Part 3: ML Parameter Estimation

Aim: The key question answered here is: *Given a partially observed stochastic dynamical system, how does one estimate the parameters of the system?*
Also joint recursive parameter and state estimation algorithms are described.

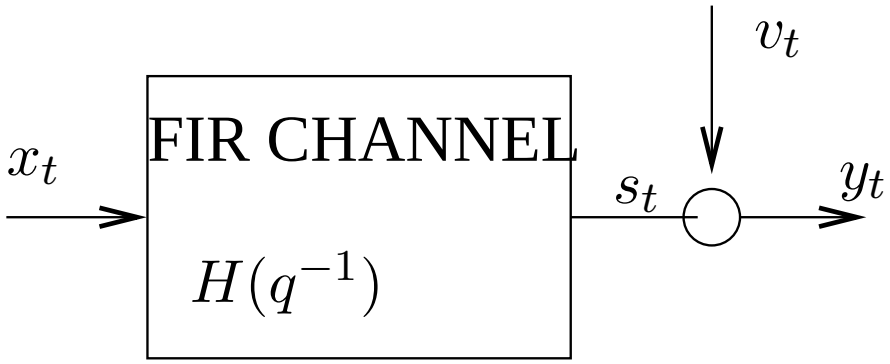
The algorithms make extensive use of filtering.

OUTLINE

- **ML Parameter Estimation**
  - ML criterion
  - 2 Simple Examples
  - EM Algorithm
  - Case Studies
    EM algorithm for Errors-in-Variables Estimation
    Baum Welch Algorithm for HMMs

- **Recursive Parameter Estimation**

Example: Blind Deconvolution



Assumptions: Assume unknown FIR channel

$$H(q^{-1}) = h_0 + h_1 q^1 + \cdots + h_L q^{-L}$$

Digital input x_t is assumed Markov (possibly unknown probabilities and state levels)

v_t is white Gaussian (possibly unknown variance).

EM algorithm for ML estimation can be used to compute:
(i) Parameters (channel, trans prob, noise var, levels)
(ii)and simultaneously provide optimal state estimate.

The EM algorithm is *off-line* and operates on a fixed batch of data. This motivates the need for a *recursive* or *online* joint parameter and state estimation algorithm – in particular the channel characteristics might change slowly with time – then a recursive estimator is necessary. Recursive EM algorithms can also be obtained.

ML Estimation

Given a sequence of measurements $Y_N \stackrel{\triangle}{=} (y_1, \ldots, y_N)$ likelihood function

$$L(\theta, N) \stackrel{\triangle}{=} p(Y_N; \theta), \qquad \theta \in \Theta$$

where Θ is the parameter space.

Likelihood function is a measure of the plausibility of the data under parameter θ. Our aim is to pick θ which makes data most plausible.

Aim: Compute maximum likelihood (ML) parameter estimate

$$\theta^{ML}(N) \stackrel{\triangle}{=} \arg\max_{\theta \in \Theta} L(\theta, N)$$

Often it is more convenient to maximize $\log L(\theta, N)$. Clearly $\arg\max_\theta L(\theta, N) = \arg\max_\theta \log L(\theta, N)$.

Why ML Estimation? MLE often has 2 nice properties

1. *Strong Consistency*: Let θ^* be true parameter. Then

$$\lim_{N \to \infty} \theta^{ML}(N) \to \theta^* \quad w.p.1$$

2. *Asymptotic Normality*: The MLE is normally distributed about the true parameter:

$$\sqrt{N}(\theta^{ML}(N) - \theta^*) \to N(0, I_{\theta^*}^{-1})$$

where I_{θ^*} is the Fisher Information Matrix.

Perspective – Point Estimation

Let θ^* denote the true model. MLE is an example of a "point" estimate. Given data vector Y, we seek an estimator $g(Y)$ of the true parameter $\theta^* \in \Theta$.

Unbiased: An estimator $g(Y)$ is unbiased if

$$\mathbf{E}_{Y|\theta^*}\{g(Y)\} = \theta^*, \qquad \text{for any } \theta^* \in \Theta$$

Cramer Rao Bound: Any unbiased estimator satisfies

$$\text{cov}(g) = \mathbf{E}_{Y|\theta^*}\{(g(Y) - \theta^*)(g(Y) - \theta^*)'\} \geq I_{\theta^*}^{-1}$$

Efficient Estimator: $g(Y)$ is an efficient estimator if it meets the CR bound.

