Divide & Conquer:
Dynamic Program
EM

Divide & conquer:
Solve a # of smaller subproblems
combine

Merge Set

Split data set in two
sort both recursively
merge results

M(n) = 2M(n/2) + Cn

152n needs
O(n) work per level
O(n log n)

Recurrence for computing
resources
Dynamic Progr.

- keep table of subproblems
- occurrence of current problem, i.e. previously solved problems

Pieces used can occur again.

\( L \subseteq x = a, b, c, x, \ldots \)

\( y = b, c, b, b, a, \ldots \)

\( y, y_m \)

\( T[i,j] \) longest common subsequence

\( w_i, y_1, \ldots, x_i \)

\( y_1, \ldots, y_j \)

\( \text{initialized} \)

\( \text{initialized} \)
\[ T(i, j) = T(i-1, j-1) + 1 \text{ if } i, j > 0 \land x_i = y_j \]

\[ = \max(T(i, j-1), T(i-1, j)) \text{ if } i, j > 0 \land x_i \neq y_j \]

\[ = 0 \text{ if } i = 0 \text{ or } j = 0 \]

- What's table - Recurrence
- Initialization
- Rec. & fill in
KNAPSACK

Items, $s_1, \ldots, s_n$

Does there exist a nonempty subset of size $k$ such that \sum_{i=1}^{k} s_i = \text{some number}$

Weakly NP-complete

$T(i, z) =$ true

if $\exists$ subset of $s_1, \ldots, s_i$ summing to $z$

$T(i, z) = \begin{cases} T(i-1, z) & \text{if } i = 0, z > 0 \\ T(i, z-s_i) \lor T(i-1, z) & \text{if } i > 0, z \geq 1 \end{cases}$

\[ z \]

\[ 0 \quad 1 \quad 2 \quad \ldots \quad k \]

$T$
EM algorithm (PART I)

- Expectation Maximization:
  iterative algorithm for maximizing likelihood

- Given: vectors of "visible" variables \( v_n \)
- Hidden: vectors of "hidden" variables \( h_n \)
  where \( n \) is example index
- Complete data set: \( V = \{ v_n \}, H = \{ h_n \} \)

- Model specifies a joint distribution:

\[
P(U, H | \Theta)
\]

\( \Theta \) parameters of model

- Usually i.i.d. data

\[
P(U, H | \Theta) = \prod_{n} P(v_n, h_n | \Theta)
\]

Then whole analysis "decomposes"!

- \( P(U | \Theta) = \sum_H P(U, H | \Theta) \)

\[
= \sum_H P(U | H, \Theta) P(H | \Theta)
\]

Goal:

- Maximize \( \ln P(U | \Theta) \)

Note: its log of a sum over the hidden variables
Example 1: Mixture of \( m \) fixed densities

\[ P(x|\theta) = \sum_i \theta_i P(x|i) \]

\[ \theta = \{\theta_1, \theta_2, \ldots, \theta_m\} \]

Mixtures coefficients

Statistics often use \( \theta \) for their model parameter.

Formally:

\[ P(x|\theta) = \sum_i P(x,i|\theta) \]

\[ = \sum_i P(x|i, \theta) P(i|\theta) \]

\[ \sum_i \text{fixed densities} \]

\( \theta \)

Example 2: Mixture of Gaussians

\[ P(x|\theta) = \sum_i \theta_i P(x|i, \theta) \]

\[ = \sum_i \theta_i \frac{1}{(2\pi)^{d/2} \Sigma_i^{1/2}} e^{-\frac{1}{2} (x-\mu_i)^T \Sigma_i^{-1} (x-\mu_i)} \]

\( \theta = \{\theta_1, \theta_2, \ldots, \theta_m\} \)

\( \mu = \{\mu_1, \mu_2, \ldots, \mu_k\} \)

\( \Sigma = \{\Sigma_1, \Sigma_2, \ldots, \Sigma_k\} \)

\( i \in \{1, 2, \ldots, \# \text{of Gaussians}\} \)

\( \Sigma_i \) is covariance matrix

\( \mu_i \) is mean

Point \( x \in \mathbb{R}^d \) visible

\( i \in \{1, 2, \ldots, \# \text{of Gaussians}\} \) hidden
Loss: \(- \ln P(V|\theta)\)

\[ = -\ln \sum_{H} P(V, H|\theta) \]

Hard to minimize because log of sum!

