

1. Probing Gene Regulation: Technologies and Datasets
   - Partitioning (k-means) vs. Agglomerative (Hierarchical) methods
2. The clustering problem: Finding structure in unlabelled data
   - K-means clustering: Algorithmic formulation
     - The basic algorithm, Update rule, optimality criterion, Fuzzy k-means.
   - K-means clustering: Machine learning formulation
     - Generative models and learning, Expectation Maximization.
   - Hierarchical Clustering: Algorithmic formulation
     - Basic algorithm, Distance measures, Evaluating clustering results
3. The classification problem: Partition data into known labels
   - Generative (model) vs. Discriminative (no model) classification
     - Generative (model) classification
       - Bayesian classification
         - Discriminant function: class priors, and class-conditional distributions
         - Training and testing, Combine multi features, Classification in practice
     - Discriminative classification: Support Vector Machines
       - SVM formulation, Margin maximization, Finding the support vectors
       - Non-linear discrimination, Kernel functions, SVMs in practice

Clustering vs Classification

- Objects characterized by one or more features
- Classification
  - Have labels for some points
  - Want a “rule” that will accurately assign labels to new points
  - Supervised learning
- Clustering
  - No labels
  - Group points into clusters based on how “near” they are to one another
  - Identify structure in data
  - Unsupervised learning
Today: Gene Expression Clustering & Classification

1. Probing Gene Regulation: Technologies and Datasets
2. The clustering problem: Finding structure in unlabelled data
   - Partitioning (k-means) vs. Agglomerative (hierarchical) methods
   a) K-means clustering: Algorithmic formulation
      - The basic algorithm: Update rule, optimality criterion. Fuzzy k-means.
   b) K-means clustering: Machine learning formulation
      - Generative models and learning. Expectation Maximization.
   c) Hierarchical Clustering: Algorithmic formulation
      - Basic algorithm, Distance measures. Evaluating clustering results
3. The classification problem: Partition data into known labels
   - Generative (model) vs. Discriminative (no model) classification
   a) Generative classification: Bayesian classification
      - Discriminant function: class priors, and class-conditional distributions
      - Training and testing, Combine mutl features, Classification in practice
   b) Discriminative classification: Support Vector Machines
      - SVM formulation, Margin maximization, Finding the support vectors
      - Non-linear discrimination, Kernel functions, SVMs in practice

Informative features provide discrimination

- Each object can be associated with multiple features
- We will look at the case of just one feature for now

Evaluate \( P(\text{class}|x) \) as a function of \( P(x|\text{class}) \) and \( P(\text{class}) \)

Using Bayes’ Rule: \( P(\text{class}|x) = \frac{P(x|\text{class})P(\text{class})}{P(x)} \)

Class Priors:

We model prior probabilities to quantify the expected \textit{a priori} chance of seeing a class

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

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

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

\[ P(\text{milo}) = \frac{1500}{21000} \quad P(\sim \text{milo}) = \frac{19500}{21000} \]

Evidence

Total evidence is \( P(\text{Feature}) = \sum_i P(\text{Feature}|\text{Class}_i)P(\text{Class}_i) \)

But it does not need to be known for classification

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

The Bayes Decision Rule is simply choose Class 1 if:

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

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

\( \Rightarrow P(\text{Feature}) \) 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

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

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

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

If \( G(X) > 0 \), we classify as Class 1

Today: Gene Expression Clustering & Classification

1. Probing Gene Regulation: Technologies and Datasets
   - Partitioning (k-means) vs. Agglomerative (hierarchical) methods
2. The clustering problem: Finding structure in unlabeled data
   a) K-means clustering: Algorithmic formulation
      - The basic algorithm: Update rule, optimality criterion. Fuzzy k-means.
   b) K-means clustering: Machine learning formulation
      - Generative models and learning. Expectation Maximization.
   c) Hierarchical Clustering: Algorithmic formulation
      - Basic algorithm, Distance measures. Evaluating clustering results
3. The classification problem: Partition data into known labels
   a) Generative classification: Bayesian classification
      - Discriminant function: class priors, and class-conditional distributions
   b) Discriminative classification: Support Vector Machines
      - 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

