EM: Expectation Maximization

Sun Kim

Computer Science and Engineering
Bioinformatics Institute
Seoul National University

EM?

Many people say “I know EM is used in many applications but I do not know what it is.”

I hope I can help you understand EM today

Basics

- Model
- Likelihood
- Prior probability
- Posterior probability
- Latent variables

Coin Model

- Tossing a coin produces either
  - head (앞면) or tail (뒷면)
- A model for a coin ?
  - Prob[head], Prob[tail]
- A fair or loaded coin
  - Fair coin: Prob[head] = 0.5, Prob[tail] = 0.5
  - Loaded coin at Casino: Prob[head] = 0.6, Prob[tail] = 0.4

Likelihood

- We have a sequence of tossing a coin:
  HTHHHTT

- Is this generated using a fair coin or a loaded coin?
  - For a fair coin,
    Pr[HTHHHTT | Fair] = 0.5 * 0.5 * 0.5 * 0.5 * 0.5 * 0.5 * 0.5 = .0078125
  - For a loaded coin,
    Pr[HTHHHTT | Loaded] = 0.6 * 0.4 * 0.6 * 0.6 * 0.6 * 0.4 * 0.4 = .0082944

    카지노 딜러가 나를 속였군. 나쁜놈!

Prior Probability

- 카지노 딜러 왈 "**너 기계학습 여름 학교 강의 이해 했니?**"

- The government regulates Pr[Fair]= 0.99, Pr[Loaded]= 0.01
  ➔ **Prior probability**

Posterior Probability

- **"카지노 딜러가 속였을까요?"** 를 수식으로 나타내면
  - **Pr[HTHHHTT | Fair]** vs. **Pr[HTHHHTT | Loaded]**
- 그런데 이건 계산이 직접 안되어서 Bayes rule을 이용해야함.
- Pr[Fair | HTHHHTT]
  = Pr[HTHHHTT | Fair] * Pr[Fair] / P[HTHHHTT]
- **Okay, but Pr[HTHHHTT] ??**
- Pr[HTHHHTT]
  = Pr[HTHHHTT | Fair] * Pr[Fair]+ Pr[HTHHHTT | Loaded] * Pr[Loaded]

Posterior Probability

- **Weighted likelihood로 생각하면 쉬움.**

- Pr[Fair | HTHHHTT] = Pr[HTHHHTT | Fair] * Pr[Fair] / P[HTHHHTT]
- Pr[Loaded | HTHHHTT] = Pr[HTHHHTT | Fair] * Pr[Loaded] / P[HTHHHTT]

Pr[Fair | HTHHHTT] vs. **Pr[Loaded | HTHHHTT]**

➔➔

Pr[HTHHHTT | Fair]*Pr[Fair] vs. **Pr[HTHHHTT|Fair]*Pr[Loaded]**

ML vs. MAP

- When we set model parameters,
- **ML (maximum likelihood)**
  - Among all possible model parameter configurations,
  - Select one that is ML.

- **MAP (maximum a posteriori)**
  - Among all possible model parameter configurations,
  - Select one that is MAP.

Three Problems to be reviewed today

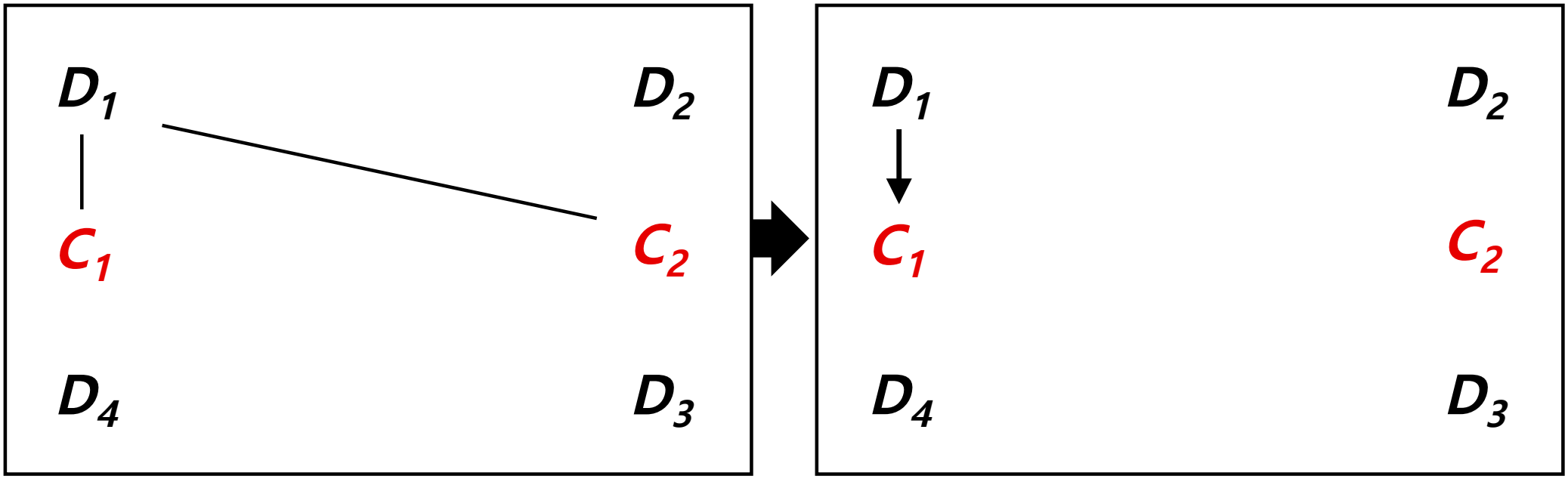
- Gaussian Mixture
- Baum-Welch algorithm for HMM model parameter estimation
- Motif prediction

Problem 1: K Clustering Algorithms

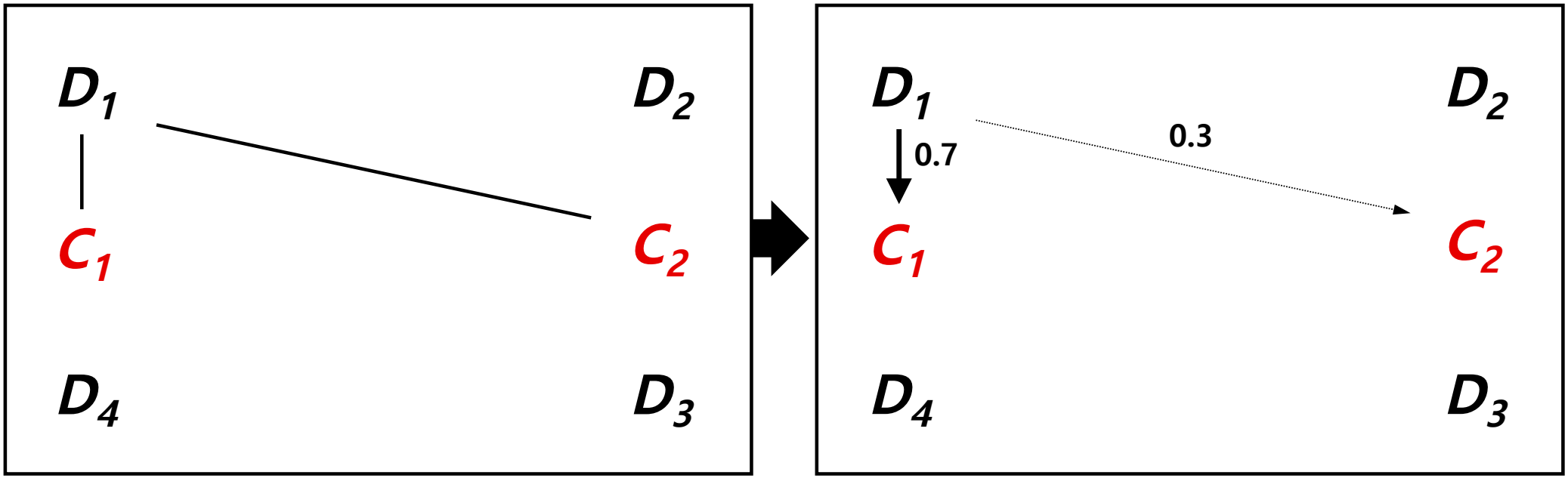
K Clustering Algorithms and EM

- K means clustering algorithm
- A probabilistic version of K means clustering algorithm
- K Gaussian mixtures algorithm

