Scoring Alignments
Clustering

Scoring Alignments

A Genome Sequence

```
GCGTCTGACGGCGCACCGTTCGCGCTGCCGGCACCCCGGGCTCCATAATGAAAATCATGT
TCAGTAAGCTACACTCTGCATATCGGGCTACCAACGAAATGGAGTATCGGTCATGATCTT
GCAAGCCGTGCCTAAAAGCTTGGCCGCAGGGCCGAGTATAATTGGTCGCGGTCGCCTCGA
AGTTAGCTTATGCAATGCAGGAGGTGGGGCAAAGTTCAGGCGGATCGGCCGATGGCGGGC
GTAGGTGAAGGAGACAGCGGAGGCGTGGAGCGTGATGACATTGGCATGGTGGCCGCTTCC
CCCGTCGCGTCTCGGGTAAATGGCAAGGTAGACGCTGACGTCGTCGGTCGATTTGCCACC
TGCTGCCGTGCCCTGGGCATCGCGGTTTACCAGCGTAAACGTCCGCCGGACCTGGCTGCC
GCCCGGTCTGGTTTCGCCGCGCTGACCCGCGTCGCCCATGACCAGTGCGACGCCTGGACC
GGGCTGGCCGCTGCCGGCGACCAGTCCATCGGGGTGCTGGAAGCCGCCTCGCGCACGGCG
ACCACGGCTGGTGTGTTGCAGCGGCAGGTGCAACTGGCCGATAACGCCTTGGGCTTCCTG
TACGACACCGGGCTGTACCTGCGTTTTCGTGCCACCGGACCTGACGATTTCCACCTCGCG
TATGCCGCTGCGTTGGCTTCGACGGGCGGGCCGGAGGAGTTTGCCAAGGCCAATCACGTG
```

GGAGGTGGGGCAAAGTTCAGG

A fragment of a gene

A Genome Sequence

```
GCGTCTGACGGCGCACCGTTCGCGCTGCCGGCACCCCGGGCTCCATAATGAAAATCATGT
TCAGTAAGCTACACTCTGCATATCGGGCTACCAACGAAATGGAGTATCGGTCATGATCTT
GCAAGCCGTGCCTAAAAGCTTGGCCGCAGGGCCGAGTATAATTGGTCGCGGTCGCCTCGA
AGTTAGCTTATGCAATGCAGGAGGTGGGGCAAAGTTCAGGCGGATCGGCCGATGGCGGGC
GTAGGTGAAGGAGACAGCGGAGGCGTGGAGCGTGATGACATTGGCATGGTGGCCGCTTCC
CCCGTCGCGTCTCGGGTAAATGGCAAGGTAGACGCTGACGTCGTCGGTCGATTTGCCACC
TGCTGCCGTGCCCTGGGCATCGCGGTTTACCAGCGTAAACGTCCGCCGGACCTGGCTGCC
GCCCGGTCTGGTTTCGCCGCGCTGACCCGCGTCGCCCATGACCAGTGCGACGCCTGGACC
GGGCTGGCCGCTGCCGGCGACCAGTCCATCGGGGTGCTGGAAGCCGCCTCGCGCACGGCG
ACCACGGCTGGTGTGTTGCAGCGGCAGGTGCAACTGGCCGATAACGCCTTGGGCTTCCTG
TACGACACCGGGCTGTACCTGCGTTTTCGTGCCACCGGACCTGACGATTTCCACCTCGCG
TATGCCGCTGCGTTGGCTTCGACGGGCGGGCCGGAGGAGTTTGCCAAGGCCAATCACGTG
```

GGAGGTGGGGCAAAGTTCAGG

A fragment of a gene

Probabilistic Model of Alignments

Focus on *protein alignments without gaps*

Given an alignment, two possibilities

R: sequences related by evolution

U: sequences unrelated

Model for Unrelated Sequences

Assume each position sampled independently from the same random distribution

Let q_a be the <u>probability of amino acid *a*</u>

Then the probability of an *n* character alignment of X and Y under U is:

$$P(x, y \mid U) = \prod_{i=1}^{n} q_{x_i} \prod_{y=1}^{n} q_{y_i} = \prod_{i=1}^{n} q_{x_i} q_{y_i}$$

Model for Related Sequences

Each pair of aligned amino acids *evolved from a common ancestor*

Let p_{ab} be <u>the probability that amino a and b evolved from the same ancestral amino acid</u>

Then the probability of an *n* character alignment of X and Y under R is:

$$P(x, y \mid R) = \prod_{i=1}^{n} p_{x_i, y_j}$$

Probabilistic Model of Alignments

How do we decide between U and R?

(Hint: think of U and R as parameters of a model…)

Maximum Likelihood (Likelihood Ratio)

$$\frac{P(x, y \mid R)}{P(x, y \mid U)} = \frac{\prod_{i=1}^{n} p_{x_i, y_j}}{\prod_{i=1}^{n} q_{x_i} q_{y_i}}$$

$$l = \log \frac{P(x, y \mid R)}{P(x, y \mid U)}$$

Probabilistic Model of Alignments

How do we decide between U and R?