I.I.D. case: Decomposition into a sum

\[ V = \{v_n\}, \quad H = \{h_n\} \quad \text{SEQUENCES OF ONE FIXED LENGTH \((v_n, h_n)\)} \]

\noindent \text{ASSUMPTION:} \quad P(V, H|\theta) = \prod_{n=1}^{N} P(v_n, h_n|\theta)

\noindent \text{THIS IMPLIES:}

- \[ P(V|\theta) = \sum_{H} P(V, H|\theta) \]

\[ = \sum_{H} \prod_{n} P(v_n, h_n|\theta) \]

\[ = \sum_{h_1, h_2, \ldots, h_N} \prod_{n} P(v_n, h_n|\theta) \]

\[ = \prod_{n} \sum_{h_n} P(v_n, h_n|\theta) \]

\[ = \prod_{n} P(v_n|\theta) \]

- \[ P(H|V, \theta) = \frac{P(H, V|\theta)}{P(V|\theta)} = \prod_{n} \frac{P(h_n, v_n|\theta)}{P(v_n|\theta)} = \prod_{n} \frac{P(h_n|v_n, \theta)}{P(v_n|\theta)} \]

- \[ -\ln P(V|\theta) = -\sum_{n} \ln \frac{P(v_n|\theta)}{P(v_n|\theta)} \]

Product becomes sum

- Loss is essentially a relative entropy

\[ \sum_{n} \frac{1}{N} \ln \frac{N}{P(v_n|\theta)} = -\frac{1}{N} \sum_{n} \ln (v_n|\theta) - \ln N \]
Example 3: Hidden Markov Models

\[
P(abca, 1231 | \Theta) = 0.1 \cdot 0.1 \cdot 0.3 \cdot 0.6 \cdot 0.2 \cdot 0.1 \cdot 0.4 \cdot 0.1
\]

PARAMETERS: \( \Theta = \{ \Theta_i \} \)

\[
P(x, s | \Theta) = \prod_{i=1}^{m} \Theta_i \cdot n_i(x, s)
\]

where \( n_i \) is # of times \( \Theta_i \) occurs in \( (x, s) \)

\[
P(x | \Theta) = \sum_s P(x, s | \Theta)
\]

sum over all hidden paths
EM in KW framework: Θ new, Θ old

\[ \sum_H P(H|U, \Theta) \ln \frac{P(H|U, \Theta)}{P(H|U\Theta)} + \eta (-\ln P(U|\Theta)) \tag{*} \]

Divergence that motivates EM

Too hard to minimize!

\[ = \sum_H P(H|U, \Theta) \ln \frac{P(H|U, \Theta)}{P(H|U\Theta)} - \eta \ln P(U|\Theta) \]

\[ = \sum_H P(H|U, \Theta) \ln \frac{P(H|U, \Theta)}{P(H|U\Theta)} + \ln P(U|\Theta) - \ln P(U|\Theta) - \eta \ln P(U|\Theta) \]

(\eta = 1)

\[ = \sum_H P(H|U, \Theta) \ln \frac{P(H|U, \Theta)}{P(H|U\Theta)} - \ln P(U|\Theta) \]

\[ \text{constant} \]

Easier to minimize!

Joints frequently are products!