K means clustering algorithm



A probabilistic version of K means clustering algorithm



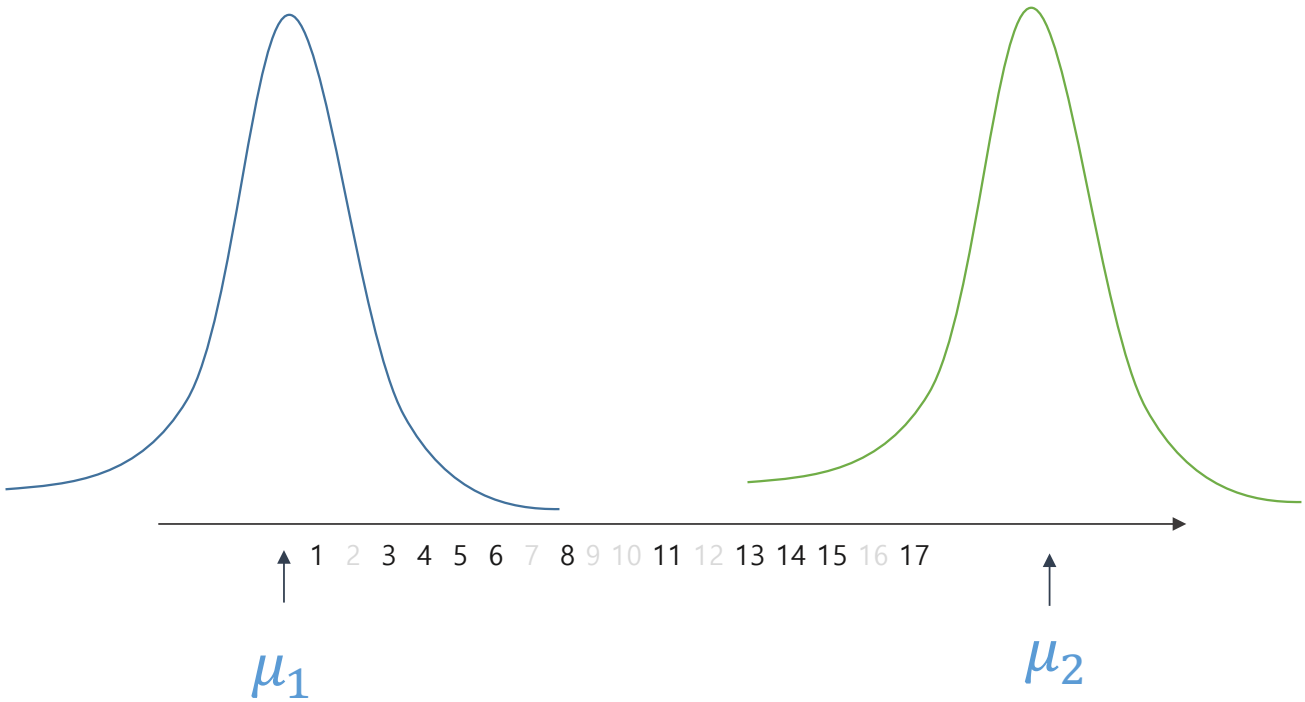
민주적인 결정 : 모든 데이터가 참여해서 cluster center를 정함.
➔ model parameter수가 증가함.

Two Gaussian Mixture EM Clustering

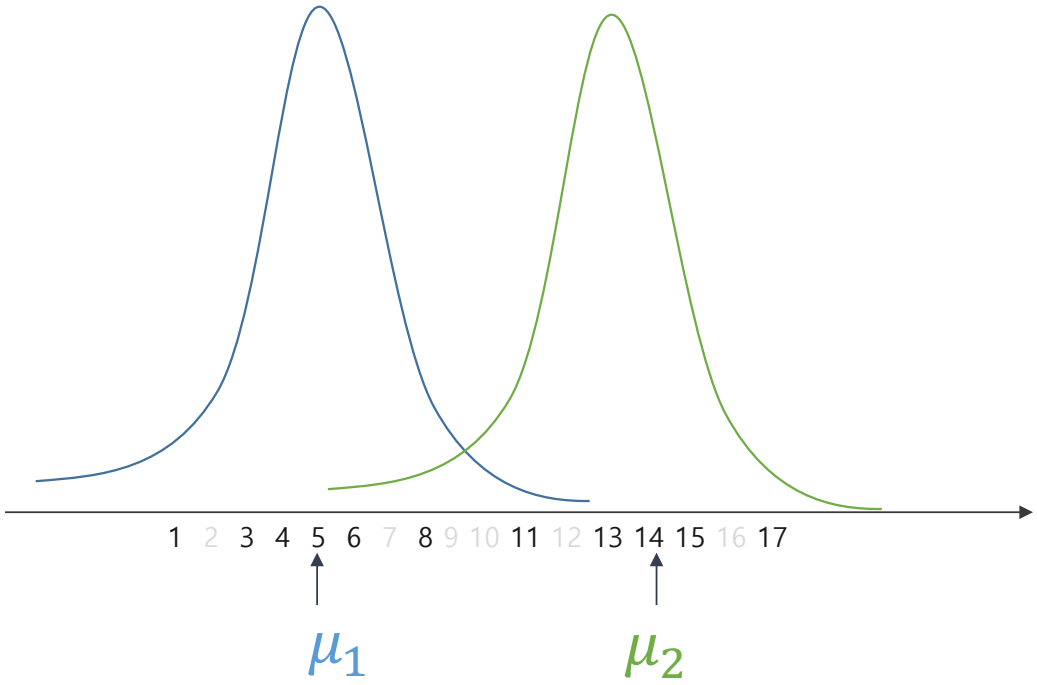
아래 숫자들이 2개의 Gaussian distribution에 의해 생성이 되었는데, 2개의 그룹으로 클러스링을 하고자 한다.

1 3 4 5 6 8 9 11 12 13 14 15 17 18

ESTIMATION1: 아래 2개의 Gaussian distribution



ESTIMATION2: 아래 2개의 Gaussian distribution



이 estimation 이 더 좋은가요? 왜?

If we know which distribution the number are from.

1	3	4	5	6	8	11	13	14	15	17
G1	G1	G1	G1	G1	G1					
						G2	G2	G2	G2	G2

μ_1 = (1 + 3 + 4 + 5 + 6 + 8) / 6 = 4.5

μ_2 = (11 + 13 + 14 + 15 +17)/ 5 = 14

Of course, we do not know which distribution the numbers are from.

1	3	4	5	6	8	11	13	14	15	17
G1	G1	G1	G1	G1	G1	G1	G1	G1	G1	G1
G2	G2	G2	G2	G2	G2	G2	G2	G2	G2	G2

Democracy again using latent variables:

Of course, we do not know which distribution the numbers are from.

1	3	4	5	6	8	11	13	14	15	17
P[G1]	P[G1]	P[G1]	P[G1]	P[G1]	P[G1]	P[G1]	P[G1]	P[G1]	P[G1]	P[G1]
P[G2]	P[G2]	P[G2]	P[G2]	P[G2]	P[G2]	P[G2]	P[G2]	P[G2]	P[G2]	P[G2]

STEP 1: Estimate P[1 from G1], P[3 from G1], …

STEP 2: μ_1 = (P[1 from G1] * 1 + P[3 from G1] * 3 + …) / (P[1 from G1] + P[3 from G1] + … + P[17 from G1])

개념을 이해 하셨으면 이제 구체적인 계산 방법을 논의 합니다.

The material is from
Andrew Rosenberg at Queens College / CUNY
http://eniac.cs.qc.cuny.edu/andrew/ml/syllabus.html

Mixture Models

- Formally a Mixture Model is the weighted sum of a number of pdfs where the weights are determined by a distribution, π

$$p(x) = \pi_0 f_0(x) + \pi_1 f_1(x) + \pi_2 f_2(x) + \ldots + \pi_k f_k(x)$$

$$\text{where } \sum_{i=0}^{k} \pi_i = 1$$

$$\boxed{p(x) = \sum_{i=0}^{k} \pi_i f_i(x)}$$

Mixture Models

- Formally a Mixture Model is the weighted sum of a number of pdfs where the weights are determined by a distribution, π

$$p(x) = \pi_0 f_0(x) + \pi_1 f_1(x) + \pi_2 f_2(x) + \ldots + \pi_k f_k(x)$$

$$\text{where } \sum_{i=0}^{k} \pi_i = 1$$

$$\boxed{p(x) = \sum_{i=0}^{k} \pi_i f_i(x)}$$

Gaussian Mixture Models

- GMM: the weighted sum of a number of Gaussi ans where the weights are determined by a dist ribution, π

$$p(x) = \pi_0 N(x|\mu_0, \Sigma_0) + \pi_1 N(x|\mu_1, \Sigma_1) + \ldots + \pi_k N(x|\mu_k, \Sigma_k)$$

$$\text{where } \sum_{i=0}^{k} \pi_i = 1$$

$$\boxed{p(x) = \sum_{i=0}^{k} \pi_i N(x|\mu_k, \Sigma_k)}$$

Expectation Maximization

- Both the training of GMMs and Graphical Models with latent variables can be accomplished using Expectation Maximizatio n
  - Step 1: Expectation (E-step)
    - Evaluate the "responsibilities" of each cluster with the current parameters
  - Step 2: Maximization (M-step)
    - Re-estimate parameters using the existing "responsibilities"
- Similar to k-means training.

Latent Variable Representation

- We can represent a GMM involving a latent variable

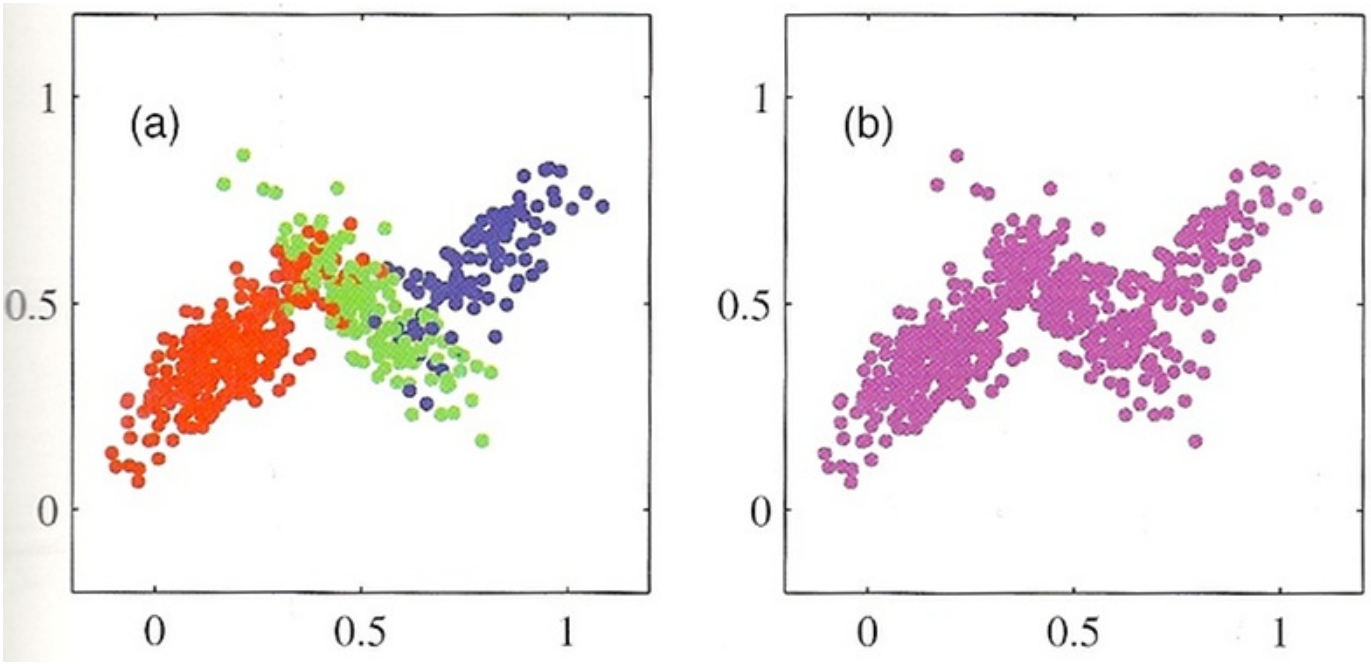
$$p(x) = \sum_{i=0}^{k} \pi_i N(x|\mu_k, \Sigma_k) = \sum_{z} p(z)p(x|z)$$

$$p(z) = \prod_{k=1}^{K} \pi_k^{z_k} \qquad p(x|z) = \prod_{k=1}^{K} N(x|\mu_k, \Sigma_k)^{z_k}$$

- What does this give us?