Remark: Often the MLE is an *efficient* estimator.

We will discuss numerical algorithms for MLE estimation for two partially observed stochastic dynamical systems:
1. Linear Gaussian State Space Models.
2. Hidden Markov Models.

The consistency and asymptotic normality of the MLE for Linear Gaussian State Space Models is shown in several textbooks – e.g. Caines, Hannan & Deistler.

The consistency of the MLE for HMMs was proved in

1992 by Leroux. The proof is considerably more difficult. Asymptotic normality was only proved very recently.

2 Simple Examples

For partially observed models MLE needs to be numerically computed (as shown later). For fully observed models MLE can sometimes be analytically computed. Here are 2 examples.

1. **MLE for Gaussian Linear Model**: Suppose

$$Y = \Psi\theta + \epsilon, \qquad \epsilon \sim N(0_N, \Sigma_{N\times N})$$

Then likelihood function is

$$p(Y;\theta) = (2\pi)^{-N/2}|\Sigma|^{-1/2}\exp\left(-\frac{1}{2}(Y-\Psi\theta)'\Sigma^{-1}(Y-\Psi\theta)\right)$$

It is more convenient to maximize the log likelihood.

$$\log p(Y;\theta) = -\frac{N}{2}\log 2\pi - \frac{1}{2}\log|\Sigma| - \frac{1}{2}(Y-\Psi\theta)'\Sigma^{-1}(Y-\Psi\theta)$$

Setting $\frac{d}{d\theta}\log p(Y;\theta) = 0$ yields

$$\theta^{ML} = (\Psi'\Sigma^{-1}\Psi)^{-1}\Psi'\Sigma^{-1}Y$$

which coincides with least squares parameter estimate.

2. **MLE for Markov Chain**: Suppose $Y_N = (y_1, \ldots, y_N)$ is a S state chain. Parameter is

transition prob matrix $\theta = (a_{ij}, \quad i,j \in \{1,\ldots,S\})$

Note the parameter constraints:

$$\sum_{j=1}^{S} a_{ij} = 1, \qquad 0 \le a_{ij} \le 1$$

The likelihood and log likelihood functions are

$$p(Y_N;\theta) = p(y_n|y_{n-1};\theta)p(y_{n-1}|y_{n-2};\theta)\cdots p(y_1|y_0;\theta)p(y_0;\theta)$$

$$\log p(Y_N;\theta) = \sum_{k=1}^{N} \log p(y_k|y_{k-1};\theta) + \log p(y_0;\theta)$$

$$= \sum_{k=1}^{N}\sum_{i=1}^{S}\sum_{j=1}^{S} \delta(y_{k-1}=i, y_k=j)\log a_{ij} + \sum_{i=1}^{S}\delta(y_0 - i)\pi_0(i)$$

$$= \sum_{i=1}^{s}\sum_{j=1}^{S} J_{ij}(N)\log a_{ij} + \sum_{i=1}^{S}\delta(y_0 - i)\pi_0(i)$$

(note $f(y_k, y_{k-1}) = \sum_{i=1}^{s}\sum_{j=1}^{s} f(i,j) I(y_k = j, y_{k-1} = i)$);
$J_{ij} = \#$ jumps from state i to state j from time 1 to N.

Then $\frac{d}{da_j}\log p(Y_N;\theta) = 0$ subject to constraint yields

$$\boxed{a_{ij} = \frac{J_{ij}(N)}{\sum_{j=1}^{S} J_{ij}(N)} = \frac{J_{ij}(N)}{D_i(N)} = \frac{\#\text{jumps from } i \text{ to } j}{\#\text{of visits in } i}}$$

Numerical Algorithms for MLE

In HMM and Gaussian State space model, the MLE can be computed numerically. 2 algorithms are widely used:

1 Newton Raphson (NR) Algorithm for HMM

MLE: Given data $Y_T = (y_1, \ldots, y_T)$ and initial parameter estimate $\theta^{(0)} \in \Lambda$.