(Hint: think of U and R as parameters of a model…)

Maximum Likelihood (Likelihood Ratio)

$$\frac{P(x,y\,|\,R)}{P(x,y\,|\,U)} = \frac{\prod_{i=1}^{n} p_{x_i,y_j}}{\prod_{i=1}^{n} q_{x_i} q_{y_i}}$$

$$l = \log\frac{P(x,y\,|\,R)}{P(x,y\,|\,U)} = \log\frac{\prod_{i=1}^{n} p_{x_i,y_j}}{\prod_{i=1}^{n} q_{x_i} q_{y_i}} = \sum_i \log\frac{p_{x_i,y_j}}{q_{x_i} q_{y_i}}$$

The Substitution Matrix

In general, the score of a match or mismatch is given by s(x,y) – called a substitution matrix

And the score of an ungapped sequence is:

$$S = \sum_i s(x_i, y_j)$$

The Substitution Matrix

In general, the score of a match or mismatch is given by s(x,y) – called a substitution matrix

And the score of an ungapped sequence is:

$$S = \sum_i s(x_i, y_j) \equiv \quad l = \log \frac{P(x, y \mid R)}{P(x, y \mid U)}$$

**** We can always choose to interpret s as l***

$$\boxed{s(x_i, y_j) = \log\left(\frac{p_{ab}}{q_a q_b}\right)}$$

Probabilistic Model of Alignments

$$s(x_i, y_j) = \log\left(\frac{p_{ab}}{q_a q_b}\right)$$

Any score for a match/mismatch a,b can be interpreted the log of the ratio of

- The probability of *a,b* evolving from an ancestor
- Probability of a and b each occuring independently

We now have a principled way of thinking about substitution matrices

Substitution Matrices

- two popular sets of matrices for protein sequences
  - PAM matrices [Dayhoff *et al.,* 1978]
  - BLOSUM matrices [Henikoff & Henikoff, 1992]
- both try to capture the the relative substitutability of amino acid pairs in the context of evolution

BLOSUM62

Comm ... ow weights

Rar ... igh weights

Negative for less likely substitutions

Positive for more likely substitutions

	A	R	N	D	C	Q	E	G	H	I	L	K	M	F	P	S	T	W	Y	V	X
A	4																				
R	-1	5																			
N	-2	0	6																		
D	-2	-2	1	6																	
C	0	-3	-3	-3	9																
Q	-1	1	0	0	-3	5															
E	-1	0	0	2	-4	2	5														
G	0	-2	0	-1	-3	-2	-2	6													
H	-2	0	1	-1	-3	0	0	-2	8												
I	-3	-3	-3	-3	-1	-3	-3	-4	-3	4											
L	-1	-2	-3	-4	-1	-2	-3	-4	-3	2	4										
K	-1	2	0	-1	-3	1	1	-2	-1	-3	-2	5									
M	-1	-1	-2	-3	-1	0	-2	-3	-2	1	2	-1	5								
F	-2	-3	-3	-3	-2	-3	-3	-3	-1	0	0	-3	0	6							
P	-1	-2	-2	-1	-3	-1	-1	-2	-2	-3	-3	-1	-2	-4	7						
S	1	-1	[illegible]	0	-1	0	0	0	-1	-2	-2	0	-1	-2	-1	4					
T	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	-1	-1	-2	-1	1	5				
W	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	-3	-1	1	-4	-3	-2	11			
Y	-2	-2	-2	-3	-2	-1	-2	-3	2	-1	-1	-2	-1	3	-3	-2	-2	2	7		
V	0	-3	-3	-3	-1	-2	-2	-3	-3	3	1	[illegible]	1	-1	-2	-2	0	-3	-1	4	
X	0	-1	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	-2	0	0	-2	-1	-1	-1

CLUSTERING

Structure in High-Dimensional Data

Structure in High-Dimensional Data

Topologically Clean Distance Fields. A. Gyulassy, V. Natarajan, Mark Duchaineau , Valerio Pascucci, Eduardo M. Bringa, Andrew Higginbotham, and Bernd Hamann, IEEE Transactions on Visualization and Computer Graphics.

