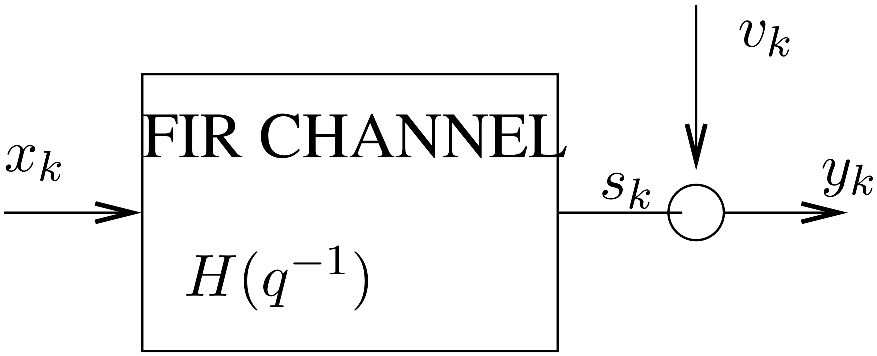
# Part IV: ML Parameter Estimation

*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.

*OUTLINE*

- **ML Parameter Estimation**
  - ML criterion
  - 2 Simple Examples
  - Gradient algorithms
  - EM Algorithm
  - Baum Welch Algorithm for HMMs

# Example: Blind Deconvolution




**Assumptions**:

$$y_k = x_k + h_1 x_{k-1} + \ldots + h_L x_{k-L} + v_k$$

Assume unknown FIR channel coefficients $h_1, \ldots, h_L$. Digital input $x_k$ is assumed Markov (possibly unknown probabilities and state levels)

$v_k$ is iid Gaussian (possibly unknown variance).

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

---

MLE computation is *off-line* and operates on a fixed batch of data.

# ML Estimation

Given a sequence of measurements $Y_N \stackrel{\text{defn}}{=} (y_1, \dots, y_N)$ likelihood function

$$L(\theta, N) \stackrel{\text{defn}}{=} p(Y_N|\theta), \qquad \theta \in \Theta$$

where $\Theta$ is the parameter space.

Likelihood function is a measure of the plausibility of the data under parameter $\theta$.

*Aim*: Compute ML parameter estimate

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

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 $\theta^*$ 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:

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

where $I_{\theta^*}$ is the Fisher Information Matrix.

# 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

$$Y = \Psi\theta + \epsilon, \qquad \epsilon \sim N(0_{N\times 1}, \Sigma_{N\times N})$$

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.

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

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

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

which coincides with least squares parameter estimate.

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

$$\text{transition prob matrix } \theta = (P_{ij}, \quad i, j \in \{1, \ldots, X\})$$

Note the parameter constraints:

$$\sum_{j=1}^{X} P_{ij} = 1, \qquad 0 \leq P_{ij} \leq 1$$

---

The likelihood and log likelihood functions are

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

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

$$= \sum_{k=1}^{N} \sum_{i=1}^{X} \sum_{j=1}^{X} I(y_{k-1} = i, y_k = j) \log P_{ij} + \sum_{i=1}^{X} I(y_0 = i)\pi_0(i)$$

$$= \sum_{i=1}^{X} \sum_{j=1}^{X} J_{ij}(N) \log P_{ij} + \sum_{i=1}^{X} I(y_0 = i)\pi_0(i)$$

$J_{ij}$ = # jumps from state $i$ to state $j$ from time 1 to $N$.

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

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

# Numerical Algorithms for MLE

**Aim**. Consider a HMM with state sequence $x_0, \ldots, x_N$.
Given observations $Y_N = y_1, \ldots, y_N$, compute MLE $\theta = (P, B)$.