For iterations $I = 1, 2, \ldots,$, given model $\theta^{(I)}$ at iteration I:

- Compute $L(\theta)$, $\nabla_\theta L(\theta)$, $\nabla^2_\theta L(\theta)$ at $\theta = \theta^{(I)}$ recursively using optimal filter as follows
  (i) Compute HMM filter α^θ_k, $k = 1, \ldots, T$ as

$$\alpha_{k+1}(j) = P(x_{k+1} = q_j, Y_{k+1}) = \sum_{i=1}^{S} \alpha_k(i) a_{ij} b_j(y_{k+1})$$

  Likelihood $L(\theta) = P(Y_T|\theta) = \sum_{i=1}^S \alpha^\theta_T(i)$
  (ii) Compute derivative $\nabla_\theta L(\theta) = \sum_{i=1}^S R^\theta_T(i)$ where filter sensitivity $R^\theta_k(i) = \nabla_\theta \alpha^\theta_k(i)$, $k = 1, \ldots, T$ is

$$R^\theta_{k+1}(j) = \big(\nabla_\theta(b^\theta_j(y_{k+1})\big) \sum_{i=1}^{S} a^\theta_{ij} \alpha^\theta_k(i)$$

$$+ b^\theta_j(y_{k+1}) \sum_{i=1}^{S} \big(\nabla_\theta a^\theta_{ij}\big) \alpha^\theta_k(i) + b^\theta_j(y_{k+1}) \sum_{i=1}^{S} a^\theta_{ij} R^\theta_k(i)$$

- Update parameter estimate via Newton Raphson as:

$$\theta^{(I+1)} = \theta^{(I)} + \left[\nabla^2_\theta L(\theta)\right]^{-1} \nabla_\theta L(\theta)\Big|_{\theta=\theta^{(I)}}$$

Often Hessian $\nabla^2 L$ is approximated (Pseudo Newton).

2. Expectation Maximization (EM) Algorithm:

- Developed in 1976 by Dempster, Laird, Rubin.
- Widely used in last 15 years
- Recent variants based on MCMC yield Stochastic EM algorithms that are globally convergent. Algorithms such as Data augmentation can be viewed as stochastic versions of EM.
- EM can also be used for MAP state estimation in Generalized HMMs (Jump Markov Linear Systems)

Aside: Optimal Fixed Interval Smoother Given state space model:

$$x_{k+1} = f(x_k) + w_k$$
$$y_k = h(x_k) + v_k \quad Y_k = (y_1, \ldots, y_k)$$

Recall optimal smoother computes $\mathbf{E}\{x_k|Y_T\}$ for $T > k$.
There are 3 types of smoothing problems:
1. Fixed point smoother: For fixed k_0, compute $E\{x_{k_0}|Y_k\}$ for $k = k_0, k_0+1, \ldots$.

2. Fixed interval smoother: Compute $\mathbf{E}\{x_k|Y_T\}$ for $k = 1, \ldots, T$ (we will use this in the EM algorithm below).

3. Fixed lag smoother: For fixed lag $\Delta > 0$, compute $\mathbf{E}\{x_k|Y_{k+\Delta}\}$.

Derivation of Fixed Interval Smoother. Recall $\alpha_k(x) = p(x_k, Y_k)$, computed recursively as

$$\alpha_{k+1}(x) = p(y_{k+1}|x_{k+1}) \int \alpha_k(z) p(x_{k+1} = x|x_k = z) dz$$

Let $\beta_k(x) = p(Y_{k+1:T}|x_k)$ where $Y_{k+1:T} = (y_{k+1}, \ldots, y_T)$. Compute β via the backward recursion $k = T, T-1, \ldots, 0$.

$$\beta_k(z) = \int p(y_{k+1}|x_{k+1} = x) p(x_{k+1} = x|x_k = z) \beta_{k+1}(x) dx,$$

Initialized by $\beta_T(z) = 1 \quad \forall z$.

