**University of Illinois** Fall 2018

# ECE 544NA: Pattern Recognition
# Lecture 18: October 25

Lecturer: Alexander Schwing Scribe: Chenpeng Zhao

## Objective

The goal of unsupervised learning is to find some structure in X, that is to get value of P(X).In unsupervised learning, many interesting/useful models include"hidden variables", i.e.variables in the model that is not observed. We want to learn these variables with the model parameters in order to determine the distribution. A very broad family of algorithm is the Expectation Maximization family. We may have medthods for learning in a particular probabilistic model with hidden variables, but there are two questions: (1)can we apply this idea more generally and (2)why is it even a reasonable thing to do? EM is a family of algorithms of performing maximum likelihood estimation in probabilistic models with hidden variables.
Besides, we are able to know one method to get lower bound, which is Majorize-Minimize method. What's more, we are going to get to know about Concave-convex procedure(CCCP), which will get identical result as EM.

## Introduction of Expectation Maximization

Expectation Maximization is an iterative algorithm, where the model depends on unobserved latent variables. The EM iteration alternates between performing an expectation (E) step, which creats a function for the expectation of the log-likelihood evaluated using the current estimate for parameters, and Maximization step which computes parameters parameters maximizing the expected log-likelihood found on the E step.

## Recall Mixtures

In mixtures, we model distribution as follows: For every example we draw Z from a categorical distribution having parameter .After we have the latent variable representing a particular category, we draw X given Z from distribution $P(X|Z;\theta_x)$where $\theta_x$corresponds to the parameters that describe the underlying distribution

$$z_i \sim categorical(\pi) \tag{1}$$

$$x_i|z_i \sim P(X|Z;\theta_x) \tag{2}$$

$P(X|Z;\theta_x)$refers to some distribution. For example, Gaussian distribution for mixtures of gaussian, Multinomial distribution for describing text documents and topic models. The latent variable Z takes values from the set [1,2,....K]and the observed variables $X \in R^d$

When the give data is independent and identically distributed and we are using a mixture model, in that case, we can represent the Posterior distribution over Z in a product form:

$$P(Z_1, Z_2, ....Z_n|X_1, X_2, ......;\theta) = \prod_{i=1}^{n} P(Z_i = k|X_i) \tag{3}$$

we learn that the probability distribution function is P(Z),we can write the expressions:

$$P(Z|X) = \frac{P(X|Z)P(Z)}{P(X)} \tag{4}$$

where

$$P(Z) = P(Z_1, Z_2, ...., Z_n) = \prod_{i=1}^{n} P(Z_i) \tag{5}$$

Therefore, the expression for the posterior distribution can be written as follows:

$$P(Z_1, Z_2, ....Z_n|X_1, X_2, ......;\theta) = \frac{\prod_{i=1}^{n} P(X_i|Z_i, \theta) \prod_{i=1}^{n} P(Z_i)}{\sum_z \prod_{i=1}^{n} P(X_i|Z_i, \theta) \prod_{i=1}^{n} P(Z_i)} \tag{6}$$

**EM for Gaussian Mixture Model**

Gaussian mixture models are a probabilistic model for representing normally distributed subpopulation within an overall population. Since subpopulation assignment is not known, this constitutes a form of unsupervised learning. for example:

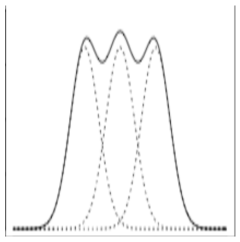

Figure 1: A Gaussian mixture of three normal distributions

After learning basic idea of mixture model, we can have a better understanding of following explanations.