Note likelihood for HMM is

$$\boxed{L(\theta) = p(Y_N|\theta) = \mathbf{1}' B_{y_N} P' B_{y_{N-1}} P' \cdots B_{y_1} P' \pi_0}$$

$L(\theta)$ can be computed numerically using un-normalized HMM filter. With $\alpha_k^\theta(i) = P(x_k = i, Y_k|\theta)$, HMM filter is

$$\alpha_{k+1}^\theta = B_{y_{k+1}}^\theta P^{\theta\prime} \alpha_k^\theta, \qquad \alpha_0^\theta = \pi_0$$

$$\boxed{\text{So likelihood is } L(\theta) = \mathbf{1}' \, \alpha_N^\theta = \sum_{i=1}^X P(x_N = i, Y_N|\theta)}$$

MLE can be computed numerically. 2 algorithms are widely used: (i) Newton Raphson (ii) Expectation Maximization

**Aside**: Consider unconstrained optimization: $\max_{\theta \in \mathbb{R}^d} F(\theta)$

1. **Steepest Ascent Gradient Algorithm** Scalar step size

$$\theta_{n+1} = \theta_n + \epsilon_n \nabla F(\theta_n), \qquad \epsilon_n \geq 0, \epsilon_n \to 0, \sum_n \epsilon_n = \infty$$

2. **Newton Raphson** Matrix step size (inverse of Hessian)

$$\theta_{n+1} = \theta_n + \left[\nabla^2 F(\theta_n)\right]^{-1} \nabla F(\theta_n)$$

Then $\{\theta_n\}$ converges to local stationary point.

**1 Newton Algorithm (General Purpose Optimization) for HMM MLE**: Given data $Y_N = (y_1, \dots, y_N)$ and initial parameter estimate $\theta^{(0)} \in \Theta$.

For iterations $I = 1, 2, \dots,$, 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) Run un-normalized HMM filter $\alpha^\theta_k$, $k = 1, \dots, N$

$$\alpha^\theta_{k+1}(j) = P(x_{k+1} = q_j, Y_{k+1}|\theta) = \sum_{i=1}^{X} \alpha^\theta_k(i) P_{ij} b_j(y_{k+1})$$

  Likelihood $L(\theta) = P(Y_N|\theta) = \sum_{i=1}^{X} \alpha^\theta_N(i)$

  (ii) Compute derivative $\nabla_\theta L(\theta) = \sum_{i=1}^{X} R^\theta_N(i)$ where filter sensitivity $R^\theta_k(i) = \nabla_\theta \alpha^\theta_k(i)$, $k = 1, \dots, N$ is

$$R^\theta_{k+1}(j) = \left(\nabla_\theta b^\theta_j(y_{k+1})\right) \sum_{i=1}^{X} P^\theta_{ij} \alpha^\theta_k(i)$$
$$+ b^\theta_j(y_{k+1}) \sum_{i=1}^{X} (\nabla_\theta P^\theta_{ij}) \alpha^\theta_k(i) + b^\theta_j(y_{k+1}) \sum_{i=1}^{X} P^\theta_{ij} R^\theta_k(i)$$

- Update parameter estimate via Newton Raphson as:

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

`fmincon` in Matlab is general purpose optimization algorithm.

**Scaling to avoid numerical underflow.**

Recall un-normalized HMM filter is

$$\alpha_k = [P(x_k = i, y_{1:k}), i = 1, \ldots, X] = B_{y_k} P' B_{y_{k-1}} P' \cdots B_{y_1} P' \pi_0$$

Recall normalized HMM filter is

$$\pi_k = [P(x_k = i | y_{1:k}), i = 1, \ldots, X] = \frac{B_{y_k} P' \pi_{k-1}}{\sigma_k}$$

where normalization term $\boxed{\sigma_k = \mathbf{1}' B_{y_k} P' \pi_{k-1}}$.
We can relate $\alpha_k$ and $\pi_k$ as follows:

$$\begin{aligned}
\pi_1 &= \frac{B_{y_1} P' \pi_0}{\sigma_1} = \frac{\alpha_1}{\sigma_1} \\
\pi_2 &= \frac{B_{y_2} P' \pi_1}{\sigma_2} = \frac{B_{y_2} P' B_{y_1} P' \pi_0}{\sigma_2 \sigma_1} = \frac{\alpha_2}{\sigma_2 \sigma_1} \\
\pi_k &= \frac{\alpha_k}{\prod_{t=1}^k \sigma_t}
\end{aligned}$$