Then clearly fixed interval smoothed estimate is

$$p(x_k = x|Y_T) = \gamma_k(x) = \frac{\alpha_k(x)\beta_k(x)}{\int \alpha_k(x)\beta_k(x)dx}$$

$$\mathbf{E}\{x_k|Y_T\} = \int x \gamma_k(x) dx$$

Similarly $\gamma_k(x_k, x_{k+1}) = p(x_k, x_{k+1}|Y_T)$ is computed as:

$$\gamma_k(x_k, x_{k+1}) = \frac{\alpha_k(x_k) p(x_{k+1}|x_k) p(y_{k+1}|x_{k+1}) \beta_k(x_{k+1})}{\int \int [\text{Numerator}] dx_k dx_{k+1}}$$

Example: HMM Smoothing: For S state HMM θ

$$\alpha_{k+1}(j) = P(x_{k+1} = q_j | Y_{k+1}) = \sum_{i=1}^{S} \alpha_k(i) a_{ij} b_j(y_{k+1})$$

$$\beta_k(i) = p(Y_{k+1:T} | x_k = q_i) = \sum_{j=1}^{S} \beta_{k+1}(j) a_{ij} b_j(y_{k+1})$$

$$\gamma_k(i) = P(x_k = q_i | Y_T) = \frac{\alpha_k(i)\beta_k(i)}{\sum_{i=1}^{S} \alpha_k(i)\beta_k(i)}$$

$$\begin{aligned}\gamma_k(i,j) &= P(x_k = q_i, x_{k+1} = q_j | Y_T) \\ &= \frac{\alpha_k(i) a_{ij} b_j(y_{k+1}) \beta_{k+1}(j)}{\sum_{i=1}^{S}\sum_{j=1}^{S} \alpha_k(i) a_{ij} b_j(y_{k+1}) \beta_{k+1}(j)}\end{aligned}$$

Expected duration time in state i given data Y_T is

$$\mathbf{E}\{D_T(i) | Y_T\} = \sum_{k=1}^{T} \gamma_k(i)$$

Expected number of jumps from state i to state j

$$\mathbf{E}\{N_T(i,j) | Y_T\} = \sum_{k=1}^{T} \gamma_k(i,j)$$

Also note that $\gamma_k(i) = \sum_{j=1}^{S} \gamma_k(i,j)$. So

$$\sum_{j=1}^{S} \mathbf{E}\{N_T(i,j) | Y_T\} = \mathbf{E}\{D_T(i) | Y_T\}$$

Implementation: α_k and β_k are S dimensional vectors. Computational cost: $O(S^2T)$, memory cost $O(ST)$. Called forward backward algorithm in Rabiner's HMM paper. For Kalman case, β_k and γ_k are Gaussian. β_k computed by backward Kalman filter.

EM Algorithm for HMM maximum likelihood parameter estimation of transition probability: Consider HMM with unknown trans probability $\theta^* = A$. Choose initial $\theta^{(0)}$.

For iterations $I = 1, 2, \ldots$:

Step 1: Use model $\theta = \theta^{(I)}$ to compute $\alpha_k^\theta(i)$, $\beta_k^\theta(i)$, $\gamma_k^\theta(i)$, $\hat{D}_T^\theta(i) = \sum_{k=1}^T \gamma_k^\theta(i)$, $\hat{N}_T^\theta(i,j) = \sum_{k=1}^T \gamma_k^\theta(i,j)$. Called the Expectation (E) step.

Step 2: In analogy to MLE of Markov chain, compute new model $\theta^{(I+1)}$ as

$$a_{ij}^{(I+1)} = \frac{\hat{N}_T^\theta(i,j)}{\hat{D}_T^\theta(i)} = \frac{\mathbf{E}\{N_T(i,j)|Y_T,\theta\}}{\mathbf{E}\{D_T(i)|Y_T,\theta\}}, \quad \text{where } \theta = \theta^{(I)}$$

This can be interpreted as maximizing the complete data likelihood function (see later). Maximization (M) step. Go to Step 1.

Above update is guaranteed to generate valid transition probability estimates since $\sum_{j=1}^S \hat{N}_T(i,j) = \hat{D}_T(i)$. Unlike Newton Raphson, no constraints are required.

EM Algorithm

Consider partially observed stoch dynamical system

$$x_{k+1} = f(x_k; \theta) + w_k$$
$$y_k = h(x_k) + v_k$$

Let $X_T = (x_1, \ldots, x_T)$, $Y_T = (y_1, \ldots, y_T)$.
Aim: Given a sequence of observations Y_T compute MLE.