TODO: plate notation

GMM data and Latent variables



One last bit

- We have representations of the joint p(x,z) and the marginal, p(x)…
- The conditional of p(z|x) can be derived using B ayes rule.
  - The **responsibility** that a mixture component takes f or explaining an observation x.

$$\tau(z_k) = p(z_k = 1|x) = \frac{p(z_k = 1)p(x|z_k = 1)}{\sum_{j=1}^{K} p(z_j = 1)p(x|z_j = 1)}$$

$$= \frac{\pi_k N(x|\mu_k, \Sigma_k)}{\sum_{j=1}^{K} \pi_j N(x|\mu_j, \Sigma_j)}$$

Maximum Likelihood over a GMM

- As usual: Identify a likelihood function

$$\ln p(x|\pi,\mu,\Sigma) = \sum_{n=1}^{N} \ln \left\{ \sum_{k=1}^{K} \pi_k N(x_n|\mu_k,\Sigma_k) \right\}$$

- And set partials to zero…

Maximum Likelihood of a GMM

- Optimization of means.

$$\ln p(x|\pi,\mu,\Sigma) = \sum_{n=1}^{N} \ln \left\{ \sum_{k=1}^{K} \pi_k N(x_n|\mu_k,\Sigma_k) \right\}$$

$$\begin{aligned} \frac{\partial \ln p(x|\pi,\mu,\Sigma)}{\partial \mu_k} &= \sum_{n=1}^{N} \frac{\pi_k N(x_n|\mu_k,\Sigma_k)}{\sum_j \pi_j N(x_n|\mu_j,\Sigma_j} \Sigma_k^{-1}(x_k - \mu_k) = 0 \\ &= \sum_{n=1}^{N} \tau(z_{nk}) \Sigma_k^{-1}(x_k - \mu_k) = 0 \end{aligned}$$

$$\boxed{\mu_k = \frac{\sum_{n=1}^{N} \tau(z_{nk}) x_n}{\sum_{n=1}^{N} \tau(z_{nk})}}$$

Maximum Likelihood of a GMM

- Optimization of covariance

$$\ln p(x|\pi,\mu,\Sigma) = \sum_{n=1}^{N} \ln \left\{ \sum_{k=1}^{K} \pi_k N(x_n|\mu_k,\Sigma_k) \right\}$$

$$\boxed{\Sigma_k = \frac{1}{\sum_{n=1}^{N} \tau(z_{nk})} \sum_{n=1}^{N} \tau(z_{nk})(x_k-\mu_k)(x_k-\mu_k)^T}$$

- Note the similarity to the regular MLE without **responsibility terms**.

Maximum Likelihood of a GMM

- Optimization of mixing term

$$\ln p(x|\pi,\mu,\Sigma) + \lambda\left(\sum_{k=1}^{K}\pi_k - 1\right)$$

$$0 = \sum_{n=1}^{N}\frac{\pi_k N(x_n|\mu_k,\Sigma_k)}{\sum_j \pi_j N(x_n|\mu_j,\Sigma_j} + \lambda$$

$$\boxed{\pi_k = \frac{\sum_{n=1}^{N}\tau(z_n k)}{N}}$$

MLE of a GMM

$$\boxed{\mu_k = \frac{\sum_{n=1}^{N} \tau(z_{nk}) x_n}{N_k}}$$

$$\boxed{\Sigma_k = \frac{1}{N_k} \sum_{n=1}^{N} \tau(z_{nk})(x_k - \mu_k)(x_k - \mu_k)^T}$$

$$\boxed{\pi_k = \frac{N_k}{N}}$$

$$N_k = \sum_{n=1}^{N} \tau(z_n k)$$

EM for GMMs

- Initialize the parameters
  - Evaluate the log likelihood
- Expectation-step: Evaluate the responsibilities
- Maximization-step: Re-estimate Parameters
  - Evaluate the log likelihood
  - Check for convergence

EM for GMMs

- E-step: Evaluate the Responsibilities

$$\tau(z_{nk}) = \frac{\pi_k N(x_n|\mu_k, \Sigma_k)}{\sum_{j=1}^{K} \pi_j N(x_n|\mu_j, \Sigma_j)}$$

EM for GMMs

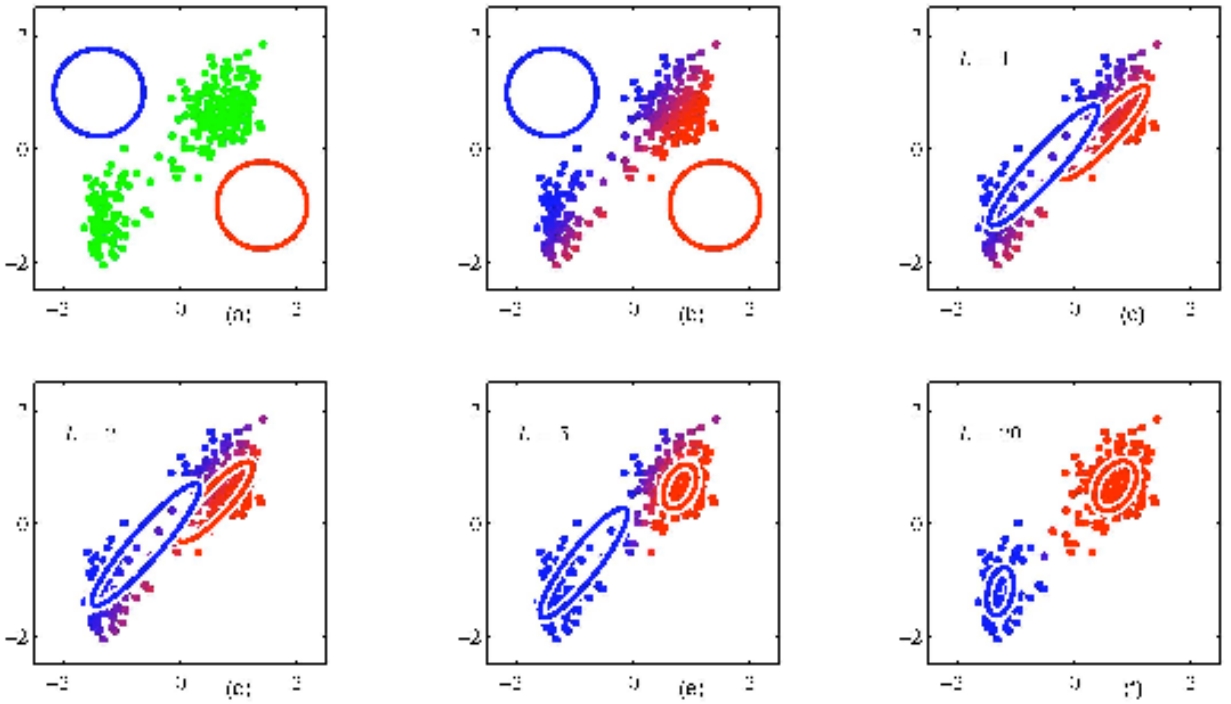
- M-Step: Re-estimate Parameters

$$\mu_k^{new} = \frac{\sum_{n=1}^{N} \tau(z_{nk}) x_n}{N_k}$$

$$\Sigma_k^{new} = \frac{1}{N_k} \sum_{n=1}^{N} \tau(z_{nk})(x_k - \mu_k^{new})(x_k - \mu_k^{new})^T$$

$$\pi_k^{new} = \frac{N_k}{N}$$

Visual example of EM



Relationship to K-means

- K-means makes **hard** decisions.
  - Each data point gets assigned to a single cluster.
- GMM/EM makes **soft** decisions.
  - Each data point can yield a posterior p(z|x)
- Soft K-means is a special case of EM.

Soft means as GMM/EM

- Assume equal covariance matrices for every mixture component: $\epsilon \mathbf{I}$
- Likelihood: $$p(x|\mu_k, \Sigma_k) = \frac{1}{(2\pi\epsilon)^{M/2}} \exp\left\{-\frac{1}{2\epsilon}\|x - \mu_k\|^2\right\}$$
- Responsibilities: $$\tau(z_{nk}) = \frac{\pi_k \exp\left\{-\|x_n - \mu_k\|^2/2\epsilon\right\}}{\sum_j \pi_j \exp\left\{-\|x_n - \mu_j\|^2/2\epsilon\right\}}$$
- As epsilon approaches zero, the responsibility approaches unity.