EM simplification requires \( \eta = 1 \)

We will see:

\( \eta = 1 \) too slow

\( \eta = 1 \) too large
\[ \sum_{H} P(H | V, \Theta) \ln \frac{P(H | V, \Theta)}{P(H | V, \Theta)} \]

\[ = \sum_{\{h_n\}} \prod_{n} P(h_n | v_n, \Theta) \ln \frac{\prod_{n} P(h_n | v_n, \Theta)}{\prod_{n} P(h_n | v_n, \Theta)} \]

\[ = \sum_{\{h_n\}} \sum_{n} P(h_n | v_n, \Theta) \ln \frac{P(h_n, v_n | \Theta)}{P(h_n, v_n | \Theta)} \]

\[ = \sum_{\{h_n\}} \sum_{n} P(h_n | v_n, \Theta) \ln \frac{P(h_n, v_n | \Theta)}{P(h_n, v_n | \Theta)} + \sum_{n} \ln P(v_n | \Theta) - \sum_{n} \ln P(v_n | \Theta) \]

When \( \eta = 1 \), (x) becomes

\[ \sum_{\{h_n\}} \sum_{n} P(h_n | v_n, \Theta) \ln \frac{P(h_n, v_n | \Theta)}{P(h_n, v_n | \Theta)} - \sum_{n} \ln P(v_n | \Theta) \]

\[ = \sum_{x, y} p(x) p(y) \ln \frac{p(x) p(y)}{q(x) q(y)} \]

\[ = \sum_{x, y} p(x) p(y) \left( \ln \frac{p(x)}{q(x)} + \ln \frac{p(y)}{q(y)} \right) \]

\[ = \sum_{x} p(x) \ln \frac{p(x)}{q(x)} + \sum_{y} p(y) \ln \frac{p(y)}{q(y)} \]
Minimize as a function of $\bar{\theta}$

$$\sum_H P(H \mid V, \theta) \ln \frac{P(H \mid V, \theta)}{P(H \mid V \bar{\theta})} - \ln P(V \mid \bar{\theta})$$

$$= \sum_H P(H \mid V, \theta) \ln \frac{P(H, V \mid \theta)}{P(H, V \mid \bar{\theta})} - \ln P(V \mid \bar{\theta})$$

$$= -\sum_H P(H \mid V, \theta) \ln P(H, V \mid \theta) + \sum_H P(H \mid V, \theta) \ln P(H, V \mid \bar{\theta})$$

piece that depends on $\bar{\theta}$

$$+$$

constant w/ $\bar{\theta}$

**Estimation Step:**

Compute posterior $P(H \mid V, \theta)$

**Maximization Step:**

$$\hat{\theta} := \arg \max_{\theta} \sum_H P(H \mid V, \theta) \ln P(H, V \mid \theta)$$

IRD CASE:

$$P(V, H \mid \theta) = \prod_{n=1}^{N} P(u_n \mid u_n, \theta)$$

($\star$) becomes:

$$-\sum_n \sum u_n P(h_n \mid u_n, \theta) \ln P(h_n \mid u_n, \theta)$$

$$+$$

constant

**E-step:**

Compute posteriors $P(h_n \mid u_n, \theta)$

**M-step:**

$$\hat{\theta} := \arg \max_{\theta} \sum_n \sum u_n P(h_n \mid u_n, \theta) \ln P(u_n, v_n \mid \theta)$$
Example 1:

Parameters $\Theta = \{\theta_1, \ldots, \theta_m\}$
visible $x$
hidden $i$
fixed densities: $P(x_i \mid \Theta) = P(x_i) = \sum_i P(x_i \mid \Theta)$

$$P(x \mid \Theta) = \sum_i P(x_i \mid \Theta) \prod_i P(x_i \mid \Theta)$$

$$= \sum_i P(x_i \mid \Theta) \prod_i P(x_i \mid \Theta)$$

$$= \sum_i \theta_i P(x_i \mid \Theta)$$

**E-step:**

$$P(i \mid x_n, \Theta) = \frac{P(i, x_n \mid \Theta)}{P(x_n \mid \Theta)} = \frac{\theta_i P(x_i \mid \Theta)}{\sum_j \theta_j P(x_j \mid \Theta)}$$

