ECE 5412. Bayesian Estimation and Stochastic Optimization

Prof. Vikram Krishnamurthy
Electrical & Computer Engineering
Cornell University
email: vikramk@cornell.edu.
Rhodes 330. Updated November 26, 2019

This document contains all the slides used in class.

These slides are work in progress and will be updated regularly. There may be typo errors. Your feedback is appreciated.

Classes: Tue and Thu from 8:40 to 9:55 am in Gates G01.

Website: `canvas.cornell.edu`

Office hours: Fri 3:30 to 4:45 pm or email me for appointment.

---

1. We cover advanced topics in machine learning, data science, signal processing and reinforcement learning - you need to understand the underlying mathematics (why it works) and algorithms (how it works). This is **not** applied machine learning

2. We cover fundamentals in detail & outline applications. We won't spend time on specific applications or algorithm implementation. Some assignments focus on applications.

3. The slides are not lecture notes! The material is dense and needs careful thought. Please dont skip class.

Assessment

- Entrance Exam Today: You need to get at least 7/12.
- 12 Quizzes $= 12 \times 3 = 36$ % (First 10 mins of each class). We choose your best 12 out of 20 quizzes.
- 4 Assignments $= 4 \times 5 = 20$ %
- Final Exam (take home) $= 44$ %

Why Bayesian Estimation & Stochastic Opt?

`http://signalprocessingsociety.org/our-story/signal-processing-101`

Machine learning and Data Science: classifiers, logistic models, big data problems,

Network Science: How to model, estimate and control information in social networks.

Wireless Comms, Sensing Systems localization & tracking in robotics.

Geophysics: ground penetration radar, deconvolution

Aerospace/Defense Systems: Surveillance, Radar and Sonar Tracking of manuevering targets.

Mathematical finance and economics: predicting stock market, options pricing. Merton & Scholes won the 1997 Nobel prize in economics.
See `www.ams.org/new-in-math/nobel1997econ.html`

Objectives

Graduate course for beginning PhD students and MEng students who want to do advanced fundamental research.
(i) Understand basics of stochastic state space models, Hidden Markov processes and MCMC stochastic simulation
(ii) Least Squares Inference, Nonlinear Bayesian filters
(iii) Maximum Likelihood Estimation and EM algorithms
(iv) Stochastic optimization & Stochastic Gradient Algorithms.

Prerequisites: (i) Detailed working knowledge of undergrad probability & random processes, linear algebra, advanced calculus, basic optimization, signals & systems
(ii) Get at least 7/12 in today's entrance exam.
(iii) Strong in abstract thinking and math.

1. This course: derivation/construction of new algorithms, and mathematical proof of performance. *Don't do this course if you just want to implement machine learning algorithms for computer vision.*
2. On completion of course: you will be able to read and understand state-of-the-art research papers. *Don't do this course if you will not use the material in your research.*
3. This course emphasizes brain-work, mathematical, analytic thinking, and small amts of Matlab/Python simulation. *Don't do this course if you like implementation, coding and are weak in math.*

Books

Most material on internet/wikipedia. Graduate level books:

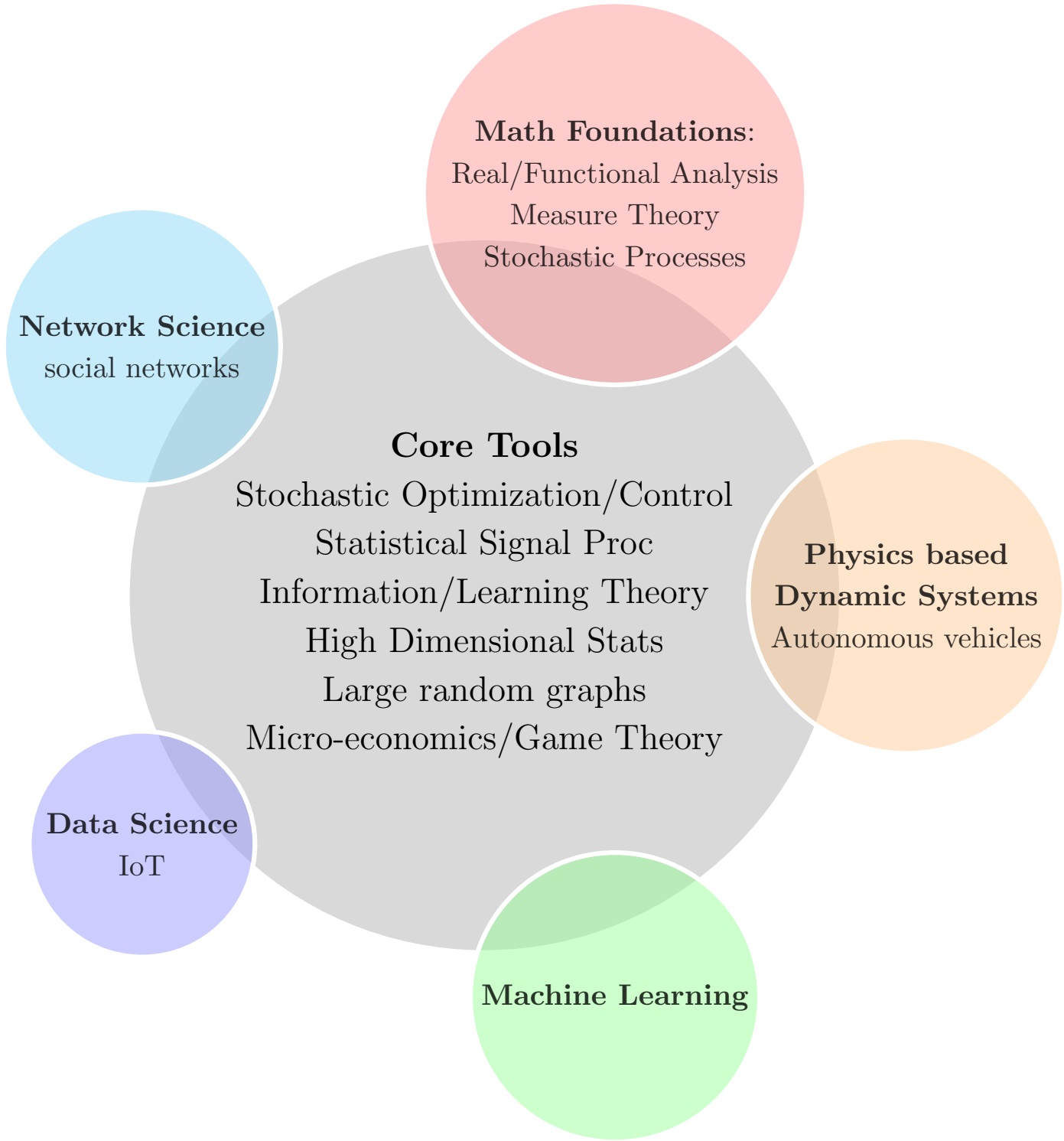
1. V. Krishnamurthy, Partially Observed Markov Decision Processes, Cambridge Univ Press, 2016.
2. L. Lung, System Identification for the user.
3. S. Ross, Simulation
4. Robert & Casella, Monte Carlo Statistical Methods
5. M. Jackson, Social and Economic Networks
6. Wainwright, High Dimensional Statistics - A non-asymptotic viewpoint
7. Giraud, Introduction to High Dimensional Statistics

Background undergraduate material: You should be *very familiar* with:

1. Probability & Random Processes: E.g. MIT OCW Probabilistic Systems Analysis and Applied Probability
2. Linear Algebra: E.g. MIT OCW Linear Algebra
3. Undergrad signals and systems: E.g. entire Oppenheim & Willsky textbook
4. Basic Dynamical Systems and State Space Models: Undergrad Control systems e.g. Franklin & Powell textbook
5. Basic optimization & engineering math. E.g. MIT OCW Engineering Math: Differential Equations and Linear Algebra

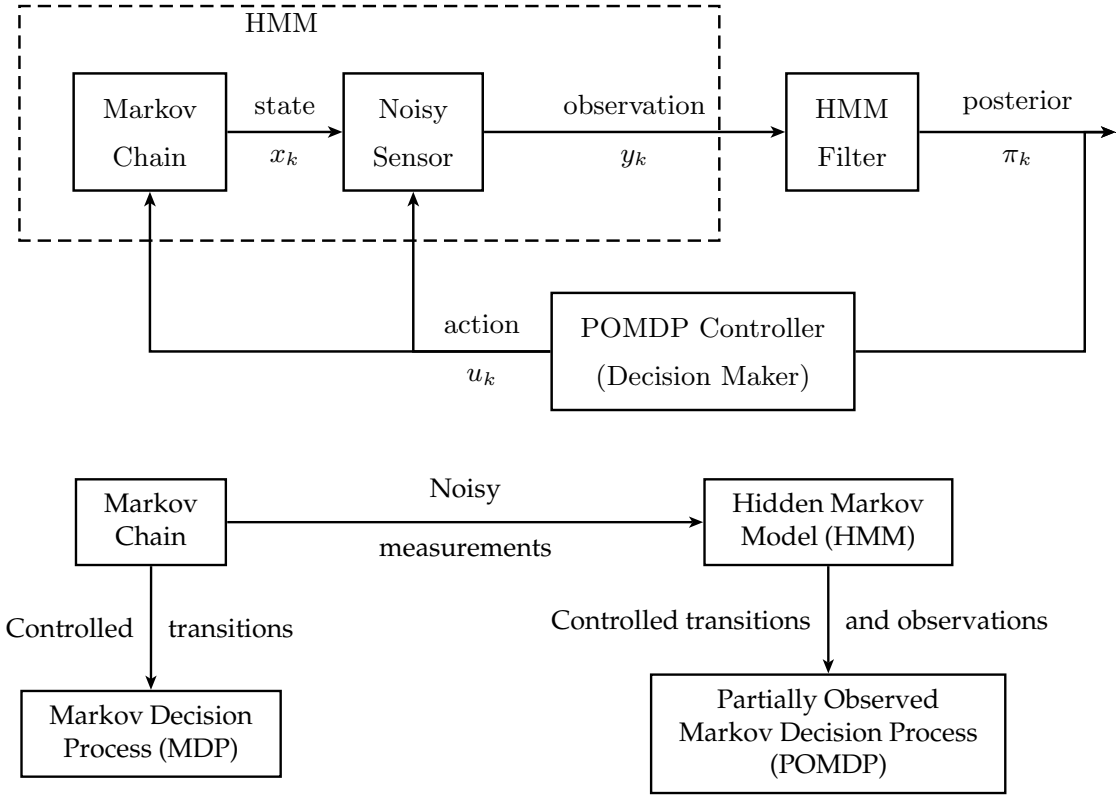
Core Graduate Courses

for meaningful PhD research...



Big Picture

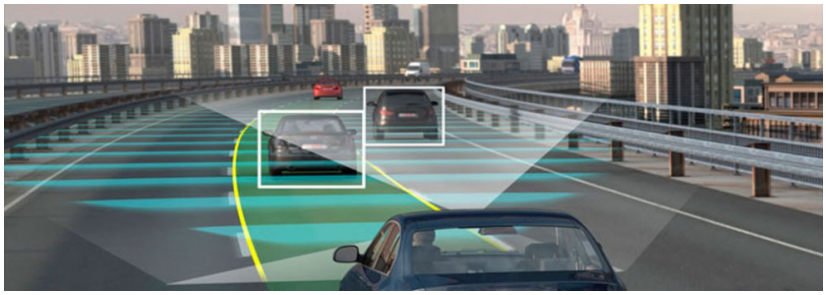
Smart (Cognitive) Autonomous Reconfigurable Sensing.



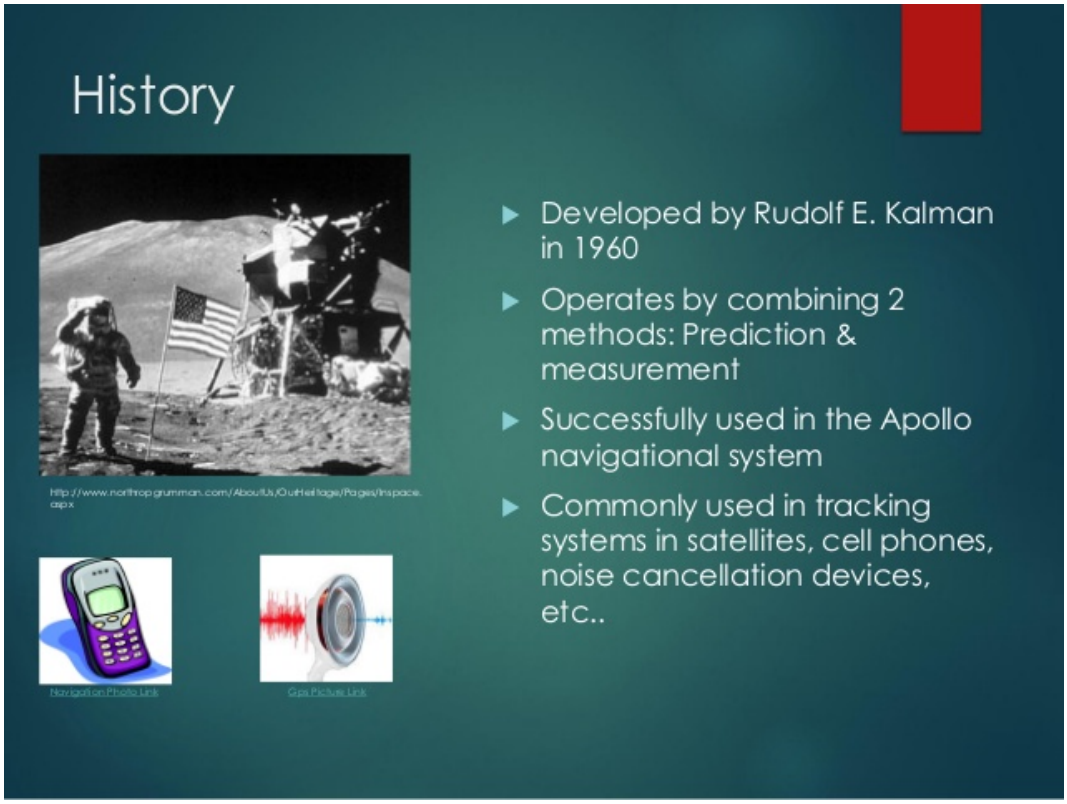
MCMC, Bayesian filtering, ML estimation, stochastic optimization are ubiquitous is sensing, data science, network science, machine learning

Examples of Bayesian Inference

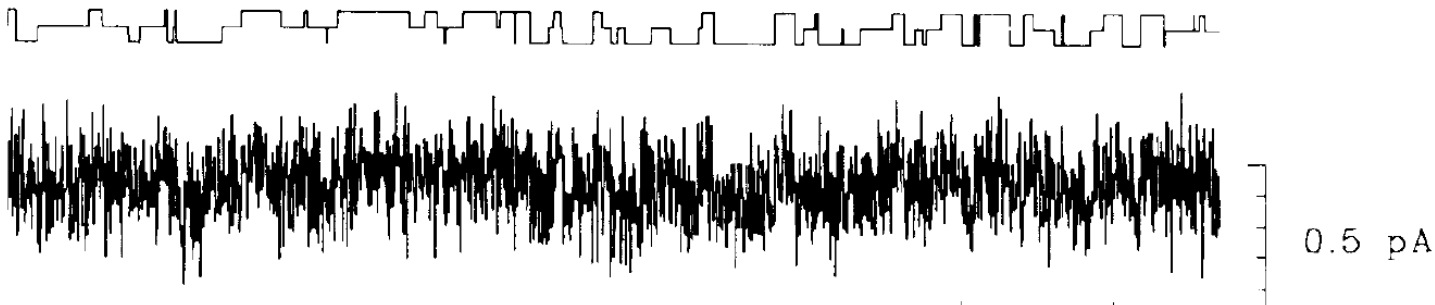
Tracking. Given noisy measurements of random process x_k, how to estimate x_k? Self driving cars [Optimal Filtering]



Kalman filter



Neurobiology: Ion channel gating modeled as Hidden Markov Model



Examples of Bayesian Feedback Systems.

Ex 1: Recommender System

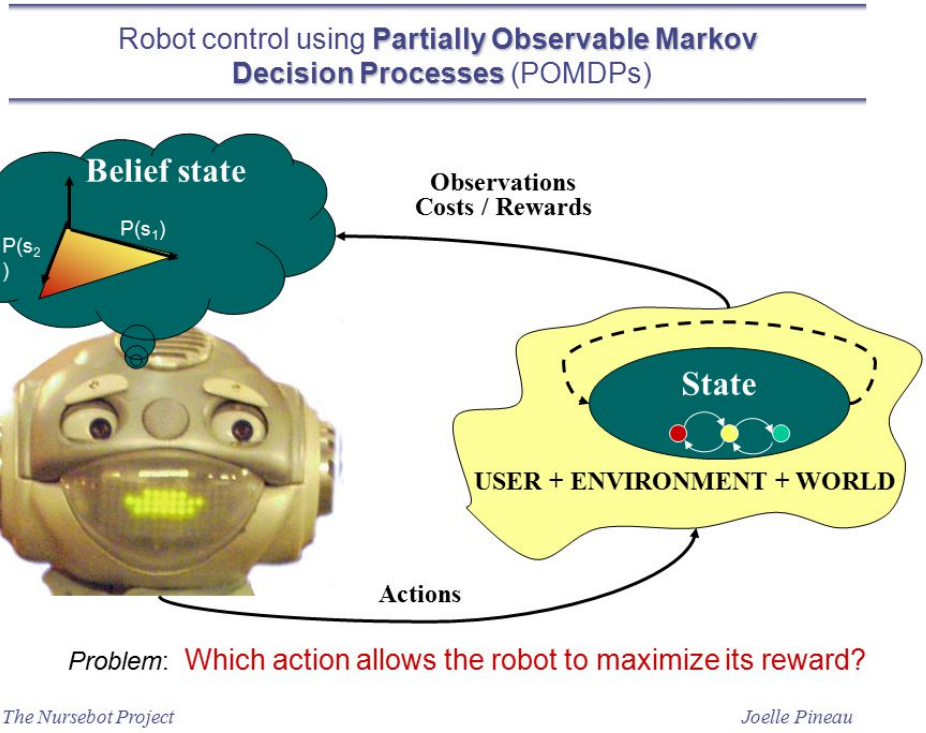
(i). build model of users tastes based on past feedback;

(ii). recommend items to users using this model;

(iii). gather feedback from users about recommended products to refine model in step 1.

Ex 2: Robotics

`http://www.cs.cmu.edu/~ggordon/jpineau-isrr05.pdf`



Ex 3: Biostatistics, Adaptive Modulation, Optimal search, Controlled Sensing, Robotics, Smart Radars, Ad. Placement, Sampling a social network, Learning Systems

For more practical details, see "Success stories in control"

`http://ieeecss.org/general/IoCT2-report`

Big Data: High Dimensional Statistical Inference

1. Classical setting: $x_k \in \mathbb{R}^n$ iid. Then as number of data points $N \to \infty$, (so $n \ll N$) following asymptotics hold:

$$\frac{1}{N}\sum_{k=0}^{N} x_k = \hat{\mu}_N \to \mathbb{E}\{X_k\} = \mu \quad \text{(SLLN)}$$

$$\frac{1}{\sqrt{N}}\sum_{k=0}^{N} (x_k - \mu) \to N(0, \Sigma) \quad \text{(CLT)}$$

Big Data: $n = N$ or $n > N$. Asymptotic statistics dont apply.

Important Questions: 1. *How to mathematically guarantee performance?* Concentration of measure (finite N analysis). *Hoeffding inequality.* $X_k \in [a, b]$ iid. Then for any $N > 0$,

$$\boxed{P(|\hat{\mu}_N - \mu| > \epsilon) \leq 2\exp\left(-\frac{2N\epsilon^2}{(b-a)^2}\right)}$$

2. *How to exploit sparsity?* Suppose parameter dimension $n = 10^8$, number of data points $N = 10^5$. But only 5 parameters in parameter are non-zero.

Netflix problem: Suppose matrix has missing entries but is rank sparse. How to reconstruct matrix?

3. Some other fascinating properties of Big Data.

Line of length 1: 0.9 length lies in interval $[0, 0.9]$
Square of area 1: 0.9^2 area lies in interval $[0, 0.9]^2$.
Cube of area 1: 0.9^3 volume lies in interval $[0, 0.9]^3$.
As dimension increases, most volume lies close to surface!
4. Volume of unit sphere goes to zero as dimension increases
5. For multi-dimension Gaussian, most of mass lies in tails.

6. The deep mathematics for big-data comes from "Probability in Banach Spaces" (which we won't cover)
Dvoretsky's theorem: Consider a symmetric convex body S. Take a subspace V. Then $V \cap S$ looks like a sphere (or ellipsoid) in high dimensions. (2 dim slice thru 3 dim cube is not an ellipsoid. But as the dim of the S and V increase gets closer to ellipsoid.)

Big Data sets: Crowdsourcing data (Cornell Ornithology), Consumer preference data, Images & videos, Biotech data (DNA microarrays), social network data

Outline

- Part 1. Things to know. Stochastic Processes, statistical inference, simulation
- Part 2. Least Squares, PCA, Sparsity
- Part 3. Bayesian Filtering
- Part 4. Maximum Likelihood Estimation
- Part 5. Stochastic Optimization

Regarding this course: Measure-theoretic probability is not used. Everything is in discrete time.

Continuous-time stochastic calculus is technically more difficult

Part I. Things to Know

1. Probabilistic Models
2. Random Variables
3. Elementary Stochastic Simulation
4. Random Processes (IID & Markov Chains)
5. Statistical Inference & Stochastic Convergence
6. Stochastic Difference Equations
7. Linear Time Series Models

Elementary logic:

$$a \implies b$$

a is a sufficient condition for b.

b is necessary condition for a.

