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

Expression Analysis Data Matrix

- Measure 20,000 genes in 100s of conditions

RNA-Seq: De novo tx reconstruction / quantification

- Begin with probes for each predicted gene
- 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

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

Gene similarity questions

Study resulting matrix
Clustering vs. Classification

- **Supervised learning**
  - Conditions
  - Genes
  - Alizadeh, Nature 2000

  - Proliferation genes in transformed cell lines
  - B-cell genes in blood cell lines
  - Lymph node genes in diffuse large B-cell lymphoma (DLBCL)
  - Chronic lymphocytic leukemia

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

**Goal of Classification:** Extract features from the data that best assign new elements to one of well-defined classes.

---

**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

---

**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

---

**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 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 $k$

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

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

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

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

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 clusters.

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

To achieve this, minimize each cluster term separately:

\[
\sum_{x_i \text{ with label } k} (x_i - \mu_k)^2 = \sum x_i^2 - 2x_i \mu_k + \mu_k^2 = \sum x_i^2 - \mu_k^2
\]

Optimum \( \mu_k = \frac{\sum x_i}{N_k} \), 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

\( \Rightarrow \) For each point calculate the probability of membership for each category \( K \):

\[
P(K | x_i, \mu_k)
\]

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

Solution is exactly the k-means algorithm!

K-Means as a Generative Model

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

Generate \( \Rightarrow \) Estimate

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

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

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 \( \Rightarrow \) Estimate memberships

M step: If assignments known \( \Rightarrow \) Compute centroids

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

Solution is exactly the k-means algorithm!

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)
   - Machine learning formulation: Generative models, Expectation Maximization

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

M step: assignments known \( \Rightarrow \) compute centroids

Choose \( \mu_k \) and labeled centers \( \Rightarrow \) compute centroids

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

Solution is the centroid of the \( x_i \)

Equivalent

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

Solution is exactly the k-means algorithm!
EM is much more general than fuzzy K-means

<table>
<thead>
<tr>
<th>Algorithmic vs. machine learning formulations</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>K-means</strong></td>
</tr>
<tr>
<td><strong>algorithmic</strong></td>
</tr>
<tr>
<td><strong>formulation</strong></td>
</tr>
<tr>
<td><strong>Initialization</strong></td>
</tr>
<tr>
<td><strong>E-step</strong></td>
</tr>
<tr>
<td><strong>M-step</strong></td>
</tr>
</tbody>
</table>

**Three options for assigning points, and their parallels across K-means, HMMs, Motifs**

<table>
<thead>
<tr>
<th>Update rule</th>
<th>Algorithm implementing E step in each of the three settings</th>
<th>Update model parameters (M step) to maximize likelihood</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Expression clustering</strong></td>
<td><strong>HMM learning</strong></td>
<td><strong>Motif discovery</strong></td>
</tr>
<tr>
<td>Pick a best match to each position</td>
<td>K-means: Assign each point to nearest cluster</td>
<td>Viterbi training: label sequence with best path</td>
</tr>
<tr>
<td>Assign each point to all clusters, 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 from all positions as a probability</td>
<td>MEME: Use all positions as a motif occurrence weighted by motif match score</td>
<td>Average of all points, weighted by membership</td>
</tr>
<tr>
<td>N/A: Assign to a random cluster, sample by probability</td>
<td>Gibbs sampling: Use one position for the motif, by sampling from the match scores</td>
<td>Average of those points assigned to label (a sample)</td>
</tr>
<tr>
<td>N/A: Sample a single label for each position, according to posterior probability</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Sample one label at random, based on their relative probability</strong></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**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

