Part 3: Optimal Filtering

Aim: The key question answered here is:
Given a stochastic signal observed in noise, how does one construct an optimal estimator of the signal?
The key results will be covered using elementary concepts in probability and stochastic processes. Optimal Filters are used in telecommunication systems, radar tracking systems, speech processing.

- Review of key tools in probability and stochastic processes: Bayes' rule, Conditional Expectation
- Summarize 4 basic stochastic processes: white noise, Markov processes, Hidden Markov Models (HMMs) and state space models.
- Develop the key results in discrete time filtering – including Kalman filter and HMM filter.
- Briefly describe sequential MCMC based particle filters.

Note: A more rigorous development in continuous time involves martingale theory, Girsanov's theorem, etc. This is not covered here

Optimal filtering in Signal Processing

Wiener Filter: Nobert Wiener (MIT) 1940s:
Model $Y = S + W$, S is signal W is noise.

$$\min_F \mathbb{E}\|S - FY\|^2$$

Widely used in LMMSE detection.

Kalman Filter: (1960s) Model S and N in time domain (state space models). The Kalman filter is probably the single most used algorithm in signal processing.

Hidden Markov Filter: Developed by statisticians (L. Baum, T. Petrie) in 1960s
Significant application in Electrical Engg in 1990s in speech recognition, channel equalization, tracking, etc

Sequential Markov Chain Monte Carlo Methods: Particle filters – randomized (simulation based) algorithms – applications in target tracking – late 1990s.

Stochastic Filtering theory studies optimal filtering. Also called recursive Bayesian estimation.
Journals: IEEE Trans Signal Processing; Automatic Control, Information Theory; Aerospace.

In continuous-time stochastic filtering theory involves stochastic calculus – widely used in mathematical finance.

Perspective

Given a partially observed stochastic dynamical system

$$x_{k+1} = A_k(x_k) + \Gamma_k(x_k)w_k, \quad x_0 \sim \pi_0(\cdot)$$
$$y_k = C_k(x_k) + D_k(x_k)v_k,$$

or equivalently in transition density form

$$p(x_{k+1}|x_k) = p_w\left(\Gamma_k^{-1}(x_k)\,[x_{k+1} - A_k(x_k)]\right)$$
$$p(y_k|x_k) = p_v\left(D_k^{-1}(x_k)\,[y_k - C_k(x_k)]\right).$$

Assume known model: The aim is estimate state given observations $y_{1:k} = (y_1, \ldots, y_k)$.

State estimation has two broad philosophies

- **Bayesian State Estimation:** Model based **optimal filtering** such as Kalman Filters, Hidden Markov Model filters, particle filters. This is what we will cover.
- **Adaptive filtering:** Stochastic Approximation e.g. LMS, RLS. x_k is assumed to vary very slowly with unknown dynamics. We will briefly touch on this in recursive parameter estimation.

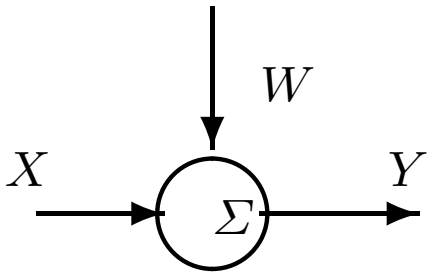
1 Results in Stochastic Processes

Likelihood of Observation:

Suppose

$$Y = X + W$$

where Y noisy measurement, X signal, W noise.
Assume noise W is independent of signal X.



Then conditional density of Y given X is given by "*Likelihood formula*"

$$\boxed{p_{Y|X}(y|x) = p_W(y - x)}$$

Remarks:

1. $p_{Y|X}(y|x)$ is called *observation probability* or *observation likelihood*. It denotes the likelihood of that observation Y came from signal X.
2. Likelihood formula says that observation probability

depends on noise density p_W.

3. Likelihood formula is of fundamental importance in communication systems, signal processing.

Example: For additive Gaussian channel $Y = X + W$ where $W \sim N(0, \sigma^2)$. Hence from likelihood formula

$$p_{Y|X}(y|x) = p_W(y - x) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left[-\frac{1}{2}\frac{(y-x)^2}{\sigma^2}\right]$$

---

4. Suppose $Y = f(X, W)$ and W can be expressed as $W = g(X, Y)$. Then likelihood formula is

$$p_{Y|X}(y|X) = p_W(g(X, Y))$$

One problem with the above likelihood probabilities is that they make no assumption on the prior information of X.

How can we use apriori information on $p_X(x)$ together with likelihood probabilities to compute a better estimate?

Bayes' rule gives the answer.

1.1 Bayes Rule

(The most important result in statistical inference).

Allows to reverse conditioning

$$p_{X|Y}(x|y) = \frac{p_{Y|X}(y|x)\, p_X(x)}{p_Y(y)}$$
$$= \frac{p_{Y|X}(y|x) p_X(x)}{\int_{\mathbb{R}} p_{Y|X}(y|\zeta) p_X(\zeta)\, d\zeta}$$

This is the single most important result which is at the heart of all statistical inference and filtering.

In words: Suppose we know

- probability of receiving observation y given message x was transmitted: $p_{Y|X}(y|x)$ (observation probability)
- the probability that the source transmitted message x: $p_X(x)$ (called *a priori* probability)

Suppose we received an observation y.
Then Bayes' rule tells us how to compute the conditional probability that the message sent was x given that the received message is y: $p_{X|Y}(x|y)$ (called *a posteriori* probability)

1.2 Conditional Expectation

Essential tool in filtering, estimation and control.

(i) Definition: Math rigorous defn: (i) $E\{X|\mathcal{F}\}$ is measurable wrt $\mathcal{F}$.

$$(ii)\ \mathbb{E}\{I_A \mathbb{E}\{X|\mathcal{F}\}\} = \mathbb{E}\{I_A X\}, \quad \forall A \in \mathcal{F}$$

Engg defn:

$$\mathbb{E}\{X|Y=y\} = \int_{\mathbb{R}} x p_{X|Y}(x|y) dx$$

is a function of y.
More generally

$$\mathbb{E}\{g(X,Y)|Y=y\} = \int_{\mathbb{R}} g(x,y) p_{X|Y}(x|y) dx$$

(ii) Smoothing Property: If $\mathcal{F}_1$, $\mathcal{F}_2$ are two sigma algebras with $\mathcal{F}_1 \subset \mathcal{F}_2$, then

$$\mathbb{E}\{\mathbb{E}\{X|\mathcal{F}_2\}|\mathcal{F}_1\} = \mathbb{E}\{X|\mathcal{F}_1\}$$

Example: $\mathcal{F}_1 = (\Omega, \emptyset\}$, then
$\mathbb{E}\{\mathbb{E}\{X|\mathcal{F}_2\}|\mathcal{F}_1\} = \mathbb{E}\{X|\mathcal{F}_1\} = \mathbb{E}\{X\}$ (unconditional expectation).

