6.047/6.878/HST.507
Computational Biology: Genomes, Networks, Evolution

Lecture 7

Gene expression analysis: Clustering and Classification

Module II: Gene expression analysis and networks

- Computational foundations:
  - Unsupervised Learning: Expectation Maximization
  - Supervised learning: generative/discriminative models
  - Read mapping, significance testing, splice graphs
  - Folding: DP self-alignment, Context Free grammars
- Biological frontiers:
  - L6: RNA-Seq analysis, quantifying transcripts, isoforms
  - L7: Gene expression analysis: cluster genes/conditions
  - L8: Networks I: Bayesian Inference, deep learning
  - L9: Networks II: Network structure, spectral methods

Today: Gene Expression Clustering & Classification

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

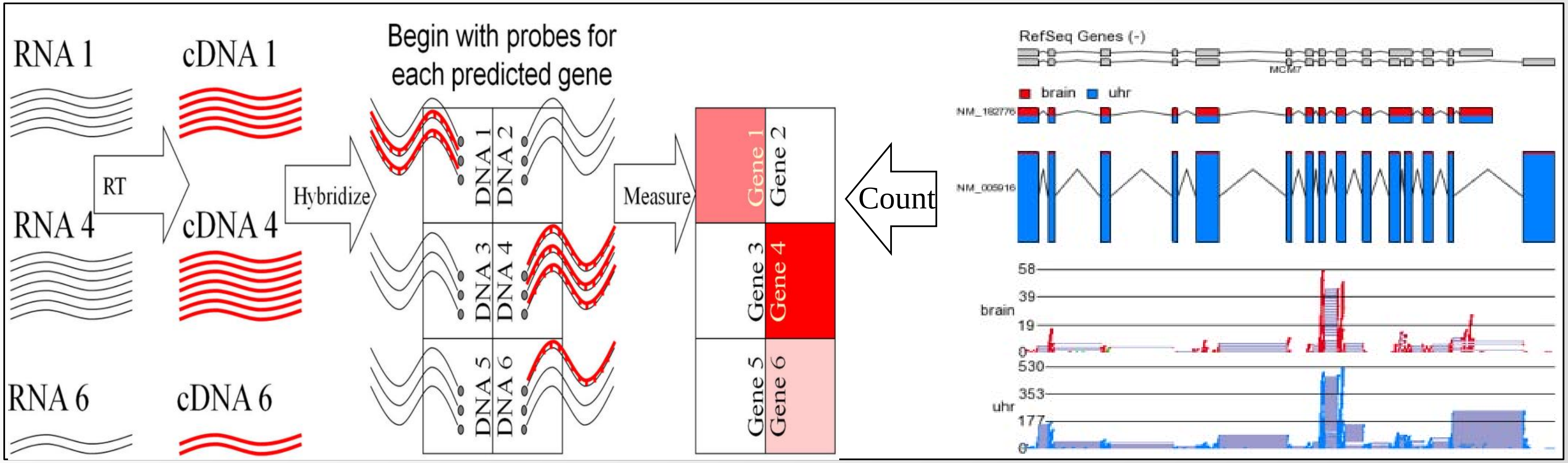
2. **K-means clustering (clustering by partitioning)**
   - Algorithmic formulation: Update rule, optimality criterion. Fuzzy k-means.
   - Machine learning formulation: Generative models, Expectation Maximization.

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

RNA-Seq: De novo tx reconstruction / quantification



Microarray technology

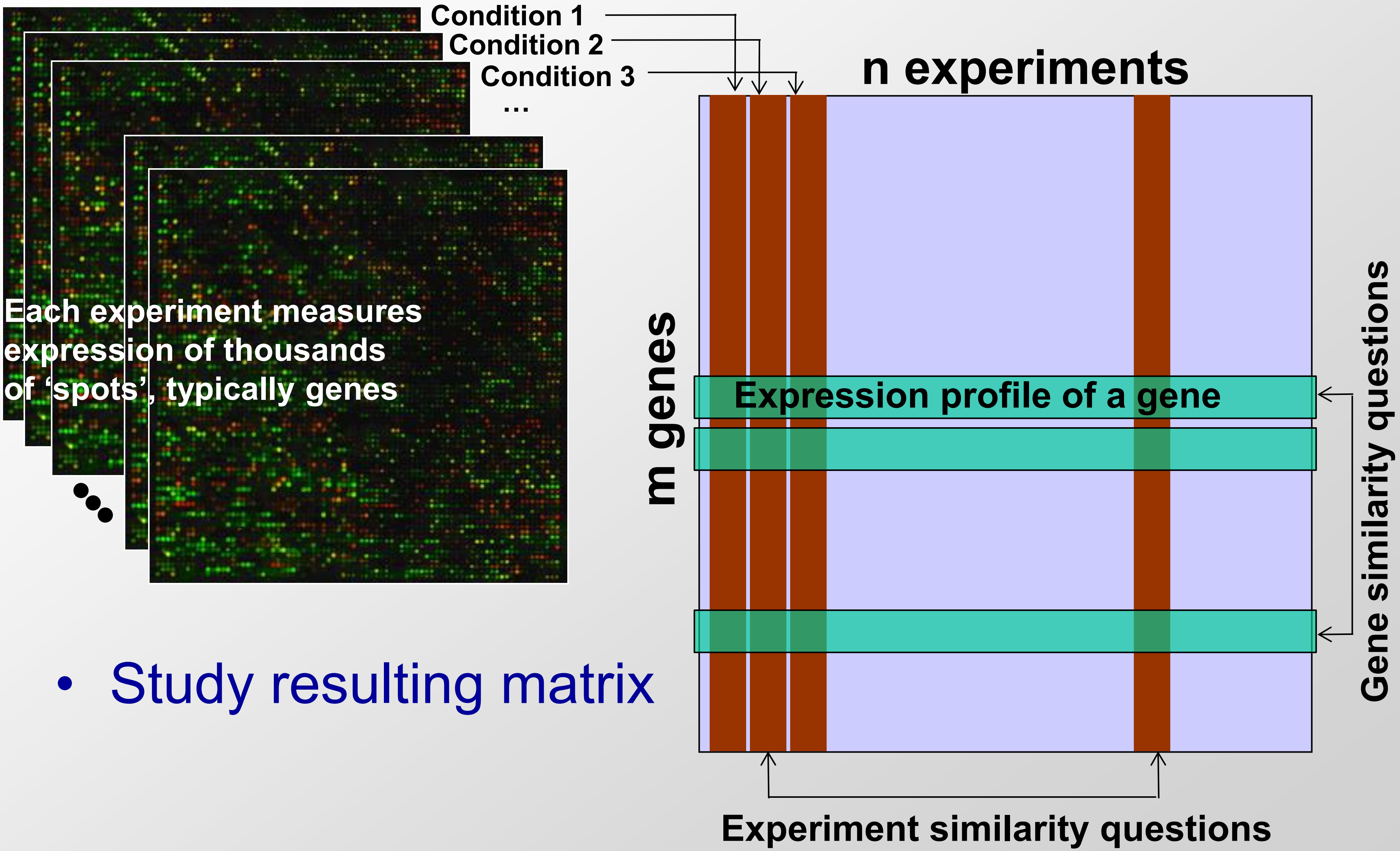
- Synthesize DNA probe array, complementary hybridization
- Variations:
  - One long probe per gene
  - Many short probes per gene
  - Tiled k-mers across genome
- Advantage:
  - Can focus on small regions, even if few molecules / cell

RNA-Seq technology:

- Sequence short reads from mRNA, map to genome
- Variations:
  - Count reads mapping to each known gene
  - Reconstruct transcriptome *de novo* in each experiment
- Advantage:
  - Digital measurements, de novo

Expression Analysis Data Matrix

- Measure 20,000 genes in 100s of conditions



Condition 1

Condition 2

Condition 3

…

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

n experiments

m genes

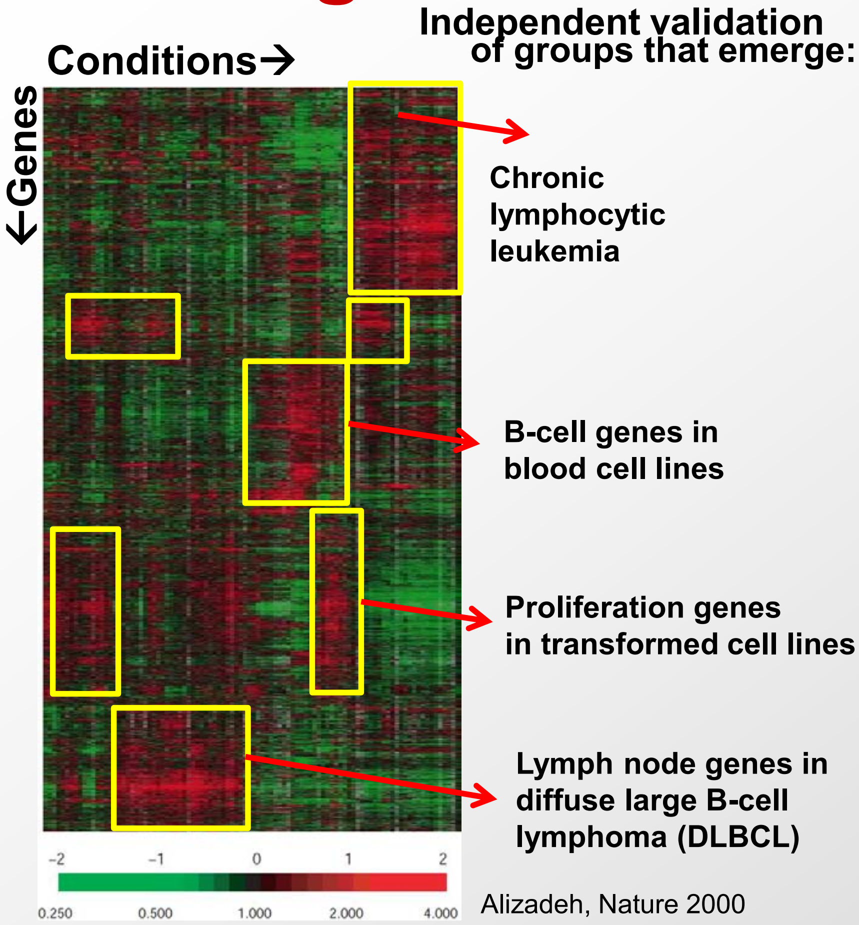
Expression profile of a gene

Gene similarity questions

Experiment similarity questions

- Study resulting matrix

Clustering vs. Classification



Independent validation of groups that emerge:

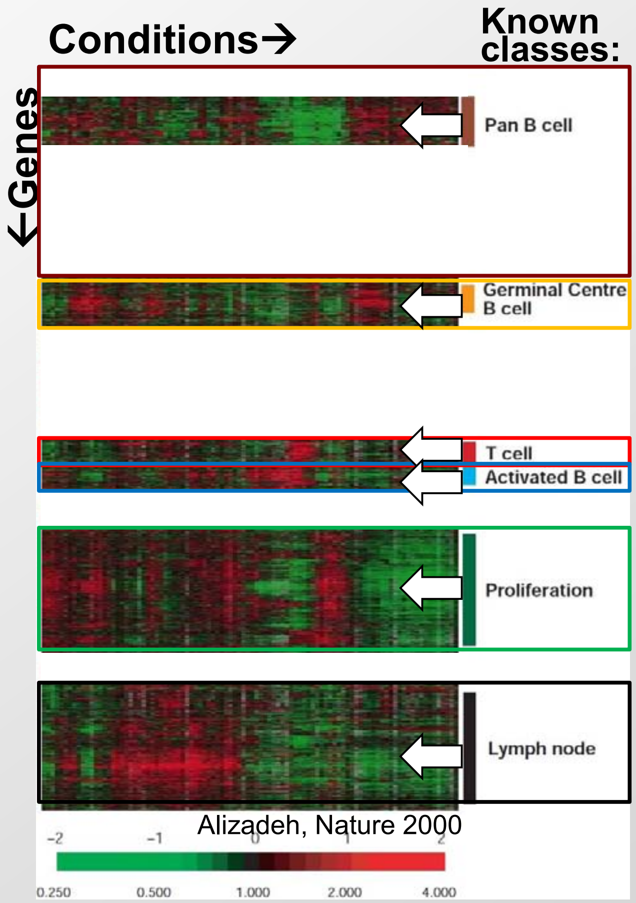
Chronic lymphocytic leukemia

B-cell genes in blood cell lines

Proliferation genes in transformed cell lines

Lymph node genes in diffuse large B-cell lymphoma (DLBCL)

Alizadeh, Nature 2000



Known classes:

Alizadeh, Nature 2000

Courtesy of Macmillan Publishers Limited. Used with permission.
Source: Alizadeh, Ash A., Michael B. Eisen, R. Eric Davis, Chi Ma, Izidore S. Lossos, Andreas Rosenwald, Jennifer C. Boldrick et al. "Distinct types of diffuse large B-cell lymphoma identified by gene expression profiling." Nature 403, no. 6769 (2000): 503-511.

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

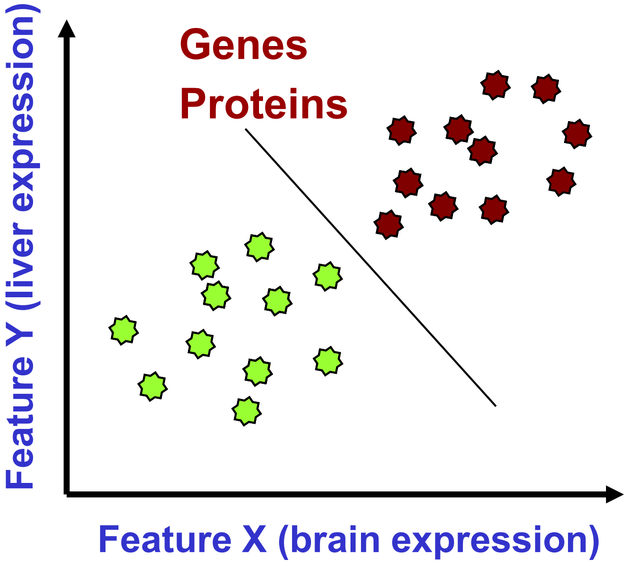
- **Unsupervised learning**

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

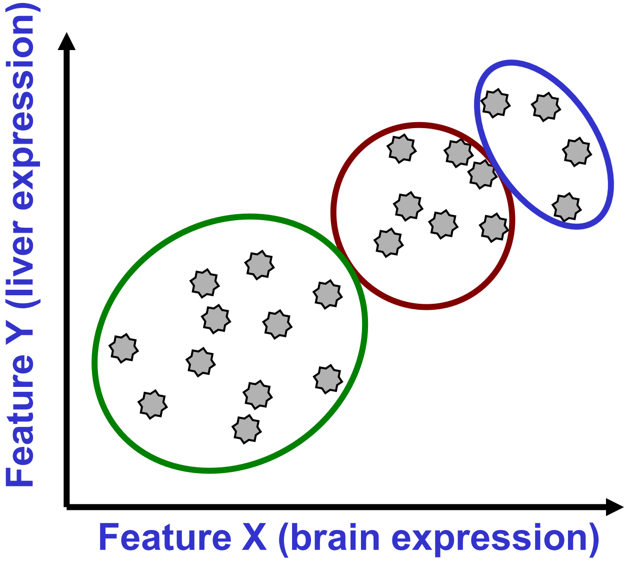
- **Supervised learning**

Clustering vs Classification

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



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



Two approaches to clustering

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

- Agglomerative (e.g. hierarchical clustering)
  - A set of nested clusters organized as a hierarchy