$a \implies b$ does not imply $b \implies a$.

If $a \implies b$ and $b \implies a$, then $a \equiv b$ (iff)

$a \implies b \equiv \neg b \implies \neg a$.

Any mathematical definition is "iff"

Denumerable Sets and Continuum

Examples of denumerable sets (countable infinity $\aleph_0$):
1. $\mathcal{Z} = \{\ldots, -1, 0, 1 \ldots\}$: set of integers.
2. $\mathcal{Q} = \{a/b,\ a \in \mathcal{Z}, b \in \mathcal{Z}, b \neq 0\}$.

The cardinality of $\mathbb{R}$ is denoted c – uncountable infinity.
Examples: Closed interval $[0, 1]$; open interval (a, b) where $a, b \in \mathbb{R}$; unions of open intervals, etc.
c is much larger than $\aleph_0$. Roughly speaking $\aleph_0/c = 0$!

Given finite set $A = \{a_1, \ldots, a_n\}$, the power set denoted 2^A comprises of all subsets of A and the empty set $\emptyset$.
$\text{card}(2^A) = 2^n$.
Example: Given $A = \{1, 2\}$, $2^A = \{\{1\}, \{2\}, \{1, 2\}, \emptyset\}$.

Cantor's Theorem: $2^{\aleph_0} = c$.

What is 2^c?? Contains several types of bizarre elements.
Only want those subsets of 2^c that are closed under complementation and countable unions: *measurable sets*.

Probability theory $\subset$ Measure Theory
We wont cover/use "Measure Theoretic Probability" .

Part I.1. Random Variables

Model for Probabilistic Experiment: $(\Omega, \mathcal{F}, P)$.

1. For finite Ω, suffices to choose $\mathcal{F} = 2^{\Omega}$ (powerset)
2. For denumerable Ω, event space $\mathcal{F} = 2^{\Omega} = c$.
3. $\Omega = \mathbb{R}$: $2^{\Omega} = \{\text{measurable sets, non measurable sets}\}$
   Can define prob measure only on *measurable sets*.
   Event space σ-algebra: $\mathcal{F}$ = measurable sets: closed under complement, countable unions and intersections.

Result: If $\Omega = \mathbb{R}$, all elements of $\mathcal{F}$ (events = measurable sets) are obtained as complement, countable unions and intersections of $(-\infty, x]$, $x \in \mathbb{R}$. Then $\mathcal{F}$: Borel algebra.

Probability measure $P : \mathcal{F} \to [0,1]$: $P(\Omega) = 1$, $P(\emptyset) = 0$. For $A_i \in \mathcal{F}$, s.t. $A_i \cap A_j = \emptyset$, $P(\cup_{i=1}^{\infty} A_i) = \sum_{i=1}^{\infty} P(A_i)$ "countably additive normalized set-function (measure)"

Random Variable $X(\omega)$: $X : \Omega \to \mathbb{R}$ is real valued rv if $\{\omega : X(\omega) \leq x\} \in \mathcal{F}$ for all $x \in \mathbb{R}$.

Main outcome. Sufficient to define CDF

$$P((-\infty, x]) = F_X(x), \quad x \in \mathbb{R}$$

From CDF, can compute prob of any measurable event.

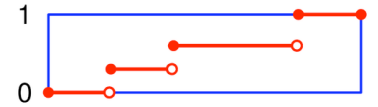
Trivia. Non-measurable sets require axiom of choice.

Strange results: Banach Tarski paradox. Pollard: User's Guide to Measure Theoretic Probability; Billingsley: Probability & Measure.

I.1. Things to know

CDF $F_X(x) = P(X(w) \leq x)$, $x \in \mathbb{R}$ is continuous on the right with limit on the left:

$$\lim_{\epsilon \downarrow 0} F_X(x+\epsilon) = F_X(x)$$



Example of cdf $X \in \{1, 2, 4\}$

Multivariate Random Variable $X(w) \in \mathbb{R}^n$. Then CDF is

$$F_X(x) = P(X_1(w) \leq x_1, X_2(w) \leq x_2, \ldots, X_n(w) \leq x_n)$$

$$n\text{-variate PDF is } f_X(x) = \frac{\partial^n}{\partial x_1, \ldots, \partial x_n} F_X(x)$$

Example: Consider two rvs X and Y. Then

$$F_{X,Y}(x,y) = P(X \leq x, Y \leq y),$$

$$f_{X,Y}(x,y) = \frac{\partial^2}{\partial x \partial y} F_{X,Y}(x,y)$$

Independence: The σ-algebras $\mathcal{B}_1, \mathcal{B}_2, \ldots, \mathcal{B}_n$ are indpt if for any sets $A_i \in \mathcal{B}_i$, $i = 1, \ldots, n$, $P(\cap_i A_i) = \prod_i P(A_i)$.
Events $A_1, \ldots . A_n$ are mutually indpt if σ-algebras $\mathcal{B}_1, \ldots, \mathcal{B}_n$ are indpt where $\mathcal{B}_i = \{\emptyset, \Omega, A_i, A_i^c\}$.
Equivalently: Events $A_1, \ldots . A_n$ are mutually indpt if

$$P(\cap_{i=1}^n A_i^*) = \prod_{i=1}^n P(A_i^*)$$

for all possible choices of A_i^* as either A_i or $A_i^c = \Omega - A_i$.

RVs $X_1, \ldots, X_n$ are indpt if

$$F_{X_1,\ldots,X_n}(x_1, \ldots, x_n) = F_{X_1}(x_1) \cdots F_{X_n}(x_n)$$

Then

1. $p_{X_1,\ldots,X_n}(x_1, \ldots, x_n) = p_{X_1}(x_1) \cdots p_{X_n}(x_n)$.
2. $F_{X_j|\{X_i, i\neq j\}}(x_j|\{x_i, i \neq j\}) = F(X_j)$.

Marginalization: Essential tool in Bayesian inference

- Given $f_{X,Y}(x, y)$ how to compute $f_X(x)$:
  Ans: $f_X(x) = \int_{-\infty}^{\infty} f_{X,Y}(x, y)dy$ **(marginal density)**
- Given $F_{X,Y}(x, y)$ how to compute $F_X(x)$:
  Ans: $F_X(x) = F_{X,Y}(x, \infty)$

Example. $$f_{X,Y}(x, y) = \begin{cases} 2e^{-x}e^{-y} & 0 \leq y \leq x < \infty \\ 0 & \text{elsewhere} \end{cases}$$

Compute the marginals and show that X and Y are dependent.
Soln:

$$f_X(x) = \int_0^{\infty} f_{X,Y}(x, y)dy = 2e^{-x}(1 - e^{-x}), \quad 0 \leq x < \infty$$

$$f_Y(y) = \int_0^{\infty} f_{X,Y}(x, y)dx = 2e^{-2y}, \quad 0 \leq y < \infty$$

Statistics of Vector Random Variables

1. Expected Value: $X(w) = (X_1(w), \ldots, X_n(w))' \in \mathbb{R}^n$ with multivariate pdf $f_X(x)$ where $x = (x_1, \ldots, x_n)$

$$\mathbb{E}\{X\} = \Big[\mathbb{E}\{X_1\} \cdots \mathbb{E}\{X_n\}\Big]', \ \mathbb{E}\{X_i\} = \int_{-\infty}^{\infty} x_i f_{X_i}(x_i) dx_i$$

$$\text{where marginal } f_{X_1}(x_1) = \int f_X(x_1, \ldots, x_n) dx_2 \cdots dx_n$$

Note that $\mathbb{E}\{X\} \in \mathbb{R}^n$ is deterministic (non-random).

Remarks:

1. A more useful/rigorous defn involves Lebesgue integrals
2. $\mathbb{E}\{g(X)\} = \int_{\mathbb{R}}^n g(x) f_X(x) dx$
3. *Linearity*: $\mathbb{E}\{aX + bY\} = a\mathbb{E}\{X\} + b\mathbb{E}\{Y\}$

2. Covariance: $X \in \mathbb{R}^n$, with $\mathbb{E}\{X\} = \mu \in \mathbb{R}^n$. Then

$$\text{cov}(X) = \mathbb{E}\{(X - \mu)(X - \mu)'\} \in \mathbb{R}^{n \times n}$$

symmetric positive semi-definite matrix.

Defn: Symmetric matrix $A \in \mathbb{R}^{n \times n}$ is *positive semidefinite* if for all $x \in \mathbb{R}^n \neq 0$, $x'Ax \geq 0$.

Symmetric matrix $A \in \mathbb{R}^{n \times n}$ is *positive definite* (pd) if for all $x \in \mathbb{R}^n \neq 0$, $x'Ax > 0$.

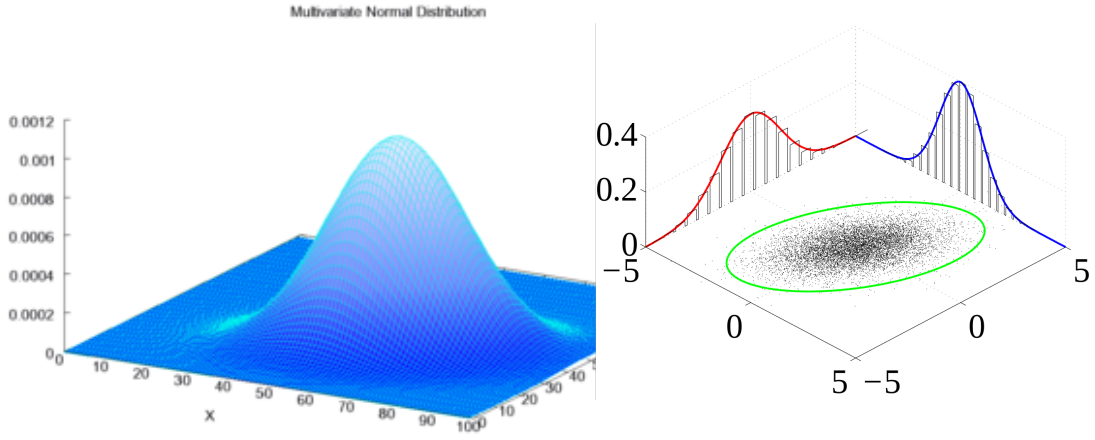
Prove that A is pd: (i) Iff $\lambda_i(A) > 0, \quad \forall i$. (ii) Iff A^{-1} is pd. (iii) All eigenvectors are orthogonal (since symmetric).

3. $\text{var}(X) = \text{trace}(\text{cov}(X)) = \mathbb{E}\{(X - \mu)'(X - \mu)\} \in \mathbb{R}_+$.
Standard deviation = Square root of $\text{var}(X)$.

Example 1. Multivariate Gaussian. An n-dimensional Gaussian random variable $X \sim N(\mu, \Sigma)$ has pdf

$$\frac{1}{(2\pi)^{n/2}|\Sigma|^{1/2}} \exp\Big(-\frac{1}{2}(x-\mu)'\Sigma^{-1}(x-\mu)\Big), \quad x \in \mathbb{R}^n$$

where $\mu \in \mathbb{R}^n$ denotes the mean and Σ is an $n \times n$ symmetric positive definite matrix denoting the covariance.



Gaussian pdf is bell-shaped curve symmetric about μ.

1. Show that Gaussian density integrates to 1.
2. Show that $\mathbb{E}\{X\} = \mu$, $\text{cov}(X) = \Sigma$.
3. Show that if Σ is diagonal, then the individual components $x_1, x_2, \ldots, x_n$ are independent random variables.
4. Show that the individual components x_i are univariate Gaussians. Therefore $(x_1, \ldots, x_n)$ jointly Gaussian implies that the marginals are Gaussian.
5. However, x_1, x_2 being individually Gaussian does not imply that (x_1, x_2) is jointly Gaussian. Give an example.

Gaussians are widely used because:

Result 1: Gaussian input to linear system yields Gaussian output.

Example: If $Y = AX + b$, $A \in \mathbb{R}^{n\times n}$ and $X \sim N(\mu_n, \Sigma_{n\times n})$, then $Y \sim N(A\mu + b, A\Sigma A')$.

Because $\mathbb{E}\{\cdot\}$ is linear, $\mathbb{E}\{Y\} = \mathbb{E}\{AX + b\} = A\mathbb{E}\{X\} + b$.
Similarly $\text{cov}\{Y\} = A\Sigma A'$ is easy to show.
Scalar case: $Y = aX + b$, let us compute $f_Y(y)$?

First compute cdf F_Y of Y. Assume $a > 0$. For any ζ

$$F_Y(\zeta) = P(Y \le \zeta) = P(aX+b \le \zeta) = P(X \le \frac{\zeta - b}{a}) = F_X\left(\frac{\zeta - b}{a}\right)$$

Take derivatives of both sides wrt ζ yields pdf:

$$f_Y(\zeta) = \frac{d}{d\zeta} F_Y(\zeta) = \frac{1}{a} f_X\left(\frac{\zeta - b}{a}\right)$$

For $a < 0$, pdf is $f_Y(\zeta) = \frac{1}{-a} f_X\left(\frac{\zeta-b}{a}\right)$. So in general for any a

$$\boxed{f_Y(\zeta) = \frac{1}{|a|} f_X\left(\frac{\zeta - b}{a}\right)}$$

So if X is $N(\mu, \sigma^2)$, then Y is $N(a\mu + b, a^2\sigma^2)$.

Result 2: Central Limit Theorem (see later)

Exercise (Gaussian tails): If $X \sim \boldsymbol{N}(0, 1)$, then

$$P(|X| \ge x) \le e^{-x^2/2}$$

Example 2. Exponential Random Variable

$$\text{CDF: } F_X(x) = P(X \le x) = 1 - e^{-\lambda x}, \quad x \ge 0$$

$\lambda > 0$: rate parameter (number of events per unit time).

$$\text{PDF: } f_X(x) = \lambda e^{-\lambda x}, \; x \ge 0$$

$$\mathbb{E}\{X\} = \int_0^\infty x f_X(x) dx = \frac{1}{\lambda}, \quad \text{var}\{X\} = \frac{1}{\lambda^2}$$

So the average time interval between events is $1/\lambda$.
Memoryless property of exponential pdf:

$$P(X > s + t | X > t) = P(X > s)$$

That is there is no memory of t.
Example 1: If the reliability of a car was exponential distributed, then used car is as good as brand new car.
Example 2: Suppose $1/\lambda = 10$, i.e., a bus comes every 10 mins on average. If you have already waited 9 mins, prob that you need to wait more than 1 additional minute for the bus to come, is the same as if you have not waited for 9 mins.

Proof: $P(X > s + t | X > t) = \dfrac{P(X > s + t \text{ and } X > t)}{P(X > t)}$

$$= \frac{P(X > s + t)}{P(X > t)} = \frac{e^{-\lambda(s+t)}}{e^{-\lambda t}} = e^{-\lambda s} = P(X > s)$$

1. Poisson rv $P(X=k) = \frac{\lambda^k e^{-\lambda}}{k!}$ has exponential waiting times.
2. $X_1, \ldots, X_n$ indpt exponential rv. Then $\min\{X_1, \ldots, X_n\}$ is exponential with $\lambda = \sum_i \lambda_i$. But max is not exponentially distributed.
3. Amongst all pdfs with support on $\mathbb{R}_+$ with finite mean μ, exponential density has largest differential entropy:

$$\left.\begin{array}{l} \text{argmax}_p - \int p(x) \log p(x) dx, \\ \int_{\mathbb{R}_+} x p(x) dx = \mu, \quad x \in \mathbb{R}_+ \end{array}\right\} = \lambda e^{-\lambda x}, \quad \lambda = 1/\mu.$$

4. Amongst all pdfs with support on $\mathbb{R}$ with finite power σ^2, Gaussian density has largest differential entropy:

$$\left.\begin{array}{l} \text{argmax}_p - \int p(x) \log p(x) dx, \\ \int_{\mathbb{R}_+} x^2 p(x) dx = \sigma^2, \quad x \in \mathbb{R} \end{array}\right\} = \boldsymbol{N}(\mu, \sigma^2).$$

(Recall differential entropy does not depend on μ).

Result: If X, Y are indpt rvs then pdf of $Z = X + Y$ is

$$p_Z(z) = p_X \otimes p_Y = \int_{-\infty}^{\infty} p_X(z-t)\, p_Y(t)\, dt$$

where $\otimes$ denotes convolution

If X, Y are indpt Gaussian rvs, $X + Y$ is Gaussian.
(Convolution of Gaussian densities is Gaussian).

Are there other pdfs which are closed under convolution?

Stable Distributions (Advanced)

Stable. Suppose $X \sim p$ and $Y \sim p$ are indpt. Then p is stable if $X + Y \sim p$. (Recall pdf of $X + Y$ is convolution of pdfs).

Ex1. Gaussian. (so convolution of Gaussian pdfs is Gaussian).
Ex2. Cauchy (Lorentz) density: $p(x) = \frac{a}{\pi} \frac{1}{a^2+x^2}$
General class of zero mean, symmetric α-stable Levy distributions:

$$\text{characteristic fn: } \phi(w) = \mathbf{F}(p(x)) = e^{-\gamma|w|^\alpha}, \quad \alpha \in (0, 2]$$

$$\text{pdf: } p(x) = \mathbf{F}^{-1}[\phi(w)] = \frac{1}{\pi} \int_0^\infty e^{-\gamma|w|^\alpha} \cos(wx) dw$$

Widely used: finance, economics, internet traffic, network science.
α-stable distributions have remarkable properties:

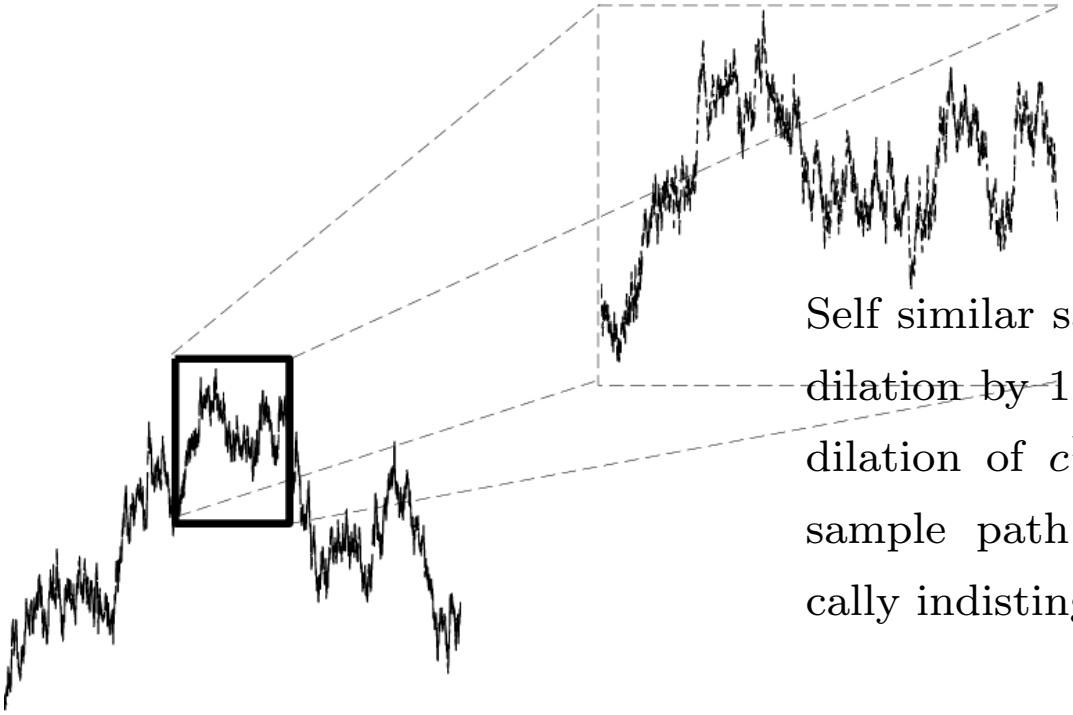
1. Have infinite variance (except for Gaussian $\alpha = 2$).
   Cauchy $\alpha = 1$ has infinite mean.
2. Exhibit power law density: For $\alpha \in (1, 2)$ and large $|x|$,

   $$p(x) \propto |x|^{-(1+\alpha)} \qquad \text{power law density}$$

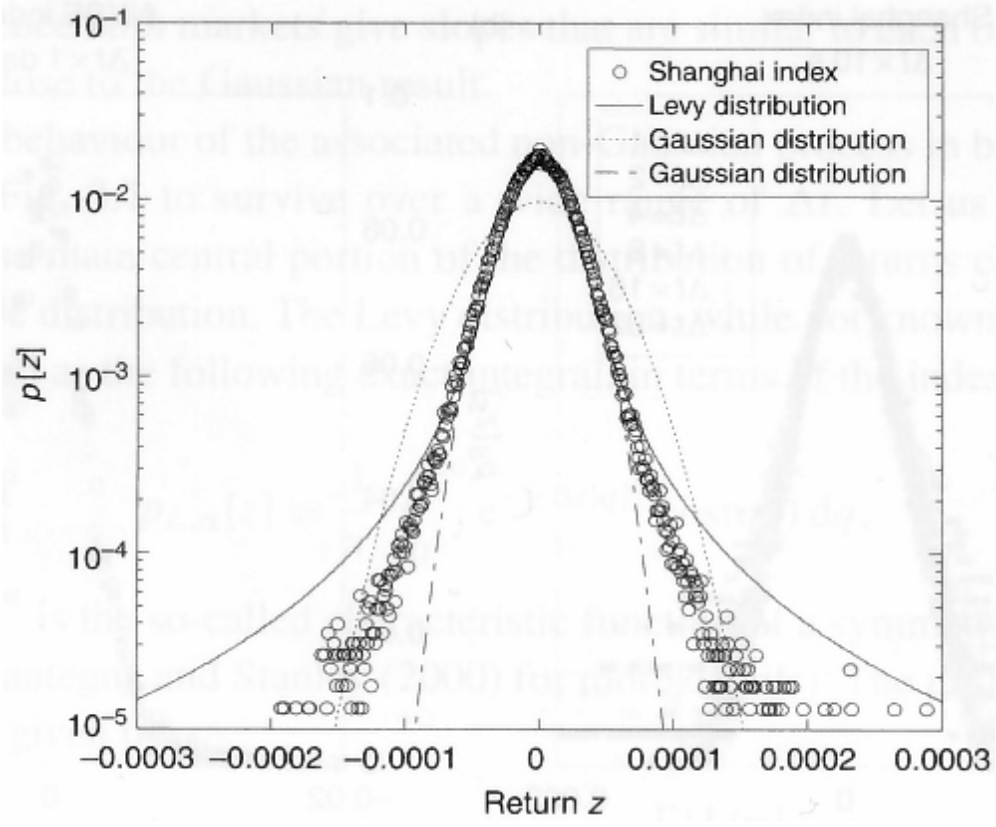
   Examples of power law densities: Pareto law 80-20 rule; degree distribution of social networks, etc.
3. Clearly $\phi^n(w) = \phi(n^{1/\alpha} w)$, $\alpha \in (0, 2]$ for all n.
4. *Self similarity*: Let $S_n = \sum_{i=1}^n x_i$ where x_i are iid. Then S_n is self similar if $p(S_{cn} = s) = c^{1/\alpha} p(S_n = s)$.