Soft K-Means as GMM/EM

- Overall Log likelihood as epsilon approaches zero:

$$\mathbb{E}_z[\ln p(X, Z|\mu, \Sigma, \pi)] \rightarrow -\frac{1}{2}\sum_{n=1}^{N}\sum_{k=1}^{K} r_{nk}\|x_n - \mu_k\|^2 + const.$$

- The expectation of soft k-means is the intercluster variability
- Note: only the means are reestimated in Soft K-means.
  - The covariance matrices are all tied.

General form of EM

- Given a joint distribution over observed and lat ent variables: $p(X, Z|\theta)$
- Want to maximize: $p(X|\theta)$

1. Initialize parameters θ^{old}
2. E Step: Evaluate:

$$p(Z|X, \theta^{old})$$

3. M-Step: Re-estimate parameters (based on expectati on of complete-data log likelihood)

$$\theta^{new} = \text{argmax}_\theta \sum_Z p(Z|X, \theta^{old}) \ln p(X, Z|\theta)$$

4. Check for convergence of params or likelihood

Problem 2:
HMM Model Parameter Estimation

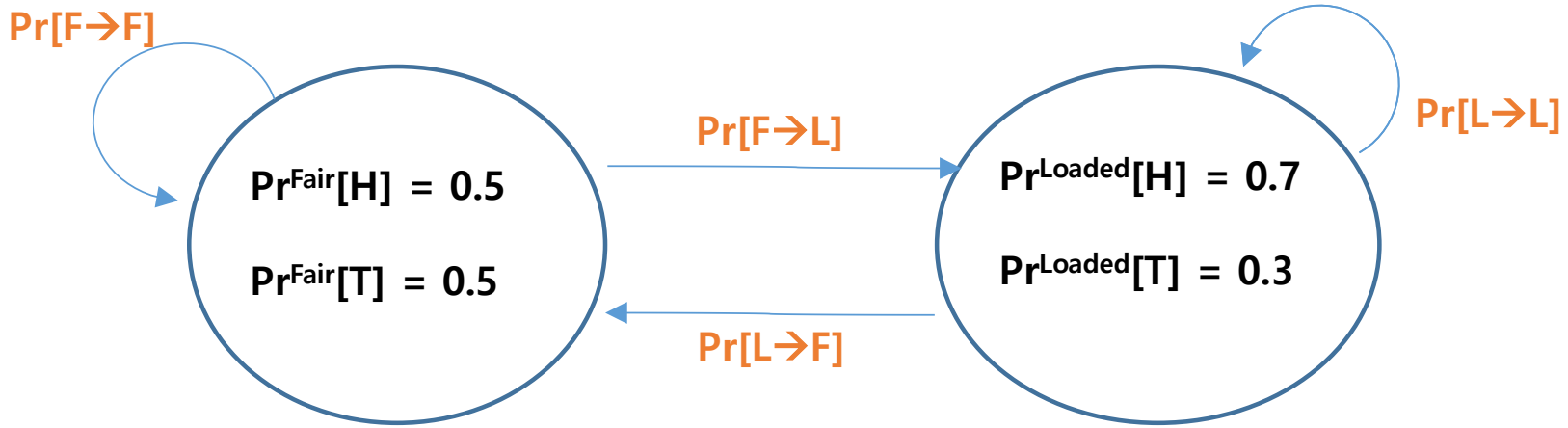
Hidden Markov Model

- We have a sequence of tossing a coin:
  HTHHHTH

- 이 번에는 카지노 딜러가 fair, loaded coin을 번갈아서 사용했는데, 이 걸 어떻게 모델을 하나?

Hidden Markov Model

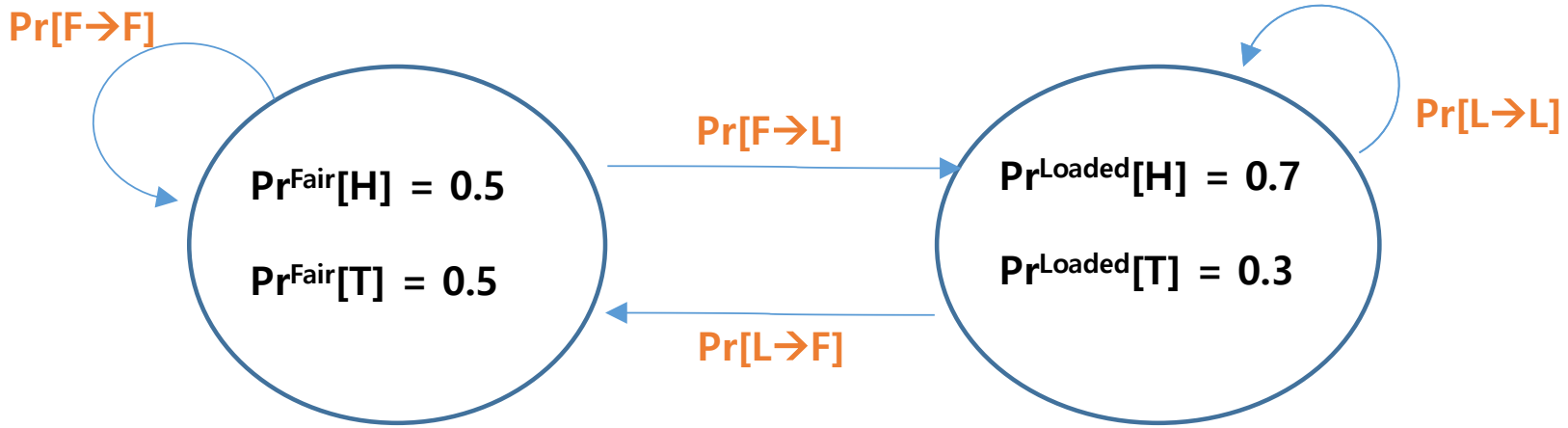
- We have a sequence of tossing a coin:
  HTHHHTH

- 

Note that there are TWO different types of model parameters.
- **Character emission**
- **State transition**

HMM1

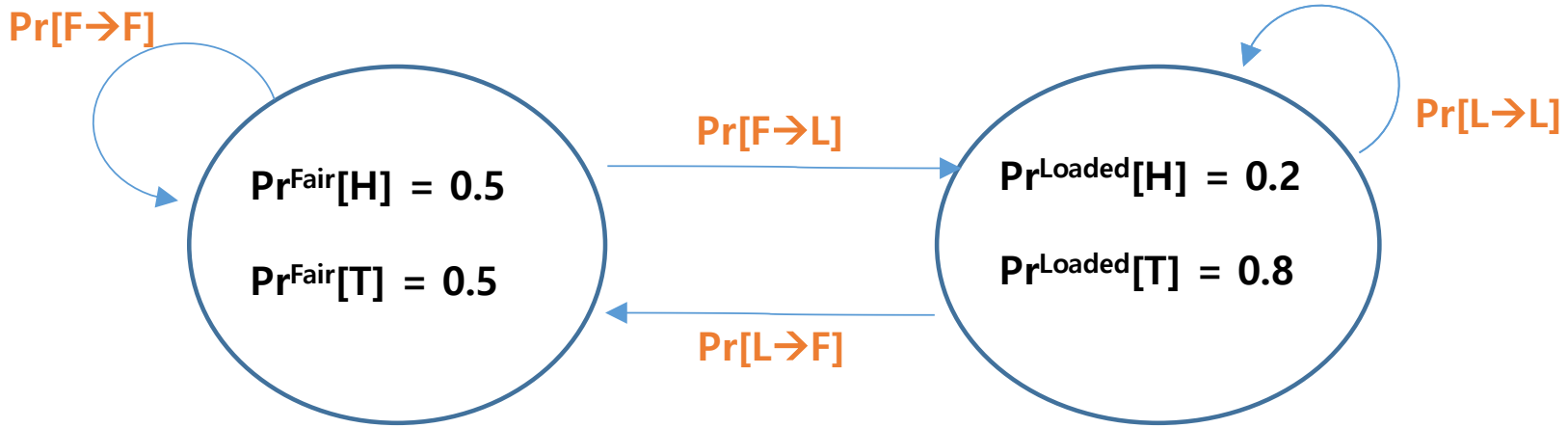
- We have a sequence of tossing a coin:
  HTHHHTH

- 

HMM1 이 좋은 가요? In terms of likelihood, Pr[HTHHHTH | HMM1].

HMM2

- We have a sequence of tossing a coin:
  HTHHHTH

- 

HMM2 이 좋은 가요? In terms of likelihood, Pr[HTHHHTH | HMM2].

HMM model parameter estimation

- **Algorithm**:
  1. 가능한 모든 HMM을 고려해서,
     1. HMM1, HMM2, ……
  2. 가장 likelihood가 좋은 HMM을 선정한다.

- 그런데 가능한 모든 HMM의 수는 무수히 많다. 따라서 위의 알고리즘은 불가능.
  - EM !!
  - Baum-Welch algorithm!

Computing Likelihood **Pr[HTHHHTH | HMM]**

H	T	H	H	H	T	H
F	F	F	F	F	F	F
L	L	L	L	L	L	L

Pr[F→F]	Pr[F→F]	Pr[F→F]	Pr[F→F]	Pr[F→F]	Pr[F→F]
Pr[F→L]	Pr[F→L]	Pr[F→L]	Pr[F→L]	Pr[F→L]	Pr[F→L]
Pr[L→F]	Pr[L→F]	Pr[L→F]	Pr[L→F]	Pr[L→F]	Pr[L→F]
Pr[L→L]	Pr[L→L]	Pr[L→L]	Pr[L→L]	Pr[L→L]	Pr[L→L]

HTHHHTH를 이 HMM으로 생성하는 경우의 수는?

너무 많아서 계산이 불가능해지는데, 다행히 dynamic programming으로 쉽게 계산 가능.
→ Forward or Backward algorithm

Baum-Welch algorithm for HMM model parameter estimation

- Set HMM model parameters randomly.
- Repeat until no improvement in likelihood.
  - Estimate model parameters in a democratic way (How?)
  - Compute likelihood

Estimate model parameters in a democratic way

- Define **latent variables**.
- (Expectation) Estimate latent variables.
- (Maximization) Estimate model parameters.