From an initial parameter estimate $\theta^{(0)}$, EM iteratively generates a sequence of estimates $\theta^{(I)}$, $I = 1, 2, \ldots$ as follows:
Each iteration consists of 2 steps:

- *E Step*: Evaluate auxiliary (complete) likelihood

$$Q(\theta^{(I)}, \theta) = E\{\ln p(X_T, Y_T; \theta) | Y_T, \theta^{(k)}\}$$

- *M step*: Maximize auxiliary (complete) likelihood, i.e, compute

$$\theta^{(I+1)} = \max_{\theta} Q(\theta^{(I)}, \theta)$$

Remark: Notice that the EM algorithm involves computing smoothed state densities – these involve forward and backward state filters. Thus optimal filtering is an important ingredient of the EM algorithm.

Advantages of EM Algorithm

- *Monotone property*: $L(\theta^{(I+1)}) \geq L(\theta^{(I)})$ (equality holds at a local maximum)
  NR does not have monotone property.
- In many cases, EM is much simpler to apply than NR. (e.g. HMMs, Error-in-variables models)
- EM is numerically more robust than NR; inverse of Hessian is not required in EM.
- Recent variants of the EM speed up convergence – SAGE, AECM, [MV97]

Dis-advantages of EM Algorithm

- Linear convergence: NR has quadratic convergence rate
- NR automatically yields estimates of parameter estimate variance. EM does not.

Example 1: EM algorithm for Linear Gaussian State Space Model Estimation

Consider scalar linear Gaussian state space model. (Easily generalized to multidimensional models.)

$$\text{State } x_t = a\, x_{t-1} + w_t$$

$$\text{Observations } y_t = x_t + v_t$$

$w_t \sim N(0, \sigma_w^2)$, $v_t \sim N(0, \sigma_v^2)$ white Gaussian processes.

Aim: Estimate $\theta = (a, \sigma_w^2, \sigma_v^2)$.

Applications: Speech coding, Econometrics [Gho89], Multisensor speech enhancement [WOFB94], errors in variables model [EK99].

EM Algorithm

E Step: The aim is to compute

$$Q(\theta^{(I)}, \theta) = E\{\ln p(Y_T, X_T|\theta)|Y_T, \theta^{(I)}\}$$

Result: The auxiliary likelihood $Q(\theta^{(I)}, \theta)$ is:

$$Q(\theta^{(I)}, \theta) = -\frac{T}{2}\ln\sigma_v^2 - \frac{1}{2\sigma_v^2}\sum_{t=1}^{T}\mathbf{E}\{(y_t - x_t)^2|Y_T, \theta^{(I)}\}$$
$$- \frac{T}{2}\ln\sigma_w^2 - \frac{1}{2\sigma_w^2}\sum_{t=1}^{T}\mathbf{E}\{(x_t - a\,x_{t-1})^2|Y_T, \theta^{(I)}\}$$

So we need to compute:
$\mathbf{E}\{x_t|Y_T, \theta\}$, $\mathbf{E}\{x_t\,x_{t-1}|Y_T, \theta\}$, $\mathbf{E}\{x_t^2|Y_T, \theta\}$, $\mathbf{E}\{x_{t-1}^2|Y_T, \theta\}$

These are obtained via a Kalman Smoother

M Step: Compute $\theta^{(k+1)} = \max_\theta Q(\theta^{(I)}, \theta)$

Setting $\partial Q/\partial\theta = 0$ yields:

$$a = \frac{\sum_{t=1}^{T}\mathbf{E}\{x_t\,x_{t-1}|Y_T, \theta^{(I)}\}}{\sum_{t=1}^{T}\mathbf{E}\{x_t^2|Y_T, \theta^{(k)}\}}$$

$$\sigma_v^2 = \frac{1}{T}\sum_{t=1}^{T}\Big(y_t^2 + E\{x_t^2|Y_T\} - 2\,E\{x_t\,y_t|Y_T, \theta^{(I)}\}\Big)$$

$$\sigma_w^2 = \frac{1}{T}\sum_{t=1}^{T}E\{(x_t - a\,x_{t-1})^2|Y_T, \theta^{(I)}\}$$

Set $\theta^{(I+1)} = (d, \sigma_v^2, \sigma_w^2)$

Remarks: (i) The update for a is similar to the Yule Walker equations (apart from conditioning on Y_T).
(ii) Estimates σ_v and σ_w are non-negative by construction.