(iii) Optimality: $\mathbb{E}\{X|Y\}$ is optimal in the following min-variance sense: Suppose X, Y are rvs.

Find the function $g(Y)$ which minimizes

$$\mathbb{E}\{(g(Y) - X)^2\}$$

Soln: $g(Y) = E\{X|Y\}$

This is the basis of optimal state estimation via filtering. Suppose X is the state observed via noisy observations Y. Then the optimal (minimum variance) state estimate of X given Y is $\mathbb{E}\{X|Y\}$.

More generally: *Bregman loss function* Suppose ϕ convex:

$$\mathbf{L}_\phi(x, \bar{x}) = \phi(x) - \phi(\bar{x}) - (x - \bar{x})' \nabla\phi(\bar{x}).$$

$$\boxed{\kappa^*(\mathbf{y}) = \mathbb{E}\{x|\mathbf{y}\} = \operatorname*{argmin}_{\kappa \in \boldsymbol{\kappa}} \mathbb{E}\{\mathbf{L}_\phi(x, \kappa(\mathbf{y}))\}.}$$

Convexity implies
$\mathbf{L}_\phi(x, \bar{x}) \geq 0$, and $\mathbf{L}_\phi(x, \bar{x}) = 0$ iff $x = \bar{x}$.
Ex1: *Quadratic loss*: $\phi(x) = x'Sx$.

$$\mathbf{L}_\phi(x, \bar{x}) = x'Sx - \bar{x}S\bar{x} - 2(x - \bar{x})'\bar{x} = \|x - \bar{x}\|^2$$

Ex 2. *Kullback Liebler divergence*: $x = (x_1, \ldots, x_X)$ with $\sum_{i=1}^X x_i = 1$. Define the negative Shannon entropy as $\phi(x) = \sum_{i=1}^X x_i \log x_i$.

$$\begin{aligned}\mathbf{L}_\phi(x, \bar{x}) &= \sum_{i=1}^X x_i \log_2 x_i - \sum_{i=1}^X \bar{x}_i \log_2 \bar{x}_i - (x - \bar{x})' \nabla\phi(\bar{x}) \\ &= \sum_{i=1}^X x_i \log_2 x_i - \sum_{i=1}^X \bar{x}_i \log_2 \bar{x}_i\end{aligned}$$

Proof: Denote $\hat{x} = \mathbb{E}\{x|\mathbf{y}\}$. We will show:

$$\mathbb{E}\{\mathbf{L}_\phi(x,\kappa)\} - \mathbb{E}\{\mathbf{L}_\phi(x,\hat{x})\} = \mathbb{E}\{\mathbf{L}_\phi(\hat{x},\kappa)\}.$$

The result then follows since $\mathbb{E}\{\mathbf{L}_\phi(\hat{x},\kappa)\}$ is minimized if $\kappa(\mathbf{y}) = \hat{x} = \mathbb{E}\{x|\mathbf{y}\}$.

By definition of the Bregman loss function

$$\mathbb{E}\{\mathbf{L}_\phi(x,\kappa)\} - \mathbb{E}\{\mathbf{L}_\phi(x,\hat{x})\} = \mathbb{E}\{\phi(\hat{x}) - \phi(\kappa) - (x-\kappa)'\nabla\phi(\kappa) + (x-\hat{x})'\nabla\phi(\hat{x})\}$$

Last term is zero via the smoothing property of conditional expectation: $\mathbb{E}\{(x-\hat{x})'\nabla\phi(\hat{x})\} = \mathbb{E}\{\mathbb{E}\{(x-\hat{x})'\nabla\phi(\hat{x}) \mid \mathbf{y}\}\} = \mathbb{E}\{(\hat{x}-\hat{x})'\nabla\phi(\hat{x})\} = 0$. Using the smoothing property of conditional expectations yields

$$\mathbb{E}\{(x-\kappa)'\nabla\phi(\kappa)\} = \mathbb{E}\{\mathbb{E}\{(x-\kappa)'\nabla\phi(\kappa)|\mathbf{y}\}\} = \mathbb{E}\{(\hat{x}-\kappa)'\nabla\phi(\kappa)\}$$

So

$$\mathbb{E}\{\mathbf{L}_\phi(x,\kappa)\} - \mathbb{E}\{\mathbf{L}_\phi(x,\hat{x})\} = \mathbb{E}\{\phi(\hat{x}) - \phi(\kappa) - (\hat{x}-\kappa)'\nabla\phi(\kappa)\} = \mathbb{E}\{\mathbf{L}_\phi(\hat{x},\kappa)\}$$

2 Filtering

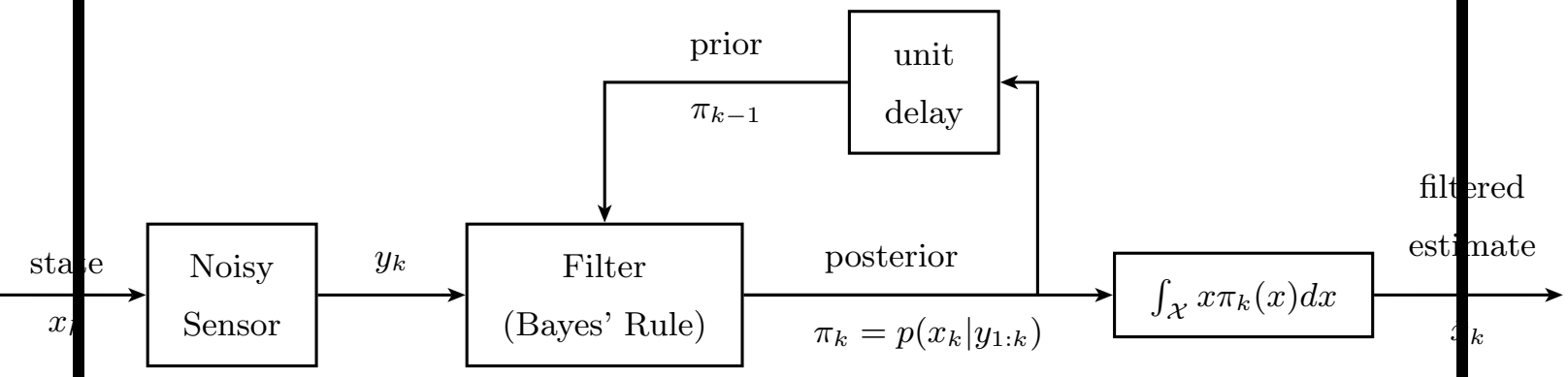


Figure 1: Schematic of optimal filter. The observation y_k from the noisy sensor is combined with the prior π_{k-1} via Bayes rule to compute the posterior $\pi_k = p(x_k|y_{1:k})$. The filtered (conditional mean) state estimate is then computed in terms of the posterior as $\int_{\mathcal{X}} x\pi_k(x)dx$.