Theorem: α-stable pdf $\equiv$ self similar.
Proof of $\Longrightarrow$: $\phi_{cn}(w) = [\phi_i(w)]^{cn} = \phi_1^n(c^{1/\alpha} w) = \phi_n(c^{1/\alpha} w)$.

Summary. α-Stable Levy pdfs: power law and self similarity.



Self similar sample path: time dilation by $1/c$ and amplitude dilation of $c^{1/\alpha}$ yields a new sample path that is statistically indistinguishable.



$Y(k)$: time series of index
$Z(k) = Y(k+1) - Y(k)$;
time step $\Delta t = 1$ min.

1. almost symmetric
2. Non-Gaussian for small changes and time intervals from 1 to 10^4 trading minutes.
3. NYSE, NASDAQ: pdf for change x has power law with $\alpha > 2$.

PDF of returns for high frequency trading Shanghai index price changes with stable symmetric Levy distribution $\alpha = 1.44$.
Johnson, Jefferies and Hui (2003)

Infinitely Divisible Distributions

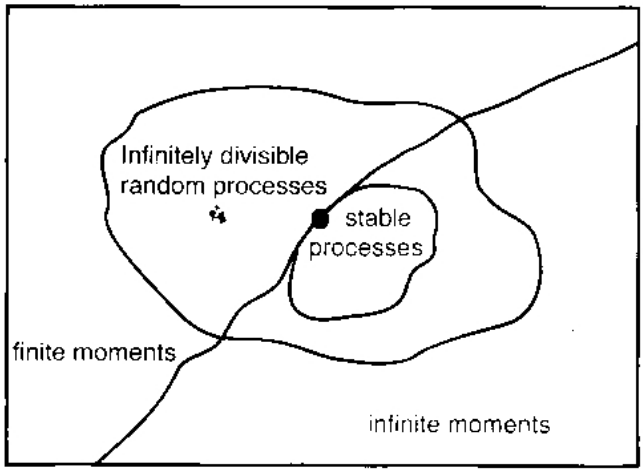
A distribution F is infinitely divisible if for every positive integer k, there exist iid rvs $X_{k,1}, \ldots, X_{k,k}$ such that $S_k = X_{k,1} + \cdots + X_{k,k}$ has cdf F.
Equivalently for every k, there exists pdf p_k s.t.

$$p = p_k \otimes p_k \cdots \otimes p_k$$

Examples: Gaussian, Poisson, Gamma, Exponential, Geometric.
Proof. Check that $(\phi_X(t))^{1/k}$ is a valid characteristic function.

1. Every stable distribution is infinitely divisible.
2. Uniform, binomial are not infinitely divisible.

A pdf with bounded range is not infinitely divisible (unless delta function, i.e. rv is deterministic).



Exponential family

Normal, Bernoulli, Gamma, Chi-Squared, Beta, Binomial, Poisson, ...

$$\boxed{p_\theta(x) = \exp\big(\theta' T(x) - A(\theta)\big)\, h(x)}, \quad x \in \mathbb{R}^n$$

θ: vector of natural parameters, $T(x)$ sufficient statistic;
$A(\theta)$: log partition function = log of normalization factor;
convex in θ.
Note: x and θ only interact in $\theta' T(x)$.
Examples:(i) Multivariate Normal: $X \sim N(\mu, \Sigma)$ then $x \in \mathbb{R}^n$

$$p_\theta(x) = \frac{1}{(2\pi)^{n/2}|\Sigma|^{1/2}} \exp\big(-\frac{1}{2}(x-\mu)'\Sigma^{-1}(x-\mu)\big), \quad \Sigma \succeq 0$$

$$h(x) = (2\pi)^{-n/2},\ T(x) = \begin{bmatrix} x \\ xx' \end{bmatrix},\ \theta = \begin{bmatrix} \Sigma^{-1}\mu \\ -\frac{1}{2}\Sigma^{-1} \end{bmatrix}.$$

(ii) Bernoulli: $p_\theta(x) = \alpha^x (1-\alpha)^{1-x}$
Then $T(x) = x$, $\theta = \log \frac{\alpha}{1-\alpha}$, $A(\theta) = \log(1 + e^\theta)$.

Properties of Exponential Family.

1. Exponential family pdfs are closed under multiplication.

$$\prod_{i=1}^n \exp\big(\theta_i' T(x) - A(\theta_i)\big)\, h(x) = \exp\Big(\sum_{i=1}^n \theta_i' T(x) - \tilde{A}(\theta_1, \ldots, \theta_n)\Big)\, \tilde{h}(x)$$

Product of Gaussian densities is Gaussian.

2. $\nabla_\theta A(\theta) = \mathbb{E}_{p_\theta}\{T(X)\}$ and $\nabla^2_\theta A(\theta) = \mathrm{cov}_{p_\theta}\{T(X)\} \succeq 0$.
Therefore $A(\theta)$ is convex.

Recall exponential family $p_\theta(x) = \exp\big(\theta' T(x) - A(\theta)\big)\, h(x)$

3. Stein' Identity. Useful for computing moments. If X belongs to exponential family, and g is differentiable, then

$$\mathbb{E}\Big\{ \big[\frac{\nabla h(X)}{h(X)} + \theta' \nabla T(X)\big] g(X) \Big\} = -\mathbb{E}\{\nabla g(X)\}$$

Example. $X \sim \boldsymbol{N}(0,1) \implies \mathbb{E}\{g(X)(X-\mu)\} = \sigma^2 \mathbb{E}\{\nabla g(X)\}$

So $g(x) = 1$ implies $\mathbb{E}\{X\} = \mu$,
$g(x) = x$ implies $\mathbb{E}\{X^2\} = \sigma^2 + \mu^2$.

4. Bayesian Conjugate Priors (see later)

Are Uniform, Exponentials and Gaussians related?

Result: Suppose $\Theta \sim U(0, 2\pi)$ (uniform pdf) and
$R \sim \lambda e^{-\lambda r}$ (exponential pdf with $\lambda = 1/2$) are indpt rvs.
Then $X = \sqrt{R}\cos\Theta$ and $Y = \sqrt{R}\sin\Theta$ are indpt $N(0,1)$ rvs.

Proof: Because R and Θ are indpt, joint pdf is

$$f_{R,\theta}(r,\theta) = \frac{1}{2} e^{-r/2} \frac{1}{2\pi}, \quad 0 \le r < \infty, \quad 0 \le \theta \le 2\pi$$

Recall function of rv formula: $(X,Y) = g(R,\theta)$ implies pdf

$$\boxed{f_{X,Y}(x,y) = f_{R,\Theta}\left(g^{-1}(X,Y)\right) \times J}$$

$$\text{where Jacobian } J = \text{determinant}\left[\frac{d}{d[X,Y]} g^{-1}(X,Y)\right]$$

Here $(R,\theta) = g^{-1}(X,Y)$ is $R = X^2 + Y^2$, $\theta = \tan^{-1}(Y/X)$
Exercise: Show that Jacobian $J = 2$ and then result follows.

$$f_{X,Y}(x,y) = \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{1}{2}x^2\right) \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{1}{2}y^2\right)$$

Gaussians, Expo, Chi-Squared: $X \sim \boldsymbol{N}(0, I_n)$, then $\|X\|^2 \sim \chi^2(n)$; chi-squared with n degrees of freedom.
$Z \sim \chi^2(2)$ is exponential density; $Z^2 \sim \chi^2(2)$, then Z has Rayleigh density.

Two famous Probabilistic Inequalities.
Markov Inequality: For $\phi : \mathbb{R} \to \mathbb{R}_+$ increasing and $X \in \mathbb{R}$

$$P(X \geq t) \leq \frac{1}{\phi(t)} \mathbb{E}\{\phi(X)\}$$

Proof: $P(X \geq t) \leq \mathbb{E}\{\frac{\phi(X)}{\phi(t)} \mathbf{1}_{X \geq t}\} \leq \frac{1}{\phi(t)} \mathbb{E}\{\phi(X)\}$
Example: Chebyshev ineq: $P(|X - \mu| \geq t) \leq \mathbb{E}|X - \mu|^2 / t^2$.
Used with Borel Cantelli to prove almost sure convergence.

Jensen's Inequality: For convex $\phi : \mathbb{R}^n \to \mathbb{R}$ and $X \in \mathbb{R}^n$,

$$\phi(\mathbb{E}\{X\}) \leq \mathbb{E}\{\phi(X)\}$$

Example 1. $\mathbb{E}^2\{X\} \leq \mathbb{E}\{X^2\}$. So $\mathrm{Var}(X) \geq 0$. Also finite variance implies finite mean.
Example 2. $\mathbb{E}\{\log X\} \leq \log \mathbb{E}\{X\}$ since log is concave.
KL Divergence $D(p||q) = \mathbb{E}_p\{\log \frac{p}{q}\} \geq 0$.
Example 3. $\max_i \mathbb{E}\{X_i\} \leq \mathbb{E}\{\max_i X_i\}$.
Example 4. Geometric mean is smaller than arithmetic mean

$$(x_1 x_2 \ldots, x_n)^{1/n} \leq \frac{x_1 + x_2 + \ldots + x_n}{n}$$

Example 5. Cauchy Schwartz: $|\mathbb{E}\{XY\}|^2 \leq \mathbb{E}\{X^2\}\mathbb{E}\{Y^2\}$

$$\text{correlation coeff } \rho_{XY} = \frac{\mathrm{cov}(X, Y)}{\big(\mathrm{Var}(X)\,\mathrm{Var}(Y)\big)^{1/2}} \in [-1, 1]$$

Convex Functions

Definition (i) $f : \mathbb{R}^d \to \mathbb{R}$ is convex if for all $x_1, x_2 \in \mathbb{R}^d$,

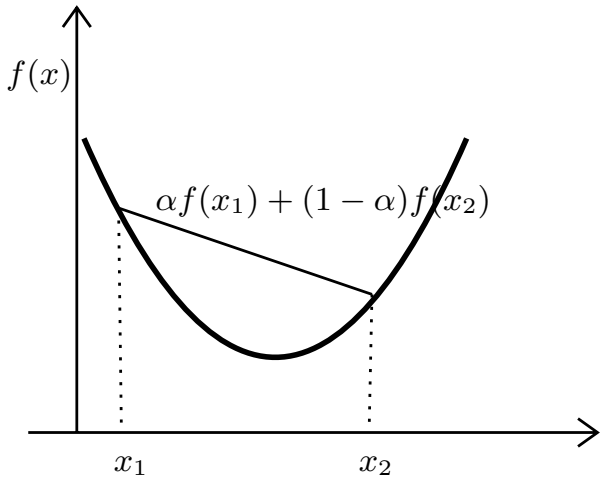
$$f(\alpha x_1 + (1-\alpha)x_2) \leq \alpha f(x_1) + (1-\alpha) f(x_2) \text{ for } \alpha \in [0,1]$$

(ii) Differentiable function f is convex if

$$f(y) \geq f(x) + (y-x)' \nabla f(x), \quad \text{for all } x, y \in \mathbb{R}^d$$

(iii) Twice differentiable function f is convex if Hessian matrix $\nabla^2 f(x)$ is positive semidefinite for all $x \in \mathbb{R}^d$.

Examples of convex functions: Ax, e^x, $-\log(x)$, $x'Qx$ where Q is positive definite, etc.



Proof of Jensen's Inequality:

Convexity implies $f(X) \geq f(a) + (X-a)\nabla f(a)$.

Choose $a = E\{X\}$. Then

$$f(X) \geq f(\mathbb{E}\{X\}) + (X - \mathbb{E}\{X\})\, \nabla f(\mathbb{E}\{X\})$$

$$\implies \mathbb{E}\{f(X)\} \geq \mathbb{E}\{f(\mathbb{E}\{X\})\} + \underbrace{\mathbb{E}\{(X - \mathbb{E}\{X\})\}}_{0}\, \nabla f(\mathbb{E}\{X\})$$

Part I.2. Stochastic Simulation: Scalar RV

$U[0,1]$ uniform pdf with support on $[0,1]$. Matlab `rand(n)` generates an $n \times n$ $U[0,1]$ matrix.

Aim: Given $U[0,1]$ random numbers, generate samples of random variables/processes with specified distributions.

Why? Prototyping; Discrete optimization; Model Validation. In this course: Bayesian inference and Monte-Carlo for computing multidimensional integrals efficiently. Given function $\phi : \mathbb{R}^X \to \mathbb{R}$, let $p(\cdot)$ denote pdf on $\mathbb{R}^X$. Then

$$\int_{\mathbb{R}^X} \phi(x)dx = \int_{\mathbb{R}^X} \frac{\phi(x)}{p(x)} p(x)dx = \mathbb{E}_p \left\{ \frac{\phi(x)}{p(x)} \right\}$$

Simulating iid samples $\{x_k\}$, $k = 1, \ldots, N$, from the pdf $p(\cdot)$, Monte-Carlo estimates integral as

$$\boxed{\frac{1}{N} \sum_{k=1}^{N} \frac{\phi(x_k)}{p(x_k)} \text{ where } x_k \sim p(x)} \to \int_{\mathbb{R}^X} \phi(x)dx \text{ as } N \to \infty \text{ w.p.1}$$

Classical Monte-Carlo: iid samples $\{x_k\}$ Markov-chain Monte-Carlo (MCMC): $\{x_k\}$ geometrically ergodic Markov chain with stationary distribution $p(\cdot)$.

Simulation of Random Variables

Given $U[0,1]$ random numbers, three elementary methods:

(i) Inverse Transform Method Aim: Generate rv $x \sim F$.

Step 1: Generate $u \sim U[0,1]$.
Step 2: Generate $x = F^{-1}(u)$.

Define $F^{-1}(u) = \min\{x : F(x) = u\}$ if $F^{-1}(\cdot)$ is not unique.

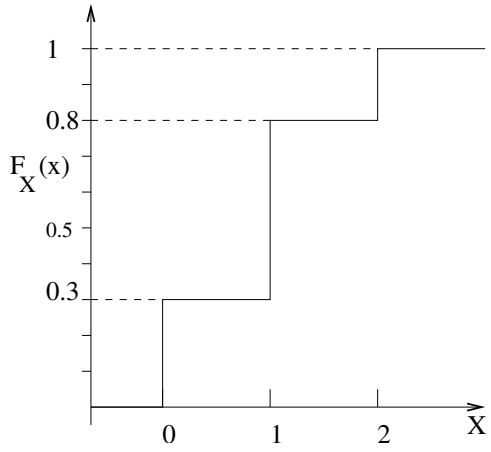
Therefore, if p_i, $i = 1, \ldots, m$ is a probability mass function, then the inverse transform method generates $x \sim p$ as

1. Generate $u \sim U[0,1]$.
2. Generate $x = l^* = \min\{l : u \leq \sum_{k=1}^{l} p_i\}$.

Example 0: Given $U[0,1]$ generator, generate discrete rv X with $P(X=0) = 0.3$, $P(X=1) = 0.5$, $P(X=2) = 0.2$.

Solution: Generate $u \sim U[0,1]$.

$$\text{Set } X = \begin{cases} 0 & \text{if } u < 0.3 \\ 1 & \text{if } 0.3 \leq u < 0.8 \\ 2 & \text{otherwise} \end{cases}$$



Example 1: Generate rv with cdf $F(x) = x^n$, $0 \leq x \leq 1$.
Soln: $u = F(x) = x^n$ or equivalently, $x = u^{1/n}$. So:
(i) generate rv $u \sim U[0,1]$.
(ii) Compute $u^{1/n}$. This has distribution $F(x)$.

Example 2: Exponentially distributed. $F(x) = 1 - e^{-\lambda x}$, $x \geq 0$, $\lambda > 0$ can be generated as $x = -\frac{1}{\lambda}\log(1-u)$.
Example 3: Generate normal random variables as follows. If $\Theta \sim U[0, 2\pi]$ (uniform pdf) and $R \sim \lambda e^{-\lambda r}$ (exponential pdf) with $\lambda = 1/2$ are independent random variables, then it can be shown that $X = \sqrt{R}\cos\Theta$ and $Y = \sqrt{R}\sin\Theta$ are independent $\boldsymbol{N}(0,1)$ random variables. So if $u_1, u_2 \sim U[0,1]$ are independent, then $x = \sqrt{-2\log u_1}\cos(2\pi u_2)$ and $Y = \sqrt{-2\log u_1}\sin(2\pi u_2)$ are independent $\boldsymbol{N}(0,1)$ random variables.

Example 4: To generate discrete rv, Step 2 comprises of $m-1$ `if` statements and can be inefficient in run time execution if m is large. However, for *discrete uniform mass function* can be implemented efficiently. In this case $p_i = 1/m$ for all $i = 1, 2, \ldots, m$. Then, the above method yields

$$x = l \text{ if } \frac{l-1}{m} \leq u < \frac{l}{m} \text{ or equivalently } x = \text{Int}(mu) + 1. \quad (1)$$

Proof. Let $\bar{F}$ denote the cdf of x generated by the algorithm:

$$\bar{F}(\zeta) = \mathbb{P}\{x \leq \zeta\} = \mathbb{P}\{F^{-1}(u) \leq \zeta\} = \mathbb{P}\{F(F^{-1}(u)) \leq F(\zeta)\}.$$

since F is a monotone non decreasing and so $\alpha \leq \beta$ is equivalent to $F(\alpha) \leq F(\beta)$. Thus, $\bar{F}(\zeta) = \mathbb{P}\{u \leq F(\zeta)\} = F(\zeta)$ where the last equality follows since u is uniformly distributed in $[0,1]$.

How to check if your simulation is correct?

1. Empirical cdf: $\hat{F}_n(x) = \frac{1}{n}\sum_{k=1}^n I(x_k \le x) \to F(x)$
   Matlab: `[F,z] = ecdf(x)` returns empirical cdf `F` evaluated at grid points `z` using the data in the vector `x`.
   Then `plot(z,F)`
2. Empirical pdf: $p_n(x) = \frac{1}{n}\sum_{k=1}^n I(x_k \in [x-\Delta, x+\Delta))$.
   Matlab: Given $x = [x_1, \ldots, x_n]$, use `hist(x, nbins, 1)`

Simulating Gaussian rv (Box Muller eqns) Recall

Result: If $\Theta \sim U(0, 2\pi)$ (uniform pdf) and
$R \sim \lambda e^{-\lambda r}$ exponential pdf with $\lambda = 1/2$ are indpt rvs
then $X = \sqrt{R}\cos\Theta$ and $Y = \sqrt{R}\sin\Theta$ are indpt $N(0,1)$ rvs.

1. Generate $U_1 \sim U[0,1]$, $U_2 \sim U[0,1]$ independently
2. Set $d = -2\log U_1$ (inverse transform method for expo)
   Set $\theta = 2\pi U_2$. So obviously $\theta \sim U[0, 2\pi]$
3. Set $X = \sqrt{d}\cos\theta = \sqrt{-2\log U_1}\cos(2\pi U_2)$ and
   $Y = \sqrt{d}\sin\theta = \sqrt{-2\log U_1}\sin(2\pi U_2)$

Then $X \sim N(0,1)$ and $Y \sim N(0,1)$ are indpt rvs.

Simulating Cauchy: $p(x) = \frac{1}{\pi(1+x^2)}$, $F(x) = \frac{1}{2} + \arctan(x)/\pi$.
So $F^{-1}(u) = \tan(\pi(u - \frac{1}{2}))$.

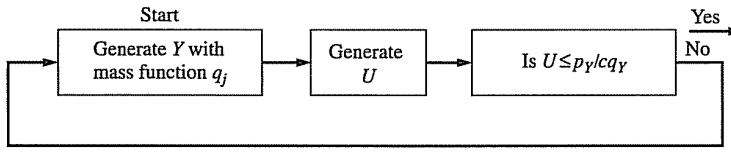
Simulating symmetric α-stable pdfs: $0 \le \alpha \le 2$, $\alpha \ne 1$; see Chambers, Mallows and Stuck, A method for simulating stable rvs, JASA, June 1976.

(ii) Acceptance Rejection Method

Suppose one can generate samples from pdf q. How can random samples be simulated from pdf p? Assume $\max_\zeta \frac{p(\zeta)}{q(\zeta)} < \infty$.

Acceptance Rejection Algorithm Let c denote a constant such that $c \geq \max_\zeta \frac{p(\zeta)}{q(\zeta)}$. Then:

Step 1. Generate $y \sim q$.
Step 2. Generate $u \sim U[0,1]$.
Step 3. If $u < \frac{p(y)}{c\, q(y)}$, set $x = y$.
Otherwise return to step 1.