Latent variables

- What were the latent variables for the Gaussian mixture problem?

1	3	4	5	6	8	11	13	14	15	17
P[G1]	P[G1]	P[G1]	P[G1]	P[G1]	P[G1]	P[G1]	P[G1]	P[G1]	P[G1]	P[G1]
P[G2]	P[G2]	P[G2]	P[G2]	P[G2]	P[G2]	P[G2]	P[G2]	P[G2]	P[G2]	P[G2]

STEP 1: Estimate P[1 from G1], P[3 from G1], …

STEP 2: μ_1 = (P[1 from G1] * 1 + P[3 from G1] * 3 + …) / 11

Latent variables for HMM model parameter estimation

- Unfortunately, complicated.
- But if you understand why we need latent variables, you are okay.

- **Why?**
- To estimate model parameters in M-step !
  - **$Pr^{Fair}[H]$, $Pr^{Fair}[T]$, $Pr^{Loaded}[H]$, $Pr^{Loaded}[T]$**
  - **Pr[F→F], Pr[F→L], Pr[L→F], Pr[L→L]**

- Then, how to define latent variables in E-step?
  - Of course, different latent variables for different types of parameters.

Latent variables for **"counting"**

H	T	H	H	H	T	H
#F1	#F2	#F3	#F4	#F5	#F6	#F7
#L1	#L2	#L3	#L4	#L5	#L6	#L7

#[F1→F2] #[F2→F3] #[F3→F4] #[F4→F5] #[F5→F6] #[F6→F7]
#[F1→L2] #[F2→L3] #[F3→L4] #[F4→L5] #[F5→L6] #[F6→L7]
#[L1→F2] #[L2→F3] #[L3→F4] #[L4→F5] #[L5→F6] #[L6→F7]
#[L1→L2] #[L2→L3] #[L3→L4] #[L4→L5] #[L5→L6] #[L6→L7]

HTHHHTH를 이 HMM으로 생성하는 경우의 수는?

너무 많아서 계산이 불가능해지는데, 다행히 dynamic programming으로 쉽게 계산 가능.
→ Forward or Backward algorithm

How to estimate Latent variables for "counting"

H	T	H	H	H	T	H
?	?	?	#F4	?	?	?
?	?	?	#L4	?	?	?

#[?→?] #[?→?] #[?→?] #[F4→F5] #[?→?] #[?→?]
#[F4→L5]
#[L4→F5]
#[L4→L5]

#F4를 계산하려면 앞, 뒤로 경우의 수가 너무 많음.
다행히 dynamic programming으로 쉽게 계산 가능.
→ Forward or Backward algorithm

M-step: maximum likelihood estimation of model parameters.

H	T	H	H	H	T	H
#F1	#F2	#F3	#F4	#F5	#F6	#F7
#L1	#L2	#L3	#L4	#L5	#L6	#L7

#[F1→F2]	#[F2→F3]	#[F3→F4]	#[F4→F5]	#[F5→F6]	#[F6→F7]
#[F1→L2]	#[F2→L3]	#[F3→L4]	#[F4→L5]	#[F5→L6]	#[F6→L7]
#[L1→F2]	#[L2→F3]	#[L3→F4]	#[L4→F5]	#[L5→F6]	#[L6→F7]
#[L1→L2]	#[L2→L3]	#[L3→L4]	#[L4→L5]	#[L5→L6]	#[L6→L7]

$$\mathbf{Pr^{Fair}[H] = (\#F1+\#F3+\#F4+\#F5+\#F7)/ (\#F1+\#F3+\#F4+\#F5+\#F7+ \#L1+\#L3+\#L4+\#L5+\#L7)}$$

More formal material by Andrew McCallum at University of Massachusetts Amherst

- https://people.cs.umass.edu/~mccallum/courses/inlp2004a/lect10-hmm2.pdf

Larry Moss at Indiana University Bloomington

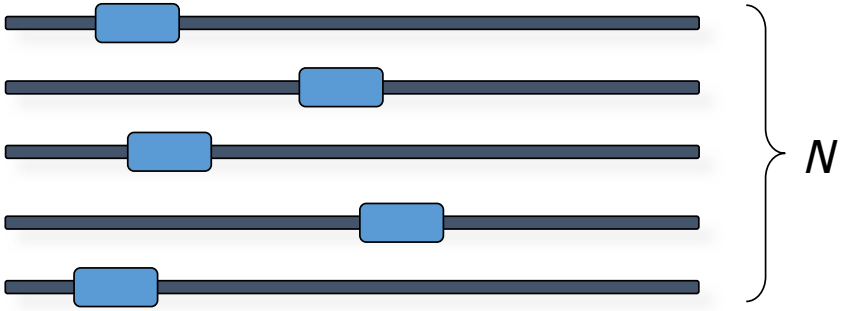
- A good example of HMM model parameter estimation

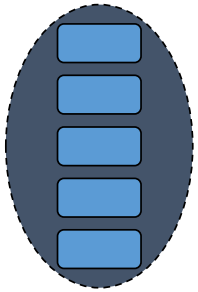
- http://www.indiana.edu/~iulg/moss/hmmcalculations.pdf

Problem 3:
Motif Discovery

Motif Discovery Problem: a search problem

Input: *N* sequences

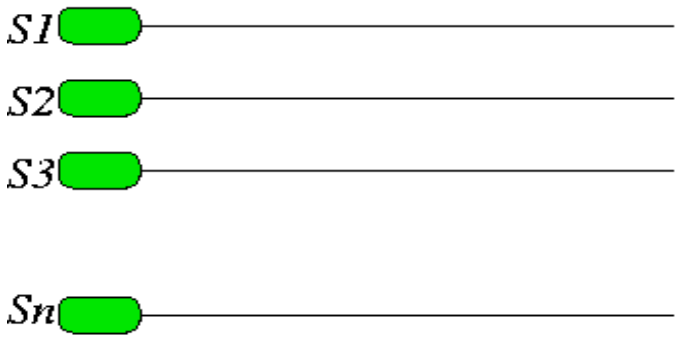




Output: set of conserved regions

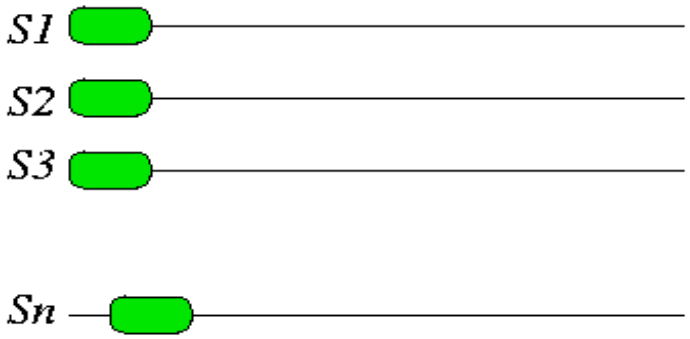
Motif search in unaligned seqs: an optimization problem

We are looking for a motif model with a set of fixed length sequences.



Motif model 1, M1

Score(M1) = H(M1 || R)



Motif model 2, M2

Score(M2) = H(M2 || R)

Solution: select a model Mi with the best solution

MEME, a motif discovery algorithm using EM in 1994 and it is still the best, showing how powerful EM is.

From: Machine Learning Journal, 21, 51–83 (1995)

Unsupervised Learning of Multiple Motifs in Biopolymers Using Expectation Maximization

TIMOTHY L. BAILEY tbailey@cs.ucsd.edu

CHARLES ELKAN elkan@cs.ucsd.edu

Department of Computer Science and Engineering, University of California, San Diego, La Jolla, California 92093-0114

Editor: Lawrence Hunter

Abstract. The MEME algorithm extends the expectation maximization (EM) algorithm for identifying motifs in unaligned biopolymer sequences. The aim of MEME is to discover new motifs

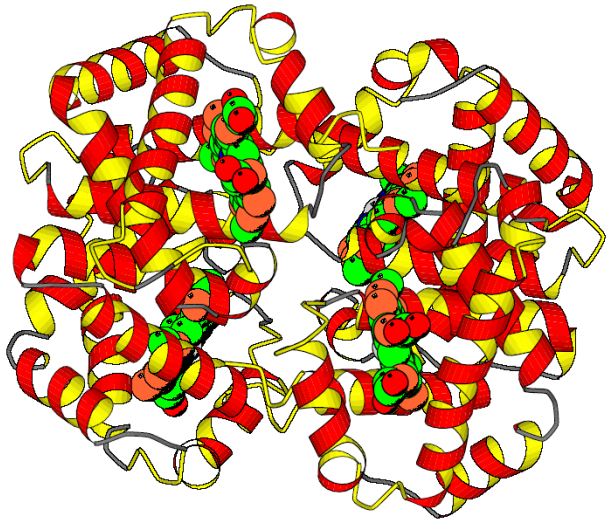
From
Eric Xing at Carnegie Mellon University