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)**
   - Algorithmic formulation: Update rule, optimality criterion. Fuzzy k-means.
   - 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

K-Means Clustering

The Basic Idea

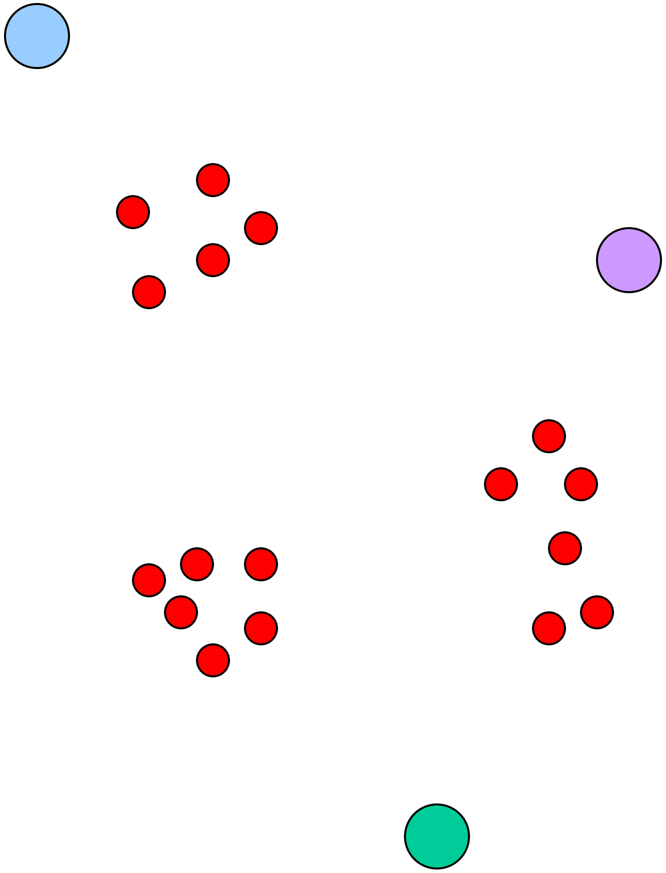
- Assume a fixed number K of clusters
- Partition points into K compact clusters

The Algorithm

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

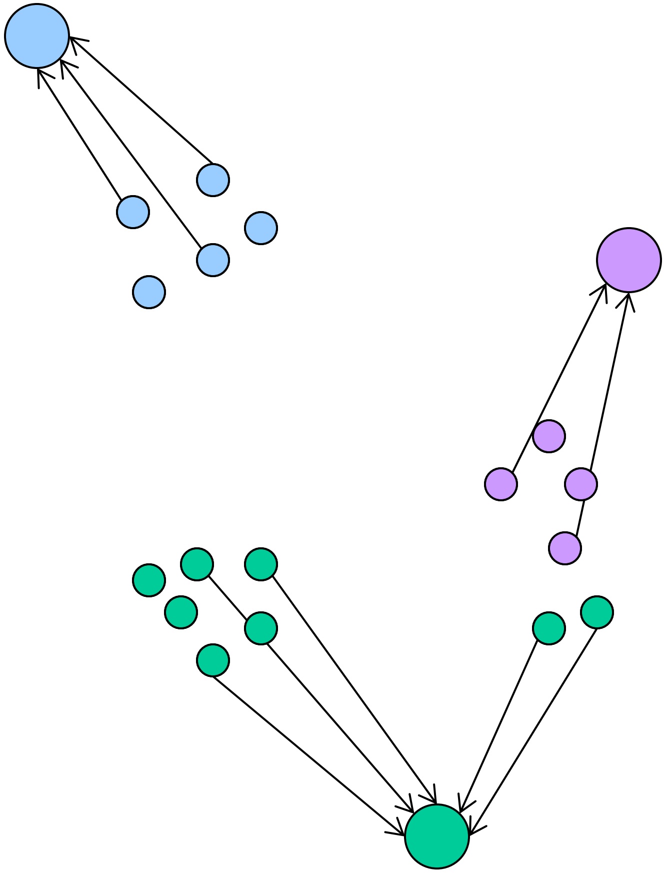
K-Means Algorithm Example

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



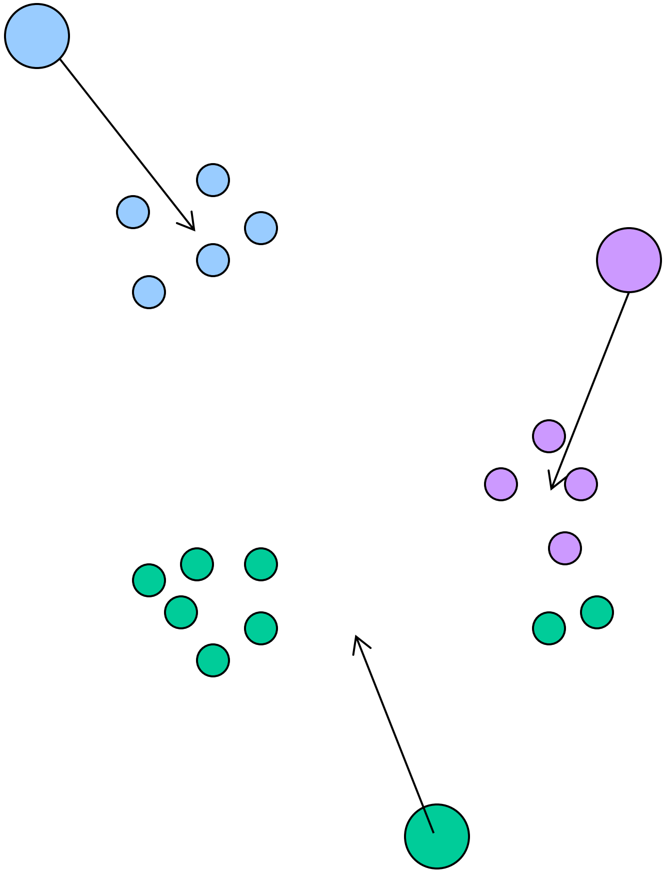
K-Means Algorithm Example

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



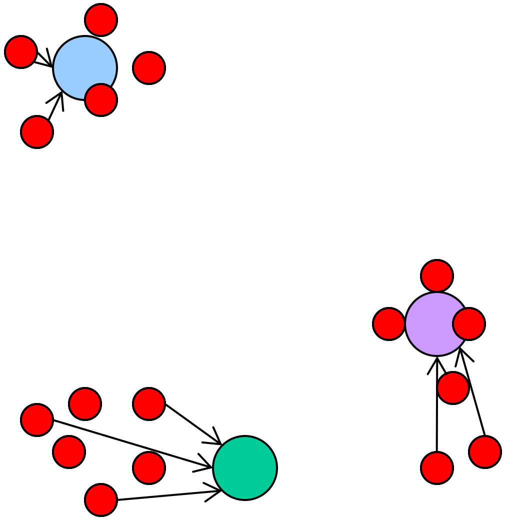
K-Means Algorithm Example

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



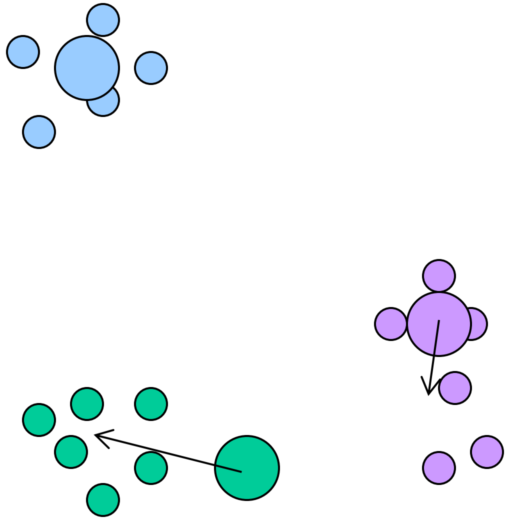
K-Means Algorithm Example

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



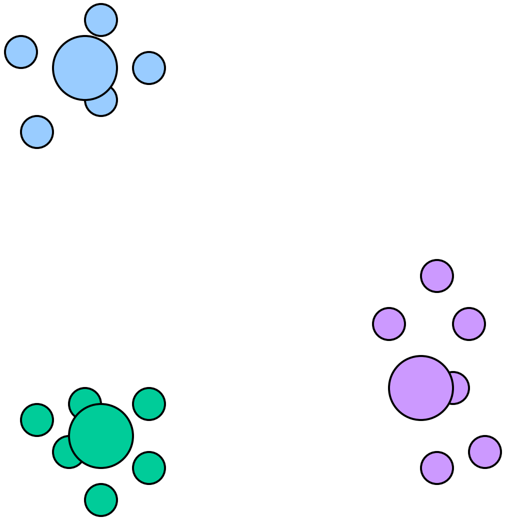
K-Means Algorithm Example

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

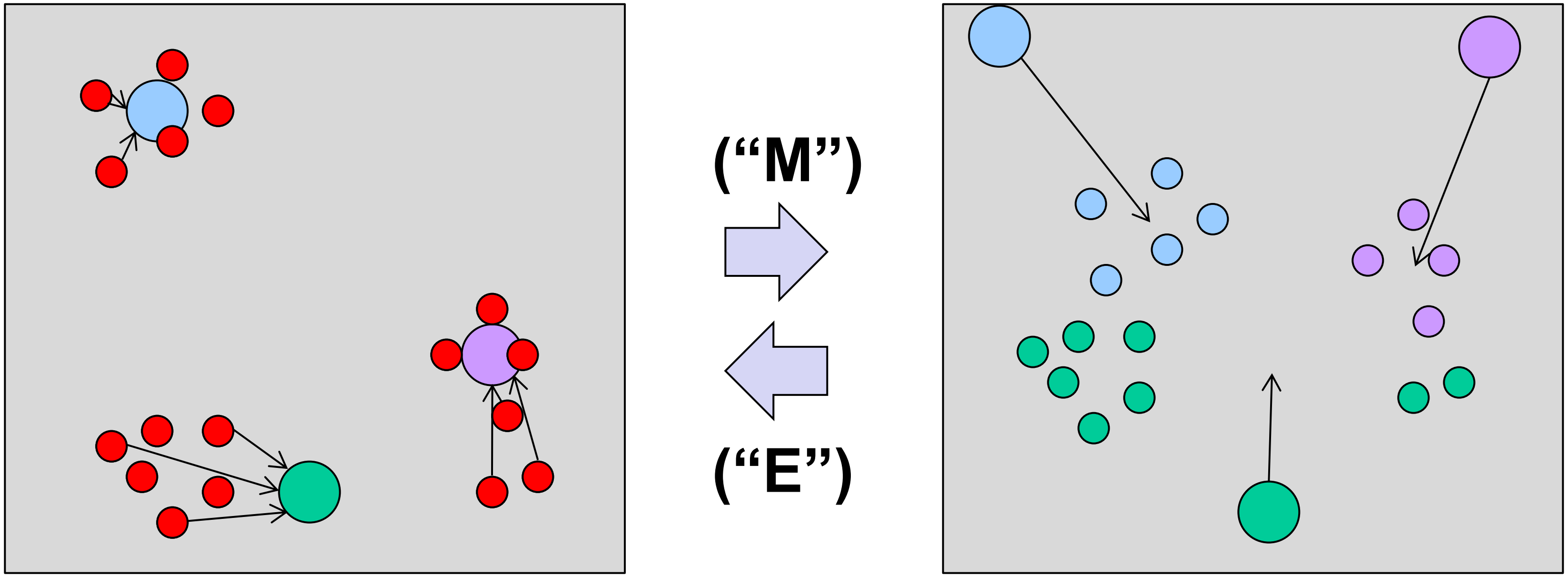


K-Means Algorithm Example

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



K-means update rules



("M")

("E")

Re-assign each point $\mathbf{x}_i$ to nearest center k

➔ Minimize distance from $\mathbf{x}_i$ to $\boldsymbol{\mu}_k$:

$$d_{i,k} = \left(\mathbf{x}_i - \boldsymbol{\mu}_k\right)^2$$

Update center $\boldsymbol{\mu}_k$ to the mean of the points assigned to it:

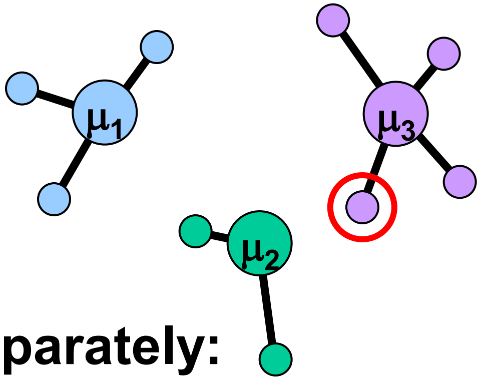
$$\boldsymbol{\mu}_k(n+1) = \sum_{\mathrm{x_i\ \ with\ label\ j}} \frac{\mathbf{x}_\mathrm{i}}{\left|\mathrm{x}^\mathrm{k}\right|}$$

where: $\left|\mathrm{x}^\mathrm{k}\right| = \#\mathbf{x}_\mathrm{i}$ 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

$$\mathrm{COST}(\mathbf{x}_1,\mathbf{x}_2,\mathbf{x}_3,\ldots,\mathbf{x}_n)=\sum_{\boldsymbol{\mu}_k}\sum_{\mathbf{x}_i\text{with label k}}(\mathbf{x}_\mathrm{i}-\boldsymbol{\mu}_k)^2$$



To achieve this, minimize each cluster term separately:

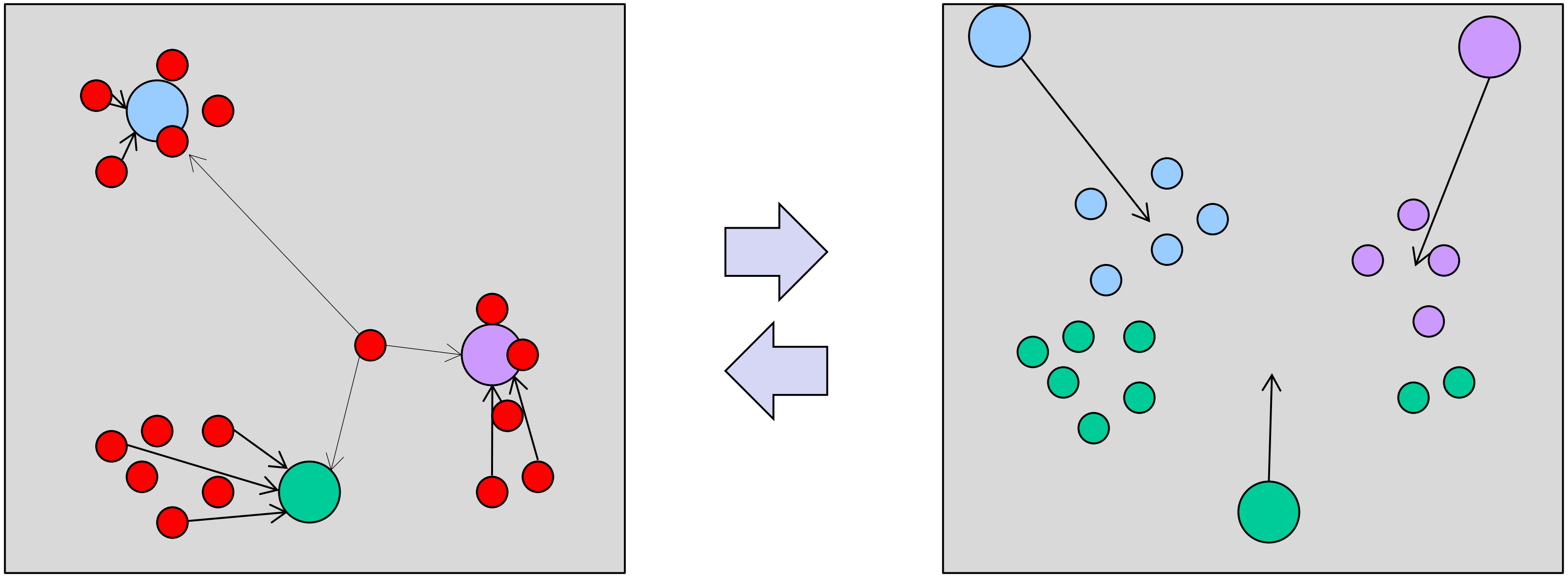
$$\sum_{\mathbf{x}_i\text{with label k}}(\mathbf{x}_\mathrm{i}-\boldsymbol{\mu}_k)^2=\sum_{\mathbf{x}_i\text{with label k}}\mathbf{x}_\mathrm{i}^2-2\mathbf{x}_i\mathbf{u}_k+\boldsymbol{\mu}_k^2=\sum\mathbf{x}_\mathrm{i}^2-\mathbf{u}_k\sum 2\mathbf{x}_i+\left|\mathbf{x}^k\right|\mathbf{u}_k^2$$