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

Clustering vs Classification

- Objects characterized by one or more features

- Classification
  - Have labels for some points
  - Want a "rule" that will accurately assign labels to new points
  - Supervised learning

- Clustering
  - No labels
  - Group points into clusters based on how "near" they are to one another
  - Identify structure in data
  - Unsupervised learning

Today

- K-means clustering
- Expectation Maximization

Expression Microarrays

- A way to measure the levels of mRNA in every gene
- Two basic types
  - Affymetrix gene chips
  - Spotted oligonucleotides
- Both work on same principle
  - Put DNA probe on slide
  - Complementary hybridization

Expression Microarray Data Matrix

- Genes are typically given as rows
- Experiment are given by columns

Clustering Expression Data

- Cluster Experiments
  - Group by similar expression profiles

- Cluster Genes
  - Group by similar expression in different conditions

ALL
MLL
-3σ
-2σ
-1σ
0
+1σ
+2σ
+3σ

A
G
DLBCL
Germinal centre B
NL lymph node/tonsil
Activated blood B
Resting/activated T
Transformed cell lines
FL
Resting blood B
CLL
Pan B cell
Germinal Centre B cell
T cell
Activated B cell
Proliferation
Lymph node
-2
-1
0
1
2
0.250
0.500
1.000
2.000
4.000

Why Cluster Genes by Expression?

- Data Exploration
  - Summarize data
  - Explore without getting lost in each data point
  - Enhance visualization
- Co-regulated Genes
  - Common expression may imply common regulation
  - Predict *cis*-regulatory promoter sequences
- Functional Annotation
  - Similar function from similar expression

Clustering and Classification in Genomics

- Classification
  - Microarray data: classify cell state (i.e. AML vs ALL) using expression data
  - Protein/gene sequences: predict function, localization, etc.

- Clustering
  - Microarray data: groups of genes that share similar function have similar expression patterns – identify regulons
  - Protein sequence: group related proteins to infer function
  - EST data: collapse redundant sequences

Clustering Algorithms

- Partitioning
  - Divides objects into non-overlapping clusters such that each data object is in exactly one subset
- Agglomerative
  - A set of nested clusters organized as a hierarchy

K-Means Clustering

The Basic Idea

- Assume a fixed number of clusters, K
- Goal: create “compact” clusters

K-Means Algorithm

- Randomly
  Initialize Clusters

Slide credit: Serafim Batzoglou

K-Means Algorithm

- Randomly Initialize Clusters
- Assign data points to nearest clusters

Slide credit: Serafim Batzoglou

K-Means Algorithm

- Randomly Initialize Clusters
- Assign data points to nearest clusters
- Recalculate Clusters

Slide credit: Serafim Batzoglou

K-Means Algorithm

- Randomly Initialize Clusters
- Assign data points to nearest clusters
- Recalculate Clusters

Slide credit: Serafim Batzoglou

K-Means Algorithm

- Randomly Initialize Clusters
- Assign data points to nearest clusters
- Recalculate Clusters
- Repeat…

Slide credit: Serafim Batzoglou

K-Means Algorithm

- Randomly Initialize Clusters
- Assign data points to nearest clusters
- Recalculate Clusters
- Repeat…

Slide credit: Serafim Batzoglou

K-Means Algorithm

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

Slide credit: Serafim Batzoglou

More Formally

1. Initialize K centers $\mathbf{u}_k$

For each iteration n until convergence

2. Assign each $\mathbf{x}_i$ the label of the nearest center, where the distance between $\mathbf{x}_i$ and $\mathbf{u}_k$ is

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

3. Move the position of each $\mathbf{u}_k$ to the centroid of the points with that label

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

Cost Criterion

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

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

Minimizing this means minimizing cost of each cluster separately:

$$C_k\left(u_k\right)=\sum_{x_i \text{ with label k}}\left(\mathrm{x_i}\text{-}\mathrm{u_k}\right)^2 \quad\Rightarrow\quad \frac{dC_k\left(u_k\right)}{du_k}=2\sum_{X_i \text{ with label k}}\left(\mathrm{x_i}\text{-}\mathrm{u_k}\right)$$

$$\frac{dC_k\left(u_k\right)}{du_k}=0 \quad\Rightarrow\quad \mathrm{u_k}=\frac{1}{\left|x_i\right|}\sum_{X_i \text{ with label k}} x_i$$

the centroid

What About Ties?

Fuzzy K-Means

- Initialize K centers $\mathbf{u}_k$
- For each point calculate the probability of membership for each category

$$P(\text{label k} \mid x_i, u_k)$$

- Move the position of each $\mathbf{u}_k$ to the weighted centroid :

$$u_k(n+1) = \sum_{\text{All } x_i} x_i P(\text{label k} \mid x_i, u_k) \Big/ \sum_{\text{All } x_i} P(\text{label k} \mid x_i, u_k)$$

- Iterate

K-means

Fuzzy K-means

Fuzzy K-Means

- Initialize K centers $\mathbf{u}_k$
- For each point calculate the probability of membership for each category

$$P(\text{label k} \mid x_i, u_k)$$

K-means

- Move the position of each $\mathbf{u}_k$ to the weighted centroid :

u

Of course, K-Means just special case where

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

K-Means from a Generative Model

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

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

Unsupervised Learning

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

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

We need to learn *both* the means (μ_i) and labels from the data

If We Have Labeled Points

Need to estimate unknown gaussian centers from data

In general, how could we do this?
How could we "estimate" the "best" $u_{k?}$

Choose u_k to maximize probability of the data

If We Have Labeled Points

Need to estimate unknown gaussian centers from data

In general, how could we do this?
How could we "estimate" the "best" u_k?

