Lecture 10
Gene expression analysis: Clustering and Classification

Module III: Regulation, Epigenomics, Networks

- Computational Foundations: Machine learning, dealing with noisy data
  - L10: Clustering/classification: Supervised/unsupervised learning
  - L12: Expectation Maximization (EM), Gibbs Sampling, Information theory
  - L13: Network algorithms, matrix operations, linear algebra, deconvolution
  - L14: Normalization, Reproducibility, False Discovery Rate, Integration

- Biological frontiers: Gene Regulation, Regulatory Systems Genomics
  - L10: Gene expression analysis.
  - L11: Epigenomics and Chromatin state.
  - L12: Regulatory motif discovery.
  - L13: Biological Network inference.
  - L14: Integrative genomics and the ENCODE project.

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. Naive Bayes classification (generative approach to classification)
   - Discriminant function: class priors, and class-conditional distributions

5. (optional) Support Vector Machines (discriminative approach)
   - SVM formulation, Margin maximization, Finding the support vectors

RNA-Seq: De novo tx reconstruction / quantification

- Measure 20,000 genes in 100s of conditions

Expression Analysis Data Matrix

- Each experiment measures expression of thousands of "spots", typically genes
- Study resulting matrix
Clustering vs. Classification

**Unsupervised learning**
- Independent validation of groups that emerge:
  - B-cell genes in blood cell lines
  - Lymph node genes in diffuse large B-cell lymphoma (DLBCL)
- 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

**Supervised learning**
- Known classes:
  - Independent validation of groups that emerge:
    - Feature Y (liver expression)
    - Feature X (brain expression)
- Clustering (unsupervised learning)
  - 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

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

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

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
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 \( k \)

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

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

\[
    \mu_k(n+1) = \frac{1}{\#x_i} \sum_{x_i \text{ with label } k} x_i
\]

where: \( \#x_i \) = \# \( x_i \) 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.

\[
\text{COST}(x_1, x_2, \ldots, x_n) = \sum_{k=1}^{K} \sum_{x_i \text{ with label } k} (x_i - \mu_k)^2
\]

To achieve this, minimize each cluster term separately:

\[
\sum_{k=1}^{K} \sum_{x_i \text{ with label } k} (x_i - \mu_k)^2 = \sum_{k=1}^{K} -u_k \sum_{x_i \text{ with label } k} x_i + |u_k|^2.
\]

Optimum \( u_k = \sum_{x_i \text{ with label } k} x_i \), the centroid

However: Some points can be almost halfway between two centers \( \Rightarrow \) Assign probabilistically

**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 \( K \):

\[
P(\text{label } K \mid x\mu_k)
\]

Update center \( \mu_k \) to the weighted mean of the points assigned to it:

\[
\mu_k(n+1) = \frac{1}{\sum_{x_i \text{ with label } k} P(\text{label } K \mid x\mu_k)} \sum_{x_i \text{ with label } k} P(\text{label } K \mid x\mu_k) x_i
\]

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

\[
P(\text{label } K \mid x\mu_k) = \begin{cases} 
1 & \text{if } x\mu_k \text{ closest to } \mu_k \\
0 & \text{otherwise}
\end{cases}
\]

**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. Naive Bayes classification (generative approach to classification)
   - Discriminant function: class priors, and class-conditional distributions
   - Training and testing, Combine multiple 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

**If centers are known \( \Rightarrow \) Estimate memberships**
If assignments known \( \Rightarrow \) Compute centroids

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

\[
\arg \max_{\mu_k} \left\{ \log \prod \left( \frac{1}{\sqrt{2\pi \sigma^2}} \exp \left( \frac{-1}{2} \left( \frac{x - \mu_k}{\sigma} \right)^2 \right) \right) \right\} = \arg \max_{\mu_k} \sum_i \left\{ \frac{-1}{2} \left( \frac{x_i - \mu_k}{\sigma} \right)^2 + \log \left( \frac{1}{\sqrt{2\pi \sigma^2}} \right) \right\}
\]

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

**Algorithmic vs. machine learning formulations**

<table>
<thead>
<tr>
<th>K-means</th>
<th>Fuzzy K-means</th>
</tr>
</thead>
<tbody>
<tr>
<td>algorithmic formulation</td>
<td>probabilistic interpretation</td>
</tr>
<tr>
<td>initialization</td>
<td>Initialize K centers ( \mu_k )</td>
</tr>
<tr>
<td>E-step: Estimate prob of hidden labels (point assignments to classes)</td>
<td>Estimate most likely missing label given previous assignments</td>
</tr>
<tr>
<td>M-step: Update parameters to maximize likelihood estimates given assignments</td>
<td>Move ( \mu_k ) to centroid of all points with that label</td>
</tr>
<tr>
<td>iteration</td>
<td>iterate</td>
</tr>
</tbody>
</table>