Optimum $\mathbf{u}_k=\sum_{\mathbf{x}_i\text{with label k}}\frac{\mathbf{x}_i}{\left|\mathbf{x}^k\right|}$, the centroid

However: Some points can be almost halfway between two centers ➔ Assign partial weights

⇨ **Fuzzy K-means**

Fuzzy K-means update rule



Re-assign each point $\mathbf{x}_i$

to **all** centers, **weighted by distance**

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

$$P(\text{label K} \mid \mathbf{x}_i, \boldsymbol{\mu}_k)$$

Update center $\boldsymbol{\mu}_k$ to the **weighted mean** of the points assigned to it:

$$\boldsymbol{\mu}_k(n+1) = \sum_{\mathrm{x_i}\ \text{with label j}} \mathbf{x}_i\, P(\boldsymbol{\mu}_k \mid \mathbf{x}_i)^b \Big/ \sum_{\mathrm{x_i}\ \text{with label j}} P(\boldsymbol{\mu}_k \mid \mathbf{x}_i)^b$$

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

$$P(\text{label K} \mid \mathbf{x}_i, \boldsymbol{\mu}_k) = \begin{cases} 1 & \text{if } \mathbf{x}_i \text{ is closest to } \boldsymbol{\mu}_k \\ 0 & \textbf{otherwise} \end{cases}$$

Today: Gene Expression Clustering & Classification

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

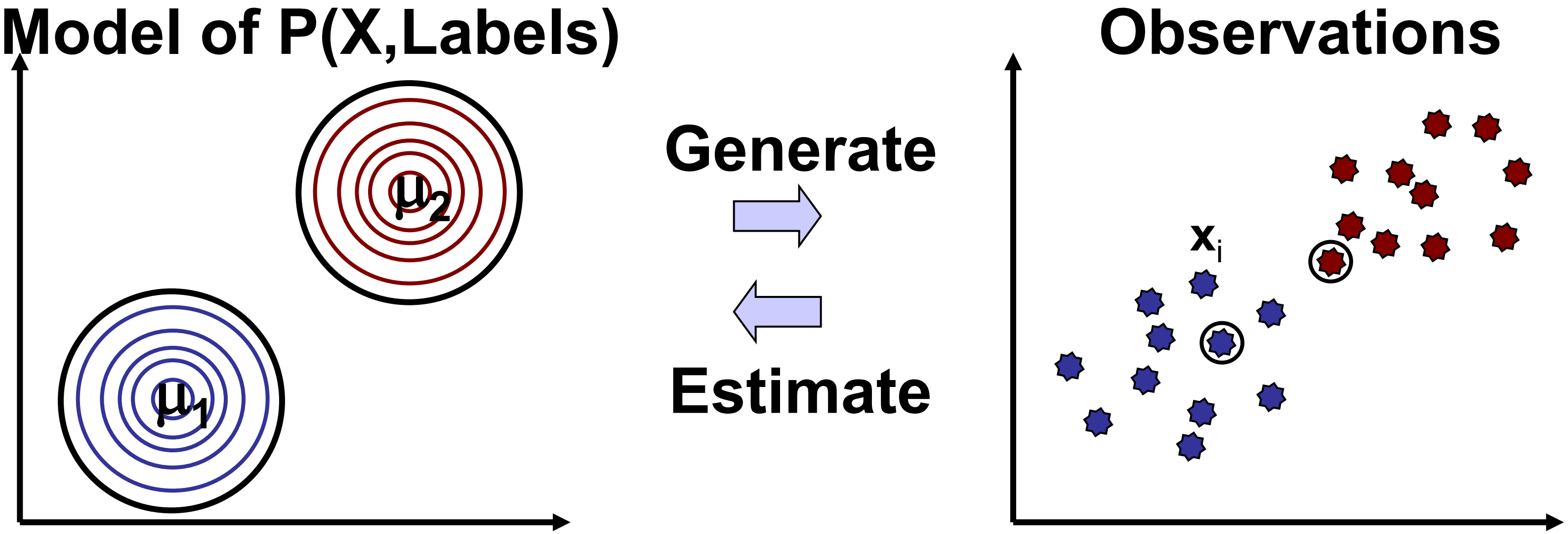
2. **K-means clustering (clustering by partitioning)**
   - Algorithmic formulation: Update rule, optimality criterion. Fuzzy k-means.
   - Machine learning formulation: Generative models, Expectation Maximization.

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

K-Means as a Generative Model



Model of P(X,Labels)

Generate

Estimate

Observations

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

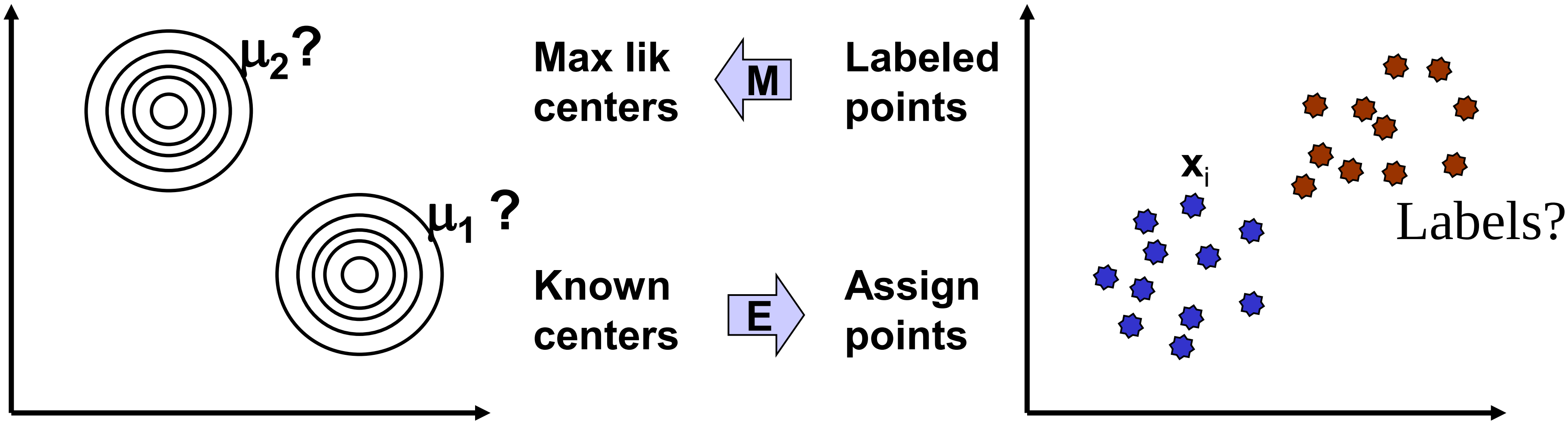
$$P\left(\mathbf{x}_i \mid \mathbf{u}_j\right) = \frac{1}{\sqrt{2\pi}} \exp\left\{ -\frac{\left(\mathbf{x}_i - \mathbf{u}_j\right)^2}{2} \right\}$$

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

EM solution: iteratively estimate one from the other

E step: If centers are known ➔ Estimate memberships

M step: If assignments known ➔ Compute centroids



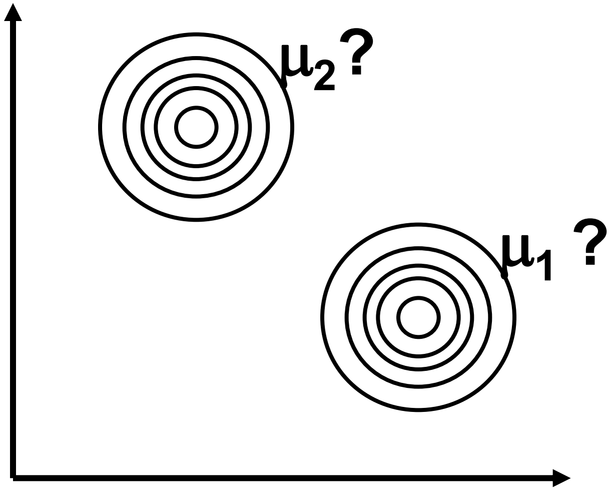
Max lik centers ⇐ M ⇐ **Labeled points**

Known centers ⇒ E ⇒ **Assign points**

Choose μ_k and labels that maximize P(data|model)

Solution is exactly the k-means algorithm!

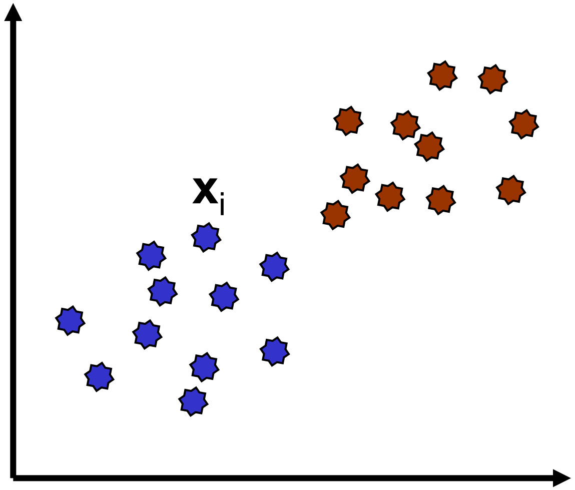
M step: assignments known ➔ compute centroids



Max lik centers

M

Labeled points



Choose μ_k and labels that maximize P(data|model)

$$\boxed{\arg\max_{\boldsymbol{\mu}}} \left\{\log \prod_{\mathrm{i}} \mathrm{P}\left(\mathbf{x}_i \,|\, \boldsymbol{\mu}\right)\right\} = \arg\max_{\boldsymbol{\mu}} \sum_i \left\{-\frac{1}{2}\left(\mathbf{x}_i - \mathbf{u}\right)^2 + \log\left(\frac{1}{\sqrt{2\pi}}\right)\right\}$$

Seeking the max likelihood estimate of the cluster mean

$$= \arg\min_{\boldsymbol{\mu}} \sum_i \boxed{\left(\mathbf{x}_i - \mathbf{u}\right)^2}$$

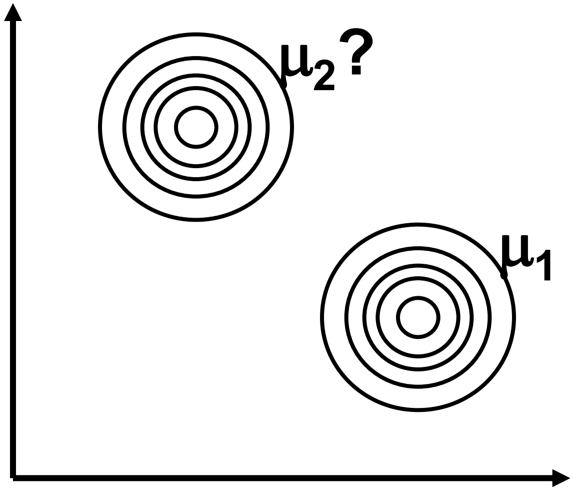
Solution is the centroid of the x_i

Equivalent

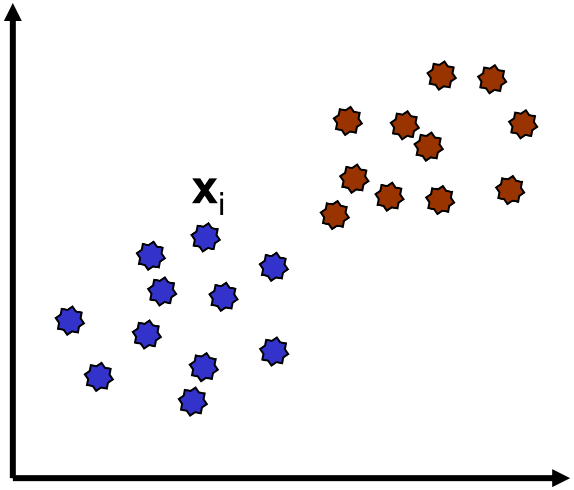
EM solution

K-means solution

E step: centers known ➔ Estimate memberships



Known centers E **Assign points**



Choose μ_k and labels that maximize P(data|model)

$$\boxed{\underset{k}{\arg\max}}\, P_k(\mathbf{x}_i \mid \boldsymbol{\mu}_i) = \underset{k}{\arg\max}\ \frac{1}{\sqrt{2\pi}}\exp\left\{-\frac{(\mathbf{x}_i-\mathbf{u}_k)^2}{2}\right\} = \underset{k}{\arg\min}\boxed{(\mathbf{x}_i-\mathbf{u}_k)^2}$$

Seeking the label k that maximizes likelihood of point

Solution is the nearest center

Equivalent

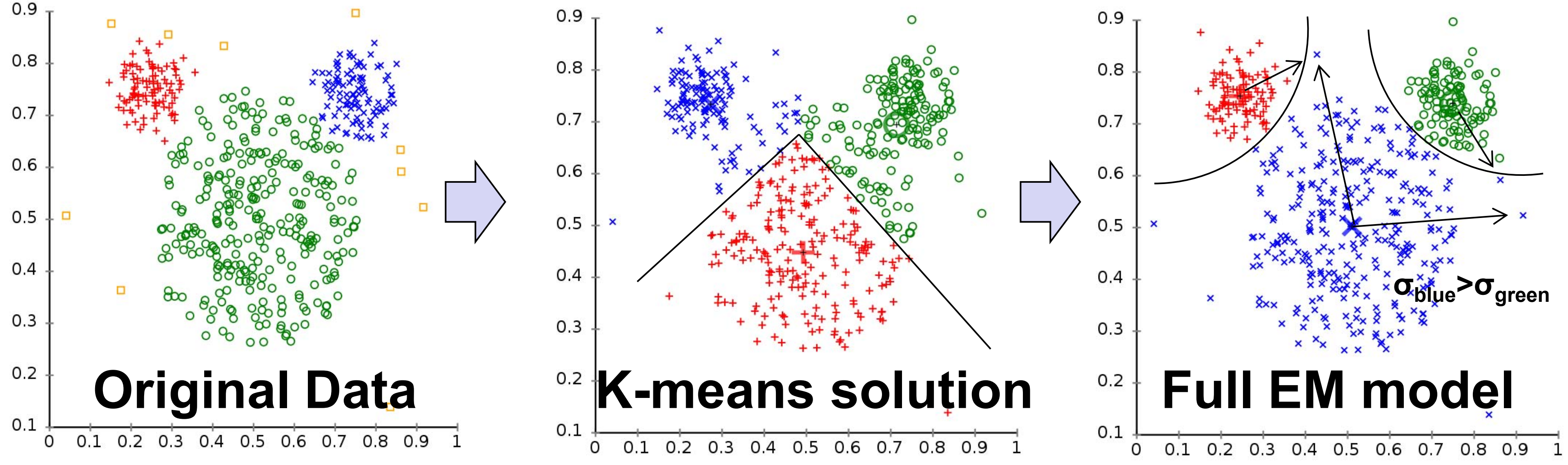
EM solution ⟺ **K-means solution**

Algorithmic vs. machine learning formulations

	K-means		Fuzzy K-means	
	algorithmic formulation	probabilistic interpretation	algorithmic formulation	probabilistic interpretation
Initialization	Initialize K centers $\boldsymbol{\mu}_k$	Initialize model parameters	Initialize K centers $\boldsymbol{\mu}_k$	Initialize model parameters
E-step: Estimate prob of hidden labels (point assignments to classes)	Assign $\mathbf{x}_i$ label of nearest center distance $d_{i,k} = (\mathbf{x}_i - \boldsymbol{\mu}_k)^2$	Estimate most likely missing label given previous parameters	Calculate probability of membership for each point to each class $\mathrm{P(label\ K} \mid \mathbf{x}_i, \boldsymbol{\mu}_k)$	Estimate probability over missing labels given previous parameters
M-step: Update params to max likelihood estimates given assignments	Move $\boldsymbol{\mu}_k$ to centroid of all points with that label	Choose new max likelihood params given points in label	Move $\boldsymbol{\mu}_k$ to weighted centroid of all points, each weighted by P(label)	Choose new params to maximize expected likelihood given label estimates
Iteration	Iterate	Iterate	Iterate	Iterate