Example 1. Discrete rv: Want to simulate $Y \in \{1, 2, \ldots, 10\}$ with probs $\{0.11, 0.12, 0.08, 0.12, 0.10, 0.09, 0.10, 0.10\}$.

Generate $q_j = 1/10$, $j = 1, \ldots, 10$. $c = \max p_j/q_j = 1.2$.
Step 1: Generate uniform discrete rv (see previous page).
$U_1 \sim U[0,1]$, $X = \text{Int}(10U_1) + 1$.
Step 2: Generate U
Step 3: If $U < p_X/(cq_X)$, set $Y = X$ and stop. Else goto step 1.

Remarks: (i) Acceptance rejection operates on pdf while inverse transform method operates on cdf.
(ii) Self-normalizing: $p(\cdot)$ does not need to be normalized.
(iii) Clearly $c \geq 1$: $c \geq \max_\zeta \frac{p(\zeta)}{q(\zeta)} \implies c \geq \frac{p(\zeta)}{q(\zeta)} \implies c \geq 1$.
(iv) Expected number of iterations is c. c need not be tight. e.g., $2c$ works.
(v) Matlab code for Acceptance Rejection
https://www.mathworks.com/examples/statistics/mw/

Each iteration independently yields a probability of acceptance of $\frac{1}{c}$. So the number of iterations to accept is a geometric random variable[a] with mean c and variance $c(c-1)$.

Example 2: Generate rv with cdf $F(x) = x^n, \quad 0 \le x \le 1$.
Soln: Choose $q(x) = U[0,1]$. Then

$$\max_{\zeta} \frac{p(\zeta)}{q(\zeta)} = \max_{\zeta \in [0,1]} n\zeta^{n-1} = n$$

So choosing $c = n$, Step 1 and Step 2 generate two independent $U[0,1]$ samples y and u. Step 3 sets $x = y$ if $u < y^{n-1}$.

Example 3: Generate rv with pdf $p(x) = \frac{2}{\sqrt{2\pi}} e^{-x^2/2}, \quad x \ge 0$.
from exponential pdf $q(x) = e^{-x}$, $x \ge 0$.

Why? Once we generate $X \sim p(x)$, then $\pm X \sim N(0,1)$.

$$c = \max_{\zeta} \frac{p(\zeta)}{q(\zeta)} = \max_{\zeta \in \mathbf{R}} \sqrt{\frac{2}{\pi}} e^{\zeta - \zeta^2/2} = \sqrt{\frac{2e}{\pi}}.$$

Step 1: simulate exponentially distributed rv y (use inverse transform method). Step 2: generate uniform rv u.
Step 3: Set $x = y$ if $u \le e^{-(y-1)^2/2}$.

---

[a] Geometric distribution models number of trials to the first success when trials are iid with success probability p: So for $n \ge 1$, the probability of n trials to the first success is $p(1-p)^{n-1}$. The expected value and variance of a geometric random variable are $1/p$ and $\frac{1-p}{p^2}$.
B. D. Flury, *Acceptance-Rejection Sampling Made Easy*, SIAM, 1990.

Proof. $\mathbb{P}(x \le \zeta) = \mathbb{P}(y \le \zeta | u \le \dfrac{p(y)}{c\,q(y)}) = \dfrac{\mathbb{P}\left(y \le \zeta, u \le \frac{p(y)}{c\,q(y)}\right)}{\mathbb{P}\left(u \le \frac{p(y)}{cq(y)}\right)}$

$$= \frac{\text{Prob of } y \le \zeta \text{ and accept}}{\text{Prob of accept}} = \frac{\int_{-\infty}^{\zeta} \int_0^{\frac{p(y)}{cq(y)}} du\, q(y) dy}{\int_{-\infty}^{\infty} \int_0^{\frac{p(y)}{cq(y)}} du\, q(y) dy}$$

$$= \frac{\frac{1}{c}\int_{-\infty}^{\zeta} p(y) dy}{\frac{1}{c}\int_{-\infty}^{\infty} p(y) dy}$$

(iii) Composition Method

Aim: Simulate from convex combination of cdfs:
$F(\zeta) = \sum_{i=1}^n p_i F_i(\zeta)$ where $p_i \ge 0$, and $\sum_{i=1}^n p_i = 1$.

1. Generate the integer random sample $i^* \in \{1, \ldots, n\}$ with probability mass function $p_1, \ldots, p_n$
2. Then generate a random sample x from distribution F_{i^*}.

Continuum convex combination: Suppose one can simulate samples from $x \sim p_{x|y}(x|y)$ and $y \sim p_y(y)$. Then samples from

$$p_x(\zeta) = \int_{\mathbb{R}^m} p_{x|y}(\zeta|y) p_y(y) dy \tag{2}$$

can be simulated via the following composition method:

Step 1: Simulate $y^* \sim p_y(\cdot)$.
Step 2: Simulate $x \sim p_{x|y}(\cdot|y^*)$.

The nice property of the above algorithm is that we do not need to compute the integral in (2) in order to simulate from $p_x(\zeta)$.

Example 1. Simulate from cdf $F(x) = \frac{x+x^3+x^5}{3}$, $0 \le x \le 1$.
Example 2. Simulate from $\int_0^\infty x^y e^{-y} dy$, $0 \le x \le 1$.
Soln. (i) Simulate y^* from pdf e^{-y}.
(ii) Simulate x from cdf x^{y^*}, $0 \le x \le 1$.

Example 3. Randomized linear algebra. How to estimate $p'x$ where p is a probability vector and $x \in \mathbb{R}^n_+$?
Standard computations: $\theta = p'x$ requires $O(n)$ multiplications.
Composition method: Note $p'x = \mathbb{E}_p\{x\}$. For $k = 1, 2, \ldots, N$:

1. Generate $i_k \sim p$. Then $x_{i_k} \sim p'x$.
2. Then for sufficiently large N,

$$\hat{\theta} = \frac{1}{N} \sum_{k=1}^{N} x_{i_k} \to p'x$$

Concentration inequality specifies what N to choose.
Hoeffding inequality. $X_k \in [a, b]$ iid. Then for any $N > 0$,

$$\boxed{P(\Big|\frac{1}{N}\sum_{k=1}^{N} x_k - \mathbb{E}\{X_k\}\Big| > \epsilon) \le 2\exp\left(-\frac{2N\epsilon^2}{(b-a)^2}\right)}$$

E.g. $a = 0, b = 1$, prob $\le 10^{-4}$, $\epsilon = 10^{-3}$: need $N > 5 \times 10^6$.
So if $n = 10^{10}$, randomized method is more efficient.

We will return to simulation for:

1. Multivariate distributions (via MCMC)
2. For Bayesian inference (particle filters).

Part I-3. Discrete Time Random Processes

A random (stochastic) process $X_k(\omega)$, $\omega \in \Omega$ is a family of random variables indexed by discrete time $k = 0, 1, \ldots$.

Remarks: 1. Each outcome $\omega \in \Omega$ yields sample path $X_k(\omega)$.

(i) Fixing w, yields a deterministic function of time: sample path

(ii) Fixing time n yields a random variable

(iii) Fixing both n and w yields a constant

2. Image processing: spatial random processes on poset

A random process is completely characterized by joint distribution over time: For time $1, 2, \ldots, T$, joint pdf over $\mathbb{R}^T$ is

$$
\begin{aligned}
& p_{X_1, X_2, \ldots, X_T}(x_1, x_2, \ldots, x_T; 1, 2, \ldots, T) \\
\overset{\text{defn}}{=} & p(X_1 = x_1, X_2 = x_2, \ldots, , X_T = x_T) \\
= & p(X_T = x_T \mid \underbrace{X_1 = x_1, \ldots, X_{T-1} = x_{T-1}}_{\text{history}(1:T-1)}) \times \\
& p(X_{T-1} = x_{T-1} \mid \underbrace{X_1 = x_1, \ldots, X_{T-2} = x_{T-2}}_{\text{history}(1:T-2)}) \times \\
& \cdots \times p(X_2 = x_2 \mid X_1 = x_1) \times P(X_1 = x_1)
\end{aligned}
$$

using formula $p(x, y, z) = p(z|x, y)p(y|x)p(x)$.

Two tractable stochastic models:

1. IID: $p(X_k)$ on $\mathbb{R}$ suffices to construct joint pdf on $\mathbb{R}^T$
2. Markov: $p(X_k|X_{k-1})$ on $\mathbb{R}^2$ to construct joint pdf on $\mathbb{R}^T$.

Math course. Discrete time: construct cdf on $\mathbb{R}^\infty$: cylinder sets, Kolmogorov consistency theorem. Cont-time: cdf on function space

IID and Markov processes

IID. X_k is indpt and identically distributed (iid) on state space $\mathcal{X}$ if conditional densities satisfy:
(i) $p(X_{n+1} = x|x_1, x_2, \ldots, x_n) = p(X_{n+1} = x)$
(ii) $p(X_{n+1} = x)$ has the same pdf/pmf for all n.
Note $\int_{\mathcal{X}} p(X_n = x)dx = 1$.

Markov. X_k is Markov on state space $\mathcal{X}$ if for all n,

$$p(X_{n+1} = x|x_1, x_2, \ldots, x_n) = p(X_{n+1} = x|x_n) \quad \forall x \in \mathcal{X}$$

Initial density: $\pi_0(x) = p(X_0 = x)$.

Remarks (i) IID processes have memoryless probability laws.
Examples: dice or coin tosses, noise to a first approx
(ii) Markov processes have one-step memory probability laws.
Examples: Most real world signals are Markovian - stock market, speech, video, moving target/vehicle, queuing system
(iii) IID is a special case of Markov

For IID processes, joint distribution of $X_1, \ldots, X_T$ factorizes

$$p_{X_1,X_2,\ldots,X_T}(x_1, x_2, \ldots, x_T; 1, 2, \ldots, T) = p(X_T = x_T)\, p(X_{T-1} = x_{T-1}) \times \cdots \times p(X_0 = x_0)$$

For Markov processes, joint distribution of $X_1, \ldots, X_T$ factorizes

$$p_{X_1,X_2,\ldots,X_T}(x_1, x_2, \ldots, x_T; 1, 2, \ldots, T) = p(X_T = x_T|x_{T-1})\, p(X_{T-1} = x_{T-1}|X_{T-2}) \times \cdots \times p(X_1 = x_1|x_0) \times p(X_0 = x_0)$$

Finite state Markov chains

1. Finite state space $\mathcal{X} = \{1, \ldots, X\}$, i.e., X alphabets.
2. Initial distribution vector π_0 (X dim col vector)

$$\pi_0 = [P(X_0 = 1), P(X_0 = 2), \ldots, P(X_0 = X)]'$$

   Clearly $\pi_0' \mathbf{1} = 1$ i.e., $\sum_{j=1}^{X} \pi_0(j) = 1$.
3. Transition probabilities: $P_{ij} = P(X_k = j | X_{k-1} = i)$ where $0 \le P_{ij} \le 1$, $\sum_{j=1}^{X} P_{ij} = 1$ for $i = 1, 2, \ldots, X$.
   $X \times X$ transition probability matrix P with elements

$$P_{ij} = P(X_k = j | X_{k-1} = i), \quad 0 \le P_{ij} \le 1, \sum_{j=1}^{X} P_{ij} = 1 \equiv P\mathbf{1} = \mathbf{1}$$

Remarks

1. P always has one eigenvalue at 1 for eigenvector $\mathbf{1}$.
2. P^k is always a stochastic matrix for any integer $k \ge 0$.
3. If all rows of P are identical then iid
4. More precisely, first-order homogeneous Markov chain.

Example: Three state Markov chain with absorbing state:

$$\mathcal{X} = \{1, 2, 3\}, \ \pi_0 = \begin{bmatrix} 0.3 \\ 0.7 \\ 0 \end{bmatrix}, \quad P = \begin{bmatrix} 0.8 & 0.1 & 0.1 \\ 0.6 & 0.4 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

States 1 and 2 are transient; state 3 is absorbing.

Q: Given N point sample path how to estimate transition prob?

$$\hat{P}_{ij} = \frac{\text{number of jumps from } i \text{ to } j}{\text{number of times Markov chain is in } i}$$

Simulating a Markov chain

1. IID process: Repeated use of inverse transform/acceptance rejection where uniform numbers at each step are independent.

2. Markov chain: Let P_i denote the i-th row of P. Since given X_k, state X_{k+1} is conditionally independent of the past

1. Generate $X_0 \sim \pi_0$.
2. For $k = 1, 2, \ldots$, generate $X_k \sim P_{x_{k-1}}$.

Step 1 and 2 using inverse transform or acceptance rejection.

Markov chain State Properties:

1. *Recurrent and Transient States*: A state is *recurrent* if it is visited infinitely often. Otherwise state is called *transient*.

Result: State i is recurrent if $\sum_{k=1}^{\infty} P_{ii}^k = \infty$. Otherwise transient.

Proof: Expected number of visits to state j if started in state i is

$$v_{ij} = \sum_{k=0}^{\infty} \mathbb{E}\{\mathbf{1}_{X_k=j} | X_0 = i\} = \sum_{k=0}^{\infty} P\{X_k = j | X_0 = i\} = \sum_{k=0}^{\infty} P_{ij}^k$$

Visit matrix $V = \sum_{k=0}^{\infty} P^k$.

2. *Periodic and Aperiodic States*: A state i of a Markov chain has period n if $P(X_k = i | X_0 = i) \neq 0 \quad , k = 0, n, 2n, 3n, \ldots$ and $P(X_k = i | X_0 = i) = 0$ otherwise.
If period $n = 1$, state is *aperiodic*. If all states are aperiodic, Markov chain is called aperiodic.

Chapman Kolmogorov Theorem - Optimal Prediction

Result: (Chapman Kolmogorov (CK) eqn) Given π_0 and P, the state probability vector at time k

$$\pi_k = [P(X_k = g_1), \ldots, P(X_k = g_X)]'$$

can be computed as

$$\boxed{\pi_{k+1} = P'\pi_k = {P'}^{k+1}\pi_0}$$

Then predicted state mean at time k is

$$\hat{X}_k = \mathbb{E}\{X_k\} = g'\pi_k$$

Predicted mean is $\hat{x}_k$ is MMSE optimal predictor:

$$\mathbb{E}\{(X_k - \hat{X}_k)^2\} \leq \mathbb{E}\{(X_k - \phi(\pi_0))^2\}.$$

Proof of CK: From total probability rule

$$\pi_{k+1}(j) = P(X_{k+1} = g_j) = \sum_{i=1}^{X} P(X_{k+1} = g_j | X_k = g_i)\, P(X_k = g_i)$$

$$= \sum_{i=1}^{X} P_{ij}\pi_k(i)$$

Remark: State probability vector π_k evolves as LTI system with state matrix P. Clearly, $\pi_k'\mathbf{1} = 1$ for all k.

Convergence of Markov Chains

Limiting Distribution *Limiting distribution* is

$$\lim_{k\to\infty} \pi_k = \lim_{k\to\infty} {P'}^k \pi_0.$$

This limiting distribution may not exist. For example if

$$P = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \pi_0 = \begin{bmatrix} \pi_0(1) \\ \pi_0(2) \end{bmatrix}, \quad \text{then } \pi_k = \begin{cases} \begin{bmatrix} \pi_0(2) & \pi_0(1) \end{bmatrix}' & k \text{ odd} \\ \begin{bmatrix} \pi_0(1) & \pi_0(2) \end{bmatrix}' & k \text{ even} \end{cases}$$

and so $\lim_{k\to\infty} \pi_k$ does not exist unless $\pi_0(1) = \pi_0(2) = 1/2$.

Stationary Distribution X-dimensional vector π_∞

$$\pi_\infty = P'\pi_\infty, \quad \mathbf{1}'\pi_\infty = 1$$

So π_∞ is normalized right eigenvector of P' with eigenvalue 1. Equivalently, choosing $\pi_0 = \pi_\infty$ implies $\pi_k = \pi_\infty$ for all k. Stationary distribution also called the *invariant, equilibrium or steady-state distribution.*

Limiting distributions are a subset of stationary distributions. For example, for above P, $\pi_\infty = \begin{bmatrix} 0.5 & 0.5 \end{bmatrix}'$ is a stationary distribution but there is no limiting distribution.

Markov chain properties:

1. A Markov chain is regular (primitive) if for some $k \geq 1$, all elements of P^k are strictly positive.
2. A Markov chain is irreducible if for each $i, j \in \mathcal{X}$, there exists $k \geq 1$ such that $P^k_{ij} > 0$.

Theorem 1 (Perron-Frobenius). *Consider a finite-state Markov chain with regular transition matrix P. Then:*

1. *The eigenvalue 1 has algebraic & geometric multiplicity of one.*
2. *All remaining eigenvalues of P have modulus strictly smaller than 1.*
3. *The eigenvector of P' corresponding to eigenvalue of 1 can be chosen with non-negative elements.*
4. *$P^k = \mathbf{1}\pi'_\infty + O(|\lambda_2|^k)$ where λ_2 is the second largest eigenvalue modulus (SLEM).*
5. *Limiting distribution and stationary distribution coincide.*

$$\pi_\infty(i) = \lim_{N\to\infty} \frac{1}{N} \sum_{k=1}^{N} I(x_k = i)$$

is fraction of time Markov chain spends in state i.

Statement 4 says if the transition matrix P is regular, state probability vector π_k forgets initial condition geometrically fast.

$$\pi_k = {P'}^k \pi_0 = \pi_\infty \mathbf{1}'\pi_0 + O(|\lambda_2|^k)\pi_0 = \pi_\infty + O(|\lambda_2|^k)\pi_0.$$

So k-step ahead predictor of a Markov chain forgets initial condition geometrically fast in terms of the second largest eigenvalue modulus, $|\lambda_2|$.
Equivalently, π_k converges geometrically fast to π_∞.

Proof of (ii): Define spectral radius $\rho(A) = \max_i |\lambda_i|$
Lemma 1: $\rho(A) \leq \|A\|_\infty$ where $\|A\|_\infty = \max_i \sum_j |a_{ij}|$
Proof: $|\lambda|\|x\| = \|\lambda x\| = \|Ax\| \leq \|A\|\|x\| \implies |\lambda| \leq \|A\| \quad \forall \lambda.$

In our case $\|A\|_\infty = 1$ and A has an eigenvalue at 1. So $\rho(A) = 1$.

Proof of (iii): For positive matrix A, $A'\pi = \pi$ implies $A'|\pi| = |\pi|$
Proof: $|\pi| = |A'\pi| \leq |A'||\pi| = A'|\pi|$ So $A'|\pi| - |\pi| \geq 0$.
But $A'|\pi| - |\pi| > 0$ is impossible, since it implies $1'A'|\pi| > 1'|\pi|$, i.e., $1'|\pi| > 1'|\pi|$.

Dobrushin Coefficient of Ergodicity

Upper bound for SLEM.

$$\rho(P) = \frac{1}{2} \sup_{i,j} \sum_k |P_{ik} - P_{jk}| = \sup_{i,j} \|P' e_i - P' e_j\|_{\text{TV}}.$$

Note: $0 \le \rho(P) \le 1$.

$\rho(P)$ is max variational dist between two rows of P.
Given pmfs α and β variational distance

$$\|\alpha - \beta\|_{\text{TV}} = \frac{1}{2}\|\alpha - \beta\|_1 = \frac{1}{2} \sum_{i \in \mathcal{X}} |\alpha(i) - \beta(i)|$$

If $\rho(P) < 1$, then Markov chain is geometrically ergodic. This implies SLLN: $\mu_n = \frac{1}{n} \sum_{k=1}^n X_k \to \pi_\infty$.

Example 1: If $P = \begin{bmatrix} P_{11} & 1 - P_{11} \\ 1 - P_{22} & P_{22} \end{bmatrix}$, then
$\rho(P) = |1 - P_{11} - P_{22}| = |\lambda_2|$.
Example 2: If $P_{ij} \ge \epsilon$ for all i, j, then $\rho(P) \le 1 - \epsilon$. All non-zero transition probabilities: trivially irreducible and aperiodic.

Example 3. Doeblin/Minorization condition: (advanced).

$$P_{ij} \ge \epsilon \kappa_j, \quad \sum_j \kappa_j = 1, \kappa_j \ge 0 \implies \rho(P) \le 1 - \epsilon$$

Remark: $\rho(P) = \sup_{i \ne j} \frac{W(P' e_i, P' e_j)}{W(e_i, e_j)}$; W is Wasserstein distance.

Example: Google Page Rank Algorithm

Search engine: webpage importance = # pages pointing to it.
Inventors: Larry Page & Sergey Brin. Assume surfer walks www according to a Markov chain.
Given webpage transition graph, how to compute page rank?

1. Construct transition matrix A as: If node i connects to L nodes $j_1, \ldots, j_L$ set trans probs

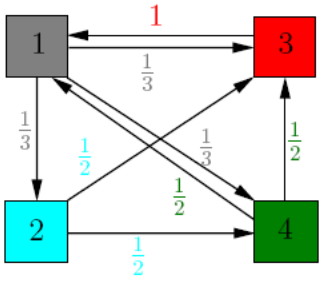
$$a_{ij_1} = a_{ij_2} = \cdots = a_{ij_L} = 1/L$$

2. Non-regular transition matrix: page rank transition matrix: for damping factor $0 \leq p \leq 1$, (usually $p = 0.15$)

$$M = (1-p)A + pB, \text{ where } B = \frac{1}{n}\begin{bmatrix} 1 & 1 & \cdots & 1 \\ \vdots & \vdots & \ddots & \vdots \\ 1 & 1 & \cdots & 1 \end{bmatrix}$$

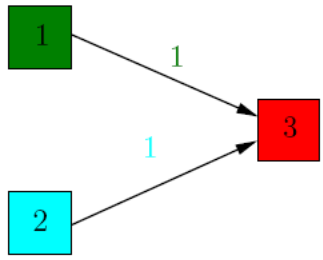
3. Page rank = stationary pmf $\pi[\infty]$ for trans prob matrix M. Fraction of time webpage is visited over a long time horizon.