So likelihood $L(\theta, N) = P(y_1, \ldots, y_N | \theta)$ can be computed as

$$L(\theta, N) = \mathbf{1}' \alpha_N^\theta = \mathbf{1}' \pi_N \prod_{t=1}^N \sigma_t = \prod_{t=1}^N \sigma_t.$$

$$\boxed{\log L(\theta, N) = \sum_{t=1}^N \log \sigma_t}$$

## 2. Expectation Maximization (EM) Algorithm:

- Developed in 1976 by Dempster, Laird, Rubin. Widely used in last 25 years
- Recent variants based on MCMC yield Stochastic EM algorithms that are globally convergent.

**Aside: Optimal Fixed Interval Smoother.** Consider HMM $\theta = (P, B)$ with unknown state sequence $(x_0, \ldots, x_N)$ and observation sequence $Y_N = (y_1, \ldots, y_N)$.
**Aim**. Fixed interval smoother: Compute $P(x_k | Y_N, \theta)$ for $k = 1, \ldots, N$ (we will use this in the EM algorithm below).

**HMM Smoothing**: For $X$ state HMM with model $\theta = (P, B)$

$$\boxed{\begin{aligned}
\alpha^\theta_{k+1}(j) &= P(x_{k+1} = q_j, Y_{k+1} | \theta) = \sum_{i=1}^{X} \alpha^\theta_k(i) P_{ij} b_j(y_{k+1}) \\
\beta^\theta_k(i) &= p(Y_{k+1:N} | x_k = q_i, \theta) = \sum_{j=1}^{X} \beta^\theta_{k+1}(j) P_{ij} b_j(y_{k+1}) \\
\gamma^\theta_k(i) &= P(x_k = q_i | Y_N, \theta) = \frac{\alpha^\theta_k(i)\beta^\theta_k(i)}{\sum_{i=1}^{X} \alpha^\theta_k(i)\beta^\theta_k(i)} \\
\gamma^\theta_k(i,j) &= P(x_k = q_i, x_{k+1} = q_j | Y_N, \theta) \\
&= \frac{\alpha^\theta_k(i) P_{ij} b_j(y_{k+1}) \beta^\theta_{k+1}(j)}{\sum_{i=1}^{X}\sum_{j=1}^{X} \alpha^\theta_k(i) P_{ij} b_j(y_{k+1}) \beta^\theta_{k+1}(j)}
\end{aligned}}$$

Expected duration time in state $i$ given data $Y_N$ is

$$\boxed{\mathbb{E}\{D_N^\theta(i)|Y_N\} = \sum_{k=1}^{N} \gamma_k^\theta(i)}$$

Expected number of jumps from state $i$ to state $j$

$$\boxed{\mathbb{E}\{J_N^\theta(i,j)|Y_N\} = \sum_{k=1}^{N} \gamma_k^\theta(i,j)}$$

Note $\gamma_k^\theta(i) = \sum_{j=1}^X \gamma_k^\theta(i,j)$. So $\sum_{j=1}^X \mathbb{E}\{J_N^\theta(i,j)|Y_N\} = \mathbb{E}\{D_N^\theta(i)|Y_N\}$

**Implementation**: For $X$-Markov chain given observations $Y_N = (y_1, \ldots, y_N)$, forward filter $\alpha_k^\theta$ and backward filter $\beta_k^\theta$ are $X$ dimensional vectors. Their computation is called *forward backward algorithm.*