Getting Priors

Three general approaches
1. Estimate priors by counting fraction of classes in training set
   - \( P(\text{Class}1) = 13/23 \)
   - \( P(\text{Class}2) = 10/23 \)
   - But sometimes fractions in training set are not representative of world

2. Estimate from "expert" knowledge
   - Example: \( P(\text{mito}) = 1500/21000 \)
   - \( P(\neg \text{mito}) = 19500/21000 \)

3. We have no idea – use equal (uninformative) priors
   - \( P(\text{Class}1) = P(\text{Class}2) \)

Distributions Over Many Features

Estimating \( P(X_1, X_2, X_3, \ldots, X_8|\text{Class}1) \) 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
Today: Gene Expression Clustering & Classification

1. Probing Gene Regulation: Technologies and Datasets
   - Partitioning (k-means) vs. Agglomerative (hierarchical) methods

2. The clustering problem: Finding structure in unlabelled data
   a) K-means clustering: Algorithmic formulation
      - The basic algorithm. Update rule, optimality criterion. Fuzzy k-means.
   b) K-means clustering: Machine learning formulation
      - Generative models and learning. Expectation Maximization.
   c) Hierarchical Clustering: Algorithmic formulation
      - Basic algorithm, Distance measures. Evaluating clustering results

3. The classification problem: Partition data into known labels
   a) Generative classification: Bayesian classification
      - Discriminant function: class priors, and class-conditional distributions
      - Training and testing, Combine multiple features. Classification in practice
   b) Discriminative classification: Support Vector Machines
      - 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 $P(X_1,X_2,X_3,...,X_8|\text{Class})$ and our discriminant function becomes

$$G(X) = \log \frac{P(X_1,X_2,X_3,...,X_8|\text{Class})P(\text{Class})}{P(X_1,X_2,X_3,...,X_8|\text{Class})P(\text{Class})} > 0$$

Naïve Bayes Discriminant Function

Thus, with the Naïve Bayes assumption, we can now rewrite, this:

$$G(X_1,...,X_8) = \log \frac{P(X_1,X_2,X_3,...,X_8|\text{Class})P(\text{Class})}{P(X_1,X_2,X_3,...,X_8|\text{Class})P(\text{Class})} > 0$$

As this:

$$G(X_1,...,X_8) = \log \frac{\prod P(X_i|\text{Class})P(\text{Class})}{\prod P(X_i|\text{Class})P(\text{Class})} > 0$$

Binary Classification Errors

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

Sensitivity = TP/(TP+FN)  Specificity = 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

Today: Gene Expression Clustering & Classification

1. Probing Gene Regulation: Technologies and Datasets
   - Partitioning (k-means) vs. Agglomerative (hierarchical) methods

2. The clustering problem: Finding structure in unlabelled data
   a) K-means clustering: Algorithmic formulation
      - The basic algorithm. Update rule, optimality criterion. Fuzzy k-means.
   b) K-means clustering: Machine learning formulation
      - Generative models and learning. Expectation Maximization.
   c) Hierarchical Clustering: Algorithmic formulation
      - Basic algorithm, Distance measures. Evaluating clustering results

3. The classification problem: Partition data into known labels
   a) Generative classification: Bayesian classification
      - Discriminant function: class priors, and class-conditional distributions
      - Training and testing, Combine multiple features. Classification in practice
   b) Discriminative classification: Support Vector Machines
      - 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

Individual Feature Distributions

Instead of a single big distribution, we have a smaller one for each feature (and class):
- Targeting signal
- Protein domains
- Mass Spec
- Homology
- Induction
- Motifs

P(Target|Mito) P(Target|~Mito)
P(Domain|Mito) P(Domain|~Mito)
P(CE|Mito) P(CE|~Mito)
P(Mass|Mito) P(Mass|~Mito)
P(Homology|Mito) P(Homology|~Mito)
P(Induc|Mito) P(Induc|~Mito)
P(Motif|Mito) P(Motif|~Mito)