2.1 The Problem

Given a stochastic dynamical system

$$\begin{aligned} x_{k+1} &= A_k(x_k) + \Gamma_k(x_k)w_k, \quad x_0 \sim \pi_0(\cdot) \\ y_k &= C_k(x_k) + D_k(x_k)v_k. \\ p(x_{k+1}|x_k) &= p_w\left(\Gamma_k^{-1}(x_k)\,[x_{k+1} - A_k(x_k)]\right) \\ p(y_k|x_k) &= p_v\left(D_k^{-1}(x_k)\,[y_k - C_k(x_k)]\right). \end{aligned}$$

Assume model and parameters are known.

Aim: Compute the min-variance state estimate $\hat{x}_{k|l}$ given the sequence of observations $Y_l = y_1, \ldots, y_l$.
As shown earlier $\hat{x}_{k|l} = \mathbb{E}\{x_k|Y_l\}$.

There are 3 problems of interest:

- **Filtering**: If $k = l$
- **Prediction**: If $k > l$
- **Smoothing**: If $k < l$.

We focus here on filtering. From these, smoothers and predictors can easily be obtained.

Remark: If x_k does not evolve – regression problem.
Note: Unbiased estimator.

2.2 Filtering solution

Model:

$$p(x_{k+1}|x_k) = p_w\left(\Gamma_k^{-1}(x_k)\,[x_{k+1} - A_k(x_k)]\right)$$
$$p(y_k|x_k) = p_v\left(D_k^{-1}(x_k)\,[y_k - C_k(x_k)]\right).$$

Aim: Recursively compute

$$\hat{x}_k = \mathbb{E}\{x_k|y_{1:k}\} = \int_{\mathcal{X}} x_k\, p(x_k|y_{1:k})dx_k, \quad k = 1, 2\ldots$$

Denoting $\pi_k(x) = p(x_k = x|y_{1:k})$ we want recursion on π_k

Main Result:

$$\pi_{k+1}(x_{k+1}) = \frac{p(y_{k+1}|x_{k+1})\int_{\mathcal{X}} p(x_{k+1}|x_k)\,\pi_k(x_k)\,dx_k}{\int_{\mathcal{X}} p(y_{k+1}|x_{k+1})\int_{\mathcal{X}} p(x_{k+1}|x_k)\,\pi_k(x_k)\,dx_k dx_{k+1}}$$
$$\hat{x}_{k+1} = \int_{\mathcal{X}} x\,\pi_{k+1}(x)dx$$

In terms of prediction and measurement update steps:

$$\pi_{k+1|k}(x_{k+1}) \stackrel{\text{defn}}{=} p(x_{k+1}|y_{1:k}) = \int_{\mathcal{X}} p(x_{k+1}|x_k)\,\pi_k(x_k)\,dx_k$$
$$\pi_{k+1}(x_{k+1}) = \frac{p(y_{k+1}|x_{k+1})\pi_{k+1|k}(x_{k+1})}{\int_{\mathcal{X}} p(y_{k+1}|x_{k+1})\pi_{k+1|k}(x_{k+1})dx_{k+1}}$$

- With only 2 exceptions (Kalman and HMM filter) Step 1 is not finite dimensional computable.
- Denominator= normalization = model likelihood

Un-normalized Filter Update

$$\tilde{\pi}_k(x) = p(x_k = x, y_{1:k}).$$

Clearly $\pi_k(x) = \tilde{\pi}_k(x) / \int_{\mathcal{X}} \tilde{\pi}_k(x)$

$$\boxed{\tilde{\pi}_{k+1}(x) = \int_{\mathcal{X}} p(y_{k+1}|x_{k+1} = x) p(x_{k+1} = x|x_k) \tilde{\pi}_k(x_k) dx_k.}$$

$$\hat{x}_{k+1} = \frac{\int_{\mathcal{X}} x\, \tilde{\pi}_{k+1}(x) dx}{\int_{\mathcal{X}} \tilde{\pi}_{k+1}(x) dx}.$$

Example

$$x_{k+1} = x_k + w_k, \quad w_k \sim N(0,1)$$

$$y_k = x_k + v_k, \quad v_k \sim N(0,1)$$

$$\text{Then } p(y_k|x_k) = \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{1}{2}(y_k - x_k)^2\right)$$

$$p(x_{k+1}|x_k) = \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{1}{2}(x_{k+1} - x_k)^2\right)$$

$$\tilde{\pi}_{k+1}(x_{k+1}) = \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{1}{2}(y_{k+1} - x_{k+1})^2\right) \times \int_{-\infty}^{\infty} \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{1}{2}(x_{k+1} - x_k)^2\right) \tilde{\pi}(x_k) dx_k$$

Can you guess a closed form expression for $\tilde{\pi}_k(x)$ so that $\tilde{\pi}_{k+1}(x)$ has the same form?

2.3 Kalman Filter

$$x_{k+1} = A_k x_k + f_k u_k + w_k, \quad x_0 \sim \pi_0$$
$$y_k = C_k x_k + g_k u_k + v_k.$$

$w_k \sim \boldsymbol{N}(0, Q_k)$, $v_k \sim \boldsymbol{N}(0, R_k)$ and initial density $\pi_0 \sim \boldsymbol{N}(\hat{x}_0, \Sigma_0)$ is Gaussian.

$$\hat{x}_{k+1|k} = A_k \hat{x}_k + f_k u_k, \quad y_{k+1|k} = C_{k+1} \hat{x}_{k+1|k} + g_{k+1} u_{k+1}$$
$$\Sigma_{k+1|k} = A_k \Sigma_{k|k} A_k' + Q_k$$
$$S_{k+1} = C_{k+1} \Sigma_{k+1|k} C_{k+1}' + R_{k+1}$$
$$\hat{x}_{k+1} = \hat{x}_{k+1|k} + \Sigma_{k+1|k} C_{k+1}' S_{k+1}^{-1} (y_{k+1} - y_{k+1|k})$$
$$\Sigma_{k+1} = \Sigma_{k+1|k} - \Sigma_{k+1|k} C_{k+1}' S_{k+1}^{-1} C_{k+1} \Sigma_{k+1|k}$$

$p(x_{k+1}|y_{1:k}) = \boldsymbol{N}(\hat{x}_{k+1|k}, \Sigma_{k+1|k})$ where

$\hat{x}_{k+1|k} = \mathbb{E}\{x_{k+1}|y_{1:k}\}, \quad \Sigma_{k+1|k} = \mathbb{E}\{(\hat{x}_{k+1} - x_{k+1})(\hat{x}_{k+1} - x_{k+1})'$

$$\hat{x}_{k+1|k} = (A_k - K_k C_k)\hat{x}_{k|k-1} + K_k (y_k - g_k u_k) + f_k u_k,$$
$$K_k = A_k \Sigma_{k|k-1} C_k' (C_k \Sigma_{k|k-1} C_k' + R_k)^{-1}$$
$$\Sigma_{k+1|k} = A_k \left(\Sigma_{k|k-1} - \Sigma_{k|k-1} C_k' (C_k \Sigma_{k|k-1} C_k' + R_k)^{-1} C_k \Sigma_{k|k-1}\right)$$

Riccati equation for covariance update.