**M-step:**

Maximize

$$\sum_{n, x_n} \sum_i P(i \mid x_n, \Theta) \ln P(x_n \mid i \bar{\Theta}) + \lambda (\sum \bar{\theta}_i - 1)$$

$$= \sum_{n, x_n} \sum_i P(i \mid x_n, \Theta) \ln \bar{\theta}_i P(x_i \mid \Theta) + \lambda (\sum \bar{\theta}_i - 1)$$

$$\text{constant}$$

$$\frac{\partial}{\partial \bar{\theta}_j} = \sum_n \frac{P(i \mid x_n, \Theta)}{\bar{\theta}_j} + \lambda = 0$$

$$\Rightarrow \sum_n P(i \mid x_n, \Theta) + \lambda \bar{\theta}_j = 0 \quad (\ast)$$

$$\sum_{i, n} P(i \mid x_n, \Theta) + \lambda \sum_i \bar{\theta}_i = 0$$

$$\lambda = -N$$

From (\ast): \( \bar{\theta}_i = \frac{1}{N} \sum_n P(i \mid x_n, \Theta) \quad \text{Average Posterior} \)

$$N = 1 \quad \text{Bayes Rule}$$
Example 2: Mixture of Gaussians

\[ P(x \mid i, \theta) \text{ not independent of } \theta \]  

\[ P(x \mid \theta) = \sum_i P(i \mid \theta) P(x \mid i, \theta) \]

\[ = \sum_i \delta_i \cdot \frac{1}{(2\pi)^{1/2}} \frac{1}{\sqrt{\Sigma_i}} e^{-\frac{1}{2} (x - \mu_i)^T \Sigma_i^{-1} (x - \mu_i)} \]

E step:

\[ P(i \mid x_n, \theta) = \frac{P(i, x_n \mid \theta)}{P(x_n \mid \theta)} = \frac{\delta_i \cdot P(x \mid i, \theta)}{\sum_j \delta_j \cdot P(x \mid j, \theta)} \]

M step:

Maximize

\[ \sum_i \sum_j P(i \mid x_n, \theta) \ln P(i, x_n \mid \theta) \]

\[ = \sum_i \sum_j P(i \mid x_n, \theta) \ln \left( \frac{P(i \mid \theta) P(x_n \mid i, \theta)}{Z_i} \right) \]

3 Classes of parameters:

a) Mixture coefficients \( \delta_i \)

b) Means \( \mu_i \)

c) Covariance matrices \( \Sigma_i \)

Maximize above for one class at a time while keeping the other classes fixed.
a) $\mu_i$ and $\Sigma_i$ fixed

Thus $P(x_n | i, \bar{\theta}) = P(x_n | i, \theta)$

Thus becomes $\sum_m \sum_i P(i, l | x_n, \theta) \ln \bar{\delta}_i + \text{constant}$

Minimize as before with Lagrangian:

$$\delta_i = \frac{1}{N} \sum_m P(i | l, x_n, \theta)$$

b) and c): $\delta_i$ fixed

Thus $P(i | \theta) = P(i | \theta) = \delta_i$

Thus becomes $\sum_m \sum_i P(i | x_n, \theta) \ln P(x_n | i, \bar{\theta}) + \text{constant}$

$$\sum_m \sum_i P(i | x_n, \theta) \ln \left( \frac{1}{\sum \delta_i} e^{-\frac{1}{2} (x_n - \mu_i)^T \Sigma_i^{-1} (x_n - \mu_i)} \right) + \text{constant}$$

$$\sum_m \sum_i P(i | x_n, \theta) \left[ -\frac{1}{2} \ln \sum \delta_i - \frac{1}{2} (x_n - \bar{\mu}_i)^T \Sigma_i^{-1} (x_n - \bar{\mu}_i) \right] + \text{constant}$$

$$\Sigma_i = \Sigma_i$$
$$\bar{\mu}_i = \mu_i$$
\[ \frac{\partial}{\partial \mu_i} = \sum_{m} P(i|\mathbf{x}_m, \theta) \Sigma_i^{-1} (x_m - \tilde{\mu}_i) = 0 \]

\[ \Leftrightarrow \sum_{m} P(i|\mathbf{x}_m, \theta) (x_m - \tilde{\mu}_i) = 0 \]