Example 1: Suppose $p = 0$, then



$$M = A = \begin{bmatrix} 0 & 1/3 & 1/3 & 1/3 \\ 0 & 0 & 1/2 & 1/2 \\ 1 & 0 & 0 & 0 \\ 1/2 & 0 & 1/2 & 0 \end{bmatrix}, \quad \pi[\infty] = \begin{bmatrix} 0.39 \\ 0.13 \\ 0.29 \\ 0.19 \end{bmatrix}$$

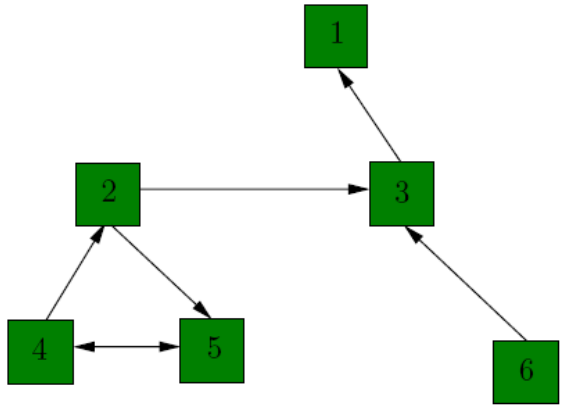
Example 2 (Dangling nodes):



$$M = (1-p)\begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} + pB$$

For $p = 0.15$ compute $\pi[\infty]$.

Exercise: Compute page ranks for following for $p = 0.15$.



Homework: Read carefully thru

`https://www.mathworks.com/help/matlab/examples/use-page-rank-algorithm-to-rank-websites.html`

Matlab defines damping factor as $1-p$ instead of p.

Recall $M = (1-p)A + pB$ where B is uniform.

B are called restart (teleportation) probabilities

Personalized page rank: Choose B non-uniform

Part I.4. Statistical Inference

Two fundamental results:
(i) Law of large numbers (ii) Central Limit Theorem.
Law of Large Numbers (LLN). LLN relates

$$\text{statistics (real world)} \leftrightarrows \text{probability (mathematical model)}$$

For iid process $X_n(\omega)$ two averages can be computed:
1. Expected value (ensemble average) for fixed n: (from pdf)

$$\mathbb{E}\{X_n(\omega)\} = \int_{\mathbb{R}} x f_{X[n]}(x) dx = \int x f_X(x) = \mu$$

Note: Recall because iid, μ is a const.

2. Real life sample path time average (statistic) for fixed w given N observations: (in real life you live one sample path)

$$\hat{\mu}_N(\omega) = \frac{1}{N}(X[w,1] + \ldots + X[w,N])$$

Result Strong LLN: For iid process as $N \to \infty$, $\hat{\mu}_N(\omega)$ converges "strongly" to μ (statistic $\hat{\mu}_N$ computed from data converges to mean of random variable μ computed from probabilistic model.)

Aside. Stochastic Convergence

Consider $(\Omega, \mathcal{F}, \mathbb{P})$.

1. A sequence of rv $\{X_n\}$ converges in probability to rv X if for any $\epsilon > 0$, $\lim_{n\to\infty} p_n = 0$ where

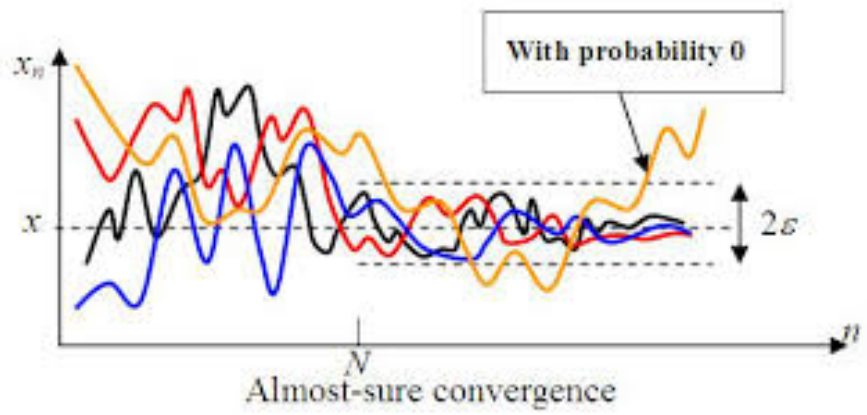
$$p_n = \mathbb{P}(\{\omega : |X_n(\omega) - X(\omega)| > \epsilon\}).$$

Equivalently: $\boxed{\lim_{n\to\infty} \mathbb{P}(|X_n - X| > \epsilon) = 0, \quad \forall \epsilon > 0}$

(ii) $\{X_n\}$ converges almost surely (with probability one) to rv X if for any $\epsilon > 0$, $\lim_{n\to\infty} a_n = 0$ where

$$\begin{aligned} a_n &= \mathbb{P}(\{\omega : \exists m \geq n \text{ such that } |X_m(\omega) - X(\omega)| > \epsilon\}) \\ &= \mathbb{P}(\{\omega : \cup_{m\geq n} |X_m(\omega) - X(\omega)| > \epsilon\}) \end{aligned}$$

Equivalently: $\boxed{\lim_{n\to\infty} \mathbb{P}(\sup_{m\geq n} |X_m - X| > \epsilon) = 0, \quad \forall \epsilon > 0}$



(iii) $\{X_n\}$ converges in distribution if $\lim_{n\to\infty} F_n(x) = F(x)$ for all x for which F is continuous. Here F_n and F are cdfs of X_n and X, respectively.

Remark:

$$\begin{aligned}\lim_{n\to\infty} a_n &= \lim_{n\to\infty} \mathbb{P}(\{\omega : \cup_{m\geq n}|X_m(\omega) - X(\omega)| > \epsilon\}) \\ &= \mathbb{P}(\{\omega : \lim_{n\to\infty} \cup_{m\geq n}|X_m(\omega) - X(\omega)| > \epsilon\}) = 0\end{aligned}$$

by sequential continuity property of probability: If $A_n \supset A_{n+1}$ then $\mathbb{P}(\lim_n A_n) = \lim_n \mathbb{P}(A_n)$).

So $X_n \to X$ a.s. equiv to: as $n \to \infty$,

$$a_n = \mathbb{P}(\{\omega : |X_n(\omega) - X(\omega)| > \epsilon \text{ infinitely often}\}) = 0.$$

Result. Almost sure convergence implies convergence in probability which in turn implies convergence in distribution.

Proof. Obviously event associated with a.s. convergence is superset of event associated with convergence in prob:

$$\{\omega : \cup_{m\geq n}|X_m(\omega) - X(\omega)| > \epsilon\} \supseteq \{\omega : |X_n(\omega) - X(\omega)| > \epsilon\}.$$

So $a_n \geq p_n$. As a result $a_n \to 0$ implies $p_n \to 0$. So almost sure convergence always implies convergence in probability

Under what additional conditions does convergence in probability imply almost sure convergence?

Equivalently: When does $p_n \to 0$ imply $a_n \to 0$?

Recall $p_n = \mathbb{P}(\{\omega : |X_n(\omega) - X(\omega)| > \epsilon\})$.

$$\begin{aligned} a_n &= \mathbb{P}(\{\omega : \cup_{m \geq n} |X_m(\omega) - X(\omega)| > \epsilon\}) \\ &\leq \sum_{m \geq n} \mathbb{P}(\{\omega : |X_m(\omega) - X(\omega)| > \epsilon\}) = \sum_{m \geq n} p_n \end{aligned}$$

where the inequality follows since $\mathbb{P}(\cup_m A_m) \leq \sum_m \mathbb{P}(A_m)$. So if as $n \to \infty$, $\sum_{m \geq n} p_m \to 0$, then $a_n \to 0$. Therefore

$$\boxed{\sum_{n=1}^{\infty} p_n < \infty \implies a_n \to 0.}$$

Summary: $\sum_n \mathbb{P}(\{\omega : |X_n(\omega) - X(\omega)| > \epsilon\}) < \infty$ implies $X_n \xrightarrow{\text{a.s.}} X$.

Borel Cantelli lemma:

1. If $\sum_n \mathbb{P}(A_n) < \infty$, then $\mathbb{P}(A_n \text{ infinitely often }) \to 0$.
   Use this as follows: $A_n = \{w : |X_n(w) - X(w)| > \epsilon\}$. Then $\sum_n \mathbb{P}(A_n) < \infty$ implies $X_n \xrightarrow{\text{a.s.}} X$
2. If $\{A_n\}$ are independent, then $\sum_n \mathbb{P}(A_n) = \infty$ implies A_n does not converge a.s.

Example 1: Suppose $\mathbb{P}(X_n = 1) = p_n$ and $\mathbb{P}(X_n = 0) = 1 - p_n$.

1. Then if $p_n \to 0$, $X_n \to 0$ in probability.
2. For $X_n \xrightarrow{wp1} 0$ we need $\sum_n p_n < \infty$.
3. X_n indpt and $p_n = 1/n$: then X_n does not converge to 0 wp1.

Example 2: Strong Law of Large Numbers for $\mathbb{E}\{|X|^4\} \leq C$.
Suppose $\{X_k\}$ iid with $\mathbb{E}\{X_k\} = \mu = 0$, $\mathbb{E}\{X_k^2\} = \sigma^2$.
Define

$$S_n = \sum_{k=1}^{n} X_k, \quad \hat{\mu}_n = S_n/n$$

SLLN states $\boxed{\hat{\mu}_n \xrightarrow{\text{a.s.}} \mu}$

1. *Weak law of large numbers*: Using Markov inequality

$$\mathbb{P}(|\hat{\mu}_n - \mu| > \epsilon) \leq \frac{\mathbb{E}\{|\mu_n - \mu|^4\}}{\epsilon^4}$$

IID assumption implies

$$\mathbb{E}\{|\hat{\mu}_n|^4\} = \frac{1}{n^4}\left[n\mathbb{E}\{X_i^4\} + 3n(n-1)\mathbb{E}\{X_i^2 X_j^2\}\right] \leq \frac{C}{n^3} + \frac{3\sigma^4}{n^2}$$

So $\mathbb{P}(|\hat{\mu}_n - \mu| > \epsilon) \to 0$, i.e. $\hat{\mu}_n \xrightarrow{\text{in prob}} \mu$.

2. *Strong law of large numbers*: To prove $\hat{\mu}_n \xrightarrow{\text{a.s.}} \mu$ use Borel Cantelli lemma. Define

$$A_n = \{|\hat{\mu}_n| > \epsilon\}$$

So clearly $\sum_n P(A_n) < \infty$. Therefore, $P(\{|\hat{\mu}_n| > \epsilon \text{ i.o}\} = 0$.
Therefore Borel Cantelli implies $\hat{\mu}_n \xrightarrow{\text{a.s.}} \mu$.

Exercises:

1. Convg in prob implies existence of subsequence that converges almost surely.
2. Convergence in distribution to a constant implies convergence in probability to the constant.

Result 1. Strong Law of Large Numbers (SLLN)

Define the sample average as $\mu_n = \frac{1}{n}\sum_{k=1}^{n} X_k$. SLLN states

Theorem 2. *(IID) Suppose $\{X_k\}$ is an i.i.d. sequence of vector random variables. Then $\lim_{n\to\infty} \mu_n = \mathbb{E}\{X_1\}$ almost surely iff $\mathbb{E}\{|X_1|\} < \infty$ where $\mathbb{E}\{X_1\} = \mu$.*

(Finite-state Markov) Suppose $\{X_n\}$ is an X-state Markov chain with state space of X-dimensional unit vectors. Assume transition matrix P is regular. Then $\lim_{n\to\infty} \mu_n = \pi_\infty$ almost surely.

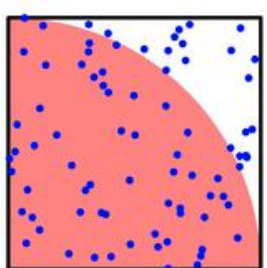
Remarks: SLLN is basis of Monte Carlo (MC) inference.

1. The i.i.d. version also called Kolmogorov's SLLN.
2. For Markov chain $\pi_\infty(i)$ is fraction of time spent in state i.
3. A random process for which SLLN holds is called "ergodic". So any iid process is ergodic. For Markov chain, need geometric ergodicity for SLLN.

Example 1. Multidimensional Monte-Carlo integration: Compute difficult integrals numerically by stochastic simulation and LLN. As $n \to \infty$,

$$\int_{\mathbf{R}^X} f(x)dx \approx \frac{1}{n}\sum_{i=1}^{n} \frac{f(x_i)}{\pi(x_i)} \text{ where } x_i \sim \pi(x) \ .$$

A dumb approximation of π



$$P(x,y) = \begin{cases} 1 & 0<x<1 \text{ and } 0<y<1 \\ 0 & \text{otherwise} \end{cases}$$

$$\pi = 4 \iint \mathbb{I}\left((x^2+y^2)<1\right) P(x,y)\, \mathrm{d}x\, \mathrm{d}y$$

In Matlab avoid "for loops". Use vectorized code;

Example 2. If not iid, then LLN may not hold.

1. Consider random process:

$$X[n] = X[n-1], \quad X[0] = \begin{cases} 0 & \text{with prob } 0.8 \\ 1 & \text{with prob } 0.2 \end{cases}$$

   Then there are only two possible outcomes
   $w = [0,0,0,0,0,0,\ldots,]$ or $w = [1,1,1,1,1\ldots,]$.
   So time average $\hat{\mu}_N = 0$ or 1. But $\mu = \mathbb{E}\{X[n]\} = 0.2$.
   Thus $\mu \neq \hat{\mu}_N$, i.e., LLN does not hold.
   Here $X[n]$ are identically distributed but not indpt. For non-iid sources, need to be more careful.
2. SLLN does not hold for Cauchy density ($\alpha = 1$) since $\mathbb{E}\{|X|\}$ is not finite.

Example 3: Estimation of a cdf from random samples. Suppose $\{X_n\}$ is an i.i.d. sequence simulated from an unknown cdf F. The empirical cdf is defined as

$$\boxed{F_n(x) = \frac{1}{n}\sum_{k=1}^{n} I(X_k \leq x)}$$

$F_n(x)$ is a natural estimator of F when F is not known. By SLLN $\lim_{n\to\infty} F_n(x) = F(x)$ almost surely for each x. Actually for estimation of cumulative distributions uniform almost sure convergence. (Glivenko-Cantelli Theorem)

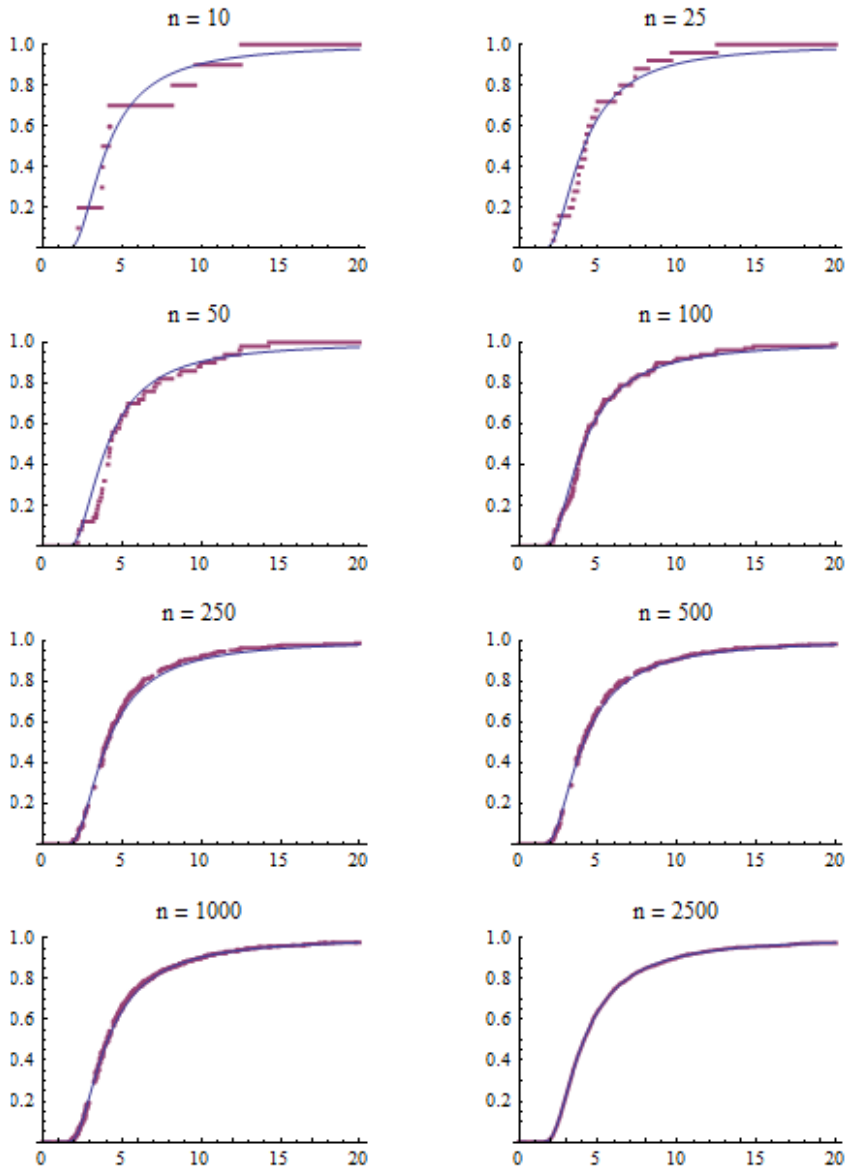
In Matlab: `[F,z] = ecdf(x)` returns empirical cdf `F` evaluated at grid points `z` using the data in the vector `x`. Then `plot(z,F)`

Theorem 3 (Glivenko-Cantelli Theorem). *Suppose* $\{X_n\}$ *is an i.i.d. sequence with cdf* F. *Then uniform SLLN holds:*

$$\lim_{n\to\infty} \sup_{x\in\mathbb{R}} |F_n(x) - F(x)| = 0 \quad \textit{almost surely.}$$

"Large number of random samples uniformly approximates cdf."

Essential idea in stochastic simulation and particle filtering.



Dvoretsky-Kiefer-Wolfowitz inequality:

$$P(\sup_x |F_n(x) - F(x)| > \epsilon) \leq 2e^{-2n\epsilon^2}$$

Example 4. Shannon-McMillan-Breiman Theorem.
Assume

1. $y_k \sim p(y|\theta^o)$ iid, $k = 1, 2 \ldots, n$.
2. We dont know θ^o and assume model θ.

SLLN gives iid version of Shannon-McMillan-Breiman theorem:

$$\frac{1}{n}\sum_{k=1}^{n} \log p(y_k|\theta) \xrightarrow{\text{wp1}} \mathbb{E}_{\theta^o}\{\log p(y|\theta)\} = \int \log p(y|\theta)\, p(y|\theta^o) dy$$

$$\frac{1}{n}\sum_{k=1}^{n} \log p(y_k|\theta^o) \xrightarrow{\text{wp1}} \mathbb{E}_{\theta^o}\{\log p(y|\theta^o)\} = \int \log p(y|\theta^o)\, p(y|\theta^o) dy$$

Negative of KL divergence is

$$\begin{aligned} -K(\theta, \theta^o) &= \mathbb{E}_{\theta^o}\{\log p(y|\theta)\} - \mathbb{E}_{\theta^o}\{\log p(y|\theta^o)\} \\ &= \int \log \frac{p(y|\theta)}{p(y|\theta^o)}\, p(y|\theta^o) dy \leq 0 \end{aligned}$$

since by Jensen inequality: $\mathbb{E}\{\log(X)\} \leq \log(\mathbb{E}\{X\})$
Also clearly $K(\theta^o, \theta^o) = 0$.
So to estimate true model, we should maximize $-K(\theta, \theta^o)$ wrt θ.
Equivalently maximize $\mathbb{E}_{\theta^o}\{\log p(y|\theta)\}$ wrt θ
Equivalently maximize $\frac{1}{n}\sum_{k=1}^{n} \log p(y_k|\theta)$ wrt θ for large n.
Equivalently maximize $p(y_1, \ldots, y_n|\theta)$ wrt θ
This is the basis of maximum likelihood estimation (Part IV).

Result 2. Central Limit Theorem

Theorem 4. *With* $\mu_n = \frac{1}{n}\sum_{k=1}^n X_k$ *and* $\stackrel{\mathcal{L}}{=}$ *denoting convergence in distribution, the following hold:*

(IID) Suppose $\{X_k\}$ *is iid with zero mean and finite variance* σ^2*. Then* $\lim_{n\to\infty}\sqrt{n}\mu_n = \frac{1}{\sqrt{n}}\sum_{k=1}^n X_k \stackrel{\mathcal{L}}{=} \boldsymbol{N}(0,\sigma^2)$*.*

Equivalently: $\mu_n \stackrel{\mathcal{L}}{=} \boldsymbol{N}(0,\sigma^2/N)$*.*

(Finite-state Markov) Suppose $\{X_n\}$ *is an* X*-state Markov chain with state space comprising of* X*-dimensional unit vectors. Suppose* P *is regular implying a unique stationary distribution* π_∞ *exists. Then for any* $g \in \mathbb{R}^X$*,*

$$\lim_{n\to\infty}\sqrt{n}g'(\mu_n - \pi_\infty) \stackrel{\mathcal{L}}{=} \boldsymbol{N}(0,\sigma^2)$$

$$\sigma^2 = 2g'\mathrm{diag}(\pi_\infty)Zg - g'\mathrm{diag}(\pi_\infty)(I + \mathbf{1}\pi_\infty')g.$$

$Z = (I - (P - \mathbf{1}\pi_\infty'))^{-1}$ *is called fundamental matrix.*

Remarks:

1. The CLT is best understood as follows: Suppose $X_1, X_2, \ldots$ are iid with $\mathbb{E}\{X_i\} = 0$, and variance $\mathrm{var}(X_i) = \sigma^2$. Then

$$\text{SLLN: } \frac{1}{n}\sum_{i=1}^n X_i \xrightarrow{\text{wp1}} 0 \qquad \text{CLT: } \frac{1}{\sqrt{n}}\sum_{i=1}^n X_i \xrightarrow{\mathcal{L}} N(0,\sigma^2)$$

$$\text{Law of iterated logarithm: } \frac{1}{\sqrt{2n\log\log n}}\sum_{i=1}^n X_i \in [-1,1]$$

2. Functional CLT (advanced): Scaled sum of stochastic processes converge to a Gaussian stochastic process

Example 1. Multipath yields Rayleigh fading wireless channel:

$$X(t) = \frac{1}{\sqrt{N}} \sum_{k=1}^{N} A_k \cos(w_c t + \Theta_k)$$

A_k: attenuation of signal on kth path. Assume iid in $[-1, 1]$.
Θ_k: propagation delay due to kth path. Assume iid $U(0, 2\pi)$.