http://www.cs.cmu.edu/~epxing/Class/10810-06/ppt/lecture6.ppt

Example: Globin Motifs

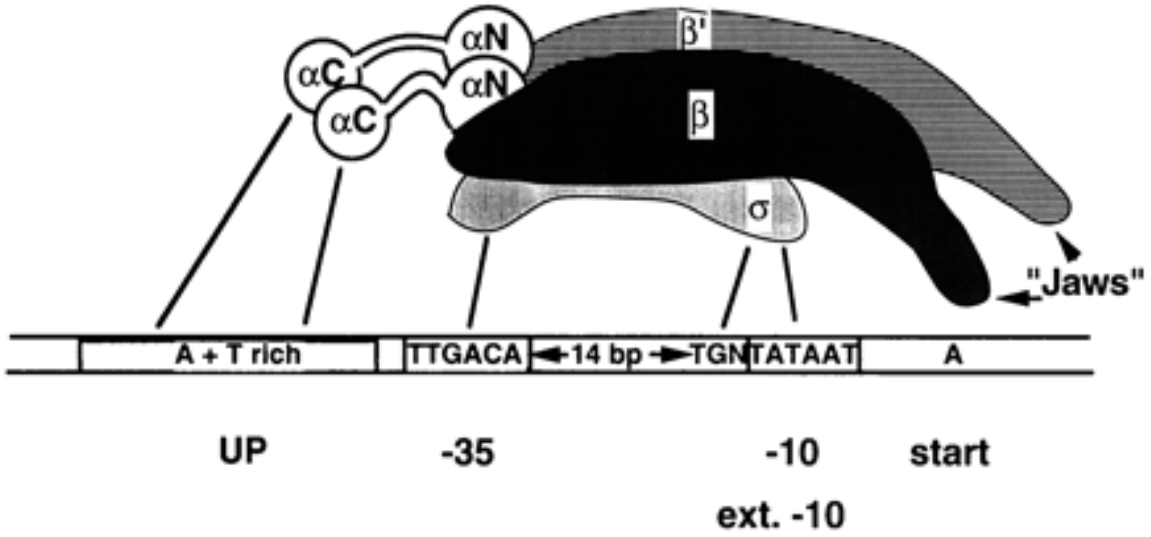
```
          xxxxxxxxxx.xxxxxxxxx.xxxxx.........xxxxxx.xxxxxxx.xxxxxxxxx.xxxxxxxxx
HAHU      V.LSPADKTN..VKAAWGKVG.AHAGE.........YGAEAL.ERMFLSF..PTTKTYFPH.FDLS.HGSA
HAOR      M.LTDAEKKE..VTALWGKAA.GHGEE.........YGAEAL.ERLFQAF..PTTKTYFSH.FDLS.HGSA
HADK      V.LSAADKTN..VKGVFSKIG.GHAEE.........YGAETL.ERMFIAY..PQTKTYFPH.FDLS.HGSA
HBHU      VHLTPEEKSA..VTALWGKVN.VDEVG.........G.EAL.GRLLVVY..PWTQRFFES.FGDL.STPD
HBOR      VHLSGGEKSA..VTNLWGKVN.INELG.........G.EAL.GRLLVVY..PWTQRFFEA.FGDL.SSAG
HBDK      VHWTAEEKQL..ITGLWGKVNvAD.CG.........A.EAL.ARLLIVY..PWTQRFFAS.FGNL.SSPT
MYHU      G.LSDGEWQL..VLNVWGKVE.ADIPG.........HGQEVL.IRLFKGH..PETLEKFDK.FKHL.KSED
MYOR      G.LSDGEWQL..VLKVWGKVE.GDLPG.........HGQEVL.IRLFKTH..PETLEKFDK.FKGL.KTED
IGLOB     M.KFFAVLALCiVGAIASPLT.ADEASlvqsswkavsHNEVEIlAAVFAAY.PDIQNKFSQFaGKDLASIKD
GPUGNI    A.LTEKQEAL..LKQSWEVLK.QNIPA.........HS.LRL.FALIIEA.APESKYVFSF.LKDSNEIPE
GPYL      GVLTDVQVAL..VKSSFEEFN.ANIPK.........N.THR.FFTLVLEiAPGAKDLFSF.LKGSSEVPQ
GGZLB     M.L.DQQTIN..IIKATVPVLkEHGVT.........ITTTF.YKNLFAK.HPEVRPLFDM.GRQ..ESLE

          xxxxx.xxxxxxxxxxxx..xxxxxxxxxxxxxx..xxxxxxx.xxxxxxx...xxxxxxxxxxxxxxx
HAHU      QVKGH.GKKVADA.LTN......AVA.HVDDMPNA...LSALS.D.LHAHKL....RVDPVNF.KLLSHCLL
HAOR      QIKAH.GKKVADA.L.S......TAAGHFDDMDSA...LSALS.D.LHAHKL....RVDPVNF.KLLAHCIL
HADK      QIKAH.GKKVAAA.LVE......AVN.HVDDIAGA...LSKLS.D.LHAQKL....RVDPVNF.KFLGHCFL
HBHU      AVMGNpKVKAHGK.KVLGA..FSDGLAHLDNLKGT...FATLS.E.LHCDKL....HVDPENF.RL.LGNVL
HBOR      AVMGNpKVKAHGA.KVLTS..FGDALKNLDDLKGT...FAKLS.E.LHCDKL....HVDPENFNRL..GNVL
HBDK      AILGNpMVRAHGK.KVLTS..FGDAVKNLDNIKNT...FAQLS.E.LHCDKL....HVDPENF.RL.LGDIL
MYHU      EMKASeDLKKHGA.TVL......TALGGILKKKGHH..EAEIKPL.AQSHATK...HKIPVKYLEFISECII
MYOR      EMKASaDLKKHGG.TVL......TALGNILKKKGQH..EAELKPL.AQSHATK...HKISIKFLEYISEAII
IGLOB     T.GA...FATHATRIVSFLseVIALSGNTSNAAAV...NSLVSKL.GDDHKA....R.GVSAA.QF..GEFR
GPUGNI    NNPK...LKAHAAVIFKTI...CESATELRQKGHAVwdNNTLKRL.GSIHLK....N.KITDP.HF.EVMKG
GPYL      NNPD...LQAHAG.KVFKL..TYEAAIQLEVNGAVAs.DATLKSL.GSVHVS....K.GVVDA.HF.PVVKE
GGZLB     Q......PKALAM.TVL......AAAQNIENLPAIL..PAVKKIAvKHCQAGVaaaH.YPIVGQELL.LGAIK

          xxxxxxxxxx.xxxxxxxxx.xxxxxxxxxxxxxxxxxxxxxxx..x
HAHU      VT.LAA.H..LPAEFTPA..VHASLDKFLASV.STVLTS..KY..R
HAOR      VV.IAR.H..CPGEFTPS..AHAAMDKFLSKV.ATVLTS..KY..R
HADK      VV.VAI.H..HPAALTPE..VHASLDKFMCAV.GAVLTA..KY..R
HBHU      VCVIAH.H..FGKEFTPP..VQAAYQKVVAGV.ANALAH..KY..H
HBOR      IVVLAR.H..FSKDFSPE..VQAAWQKLVSGV.AHALGH..KY..H
HBDK      IIVLAA.H..FTKDFTPE..CQAAWQKLVRVV.AHALAR..KY..H
MYHU      QV.LQSKHPgDFGADAQGA.MNKALELFRKDM.ASNYKELGFQ..G
MYOR      HV.IQSKHSaDFGADAQAA.MGKALELFRNDM.AAKYKEFGFQ..G
IGLOB     TA.LVA.Y..LQANVSWGDnVAAAWNKA.LDN.TFAIVV..PR..L
GPUGNI    ALLGTIKEA.IKENWSDE..MGQAWTEAYNQLVATIKAE..MK..E
GPYL      AILKTIKEV.VGDKWSEE..LNTAWTIAYDELAIIIKKE..MKdaA
GGZLB     EVLGDAAT..DDILDAWGK.AYGVIADVFIQVEADLYAQ..AV..E
```

Hemoglobin alpha subunit



DNA Motif

- **RNA polymerase-promotor interactions**
  - Transcription Initiation in *E. coli*



- In *E. coli* transcription is initiated at the ***promotor*** , whose sequence is r ecognised by the ***Sigma factor*** of RNA polymerase.

Motif Representation

Determinism 2:
Regular Expressions

- The characteristic motif of a Cys-Cys-His-His zinc finger DNA binding domain has ***regular expression***

C-X(2,4)-C-X(3)-[LIVMFYWC]-X(8)-H-X(3,5)-H

- Here, as in algebra, **X** is unknown. The 29 a.a. sequence of an example domain 1S P1 is as follows, clearly fitting the model.

1SP1:

KKFACPECPKRFMRSDHLSKHIKTHQNKK

Weight Matrix Model (WMM)

- **Weight matrix model (WMM) = Stochastic consensus sequence**

A	9	214	63	142	118	8
C	22	7	26	31	52	13
G	18	2	29	38	29	5
T	193	19	124	31	43	216

Counts from 242 known σ^{70} sites

A	.04	.88	.26	.59	.49	.03
C	.09	.03	.11	.13	.21	.05
G	.07	.01	.12	.16	.12	.01
T	.80	.08	.51	.13	.18	.83

encies: θ_{ij}

Relative frequ

less interesting

more interesting

- Weight matrices are also known as
  - Position-specific scoring matrices
  - Position-specific probability matrices
  - Position-specific weight matrices

- A motif is *interesting* if it is very different from the background distribution

Relative entropy

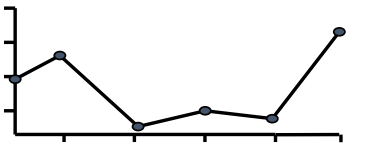
- A motif is *interesting* if it is very different from the background distribution
- Use relative entropy:

$$\sum_{\text{position } l} \left(\sum_{\text{letter } i} \theta_{li} \log_2 \frac{\theta_{li}}{\theta_{0i}} \right)$$