**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

Point-to-point (Dis)Similarity Measures

<table>
<thead>
<tr>
<th>Table 1. Gene expression similarity measures</th>
<th>d_{xy} = \sum x_i - y_i</th>
<th>d_{xy} = \left( \sum x_i - y_i \right)^2</th>
<th>d_{xy} = \left( \sum x_i - y_i \right)^2</th>
<th>d_{xy} = 1 - r_{xy}</th>
<th>d_{xy} = 1 - r_{xy}</th>
</tr>
</thead>
<tbody>
<tr>
<td>Manhattan distance (city-block distance, L1 norm)</td>
<td>\left</td>
<td>x_i - y_i \right</td>
<td></td>
<td>\left</td>
<td>x_i - y_i \right</td>
</tr>
<tr>
<td>Euclidean distance (L2 norm)</td>
<td>\sqrt{\sum (x_i - y_i)^2}</td>
<td>\sqrt{\sum (x_i - y_i)^2}</td>
<td>\sqrt{\sum (x_i - y_i)^2}</td>
<td>\sqrt{\sum (x_i - y_i)^2}</td>
<td>\sqrt{\sum (x_i - y_i)^2}</td>
</tr>
<tr>
<td>Mahalanobis distance</td>
<td>\frac{1}{\sqrt{\sum (x_i - y_i)^2}}</td>
<td>\frac{1}{\sqrt{\sum (x_i - y_i)^2}}</td>
<td>\frac{1}{\sqrt{\sum (x_i - y_i)^2}}</td>
<td>\frac{1}{\sqrt{\sum (x_i - y_i)^2}}</td>
<td>\frac{1}{\sqrt{\sum (x_i - y_i)^2}}</td>
</tr>
<tr>
<td>Pearson correlation (contingent correlation)</td>
<td>1 - r_{xy}</td>
<td>1 - r_{xy}</td>
<td>1 - r_{xy}</td>
<td>1 - r_{xy}</td>
<td>1 - r_{xy}</td>
</tr>
<tr>
<td>Spearman's rank correlation</td>
<td>1 - r_{xy}</td>
<td>1 - r_{xy}</td>
<td>1 - r_{xy}</td>
<td>1 - r_{xy}</td>
<td>1 - r_{xy}</td>
</tr>
<tr>
<td>Absolute or squared correlation</td>
<td>1 - r_{xy}</td>
<td>1 - r_{xy}</td>
<td>1 - r_{xy}</td>
<td>1 - r_{xy}</td>
<td>1 - r_{xy}</td>
</tr>
</tbody>
</table>

D’haeseleer (2005) Nat Biotech

Evaluating clusters – Hypergeometric Distribution

\[ P(\text{pos} > r) = \sum_{m=0}^{\infty} \frac{\binom{p}{m} \binom{N-p}{k-m}}{\binom{N}{k}} \]

Select k elements (at random)

• 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 +

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

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

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

---

Bayesian classification with a single feature

**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

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:**
   - $P(C|F) \text{ from } P(F|C)$
   - $P(C|F) = \frac{P(F|C) \cdot P(C)}{P(F)}$

**Classification problem: Max Probability Class**

Select the class that maximizes posterior:

$$P(Class|Feature) = \frac{Likelihood \cdot Prior}{Evidence}$$

Max-A-Posteriori (MAP) estimates

$$BestClass = \arg\max_C P(Class|Feature)$$

Scaling the above distribution based on class priors
Likelihood:

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

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

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

Class Priors:

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

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

\[ P(\text{mito}) = \frac{1500}{21000} \quad \text{and} \quad P(\neg\text{mito}) = \frac{19500}{21000} \]

Evidence

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

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

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:

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

\[ \frac{P(X \mid \text{Class1}) P(L1)}{P(X)} > \frac{P(X \mid \text{Class2}) P(L2)}{P(X)} \]

\( P(X \mid \text{Class1}) P(\text{Class1}) > P(X \mid \text{Class2}) P(\text{Class2}) \)

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

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

\[ \frac{P(X \mid \text{Class1}) P(\text{Class1})}{P(X \mid \text{Class2}) P(\text{Class2})} > 1 \]

\[ G(X) = \log \frac{P(X \mid \text{Class1}) P(\text{Class1})}{P(X \mid \text{Class2}) P(\text{Class2})} > 0 \]

\( \Rightarrow G(X) > 0, \) 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, SVM 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|Class) from Training Set

One Simple Approach

Divide X values into bins
And then we simply count frequencies

There are 13 data points

How do we get this from these?

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

Distributions Over Many Features

Estimating P(X1,X2,X3,...,X8|Class1) 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(Class1)=13/23
   P(Class2)=10/23

But sometimes fractions in training set are not representative of world

Example
   P(mito)=1500/21000
   P(~mito)=19500/21000

2. Estimate from “expert” knowledge
   P(Class1)=P(Class2)

3. We have no idea – use equal (uninformative) priors
   P(Class1)=P(Class2)

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

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|Class) become P(X1,X2,X3,...,X8|Class)
and our discriminant function becomes

G(X) = \log \frac{P(X_1, X_2, ..., X_8 | Class_1) \cdot P(Class_1)}{P(X_1, X_2, ..., X_8 | Class_2) \cdot P(Class_2)} > 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 | Class) = P(X_1 | Class)P(X_2 | Class)...P(X_n | Class)

We can thus estimate individual distributions for each feature and just multiply them together!
### Naïve Bayes Discriminant Function