Properties of the Kalman Filter

- Kalman Filter is linear, discrete-time, finite dimensional system with 2 sufficient statistics.
- Covariance $\Sigma_{k|k}$ can be precomputed since it is independent of the data.
- Stability of KF is related to stability of

$$\lambda_{k+1} = (A_k - K_k C_k)\lambda_k$$

- Steady State Kalman Filter. If A, B, C, Q and R are time-invariant, then under stability conditions K_k and Σ_k converge to a constant.
- Amongst the class of linear estimators the Kalman filter is the minimum variance estimator.
- Above derivation is algebraic. Another method is basd on projection theorem (Hilbert space approach to linear functionals).

For a detailed exposition of Kalman filters see "Optimal Filtering" by B.D.O Anderson and J.B.Moore, Prentice Hall, 1979.

Derivation of Kalman Filter

$$\boldsymbol{N}(x;\mu,P) = (2\pi)^{-X/2}|P|^{-1/2}\exp\left(-\frac{1}{2}(x-\mu)'P^{-1}(x-\mu)\right).$$

Swiss-Army-Knife for Gaussians:

$$\begin{aligned}&\boldsymbol{N}(y;Cx,R)\boldsymbol{N}(x;\mu,P)\\ &= \boldsymbol{N}(y;C\mu,CPC'+R)\ \boldsymbol{N}(x;m+\bar{K}(y-C\mu),P-\bar{K}CP)\end{aligned}$$

where in the right hand side of the above equation

$$\bar{K} = PC'(CPC'+R)^{-1}$$
$$m = \mu + \bar{K}(y-C\mu).$$

As a result, the following hold:

$$\int_{\mathcal{X}} \boldsymbol{N}(y;Cx,R)\boldsymbol{N}(x;\mu,P)dx = \boldsymbol{N}(y;C\mu,CPC'+R)$$

$$\frac{\boldsymbol{N}(y;Cx,R)\boldsymbol{N}(x;\mu,P)}{\int_{\mathcal{X}}\boldsymbol{N}(y;Cx,R)\boldsymbol{N}(x;\mu,P)dx} = \boldsymbol{N}(x;m+\bar{K}(y-C\mu),P-\bar{K}CP)$$

2.4 Hidden Markov Model Filter

Recall HMM is (P, B, π_0).

$$P_{ij} = \mathbb{P}(x_{k+1} = e_j | x_k = e_i), \quad B_{xy} = p(y_k = y | x_k = x)$$

$\mathcal{X} = \{e_1, \ldots, e_X\}$ where e_i is a X indicator vector with 1 in the i-th position.

HMM Filter: Since $\mathcal{X} = \{e_1, \ldots, e_X\}$, so

$$\pi_k(i) = \mathbb{P}(x_k = e_i | y_{1:k}), \quad i = 1, \ldots, X$$

$$\boxed{\pi_{k+1}(j) = \frac{p(y_{k+1}|x_{k+1} = e_j) \sum_{i=1}^X P_{ij} \pi_k(i)}{\sum_{l=1}^X p(y_{k+1}|x_{k+1} = e_l) \sum_{i=1}^X P_{il} \pi_k(i)}} \quad j = 1, \ldots, X,$$

In matrix-vector notation:

$$B_{y_k} = \text{diag}\left[p(y_k|x_k = e_1) \quad \cdots \quad p(y_k|x_k = e_X)\right].$$

$$\pi_k = \left[\pi_k(1) \quad \cdots \quad \pi_k(X)\right]'$$

$$\boxed{\pi_{k+1} = T(\pi_k, y_{k+1}) = \frac{B_{y_{k+1}} P' \pi_k}{\sigma(\pi_k, y_{k+1})}, \quad \sigma(\pi_k, y_{k+1}) = \mathbf{1}' B_{y_{k+1}} P' \pi_k.}$$

Compute the conditional mean estimate of $C' x_{k+1}$ as

$$C' \hat{x}_{k+1} = \mathbb{E}\{C' x_{k+1} | y_{1:k+1}\} = C' \pi_{k+1}.$$

$O(X^2)$ multiplications at each time

Belief Space:

$$\Pi(X) \stackrel{\text{defn}}{=} \left\{\pi \in \mathbb{R}^X : \mathbf{1}'\pi = 1, \quad 0 \le \pi(i) \le 1 \text{ for all } i \in \mathcal{X}\right\}$$

Unit vectors $e_1, e_2, \ldots, e_X$, that represent the X-states of the Markov chain are the vertices of this simplex Π.

Un-normalized HMM filter and Forward algorithm

$$\tilde{\pi}_{k+1} = p(x_{k+1}, y_{1:k+1}) = B_{y_{k+1}} P' \tilde{\pi}_k.$$

$$\hat{x}_{k+1} = \mathbb{E}\{x_{k+1}|y_{1:k+1}\} = \frac{\tilde{\pi}_{k+1}}{\mathbf{1}'\tilde{\pi}_{k+1}}.$$

Called *forward* algorithm.

Scaling: underflow problem remedied by scaling all the elements of $\tilde{\pi}_k$ by any arbitrary positive number. Since $\hat{x}_k$ involves the ratio of $\tilde{\pi}_k$ with $\mathbf{1}'\tilde{\pi}_k$, this scaling factor cancels out in the computation of $\hat{x}_k$.

HMM Predictor: $\pi_{k+\Delta} = {P'}^{\Delta} \pi_k$.