P(x|Model) *guaranteed* to increase each iteration of EM algo

EM is much more general than fuzzy K-means



	K-means solution	EM generalization
Cluster sizes	**Uniform** priors	Class priors $P(class_i)$
Spread of points	**Unit** distance function	*Gaussian* $(\mu_i, \boldsymbol{\sigma_i})$
Cluster shape	**Symmetric**, x-y indpt	Co-variance matrix $q_{jk} = \frac{1}{N}\sum_{i=1}^{N}(x_{ij}-\bar{x}_j)(x_{ik}-\bar{x}_k)$
Label assignment	K-means: Pick **max** Fuzzy: Full **density**	EM: Full **density** Gibbs: **sample** posterior

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

Update rule	**Update assignments (E step) ➔ Estimate hidden labels**	**Algorithm implementing E step in each of the three settings**			**Update model parameters (M step) ➔ max likelihood**
		Expression clustering	**HMM learning**	**Motif discovery**	
The hidden label is:		Cluster labels	State path π	Motif positions	
Pick a best	Assign each point to best label	**K-means:** Assign each point to nearest cluster	**Viterbi training:** label sequence with best path	**Greedy:** Find best motif match in each sequence	Average of those points assigned to label
Average all	Assign each point to all labels, probabilistically	**Fuzzy K-means:** Assign to all clusters, weighted by proximity	**Baum-Welch training:** label sequence w all paths (posterior decoding)	**MEME:** Use all positions as a motif occurrence weighed by motif match score	Average of all points, weighted by membership
Sample one	Pick one label at random, based on their relative probability	**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	Average of those points assigned to label(a sample)

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)**
   - Algorithmic formulation: Update rule, optimality criterion. Fuzzy k-means.
   - 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

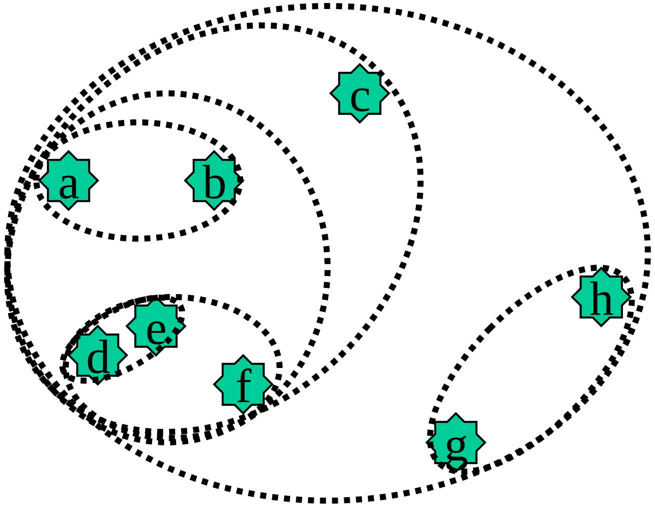
5. **(optional) Support Vector Machines (discriminative approach)**
   - SVM formulation, Margin maximization, Finding the support vectors
   - Non-linear discrimination, Kernel functions, SVMs in practice

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

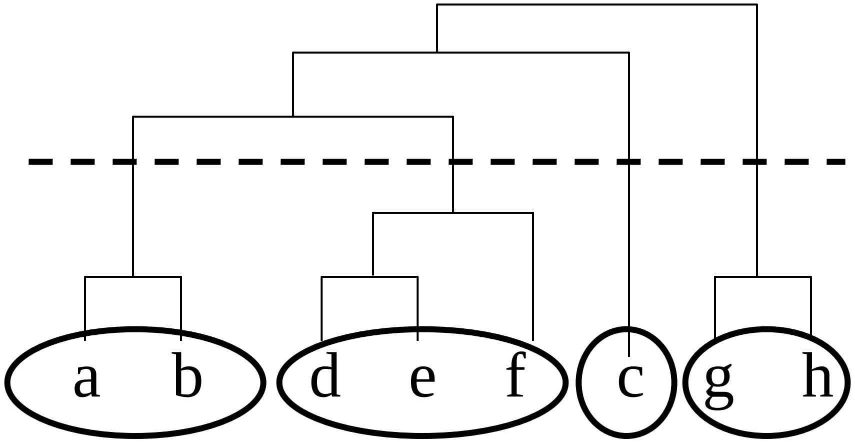
Most widely used algorithm for expression data



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

Phylogeny (UPGMA)

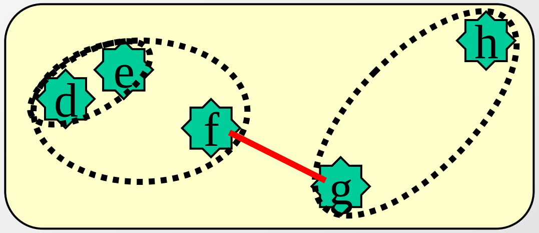
Unweighted **P**air **G**roup **M**ethod with **A**rithmetic-mean



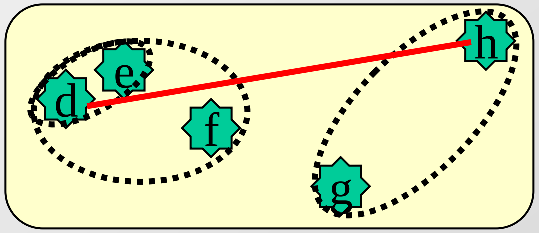
Select a “cut level” to create disjoint clusters

Distance between clusters

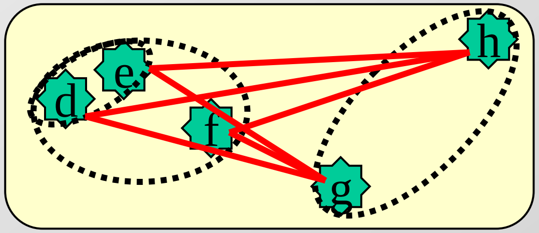
- $CD(X,Y)=\min_{x \in X,\ y \in Y} D(x,y)$
  Single-link method



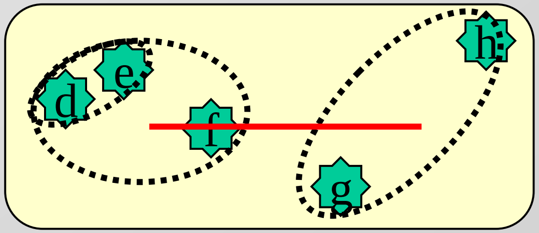
- $CD(X,Y)=\max_{x \in X,\ y \in Y} D(x,y)$
  Complete-link method



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



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



Cluster distance affects both results and runtime

Point-to-point (Dis)Similarity Measures

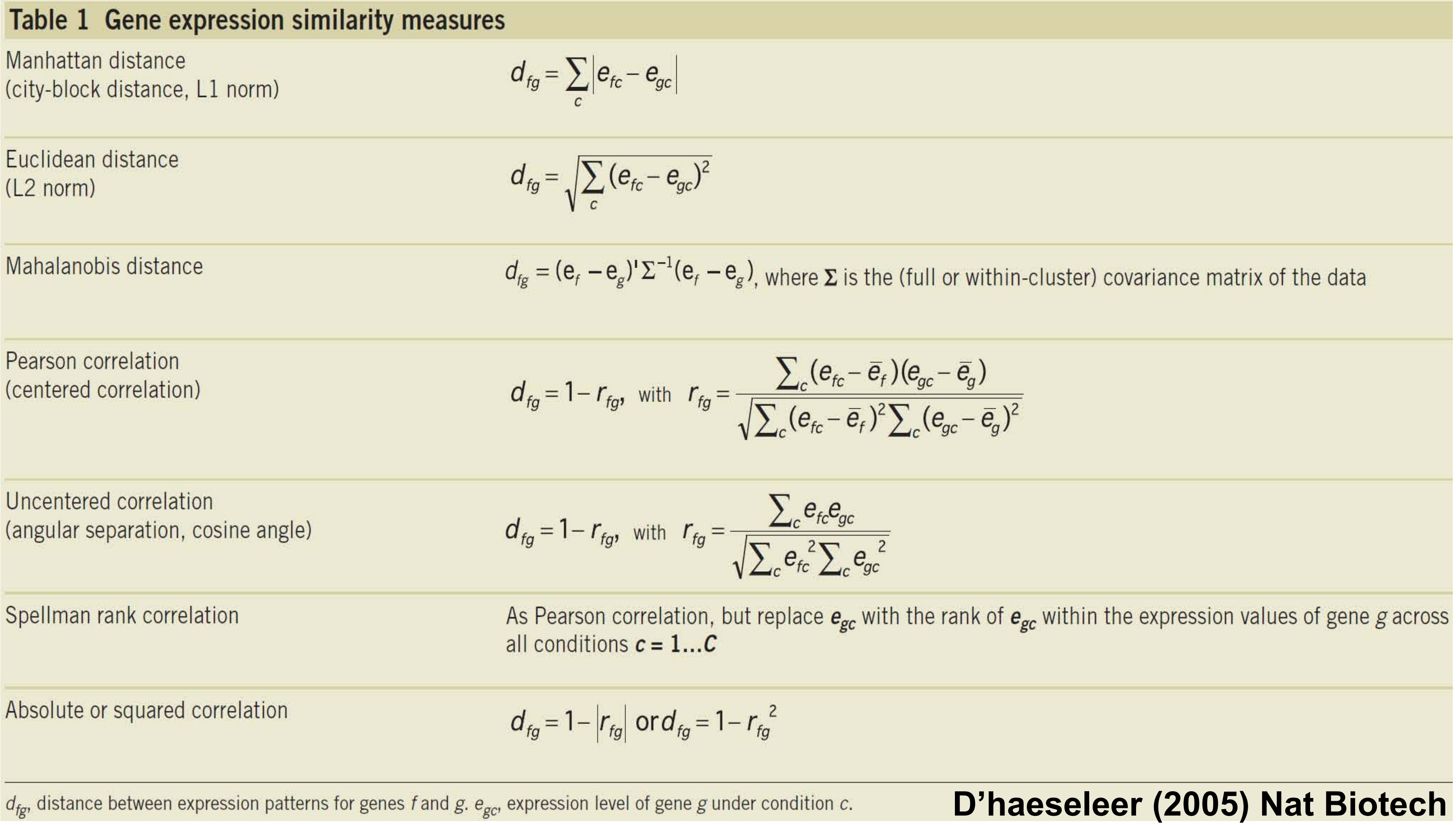


Table 1 Gene expression similarity measures

Manhattan distance (city-block distance, L1 norm)	$d_{fg} = \sum_c \lvert e_{fc} - e_{gc} \rvert$
Euclidean distance (L2 norm)	$d_{fg} = \sqrt{\sum_c (e_{fc} - e_{gc})^2}$
Mahalanobis distance	$d_{fg} = (e_f - e_g)'\Sigma^{-1}(e_f - e_g)$, where Σ is the (full or within-cluster) covariance matrix of the data
Pearson correlation (centered correlation)	$d_{fg} = 1 - r_{fg}$, with $r_{fg} = \frac{\sum_c (e_{fc} - \bar{e}_f)(e_{gc} - \bar{e}_g)}{\sqrt{\sum_c (e_{fc} - \bar{e}_f)^2 \sum_c (e_{gc} - \bar{e}_g)^2}}$
Uncentered correlation (angular separation, cosine angle)	$d_{fg} = 1 - r_{fg}$, with $r_{fg} = \frac{\sum_c e_{fc} e_{gc}}{\sqrt{\sum_c e_{fc}{}^2 \sum_c e_{gc}{}^2}}$
Spellman rank correlation	As Pearson correlation, but replace $\mathbf{e}_{gc}$ with the rank of $\mathbf{e}_{gc}$ within the expression values of gene *g* across all conditions $\mathbf{c = 1 \ldots C}$
Absolute or squared correlation	$d_{fg} = 1 - \lvert r_{fg} \rvert$ or $d_{fg} = 1 - r_{fg}{}^2$

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

D'haeseleer (2005) Nat Biotech

Courtesy of Macmillan Publishers Limited. Used with permission.
Source: D'haeseleer, Patrik. "How does gene expression clustering work?."
Nature biotechnology 23, no. 12 (2005): 1499-1502.