\[ \tilde{\mu}_i = \frac{\sum_{m} P(i|\mathbf{x}_m, \theta) x_m}{\sum_{m} P(i|\mathbf{x}_m, \theta)} \]

\[ \frac{\partial}{\partial \Sigma_i} = \sum_{m} P(i|\mathbf{x}_m, \theta) \left( -\frac{1}{2} \tilde{\Sigma}_i^{-1} + \frac{1}{2} \Sigma_i^{-1} (x_m - \tilde{\mu}_i) (x_m - \tilde{\mu}_i)^T \Sigma_i^{-1} \right) = 0 \]

\[ \Leftrightarrow \sum_{m} P(i|\mathbf{x}_m, \theta) \left( \frac{1}{2} \Sigma_i^{-1} + \frac{1}{2} (x_m - \tilde{\mu}_i) (x_m - \tilde{\mu}_i)^T \right) = 0 \]

\[ \Sigma_i = \frac{\sum_{m} P(i|\mathbf{x}_m, \theta) (x_m - \tilde{\mu}_i) (x_m - \tilde{\mu}_i)^T}{\sum_{m} P(i|\mathbf{x}_m, \theta)} \]

**High-level intuition:**

In EM parameters are often updated to their usages when generating the data.
Intuition why EM works:
Minimizing $- \ln P(V|\theta) = - \ln \sum_H P(V,H|\theta)$
is hard because "\ln" of a sum.

Minimizing $\sum_H P(H|V,\theta) \ln P(H,V|\theta)$
"easy" when $P(H,V|\theta)$ is product!

EM avoids minimizing "\ln" of a sum directly.

By adding distance, minimization simplified.

By adding distance, minimization simplified.

$B): - \ln \sum_H P(V,H|\theta)$ has many symmetries.
For example by renaming any local minima
for a mixture of $m$ distinct Gaussians becomes
$m!$ local minima.

In $- \sum_H P(H,V,\theta) \ln P(H,V|\theta)$ not as many
symmetries. Hidden variables are "coupled"
with visible variables.
Example 3: HMM's

\[ P(x_1 s_1 \theta) = \prod_{i=1}^{T} \theta_i n_i(x_i s_i) \]

Maximize

\[ \sum_{\tilde{m}} \sum_{\tilde{s}_n} P(s_n | x_n, \theta) \ln P(x_n, s_n | \tilde{\theta}) \]

\[ = \sum_{\tilde{m}} \sum_{\tilde{s}_n} P(s_n | x_n, \theta) \sum_{i} \hat{n}_i(x_n, s_n) \ln \hat{\theta}_i \]

\[ = \sum_{\tilde{m}} \sum_{i} \ln \hat{\theta}_i \sum_{\tilde{s}_n} P(s_n | x_n, \theta) \hat{n}_i(x_n, s_n) \]

Expected usage of param. \( \theta_i \)

\[ \hat{n}_i(x | \theta) \]

Computable by dynamic program.

\[ [i] = \{ j : \theta_i \text{ and } \theta_j \text{ in some "class"} \} \]

All parameters of a class must sum to one.

- two classes associated with a state
- one class associated with the initial state probabilities

Maximize

\[ \sum_{\tilde{m}} \sum_{i} \left( \ln \hat{\theta}_i \right) \hat{n}_i(x_n, \theta) + \sum_{\tilde{m}} \lambda_{\tilde{c}ij} \left( \sum_{j} \theta_j - 1 \right) \]

\[ \frac{\partial}{\partial \hat{\theta}_i} = \sum_{\tilde{m}} \frac{\hat{n}_i(x_n, \theta)}{\hat{\theta}_i} + \lambda_{\tilde{c}ij} = 0 \]

\[ \hat{\theta}_i = \frac{\sum \hat{n}_i(x_n, \theta)}{\sum \hat{n}_i(x_n, \theta)} - \lambda_{\tilde{c}ij} \]

Enforcing constraint: \( \sum_{j \in [i]} \theta_j = 1 \)

\[ \overline{\theta}_i = \frac{\sum_{j \in [i]} \hat{n}_i(x_n, \theta)}{\sum_{j \in [i]} \sum_{\tilde{m}} \hat{n}_j(x_n, \theta)} \]
EM is too slow!