HMM Smoother: $\beta_{k|N}(x) = p(y_{k+1:N}|x_k = x)$.

$$\beta_{k|N} = \left[\beta_{k|N}(1), \ldots, \beta_{k|N}(X)\right].$$

Backward recursion

$$\beta_{k|N} = P B_{y_{k+1}} \beta_{k+1|N}, \quad k = N-1, \ldots, 1, \quad \beta_{N|N} = \mathbf{1}$$

$$\pi_{k|N}(i) = \mathbb{P}(x_k = i|y_{1:N}) = \frac{\pi_k(i)\beta_{k|N}(i)}{\sum_{l=1}^m \pi_k(l)\beta_{k|N}(k)}$$

Markov Modulated Auto-regressive Time Series

Given $y_1, \ldots, y_{k-1}$, distribution of y_k depends not only on the state x_k of the Markov chain but also on

$y_{k-1}, \ldots, y_{k-d}$

Example: Linear autoregressions with Markov regime

$$y_k + a_1(x_k)\, y_{k-1} + \cdots + a_d(x_k)\, y_{k-d} = \Gamma(x_k) w_k,$$

Identical to HMM filter with observation density

$$B_{x,y_k} = p_w\left(\Gamma^{-1}(x)(y_k + a_1(x)\, y_{k-1} + \cdots + a_d(x)\, y_{k-d})\right).$$

2.5 Viterbi Algorithm for HMM State estimation

Unlike HMM filter, Viterbi algorithm generates Maximum likelihood sequence estimates. Let $X_T = (x_1, \ldots, x_k)$ and $Y_T = (y_1, \ldots, y_T)$. Then Viterbi algorithm computes

$$\begin{aligned}\hat{X}_T &= \arg\max_{X_T} p(Y_T, X_T) \\ &= \arg\max_{X_T} \prod_{k=1}^{T} p(y_k|x_k)p(x_k|x_{k-1})p(x_1)\end{aligned}$$

Solve via forward dynamic programming: For $k = 1, 2, \ldots, T$

$$\delta_{k+1}(j) = \max_i \Big[\delta_k(i)P_{ij}\Big] p(y_{k+1}|x_{k+1} = q_j)$$

$$u_{k+1}(j) = \arg\max_i \Big[\delta_k(i)P_{ij}\Big] p(y_{k+1}|x_{k+1} = q_j)$$

Terminate at $\hat{x}_T = \arg\max_i \delta_T(i)$.
Then backtrack to read off MLSE $\hat{X}_T$ as $\hat{x}_k = u_{k+1}(\hat{x}_{k+1})$.

Viterbi generates hard estimates. But hard to analyse its statistical properties.
In computer implementation use $\log \delta_k$ to avoid numerical underflow.

Comparison of Kalman and HMM filter

(i) KF is linear filter, HMM filter is nonlinear filter.

(ii) KF requires Gaussian noise and a linear state space model. In non Gaussian noise, KF is linear min-var estimator. The HMM filter does not require Gaussian noise – it works for any noise density. Also the observation equation does not have to be linear in the Markov state.
(iii) KF is optimal for correlated noise (linearly filtered white noise). HMM filter depends crucially on the whiteness of v_k
(iv) Both HMM and KF are geometrically ergodic, i.e. they forget their initial condition exponentially fast.
(v) **Martingale formulation of HMM**: Let $x_k \in \{e_1, \ldots, e_S\}$ denote states of Markov chain. Then HMM can be represented as

$$x_{k+1} = A' x_k + M_k$$
$$y_k = C x_k + v_k$$

where M_k is a finite state martingale increment.

Geometric Ergodicity of Optimal Filter

Assumption: (strong mixing - Attar, Zeitouni, 1997)

$$\sigma^- \mu_j \leq P_{ij} \leq \sigma^+ \mu_j$$

$$\text{and } 0 < \sum_{j=1}^{X} \mu_j B_{jy} < \infty \text{ for all } y \in \mathcal{Y}.$$

$0 < \sigma^- \leq \sigma^+$ and μ is a pmf.

Theorem: Consider two HMMs (P, B, π_0) and $(P, B, \bar{\pi}_0)$. Let π_k and $\bar{\pi}_k$ denote the filtered pmfs. Then

$$\|\pi_k - \bar{\pi}_k\|_{\text{TV}} \leq 2 \frac{\sigma^+}{\sigma^-} \left(1 - \frac{\sigma^-}{\sigma^+}\right)^k \|\pi_0 - \bar{\pi}_0\|_{\text{TV}}$$

$$P_{ij}(n|k) = \mathbb{P}(x_n = j | x_{n-1} = i, x_{0:n-2}, y_{1:k})$$

Fixed-interval smoothed conditional probability vector

$$\pi_{n|k} = \begin{bmatrix} \mathbb{P}(x_n = 1|y_{1:k}) & \cdots & \mathbb{P}(x_n = X|y_{1:k}) \end{bmatrix}'$$

where $n \leq k$. Chapman Kolomogorov equation

$$\pi_{n|k} = P'(n|k)\pi_{n-1|k} = \prod_{l=n}^{1} P'(l|k)\pi_{0|k}.$$

Sub-multiplicative property of the Dobrushin coefficient

$$\|\pi_{n|k} - \bar{\pi}_{n|k}\|_{\text{TV}} = \|\prod_{l=n}^{1} P'(l|k)\pi_{0|k} - \prod_{l=n}^{1} P'(l|k)\bar{\pi}_{0|k}\|_{\text{TV}}$$

$$\leq \|\pi_{0|k} - \bar{\pi}_{0|k}\|_{\text{TV}} \prod_{l=1}^{n} \rho(P(l|k)), \quad n = 1, 2 \ldots, k.$$

Step 1. Show Dobrushin coefficients $\rho(P(l|k)) < 1$.

Step 2. Establish an upper bound for $\|\pi_{0|k} - \bar{\pi}_{0|k}\|_{\text{TV}}$.

Since always smaller than 1, Step 1 suffices. Sharper bound: $\|\pi_{0|k} - \bar{\pi}_{0|k}\|_{\text{TV}}$ in terms of $\|\pi_0 - \bar{\pi}_0\|_{\text{TV}}$.

We proceed with Step 1.

Theorem: $\rho(P(l|k)) \leq 1 - \frac{\sigma^-}{\sigma^+}$

Proof:

$$P_{ij}(l|k) = \mathbb{P}(x_l = j | x_{l-1} = i, x_{0:l-2}, y_{1:k})$$

$$= \frac{P_{ij} B_{jy_l} \beta_{l|k}(j)}{\sum_{x=1}^{X} P_{ix} B_{xy_l} \beta_{l|k}(x)} \geq \frac{\sigma^-}{\sigma^+} \frac{\mu_j B_{jy_l} \beta_{l|k}(j)}{\sum_{x=1}^{X} \mu_x B_{xy} \beta_{l|k}(x)}$$

Denote

$$\epsilon = \frac{\sigma^-}{\sigma^+}, \quad \kappa_j = \frac{\mu_j B_{jy_l} \beta_{l|k}(j)}{\sum_{x=1}^{X} \mu_x B_{xy} \beta_{l|k}(x)}$$

Clearly κ_j is a probability mass function and $\epsilon \in (0, 1]$. So we have Doeblin condition $P_{ij}(l|k) \geq \epsilon \kappa_j$. Therefore $\rho(P(l|k)) \leq 1 - \epsilon$.

2.6 Approximate Filters

For general nonlinear systems – no finite dimensional filter exists.
The following approximations are widely used.