Example 2: EM algorithm for HMM Estimation

Consider S state Markov chain $x_k \in q = \{q_1, \ldots, q_S\}$ with trans prob matrix $A. = (a_{ij})$, $i, j \in \{1, \ldots, S\}$.
Consider Markov chain x_k observed in Gaussian noise:

$$y_k = x_k + v_k, \qquad v_k \sim N(0, \sigma_v^2) \text{ white}$$

Aim: Estimate $\theta = (q, A, \sigma_v^2)$.

Application: Neurobiology, Channel Equalization, Target Tracking, Speech Recognition

EM Algorithm: (called Baum Welch algorithm)
E Step: Compute $Q(\theta^{(I)}, \theta) = E\{\ln p(Y_T, X_T|\theta)|Y_T, \theta^{(I)}\}$
Result: The auxiliary likelihood $Q(\theta^{(I)}, \theta)$ is:

$$Q(\theta^{(I)}, \theta) = -\frac{T}{2}\ln\sigma_v^2 - \frac{1}{2\sigma_v^2}\sum_{t=1}^{T}\sum_{i=1}^{S}\mathbf{E}\{(y_t - q_i)^2\}\gamma_t(i)$$
$$+\sum_{t=1}^{T}\sum_{i=1}^{S}\sum_{j=1}^{S}\gamma_t(i,j)\log a_{ij}$$

where $\gamma_t(i) = p(x_t = q_i|Y_T; \theta^{(I)})$,
$\gamma_t(i,j) = p(x_t = q_i, x_{t+1} = q_j|Y_T; \theta^{(I)})$ require a HMM state smoother (forward backward HMM filter).

M Step: Setting $\partial Q/\partial\theta = 0$ yields $\theta^{(I+1)}$:

$$a_{ij} = \frac{\sum_{t=1}^{T}\gamma_t(i,j)}{\sum_{t=1}^{t}\gamma_t(i)} = \frac{\mathbf{E}\{\#\text{jumps from } i \text{ to } j|Y_T,\theta^{(I)}\}}{\mathbf{E}\{\#\text{of visits in } i|Y_T,\theta^{(I)}\}}$$

$$q_i = \frac{\sum_{t=1}^{T}\gamma_t(i)y_t}{\sum_{t=1}^{T}\gamma_t(i)}$$

$$\sigma_v^2 = \frac{1}{T}\sum_{t=1}^{T}\sum_{i=1}^{S}\gamma_t(i)(y_t - q_i)^2$$

Remarks: 1. Nice property of EM is that estimates $0 \le a_{ij} < 1$, $\sum_j a_{ij} = 1$ is guaranteed by construction. Similarly, $\sigma_v^2 \ge 0$.

2. Can generalize the above to much more general HMMs – e.g. state dependent noise, Markov Modulated ARX time series.

3. The above EM is a smoother-based EM – the statistics are computed in terms of the smoothed density γ. In 1990s filter based EMs have been developed – e.g. Elliott, James Krishnamurthy and LeGland.

4. The EM algorithm can be formulated for continuous time HMMs.

Derivation of $Q(\theta^{(I)}, \theta)$ for HMM

$$\ln p(Y_T, X_T|\theta) = \ln \prod_{t=1}^{T} p(y_t|x_t)p(x_t|x_{t-1})$$

$$= \sum_{t=1}^{T} \ln p(y_t|x_t) + \sum_{t=1}^{T} \ln p(x_t|x_{t-1})$$

$$= \sum_{t=1}^{T}\sum_{i=1}^{S} \delta(x_t = i) \ln p(y_t|x_t = i)$$

$$+ \sum_{t=1}^{T}\sum_{i}\sum_{j} \delta(x_t = i, x_{t+1} = j) \ln P(x_{t+1}) = q_j|x_t = q_i)$$

$$= \sum_{i=1}^{S}\sum_{t=1}^{T} \delta(x_t = i) \left[\ln(\frac{1}{\sqrt{2\pi}\sigma_v}) - \frac{(y_t - q_i)^2}{2\sigma_v^2} \right]$$

$$+ \sum_{i}\sum_{j}\sum_{t=1}^{T} \delta(x_t = i, x_{t+1} = j) \ln a_{ij}$$

$$Q(\theta^{(I)}, \theta) = \mathbf{E}\{\ln p(Y_T, X_T|\theta)|Y_T, \theta^{(I)}\}$$