From basic trigonometry $X(t) = X_I \cos w_c t - X_Q \sin w_c t$

$$X_I = \frac{1}{\sqrt{N}} \sum_{k=1}^{N} A_k \cos \Theta_k, \quad X_Q = \frac{1}{\sqrt{N}} \sum_{k=1}^{N} A_k \sin \Theta_k$$

As $N \to \infty$, CLT implies $X_I \sim N(0, \sigma^2)$, $X_Q \sim N(0, \sigma^2)$ where X_I and X_Q are indpt.

Then $X(t)$ expressed in terms of envelope and phase is:

$$X(t) = R \cos(w_c t + \Psi), \quad \text{where } R = \sqrt{X_I^2 + X_Q^2} \sim \text{Rayleigh}$$

and $\Psi \sim U[0, 2\pi]$.

Example 2. Empirical cdf is $F_n(x) = \frac{1}{n} \sum_{k=1}^{n} I(X_k \le x)$.
Central limit theorem says

$$\lim_{n \to \infty} \sqrt{n}(F_n(x) - F(x)) \stackrel{\mathcal{L}}{=} \boldsymbol{N}\big(0, F(x)(1 - F(x))\big) \text{ for each } x$$

Homework: Show that the variance is $F(x)(1 - F(x))$.
Hint: Define $Z_k = I(X_k \le x)$. Show $\text{Var}(Z_k) = F(x) - F^2(x)$.
Since Z_k are iid, $\text{Var}(\frac{1}{n} \sum_{k=1}^{n} Z_k) = (F(x) - F^2(x))$.

CLT is the basis of statistical significance and confidence tests.

Importance Sampling in Simulation

Why? Variance Reduction in Simulation.

Monte-Carlo evaluation of an integral: Simulate N i.i.d. samples of $x_k; k = 1, 2, \ldots, N$ from some pdf p. Then as $N \to \infty$, SLLN

$$\frac{1}{N}\sum_{k=1}^{N} c(x_k) \to \mathbb{E}_p\{c(x)\} = \int_{\mathcal{X}} c(x)p(x)dx \quad \text{with probability 1.}$$

$\frac{1}{N}\sum_{k=1}^{N} c(x_k)$ is an unbiased estimate of $\mathbb{E}_p\{c(x)\}$ for any N; however, the variance of the estimate can be large.

How to reduce the variance of the estimate?

Let $p(x)$ denote a target distribution. Then for any density $q(x)$

$$\mathbb{E}_p\{c(x)\} = \int_{\mathcal{X}} c(x)\frac{p(x)}{\cancel{q(x)}}\cancel{q(x)}\,dx$$

as long as $q(x)$ is chosen so that $p(x)/q(x)$ is finite for all x.

Importance sampling estimate of $\mathbb{E}_p\{c(x)\}$:

(i) sample $x_k; k = 1, 2, \ldots, N$ from importance distribution ("instrumental distribution") $q(x)$ st $p(x)/q(x) < \infty$ for all x.

(ii) Then importance sampling estimate is

$$\boxed{\hat{c}_N = \frac{1}{N}\sum_{k=1}^{N} c(x_k)\frac{p(x_k)}{q(x_k)}, \quad x_k \sim q.}$$

Clearly unbiased estimate.

Result. If the sequence $\{x_k\}$ is i.i.d., then via the strong law of large numbers, as $N \to \infty$, the importance sampled estimate $\hat{c}_N \to \mathbb{E}_p\{c(x)\}$ almost surely. Also, by Central Limit Theorem

$$\lim_{N\to\infty} \sqrt{N}\,(\hat{c}_N - \mathbb{E}_p\{c(x)\}) \stackrel{\mathcal{L}}{=} \boldsymbol{N}(0, \mathrm{Var}_q(c(x))),$$

$$\text{where } \mathrm{Var}_q(c(x)) = \int c^2(x)\frac{p^2(x)}{q(x)}dx - \mathbb{E}^2\{c(x)\}. \quad (3)$$

Example. Rare Event Estimation: For fixed real number α, evaluate

$$c = \mathbb{P}(x > \alpha) \quad \text{where } x \sim \boldsymbol{N}(0,1)$$

Standard MC estimator based on N i.i.d. samples is

$$\hat{c} = \frac{1}{N}\sum_{k=1}^{N} I(x_k > \alpha), \quad x_k \sim \boldsymbol{N}(0,1) \text{ i.i.d.} \quad (4)$$

If α is large we will get very few samples $x_k > \alpha$. Then standard MC has high variance (inaccurate estimate).

Choosing the importance density $q = \boldsymbol{N}(\mu, 1)$, yields

$$\hat{c} = \frac{1}{N}\sum_{k=1}^{N} I(x_k > \alpha)\exp\left(\frac{\mu^2}{2} - \mu x_k\right), \quad x_k \sim \boldsymbol{N}(\mu,1) \text{ i.i.d.}$$

Choose $\mu = \alpha$ in importance sampling estimator. Let us estimate $\mathbb{P}(x > \alpha)$ where $\alpha = 8$, $x \sim \boldsymbol{N}(0,1)$.

Standard MC estimate: Matlab simulation for $N = 50000$ points yields $\hat{c} = 0$ (in double precision) which is useless.

Importance sampling estimate: Matlab simulation for $N = 50000$ points with $\mu = \alpha = 8$, yields estimate $\hat{c} = 6.25 \times 10^{-16}$.

Example 2 (HW): Suppose w_k is a random walk on integers:

$$w_{k+1} = \begin{cases} w_k + 1 & \text{wp } 1/2 \\ w_k - 1 & \text{wp } 1/2 \end{cases}, \quad w_0 = K > 0.$$

Define hitting time $\tau = \min\{k : w_k = 0\}$.

Aim. Estimate $P(\tau \le T)$ for fixed integer T.

Standard MC: Let τ^i be hitting time for ith simulation. Then

$$\hat{\tau} = \frac{1}{N} \sum_{k=1}^{N} I(\tau^i \le T)$$

If $P(\tau \le T)$ then standard MC requires many iterations N.

Importance sampling: Consider asymmetric random walk

$$v_{k+1} = \begin{cases} v_k + 1 & \text{wp } 1-\alpha \\ v_k - 1 & \text{wp } \alpha \end{cases}, \ \alpha \in [0,1], \quad v_0 = K > 0.$$

Define $\sigma = \#\{k < \tau : v_{k+1} < v_k\}$ = number of times downhill.

Let τ^i be hitting time for ith simulation. Then IS estimate is

$$\tilde{\tau} = \frac{1}{N} \sum_{k=1}^{N} I(\tau^i \le T) \frac{1}{2^\tau \alpha^\sigma (1-\alpha)^{\tau-\sigma}}$$

Example: Choose $T = 20$, $K = 10$, $N = 100$

α	0.5 (MC)	0.6	0.7	0.8	0.9
Var($\tilde{\tau}$)	0.0268	0.00436	0.00101	0.00058	0.00439

Self-normalized Importance Sampling

Often $p(x)$ is un-normalized and normalization is computationally intractable.
Define importance weight $w(x) = \frac{p(x)}{q(x)}$.
Self normalized IS:

$$\hat{c}_N = \frac{\sum_{k=1}^N c(x_k)w(x_k)}{\sum_{k=1}^N w(x_k)}, \quad x_k \sim q.$$

The self-normalized estimate is biased for any finite N. For an i.i.d. sequence $\{x_k\}$, from strong law of large numbers

$$\frac{1}{N}\sum_{k=1}^N c(x_k)w(x_k) \to \mathbb{E}_p\{c(x)\}, \quad \frac{1}{N}\sum_{k=1}^N w(x_k) \to 1$$

implying that $\hat{c}_N \to \mathbb{E}_p\{c(x)\}$ with probability one.

Derivation:

$$\frac{\int c(x)p(x)dx}{\int p(x)dx} = \frac{\int c(x)\frac{p(x)}{q(x)}q(x)dx}{\int \frac{p(x)}{q(x)}q(x)dx} = \frac{\mathbb{E}_q\{c(x)w(x)\}}{\mathbb{E}_q\{w(x)\}}$$

Many other variance reduction methods. (not covered in this course)

1. Variance Reduction by Conditioning: Uses property that $\mathrm{Var}(X) \geq \mathrm{Var}(\mathbb{E}\{X|Z\})$.
2. Variance Reduction by Stratified Sampling: Uses property that $\mathrm{Var}(X) \geq \mathbb{E}\{\mathrm{Var}(X|Z)\}$.

Aside: Big Data Perspective

1. Classical setting: $x_k \in \mathbb{R}^n$ iid. Then as number of time points $N \to \infty$, (so $n \ll N$) following asymptotic results hold:

$$\frac{1}{N}\sum_{k=0}^{N} x_k = \hat{\mu}_N \to \mathbb{E}\{X_k\} = \mu \quad \text{(SLLN)}$$

$$\frac{1}{\sqrt{N}}\sum_{k=0}^{N}(x_k - \mu) \to N(0, \Sigma) \quad \text{(CLT)}$$

Big Data: $n = N$ or $n > N$. Asymptotic statistics dont apply.

Main tool. *Concentration Inequality* (finite N analysis).

Hoeffding inequality. $X_k \in [a, b]$ iid. Then for any $N > 0$,

$$\boxed{P(|\hat{\mu}_N - \mu| > \epsilon) \le 2\exp\left(-\frac{2N^2\epsilon^2}{(b-a)^2}\right)}$$

Also applied to martingales. By Borel-Cantelli: $\hat{\mu}_N \to \mu$ wp1.

Least Squares: $Y_N = \Psi_{N\times n}\theta_n + \epsilon$.

Classical setting: $n \ll N$ (overdetermined case)

Big data: $n/N =$ const or $n \gg N$ (underdetermined case).

Example 1: $y_k = \theta + \epsilon_k$, $k = 1, \ldots, N$ where $\epsilon_k \in [-0, 5.0.5]$ iid.

$\boxed{\theta_{LS}(N) = \sum_{k=1}^{N} y_k/N}$ Classical: as $N \to \infty$, $\theta_{LS}(N) \to \theta$ w.p1.

Hoeffding inequality: $P(|\theta_{LS}(N) - \theta| > \epsilon) \le 2\exp(-2N^2\epsilon^2)$.

How much data N for $P(|\theta_{LS}(N) - \theta| > \epsilon) < \alpha$?

$\alpha < 2\exp(-2N^2\epsilon^2)$. So choose $N^2 > -\frac{1}{2\epsilon^2}\ln(\alpha/2)$.

$\alpha = \epsilon = 0.01$, $N \approx 165$. (works for any bounded rv ϵ_k)

Holy Grail of Machine Learning. Uniform Concentration Inequality

Supervised learning: Given labeled training data
$X_i, Y_i, i = 1, 2, \ldots, n$ and classifier h s.t. $\hat{Y}_i = h(X_i) \in \{0, 1\}$.
Risk of classifier h: $R(h) = P(Y \neq \hat{Y}) = P(Y \neq h(X))$
Empirical Risk (training error): $\hat{R}(h) = \frac{1}{n} \sum_{i=1}^{n} I\big(\hat{Y}_i \neq h(X_i)\big)$

Define $h^* = \operatorname{argmin}_h R(h)$ (oracle) and $\hat{h} = \operatorname{argmin}_h \hat{R}(h)$
To prove $R(h^*)$ and $R(\hat{h})$ are close, need uniform concentration:

$$\boxed{P(\sup_{h \in \mathcal{F}} |\hat{R}(h) - R(h)| > \epsilon) \leq \delta_n, \quad \delta_n \downarrow 0} \qquad \text{(UL)}$$

Result: If UL holds, then wp $\geq 1 - \delta_n$, $|R(\hat{h}) - R(h^*)| \leq 2\epsilon$.

Proof: $R(\hat{h}) - R(h^*) = R(\hat{h}) - \hat{R}(\hat{h}) + \hat{R}(\hat{h}) - R(h^*)$

$\leq R(\hat{h}) - \hat{R}(\hat{h}) + \hat{R}(h^*) - R(h^*)$ since $\hat{h}$ minimizes $\hat{R}$

$\implies |R(\hat{h}) - R(h^*)| \leq |R(\hat{h}) - \hat{R}(\hat{h})| + |\hat{R}(h^*) - R(h^*)|$

Uniform law: wp $\geq 1 - \delta_n$, $|R(\hat{h}) - \hat{R}(\hat{h})| \leq \epsilon$, $|\hat{R}(h^*) - R(h^*)| \leq \epsilon$.

Hoeffding: $P(|\hat{R}(h) - R(h)| > \epsilon) \leq c_1 \, e^{-c_2 n \epsilon^2}$

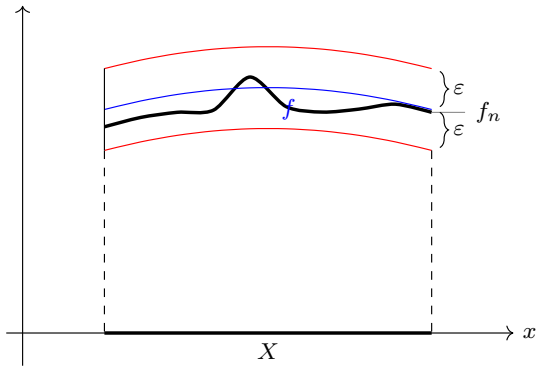
Vapnik Chervonenkis inequality (uniform conc inequality):

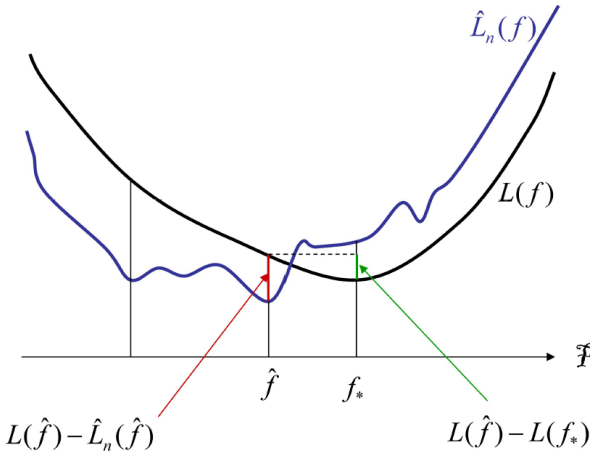
$$\boxed{P(\sup_{h \in \mathcal{F}} |\hat{R}(h) - R(h)| > \epsilon) \leq c_1 \, \mathcal{S}(\mathcal{F}, n) \, e^{-c_2 n \epsilon^2}}$$

shattering function = max num of distinct labels for n points

$$\mathcal{S}(\mathcal{F}, n) = \max_{x_1, \ldots, x_n} |\{\big(h(x_1), \ldots, h(x_n)\big), \quad h \in \mathcal{F}\}| \leq (n+1)^{V_{\mathcal{F}}}$$

Aside. Uniform convergence





Two equivalent definitions of **uniform** convergence:

1. $\{f_n\}$ converges uniformly to f for $x \in X$ if for each $\epsilon > 0$, $\exists N$ indpt of x s.t. $n > N$ implies $|f_n(x) - f(x)| \leq \epsilon$.
2. $\{f_n\}$ converges uniformly to f for $x \in X$ if for each $\epsilon > 0$, $\exists N$ indpt of x s.t. $n > N$ implies $\sup_{x \in X} |f_n(x) - f(x)| \leq \epsilon$.

$f_n(x)$ converges **pointwise** to $f(x)$ for $x \in X$ if for each $\epsilon > 0$, $\exists N$ depending on x s.t. $n > N$ implies $|f_n(x) - f(x)| \leq \epsilon$.

Ex 1: $f_n(x) = x^n$, $x \in [0, 1]$. $\{f_n\}$ converges pointwise; not uniformly.

Ex2: $\{f_n\}$ continuous & converges uniformly to f. Then f is cont.

E.g $f_n(x) = \frac{1}{n(1+x^2)}$, $x \in \mathbb{R}$.

Result: Suppose $x^* = \text{argmin}\, f(x)$ and $x_n^* = \text{argmin} = f_n(x)$.
Then UC $\max_x |f_n(x) - f(x)| \leq \epsilon \implies |f(x^*) - f(x_n^*)| \leq 2\epsilon$.

Proof:

$$
\begin{aligned}
f(x_n^*) - f(x^*) &= f(x_n^*) - f_n(x_n^*) + f_n(x_n^*) - f(x^*) \\
&\leq f(x_n^*) - f_n(x_n^*) + f_n(x^*) - f(x^*) \\
|f(x_n^*) - f(x^*)| &\leq |f(x_n^*) - f_n(x_n^*)| + |f_n(x^*) - f(x^*)| \\
&\leq \epsilon + \epsilon
\end{aligned}
$$

Part I.5. Stochastic State Space Models

Continuous state Markov Chains. Consider stochastic difference equation:

$$x_{k+1} = \phi_k(x_k, w_k), \quad x_0 \sim \pi_0$$

State x_k lies in the state space $\mathcal{X} = \mathbb{R}^X$.
State noise $\{w_k\}$: iid sequence. Assume $\{w_k\}$, and x_0 are indpt.
Then x_k is a continuous state Markov process on $\mathbb{R}^X$:

$$\mathbb{P}(x_{n+1} \in S | x_1, x_2, \ldots, x_n) = \mathbb{P}(x_{n+1} \in S | x_n)$$

for any set $S \subseteq \mathbb{R}^X$.

Define *transition density* $P_{xy} = p(x_{k+1} = y | x_k = x)$: For $S \subseteq \mathcal{X}$,

$$\mathbb{P}(x_{k+1} \in S | x_k) = \int_S p(x_{k+1} = x | x_k)\, dx, \quad \int_{\mathcal{X}} p(x_{k+1} = x | x_k)\, dx = 1.$$

How to specify transition density? Use likelihood formula:
Suppose additive noise

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

Assuming $\Gamma_k(x_k)$ is square invertible matrix,

$$\boxed{p(x_{k+1}|x_k) = |\Gamma_k^{-1}(x_k)| \;\; p_w\left(\Gamma_k^{-1}(x_k)\; [x_{k+1} - A_k(x_k)]\right)}$$

Example: If $x_{k+1} = a x_k + w_k$, $w_k \sim N(0,1)$, transition density

$$P_{x_k, x_{k+1}} = p(x_{k+1}|x_k) \propto \frac{1}{\sqrt{2\pi}} \exp\Big(-\frac{1}{2}(x_{k+1} - a x_k)^2\Big)$$

Stochastic State Space Models

Difference Equation Form. Physics based model: Two random processes $\{x_k\}$ and $\{y_k\}$, $k = 0, 1, \ldots$:

$$\boxed{\begin{aligned} x_{k+1} &= A_k(x_k) + \Gamma_k(x_k)w_k, \quad x_0 \sim \pi_0 \\ y_k &= C_k(x_k) + D_k(x_k)v_k. \end{aligned}}$$

1. *state equation*: x_k lies in the state space $\mathcal{X} = \mathbb{R}^X$.
2. *observation equation*: nonlinear noisy sensor observes the state corrupted by measurement noise $\{v_k\}$.
   $y_k \in \mathbb{R}^Y$-dimensional. y_k is a doubly stochastic process.
3. state noise $\{w_k\} \sim p_w$: X-dimensional iid sequence of random variables. Assume $\{w_k\}$, $\{v_k\}$, x_0 are independent.
4. observation noise $\{v_k\} \sim p_v$: Y-dimensional i.i.d. sequence.

Transition Density Form. Partially observed Markov Model. Two random processes $\{x_k\}$, $\{y_k\}$, $k = 0, 1, \ldots$:

$$\boxed{\begin{aligned} &\text{State transition density: } p(x_{k+1}|x_k),\ x_0 \sim \pi_0, \\ &\text{Observation likelihood: } p(y_k|x_k). \end{aligned}}$$

State $\{x_k\}$: Markov process on $\mathbb{R}^X$ with initial state $x_0 \sim \pi_0$.

$$p(x_{k+1}|x_k, \ldots, x_0) = p(x_{k+1}|x_k) \qquad \text{(Markov dynamics)}$$

Observations $\{y_k\}$: assume *memoryless sensor*

$$p(y_k|x_k, x_{k-1}, \ldots, x_0) = p(y_k|x_k) \quad \text{(conditional indpt obs)}$$

Aside: Likelihood Formula

Suppose $Y = \phi(X) + W$, and W has cdf F_W and pdf f_W. Then

$$\boxed{f_{Y|X}(y|x) = f_W(y - \phi(x))}$$

Likelihood formula: $f_{Y|X}(y|x)$ depends on noise density f_W

Proof:

$$\begin{aligned} F_{Y|X}(y|x) = P(Y \le y|X = x) &= P(\phi(X) + W \le y|X = x) \\ &= P(W \le y - \phi(x)) = F_W(y - \phi(x)) \end{aligned}$$

Take derivative wrt y:

$$\boxed{f_{Y|X}(y|x) = \frac{dF}{dy} F_{Y|X}(y|x) = f_W(y - \phi(x))}$$

Example 1: $Y = X + W$ where $W \sim N(0, \sigma^2)$. Then

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

Example 2: $Y = AX + DW$ where $A, D \in \mathbb{R}^{n \times n}$ and D is positive definite. Show that $p(y|x) = \frac{1}{|D|} p_W(y - Ax)$

Example 3: Multiplicative noise $Y = XW$.