Our model will start that we have K classes, and data from class k is drawn from a Gaussian with mean $\mu_k$and variance $\sum_k$, the choice of classes is parameterized by $\theta$weight is shown as $\pi_k$Therefore, we can choose initial parameters:

$$\theta = ((\pi_1, \mu_1, \sum_1), ...., (\pi_k, \mu_k, \sum_k)) \tag{7}$$

Alternate the following two steps util convergence:

**E**stimate the distribution of the hidden variable given the data and the current value of the parameters. Hold parameters fixed, optimally update soft assignments.$A\epsilon[0,1]^{n\times k}, A1_k = 1_n : \ for\ every\ i\epsilon \left\{1,.....n\right\},$

$$A_{ij} = p(y_i = j|x_i) \tag{8}$$

where $p_{\theta_j}$is the gaussian density with $\theta_j = (\mu_j, \sum_j)$,

$$((2\pi)^d det(\sum_j))^{-\frac{1}{2}} exp(-\frac{1}{2}(x-\mu_j)^T \sum_j^{-1}(x-\mu_j)) \tag{9}$$

**M**aximize the joint distribution of the data and the hidden variables. Hold assignment fixed, optimally update parameters: for every j$\epsilon \left\{1,.....k\right\}$

$$\begin{aligned} \pi_j &= \frac{\sum_i A_{ij}}{n} \\ \mu_j &= \frac{\sum_i A_{ij} x_i}{n\pi_j} \\ \sum_j &:= \frac{\sum_i A_{ij}(x_i-\mu_j)(x_i-\mu_j)^T}{n\pi_j} \end{aligned} \tag{10}$$

Note: GMMs often simplified via diagonal covariance matrices. Reasons: For a single gaussian, forcing axis to be independent variables, not axis aligned.

**Interpolating between k-means and GMMs**

Start with GMM but unweighted and with identity covariance.
Choose initial parameters $\theta = ((\frac{1}{k}, \mu_1, cl), ....., (\frac{1}{k}, \mu_k, cl))$Note: cl is scalar
Alternate the following two steps until convergence:

**(E step)** Hold parameters fixed, optimally update soft assignments.$A\epsilon[0,1]^{n\times k}, A1_k = 1_n$, for every i$\epsilon \left\{1,.....n\right\}$,

$$A_{ij} \propto \pi_j p_{\theta_j}(x_i) \tag{11}$$

where $p_{\theta_j}$is the gaussian density with $\theta_j = (\mu_j, \sum_j)$,

$$((2\pi)^d det(\sum_j))^{-\frac{1}{2}} exp(-\frac{1}{2}(x-\mu_j)^T \sum_j^{-1}(x-\mu_j)) \tag{12}$$

**(M step)** Hold assignments fixed, optimally update parameters: for every j$\epsilon \left\{1,.....k\right\}$,

$$\begin{aligned} \pi_j &= \frac{1}{k} \\ \mu_j &= \frac{\sum_i A_{ij} x_i}{\sum_i A_{ij}} \\ \sum_j &= cl \end{aligned} \tag{13}$$

consider the E-step, fix i$\epsilon \left\{1, .....k\right\}$ , define $r_{ij} = \frac{1}{2}\left\|x_i - \mu_j\right\|^2$

$$A_{ij} = \frac{\pi_j p_{\theta_j}(x_i)}{\sum_l \pi_l p_{\theta_l}(x_l)} = \frac{e^{-\frac{r_{ij}}{c}}}{\sum_l e^{\frac{-r_{il}}{c}}} = \frac{1}{1 + \sum_{l \neq j} e^{\frac{(r_{ij} - r_{il})}{c}}} \tag{14}$$

suppose $m_i = argmin_j r_{ij}$ unique. Then

$$lim \sum_{l \neq j} e^{\frac{r_{ij} - r_{il}}{c}} = \infty * 1[j \neq m] \tag{15}$$

That is,$A_{ij}$ becomes hard assignment as c close to 0 and decrease gradually.
In summary, k-means is obtained from E-M on GMMs via:uniform mixture weights; diagonal covariances cl with c close to zero and decrease gradually; zero-temperature hard version of the gaussian mixture model.