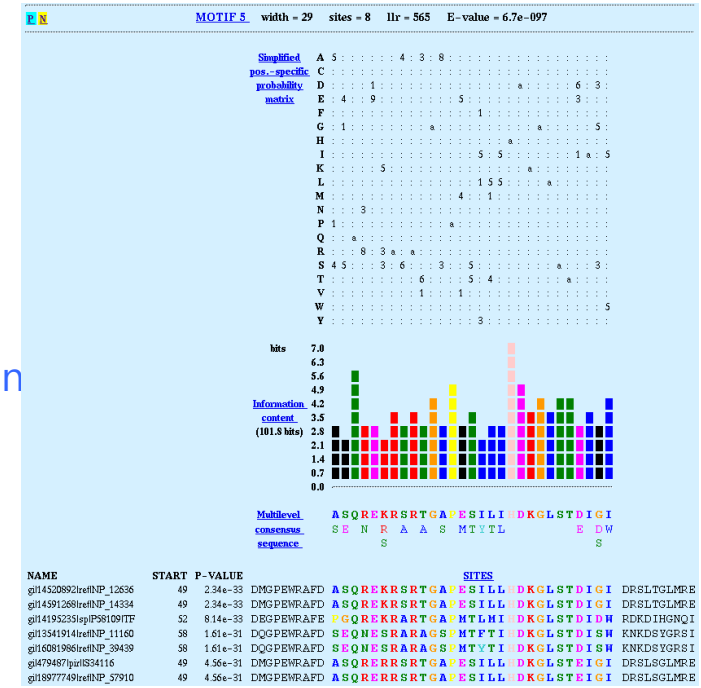
θ_{li} = probability of i in matrix position l
θ_{0i} = background frequency (in non-motif sequen

- Relative entropy is sometimes called information content

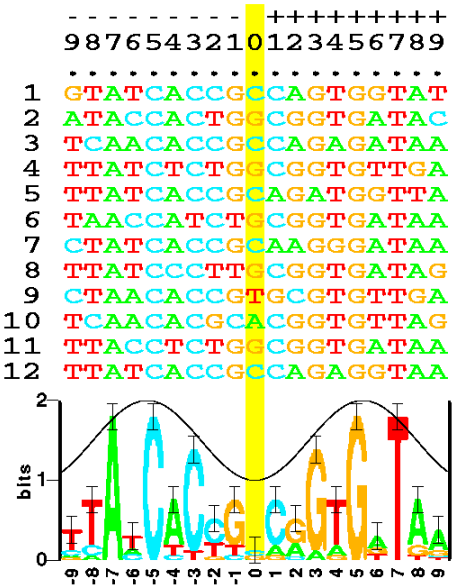
A	.04	.88	.26	.59	.49	.03
C	.09	.03	.11	.13	.21	.05
G	.07	.01	.12	.16	.12	.02
T	.80	.08	.51	.13	.18	.89

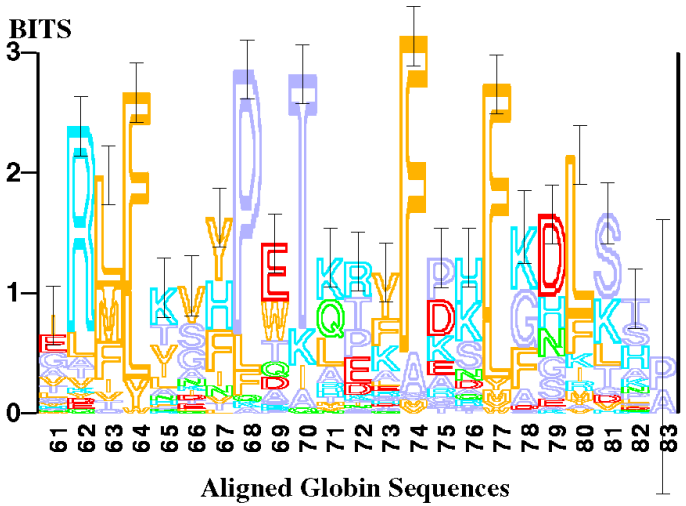


Informativeness: $2\text{-}\Sigma_i \theta_{li} \log_2 \theta_{li}/\theta_{0i}$



Sequence Logo





- Information at pos'n *l*, $H(l) = -\Sigma_{\{\text{letter i}\}}\ \theta_{li} \log_2 \theta_{li}$
- Height of x at pos'n *l*, $L(l,i) = \theta_{li}\,(2 - H(l))$
  - Examples:
    - $\theta_{lA} = 1$; H(l) = 0; L(l, A) = 2
    - A: ½; C: ¼; G: ¼; H(l) = 1.5; L(l, A) = ¼; L(l, not T) = ¼

de novo motif detection

- **Unsupervised learning**
  - Given no training examples, predict locations of all instances of ***novel*** motifs in given sequences, and learn motif models simultaneously.

- **Learning algorithms**:
  - Expectation Maximization: e.g., MEME
  - Gibbs Sampling: e.g., AlignACE, BioProspector
  - Advanced models: Bayesian network, Bayesian Markovian models

Expectation-maximization

EM

```
For each subsequence of width W
  convert subsequence to a matrix
  do {
    re-estimate motif occurrences from matrix
    re-estimate matrix model from motif occurrences
  } until (matrix model stops changing)
end
select matrix with highest score
```

Expectation-maximization

EM

```
For each subsequence of width W
  convert subsequence to a matrix
  do {
    re-estimate motif occurrences from matrix
    re-estimate matrix model from motif occurrences
  } until (matrix model stops changing)
end
select matrix with highest score
```

Sample DNA sequences

```
>ce1cg
TAATGTTTGTGCTGGTTTTTGTGGCATCGGGCGAGAATA
GCGCGTGGTGTGAAAGACTGTTTTTTTGATCGTTTTCAC
AAAAATGGAAGTCCACAGTCTTGACAG
>ara
GACAAAAACGCGTAACAAAAGTGTCTATAATCACGGCAG
AAAAGTCCACATTGATTATTTGCACGGCGTCACACTTTG
CTATGCCATAGCATTTTTATCCATAAG
>bglr1
ACAAATCCCAATAACTTAATTATTGGGATTTGTTATATA
TAACTTTATAAATTCCTAAAATTACACAAAGTTAATAAC
TGTGAGCATGGTCATATTTTTATCAAT
>crp
CACAAAGCGAAAGCTATGCTAAAACAGTCAGGATGCTAC
AGTAATACATTGATGTACTGCATGTATGCAAAGGACGTC
ACATTACCGTGCAGTACAGTTGATAGC
```

Motif occurrences

>ce1cg
taatgtttgtgctggtttttgtggcatcgggcgagaata
gcgcgtggtgtgaaagactgtttt**TTTGATCGTTTTCAC**
aaaaatggaagtccacagtcttgacag

>ara
gacaaaaacgcgtaacaaaagtgtctataatcacggcag
aaaagtccacattgatta**TTTGCACGGCGTCAC**actttg
ctatgccatagcatttttatccataag

>bglr1
acaaatcccaataacttaattattgggatttgttatata
taactttataaattcctaaaattacacaaagttaataac
TGTGAGCATGGTCATatttttatcaat

>crp
cacaaagcgaaagctatgctaaaacagtcaggatgctac
agtaatacattgatgtactgcatgta**TGCAAAGGACGTC**
ACattaccgtgcagtacagttgatagc

Motif occurrences

>ce1cg
taatgtttgtgctggtttttgtggcatcgggcgagaata
gcgcgtggtgtgaaagactgtttt**TTTGATCGTTTTCAC**
aaaaatggaagtccacagtcttgacag

>ara
gacaaaaacgcgtaacaaaagtgtctataatcacggcag
aaaagtccacattgatta**TTTGCACGGCGTCAC**actttg
ctatgccatagcatttttatccataag

>bglr1
acaaatcccaataacttaattattgggatttgttatata
taactttataaattcctaaaattacacaaagttaataac
TGTGAGCATGGTCATatttttatcaat

>crp
cacaaagcgaaagctatgctaaaacagtcaggatgctac
agtaatacattgatgtactgcatgta**TGCAAAGGACGTC**
ACattaccgtgcagtacagttgatagc

Starting point

…gactgttttt**TTTGATCGTTTTCAC**aaaaatgg…

	T	T	T	G	A	T	C	G	T	T
A	0.17	0.17	0.17	0.17	0.50	...				
C	0.17	0.17	0.17	0.17	0.17					
G	0.17	0.17	0.17	0.50	0.17					
T	0.50	0.50	0.50	0.17	0.17					

This a special initialization scheme, many others scheme, including random starts, are also valid

Re-estimating motif occurrences

TAATGTTTGTGCTGGTTTTTGTGGCATCGGGCGAGAATA

	T	T	T	G	A	T	C	G	T	T
A	0.17	**0.17**	**0.17**	0.17	0.50	...				
C	0.17	0.17	0.17	0.17	0.17					
G	0.17	0.17	0.17	0.50	**0.17**					
T	**0.50**	0.50	0.50	**0.17**	0.17					

Score = 0.50 + 0.17 + 0.17 + 0.17 + 0.17 + ...

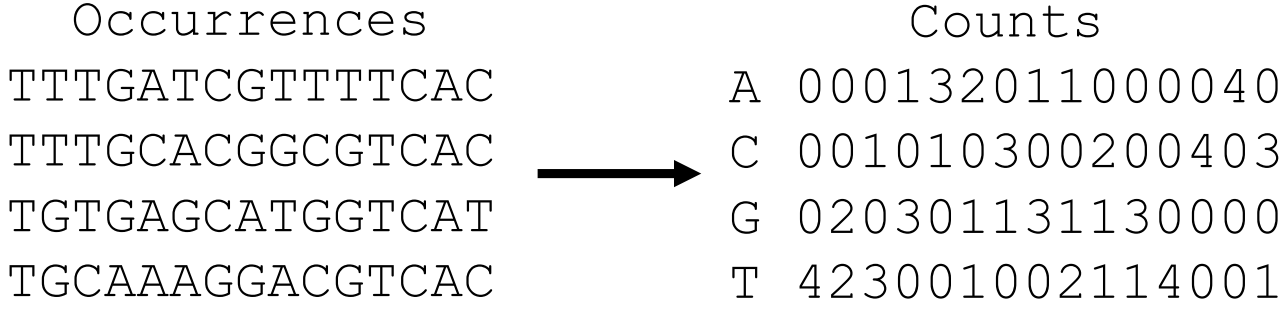
Scoring each subsequence

- Score from each sequence the subsequence with maximal sco re.