1. Computational cost: $O(X^2 N)$,
2. Memory cost: $O(XN)$.

**Simple Example. EM Algorithm for HMM. MLE of transition probability $P^*$:**

Choose initial $\theta^{(0)} = P^{(0)}$. For iterations $I = 1, 2, \ldots$:

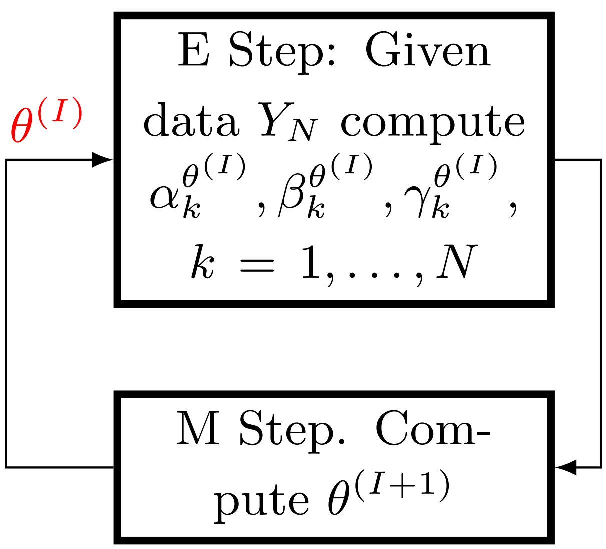
**Step 1** (E-step): Use model $\theta = \theta^{(I)}$ to compute $\alpha_k^\theta(i)$, $\beta_k^\theta(i)$, $\gamma_k^\theta(i)$, $k = 1, \ldots, N$.

Compute expected duration time $\hat{D}_N^\theta(i) = \sum_{k=1}^N \gamma_k^\theta(i)$, and expected number of jumps $\hat{J}_N^\theta(i,j) = \sum_{k=1}^N \gamma_k^\theta(i,j)$.

**Step 2**: (M-step) Compute new model $\theta^{(I+1)}$ as

$$P_{ij}^{(I+1)} = \frac{\hat{J}_N^\theta(i,j)}{\hat{D}_N^\theta(i)} = \frac{\mathbb{E}\{J_N^\theta(i,j)|Y_N\}}{\mathbb{E}\{D_N^\theta(i)|Y_N\}}, \qquad \text{where } \theta = \theta^{(I)}$$

Interpreted as maximizing complete data likelihood function. Go to Step 1.




1. Above update is guaranteed to generate valid transition probability estimates since $\sum_{j=1}^X \hat{J}_N^\theta(i,j) = \hat{D}_N^\theta(i)$.
2. Unlike Newton Raphson, no matrix inversion required.

# EM Algorithm (general formulation)

Consider partially observed stoch dynamical system

$$x_{k+1} = f(x_k;\theta) + w_k, \quad w_k \sim p_w^\theta$$
$$y_k = h(x_k;\theta) + v_k, \quad v_k \sim p_v^\theta$$

Let $X_N = (x_1,\dots,x_N)$, $Y_N = (y_1,\dots,y_N)$.
**Aim**: Given a sequence of observations $Y_N$ compute MLE

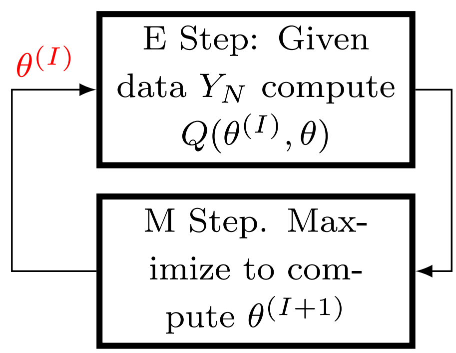
$$\theta^* = \operatorname*{argmax}_\theta L(\theta) = \operatorname*{argmax}_\theta p(Y_N|\theta)$$

---

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

Each iteration consists of 2 steps:

- *E Step*: Evaluate auxiliary (complete) likelihood
$$Q(\theta^{(I)},\theta) = \mathbb{E}\{\ln p(X_N, Y_N;\theta)|Y_N, \theta^{(I)}\}$$
- *M step*: Maximize auxiliary (complete) likelihood, i.e, compute
$$\theta^{(I+1)} = \max_\theta Q(\theta^{(I)},\theta)$$




*Remark*: EM algorithm involves computing smoothed state densities via forward and backward algorithm. Thus optimal filtering & smoothing are essential in 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, MCMC EM

**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 HMM Estimation (Baum-Welch Algorithm)

Consider $X$ state Markov chain $x_k \in q = \{q_1, \ldots, q_X\}$ with trans prob matrix $P = (P_{ij})$, $i, j \in \{1, \ldots, X\}$.
Assume Markov chain $x_k$ observed in Gaussian noise:

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

**Aim**: Estimate HMM parameters $\theta = (q, P, \sigma_v^2)$.

**Application**: Machine learning, Bioinformatics, Neurobiology, Channel Equalization, Target Tracking, Speech Recognition

**EM Algorithm for HMMs**: (called Baum Welch algorithm)
*E Step*: Compute $Q(\theta^{(I)}, \theta) = \mathbb{E}\{\ln p(Y_N, X_N|\theta)|Y_N, \theta^{(I)}\}$
*Result*: The auxiliary likelihood $Q(\theta^{(I)}, \theta)$ is:

$$\boxed{\begin{aligned} Q(\theta^{(I)}, \theta) &= -\frac{N}{2}\ln \sigma_v^2 - \frac{1}{2\sigma_v^2}\sum_{t=1}^{N}\sum_{i=1}^{X}(y_t - q_i)^2 \gamma_t^{\theta^{(I)}}(i) \\ &+ \sum_{t=1}^{N}\sum_{i=1}^{X}\sum_{j=1}^{X}\gamma_t^{\theta^{(I)}}(i,j)\log P_{ij} \end{aligned}}$$

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

*M Step*: Solving $\frac{\partial Q(\theta^{(I)},\theta)}{\partial\theta} = 0$ for $\theta^{(I+1)}$ yields $\theta^{(I+1)} = (P^{(I+1)}, q^{(I+1)}, \sigma^{2\,(I+1)})$ as:

$$P_{ij}^{(I+1)} = \frac{\sum_{t=1}^{N} \gamma_t^{\theta^{(I)}}(i,j)}{\sum_{t=1}^{t} \gamma_t^{\theta^{(I)}}(i)} = \frac{\mathbb{E}\{\#\text{jumps from } i \text{ to } j | Y_N, \theta^{(I)}\}}{\mathbb{E}\{\#\text{of visits in } i | Y_N, \theta^{(I)}\}}$$

$$q_i^{(I+1)} = \frac{\sum_{t=1}^{N} \gamma_t^{\theta^{(I)}}(i) y_t}{\sum_{t=1}^{N} \gamma_t^{\theta^{(I)}}(i)}$$

$$\sigma_v^{2\,(I+1)} = \frac{1}{N} \sum_{t=1}^{N} \sum_{i=1}^{X} \gamma_t^{\theta^{(I)}}(i)(y_t - q_i^{(I+1)})^2$$

---

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

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 $\gamma$. In 1990s filter based EMs have been developed.

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

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

$$\ln p(Y_N, X_N|\theta) = \ln \prod_{t=1}^{N} p(y_t|x_t) p(x_t|x_{t-1})$$

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

$$= \sum_{t=1}^{N} \sum_{i=1}^{X} I(x_t = i) \ln p(y_t|x_t = i)$$

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

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

$$+ \sum_{i} \sum_{j} \sum_{t=1}^{N} I(x_t = i, x_{t+1} = j) \ln P_{ij}$$

$$Q(\theta^{(I)}, \theta) = \mathbb{E}\{\ln p(Y_N, X_N|\theta) | Y_N, \theta^{(I)}\}$$