**P(\text{data} | \text{model}) guaranteed to increase each iteration of EM algo**
Fuzzy K-means only a special case of EM

- Many generalizations possible. Can vary:
  - Cluster sizes \( \rightarrow \) Class priors \( P(\text{class}_i) \) (uniform)
  - Density/spread of points \( \rightarrow \) Gaussian \( (\mu, \sigma) \) (unit \( \sigma \))
  - Cluster shape \( \rightarrow \) Co-variance (symmetric, no dependencies)
- Update rules: Max/Sampling/Density (next slide)

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) ( \rightarrow ) max likelihood</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pick a best</td>
<td>Assign each point to best label ( K )-means: Assign each point to nearest cluster</td>
<td>Viterbi training: label sequence with best path Greedy: Find best motif match in each sequence</td>
</tr>
<tr>
<td>Average all</td>
<td>Assign each point to all labels, probabilistically ( Fuzzy )-means: Assign to all clusters, weighted by proximity</td>
<td>Baum-Welch training: label sequence w all paths (posterior decoding) MEME: Use all positions as a motif occurrence weighted by motif match score</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 N/A: Sample a single label for each position, according to posterior prob. Gibbs sampling: Use one position for the motif, by sampling from the match scores</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

### Distance between clusters

- CD \((x, y)\) = min \(x, k \rightarrow y \rightarrow D(x, y)\) \(\rightarrow \) Single-link method
- CD \((x, y)\) = max \(x, k \rightarrow y \rightarrow D(x, y)\) \(\rightarrow \) Complete-link method
- CD \((x, y)\) = \(x, k \rightarrow y \rightarrow D(x, y)\) \(\rightarrow \) Average-link method
- CD \((x, y)\) = \(D(\text{avg}(x), \text{avg}(y))\) \(\rightarrow \) Centroid method

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

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

Classification problem: Max Probability Class

Select the class that maximizes posterior:

\[
P(Class|Feature) = \frac{\text{Likelihood} \times \text{Prior}}{\text{Posterior} \times \text{Evidence}}
\]

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

= \[\arg\max_C P(Feature|Class) \times P(Class)\]

Scaling the above distribution based on class priors

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:
   - \[P(Class|Feature) = \frac{P(Feature|Class) \times P(Class)}{\text{Likelihood}} \times \text{Prior} = \frac{P(Feature|Class) \times P(Class)}{P(Feature)\text{Evidence}}\]

   - Take probability ratios

Likelihood:

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

P(X|Class1) P(X|Class2)

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

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

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

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

\( P(\text{mito}) = \) 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

Total evidence is

\[ P(\text{Feature}) = \sum P(\text{Feature} | \text{Class}) P(\text{Class}) \]

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:

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

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

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

\[ \Rightarrow P(\text{Feature}) \text{ does not need to be computed for classification} \]

Discriminant Function for selecting Class 1

We can create a convenient representation of the Bayes Decision Rule

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

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

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

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 \textit{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

\textbf{One Simple Approach}

Divide \( X \) values into bins

And then we simply count frequencies

\textbf{Density Estimation}

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

\( \frac{P(X | \text{Class1})}{P(X | \text{Class2})} > 0 \)

There are 13 data points

\( \frac{2/13}{3/13} \)

\( \frac{7/13}{3/13} \)

\( \frac{1/13}{1/13} \)
Distributions Over Many Features

Estimating $P(X_1, X_2, X_3, \ldots, X_8 | \text{Class1})$ 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
   - $P(\text{Class1}) = \frac{13}{23}$
   - $P(\text{Class2}) = \frac{10}{23}$

   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
   - $P(\text{Class1}) = P(\text{Class2})$

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

Combining Multiple Features

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

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

Naïve Bayes Discriminant Function

We are going to make the following assumption:

All features are independent given the class

$P(X_1, X_2, \ldots, X_n | \text{Class}) = P(X_1 | \text{Class})P(X_2 | \text{Class}) \cdots P(X_n | \text{Class})$

We can thus estimate individual distributions for each feature and just multiply them together!

Naïve Bayes Classifier

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

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

As this:

$G(X_1, \ldots, X_7) = \log \frac{\prod P(X_i | \text{Class1}) \cdot P(\text{Class1})}{\prod P(X_i | \text{Class2}) \cdot P(\text{Class2})} > 0$
Binary Classification Errors

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

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

Individual Feature Distributions

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

$P(X_i|Mito), P(X_i|\sim\text{Mito})$

Apply to human proteome: 1,451 predictions (of which 490 are novel predictions)

Classifying A New Protein

For all 8 features

$P(X_i|Mito), P(X_i|\sim\text{Mito})$

Plug these and priors into the discriminant function

$G(X_1, \ldots, X_8) = \log \prod P(X_i|Mito) P(Mito) / \prod P(X_i|\sim\text{Mito}) P(\sim\text{Mito}) > 0$

**IF G>0, we predict that the protein is from class Mito**
Support Vector Machines (SVMs)

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

SVM Formulation

We define a vector $\mathbf{w}$ normal to the separating line.