Cluster-to-cluster distance as a function of point-to-point

Today: Gene Expression Clustering & Classification

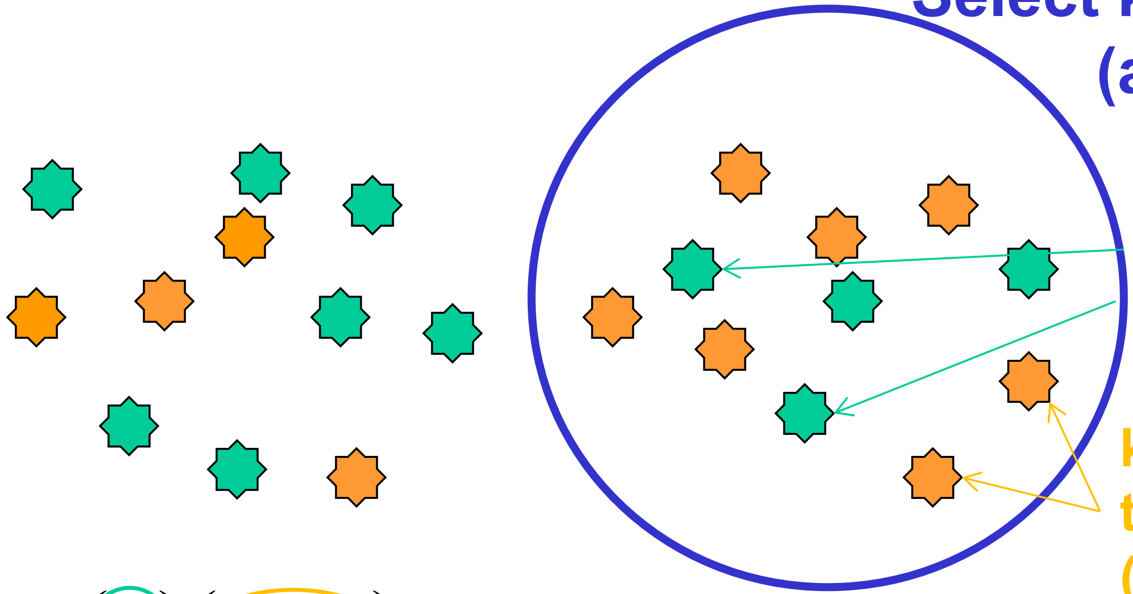
1. **Introduction to gene expression analysis**
   - Technology: microarrays vs. RNAseq. Resulting data matrices
   - Supervised (Clustering) vs. unsupervised (classification) learning
2. **K-means clustering (clustering by partitioning)**
   - Algorithmic formulation: Update rule, optimality criterion. Fuzzy k-means.
   - 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

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



Select k elements (at random)

m happen to be + (out of p +'s)

k-m happen to be - (out of N-p -'s)

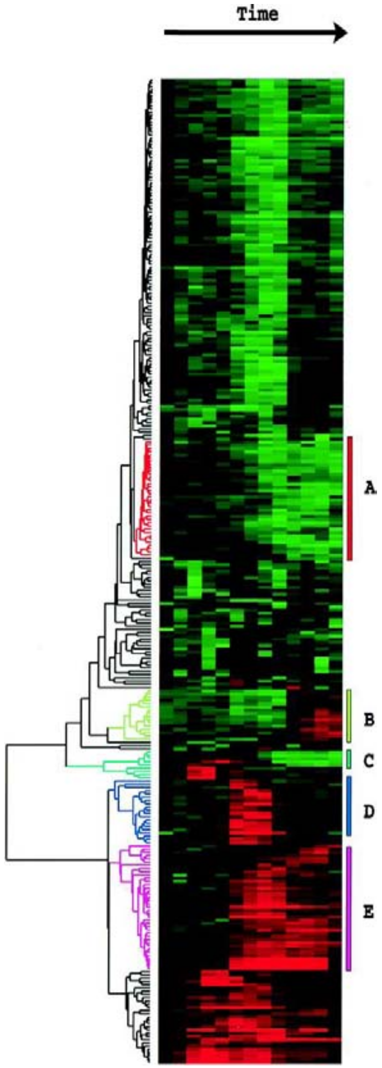
$$P(pos \geq r) = \sum_{m \geq r} \frac{\binom{p}{m}\binom{N-p}{k-m}}{\binom{N}{k}}$$

P-value of uniformity in computed cluster

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

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

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

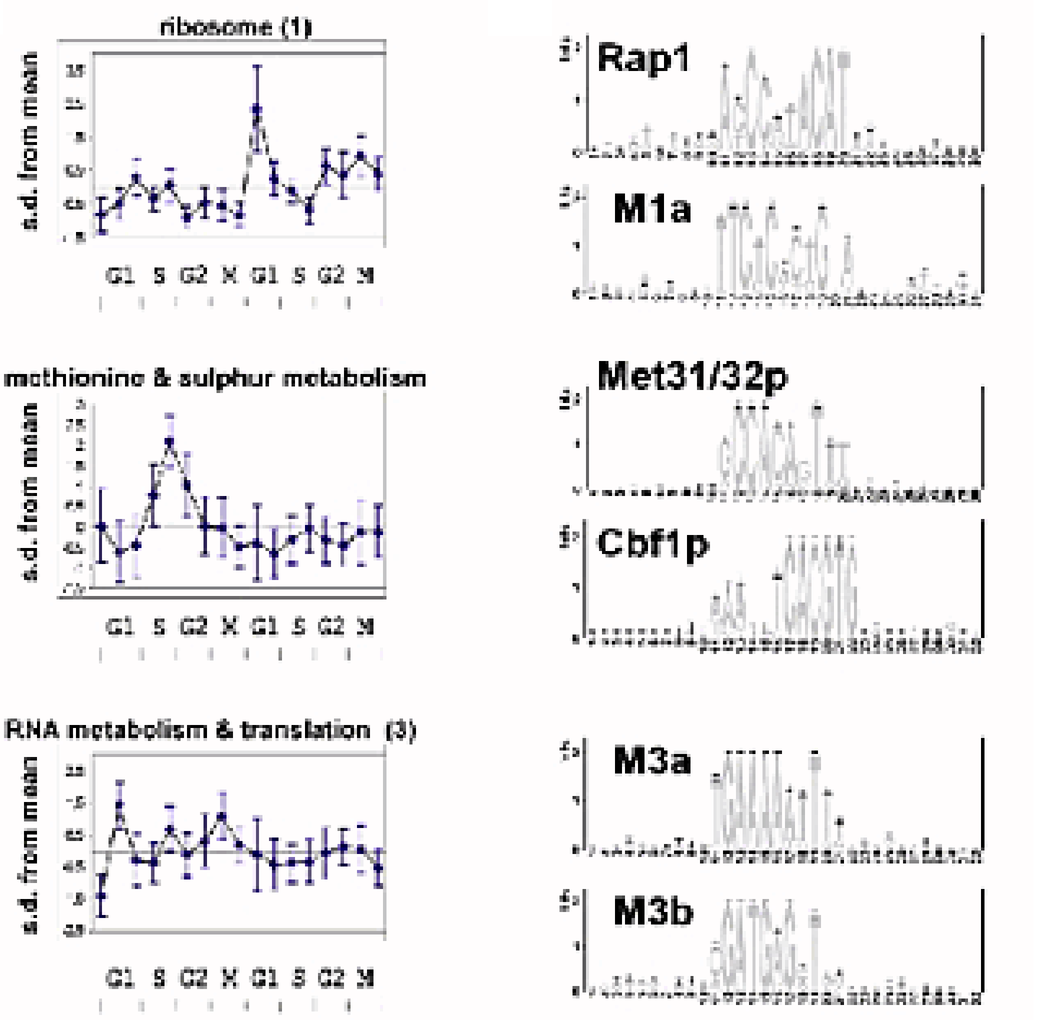
(E) Wound healing

Eisen, Michael et al. "Cluster Analysis and Display of Genome-wide Expression Patterns." PNAS 95, no. 25 (1998): 14863-14868.

(Eisen (1998) PNAS)

Evaluation based on motif content

Expression from 15 time points during yeast cell cycle



Courtesy of Nature Publishing Group. Used with permission.
Source: Tavazoie, Saeed et al. "Systematic determination of genetic network architecture." Nature Genetics 22, no. 3 (1999): 281-285.

Tavazoie & Church (1999)

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)**
   - Algorithmic formulation: Update rule, optimality criterion. Fuzzy k-means.
   - Machine learning formulation: Generative models, Expectation Maximization.

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

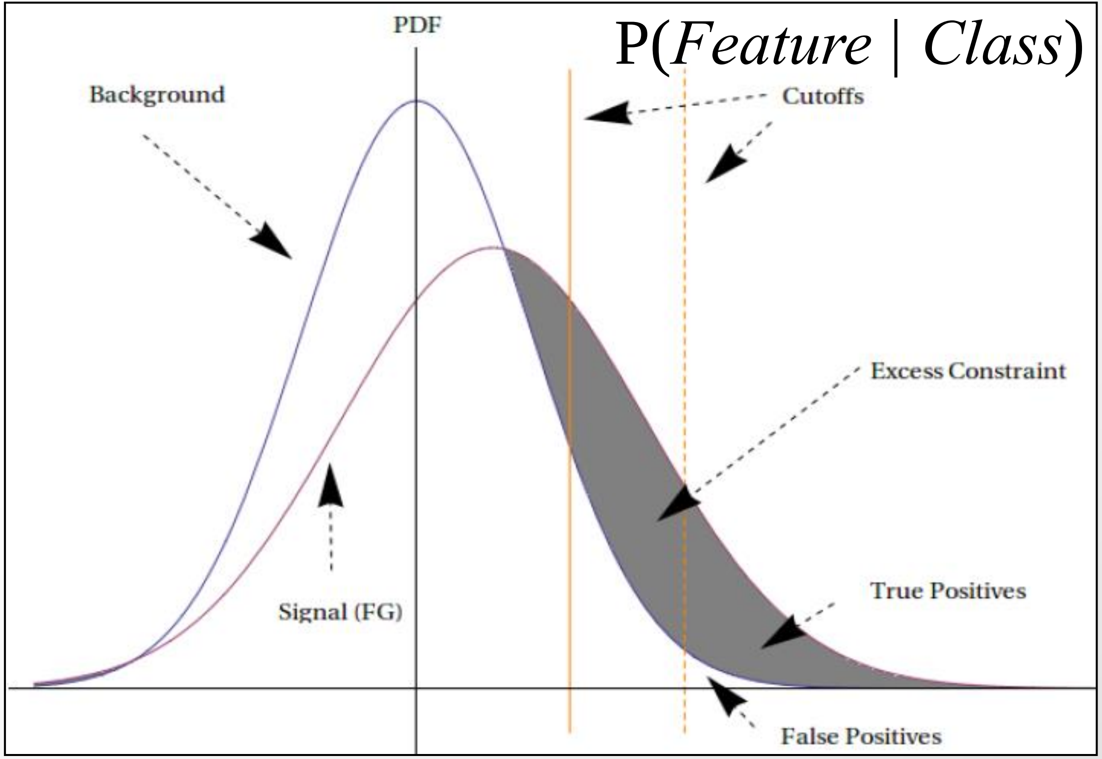
4. **Naïve Bayes classification (generative approach to classification)**
   - Discriminant function: class priors, and class-conditional distributions
   - Training and testing, Combine mult features, Classification in practice

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

Two Approaches to Classification

- Generative
  - Bayesian Classification (e.g. Naïve Bayes)
  - Pose classification problem in prob terms
  - Model feature distribution in different classes
  - Use probability calculus for making decisions
- Discriminative
  - E.g. Support Vector Machines
  - No modeling of underlying distributions
  - Make decisions using distance from boundary
- Example: Gene finding: HMMs vs. CRFs

Bayesian classification with a single feature



Ex 1: DNA repair genes show higher expression during stress

Ex 2: Protein-coding regions show higher conservation levels

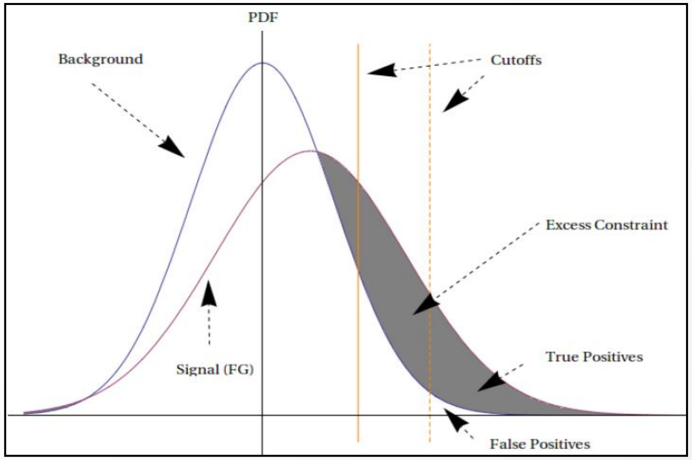
Ex 3: Regulatory regions show higher GC-content

In general: foreground signal vs. background

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

$$\underbrace{P(Class \mid Feature)}_{\textbf{Posterior}} = \frac{\overbrace{P(Feature \mid Class)}^{\textbf{Likelihood}}\overbrace{P(Class)}^{\textbf{Prior}}}{\underbrace{P(Feature)}_{\textbf{Evidence}}}$$

Classification problem: Max Probability Class



Select the class that maximizes posterior:

$$\underbrace{P(Class \mid Feature)}_{\textbf{Posterior}} = \frac{\overbrace{P(Feature \mid Class)}^{\textbf{Likelihood}}\overbrace{P(Class)}^{\textbf{Prior}}}{\underbrace{P(Feature)}_{\textbf{Evidence}}}$$

Maximum-A-Posteriori (MAP) estimates

$$BestClass = argmax_C\, P(Class|Feature)$$

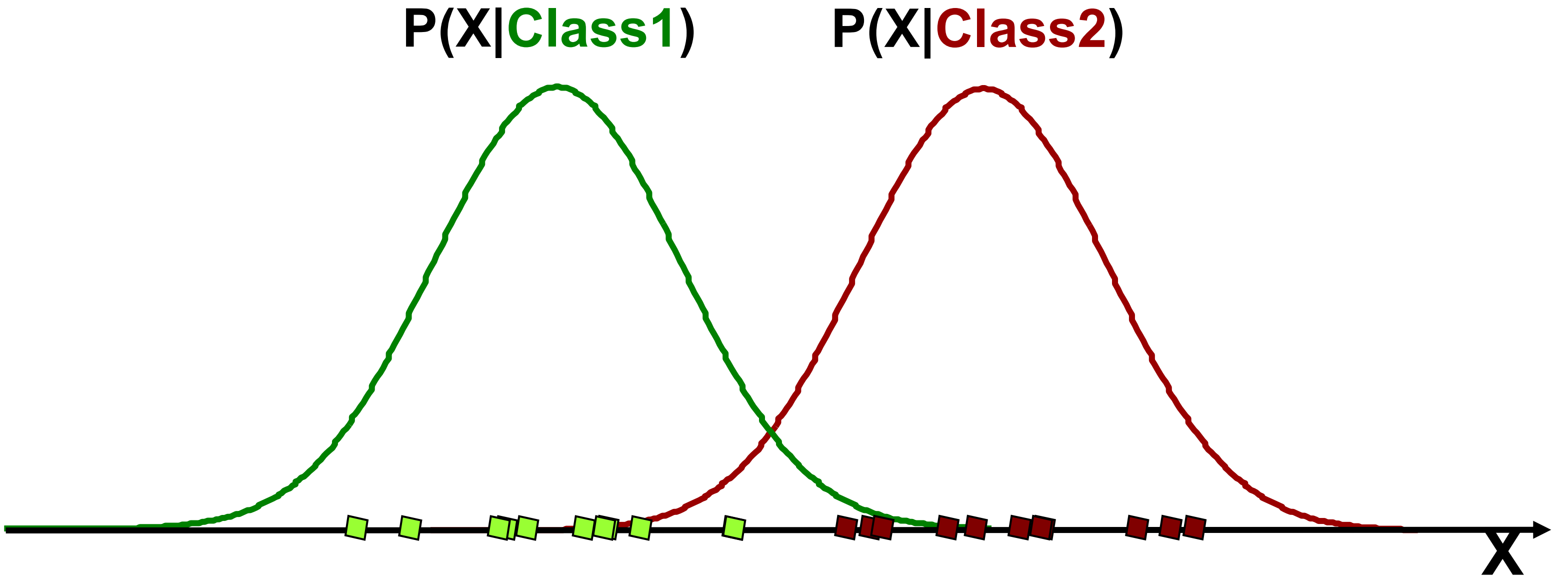
$$= argmax_C\, P(Feature|Class)\, P(Class)$$

Scaling the above distribution based on class priors

Likelihood:

$$P(Class \mid Feature) = \frac{\boxed{P(Feature \mid Class)}P(Class)}{P(Feature)}$$

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:

$$P(Class \mid Feature) = \frac{P(Feature \mid Class)\boxed{P(Class)}}{P(Feature)}$$

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

P(Class2) & P(Class1)

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

Evidence

$$P(Class \mid Feature) = \frac{P(Feature \mid Class)P(Class)}{\boxed{P(Feature)}}$$

Total evidence is P(Feature)=Σ_i P(Feature|Class$_i$)P(Class$_i$)