**Expectation Maximization introductions**

E-M framework: The goal in E-M for GMMs was to find parameters $\theta = (\theta_1, ......\theta_k)$to maximize the log-likelihood of the data:

$$l(\theta) = \sum_{i=1}^{n} lnp_\theta(x_j) = \sum_{i=1}^{n} ln(\sum_{j=1}^{k} \pi_j p\theta_j(x_i)) \tag{16}$$

We cannot solve this directly, because the summation inside, the ln is annoying (unless k=1...).We can run gradient ascent, but we will see E-M also increase $l$.

**Solutions:**

We want to maximize the log likelihood $l$,but this will turn out to be difficulty to do directly. Instead, we'll pick a surrogate function $\widetilde{l}(i.e., \forall \widetilde{l}, \widetilde{l} < l)$ that's lower bound on $l$ that's easier to maximize. We'll construct the surrogate in such a way that increasing it will force the true likelihood to also go up. After maximizing $\widetilde{l}$, we'll construct a new lower bound and optimize that.
Define latent or hidden variables($y_1, y_2, .....y_n$ identifying which Gaussian generated $x_i$ If we knew them, estimating parameters would be easy.
Considering an arbitrary probabilistic model $p(x, y|\theta)$,there x denotes the observed data, y denotes the hidden data and denotes that parameters. In the case of Gaussian Mixture Models, x was the data points, y was the unknown labels and $\theta$included the cluster prior probabilities, the cluster means and the cluster variances. Now, given access only to a number of examples ($x_1, x_2, .....x_N$,you would like to estimate the parameters ($\theta$) of the model.
Probabilistically, this means that some of the variables are unknown and therefore you need to marginalize (or sum)over their possible values. Now, your data consists only of

X=($x_1, x_2, .....x_N$ not the (x,y) pairs in D. You can then write the likelihood as:

$$\begin{aligned} p(X|\theta) &= \sum_{y_1}\sum_{y_2}....\sum_{y_N} p(X, y_1, y_2, ....y_N|\theta) \\ &= \sum_{y_1}\sum_{y_2}....\sum_{y_N}\prod_{n} p(x_n, y_n|\theta) \\ &= \prod_{n}\sum_{y_N} p(x_n, y_n|\theta) \end{aligned} \tag{17}$$

**E-M approach and theorem**

Based on what we discussed above, the natural thing to do is to take logs nad then start taking gradients. We define likelihood $l(\theta)$and a helper $\widetilde{l}(A,\theta)$:

$$\begin{aligned} l(\theta) &= \sum_{i=1}^{n} lnp_\theta(x_i) \\ \widetilde{l}(A,\theta) &= \sum_{i=1}^{n}\sum_{j=1}^{k} A_{ij} ln\frac{p_\theta(x_i,y_j)}{A_{ij}} \\ &= \sum_{i=1}^{n}\sum_{j=1}^{k} A_{ij}(lnp_\theta(x_i,y_j) - lnA_{ij}) \end{aligned} \tag{18}$$

The term $\sum_{ij} A_{ij} lnA_{ij}$does not affect argmax $\widetilde{l}(A,\theta)$
Moreover, given Lagrangian(with Lagrange multipliers $\alpha \epsilon R^n$)

$$\sum_{i=1}^{n} \alpha_i(\sum_j A_{ij} - 1) + \widetilde{l}(A,\theta) \tag{19}$$

applying $\frac{\partial}{\partial A_{ij}}$ and setting to zero gives

$$\alpha_i + lnp_\theta(x_i,y_j) - lnA_{ij} - 1 = 0 \Rightarrow A_{ij} = p_\theta(x_i,y_j)e^{\alpha_i - 1} \tag{20}$$

Thus, M and E steps maximize $\widetilde{l}(A,\theta)$, this is useful.
E step: $(A_\theta)_{ij} \propto p_\theta(x_i, y_j)$
M step: $\theta' = argmax\ \widetilde{l}(A,\theta)$
Then $\widetilde{l}(A,\theta) \leq l(\theta), l(\theta) = \widetilde{l}(A,\theta) \leq \widetilde{l}(A_\theta,\theta') \leq l(A_{\theta'},\theta') = \widetilde{l}(\theta')$
and in particular
$l(\theta_1) \leq l(\theta_2) \leq l(\theta_3)$.....