Assume all data satisfy the following:

$$\mathbf{x}_i \cdot \mathbf{w} - b \geq +1 \text{ for } y_i = +1$$
$$\mathbf{x}_i \cdot \mathbf{w} - b \leq -1 \text{ for } y_i = -1$$

We want to find the separator with the largest margin.

An Optimization Problem

For full derivation, see Burges (1998)

Minimize $L_D = \sum_i \alpha_i - \frac{1}{2} \sum_i \sum_j \alpha_i \alpha_j y_i y_j \mathbf{x}_i \cdot \mathbf{x}_j$
subject to $\sum_i \alpha_i y_i = 0$ and $\alpha_i > 0$

Solving for $\alpha_i$:

$$\alpha_i \left( y_i (\mathbf{x}_i \cdot \mathbf{w} - b) - 1 \right) = 0$$

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:

$$y_{\text{new}} = \text{sign} \left( \mathbf{w} \cdot \mathbf{x}_{\text{new}} - b \right)$$

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, \( R^d \), to other euclidean space, \( H \)

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

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, \( R^d \), to other euclidean space, \( H \)

From previous slide. SVMs only depend on dot product

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

\[
X_i \cdot X_j \rightarrow \Phi(X_i) \cdot \Phi(X_j)
\]

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

Example Kernels

- Linear
  \[
  K(x_i, x_j) = x_i^T x_j
  \]
- Polynomial
  \[
  K(x_i, x_j) = (y x_i^T x_j + r)^d
  \]
- Radial Basis Function
  \[
  K(x_i, x_j) = \exp(-y \|x_i - x_j\|^2)
  \]
- Sigmoid
  \[
  K(x_i, x_j) = \tanh(y x_i^T x_j + r)
  \]

What \( K(X_i, X_j) \) are valid kernels?
Answer given by Mercer's Condition (see Burgess 1998)

Kernels

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

\[
\Phi(X_i) \cdot \Phi(X_j) = \text{scalar!}
\]

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
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

Minimize \( L_2 = \sum a_i - \frac{1}{2} \sum a_i a_j K(x_i, x_j) \)

Step 2 – Test/Classify on new samples

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

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 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

Training Set

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 more 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.
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

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

SVM Approach

Support Vector Machine Classification of Microarray Data

Saman Bhaljee & Ryan Rifkin

Analytical Intelligence, Expository and
The Center for Biological and Computational Learning
The MIT Media Lab
Cambridge, MA 02139
http://www.ai.mit.edu

The Problem: Use the learning from examples paradigm to make class predictions and infer genes involved in
these phenomena from DNA microarray expression data. Specifically, we use a Support Vector Machine (SVM)
classifier [1] to predict cancer morphologies and treatment success and determine over-expressed genes in each.

Motivation:

Problem-Specific Motivation:

A generic approach to classification two types of acute leukemia was introduced in Golub et al. [2]. SVMs have been applied to this problem [4] and also to the problem of predicting bacterial roles of one factor-tested [5-7].

Approach: We used a SVM classifier to discriminate between two types of leukemia. The output of trained SVM is a real number in [-1,1]. The particular application of importance is to be able to test points for which the classifier is not confident enough. We introduce a confidence interval on the output of the SVM that allows us to report points with low confidence with a high confidence.

It is also important in the application of which genes are important for the classification. We have preliminary results for a feature selection algorithm for SVM classifiers.

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

Figure 9.6 The signed distance, $f(x)$, 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).
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Topics</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Introduction to gene expression analysis</td>
<td>Technology: microarrays vs. RNAseq, Resulting data matrices, Supervised (Clustering) vs. unsupervised (classification) learning</td>
</tr>
<tr>
<td>3.</td>
<td>Hierarchical Clustering (clustering by agglomeration)</td>
<td>Basic algorithm, Distance measures, Evaluating clustering results</td>
</tr>
<tr>
<td>4.</td>
<td>Naïve Bayes classification (generative approach to classification)</td>
<td>Discriminant function: class priors, and class-conditional distributions, Training and testing, Combine mult features, Classification in practice</td>
</tr>
<tr>
<td>5.</td>
<td>(optional) Support Vector Machines (discriminative approach)</td>
<td>SVM formulation, Margin maximization, Finding the support vectors, Non-linear discrimination, Kernel functions, SVMs in practice</td>
</tr>
</tbody>
</table>