But it does not need to be known for classification

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

The Bayes Decision Rule is simply choose Class1 if:

$$P(Class1 \mid X) > P(Class2 \mid X)$$

$$\frac{P(X \mid Class1)P(L1)}{P(X)} > \frac{P(X \mid Class2)P(L2)}{P(X)}$$

same

$$P(X \mid Class1)P(Class1) > P(X \mid Class2)P(Class2)$$

➔ P(Feature) does not need to be computed for classification

Discriminant Function for selecting Class1

We can create a convenient representation of the Bayes Decision Rule

$$P(X \mid Class1)P(Class1) > P(X \mid Class2)P(Class2)$$

$$\frac{P(X \mid Class1)P(Class1)}{P(X \mid Class2)P(Class2)} > 1$$

$$\boxed{G(X) = \log \frac{P(X \mid Class1)}{P(X \mid Class2)} \frac{P(Class1)}{P(Class2)} > 0}$$

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)**
   - Algorithmic formulation: Update rule, optimality criterion. Fuzzy k-means.
   - 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

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

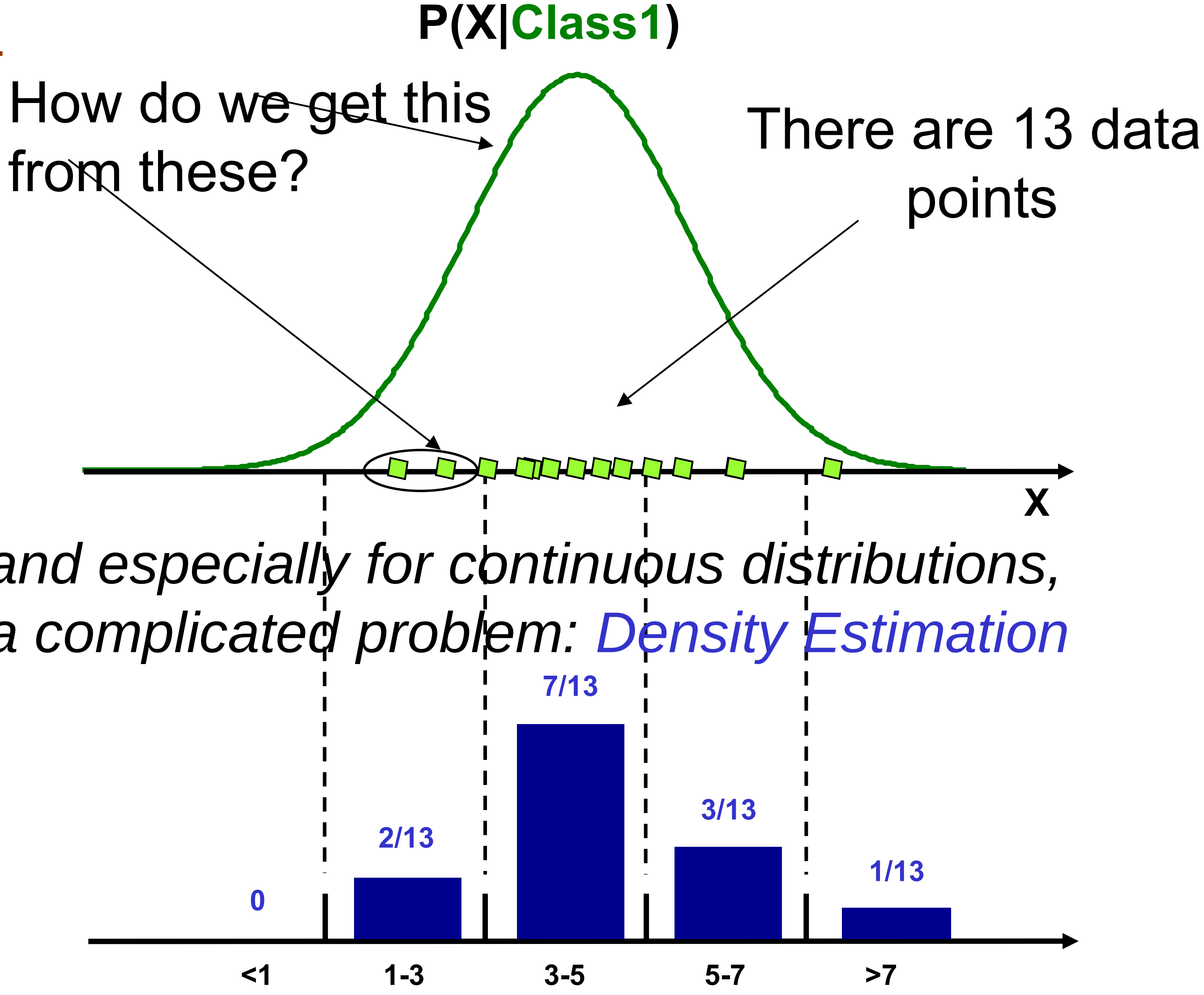
We will classify each object in the test set and count the number of each type of error

Getting P(X|Class) from Training Set

One Simple Approach

Divide X values into bins

And then we simply count frequencies



How do we get this from these?

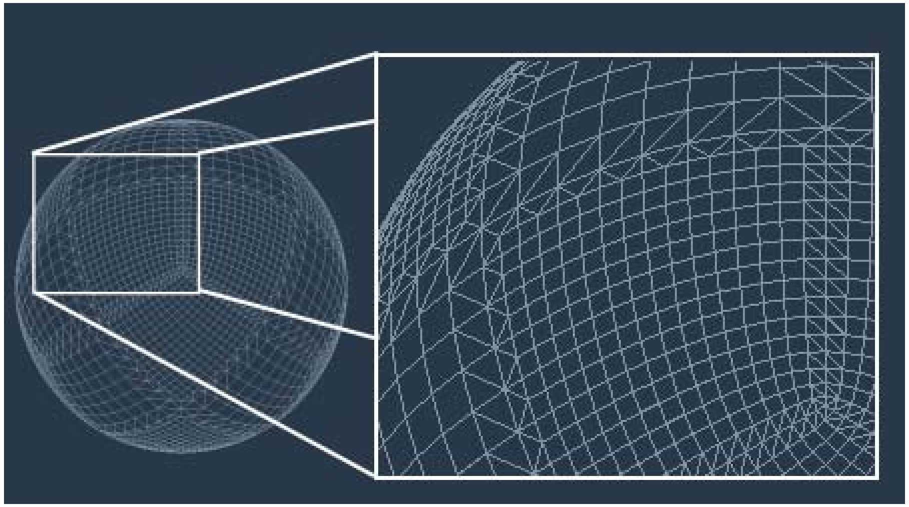
There are 13 data points

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

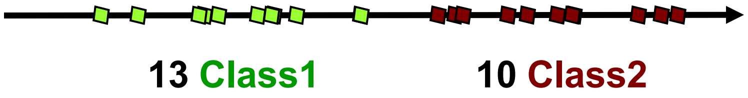
Getting Priors

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

2. Estimate from "expert" knowledge

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

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)**
   - Algorithmic formulation: Update rule, optimality criterion. Fuzzy k-means.
   - 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

Combining Multiple Features

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

Targeting signal

Protein domains

Co-expression

Mass Spec

Homology

Induction

Motifs

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

$$G(X) = \log \frac{P(X_1, X_2, ..., X_7 \mid Class1)}{P(X_1, X_2, ..., X_7 \mid Class2)} \frac{P(Class1)}{P(Class2)} > 0$$

Naïve Bayes Classifier

We are going to make the following assumption:

All features are independent given the class

$$P(X_1, X_2, \ldots, X_n \mid Class) = P(X_1 \mid Class) P(X_2 \mid Class) \ldots P(X_n \mid Class)$$

$$= \prod_{i=1}^{n} P(X_i \mid Class)$$

We can thus estimate <u>individual distributions</u> for each feature and just <u>multiply</u> them together!

Naïve Bayes Discriminant Function

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

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

As this:

$$G(X_1,\ldots,X_7) = \log \frac{\prod P(X_i \mid Class1)}{\prod P(X_i \mid Class2)} \frac{P(Class1)}{P(Class2)} > 0$$

Which can be simply computed as the sum of log scores

Binary Classification Errors

	True (Mito)	False (~Mito)
Predicted True	TP	FP
Predicted False	FN	TN

Sensitivity = TP/(TP+FN) Specificity = TN/(TN+FP)

- Sensitivity
  - Fraction of all Class1 (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

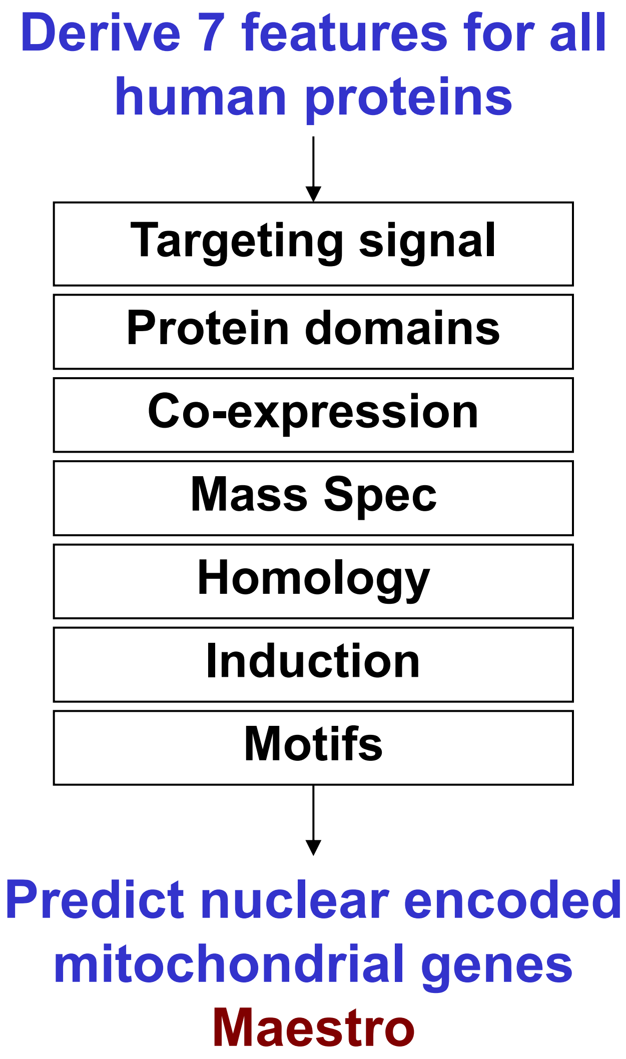
2. **K-means clustering (clustering by partitioning)**
   - Algorithmic formulation: Update rule, optimality criterion. Fuzzy k-means.
   - Machine learning formulation: Generative models, Expectation Maximization.

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

Classifying Mitochondrial Proteins



Derive 7 features for all human proteins

Targeting signal

Protein domains

Co-expression

Mass Spec

Homology

Induction

Motifs

Predict nuclear encoded mitochondrial genes

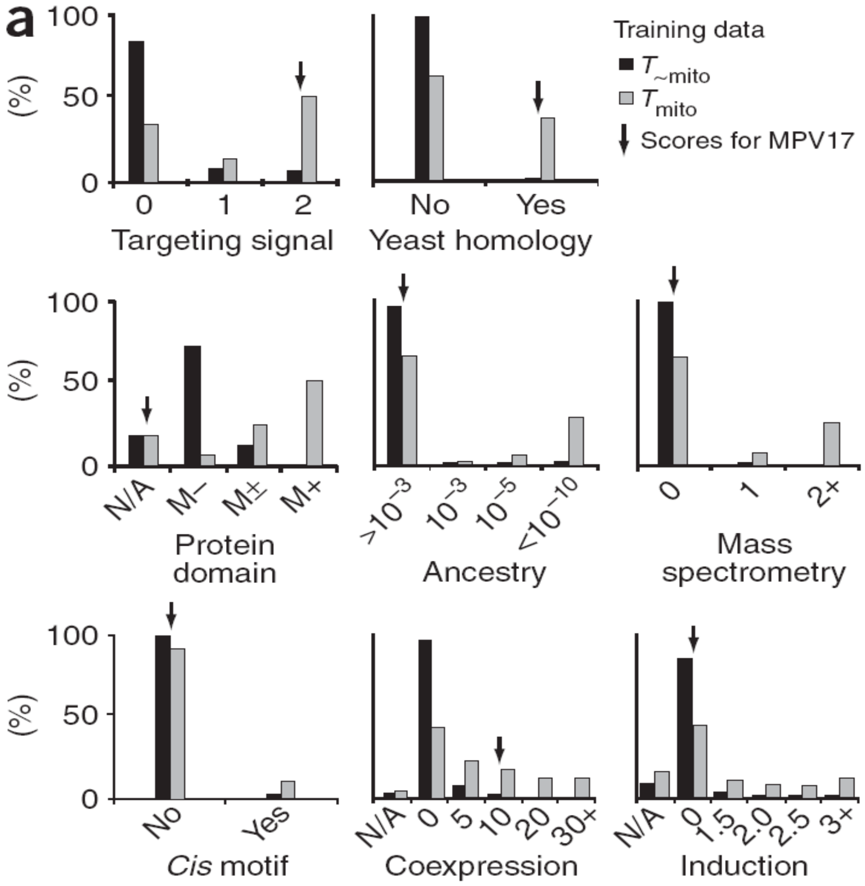
Maestro

First page of article removed due to copyright restrictions.
Source: Calvo, Sarah et al. "Systematic identification of human mitochondrial disease genes through integrative genomics." Nature Genetics 38, no. 5 (2006): 576-582.

Individual Feature Distributions

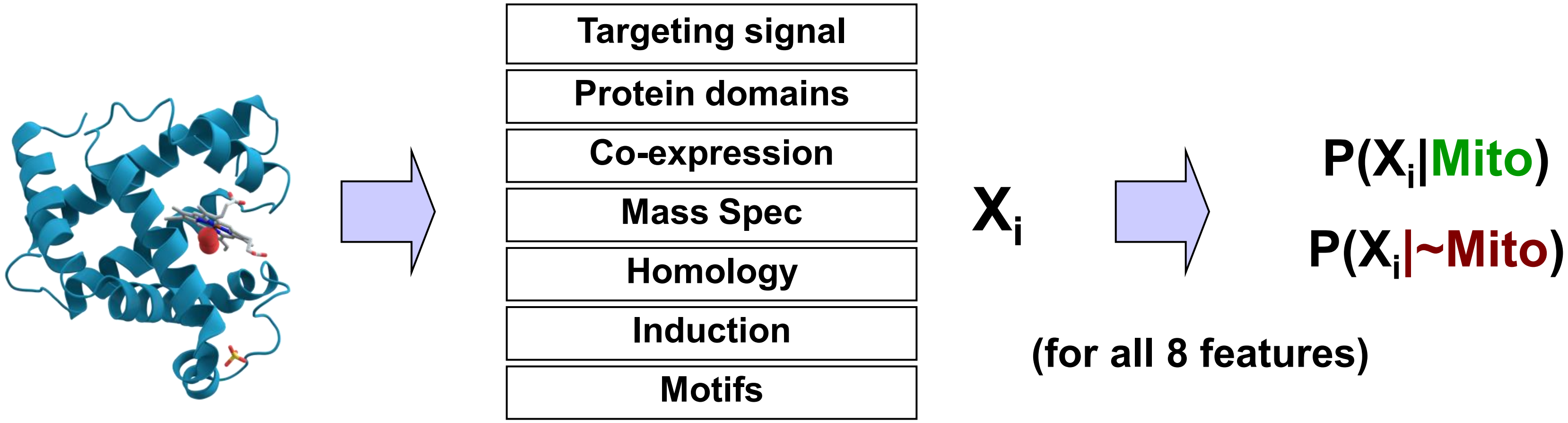
Instead of a single big distribution, we have a smaller one for each feature (and class)

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)**



Courtesy of Nature Publishing Group. Used with permission.
Source: Calvo, Sarah et al. "Systematic identification of human mitochondrial disease genes through integrative genomics." Nature Genetics 38, no. 5 (2006): 576-582.