**Jensen's inequality**

We will now construct a lower bound using Jensen's inequality. This is a very useful result that states that $f(\sum_i \lambda_i x_i) \geq \sum_i \lambda_i x_i f(x_i)$,so long as $\lambda_i \geq 0$ for all i, $\sum_i \lambda_i = 1$ and f

is concave. If this looks familiar, that's just because it's a direct result of the definition of concavity. Recall that f is concave is $f(ax+by) \geq af(x)+bf(y)$ whenever a+b=1.
For convex function:

$$f(\sum_z q(z)g(z)) \leq \sum_z q(z)f(g(z)) \tag{21}$$

For concave function:

$$f(\sum_z q(z)g(z)) \geq \sum_z q(z)f(g(z)) \tag{22}$$

We can now apply Jensen's inequality to the log likelihood by identifying the list of $A_{ij}$ This fields:

$$\begin{aligned}\widetilde{l}(A,\theta) &= \sum_{i=1}^{n}\sum_{j=1}^{k} A_{ij} ln \frac{p_\theta(x_i,y_j)}{A_{ij}} \\ &\leq \sum_{i=1}^{n} ln(\sum_{j=1}^{k} A_{ij}\frac{p_\theta(x_i,y_j)}{A_{ij}}) \\ &= \sum_{i=1}^{n} ln(\sum_{j=1}^{k} p_\theta(x_i,y_j)) \\ &= l(\theta)\end{aligned} \tag{23}$$

On the other hand:

$$\begin{aligned}\widetilde{l}(A_\theta,\theta) &= \sum_{i=1}^{n}\sum_{j=1}^{k} \frac{p_\theta(x_i,y_j)}{p_\theta(x_i)} ln(p_\theta(x_i,y_j)(\frac{p_\theta(x_i)}{p_\theta(x_i,y_j)})) \\ &= \sum_{i=1}^{n}\sum_{j=1}^{k} \frac{p_\theta(x_i,y_j)}{p_\theta(x_i)} lnp_\theta(x_i) \\ &= \sum_{i=1}^{n}(\sum_{j=1}^{k} \frac{p_\theta(x_i,y_j)}{p_\theta(x_i)}) lnp_\theta(x_i) \\ &= l(\theta)\end{aligned} \tag{24}$$

Note: that this inequality holds for any choice of function A, so long as its non-negative and sums to one. In particular it needn't even by the same function A for each n. We will need to take advantage of both of these properties.
We have succeeded in our first goal:constructing lower bound. When you go to optimize this lower bound for $\theta$, the only part of matters is the first term. The second term, $A_{ij}lnA_{ij}$,drops out as a function of $\theta$. This means that the maximization need to compute, is:

$$\theta^{new} \Leftarrow \sum_{i=1}^{n}\sum_{j=1}^{k} A_{ij} lnp(x_i,y_i|\theta) \tag{25}$$

This is exactly the sort of maximization done for Gaussian mixture models when recomputing new means, variances and cluster prior probabilities.

Remark:
1.any reasonable $A_{ij}$ will lead to a lower bound, so in order to choose one A over another, we need another criterion. Recall that we are hoping to maximize $l$, by instead maximizing a lower bound. In order to ensure that an increase in the lower bound implies an increase in loss function, we should ensure that $\widetilde{l}(x,\theta) = l(x|\theta)$. In words: $\widetilde{l}$ should be a lower bound on $l$ that makes contact at the current point,$\theta$.
2.E-M method increase $\widetilde{l}$, $l$, but hard assignment increase $\widetilde{l}$ not clear for $l$, it also optimally sets $A_{ij}$, but maximize without $A_{ij}$.
3.E-M can be derived as alternating maximization of $\widetilde{l}$ and E-M can be shown to give non-decresing likelihood $l$.
4.Easy to run E-M on more complicated latent variable models. Latent variables are very useful.$l$ was not tractable, but $\widetilde{l}$ was tractable.