Soln: $F_{Y|X}(Y \le y|X = x) = P(XW \le y|X = x) = P(W \le y/x) = F_W(y/x)$ if $x > 0$. So

$$\boxed{f_{Y|X}(y|x) = \frac{1}{|x|} f_W\left(\frac{y}{x}\right)}$$

Stochastic Difference Equation to Transition Density

$$\begin{aligned} x_{k+1} &= A_k(x_k) + \Gamma_k(x_k) w_k, \quad x_0 \sim \pi_0 \\ y_k &= C_k(x_k) + D_k(x_k) v_k. \end{aligned} \tag{5}$$

Assume $\Gamma_k(x_k)$ and $D_k(x_k)$ are square invertible matrices.
Then transition density and observation likelihood are:

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

where $|\cdot|$ denotes determinant.

Example: Moving Autonomous Vehicle. Target moves in 2 dimensional space (ship, vehicle).
Linear Gaussian state space model

$$x_{k+1} = Ax_k + Bu_k + v_k$$

$x_k \stackrel{\text{defn}}{=} [r_x[k], \dot{r}_x[k], r_y[k], \dot{r}_y[k]]'$: state vector at time kT
T is the sampling interval.
$r_x(k)$ is x-coordinate position, $\dot{r}_x(k)$ is velocity in x direction.

$$A = \begin{pmatrix} 1 & T & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & T \\ 0 & 0 & 0 & 1 \end{pmatrix}, \qquad B = \begin{pmatrix} T^2/2 & 0 \\ T & 0 \\ 0 & T^2/2 \\ 0 & T \end{pmatrix}$$

$u_k = [u_k(1), u_k(2)]' \in \mathbb{R}^2$ models acceleration (maneuver) of the target. u_k changes infrequently (e.g. when target turns).

v_k iid noise process – models uncertainly in target dynamics

Model is obtained by discretizing continuous-time model:

$$\frac{dx_1}{dt} = x_2 \ \text{(velocity)}, \quad \frac{dx_2}{dt} = u_1 \ \text{(acceleration)}$$

Discretizing with sampling interval T yields:

$$x_{k+1}(1) = x_k(1) + \frac{T}{2}(x_{k+1}(2) + x_k(2))$$
$$x_{k+1}(2) = x_k(2) + Tu_k(1)$$

So first eqn becomes: $x_{k+1}(1) = x_k(1) + Tx_k(2) + \frac{T^2}{2}u_k(1)$.

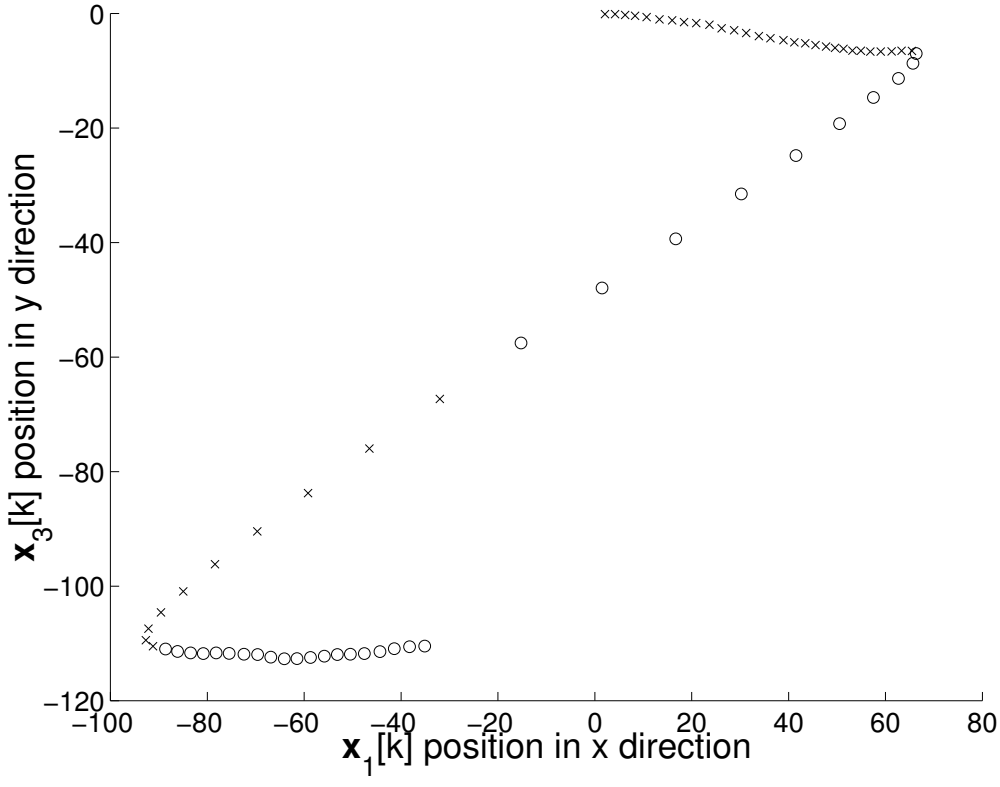


Figure 1: Manuevering Target e.g. ship

```
% matlab program to simulate a target
 T=1
 A=[1 T 0 0;0 1 0 0;0 0 1 T; 0 0 0 1];
 B=[T*T/2 0; T 0;0 T^2/2; 0 T]       ;
x(:,1)=[0 2 0 0]'     ;

for k=2:30,
  x(:,k) = A* x(:,k-1) + B * [0 0]' + .1*randn(4,1);
  plot(x(1,k),x(3,k),'x');
end;

for k=31:40,
  x(:,k) = A* x(:,k-1) + B * [-2 -1]' + .1*randn(4,1);
  plot(x(1,k),x(3,k),'o');
end;

for k=41:50,
  x(:,k) = A* x(:,k-1) + B * [2 1]' + .1*randn(4,1);
  plot(x(1,k),x(3,k),'x');
end;

for k=51:70,
  x(:,k) = A* x(:,k-1) + B * [0 0]' + .1*randn(4,1);
  plot(x(1,k),x(3,k),'o');
end;
```

Sensing the moving autonomous vehicle

1. Noisy observations of x_k are obtained at sensor (radar, sonar)

$$y_k = Cx_k + v_k, \quad v_k \sim \boldsymbol{N}(0, R).$$

where

$$x_k = [r_x[k], \dot{r}_x[k], r_y[k], \dot{r}_y[k]]'$$

Example 1. If sensor measures position, then $y_k \in \mathbb{R}^2$,

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

Example 2. Sophisticated sensors measure position and velocity observed (Doppler radar), $C = I_{4\times 4}$.

Bearings-only target tracking: only measure angle $\tan^{-1}(x_1[k]/x_3[k])$, i.e. nonlinear measurement equation.

2. Clutter: Often radar records false targets.

Homework: Read the paper "Survey of maneuvering target tracking. Part I. Dynamic models" by X Rong Li and V. Jilkov, IEEE Tans Aerospace and Electronic Systems, 2011.

Optimal Prediction: Chapman Kolmogorov Equation

Aim: How to optimally predict future state given current state probability?
Given Markov process with transition density $p(x_{k+1}|x_k)$ and initial condition π_0, compute state pdf $\pi_k(x) = p(x_k = x)$ at time k.

We call π_k the *predicted* density.

Chapman Kolmogorov: From total probability rule

$$\boxed{\pi_k(x) = \int_{\mathcal{X}} p(x_k = x|x_{k-1})\, \pi_{k-1}(x_{k-1}) dx_{k-1}, \quad \text{initialized by } \pi_0.}$$

Therefore predicted state and covariance at time k are

$$\boxed{\begin{aligned} \hat{x}_k = \mathbb{E}\{x_k\} &= \int_{\mathcal{X}} x\, \pi_k(x) dx, \\ \text{cov}(x_k) = \mathbb{E}\{(x_k - \hat{x}_k)(x_k - \hat{x}_k)'\} &= \mathbb{E}\{x_k x_k'\} - \hat{x}_k \hat{x}_k'. \end{aligned}}$$

Predicted mean is $\hat{x}_k$ is optimal in the minimum mean square error sense:

$$\mathbb{E}\{(x_k - \hat{x}_k)^2\} \leq \mathbb{E}\{(x_k - \phi(\pi_0))^2\}.$$

Hence called "optimal predictor".

Simulation-based Optimal State Predictor

Consider general stochastic state evolution model:

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

$$\implies p(x_{k+1}|x_k) = |\Gamma_k^{-1}(x_k)|\; p_w\left(\Gamma_k^{-1}(x_k)\,[x_{k+1} - A_k(x_k)]\right)$$

Chapman Kolmogorov equation for predicted state density:

$$\pi_k(x) = \int_{\mathcal{X}} p(x_k = x|x_{k-1})\, \pi_{k-1}(x_{k-1})\, dx_{k-1}.$$

Direct intergration is intractable. Can use stochastic simulation. Given random samples from the predicted state density $\pi_{k-1}(x)$ at time $k-1$, how to simulate samples from the predicted state density $\pi_k(x)$ at time k?

Solution. The composition method for stochastic simulation generates samples as follows:

1. Generate samples $x_{k-1}^{(l)}$, $l = 1, \ldots, L$ from $\pi_{k-1}(x)$.
2. Generate sample $x_k^{(l)} \sim p(x_k|x_{k-1} = x_{k-1}^{(l)})$, for $l = 1, 2, \ldots, L$.

By composition method, simulated samples $x_k^{(l)}, l = 1, \ldots, L$ are from pdf $\pi_k(x)$.

By Glivenko Cantelli Thm, for large L, these samples uniformly approximate pdf $\pi_k(x)$.

Then mean, variance, and other statistics can be estimated.

Ex 1. Linear Gaussian State Space Model

Notation:

$$\boldsymbol{N}(\zeta;\mu,\Sigma) = (2\,\pi)^{-l/2}\,|\Sigma|^{-1/2}\exp\left[-\frac{1}{2}(\zeta-\mu)'\Sigma^{-1}(\zeta-\mu)\right].$$

Sometimes shorter notation $\boldsymbol{N}(\mu,\Sigma)$ will be used.

The linear Gaussian state space model

$$\boxed{\begin{aligned} x_{k+1} &= A_k x_k + w_k, \quad x_0 \sim \pi_0 = \boldsymbol{N}(\hat{x}_0,\Sigma_0),\ w_k \sim \boldsymbol{N}(0,Q_k) \\ y_k &= C_k x_k + v_k, \quad v_k \sim \boldsymbol{N}(0,R_k). \end{aligned}}$$

$x_k \in \mathcal{X} = \mathbb{R}^X$, $y_k \in \mathcal{Y} = \mathbb{R}^Y$, $A_k \in \mathbb{R}^{X\times X}$, $C_k \in \mathbb{R}^{Y\times X}$.

Transition density form:

$$p(x_{k+1}|x_k) = p_w\left(x_{k+1} - A_k(x_k)\right) = \boldsymbol{N}(x_{k+1}; A_k x_k, Q_k)$$
$$p(y_k|x_k) = p_v\left(y_k - C_k(x_k)\right) = \boldsymbol{N}(y_k; C_k x_k, R_k).$$

Optimal Predictor Using the Chapman Kolmogorov equation $\pi_{k+1} = \boldsymbol{N}(\hat{x}_{k+1},\Sigma_{k+1})$ where

$$\boxed{\begin{aligned} \hat{x}_{k+1} &= \mathbb{E}\{x_{k+1}\} = A_k\hat{x}_k \\ \Sigma_{k+1} &= \mathrm{cov}\{x_{k+1}\} = A_k\Sigma_k A_k' + Q_k \\ \mathrm{cov}(x_{k+n},x_k) &= A^n\Sigma_k, \qquad n \geq 0 \end{aligned}}$$

Covariance update is called *Lyapunov equation.*
Same mean and covariance recursions hold for non-gaussian case

Motivation: Predicting target's coordinates (without measurements).

Proof. Evolution of covariance: Let $\tilde{x}_k = x_k - \hat{x}_k$ Then

$$\begin{aligned}
\tilde{x}_{k+1} &= A\tilde{x}_k + w_k \\
\tilde{x}_{k+1}\tilde{x}'_{k+1} &= (A\tilde{x}_k + w_k)(A\tilde{x}_k + w_k)' \\
&= A\tilde{x}_k\tilde{x}'_k A' + A\tilde{x}_k w'_k + w_k\tilde{x}'_k A' + w_k w'_k
\end{aligned}$$

Taking expectations on both sides yields result

Scalar Example In the scalar case

$$\begin{aligned}
\hat{x}_{k+1} &= A\hat{x}_k = A^{k+1}\hat{x}_0 \\
\Sigma_{k+1} &= A^2\Sigma_k + Q \\
&= A^{2k}\Sigma_0 + \frac{1-A^{2k}}{1-A^2}Q
\end{aligned}$$

$$\operatorname{cov}[x_k, x_l] = \begin{cases} A^{l-k}\Sigma_k & l \geq k \\ A^{k-l}\Sigma_k & l < k \end{cases}$$

So if $|A| < 1$, then as $k \to \infty$,

$$\hat{x}_k \to 0,\ \Sigma_k \to Q/(1-A^2),\ \operatorname{cov}(x_k, x_{\tau+k}) = Q\frac{A^\tau}{1-A^2}$$

Thus if $|A| < 1$, x_k becomes a weakly stationary process. This can be generalized to vector processes by requiring that the eigenvalues of A lie within the unit circle.

Stationary Distribution LTI Gaussian Markov model

Consider LTI Gaussian Markov process

$$x_{k+1} = Ax_k + w, \quad x_0 \sim \pi_0 = \boldsymbol{N}(\hat{x}_0, \Sigma_0),\ w \sim \boldsymbol{N}(0, Q)$$

Under reasonable conditions we expect as $k \to \infty$, $\hat{x}_k \to 0$ and asymptotic covariance satisfies algebraic Lyapunov equation:

$$\boxed{\Sigma = A\Sigma A' + Q}$$

Theorem: Suppose $Q = GG'$. If $|\lambda_i(A)| < 1$ and (A, G) is reachable, then $\Sigma > 0$ (pdf), exists and is unique.

Defn of Reachability: n dimensional state space model $x_{k+1} = Ax_k + Bu_k$ is reachable if given $x_0 = 0$, there exists a sequence $u_0, \ldots, u_{n-1}$ so that x_n takes arbitrary value. Equivalently if $\mathcal{C} = [B, AB, \ldots, A^{n-1}B]$ has full rank. Equivalently, $\sum_{i=0}^{n} A^i BB' A'^i$ is non-singular.

Result. Consider LTI Gaussian Markov process. Assume $|\lambda_i(A)| < 1$ and (A, G) is reachable. Then as $k \to \infty$, $x_k \sim \boldsymbol{N}(0, \Sigma_\infty)$ where Σ_∞ satisfies algebraic Lyapunov eq:

$$\Sigma = A\Sigma A' + Q$$

Above result does not apply to moving target model since A is unstable; $\lambda_i(A) = 1$.

Proof: Define $\bar{\Sigma} = \sum_{k=0}^{\infty} A^k \Sigma A'^k$. Since $|\lambda_i(A)| < 1$, $\bar{\Sigma}$ exists. Also $\bar{\Sigma} > 0$ since $\bar{\Sigma} > \sum_{k=0}^{n} A^k G G' A'^k > 0$ (last inequality follows from reachability assumption).

Next note $\bar{\Sigma}$ satisfies same Liapunov equation $\bar{\Sigma} = A\bar{\Sigma}A' + \Sigma$. So $\bar{\Sigma} > 0$ is a pdf solution that exists.

Uniqueness: Assume $\tilde{\Sigma}$ is another soln. Then prove that $\tilde{\Sigma} = \Sigma$. (Homework).

How to solve Algebraic Lyapunov Equation?

$$\Sigma = A\Sigma A' + Q$$

Solve linear system

$$(I - A \otimes A)\text{vec}(\Sigma) = \text{vec}(Q)$$

Note eigenvalues of $(I - A \otimes A)$ are $1 - \lambda_i \lambda_j$. Linear system is solvable if $1 - \lambda_i \lambda_j \neq 0 \quad \forall i, j$. Sufficient condition is $|\lambda_i| < 1$.

Part I.6. Linear Time Series

Stationarv Stochastic Processes

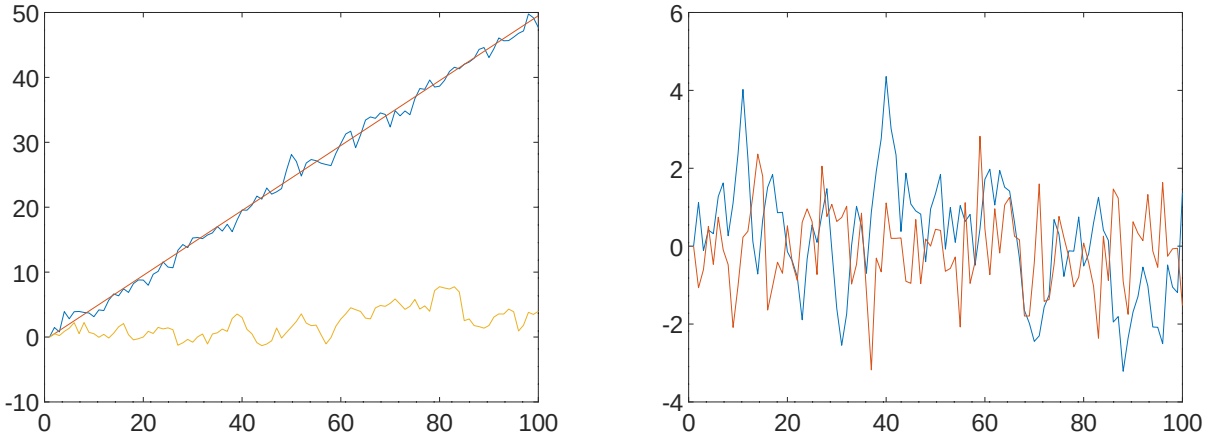


Fig 1:nonstationary time series $x_k = 0.5x_{k-1} + w_k + 0.25\,k$ and $x_k = 1.01x_{k-1} + w_k$.

Fig.2 stationary time series $x_k = ax_{k-1} + w_k$ where $a = 0.8$ and $a = 0.2$.

(i) **Strongly (Strictly) Stationary** (ss): Pick any N time instants $t_1, t_2, \ldots, t_N$. If
$F(x(t_1), x(t_2), \ldots, x(t_N)) = F(x(t_1+\tau), \ldots, x(t_N+\tau))$ for all τ, N and $t_1, t_2, \ldots, t_N$ then $X(t)$ is ss.
So joint cdf is time-shift invariant.

So $F(x(t_1)) = F((x(0)) \implies \mu(t) = \mathbb{E}\{X(t)\} =$ constant.
$F(x(t_1), x(t_2)) = F(x(0), x(|t_2 - t_1|))$. So $\text{cov}(X(t_1), X(t_2) = C(|t_1 - t_2|)$
$F(x(t_1), x(t_2), x(t_3)) = F(x(0), x(t_2 - t_1), x(t_3 - t_1))$, etc

Example 1: Show that the iid process is ss.
Soln: Joint distribution at N time instants $t_1, \ldots, t_N$ is

$F(x(t_1) = x_1, \ldots, x(t_N) = x_N) = F(x_1)F(x_2)\cdots F(x_N)$ time indpt

Example 2: Suppose X_k is iid with mean μ and variance σ^2. Is $S_k = X_1 + X_2 + \cdots X_k$ ss?
Soln: $\mathbb{E}\{S_k\} = k\mu$ and $\text{var}(S_k) = k\sigma^2$. Since mean and variance are time varying not ss.

(ii) **Weak (Wide-sense) Stationary** (ws): If:

1. $\mu(t) = \mathbb{E}\{X(t)\}$ is a constant, i.e., indpt of time.
2. $C(t_1, t_2) = \text{cov}(X(t_1), X(t_2)) = C(|t_1 - t_2|)$

For ws process $C(t, t+\tau) = C(\tau)$; function of one variable.

Remarks:
1. SS implies ws. For Gaussian processes, equivalent.
2. Variance is an even function of time t since

$$\text{var}(X(t+\tau), X(t)) = \text{var}(X(t), X(t+\tau))$$

Example: Suppose $X[k] \in \{-1, +1\}$ with prob $1/2$ if k is even.
$X[k] \in \{1/3, -3\}$ with prob $9/10$ and $1/10$ if k is odd.
Then $X[k]$ is not ss since its density varies with k.
But $\mathbb{E}\{X[k]\} = 0$ for all k. Also

$$\text{cov}(X[k_1], X[k_2]) = \begin{cases} 0 & \text{for } k_1 \neq k_2 \\ 1 & \text{if } k_1 = k_2 \end{cases}$$

Therefore $X[k]$ is ws.

Spectral Density: Spectral density is Fourier transform of autocorrelation function of a weakly stationary process.

In continuous-time: If ws process $X(t)$ has autocorrelation $R(t) = \mathbb{E}\{X(\tau)X(\tau+t)\}$, then spectral density

$$\phi_X(w) = \int_{-\infty}^{\infty} R(t)e^{-jwt}dt$$

where w is angular frequency.