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

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

# Example 2: EM algorithm for Linear Gaussian State Space Model Estimation

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

$$\text{State } x_k = a\, x_{k-1} + w_k$$
$$\text{Observations } y_k = x_k + v_k$$

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

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

**Applications**: Speech coding, Econometrics, Multisensor speech enhancement

### EM Algorithm

*E Step*: The aim is to compute

$$Q(\theta^{(I)}, \theta) = \mathbb{E}\{\ln p(Y_N, X_N|\theta)|Y_N, \theta^{(I)}\}$$

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

$$\begin{aligned} Q(\theta^{(I)}, \theta) = &-\frac{N}{2}\ln \sigma_v^2 - \frac{1}{2\sigma_v^2}\sum_{t=1}^{N} \mathbb{E}\{(y_t - x_t)^2|Y_N, \theta^{(I)}\} \\ &- \frac{N}{2}\ln \sigma_w^2 - \frac{1}{2\sigma_w^2}\sum_{t=1}^{N} \mathbb{E}\{(x_t - a\, x_{t-1})^2|Y_N, \theta^{(I)}\} \end{aligned}$$

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

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:

$$
\begin{aligned}
a &= \frac{\sum_{t=1}^N \mathbb{E}\{x_t\,x_{t-1}|Y_N,\theta^{(I)}\}}{\sum_{t=1}^N \mathbb{E}\{x_t^2|Y_N,\theta^{(k)}\}} \\
\sigma_v^2 &= \frac{1}{N}\sum_{t=1}^N \Big(y_t^2 + \mathbb{E}\{x_t^2|Y_N\} - 2\,\mathbb{E}\{x_t\,y_t|Y_N,\theta^{(I)}\}\Big) \\
\sigma_w^2 &= \frac{1}{N}\sum_{t=1}^N \mathbb{E}\{(x_t - a\,x_{t-1})^2|Y_N,\theta^{(I)}\}
\end{aligned}
$$

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 $Y_N$).
(ii) Estimates $\sigma_v$ and $\sigma_w$ are non-negative by construction.

# Models similar to HMMs

**1.** Markov Modulated AR process:

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

$z_k$: observations, $x_k$: $X$ state unobserved Markov chain. Arises in econometrics, fault detection.

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

**2.** 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.

**3.** Empirical Bayes: The **empirical Bayes** model is of the form

$$\begin{aligned} X|\Theta &\sim p(x|\theta) \\ Y|X &\sim p(y|x) \end{aligned} \tag{11}$$

There is no explicit density for the hyperparameter $\theta$. Instead MLE $\theta^* = \text{argmax}_\theta \, p(y|\theta)$ is computed. Note

$$p(y|\theta) = \int_X p(y|x) \, p(x|\theta) \, dx$$

Estimate $\theta^*$ is plugged into Bayes rule to evaluate the posterior $p(x|y, \theta^*)$. The formulation is similar to a HMM

# Proof of EM algorithm

**Theorem**: Given an observation sequence $Y_N$, and $Q(\theta^{(I)}, \theta) = \mathbb{E}\{\ln p(X_N, Y_N|\theta)|\theta^{(I)}, Y_N\}$. Then computing

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

To prove the theorem, first consider following lemma.

---

**Lemma**: For any $\theta$, $Q$ fn increases slower than log likelihood in terms of $\theta$. That is:

$$Q(\theta^{(I)}, \theta) - Q(\theta^{(I)}, \theta^{(I)}) \leq \ln P(Y_N|\theta) - \ln P(Y_N|\theta^{(I)}) \quad \text{(A)}$$

---

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

$$Q(\theta^{(I)}, \theta^{(I+1)}) \geq Q(\theta^{(I)}, \theta^{(I)}) \implies P(Y_N|\theta^{(I+1)}) \geq P(Y_N|\theta^{(I)}) \quad \text{(B)}$$

Clearly the choice $\theta^{(I+1)} = \arg\max_\theta Q(\theta^{(I)}, \theta)$ guarantees (B) and therefore $P(Y_N|\theta^{(I+1)}) \geq P(Y_N|\theta^{(I)})$.