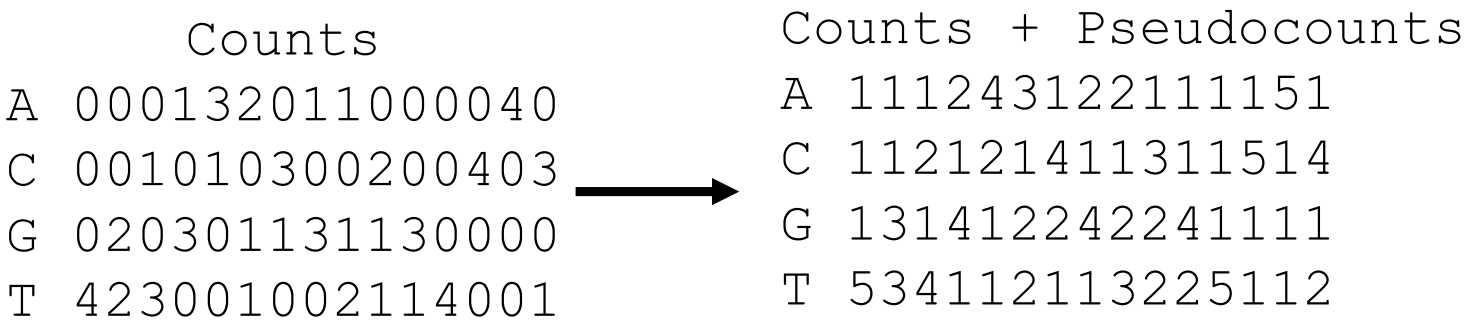
```
Sequence: TGTGCTGGTTTTTGTGGCATCGGGCGAGAATA


   Subsequences      Score
 TGTGCTGGTTTTTGT      2.95
  GTGCTGGTTTTTGTG     4.62
   TGCTGGTTTTTGTGG    2.31
    GCTGGTTTTTGTGGC     ...
```

Re-estimating motif matrix

- From each sequence, take the substring that has the maximal score
- Align all of them and count:

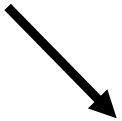


Adding pseudocounts



Converting to frequencies

```
Counts + Pseudocounts
A 111243122111151
C 112121411311514
G 131412242241111
T 534112113225112
```



```
     T    T    T    G    A    T  C  G  T  T
A 0.13 0.13 0.13 0.25 0.50 ...
C 0.13 0.13 0.25 0.13 0.25
G 0.13 0.38 0.13 0.50 0.13
T 0.63 0.38 0.50 0.13 0.13
```

Expectation-maximization

EM

```
For each subsequence of width W
   convert subsequence to a matrix
   do {
      re-estimate motif occurrences from matrix
      re-estimate matrix model from motif occurrences
   } until (matrix model stops changing)
end
select matrix with highest score
```

- **Problem:**
  - This procedure doesn't allow the motifs to move around very much. Taking the max is too brittle.
- **Solution:**
  - Associate with each start site a probability of motif occurrence.

Expectation Maximization: E-step

- **<u>Expectation</u>:**
  Find expected value of log likelihood:

$$\langle l_c(\Theta)\rangle = \sum_{n=1}^{N}\langle z_n\rangle\left(\left(\sum_{l=1}^{L}\sum_{j=1}^{4}\delta(y_{n+l-1},j)\log\theta_{l,j}\right)\right) + \sum_{n=1}^{N}(1-\langle z_n\rangle)\left(\left(\sum_{j=1}^{4}\delta(y_{n+l-1},j)\log\theta_{0,j}\right)\right)$$
$$+\sum_{n=1}^{N}\langle z_n\rangle\log\varepsilon+\left(N-\sum_{n=1}^{N}\langle z_n\rangle\right)\log(1-\varepsilon)$$

- where the expected value of Z can be computed as follows:

$$\langle z_i\rangle = p(z_i=1\mid Y) = \frac{\varepsilon p(y_i\mid\theta)}{\varepsilon p(y_i\mid\theta)+(1-\varepsilon)p(y_i\mid\theta_0)}$$

  - recall the weights for each substring in the MEME algorithm

Expectation Maximization: M-step

- **<u>Maximization</u>:**

  Maximize expected value over θ and ε independently

  For ε, this is easy:

$$\varepsilon^{NEW} = \arg\max_{\varepsilon} \sum_{n=1}^{N} \langle z_n \rangle \log \varepsilon + (N - \sum_{n=i}^{N} \langle z_i \rangle) \log(1 - \varepsilon) = \sum_{n=1}^{N} \frac{\langle z_n \rangle}{N}$$

Expectation Maximization: M-step

- For $\Theta = (\theta, \theta_0)$, define

  $c_{l,j}$ = E[# times letter *j* appears in motif position *l*]
  $c_{0,j}$ = E[# times letter *j* appears in background]

  - $c_{l,j}$ values are calculated easily from E[Z] values

  It easily follows:

$$\theta_{l,j}^{NEW} = \frac{c_{l,j}}{\sum_{j=1}^{4} c_{l,j}} \qquad \theta_{0,j}^{NEW} = \frac{c_{0,j}}{\sum_{j=1}^{4} c_{0,j}}$$

to not allow any 0's, add pseudocounts

Summary

EM

- Define **latent variables**
- Typically, estimation of latent variables is done by computing **posteriori probabilities**.
- The posteriori probabilities are used as “contributions” or “responsibilities” when estimating model parameters. Again, this is **a democratic process**!

EM

- Expectation of **what**?
- Maximization of **what**?
- At the end of each EM iteration, **what is computed**?

Latent Variables in Gaussian Mixture

- Latent variables = hidden data

1	3	4	5	6	8	11	13	14	15	17
P[G1]	P[G1]	P[G1]	P[G1]	P[G1]	P[G1]	P[G1]	P[G1]	P[G1]	P[G1]	P[G1]
P[G2]	P[G2]	P[G2]	P[G2]	P[G2]	P[G2]	P[G2]	P[G2]	P[G2]	P[G2]	P[G2]

STEP 1: Estimate P[1 from G1], P[3 from G1], …

STEP 2: μ_1 = (P[1 from G1] * 1 + P[3 from G1] * 3 + …) / (P[1 from G1] + P[3 from G1] + … + P[17 from G1])

Democracy again using latent variables:

Of course, we do not know which distribution the numbers are from.

1	3	4	5	6	8	11	13	14	15	17
P[G1]	P[G1]	P[G1]	P[G1]	P[G1]	P[G1]	P[G1]	P[G1]	P[G1]	P[G1]	P[G1]
P[G2]	P[G2]	P[G2]	P[G2]	P[G2]	P[G2]	P[G2]	P[G2]	P[G2]	P[G2]	P[G2]

STEP 1: Estimate P[1 from G1], P[3 from G1], …

Latent variables in HMM model parameter estimation

H	T	H	H	H	T	H
#F1	#F2	#F3	#F4	#F5	#F6	#F7
#L1	#L2	#L3	#L4	#L5	#L6	#L7

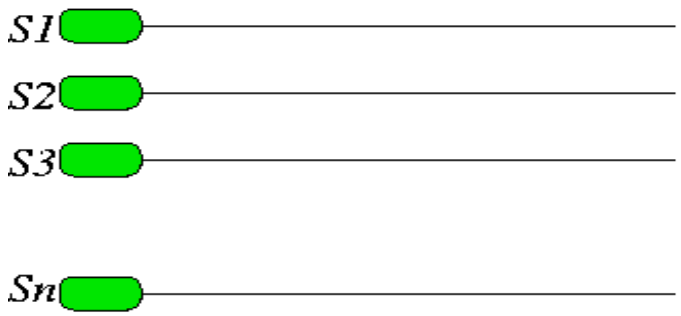
#[F1→F2] #[F2→F3] #[F3→F4] #[F4→F5] #[F5→F6] #[F6→F7]
#[F1→L2] #[F2→L3] #[F3→L4] #[F4→L5] #[F5→L6] #[F6→L7]
#[L1→F2] #[L2→F3] #[L3→F4] #[L4→F5] #[L5→F6] #[L6→F7]
#[L1→L2] #[L2→L3] #[L3→L4] #[L4→L5] #[L5→L6] #[L6→L7]

HTHHHTH를 이 HMM으로 생성하는 경우의 수는?

너무 많아서 계산이 불가능해지는데, 다행히 dynamic programming으로 쉽게 계산 가능.
→ Forward or Backward algorithm

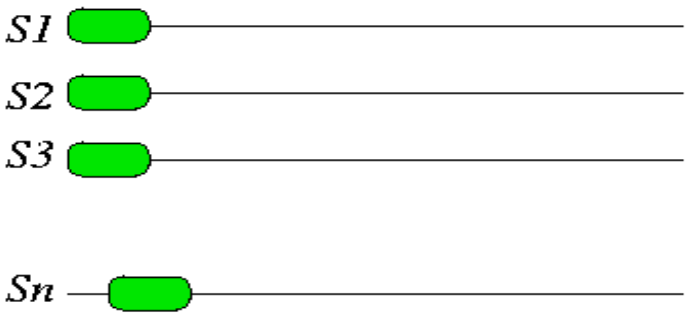
Latent variables in motif discovery

We are looking for a motif model with a set of fixed length sequences.



Motif model 1, M1

Score(M1) = H(M1 || R)



Motif model 2, M2

Score(M2) = H(M2 || R)

Solution: select a model Mi with the best solution.

Complications come from latent variables to model parameter estimation

- Gaussian mixture
  - Parametric distribution → small # of model parameters
- HMM model parameter estimation
  - Sequence data → emission & transition model parameters
  - Because of hidden states for sequence data, latent variable estimation is complicated.
- Motif discovery
  - Motif model is a matrix of numbers
  - Connecting latent variables to motif model needs a good thinking.
  - Otherwise, it is straightforward.
- **In the end, we use latent variables to estimate model parameters, which should be remembered clearly when you design an EM algorithm.**

감사합니다!