1. Deterministic Grid approximation: HMM approximation (e.g. bearings only target tracking)

$$\tilde{\pi}_{k+1}(x_{k+1}) = p(y_{k+1}|x_{k+1}) \int_{\mathbb{R}} p(x_{k+1}|x_k)\tilde{\pi}_k(x_k)dx_k$$

Discretizing x to the grid $[r_1, \ldots, r_M]$ yields

$$\tilde{\pi}_{k+1}(r_j) = p(y_{k+1}|x_{k+1} = r_j) \sum_{i=1}^{M} p(x_{k+1} = r_j|x_k = r_i)\tilde{\pi}_k(r_i)$$

Complexity: $O(M^2)$ at each time instant.
Approx error: $O(M^{-1/N})$ (N = state dim) – suffers from curse of dimensionality.

2. Extended Kalman Filter: Linearize, then run KF. Unscented Kalman filter is more sophisticated.

3. MAP estimators: Compute the MAP state estimate (modal filtering) $\arg\max_{x_1,\ldots,x_T} p(Y_T, x_1, \ldots, x_T)$.

4. Basis function approximations: (i) Gaussian sum approximations, (ii) Particle filters

Particle filters

Outline: Particle filters are a randomized grid sub-optimal algorithm for nonlinear filtering. They use the delta function basis approximation

$$p(x_{0:n}|y_{1:n}) \approx \sum_{i=1}^{N} \tilde{w}_k^{(i)} \delta(x_{0:n}^{(i)}).$$

The positions $\delta(x_{0:n}^{(i)})$ of the N particles propagate randomly according to system dynamics.
Weights $\tilde{w}_k^{(i)}$ are updated via Bayes rule.

While deterministic grid error is $O(N^{-1/X})$, for particle filter the mean square error from CLT is $O(N^{-1})$ (randomization breaks the curse of dimensionality!).

The bootstrap particle filter was invented in 1968 by D.Q Mayne. It was re-invented in 1996. Particle filters are a class of sequential Markov Chain Monte Carlo (MCMC) algorithms.

1. **Model**: $p(x_{k+1}|x_{0:k}), \quad p(y_k|y_{1:k-1}, x_{1:k})$.

Aim: Compute $\mathbb{E}\{\phi(x_{0:k})|y_{1:k}\}$ via sequential MCMC.
We estimate pdf $p(x_{0:k}|y_{1:k})$, $k = 1, 2, \ldots$.

2. **Bayesian Importance sampling**:

$$\mathbb{E}\{\phi(x_{0:k}|y_{1:k})\} = \int \phi(x_{0:k}) \frac{p(x_{0:k}|y_{1:k})}{\pi(x_{0:k}|y_{1:k})} \pi(x_{0:k}|y_{1:k}) dx_{0:k}$$

Sample $x_{0:k}^{(i)} \sim \pi(x_{0:k}|y_{1:k})$, then by SLLN

$$\sum_{i=1}^{N} \phi(x_{0:k}^{(i)}) \frac{w_k^{(i)}}{\sum_j w_k^{(j)}} \to \mathbb{E}\{\phi(x_{0:k})|y_{1:k}\}, \quad w_k^{(i)} = \frac{p(x_{0:k}^{(i)}|y_{1:k})}{\pi(x_{0:k}^{(i)}|y_{1:k})}$$

4. **Sequential Importance sampling**:

$$\pi(x_{0:k}|y_{1:k}) = \pi(x_0|y_{1:k}) \prod_{t=1}^{k} \pi(x_t|x_{0:t-1}, y_{1:k})$$

$$\text{Real time: } \pi(x_{0:k}|y_{1:k}) = \pi(x_0) \prod_{t=1}^{k} \pi(x_t|x_{0:t-1}, y_{1:t})$$

$$w_k(x_{0:k}^{(i)}) = \frac{p(x_{0:k}^{(i)}|y_{1:k})}{\pi(x_{0:k}^{(i)}|y_{1:k})} \quad \propto \frac{p(y_k|y_{1:k-1}, x_{0:k}^{(i)})\, p(x_k^{(i)}|x_{0:k-1}^{(i)})}{\pi(x_k^{(i)}|x_{0:k-1}^{(i)}, y_{1:k})} w_{k-1}(x_{0:k-1}^{(i)})$$

$$\text{particle filter: } p(x_{0:k}|y_{1:k}) \approx \sum_{i=1}^{N} \frac{w_k(x_{0:k}^{(i)})}{\sum_{j=1}^{N} w_k(x_{0:k}^{(j)})} \delta(x_{0:k}^{(i)}).$$

Summary: Particle filter algorithm

Sequential Importance Sampling step: At each time k

- Sample N particles $\tilde{x}_k^{(i)} \sim \pi(x_k|x_{0:k-1}^{(i)}, y_{1:k})$.
  Set $\tilde{x}_{0:k}^{(i)} = (x_{0:k-1}^{(i)}, \tilde{x}_k^{(i)})$.
- Update importance weights w_k and normalized importance weights $\tilde{w}_k^{(i)}$ of particles

$$w_k(\tilde{x}_{0:k}^{(i)}) \propto \frac{p(y_k|y_{1:k-1}, \tilde{x}_{0:k}^{(i)}) p(\tilde{x}_k^{(i)}|\tilde{x}_{0:k-1}^{(i)})}{\pi(\tilde{x}_k^{(i)}|x_{0:k-1}^{(i)}, y_{1:k})} w_{k-1}(x_{0:k-1}^{(i)})$$

$$\tilde{w}_t^{(i)} = \frac{w_k(\tilde{x}_{0:k}^{(i)})}{\sum_{j=1}^N w_k(\tilde{x}_{0:k}^{(j)})}.$$

Selection step: Effective no. of particles: $\hat{N} = \frac{1}{\sum_{i=1}^N \tilde{w}_k^{(i)}}$.

If $\hat{N}$ is smaller than a prescribed threshold, then

- Multiply/Discard particles $\tilde{x}_{0:k}^{(i)}$, $i = 1, \ldots, N$ with high/low normalised importance weights $\tilde{w}_k^{(i)}$ to obtain N new particles $x_{0:k}^{(i)}$, $i = 1, \ldots, N$.

Remarks: If $\phi(x_{0:n}) = x_n$, then memory required: $O(N)$.

$$\lim_{N\to\infty} \frac{\sum_{i=1}^N x_n^{(i)} w_n^{(i)}}{\sum_{i=1}^N w_n^{(i)}} \to \mathbb{E}\{x_n|y_{0:n}\} \quad \text{a.s.}$$

Implementation Issues

1. **Choice of importance function**: zillions of papers

(i) *Optimal choice*: (min var of $p(x_{0:k}|y_{0:k})/\pi(x_{0:k}|y_{0:k})$).

$$\pi(x_k|x_{0:k-1}, y_{0:k}) = p(x_k|x_{k-1}^{(i)}, y_k)$$

Then $p(x_k|x_{k-1}, y_k) = \frac{p(y_k|x_k)p(x_k|x_{k-1})}{p(y_k|x_{k-1})}$.
So $w_k^{(i)} = w_{k-1}^{(i)} p(y_k|x_{k-1}^{(i)})$.
Compute $w_k^{(i)}$, sample $x_n^{(i)}$ in parallel since indpt of $x_k^{(i)}$!