Classifying A New Protein



Targeting signal

Protein domains

Co-expression

Mass Spec

Homology

Induction

Motifs

X_i

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

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

(for all 8 features)

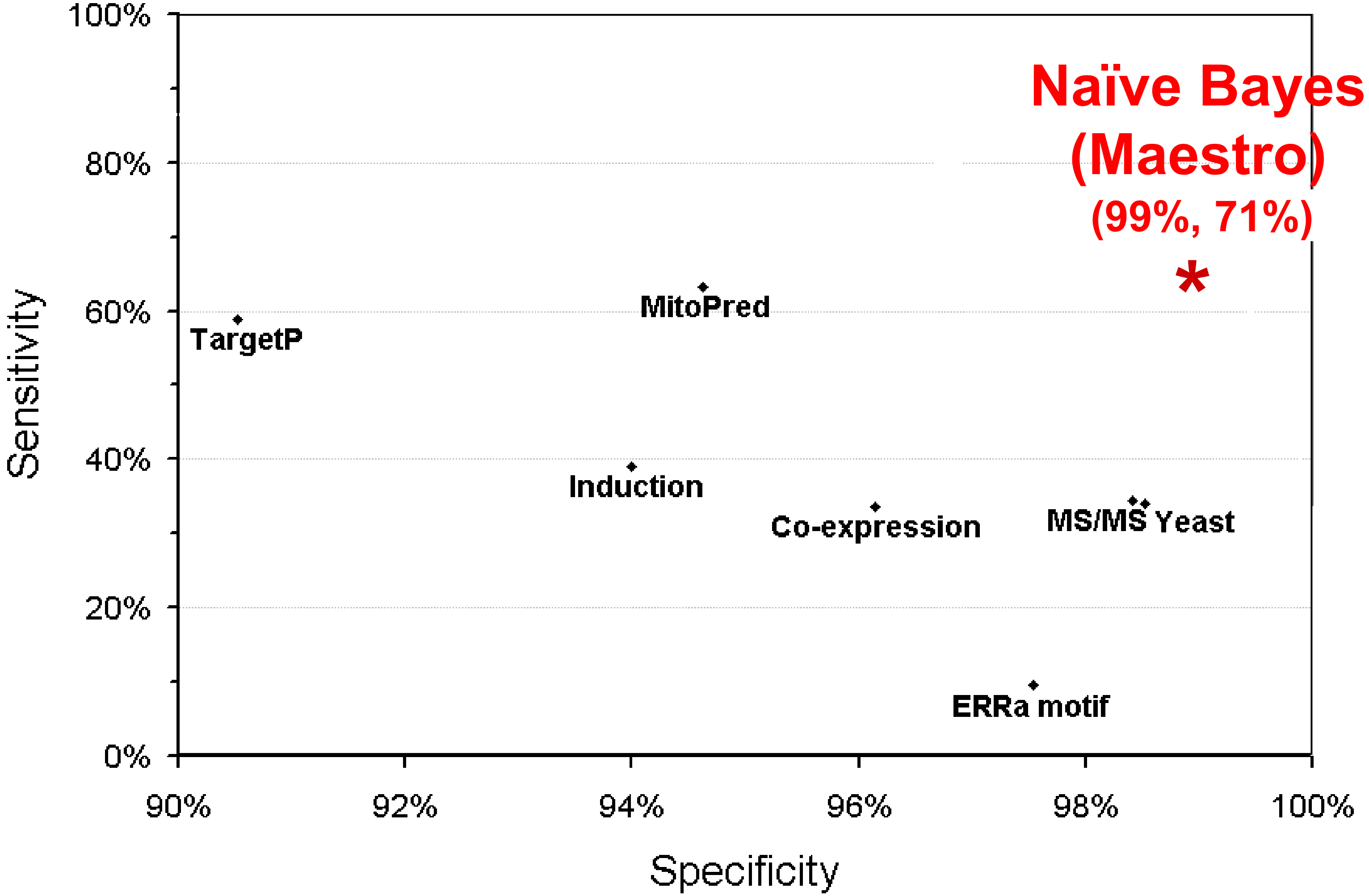
Courtesy of AzaToth; image in the public domain.

Plug these and priors into the discriminant function

$$G(X_1,\ldots,X_7)=\log\frac{\prod P(X_i\,|\,Mito)}{\prod P(X_i\,|\sim Mito)}\,\frac{P(Mito)}{P(\sim Mito)}>0$$

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)



Courtesy of Sarah Calvo. Used with permission.

Problem in genomics: not everything novel is false

Slide Credit: S. Calvo

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)**
   - Algorithmic formulation: Update rule, optimality criterion. Fuzzy k-means.
   - 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

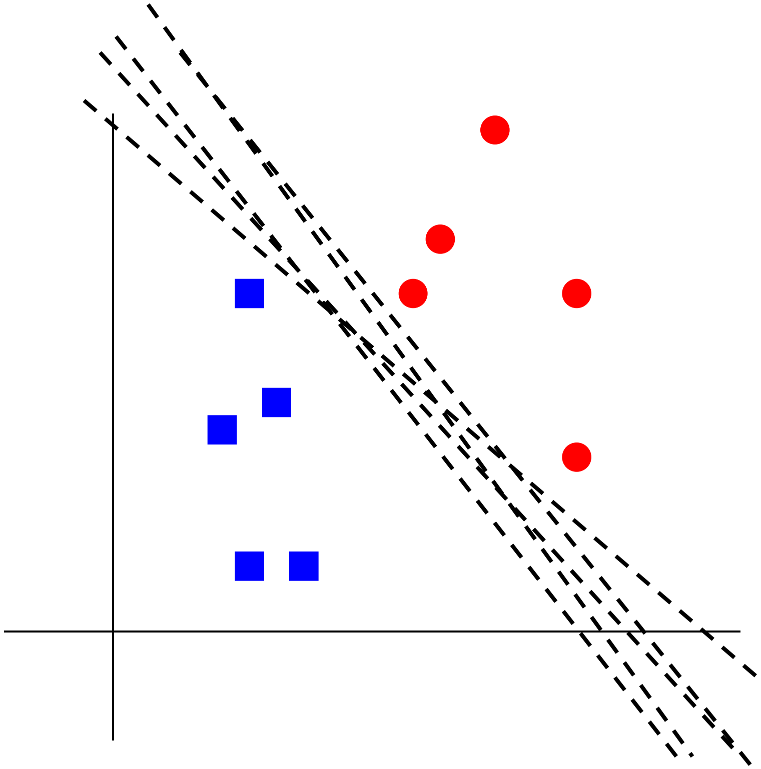
5. **(optional) Support Vector Machines (discriminative approach)**
   - SVM formulation, Margin maximization, Finding the support vectors
   - Non-linear discrimination, Kernel functions, SVMs in practice

Support Vector Machines (SVMs)

Easy to select a line

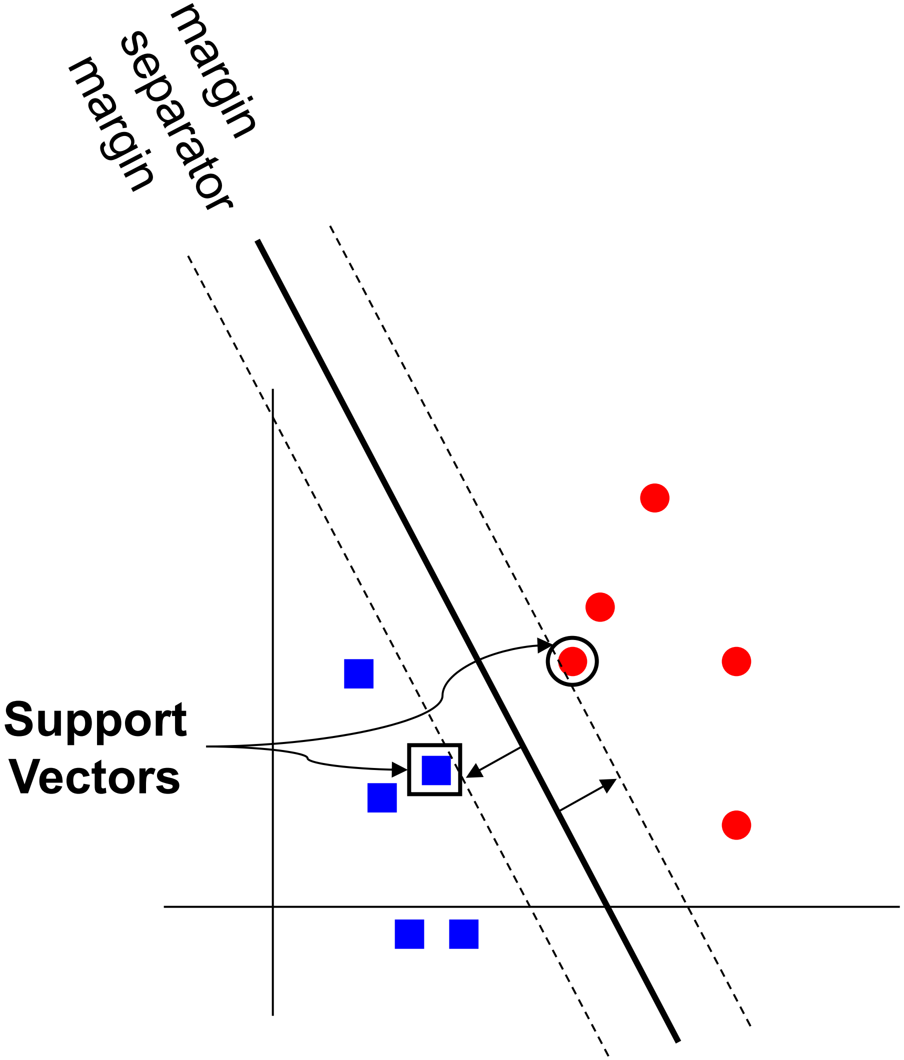
But many lines will separate these training data

What line should we choose?



Support Vector Machines (SVMs)

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



SVM Formulation

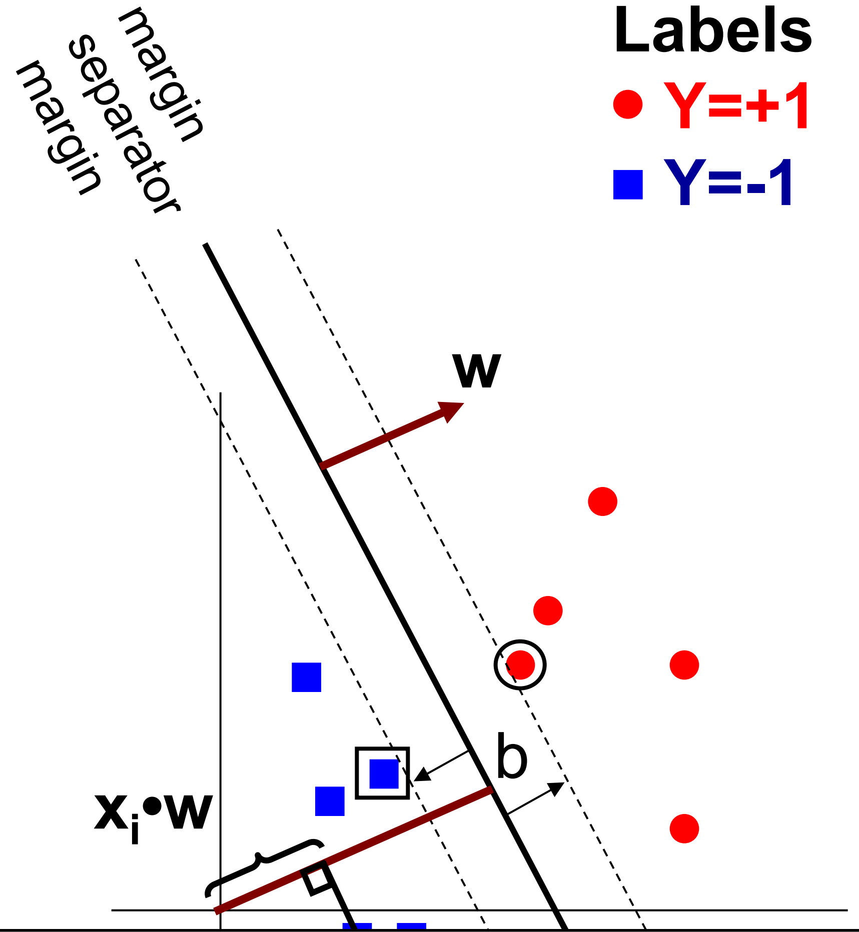
We define a vector **w** normal to the separating line

Assume all data satisfy the following:

$$\mathbf{x_i} \bullet \mathbf{w} - \mathrm{b} \geq +1 \quad \text{for } \mathrm{y_i} = +1$$

$$\mathbf{x_i} \bullet \mathbf{w} - \mathrm{b} \leq -1 \quad \text{for } \mathrm{y_i} = \text{-}1$$

⇩



We want to find the separator with the largest margin

An Optimization Probl

Only need dot product of input data!

For full derivation, see Burge

$$\text{Minimize } \mathrm{L_D} = \sum_i \alpha_i - \frac{1}{2}\sum_{i,j}\alpha_i\alpha_j y_i y_j \boxed{\mathbf{x_i} \bullet \mathbf{x_j}}$$

$$\text{subject to } \sum_i \alpha_i y_i = 0 \text{ and } \alpha_j > 0$$

Quadratic Programming

Solving for

$$\alpha_i\left(y_i\left(\mathbf{x_i} \bullet \mathbf{w} - b\right) - 1\right) = 0$$

$$\mathbf{w} = \sum_i \alpha_i y_i \mathbf{x_i}$$

Only some α_i are non-zero

$\mathbf{x}_i$ with a_i >0 are the *support vectors*

w is *determined by these data points!*

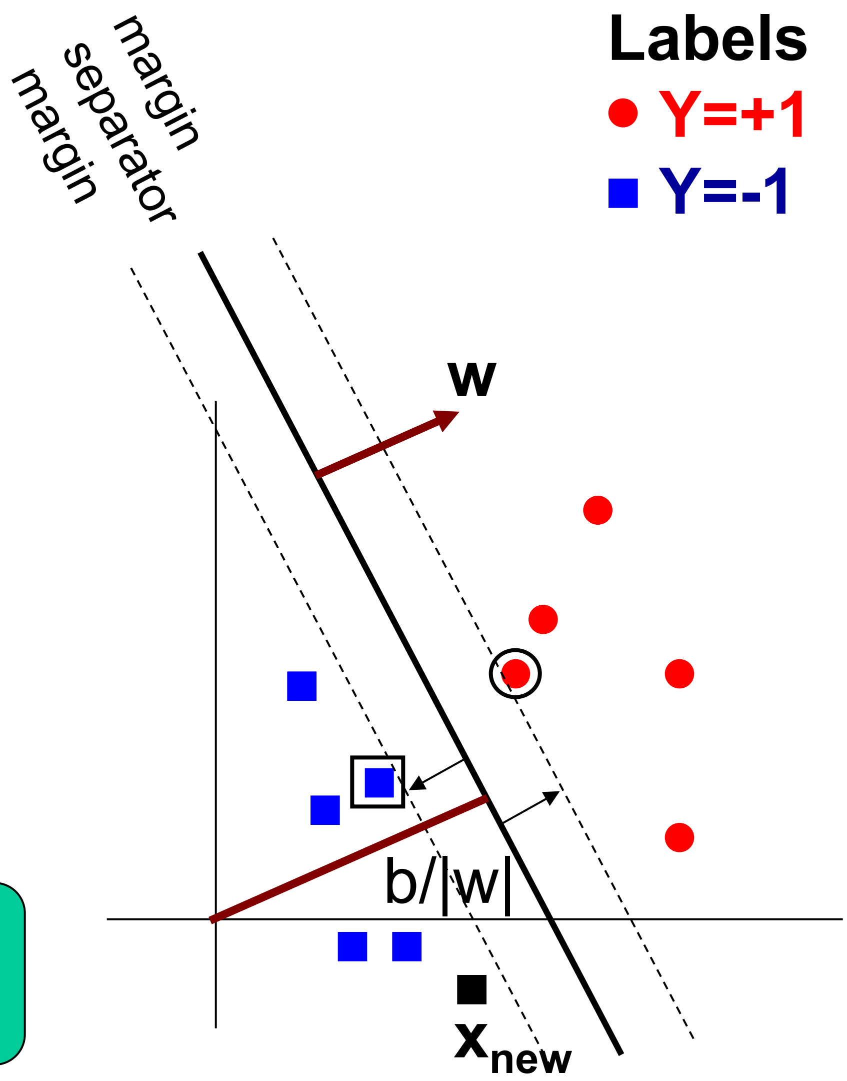
Using an SVM

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

$$y_i = \operatorname{sign}\left(\mathbf{w} \bullet \mathbf{x}_{\text{new}} - b\right)$$

$$= \operatorname{sign}\left(\sum_{\text{i}} \alpha_i y_i \boxed{\mathbf{x}_{\mathbf{i}} \bullet \mathbf{x}_{\text{new}}} - b\right)$$

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)**
   - Algorithmic formulation: Update rule, optimality criterion. Fuzzy k-means.
   - Machine learning formulation: Generative models, Expectation Maximization.