$$= \text{const} - \frac{T}{2} \ln \sigma_v^2 - \sum_{i}\sum_{t} \gamma_t^{\theta^{(I)}}(i) \frac{(y_t - q_i)^2}{2\sigma_v^2}$$

$$+ \sum_{i}\sum_{j}\sum_{t} \gamma_t^{\theta^{(I)}}(i,j) \ln a_{ij}$$

Proof of EM algorithm

Theorem: Given an observation sequence Y_T, and $Q(\theta^{(I)}, \theta) = \mathbf{E}\{\ln p(X_T, Y_T|\theta)|\theta^{(I)}, Y_T\}$. Then

$$\theta^{(I+1)} = \arg\max_{\theta} Q(\theta^{(I)}, \theta) \implies P(Y_T|\theta^{(I+1)}) \geq P(Y_T|\theta^{(I)})$$

To prove the theorem, first consider following lemma.

Lemma: For any θ, Q fn increases slower than log likelihood in terms of θ. That is:

$$Q(\theta^{(I)}, \theta) - Q(\theta^{(I)}, \theta^{(I)}) \leq \ln P(Y_T|\theta) - \ln P(Y_T|\theta^{(I)})$$

Then clearly choosing $\theta^{(I+1)}$ such that

$$Q(\theta^{(I)}, \theta^{(I+1)}) \geq Q(\theta^{(I)}, \theta^{(I)})$$

implies that $P(Y_T|\theta^{(I+1)}) \geq P(Y_T|\theta^{(I)})$. In particular, the best choice $\theta^{(I+1)} = \arg\max_\theta Q((\theta^{(I)}, \theta)$ implies $P(Y_T|\theta^{(I+1)}) \geq P(Y_T|\theta^{(I)})$.

Remark 1.: Just because likelihoods are monotone increasing does not mean EM converges. For convergence, require continuity of Q, compactness of $\theta \in \Theta$, etc, see (Wu, Annals of Statistics, 1983, pp.95–103). Wu uses Zangwill's global convergence theorem which is a standard tool in optimization theory to prove global convergence of an algorithm

Remark 2:

$$Q(\theta^{(I)},\theta) - Q(\theta^{(I)},\theta^{(I)}) = \mathbf{E}\{\ln \frac{P(Y_T, X_T|\theta)}{P(Y_T, X_T|\theta^{(I)})}|Y_T, \theta^{(I)}\}$$

is the Kullback-Liebler information measure widely used in information theory. Dempster, Laird and Rubin invented EM algorithm, 1977.

Proof of Lemma:

$$Q(\theta^{(I)},\theta) - Q(\theta^{(I)},\theta^{(I)}) = \mathbf{E}\{\ln \frac{P(Y_T, X_T|\theta)}{P(Y_T, X_T|\theta^{(I)})}|Y_T, \theta^{(I)}\}$$

$$\text{by Jensen's inequality } \leq \ln \mathbf{E}\{\frac{P(Y_T, X_T|\theta)}{P(Y_T, X_T|\theta^{(I)})}|Y_T, \theta^{(I)}\}$$

$$= \ln \int \frac{P(Y_T, X_T|\theta)}{P(Y_T, X_T|\theta^{(I)})} P(X_T|Y_T, \theta^{(I)}) dX_T$$

$$= \ln \int \frac{P(Y_T, X_T|\theta)}{P(X_T|Y_T, \theta^{(I)})P(Y_T|\theta^{(I)})} P(X_T|Y_T, \theta^{(I)}) dX_T$$

$$= \ln \int \frac{P(Y_T, X_T|\theta)}{P(Y_T|\theta^{(I)})} dX_T = \ln \frac{P(Y_T|\theta)}{P(Y_T|\theta^{(I)})}$$

Jensen's inequality:

$$f(X) \text{ convex} \implies \mathbf{E}\{f(X)\} \geq f(\mathbf{E}\{X\})$$

$$\text{Hence } f(X) \text{ concave} \implies \mathbf{E}\{f(X)\} \leq f(\mathbf{E}\{X\})$$

Consistency of MLE

Suppose $y_1, \ldots, y_T$ is an iid sequence of observations. $\theta^* \in \Theta$ true parameter. θ_T: MLE based on $y_1, \ldots, y_T$.
Aim: How to prove that $\lim_{T\to\infty} \theta_T \to \theta^*$ w.p.1. (Strong consistency of the MLE). We use Wald's approach.