Disadvantages of optimal importance function:
(a) Need to be able to sample from $p(x_k|x_{k-1}^{(i)}, y_k)$
(b) Need to be able to compute $p(y_k|x_{k-1})$ in closed form.

Example: Nonlinear dynamics, linear observation:

$$\begin{aligned} x_{k+1} &= f(x_k) + v_k, \qquad v_k \sim N(0, \Sigma_v) \\ y_k &= Cx_k + w_k, \qquad w_k \sim N(0, \Sigma_w) \end{aligned}$$

Then $p(x_k|x_{k-1}, y_k) = N(m_k, \Sigma)$ where

$$\begin{aligned} \Sigma^{-1} &= \Sigma_v^{-1} + C'\Sigma_w^{-1}C \\ m_k &= \Sigma(\Sigma_v^{-1} f(x_{k-1})_C' \Sigma_w^{-1} y_k) \\ p(y_k|x_{k-1}) &= N\left(Cf(x_{k-1}), (\Sigma_w + C\Sigma_v C')\right) \end{aligned}$$

(ii) Prior Importance function: (Mayne 1969, Tanizaki

1997): $\pi(x_k|x_{0:k-1}, y_{0:k}) = p(x_k|x_{k-1})$.
Then $w_k^{(i)} = w_{k-1}^{(i)} p(y_k|x_k^{(i)})$. Sensitive to outliers.
Particles evolve indpt of obs.

(iii) Fixed importance function: $\pi(x_k|x_{0:k-1}, y_{0:k}) = p(x_k)$
Tanizaki (1994, econometrics), Kitagawa (1987).

2. **Degeneracy**: Variance of importance weights $w_k^{(i)}$ grows with time. Most particle weights become close to zero – ill-conditioning.
Selection/Resampling Step: (zillions of papers)
(i) Discard particles with low normalized weight.
(ii) Multiply particles with high weight.
Before selection: $p(x_{0:k}|y_{0:k}) \propto \sum_i w_k^{(i)} \delta(x_{0:k}^{(i)})$
After selection: $p(x_{0:k}|y_{0:k}) \propto \sum_i \delta(x_{0:k}^{(i)})$
Methods: Sampling Importance Resampling, etc
Selection scheme increases variance - so compromise between degeneracy and variance.
3. **Variance reduction by conditioning (Rao-Blackwellization)**: Based on the idea

$$\text{var}\{\mathbb{E}\{X|Y\}\} \leq \text{var}\{\mathbb{E}\{X\}\}$$

Remark: In basic algorithm given, the particles do not interact. Resampling (which reduces the degeneracy problem) introduces dependencies in the particles. Convergence is much more difficult to prove.

Example: Jump Markov Linear System

$$z_{k+1} = A(r_{k+1})\, z_k + \Gamma(r_{k+1})\, w_{k+1} + f(r_{k+1})\, u_{k+1}$$
$$y_k = C(r_k)\, z_k + D(r_k)\, v_k + g(r_k) u_k.$$

Aim: Estimate joint posterior $p(r_{0:k}, z_{0:k}|y_{1:k})$.

Key insight: Why use a particle filter for JMLS?

$$p(r_{0:k}, z_{0:k}|y_{1:k}) = p(z_{0:k}|y_{1:k}, r_{0:k}) p(r_{0:k}|y_{1:k}).$$

$p(x_{0:k}|y_{1:k}, r_{0:k})$ is Gaussian. $p(r_{0:k}|y_{1:k})$ is X^k mixture.

Can reformulate estimation of $p(r_k, z_k|y_{1:k})$ as sampling from $p(r_{0:k}|y_{1:k})$. Use particle filter Algorithm

$$w(r_{0:t}) \propto \frac{p\left(y_t \middle| y_{1:t-1}, r_{0:t}\right) p\left(r_t \middle| r_{t-1}\right)}{\pi\left(r_t \middle| y_{1:t}, r_{0:t-1}\right)} w(r_{0:t-1})$$

to estimate $p(r_{0:k}|y_{1:k})$

Data Augmentation Algorithm for Fixed-interval Smoothing of JMLS

1. Initialization. Choose $\left(r_{0:N}^{(0)}, z_{0:N}^{(0)}\right)$ randomly.
2. Iteration. For $n = 1, 2, \ldots$: Given $\left(r_{0:N},^{(n-1)}, z_{0:N},^{(n-1)}\right)$, compute $\left(r_{0:N},^{(n)}, z_{0:N},^{(n)}\right)$, for the nth iteration as follows:
   - Simulate $z_{0:N}^{(n)} \sim p(z_{0:N}|y_{1:N}, r_{0:N}^{(n-1)})$.
   - Simulate $r_{0:N}^{(n)} \sim p\left(r_{0:N}|y_{1:N}, z_{0:N}^{(n)}\right)$.
3. State Estimation: Compute fixed-interval smoothed estimates after N iterations as

   $$\hat{r}_{0:N}^{N} = \frac{1}{N}\sum_{n=0}^{N-1} \mathbb{E}\{r_{0:N}|y_{1:N}, z_{0:N}^{(n)}\}, \quad \hat{z}_{0:N}^{N} = \frac{1}{N}\sum_{n=0}^{N-1} \mathbb{E}\{z_{0:N}|y_{1:N}$$

   These are implemented, respectively, using the HMM smoother and Kalman smoother algorithms.

2.7 Multi-agent State Estimation: Social Learning

Multi-agent system aims to estimate Markov state.
Each agent acts once in a predetermined sequential order indexed by $k = 1, 2, \ldots$.
(i) *Private Observation*: At time k, agent k records a private observation $y_k \in \mathcal{Y} = \{1, 2, \ldots, Y\}$ from the observation distribution $B_{iy} = \mathbb{P}(y|x = i)$, $i \in \mathcal{X}$.
(ii) *Private Belief*: Using public belief π_{k-1} available at time $k-1$, agent k updates private belief
$\eta_k(i) = P(x_k = i|a_1, \ldots, a_{k-1}, y_k)$

$$\eta_k = \frac{B_{y_k} P' \pi}{\mathbf{1}_X' B_y P' \pi}, \qquad \text{where } B_{y_k} = \text{diag}(P(y_k|x = i), i \in \mathcal{X}).$$