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

$$G(X_1, ..., X_n) = \log \frac{P(X_1, X_2, ..., X_n | \text{Class 1}) P(\text{Class 1})}{P(X_1, X_2, ..., X_n | \text{Class 2}) P(\text{Class 2})} > 0$$

As this:

$$G(X_1, ..., X_n) = \log \prod P(X_i | \text{Class 1}) P(\text{Class 1}) \prod P(X_i | \text{Class 2}) P(\text{Class 2}) > 0$$

Which can be simply computed as the sum of log scores

### 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. 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
- Mass Spec
- Co-expression
- Homology
- Induction
- Motifs

**Predict nuclear encoded mitochondrial genes**

**Maestro**

### Classifying A New Protein

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

- 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)

**Individual Feature Distributions**

- **Targeting signal**
  - Training ratio: No, Yes homology

**Plug these and priors into the discriminant function**

$$G(X_1, ..., X_n) = \log \prod P(X_i | \text{Mito}) P(\text{Mito}) \prod P(X_i | ~ \text{Mito}) P(~ \text{Mito}) > 0$$

**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

Support Vector Machines (SVMs)

Easy to select a line
But many lines will separate these training data
What line should we choose?

SVM Formulation

We define a vector \( \mathbf{w} \) normal to the separating line
Assume all data satisfy the following:
- \( x_i \cdot \mathbf{w} - b \geq +1 \) for \( y_i = +1 \)
- \( x_i \cdot \mathbf{w} - b \leq -1 \) for \( y_i = -1 \)
\( y_i (x_i \cdot \mathbf{w} - b) \geq 1 \)

Find the separator with the largest margin

Support Vector Machines (SVMs)

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

An Optimization Problem

For full derivation, see Burges (1998)

Minimize

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

subject to

\[
\sum_i \alpha_i y_i = 0 \quad \text{and} \quad \alpha_i > 0
\]

Solving for

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

\[
\mathbf{w} = \sum_i \alpha_i y_i x_i
\]

Only some \( \alpha_i \) are non-zero

\( \mathbf{x}_i \) with \( \alpha_i > 0 \) are the support vectors
\( \mathbf{w} \) is determined by these data points!
Using an SVM

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

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

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

**Again, only dot product of input data!**

---

**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

---

**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 i.e. \( d \) points become an infinite number of points

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

Rather difficult to work with!

---

**Kernel Mapping**

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

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

\[ \Phi(x_1) \cdot \Phi(x_2) = \text{scalar!} \]

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

- **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 \( K \)**

\[ K(x_i, x_j) \]

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

\[ L = \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j K(x_i, x_j) \]

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

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

### 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

### Classifying Tumors with Array Data

**Primary samples:**
- 38 bone marrow
- 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
Figure 3b. Genes distinguishing ALL from AML. The 50 genes most highly correlated with the ALL/AML class distinction are shown. Each row corresponds to a gene, with the columns corresponding to expression levels in different samples. Expression levels for each gene are normalized across the samples such that the mean is 0 and the standard deviation is 1. 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 appear correlated with class, no single gene is uniformly expressed across the class, illustrating the value of a multi-gene prediction method.

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 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.

SVM Approach

Support Vector Machine Classification of Microarray Data

Sinan Unlu & Ryan Rifkin

Artificial Intelligence Laboratory and The Center for Biological and Combinatorial Learning
Massachusetts Institute of Technology
Cambridge, Massachusetts 02139

http://www.ai.mit.edu

The Problem: Use the learning from example paradigm to make class predictions and infer genes involved in these predictions from microarray expression data. Specifically, we use a Support Vector Machine (SVM) classifier [1] to predict cancer subtypes or and tumor response to certain agents.

Methods

• SVM classifier (determine between two types of samples). The output of trained SVM is a linear decision function of each point in input space and coefficients for each gene. The margin is the distance between the closest data points and the hyperplane. The margin is a measure of confidence in the classification.

Results

| genes | rejects | errors | confidence level | |d|
|-------|---------|--------|-----------------|-----------|
| 7129  | 3       | 0      | ~ 93%           | .1        |
| 40    | 0       | 0      | ~ 93%           | .1        |
| 5     | 3       | 0      | ~ 92%           | .1        |

Figure 9.6 The signed distance, $d$, from the optimal separating hyperplane for the test samples. The diamonds are the correctly labeled ALL samples. The squares indicate the correctly labeled AML samples. The triangle marks the misclassified ALL case (see arrow).
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