Apply to human proteome: 1,451 predictions
(of which 490 are novel predictions)
* Naïve Bayes (Maestro)
* (99%, 71%)

Problem in genomics: not everything novel is false

Today: Gene Expression Clustering & Classification

1. Probing Gene Regulation: Technologies and Datasets
   - Partitioning (k-means) vs. Agglomerative (hierarchical) methods
     a) K-means clustering: Algorithmic formulation
        - The basic algorithm. Update rule, optimality criterion: Fuzzy k-means.
     b) K-means clustering: Machine learning formulation
        - Generative models and learning. Expectation Maximization.
     c) Hierarchical Clustering: Algorithmic formulation
        - Basic algorithm, Distance measures. Evaluating clustering results

2. The clustering problem: Finding structure in unlabelled data
   - Partitioning (k-means) vs. Agglomerative (hierarchical) methods
   - Basic algorithm, Distance measures. Evaluating clustering results

3. The classification problem: Partition data into known labels
   - Generative (model) vs. Discriminative (no model) classification
     a) Generative classification: Bayesian classification
        - Discriminant function: class priors, and class-conditional distributions
           - Training and testing. Combine multi features, Classification in practice
     b) Discriminative classification: Support Vector Machines
        - 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?
Support Vector Machines (SVMs)

A sensible choice is to select a line that maximizes the margin between classes.

Support Vectors

SVM Formulation

We define a vector \( w \) normal to the separating line.

Assume all data satisfy the following:

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

We want to find the separator with the largest margin.

An Optimization Problem

For full derivation, see Burges (1998).

Minimize:

\[ L = \sum_{i,j} a_{ij} x_i \cdot x_j \]

Subject to:

\[ \sum_{i} a_i y_i = 0 \text{ and } a_i > 0 \]

Solving for:

\[ a_i (y_i (x_i \cdot w - b) - 1) = 0 \]

Only some \( a_i \) are non-zero.

\( x_i \) with \( a_i > 0 \) 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:

\[ y_{\text{new}} = \text{sign} \left( \sum_{i} a_i y_i x_i \cdot x_{\text{new}} - b \right) \]

Again, only dot product of input data!

Today: Gene Expression Clustering & Classification

1. Probing Gene Regulation: Technologies and Datasets
2. The clustering problem: Finding structure in unlabelled data
   - Partitioning (k-means) vs. Agglomerative (Hierarchical) methods
   a) K-means clustering: Algorithmic formulation
      - The basic algorithm. Update rule, optimality criterion. Fuzzy k-means.
   b) K-means clustering: Machine learning formulation
      - Generative models and learning. Expectation Maximization.
   c) Hierarchical Clustering: Algorithmic formulation
      - Basic algorithm, Distance measures. Evaluating clustering results
3. The classification problem: Partition data into known labels
   - Generative (model) vs. Discriminative (no model) classification
   a) Generative classification: Bayesian classification
      - Discriminant function: class priors, and class-conditional distributions
      - Training and testing, Combine multi features, Classification in practice
   b) Discriminative classification: Support Vector Machines
      - 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?
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

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

Rather difficult to work with!

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

- **Linear**
  \[ K(x_i, x_j) = x_i^T x_j \]

- **Polynomial**
  \[ K(x_i, x_j) = (\gamma x_i^T x_j + r)^d \]

- **Radial Basis Function**
  \[ K(x_i, x_j) = \exp(-\gamma \|x_i - x_j\|^2) \]

- **Sigmoid**
  \[ K(x_i, x_j) = \tanh(\gamma x_i^T x_j + r) \]

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 \( K \)

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

Minimize \( L = \sum \alpha_i - \frac{1}{2} \sum \alpha_i \alpha_j y_i y_j K(x_i, x_j) \)