(iii) *Myopic Action*: Agent k takes action
$a_k \in A = \{1, 2, \ldots, A\}$ to minimize its expected cost

$$a_k = a(\pi_{k-1}, y_k) = \arg\min_{a \in A} \mathbb{E}\{c(x, a)|a_1, \ldots, a_{k-1}, y_k\}$$

$$= \arg\min_{a \in A} \{c_a' \eta_k\}.$$

Here $c_a = (c(i, a), i \in \mathcal{X})$ is cost vector
(iv) *Social Learning Filter*: Other agents use a_k to

perform social learning $\pi_k(j) = P(x_k = j|a_1, \ldots, a_k)$.

$$\pi_k = T(\pi_{k-1}, a_k), \text{ where } T(\pi, a) = \frac{R_a^\pi P'\pi}{\sigma(\pi, a)}, \ \sigma(\pi, a) = \mathbf{1}_X' R_a^\pi P'\pi$$

$$R_a^\pi = \text{diag}(P(a|x = i, \pi), i \in \mathcal{X})$$

with elements

$$P(a_k = a|x_k = i, \pi_{k-1} = \pi) = \sum_{y \in \mathcal{Y}} P(a|y, \pi) P(y|x_k = i)$$

$$\text{where } P(a_k = a|y, \pi) = \begin{cases} 1 \text{ if } c_a' B_y P'\pi \leq c_{\tilde{a}}' B_y P'\pi, & \text{for al } \tilde{a} \in \\ 0 \text{ otherwise} \end{cases}$$

Remarks: (i) Filtering with hard decisions: E.g. $A = \mathcal{X}$, $c_a = -e_a$ then $\text{argmin}_a \, c_a'\pi = \text{argmax}_a \, \pi(a)$ (MAP estimate).
(ii) Dependence of observation likelihood on prior.

- An individual agent k *herds* on the public belief π_{k-1} if it chooses its action $a_k = a(\pi_{k-1}, y_k)$ independently of its observation y_k.
- A *herd of agents* takes place at time k^*, if the actions of all agents after time k^* are identical, i.e., $a_k = a_{k^*}$ for all time $k > k^*$.
- An *information cascade* occurs at time k^*, if the public beliefs of all agents after time k^* are identical, i.e. $\pi_k = \pi_{k^*}$ for all $k < k^*$.

Result: An information cascade takes place if and only there exists some time k^*, such that for all $k \geq k^*$, the diagonal elements $\mathbb{P}(a_k|x=i)$, $i \in \mathcal{X}$ of $R^{\pi}_{a_k}$ are identical.

Remark:
(i) An information cascade implies a herd of agents, that is the action of all agents are identical after some time k^*. But a herd of agents does not imply an information cascade. This is because, even if all agents pick the same action for time k^* onwards, it is possible for the social belief π_k, $k \geq k^*$ to evolve
(ii) If all individual agents herd after some time k^*, then an information cascade occurs. If all individual agents herd from a time k^*, then no information is revealed after time k^* and social learning ceases. Equivalently, $\mathbb{P}(a|y,\pi)$ is independent of y, so

$$P(a_k = a|x=i, \pi_{k-1}=\pi) = \sum_y \mathbb{P}(y|x=i) = 1$$

for all $i \in \mathcal{X}$, and the social belief update freezes. Information cascade implies all individual agents herd

Theorem: The social learning protocol leads to an information cascade in finite time with probability 1. That is there exists a finite time k^* after which social learning ceases, i.e., public belief $\pi_{k+1} = \pi_k$, $k \geq k^*$, and all agents choose the same action, i.e., $a_{k+1} = a_k$, $k \geq k^*$.

Proof: Define $\lambda_k(i,j) = \log(\pi_k(i)/\pi_k(j))$, $i,j \in \mathcal{X}$.

$$\lambda_k(i,j) = \lambda_{k-1}(i,j) + \gamma_k(i,j)$$

$$\gamma_k(i,j) = \log \frac{P(a_k|x=i,\pi)}{P(a_k|x=j,\pi)}.$$

$\bar{\mathbf{Y}}_k$: observation symbols for which it is optimal not to choose u given belief π. So

$$\mathbb{P}(a_k = a|x,\pi) = 1 - \mathbb{P}(y_k \in \bar{\mathbf{Y}}_k|x)$$

$$\gamma_k(i,j) = \log \frac{1 - \sum_{y\in\bar{\mathbf{Y}}_k} \mathbb{P}(y|x=i)}{1 - \sum_{y\in\bar{\mathbf{Y}}_k} P(y|x=j)}.$$

If a cascade forms, then $\bar{\mathbf{Y}}_k$ is the empty set. For any y it is always optimal to pick action a. So $\gamma_k(,i,j) = 0$.

If $\bar{\mathbf{Y}}_k$ is non-empty, then $|\gamma_k(i,j)| > K$ where $K > 0$.

Define the filtration sigma-algebra $\mathcal{A}_k = \sigma(a_1,\ldots,a_k)$. $\pi_k(i) = P(x=i|a_1,\ldots,a_k) = \mathbb{E}\{I(x=i)|\mathcal{A}_k\}$ is a $\mathcal{A}_k$ martingale, since

$$\mathbb{E}\{\pi_{k+1}(i)|\mathcal{A}_k\} = \mathbb{E}\{\mathbb{E}\{I(x=i|\mathcal{A}_{k+1}\}|\mathcal{A}_k\} = \mathbb{E}\{I(x=i|\mathcal{A}_k\}$$

(via the smoothing property of conditional expectations). So by the martingale convergence theorem, there exists a random variable π_∞, such that $\pi_k \to \pi_\infty$ with probability 1 (w.p.1). Therefore $\lambda_k(i,j) \to \lambda_\infty(i,j)$ w.p.1. Now since $\lambda_k(i,j) \to \lambda_\infty(i,j)$ w.p.1, there exists some k^* such for all

$k \geq k^*$, that $|\lambda_k(i,j) - \lambda_\infty(i,j)| < K/3$ (where K is the constant defined above).

$$\text{Therefore, } |\lambda_{k+1}(i,j) - \lambda_k(i,j)| < 2K/3. \qquad (1)$$

Suppose a cascade does not form. Then $P(a|x=i,\pi)$ is different from $P(a|x=j,\pi)$ for at least one pair $i,j \in \mathcal{X}$, $i \neq j$. This implies that the set $\bar{\mathbf{Y}}_k$ is non-empty and so

$$|\gamma_{k+1}(i,j)| = |\lambda_{k+1}(i,j) - \lambda_k(i,j)| \geq K \qquad (2)$$

So (1), (2) constitute a contradiction.

Multiagent Estimation: Data Incest Problem

Rumor propagation.

$$\begin{array}{ccccc} (1,1) & \rightarrow & (1,2) & \rightarrow & (1,3) \\ & \searrow & & \nearrow & \\ (2,1) & \rightarrow & (2,2) & \rightarrow & (2,3) \end{array}$$