**Example**

An example of the algorithm in action is shown in figure.

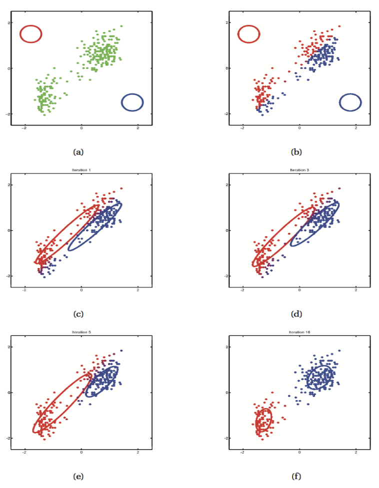

Figure 2: This is an E-M example

We start with $\mu_1 = (-1,1), \sum_1 = I, \mu_2 = (1,-1), \sum_2 = I$ ,we have learned that blue points come from cluster 1 and red points from cluster 2. More precisely, we set the color to:
$color(i) = A_{i1}blue + A_{i2}red$
so ambiguous points appear purple. After 20 iterations, the algorithm has converged on a good clustering. The data was standardized, by removing the mean and dividing by the standard deviation, before processing. This often helps convergence.

**General function:**
let's introduce distribution q(z), q(z) is any kind of distribution, and $\sum_z q(z) = 1$ and write

as follow:

$$\begin{aligned} lnp_\theta(x^i) = ln\sum p_\theta(x^i,z) = L(p_\theta(x^i,z),q(z)) + D_{kl}(q(z),p_\theta(z|x^i)) \\ L(p_\theta(x^i,z),q(z)) = \sum_z q(z)ln\frac{p_\theta(x^i,z)}{q(z)} \\ D_{kl}(q(z),p_\theta(z|x^i)) = \sum_z q(z)ln\frac{q(z)}{p_\theta z|x^i} \end{aligned} \tag{26}$$

Note:$L(p_\theta(x^i,z),q(z))$ is not equal to $D_{kl}(q(z),p_\theta(z|x^i))$, because the domain of both fomular is not equal. The domain of $p_\theta(x^i,z)$ is x and z.

Consequence for $D_{kl}$:

$$-D_{kl}(q(z),p_\theta(z|x^i)) = \sum_z q(z)ln\frac{p_\theta(z|x^i)}{q(z)} \leq 0 \tag{27}$$

Which means that $D_{kl}$ is non-negative.
Lower bound:

$$lnp_\theta(x^i) \geq L(p_\theta(x^i,z),q(z))$$

we may come up with new idea, maybe we can maximize lower bound rather than maximize the log-likelihood. thus, we can get the lower bound:

$$maxL(p_\theta(x^i,z),q(z))$$

How should we do to solve the problem. After comparsion, we can notice the difference between log-likelihood and lower bound is q(z), so we can alternately optimize q.
According to equation (24), we can know that if we want to get maximizing $L(p_\theta(x^i,z),q(z))$, we can set KL-divergence as zero. In this case, $lnp_\theta(x^i$ is upper bound and $lnp_\theta = L(p_\theta(x^i,z),q(z))$ if $q(z) = p_\theta(z|x^i)$ we can maximize the lower bound and get upper bound.