Given a set of $\mathbf{x}_i$, all with label k, we can find the maximum likelihood μ_k from

$$\arg\max_{\mu} \left\{ \log \prod_{x_i \text{ with label k}} \mathrm{P}\left(\mathbf{x}_i \mid \mu_k\right) \right\} = \arg\max_{\mu} \sum_i \left\{ -\frac{1}{2}\left(\mathbf{x}_i - \mathbf{u}_k\right)^2 + \log\left(\frac{1}{\sqrt{2\pi}}\right) \right\}$$

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

Solution is the centroid of the x_i

If We Know Cluster Centers

Need to estimate labels for the data

Learn?

Distance measure for K-means

$$\arg\max_{k} \; P\left(\mathbf{x}_i \mid \mu_k\right) = \arg\max_{k} \; \frac{1}{\sqrt{2\pi}} \exp\left\{-\frac{\left(\mathbf{x}_i - \mathbf{u}_k\right)^2}{2}\right\} = \arg\min_{k} \left(\mathbf{x}_i - \mathbf{u}_k\right)^2$$

What If We Have Neither?

What If We Have Neither?

An idea:

What If We Have Neither?

An idea:

1. **Imagine we start with some u^0**

μ^0

What If We Have Neither?

An idea:

1. **Imagine we start with some u^0**
2. **We *could* calculate the most likely labels for x_i^0 given these u_k^0**

$$\mu^0$$

$$labels_i^0 = \arg\max_k \; \mathrm{P}_k\left(\mathbf{x}_i \mid \mu_k^0\right)$$

What If We Have Neither?

An idea:

1. **Imagine we start with some u^0**
2. **We *could* calculate the most likely labels for x_i^0 given these u_k^0**
3. **We *could* then use these labels to choose u_k^1 by <u>maximizing likelihood</u>**

$$\mu^0$$

$$labels_i^0 = \arg\max_{k} \; P_k\left(\mathbf{x}_i \mid \mu_k^0\right)$$

$$\mu_k^1 = \arg\max_{\mu} \left\{ \log \prod_{x_i \text{ with label k}} P_k\left(\mathbf{x}_i \mid \mu\right) \right\}$$

What If We Have Neither?

An idea:

1. **Imagine we start with some u^0**
2. **We *could* calculate the most likely labels for x_i^0 given these u_k^0**
3. **We *could* then use these labels to choose u_k^1 by <u>maximizing likelihood</u>**
4. **And iterate (to convergence)**

$$\mu^0$$

$$labels_i^0 = \arg\max_k \; P_k\left(\mathbf{x}_i \mid \mu_k^0\right)$$

$$\mu_k^1 = \arg\max_\mu \left\{ \log \prod_{x_i \text{ with label k}} P_k\left(\mathbf{x}_i \mid \mu\right) \right\}$$

$$labels_i^1 = \arg\max_k \; P_k\left(\mathbf{x}_i \mid \mu_k^1\right)$$

...

Lets Modify This Idea

An idea:

1. Imagine we start with some u_k^0
2. We *could* calculate the most likely labels for x_i^0 given these u_k^0
3. We *could* then use labels to choose u_k^1 by <u>maximizing likelihood</u>
4. And iterate (to convergence)

$$\mu^0$$

$$labels_i^0 = \arg\max_k \; P\left(\mathbf{x}_i \mid \mu_k^0\right)$$

$$\mu_k^1 = \arg\max_\mu \left\{ \log \prod_{\text{x with label k}} P\left(\mathbf{x}_i \mid \mu\right) \right\}$$

$$labels_i^1 = \arg\max_k \; P\left(\mathbf{x}_i \mid \mu_k^1\right)$$

...

The Modification:

1. Start with some u^0

$$\mu^0$$

Lets Modify This Idea

An idea:

1. Imagine we start with some u_k^0
2. We *could* calculate the most likely labels for x_i^0 given these u_k^0
3. We *could* then use labels to choose u_k^1 by <u>maximizing likelihood</u>
4. And iterate (to convergence)

$$\mu^0$$

$$labels_i^0 = \arg\max_k \; P\left(\mathbf{x}_i \mid \mu_k^0\right)$$

$$\mu_k^1 = \arg\max_\mu \left\{ \log \prod_{\text{x with label k}} P\left(\mathbf{x}_i \mid \mu\right) \right\}$$

$$labels_i^1 = \arg\max_k \; P\left(\mathbf{x}_i \mid \mu_k^1\right)$$

...