Assume Θ is compact (i.e., closed bounded interval in $\mathbb{R}^S$). Then MLE

$$\theta_T = \arg\max_{\theta\in\Theta} \frac{1}{T} \log p(y_1, \ldots, y_T|\theta) = \arg\max_{\theta\in\Theta} \frac{1}{T} \sum_{k=1}^{T} \log p(y_k|\theta)$$

Assuming $\mathbf{E}_{\theta^*}\{|\log p(y_k|\theta)|\} < \infty$, then by SLLN,

$$\lim_{T\to\infty} \frac{1}{T} \sum_{k=1}^{T} \log p(y_k|\theta) = \underbrace{\mathbf{E}_{\theta^*}\{\log p(y_k|\theta)\}}_{\text{Kullback-Liebler } K(\theta,\theta^*)} \quad \text{w.p.1}$$

$$\text{So } \lim_{T\to\infty} \frac{1}{T} \log p(y_1, \ldots, y_T|\theta) \to K(\theta, \theta^*)$$

Lemma: From Jensen's inequality $\arg\max_\theta K(\theta, \theta^*) = \theta^*$

So we would intuitively expect

$$\arg\max_\theta \lim_{T\to\infty} \frac{1}{T} \log p(y_1, \ldots, y_T|\theta) \to \arg\max_\theta K(\theta, \theta^*) \text{ w.p.1}$$

– that is, $\theta_T \to \theta^*$ w.p.1 . More rigorously we need

$$\lim_{T\to\infty} \sup_{\theta\in\Theta} \frac{1}{T} \log p(y_1, \ldots, y_T|\theta) \overset{\text{w.p.1}}{\to} K(\theta, \theta^*) \text{ uniform convergence}$$

Recursive Parameter Estimation

Aim: Joint parameter and state estimation in real time.

Recall for HMM case $Q(\theta^{(I)}, \theta) = \sum_{t=1}^{T} \phi_t$:

$$Q(\theta^{(I)}, \theta) = -\frac{T}{2} \ln \sigma_v^2 - \frac{1}{2\sigma_v^2} \sum_{t=1}^{T} \sum_{i=1}^{S} \mathbf{E}\{(y_t - q_i)^2\} \gamma_k(i)$$

$$+ \sum_{t=1}^{T} \sum_{i=1}^{S} \sum_{j=1}^{S} \gamma_t(i,j) \log a_{ij}$$

$$= \sum_{t=1}^{T} \phi_t \qquad \text{(this is defn of } \phi_t\text{)}$$

Recursive Estimation Algorithm:

$$\theta_{t+1} = \theta_t + \gamma_t \frac{\partial \phi_t}{\partial \theta_t}$$

where γ_t is step size.

There are 3 popular choices of step size

1. Recursive EM Algorithm: $\gamma_t = (\frac{\partial^2 Q}{\partial \theta^2})^{-1}$

2. Gradient Algorithm: $\gamma_t = 1/t$.

3. Iterate Averaging (Polyak): $\gamma_t = 1/\sqrt{t}$ followed by averaging θ_t.

4. Tracking Algorithms: Constant step size $\gamma_t = \gamma$

Example: Markov Modulated Time Series

Markov Modulated AR process:

$$z_{k+1} = a(x_k)z_k + b(x_k)w_k$$

z_k: observations, x_k: S state unobserved Markov chain.
Arises in econometrics, LPC of speech, fault detection.

1. Similar algorithm to HMM filter yields $\mathbf{E}\{x_k|z_1, \ldots, z_k\}$. Also EM and recursive EM can be used for parameter estimation.

2. Image Enhanced Tracking:

$$z_{k+1} = a(x_k)z_k + b(x_k)w_k$$
$$y_k = x_k + v_k$$

x_k is mode of target – e.g. orientation.
y_k denotes noisy measurements of mode (e.g. imager).

Aim: Estimate coordinate of target z_k given $y_1, \ldots, y_k$.

Again a filter similar to HMM filter can be derived.
Parameters can be estimated via EM algorithm

3. Markov Modulated Poisson Process: Here N_t is a Poisson process whose rate $\lambda(x_k)$ is Markov modulated. A MMPP filter is similar to a HMM filter. Also EM can be used to compute parameters.