Step 2 – Test/Classify on new samples

\[ y_{new} = \text{sign}(w \cdot x_{new}) = \text{sign}\left( \sum \alpha_i y_i K(x_i, x_{new}) \right) = \text{sign}\left( \sum \alpha_i \Phi(x_i) \cdot \Phi(x_{new}) \right) \]

Today: Gene Expression Clustering & Classification

1. Probing Gene Regulation: Technologies and Datasets
2. The clustering problem: Finding structure in unlabelled data
   - Partitioning (k-means) vs. Agglomerative (hierarchical) methods
   a) K-means clustering: Algorithmic formulation
      - The basic algorithm. Update rule, optimality criterion. Fuzzy k-means.
   b) K-means clustering: Machine learning formulation
      - Generative models and learning. Expectation Maximization.
   c) Hierarchical Clustering: Algorithmic formulation
      - Basic algorithm, Distance measures. Evaluating clustering results
3. The classification problem: Partition data into known labels
   - Generative (model) vs. Discriminative (no model) classification
   a) Generative classification: Bayesian classification
      - Discriminant function: class priors, and class-conditional distributions
      - Training and testing, Combine multi features, Classification in practice
   b) Discriminative classification: Support Vector Machines
      - SVM formulation, Margin maximization, Finding the support vectors
      - Non-linear discrimination, Kernel functions
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

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

Supplementary fig. 2. Expression levels of predictive genes in independent dataset. The expression levels of the 50 genes most highly correlated with the ALL/AML class distinction in the initial dataset were determined in the independent dataset. Each row corresponds to a gene, with the columns corresponding to expression levels in different samples. The expression level of each gene in the independent dataset is shown relative to the mean of expression levels for that gene in the initial dataset. Expression levels greater than the mean are shaded in red, and those below the mean are shaded in blue. The scale indicates standard deviations above or below the mean. The top panel shows genes highly expressed in ALL, the bottom panel shows genes more highly expressed in AML. Note that while these genes as a group are highly correlated with the class distinction, no single gene is uniformly expressed across the class, illustrating the value of a multi-gene prediction method.

SVM Approach

Support Vector Machine Classification of Microarray Data
**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**

<table>
<thead>
<tr>
<th>genes</th>
<th>rejects</th>
<th>errors</th>
<th>confidence level</th>
</tr>
</thead>
<tbody>
<tr>
<td>7129</td>
<td>3</td>
<td>0</td>
<td>≈ 93%</td>
</tr>
<tr>
<td>40</td>
<td>0</td>
<td>0</td>
<td>≈ 93%</td>
</tr>
<tr>
<td>5</td>
<td>3</td>
<td>0</td>
<td>≈ 92%</td>
</tr>
</tbody>
</table>

*Figure 4.3: The signed distance, Δd, from the optimal separating hyperplane for the test samples. The squares are the correctly labeled ALL samples. The circles indicate the correctly labeled AML samples. The triangle marks the misclassified ALL case (see arrow).*

**Today: Gene Expression Clustering & Classification**

1. Probing Gene Regulation: Technologies and Datasets
   - The clustering problem: Finding structure in unlabelled data
     - Partitioning (k-means) vs. Agglomerative (hierarchical) methods
   a) K-means clustering: Algorithmic formulation
      - The basic algorithm, update rule, optimality criterion, Fuzzy k-means.
   b) K-means clustering: Machine learning formulation
      - Generative models and learning, Expectation Maximization.
   c) Hierarchical Clustering: Algorithmic formulation
      - Basic algorithm, Distance measures, Evaluating clustering results
  2. The classification problem: Partition data into known labels
     - Generative (model) vs. Discriminative (no model) classification
   a) Generative classification: Bayesian classification
      - Discriminant function: class priors, and class-conditional distributions
      - Training and testing, Combine mult features, Classification in practice
   b) Discriminative classification: Support Vector Machines
      - SVM formulation, Margin maximization, Finding the support vectors
      - Non-linear discrimination, Kernel functions, SVMs in practice

**Semi-Supervised Learning**

Common Scenario
- Few labeled
- Many unlabeled
- Structured data

What if we cluster first?

Then clusters can help us classify