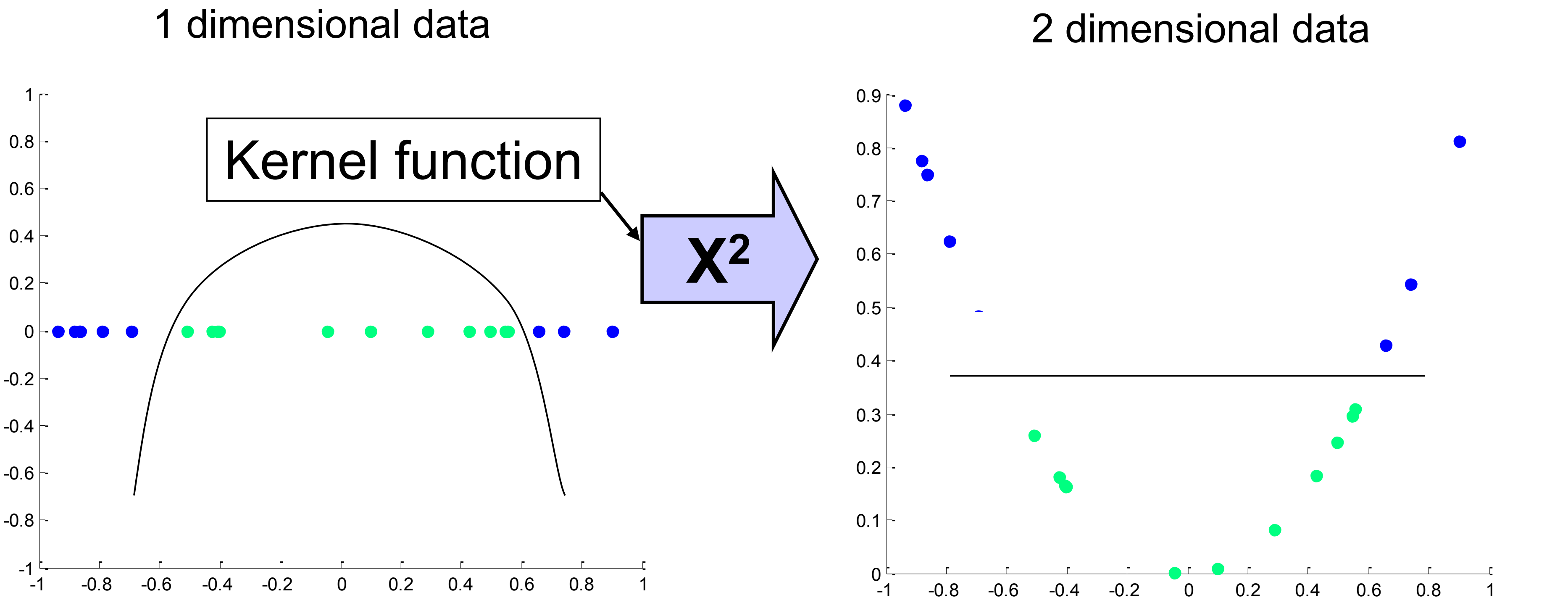
3. **Hierarchical Clustering (clustering by agglomeration)**
   - Basic algorithm, Distance measures. Evaluating clustering results

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

Non-linear Classifier

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



Noble, 2006. NATURE BIOTECHNOLOGY 24:1565.

Kernel Mapping

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

$\Phi(x)$: R^d -> H

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

$$\mathbf{X=(x_1,x_2)} \quad \Longrightarrow \quad \mathbf{\Phi(X)=(\phi_1,\phi_2,\phi_3,\ldots.\phi_\infty)}$$

Rather difficult to work with!

Kernel Mapping

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

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

$$\Phi(x): R^d \to H$$

$$\mathbf{X_i \cdot X_j} \xrightarrow{\textbf{becomes}} \Phi(\mathbf{X_i}) \cdot \mathbf{\Phi}(\mathbf{X_j})$$

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

$$\mathbf{K(X_i,X_j)} = \Phi(\mathbf{X_i}) \cdot \mathbf{\Phi}(\mathbf{X_j})$$

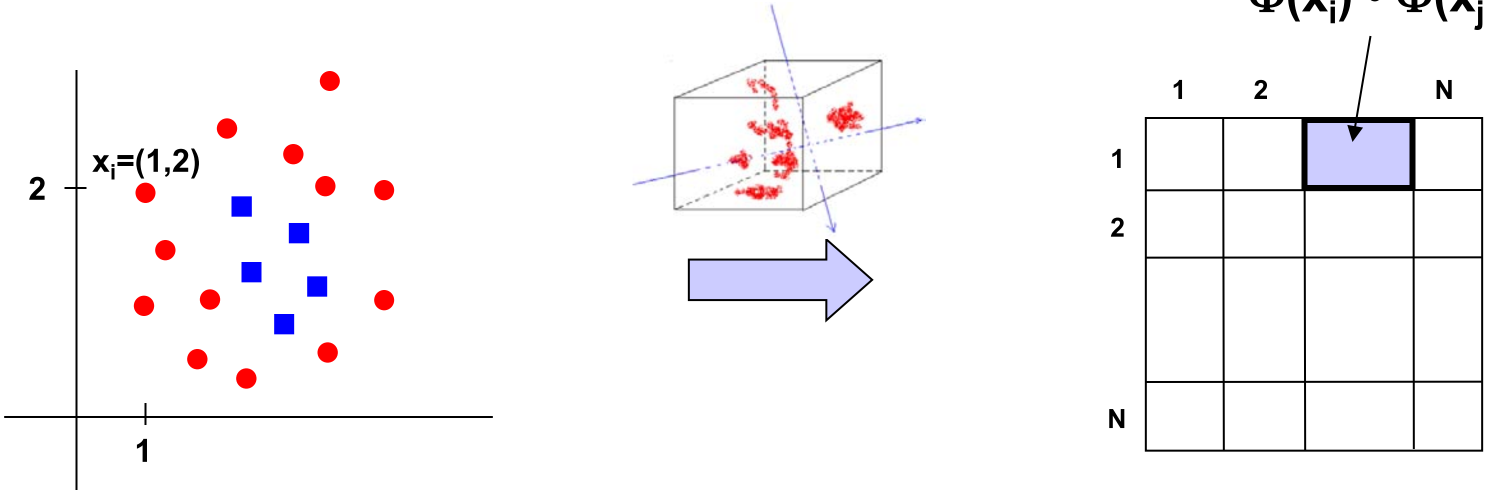
We can just use K and never know Φ(x) explicitly!

Φ(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

$\Phi(x_i) \bullet \Phi(x_j)$ = scalar!



$K(X_i,X_j)$

We have then done two very useful things:

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

Example Kernels

$$K\left(\mathbf{x}_i, \mathbf{x}_j\right) = \mathbf{x}_i^T \mathbf{x}_j$$

Linear

$$K\left(\mathbf{x}_i, \mathbf{x}_j\right) = \left(\gamma \mathbf{x}_i^T \mathbf{x}_j + r\right)^d$$

Polynomial

$$K\left(\mathbf{x}_i, \mathbf{x}_j\right) = \exp\left(-\gamma \left\|\mathbf{x}_i - \mathbf{x}_j\right\|^2\right)$$

Radial Basis Function

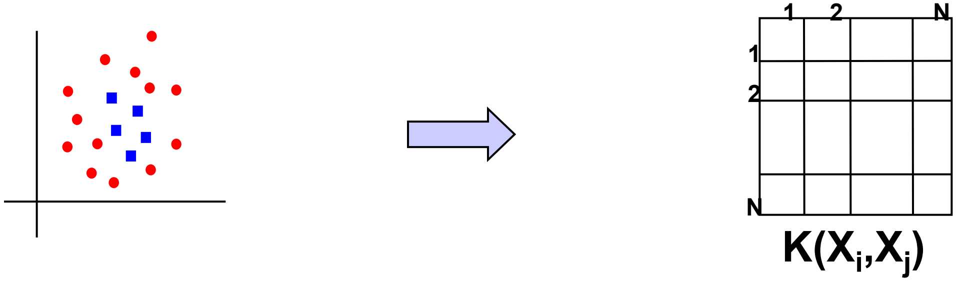
$$K\left(\mathbf{x}_i, \mathbf{x}_j\right) = \tanh\left(\gamma \mathbf{x}_i^T \mathbf{x}_j + r\right)$$

Sigmoid

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

Using (Non-Linear) SVMs

Step 1 – Transform data to Kernel Matrix K



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

$$\text{Minimize } \mathrm{L_D} = \sum_i \alpha_i - \frac{1}{2}\sum_{i,j}\alpha_i\alpha_j y_i y_j \mathbf{x_i}\bullet\mathbf{x_j} = \sum_i \alpha_i - \frac{1}{2}\sum_{i,j}\alpha_i\alpha_j y_i y_j \mathrm{K}\left(\mathbf{x_i},\mathbf{x_j}\right)$$

Step 2 – Test/Classify on new samples

$$y_{new} = \mathrm{sign}\left(\mathbf{w}\bullet\mathbf{x}_{\mathrm{new}}\right) = \mathrm{sign}\left(\sum_{\mathrm{i}}\alpha_i y_i \mathbf{x_i}\bullet\mathbf{x}_{\mathrm{new}}\right) = \mathrm{sign}\left(\sum_{\mathrm{i}}\alpha_i y_i \mathrm{K}\left(\mathbf{x_i},\mathbf{x}_{\mathrm{new}}\right)\right)$$

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)**
   - Algorithmic formulation: Update rule, optimality criterion. Fuzzy k-means.
   - 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

Classifying Tumors with Array Data

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

- Independent samples:
  - 34 leukemia samples
  - 24 bone marrow
  - 10 peripheral blood samples

- Assay ~6800 Genes

Excerpt of article removed due to copyright restrictions.
Source: Golub, Todd R. et al. "Molecular classification of cancer: Class discovery and class prediction by gene expression monitoring." Science 286, no. 5439 (1999): 531-537.

Weighted Voting Classfication

General approach of Golub et al (1999) paper:

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

Results

Initial Samples

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

Independent Samples

- 29 of 34 samples are strongly predicted with 100% accuracy.
- 5 not predicted

Training Set

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

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

SVM Approach

Text and table removed removed due to copyright restrictions.
Source: Mukherjee, Sayan et al. "Support vector machine classification of microarray data." CBCL Paper #182/AI Memo #1677(1999).

Methods

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

- Linear SVM

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

Results

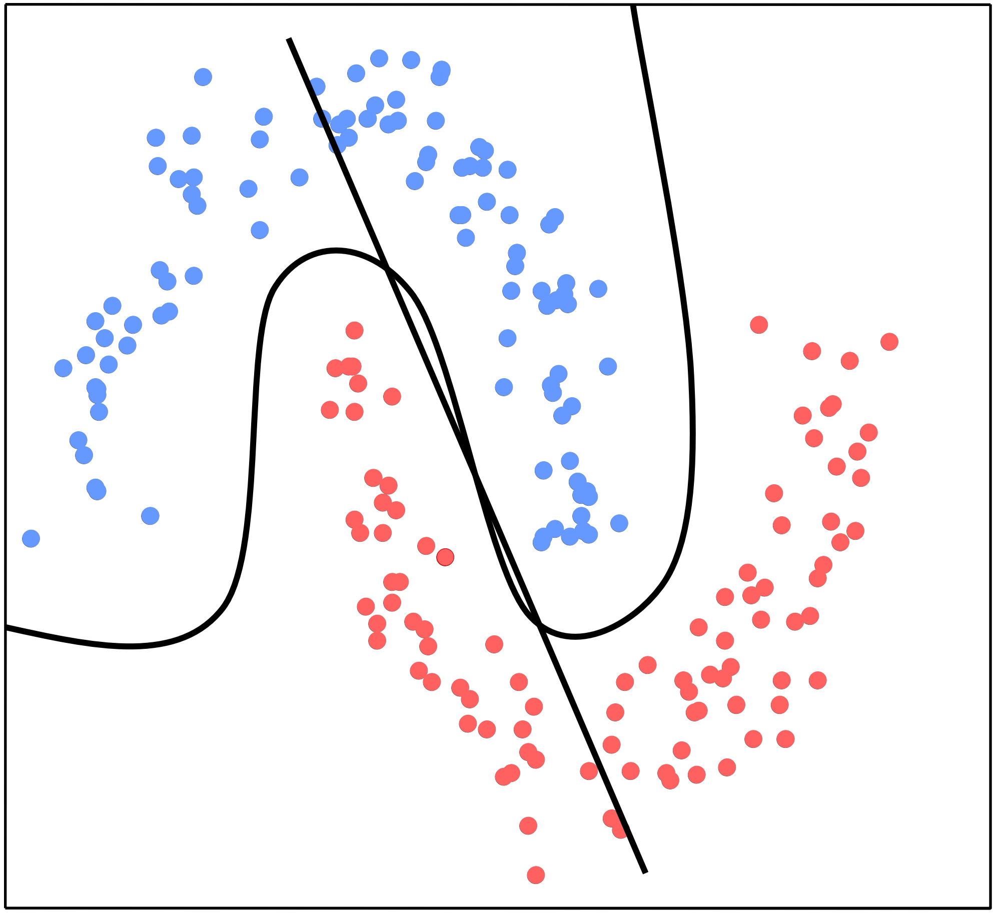
Text and table removed removed due to copyright restrictions.
Source: Mukherjee, Sayan et al. "Support vector machine classification of microarray data." CBCL Paper #182/AI Memo #1677(1999).

Results

Figure 9.6 removed due to copyright restrictions.
Source: Mukherjee, Sayan. "Classifying Microarray Data Using Support Vector Machines."

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)**
   - Algorithmic formulation: Update rule, optimality criterion. Fuzzy k-means.
   - 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