---

*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*: Kullback-Liebler information interpretation.

$$Q(\theta^{(I)}, \theta) - Q(\theta^{(I)}, \theta^{(I)}) = \mathbb{E}\{\ln \frac{P(Y_N, X_N|\theta)}{P(Y_N, X_N|\theta^{(I)})} |Y_N, \theta^{(I)}\}$$

is the Kullback-Liebler information measure widely used in information theory.

**Proof of Lemma**:

$$Q(\theta^{(I)}, \theta) - Q(\theta^{(I)}, \theta^{(I)}) = \mathbb{E}\{\ln \frac{P(Y_N, X_N|\theta)}{P(Y_N, X_N|\theta^{(I)})} |Y_N, \theta^{(I)}\}$$

$$\text{by Jensen's inequality } \leq \ln \mathbb{E}\{\frac{P(Y_N, X_N|\theta)}{P(Y_N, X_N|\theta^{(I)})} |Y_N, \theta^{(I)}\}$$

$$= \ln \int \frac{P(Y_N, X_N|\theta)}{P(Y_N, X_N|\theta^{(I)})} P(X_N|Y_N, \theta^{(I)}) dX_N$$

$$= \ln \int \frac{P(Y_N, X_N|\theta)}{\cancel{P(X_N|Y_N, \theta^{(I)})} P(Y_N|\theta^{(I)})} \cancel{P(X_N|Y_N, \theta^{(I)})} dX_N$$

$$= \ln \int \frac{P(Y_N, X_N|\theta)}{P(Y_N|\theta^{(I)})} dX_N = \ln \frac{P(Y_N|\theta)}{P(Y_N|\theta^{(I)})}$$

---

**Jensen's inequality**:

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

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

---

1. Dempster, Laird and Rubin invented EM algorithm, 1977.
2. EM is a special case of Minorization Maximization algorithms (Hunter & Lange, American Statistician, 2004).
3. EM can be implemented without smoother by forward-only filter-based E-step

# Consistency of MLE (advanced)

Suppose $y_1, \ldots, y_N$ is an iid sequence of observations. $\theta^* \in \Theta$ true parameter. MLE $\theta_N$ is based on $y_1, \ldots, y_N$.
**Aim**: Prove that $\lim_{N\to\infty} \theta_N \to \theta^*$ w.p.1. (Strong consistency of the MLE). Modern approach described below is due to Wald.

Assume $\Theta$ is compact (i.e., closed bounded interval in $\mathbb{R}^X$).

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

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

$$\lim_{N\to\infty} \frac{1}{N} \sum_{k=1}^{N} \log p(y_k|\theta) = \underbrace{\mathbb{E}_{\theta^*}\{\log p(y_k|\theta)\}}_{K(\theta,\theta^*)=-D_{KL}(\theta^*,\theta)} \quad \text{w.p.1}$$

$$\text{So} \lim_{N\to\infty} \frac{1}{N} \log p(y_1, \ldots, y_N|\theta) \to K(\theta, \theta^*) \quad \text{w.p.1}$$

**Lemma**: Jensen's inequality implies $\arg\max_\theta K(\theta, \theta^*) = \theta^*$.
Equivalently, $\arg\max_\theta K(\theta, \theta^*) = \operatorname{argmin}_\theta D_{KL}(\theta^*, \theta)$.

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

– i.e., $\theta_N \to \theta^*$ w.p.1 . More rigorously, require uniform SLLN

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

Sufficient condition is stochastic equicontinuity of $\{l_n(\theta)\}$:

$$P(\sup_{|\theta-\bar{\theta}|\leq\delta} |l_n(w,\theta) - l_n(w,\bar{\theta})| \leq \epsilon) = 1, \quad n > N(w)$$