Synthetic Data 1: 2 State HMM

![Diagram of 2-state HMM with transition probabilities](image)

(0.63, 0.37)  (0.37, 0.63)

![Graph showing NLL vs iteration](image)

Entropic Update (eta=1.5)  
Baum-Welch (EM)

Later
Synthetic Data 1 (cont.)

Baum-Welch (EM)

Entropic Update (eta=1.5)
Framework for Parameter Update

- Add a penalty term to loss which will keep the new parameter set “close” to the old set.

\[
\Theta^{t+1} = \arg\min_{\Theta} U^t(\Theta) \quad (*)
\]

\[
U^t(\Theta) = \Delta(\Theta, \Theta^t) + \eta \text{ loss}(S|\Theta)
\]

\(\eta\) is a non-negative trade-off parameter that becomes the learning rate of the algorithm.

Any update of the form \((*)\) is called Implicit Update.
Minimal Properties of the Divergence

(i) \( \Delta(\Theta, \Theta) = 0 \)
(ii) \( \Delta(\bar{\Theta}, \Theta) > 0 \) whenever \( \bar{\Theta} \neq \Theta \).

Key Lemma

If \( U^t(\bar{\Theta}) < U^t_{\theta} \)
then \( \text{loss}(S|\bar{\Theta}) < \text{loss}(S|\theta) \).

Proof:

\[
U^t(\bar{\Theta}) = \Delta(\bar{\Theta}, \theta) + \eta \text{ loss}(S|\bar{\Theta})
\]
\[
< U^t_{\theta} = \Delta(\theta, \theta) + \eta \text{ loss}(S|\theta) \quad (i)
\]
This is equivalent to

\[
\text{loss}(S|\bar{\Theta}) = \text{loss}(S|\theta) - \frac{\Delta(\bar{\Theta}, \theta)}{\eta}
\]
\[
\Rightarrow \text{loss}(S|\bar{\Theta})
\]

For any implicit update s.t. \( \theta^{t+1} = \theta^t \),

\[
\text{loss}(S|\theta^{t+1}) < \text{loss}(S|\theta^t)
\]
What's good about EM:

- Implicit update. Thus negative log likelihood decreases in each iteration
- Simple and elegant

Bad: Converges too slowly

\[ \sum_{H} P(H | U, \Theta) \ln \frac{P(H | U, \Theta)}{P(H | V, \Theta)} - \eta \ln \frac{P(V | \Theta)}{P(V | \Theta)} \]

Simplification \[ \text{for } \eta = 1 \]

\[ - \sum_{H} P(H | U, \Theta) \ln P(H | V, \Theta) + \text{const.} \]

Want \( \eta > 1 \). In that case simplification does not work!!!

Idea 1: Use \( \eta > 1 \) and approximate \( -\ln (V | \Theta) \) by 1. order Taylor. Does not seem to work.

Idea 2: Use \( \eta > 1 \), different distance, and 1. order Taylor.

\[ \sum_{H} P(H | V, \Theta) \ln \frac{P(H | V, \Theta)}{P(H | V, \Theta)} \]

\[ - \eta \left( \ln P(V | \Theta) + (\Theta - \Theta) \frac{\partial \ln P(V | \Theta)}{\partial \Theta} \right) \]

Explicit!
Distance we use:
- different direction of entropy
- $V$ integrated over domain
- avoids "ln" of sum in different way.

In all three examples our method converges faster.

\[
\frac{\partial}{\partial \eta} \ln \mathcal{P}(\mathbf{V} | \mathbf{E}_\eta) \bigg|_{\eta=0} < 0 \quad \text{unless at extremum}
\]

provided $\eta$ is close enough to 0, then loss decreases.

We don't know why our method is so good.