The Modification:

1. Start with some u^0
2. **Calculate the P(labels I x_i)** $P\left(label_i^0 = k \middle| \mathbf{x}_i, \mu^0\right)$

$$\mu^0$$

Lets Modify This Idea

An idea:

1. Imagine we start with some u_k^0
2. We *could* calculate the most likely labels for x_i^0 given these u_k^0
3. We *could* then use labels to choose u_k^1 by <u>maximizing likelihood</u>
4. And iterate (to convergence)

$$\mu^0$$

$$labels_i^0 = \arg\max_k \; \mathrm{P}\left(\mathbf{x}_i \mid \mu_k^0\right)$$

$$\mu_k^1 = \arg\max_\mu \left\{ \log \prod_{\text{x with label k}} \mathrm{P}\left(\mathbf{x}_i \mid \mu\right) \right\}$$

$$labels_i^1 = \arg\max_k \; \mathrm{P}\left(\mathbf{x}_i \mid \mu_k^1\right)$$

...

The Modification:

1. Start with some u^0
2. **Calculate the P(labels I x_i)**

$$P\left(label_i^0 = k \middle| \mathbf{x}_i, \mu^0\right) = \mathrm{P}\left(\mathbf{x}_i \mid \mu^0, label_i^0 = k\right) \mathrm{P}\left(label_i^0 = k\right) \Big/ \mathrm{P}\left(\mathbf{x}_i \middle| \mu^0\right)$$

μ^0 — **Prior** — **Likelihood** — **Evidence**

Bayes Rule

Lets Modify This Idea

An idea:

1. Imagine we start with some u_k^0
2. We *could* calculate the most likely labels for x_i^0 given these u_k^0
3. We *could* then use labels to choose u_k^1 by <u>maximizing likelihood</u>
4. And iterate (to convergence)

$$\mu^0$$

$$labels_i^0 = \arg\max_k \; P\left(\mathbf{x}_i \mid \mu_k^0\right)$$

$$\mu_k^1 = \arg\max_\mu \left\{ \log \prod_{\text{x with label k}} P\left(\mathbf{x}_i \mid \mu\right) \right\}$$

$$labels_i^1 = \arg\max_k \; P\left(\mathbf{x}_i \mid \mu_k^1\right)$$

...

The Modification:

1. Start with some u^0
2. **Calculate the P(labels I x_i)**

$$\mu^0$$

$$P\left(label_i^0 = k \middle| \mathbf{x}_i, \mu^0\right) = \mathrm{P}\left(\mathbf{x}_i \mid \mu^0, label_i^0 = k\right) \underbrace{\mathrm{P}\left(label_i^0 = k\right)}_{\text{Same for all k}} \Big/ \mathrm{P}\left(\mathbf{x}_i \middle| \mu^0\right)$$

How do we calculate this?

Lets Modify This Idea

An idea:

1. Imagine we start with some u_k^0
2. We *could* calculate the most likely labels for x_i^0 given these u_k^0
3. We *could* then use labels to choose u_k^1 by <u>maximizing likelihood</u>
4. And iterate (to convergence)

$$\mu^0$$

$$labels_i^0 = \arg\max_k \; P\left(\mathbf{x}_i \mid \mu_k^0\right)$$

$$\mu_k^1 = \arg\max_\mu \left\{ \log \prod_{\text{x with label k}} P\left(\mathbf{x}_i \mid \mu\right) \right\}$$

$$labels_i^1 = \arg\max_k \; P\left(\mathbf{x}_i \mid \mu_k^1\right)$$

...

The Modification:

1. Start with some u^0
2. **Calculate the P(labels I x_i)**
3. **Use this <u>probability</u> to choose u_k^1**

$$\mu^0$$

$$P\left(label_i^0 = k \middle| \mathbf{x}_i, \mu^0\right) = P\left(\mathbf{x}_i \mid \mu^0, label_i^0 = k\right) P\left(label_i^0 = k\right) \Big/ P\left(\mathbf{x}_i \middle| \mu^0\right)$$

$$P\left(\mathbf{x}_i \mid \mu\right)$$

Lets Modify This Idea

An idea:

1. Imagine we start with some u_k^0
2. We *could* calculate the most likely labels for x_i^0 given these u_k^0
3. We *could* then use labels to choose u_k^1 by <u>maximizing likelihood</u>
4. And iterate (to convergence)

$$\mu^0$$

$$labels_i^0 = \arg\max_k \; P\left(\mathbf{x}_i \mid \mu_k^0\right)$$

$$\mu_k^1 = \arg\max_\mu \left\{ \log \prod_{\text{x with label k}} P\left(\mathbf{x}_i \mid \mu\right) \right\}$$

$$labels_i^1 = \arg\max_k \; P\left(\mathbf{x}_i \mid \mu_k^1\right)$$

...