Alternative to show that: $q(z) = p_\theta(z|x^i)$

$$max\sum_z q(z)lnp_\theta(x^i,z) + H(q(z)) \quad s.t. \begin{Bmatrix} q(z) \geq 0 \\ \sum_z q(z) = 1 \end{Bmatrix} \tag{28}$$

solution:

$$q(z) = \frac{p_\theta(x^i,z)}{\sum_z p_\theta(x^i,z)} = p_\theta(z|x^i) = r_i \tag{29}$$

In the Gaussian case:

$$\begin{aligned} L(p_\theta(x^i,z),q(z)) &= \sum_z q(z) ln \frac{p_\theta(x^i,z)}{q(z)} \\ &= \sum_z q(z) ln \frac{\prod_{k=1}^{K} \pi_k^{z_{ik}} N(x^i|\mu_k,\sigma_k)^{z_{ik}}}{q(z)} \\ &= \sum_z q(z) ln \pi_k^{z_{ik}} N(x^i|\mu_k,\sigma_k)^{z_{ik}} + H(q(z)) \\ &= \sum_k r_{ik} ln \pi_k N(x^i|\mu_k,\sigma_k) - \sum_k r_{ik} ln r_{ik} \end{aligned} \tag{30}$$

Now we can infer general case of EM:

$$p_\theta(x^i,z) = \frac{1}{Z(\theta)} exp F(x^i,z,\theta) \tag{31}$$

$Z(\theta)$ is partition function

$$\begin{aligned} -L(p_\theta(x^i,z),q(z)) &= -\sum_z q(z) ln \frac{p_\theta(x^i,z)}{q(z)} \\ &= ln Z(\theta) - \sum_z q(z) F(x^i,z,\theta) - H(q(z)) \end{aligned} \tag{32}$$

**Concave-convex procedure (CCCP)**

Introductions: if we want to fit a model parametrized by $\theta$. Often we obtain the point estimiate $\theta$ by maximizing some energy function:

$$\theta = argmin E(\theta)$$

For convex energy function there exist unique optimum, which is easy to compute. However for most problem, E($\theta$) is non-convex, and it is difficult to develop fast optimization that is guaranteed to converged to a local optimum.
Therefore, people have tried to simplify the problem and proposed algorithms on it. A special case that has been discussed in the literature is the difference of convex functions programming (DC programming): the energy function is constrained to be the difference between two convex functions:

$$E(\theta) = E_{vex}(\theta) + E_{cave}(\theta)$$

where $E_{vex}(\theta) and - E_{cave}(\theta)$ denote two arbitrary convex functions. It is a weaker constraint than convexity as in original CCCP paper[Yuille and Rangarajan, 2013] the author proved the following algorithm.

As shown in figure 3, this procedure always decrease the enegy function: $\triangle = L_{vex}(\theta) - L_{cave}(\theta)$ denotes the distance along the function axis between the two lines that are tangent to the curve of $E_{cave}(E_{vex})$ at $\theta_t(\theta_{t+1})$ , respectively, and form convexity we have $E(\theta_t) \geq \triangle$ and $\triangle \geq E_(\theta_{t+1})$. Since the function in the form $E(\theta) = E_{vex}(\theta) + E_{cave}(\theta)$ are lower bound, the CCCP procedure is guaranteed to converge to a local optimum.

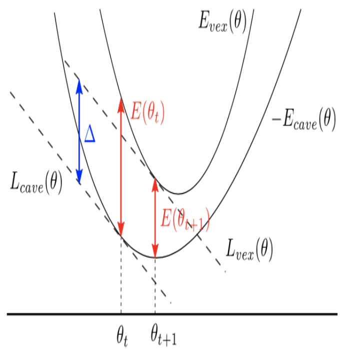

Figure 3: An illustration of the concave-convex procedure

## Combining CCCP with majorisation-minimisation

CCCP is a special case of majorisation-minimisation algorithm, which is a surrogate type optimisation method. Figure 4 illustrate the update procedure of the MM method, where the each iteration we first find a surrogate objective $E_t(\theta)$ that majorises the original objective at the current solution $\theta_t$ then apply any optimization algorithm to the surrogate for the next update $\theta_{t+1}$.