The Modification:

1. Start with some u^0
2. **Calculate the P(labels I x_i)**
3. **Use this <u>probability</u> to choose u_k^1**

$$\mu^0$$

$$P\left(label_i^0 = k \middle| \mathbf{x}_i, \mu^0\right) = P\left(\mathbf{x}_i \mid \mu^0, label_i^0 = k\right) P\left(label_i^0 = k\right) \Big/ P\left(\mathbf{x}_i \middle| \mu^0\right)$$

$$P\left(\mathbf{x}_i \mid \mu, label_i = k\right)$$

Lets Modify This Idea

An idea:

1. Imagine we start with some u_k^0
2. We *could* calculate the most likely labels for x_i^0 given these u_k^0
3. We *could* then use labels to choose u_k^1 by <u>maximizing likelihood</u>
4. And iterate (to convergence)

$$\mu^0$$

$$labels_i^0 = \arg\max_k \; P\left(\mathbf{x}_i \mid \mu_k^0\right)$$

$$\mu_k^1 = \arg\max_\mu \left\{ \log \prod_{\text{x with label k}} P\left(\mathbf{x}_i \mid \mu\right) \right\}$$

$$labels_i^1 = \arg\max_k \; P\left(\mathbf{x}_i \mid \mu_k^1\right)$$

...

The Modification:

1. Start with some u^0
2. **Calculate the P(labels I x_i)**
3. **Use this <u>probability</u> to choose u_k^1**

$$\mu^0$$

$$P\left(label_i^0 = k \middle| \mathbf{x}_i, \mu^0\right) = P\left(\mathbf{x}_i \mid \mu^0, label_i^0 = k\right) P\left(label_i^0 = k\right) \Big/ P\left(\mathbf{x}_i \middle| \mu^0\right)$$

$$P\left(\mathbf{x}_i \mid \mu, label_i = k\right) P\left(label_i = k \middle| \mathbf{x}_i, \mu^0\right)$$

Lets Modify This Idea

An idea:

1. Imagine we start with some u_k^0
2. We *could* calculate the most likely labels for x_i^0 given these u_k^0
3. We *could* then use labels to choose u_k^1 by <u>maximizing likelihood</u>
4. And iterate (to convergence)

$$\mu^0$$

$$labels_i^0 = \arg\max_k \; \mathrm{P}\left(\mathbf{x}_i \mid \mu_k^0\right)$$

$$\mu_k^1 = \arg\max_\mu \left\{ \log \prod_{\text{x with label k}} \mathrm{P}\left(\mathbf{x}_i \mid \mu\right) \right\}$$

$$labels_i^1 = \arg\max_k \; \mathrm{P}\left(\mathbf{x}_i \mid \mu_k^1\right)$$

...

The Modification:

1. Start with some u^0
2. **Calculate the P(labels I x_i)**
3. **Use this <u>probability</u> to choose u_k^1**

$$\mu^0$$

$$P\left(label_i^0 = k \middle| \mathbf{x}_i, \mu^0\right) = \mathrm{P}\left(\mathbf{x}_i \mid \mu^0, label_i^0 = k\right) \mathrm{P}\left(label_i^0 = k\right) \Big/ \mathrm{P}\left(\mathbf{x}_i \middle| \mu^0\right)$$

$$\sum_k \underbrace{\mathrm{P}\left(\mathbf{x}_i \mid \mu, label_i = k\right) P\left(label_i = k \middle| \mathbf{x}_i, \mu^0\right)}_{\text{This is a likelihood}}$$

Lets Modify This Idea

An idea:

1. Imagine we start with some u_k^0
2. We *could* calculate the most likely labels for x_i^0 given these u_k^0
3. We *could* then use labels to choose u_k^1 by <u>maximizing likelihood</u>
4. And iterate (to convergence)

$$\mu^0$$

$$labels_i^0 = \arg\max_k \; \mathrm{P}\left(\mathbf{x}_i \mid \mu_k^0\right)$$

… that replaces this likelihood

$$\mu_k^1 = \arg\max_\mu \left\{ \log \prod_{\text{x with label k}} \underbrace{\mathrm{P}\left(\mathbf{x}_i \mid \mu\right)} \right\}$$

$$labels_i^1 = \arg\max_k \; \mathrm{P}\left(\mathbf{x}_i \mid \mu_k^1\right)$$

…

The Modification:

1. Start with some u^0
2. **Calculate the P(labels I x_i)**
3. **Use this <u>probability</u> to choose u_k^1 by <u>maximizing expected likelihood</u>**

$$\mu^0$$

$$P\left(label_i^0 = k \middle| \mathbf{x}_i, \mu^0\right) = \mathrm{P}\left(\mathbf{x}_i \mid \mu^0, label_i^0 = k\right) \mathrm{P}\left(label_i^0 = k\right) \Big/ \mathrm{P}\left(\mathbf{x}_i \middle| \mu^0\right)$$

$$\underbrace{\sum_k \mathrm{P}\left(\mathbf{x}_i \mid \mu_k, label_i = k\right) P\left(label_i = k \middle| \mathbf{x}_i, \mu^0\right)}$$

This is an <u>expected value</u> of the likelihood

Lets Modify This Idea

An idea:

1. Imagine we start with some u_k^0
2. We *could* calculate the most likely labels for x_i^0 given these u_k^0
3. We *could* then use labels to choose u_k^1 by <u>maximizing likelihood</u>
4. And iterate (to convergence)

$$\mu^0$$

$$labels_i^0 = \arg\max_k \; \mathrm{P}\left(\mathbf{x}_i \mid \mu_k^0\right)$$

$$\mu_k^1 = \arg\max_\mu \left\{ \log \prod_{\text{x with label k}} \mathrm{P}\left(\mathbf{x}_i \mid \mu\right) \right\}$$

$$labels_i^1 = \arg\max_k \; \mathrm{P}\left(\mathbf{x}_i \mid \mu_k^1\right)$$

...

The Modification:

1. Start with some u^0
2. **Calculate the P(labels I x_i)**
3. **Use this <u>probability</u> to choose u_k^1 by <u>maximizing expected likelihood</u>**

$$\mu^0$$

$$P\left(label_i^0 = k \middle| \mathbf{x}_i, \mu^0\right) = \mathrm{P}\left(\mathbf{x}_i \mid \mu^0, label_i^0 = k\right) \mathrm{P}\left(label_i^0 = k\right) \Big/ \mathrm{P}\left(\mathbf{x}_i \middle| \mu^0\right)$$

$$\mu^1 = \arg\max_\mu \left\{ \log \prod_i \sum_k \mathrm{P}\left(\mathbf{x}_i \mid \mu, label_i = k\right) P\left(label_i = k \middle| \mathbf{x}_i, \mu^0\right) \right\}$$

Lets Modify This Idea

An idea:

1. Imagine we start with some u_k^0
2. We *could* calculate the most likely labels for x_i^0 given these u_k^0
3. We *could* then use labels to choose u_k^1 by <u>maximizing likelihood</u>
4. And iterate (to convergence)

$$\mu^0$$

$$labels_i^0 = \arg\max_k \; \mathrm{P}\left(\mathbf{x}_i \mid \mu_k^0\right)$$

$$\mu_k^1 = \arg\max_\mu \left\{ \log \prod_{\text{x with label k}} \mathrm{P}\left(\mathbf{x}_i \mid \mu\right) \right\}$$

$$labels_i^1 = \arg\max_k \; \mathrm{P}\left(\mathbf{x}_i \mid \mu_k^1\right)$$

...

The Modification:

1. Start with some u^0
2. **Calculate the P(labels I x_i)**
3. **Use this <u>probability</u> to choose u_k^1 by <u>maximizing expected likelihood</u>**
4. Iterate to convergence

$$\mu^0$$

$$P\left(label_i^0 = k \middle| \mathbf{x}_i, \mu^0\right) = \mathrm{P}\left(\mathbf{x}_i \mid \mu^0, label_i^0 = k\right) \mathrm{P}\left(label_i^0 = k\right) \Big/ \mathrm{P}\left(\mathbf{x}_i \middle| \mu^0\right)$$

$$\mu^1 = \arg\max_\mu \left\{ \log \prod_i \sum_k \mathrm{P}\left(\mathbf{x}_i \mid \mu, label_i = k\right) P\left(label_i = k \middle| \mathbf{x}_i, \mu^0\right) \right\}$$

...

Expectation Maximization (EM)

1. **Initialize parameters**

2. **E Step Estimate <u>probability of hidden labels</u> , Q, given parameters and data (X)**

$$Q = P\left(labels \mid X, parameters^{t-1}\right)$$

3. **M Step Choose new parameters to maximize <u>expected likelihood</u> of parameters given the data, labels and Q**

$$parameters^{t} = \underset{parameters}{\operatorname{argmax}}\, E_{Q}\left[P\left(X \mid labels, parameters\right)\right]$$

Expectation Maximization (EM)

1. **Initialize parameters**

2. **E Step Estimate <u>probability of hidden labels</u> , Q, given parameters and data (X)**

$$Q = P\left(labels \mid X, parameters^{t-1}\right)$$

3. **M Step Choose new parameters to maximize <u>expected likelihood</u> of parameters given the data, labels and Q**

$$parameters^{t} = \underset{parameters}{\operatorname{argmax}}\, E_Q\left[P\left(X \mid labels, parameters\right)\right]$$

$$= \underset{parameters}{\operatorname{argmax}} \sum_{labels} \underbrace{P\left(X \mid labels, parameters\right)}_{L(parameters \mid x,\ labels)} P\left(labels \mid X, parameters^{t-1}\right)$$

Often written as a likelihood function

Expectation Maximization (EM)

1. **Initialize parameters**

2. **E Step Estimate <u>probability of hidden labels</u> , Q, given parameters and data (X)**

$$Q = P\left(labels \mid X, parameters^{t-1}\right)$$

3. **M Step Choose new parameters to maximize <u>expected likelihood</u> of parameters given the data, labels and Q**

$$parameters^{t} = \underset{parameters}{\operatorname{argmax}} E_Q\left[P\left(X \mid labels, parameters\right)\right]$$

$$= \underset{parameters}{\operatorname{argmax}} \sum_{labels} P\left(X \mid labels, parameters\right) P\left(labels \mid X, parameters^{t-1}\right)$$