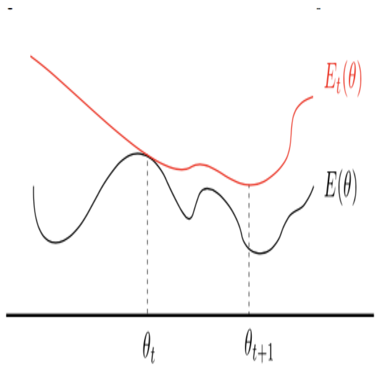

Figure 4: An illustration of the MM procedure

At each iteration we construct a surrogate objective $E_t(\theta) = E_{vex}(\theta_{t+1}) - L_{cave}(\theta)$Notice that $L_{cave}(\theta)$ depends on $E_{cave}(\theta) and \theta_t$, and more importantly the slope of this linear function is given by the negative gradient of the concave part $-\bigtriangledown E_{cave}(\theta_t)$. From convexity of $-\bigtriangledown E_{cave}(\theta)$ it is straightforward to see $E_t(\theta)$ is a convex function that majorises $E(\theta)$ at

$\theta_t$. Next we obtain the update by zeroing the gradient of $E_t(\theta)$ means:

$$\bigtriangledown E_t(\theta_{t+1}) = \bigtriangledown E_{vex}(\theta_{t+1}) + \bigtriangledown E_{cave}(\theta_t) = 0 \tag{33}$$

which has similar function model as MM step. To summarize CCCP is a MM algorithm that partially linearizes the objective function and zeros the gradient on the surrogate.
Apparently, there exist infinite number of functions that majorizes the objective at the given location, and it is still an open question that how to design the heuristic which returns the one that provides good next-step updates and is easy to compute.

Let's combine CCCP with general case:
$model : p_\theta(x^i, z) = \frac{1}{Z(\theta)} expF(x^i, z, \theta)$
maximum likelihood (marginalizing over latent space):

$$min - ln \sum_z \frac{expF(x^i, z, \theta)}{Z(\theta)} \tag{34}$$

$$min \qquad \underbrace{lnZ(\theta)}_{convex\ if\ F\ linear\ in\ \theta} \qquad \underbrace{- ln \sum_z expF(x^i, z, \theta)}_{convex\ if\ F\ linear\ in\ \theta} \tag{35}$$

We should initialize $\theta$
Repeats: decompose concave part into "convex+concave" at current $\theta$
Solve convex program
Solutions to decompose:

$$\begin{aligned} ln \sum_z expF(x^i, z, \theta) &= ln \sum_z q(z) \frac{expF(x^i, z, \theta)}{q(z)} \\ &= max(\sum_z q(z) F(x^i, z, \theta) + H(q(z))) \end{aligned} \tag{36}$$

According to above equation, it may easily remind us the inference programming.
Recall inference:

$$max \sum_{r, y_r} b_r(y_r) f_r(y_r) \qquad s.t. \qquad \begin{Bmatrix} b_r(y_r) \geq 0 \\ \sum_{y_r} b_r(y_r) = 1 \\ \sum_{\frac{y_p}{y_r}} b_p(y_p) = b_r y_r \end{Bmatrix}$$

Finally, we can get the results in:

$$min\ lnZ(\theta) - \sum_z q(z) F(x^i, z, \theta) - H(q(z)) \tag{37}$$

**Summary**

1.Generalizing EM
2.Getting to know its relationship with CCCP
3.Seeing the variational form of the partition function
4.Observing its similarity to inference

## References

[1] C. Bishop; Pattern Recognition and Machine Learning *Chapter 9.3, 9.4* .

[2] Yuille, A. L. and Rangarajan, A. (2003). The concave-convex procedure. *Neural Computation*, 15(4):915936.

[3] K. Murphy; Machine Learning: A Probabilistic Perspective.

[4] *http://yingzhenli.net/home/misc/notes/cccp$_m$m.pdf*

[5] A course in machine learning.
*http://ciml.info/*