4. **Iterate**

Expectation Maximization (EM)

1. **Initialize parameters**

2. **E Step Estimate <u>probability of hidden labels</u> , Q, given parameters and data (X)**

$$Q = P\left(labels \mid X, parameters^{t-1}\right)$$

3. **M Step Choose new parameters to maximize <u>expected likelihood</u> of parameters given the data, labels and Q**

$$parameters^{t} = \underset{parameters}{\operatorname{argmax}}\, E_Q\left[P\left(X \mid labels, parameters\right)\right]$$

$$= \operatorname{argmax} \sum_{\cdot} P\left(X \mid labels, parameters\right) P\left(labels \mid X, parameters^{t-1}\right)$$

P(X|Model) *guaranteed* to increase each iteration

Expectation Maximization (EM)

Remember the basic idea!

1.E Step: Use model to estimate distribution of missing data
2.M Step: Use estimate to update model
3.Repeat until convergence

Model is the gaussian distributions

Missing data are the data point labels

Revisting Fuzzy K-Means

Recall that instead of assigning each point x_i to a label k, we calculate the probability of each label for that point (fuzzy membership):

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

Recall that given a set of x_i, all with label k, we select a new μ_k with the update:

$$u_k(n+1) = \sum_{\text{All } x_i} x_i P(\text{label k} \mid x_i, u_k) \Big/ \sum_{\text{All } x_i} P(\text{label k} \mid x_i, u_k)$$

It can be shown that this update rule follows from assuming the gaussian mixture generative models and performing *Expectation-Maximization*

Revisiting Fuzzy K-Means

1. Initialize K centers $\mathbf{u}_k$
2. For each point calculate the probability of membership for each category

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

3. Move the position of each $\mathbf{u}_k$ to the weighted centroid :

$$u_k(n+1) = \sum_{\text{All } x_i} x_i P(\text{label k} \mid x_i, u_k) \Big/ \sum_{\text{All } x_i} P(\text{label k} \mid x_i, u_k)$$

4. Iterate

Generative Model Perspective

1. Initialize parameters
2. E Step Estimate probability over missing labels given previous parameter
3. M Step Choose new parameters to maximize expected likelihood of parameters given estimated probability over labels
4. Iterate

Revisiting K-Means

Generative Model Perspective

1. Initialize K centers $\mathbf{u}_k$

2. Assign each $\mathbf{x}_i$ the label of the nearest center, where the distance between $\mathbf{x}_i$ and $\mathbf{u}_k$ is ⇨ The most likely label k for a point x_i

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

3. Move the position of each $\mathbf{u}_k$ to the centroid of the points with that label ⇨ Maximum likelihood parameter μ_k given most likely label

4. Iterate

This is a form of Viterbi Learning

Revisiting K-Means

Generative Model Perspective

1. Initialize K centers $\mathbf{u}_k$
2. Assign each $\mathbf{x}_i$ the label of the nearest center, where the distance between $\mathbf{x}_i$ and $\mathbf{u}_k$ is

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

⇨ The most likely label k for a point x_i

P(x|Model) *not guaranteed* to increase each iteration

P(x, labels | Model) increases each iteration

Expectation Maximization (EM)

Remember the basic idea!

1.E Step: Use model to estimate distribution of missing data
2.M Step: Use estimate to update model
3.Repeat until convergence

You will see this again with HMMs

Baum-Welch Learning (EM)
Viterbi Learning (Like K-means)

K-Means, Viterbi learning & EM

K-Means and Fuzzy K-means are two related methods that can be seen performing unsupervised, re-estimation learning

Fuzzy K-means => Expectation-Maximization (EM)
K-means => Viterbi Learning

Reveal assumptions about underlying data model

K-means assumes equally likely Gaussians with unit variance

Can relax assumptions by relaxing constraints on model

- **Including explicit covariance matrix**
- **Relaxing assumption that all gaussians are equally likely**

Implications: Non-globular Clusters

Actual Clustering

K-means (K = 2)

Evaluating Cluster Performance

In general, it depends on your goals in clustering

- Robustness
  - Select random samples from data set and cluster
  - Repeat
  - Robust clusters show up in all clusters
- Category Enrichment
  - Look for categories of genes “over-represented” in particular clusters
  - Also used in Motif Discovery

Evaluating clusters – Hypergeometric Distribution

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

- N experiments, p labeled **positive**
- Cluster: k elements, m labeled **positive**
- P-value of *single* cluster containing k elements with at least r **positive**

Similar Genes Can Cluster

Clustered 8600 human genes using expression time course in fibroblasts

(A) Cholesterol biosynthesis

(B) Cell cycle

(C) Immediate early response

(D) Signaling and angiogenesis

(E) Wound healing

(Eisen (1998) PNAS)

Clusters and Motif Discovery

Expression from 15 time points during yeast cell cycle

Tavazoie & Church (1999)

Next Lecture

The other side of the coin… Classification