In discrete-time: If ws process $X[k]$ has autocorrelation $R[k]$ then spectral density is Discrete Fourier Transform (DFT)

$$\phi_X(e^{jw}) = \sum_{k=-\infty}^{\infty} R[k]e^{-jwk}, \quad R[k] = \mathbb{E}\{X[n]X[n+k]\}$$

An important result for signal to noise ratio calculations

Parseval's Theorem: Average power in frequency band $[w_1, w_2]$ is

$$P_X = \frac{1}{2\pi}\int_{w_1}^{w_2} \phi_X(e^{jw})dw$$

Homework: Work thru the example in
`https://www.mathworks.com/help/signal/ug/power-spectral-density-estimates-using-fft.html`

Discrete-time White Noise: $e[n]$ is white noise if:
(i) $\mathbb{E}\{e[n]\} = 0$. (ii) $e[n]$ and $e[m]$ are indpt rvs for $n \neq m$.

Covariance function of white noise is

$$C[n] = \sigma^2 \delta[n] = \begin{cases} \sigma^2 & \text{if } n = 0 \\ 0 & n \neq 0 \end{cases}$$

So spectral density $\phi(w) = \sigma^2$ is constant for all frequencies w – white noise is "wideband noise". Its power is finite:

$$\frac{1}{2\pi} \int_{-\pi}^{\pi} \phi_X(e^{jw}) dw = \sigma^2$$

Analogy: spectral property of white light.
Cont-time white noise does not exist. Its integral is Brownian motion (Wiener process) which is a fractal function of unbounded variation.

ARMA Time Series Models

ARMA = Auto-regressive Moving Average. These are input/output models.

1. MA Processes (Finite Impulse Response Filter)

$$y[k] = b_1 u[k-1] + b_2 u[k-2] + \cdots + b_n u[k-n] + w[k]$$

Transfer function:

$$\frac{Y(z^{-1})}{U(z^{-1})} = b_1 z^{-1} + b_2 z^{-2} + \cdots + b_n z^{-n}$$

Example: Modelling of Communication Channels:

$$y[k] = b_1 u[k-1] + b_2 u[k-2] + \cdots + b_n u[k-n] + w[k]$$

$u[k] \in \{-1, 1\}$ denotes digital input at time k.
$w[k]$ denotes channel noise
$b_2 u[k-2] + \cdots + b_n u[k-n]$ is called ISI (inter-symbol interference)
Equalization/Deconvolution Problem: Send a known test sequence $\{u[1], \ldots, u[T]\}$ and obtain output $\{y[1], \ldots, y[T]\}$. Given $\{u[1], \ldots, u[T]\}$ and $\{y[1], \ldots, y[T]\}$, a channel equalizer computes channel parameter $b_1, \ldots, b_n$.
We will show how to use *Least Squares Estimation* for this

2. AR Processes: (Infinite Impulse Response)

$$y[k] = a_1 y[k-1] + \dots a_m y[k-m] + w[k]$$

Transfer function (all pole model)

$$\frac{Y(z^{-1})}{U(z^{-1})} = \frac{z^{m-1}}{z^m - a_1 z^{m-1} - a_2 z^{m-2} - a_m}$$

Weakly Stationary if roots of $z^m - a_1 z^{m-1} - a_2 z^{m-2} - a_m$ lie inside unit circle.

Example: Linear Predictive Coding (LPC) of Speech.

$$y[k] = a_1 y[k-1] + \dots a_m y[k-m] + w[k]$$

---

3. ARMA Processes:

$$y[k] = a_1 y[k-1] + \cdots + a_m y[k-m] + b_1 u[k-1] + \cdots b_n u[k-n]$$

Widely used in econometrics (market cycles), general comms channels, adaptive control (ARMAX models).

4. GARCH. Generalized auto-regressive conditional hetroskedastic process x_k: Engle [2003 Nobel prize]

$$\sigma_k^2 = \alpha_0 + \alpha_1 x_{k-1}^2 + \alpha_2 x_{k-2}^2 + \beta_1 \sigma_{k-1}^2 + \beta_2 \sigma_{k-2}^2$$

$$x_k = \eta_k \sigma_k, \quad \eta_k \sim N(0,1)$$

$$\alpha_i, \beta_i \in \mathbb{R}_+$$

Final Remarks: 1. Convert between state space model and ARMA model: Given $x[k] \in \mathbb{R}^n$

$$x[k+1] = A_{n\times n}x[k] + B_{n\times 1}u[k]$$
$$y[k] = C'x[k]$$

$$\text{Transfer function:} \qquad \frac{Y(z^{-1})}{U(z^{-1})} = C'(zI - A)^{-1}B$$

From transfer function the ARMA model can be read off.
Given ARMA model: infinite equivalent state space models: diagonal form, controller canonical form (realization theory)

Example: Given transfer function

$$\frac{Y(z^{-1})}{U(z^{-1})} = \frac{b_1 z^{-1} + b_2 z^{-2}}{1 + a_1 z^{-1} + a_2 z^{-2}}$$

or equivalently:

$$y[k] + a_1\, y[k-1] + a_2\, y[k-2] = b_1\, u[k-1] + b_2\, u[k-2]$$

State space model: Controller canonical form is

$$x[k+1] = \begin{bmatrix} -a_1 & -a_2 \\ 1 & 0 \end{bmatrix} x[k] + \begin{bmatrix} 1 \\ 0 \end{bmatrix} u[k]$$
$$y[k] = \begin{bmatrix} b_1 & b_2 \end{bmatrix} x[k]$$

Observer canonical form is

$$x[k+1] = \begin{bmatrix} -a_1 & 1 \\ -a_2 & 0 \end{bmatrix} x[k] + \begin{bmatrix} b_1 \\ b_2 \end{bmatrix} u[k]$$
$$y[k] = \begin{bmatrix} 1 & 0 \end{bmatrix} x[k]$$

2. State Space Models such as

$$x[k+1] = Ax[k] + Bu[k] + w[k]$$
$$y[k] = Cx[k] + v[k]$$

are useful for **state estimation**. Given noisy observations $y[1], \ldots, y[k]$ how to estimate the state $x[k]$? Kalman filter

3. State Space Models are not unique. There are a continuum of state space models that give the same input/output response.

$$x[k+1] = A_{n\times n}x[k] + B_{n\times 1}u[k]$$
$$y[k] = C'x[k]$$

Choose new state $\bar{x}[k] = Tx[k]$ for any invertible matrix T.
Then can write equivalent model as

$$\bar{x}[k+1] = \bar{A}_{n\times n}\bar{x}[k] + \bar{B}_{n\times 1}u[k]$$
$$y[k] = \bar{C}'\bar{x}[k]$$

where $\bar{A} = TAT^{-1}$, $\bar{B} = TB$, $\bar{C}' = C'T^{-1}$ (similarity transformation).
Example: choose T^{-1} as matrix of eigenvectors of A.

4. Input output models e.g. ARMA models such as

$$y[k] = a_1 y[k-1] + \cdots + a_m y[k-m] + b_1 u[k-1] + \cdots b_n u[k-n]$$

useful in **parameter estimation**: given $u[1], \ldots, u[k]$ and $y[1], \ldots y[k]$ estimate parameters $[a_1, a_2, \ldots, a_m, b_1, \ldots, b_n]$.

ARMA representation requires $(m+n)$ parameters whereas general state space requires (m^2+m+n) parameters in general.
If numerator & denominator polynomials are coprime, ARMA unique

I.7: Martingales (Advanced)

Martingales are more general than iid.

Definition. Suppose $\mathcal{F}_n = (X_1, \ldots, X_n)$. Then $\{Z_n\}$ is a mtg process wrt $\mathcal{F}_n$ if $\mathbb{E}\{Z_{n+1}|\mathcal{F}_n\} = Z_n$ and $\mathbb{E}|Z_n| < \infty$.

Therefore mtg has constant mean: $\mathbb{E}\{Z_{n+1}\} = \mathbb{E}\{Z_n\} = \mathbb{E}\{Z_0\}$.

If $\mathbb{E}\{Z_{n+1}|\mathcal{F}_n\} \leq Z_n$ then $\{Z_n\}$ is a supermartingale

Martingales are useful (ML, finance) because

1. Martingale representation theorem: Every Markov process can be decomposed into martingales.
   Example. Doob Decomposition. A random process $\{X_n\}$ can be decomposed into $X_n = A_n + Z_n$ where A_n is predictable and Z_n is a mtg.
   How? Clearly $M_n = X_n - \mathbb{E}\{X_n|\mathcal{F}_{n-1}\}$ is a mtg inc.
   So $Z_n = X_0 + \sum_{k=1}^n M_k$ is a mtg
   Define $\Delta_k = \mathbb{E}\{X_k|\mathcal{F}_{k-1}\} - X_{k-1}$. Then $A_n = \sum_{k=1}^n \Delta_k$ predictable.
   Therefore $X_n = Z_n + A_n$
   Example. Doob-Meyer Decomposition. Supermtg = sum of mtg and increasing predictable process.
2. Martingale convergence theorems: If Z_k is a martingale and $\sup_k \mathbb{E}|Z_k| < \infty$, then Z_k converges to a rv a.s.
3. Martingale statistical inference theorems: Law of large numbers and central limit theorem.
4. Probabilistic Combinatorics (concentration inequalities in machine learning)

Martingale Representation of Markov Chain

Suppose $\mathcal{X} = \{e_1, e_2, \ldots, e_X\}$. Then Markov chain can be expressed as linear difference equation

$$x_{k+1} = P' x_k + M_{k+1}, \qquad \text{where } \mathbb{E}\{M_{k+1} | X_1, \ldots, X_k\} = 0$$

M_k is a martingale increment, i.e., $Z_k = \sum_{n=1}^{k} M_n$ is a martingale process:

$$\mathbb{E}\{Z_{k+1} | Z_1, \ldots, Z_k\} = Z_k$$

Recall that a martingale has constant mean.

Optional stopping theorem. Suppose $T = \phi(\mathcal{F}_n)$ be a random finite (stopping) time and Z_n is a $\mathcal{F}_n$ mtg. Then

$$\mathbb{E}\{Z_T\} = \mathbb{E}\{Z_0\}$$

Examples of martingales (Advanced)

(i) Random walk: $\{X_n\}$ are iid zero mean. Then
$Z_n = \sum_{k=1}^n X_n$ is a mtg.
If X_n are iid with mean 1, then $Z_n = \prod_{k=1}^n X_k$ is a mtg
(iii) Likelihood Ratio Martingales: Suppose X_k iid with pdf g.

$$\text{Then } Z_n = \prod_{k=1}^n \frac{f(X_k)}{g(X_k)}, k \geq 1 \text{ is a mtg}$$

Clearly

$$\mathbb{E}\{Z_{n+1}|X_{1:n}\} = \mathbb{E}\{Z_n \frac{f(X_{n+1})}{g(X_{n+1})} \mid X_{1:n}\} = Z_n \int \frac{f(x)}{g(x)} g(x)dx = Z_n$$

(iii) Doob Martingale. Given a sequence of rvs $\{X_n\}$, let
$Z_k = \mathbb{E}\{Y|X_1, \ldots, X_k\}$. Then Z_k is a mtg wrt $X_1, \ldots, X_k$.
Localization estimate $\mathbb{E}\{\theta|X_1, \ldots, X_k\}$ is a Doob mtg.

Example: Edge Exposure Martingale in random graph. Suppose $G \in G_{n,p}$ is an Erdos Renyi random graph. How to estimate $F(G)$? number of connected components, chromatic number.

Label the $\binom{n}{2}$ edges as $1, \ldots, m$.
Define $Z_i = \mathbb{E}\{F(G)|X_1, \ldots, X_i\}$, $Z_0 = \mathbb{E}\{F(G)\}$. Then Z_i is a Doob mtg called the edge exposure mtg.
Can estimate by revealing edges one by one.
Use Azuma-Hoeffding to ascertain accuracy of estimate.

(iv) *Wald's Equation.* Suppose $\{X_n\}$ is iid and T is a stopping time. Then

$$\mathbb{E}\{\sum_{k=1}^{T} X_k\} = \mathbb{E}\{T\}\,\mathbb{E}\{X_1\}$$

Ex. $X_n \in \{-1, 1\}$ iid symmetric. Let $T = \min\{n : X_n = 1\}$. What is $\mathbb{E}\{S_T\} = \mathbb{E}\{\sum_{k=1}^{T} X_k\}$?

Intuition: Since we stop after $+1$, $\mathbb{E}\{S_T\} > 0$. This is incorrect.

From Wald Eqn: $\mathbb{E}\{S_T\} = 0$.

(iv) **Bertrand's Ballot Theorem**. A and B contest an election. A receives a votes, B receives b votes and $a > b$. Compute prob that while counting votes, A remains always ahead of B?
Ans:

$$\frac{a-b}{a+b}$$

Let S_k be difference in votes for A vs B after k votes counted.
So $S_n = a - b$ where n is total # votes.
Define $Z_k = S_{n-k}/(n-k)$. Show that Z_k is a mtg.
Define stopping time

$$T = \min\{k : Z_k = 0\}, \quad \text{or } T = n-1 \text{ otherwise}$$

2 possibilities:
Case 1. A is always ahead: $T = n-1$, $Z_T = Z_{T-1} = S_1 = 1$.
Case 2. A is not always ahead. Then at some point Z_k must be zero; so $Z_T = 0$.
So $\mathbb{E}\{Z_T\} = p \times 1 + (1-p) \times 0$. By optional sampling theorem:

$$p = \mathbb{E}\{Z_T\} = \mathbb{E}\{Z_0\} = \mathbb{E}\{S_n/n\} = \frac{a-b}{a+b}$$

Doob Martingale inequality: Stronger than Chebyshev

$$P(\max_{1\le k\le n}|Z_k|\ge\lambda)\le\frac{\mathbb{E}\{Z_k^2\}}{\lambda^2}$$

Azuma-Hoeffding inequality: Suppose $Z_n=\sum_{k=1}^n M_k+Z_0$ where $\{M_k\}$ is a martingale difference process with bounded differences satisfying $|M_k|\le\Delta_k$ almost surely where Δ_k are finite constants. Then for any $\epsilon>0$

$$\mathbb{P}(Z_n-Z_0\ge\epsilon)\le\exp\left(-\frac{\epsilon^2}{2\sum_{k=1}^n\Delta_k^2}\right)$$

$$\mathbb{P}(Z_n-Z_0\le-\epsilon)\le\exp\left(-\frac{\epsilon^2}{2\sum_{k=1}^n\Delta_k^2}\right).$$

Therefore

$$\boxed{\mathbb{P}(|Z_n-Z_0|\ge\epsilon)\le 2\exp\left(-\frac{\epsilon^2}{2\sum_{k=1}^n\Delta_k^2}\right).}$$

Example: Suppose $M_k\in\{-1,1\}$ are iid. Then $Z_n=\sum_{k=0}^n M_k$ is a mtg. Also $|M_k|\le 1$. So

$$P(Z_n\ge\epsilon)\le\exp(-\frac{\epsilon^2}{2n})$$

Equivalently the empirical mean satisfies

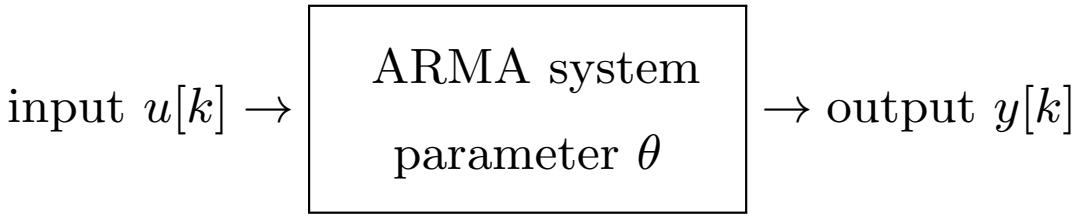
$$P(Z_n/n\ge\epsilon)\le\exp(-\frac{n\epsilon^2}{2})$$

Part II Least Squares Estimation

Outline

1. Classical Least Squares and Computational Least Squares
2. Principal Component Analysis (PCA).
3. Recursive Least Squares
4. Stochastic Least Squares
5. Bias-Variance Tradeoff and Model Order Estimation
6. LASSO and Sparsity

System Identification (Parameter Estimation) Problem



We consider LTI stochastic models

1. θ denotes parameter vector. e.g in Stochastic ARMA model

   $$y[k] = a_1 y[k-1] + a_2 y[k-2] + b_1 u[k-1] + b_2 u[k-2] + e[k]$$

   $\theta = [a_1,\ a_2,\ b_1,\ b_2]'$.
   (states are different to parameters)

2. Assume existence of true parameter θ^0
   Otherwise mis-specified problem (harder to deal with)

3. equation error model: additive output noise
   (more complicated model: errors-in-variables)

Aim: Given data (measurements)
$(u[1], u[2], \ldots, u[N], y[1], \ldots, y[N])$ estimate θ

Typically number of data points N available is large

- Off-line Estimation: Given fixed data set
  $(u[1], u[2], \ldots, u[N], y[1], \ldots, y[N])$ estimate θ
- Recursive Estimation: Recursively update θ with each new observation $u[k], y[k]$

Remarks: We consider ARMA models because they are *minimal* representations of LTI systems.

For n-th order ARMA model

$$y[k] = \sum_{i=1}^{n} a_i y[k-i] + b_i u[k-i]$$

$\theta = [a_1, \ldots, a_n, b_1, \ldots, b_n]'$ has $2n$ parameters.

Equivalent state space model

$$\begin{aligned} x[k+1] &= A_{(n\times n)} x[k] + B_{(n\times 1)} u[k] \\ y[k] &= C_{(1\times n)} x[k] \end{aligned}$$

has $n^2 + 2n$ parameters in general.

Least Squares Estimation of Static Systems

Estimate parameters of ARMA (**dynamical**) system.
To start consider least squares estimation of a **static** system.

$$\boxed{\text{Model. } Y_{N\times 1} = \Psi_{N\times n}\theta_{n\times 1}}$$

Given output $Y \in \mathbb{R}^N$ and input matrix Ψ, estimate $\theta \in \mathbb{R}^n$.

There are 3 cases:

1. Under-determined Case: $N < n$.

$$\text{Example. } y_1 = \begin{bmatrix} x_1 & 1 \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix}$$

Infinite number of solutions. Later, we consider sparse solutions.

2. Fully-determined Case: $N = n$

$$\text{Example. } \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} x_1 & 1 \\ x_2 & 1 \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix}$$

Unique solution if $\Psi_{n\times n}$ is invertible: $\theta = \Psi^{-1}Y$

3. Over-determined case: $N > n$.

$$\text{Example. } \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} = \begin{bmatrix} x_1 & 1 \\ x_2 & 1 \\ x_3 & 1 \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix}$$

In general no solution. Least Squares estimation applies to overdetermined case.

Least Squares solution applies to overdetermined case.

Step 1: Error Equation Formulation:

$$\boxed{Y_{N\times 1} = \Psi_{N\times n}\theta_{n\times 1} + \epsilon_{N\times 1}}, \qquad N > n$$

The vector ϵ are the additive measurement errors.

Step 2: Least Squares Cost Formulation:

$$\boxed{\theta_{\text{LS}} = \arg\min_{\theta\in\mathbb{R}^n} J(\theta)}$$

where $\theta \in \mathbb{R}^n$, $\Psi \in \mathbb{R}^{N\times n}$, $\epsilon \in \mathbb{R}^N$

$$\boxed{J(\theta) = \sum_{i=1}^{N} e_i^2 = \epsilon'\epsilon = (Y - \Psi\theta)'(Y - \Psi\theta)}$$

Step 3: Compute Least Square Solution Compute $\theta_{\text{LS}} = \arg\min_\theta J(\theta)$ where

$$J(\theta) = (Y - \Psi\theta)'(Y - \Psi\theta) = Y'Y - Y'\Psi\theta - \theta'\Psi'Y + \theta'\Psi'\Psi\theta$$

We need to solve $\partial J/\partial\theta = 0$.

Aside: How to compute $\frac{d}{dx}x'y$ where $x, y \in \mathbb{R}^n$?
Solution: $\frac{d}{dx}x'y = y$ (check this element-wise).
Also $\frac{d}{dx}x'Ax = Ax + A'x$.

$$\frac{\partial J}{\partial\theta} = -\Psi'Y - \Psi'Y + 2\Psi'\Psi\theta = 0$$

Therefore we have "Normal Equations" $\Psi'\Psi\theta_{\text{LS}} = \Psi'Y$.
If $\Psi'\Psi$ is invertible then least square soln is

$$\boxed{\theta_{\text{LS}} = (\Psi'\Psi)^{-1}\Psi'Y}$$

Example 1: Back to least squares example of fitting a line

$$y = ax + b$$

thru 3 points (x_1, y_1), (x_2, y_2), (x_3, y_3).

Solution: Using notation of Step 2 we have $n = 2$ (number of parameters), $N = 3$ (number of data points),

$$Y = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix}, \quad \Psi = \begin{bmatrix} x_1 & 1 \\ x_2 & 1 \\ x_3 & 1 \end{bmatrix}$$

Then from least squares solution of Step 3,

$$\theta_{\text{LS}} = \begin{bmatrix} a_{\text{LS}} \\ b_{\text{LS}} \end{bmatrix} = (\Psi'\Psi)^{-1}\Psi' Y$$

Example 2: Least squares formula for a quadratic function: $y = ax^2 + bx + c$ given points (x_i, y_i), $i = 1, \ldots, N$, $N \geq 4$.

$\theta = [a, b, c]'$ (so $n = 3$ parameters).

$$Y = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{bmatrix}, \Psi = \begin{bmatrix} x_1^2 & x_1 & 1 \\ x_2^2 & x_2 & 1 \\ \vdots & \vdots & \vdots \\ x_N^2 & x_N & 1 \end{bmatrix}$$

Remarks: $\theta_{\text{LS}} = (\Psi'\Psi)^{-1}\Psi' Y$.

1. Note $\Psi \in \mathbb{R}^{N\times n}$. Thus $\Psi'\Psi \in \mathbb{R}^{n\times n}$ and $(\Psi'\Psi)^{-1}\Psi' \in \mathbb{R}^{n\times N}$ $(\Psi'\Psi)^{-1}\Psi'$ is called the *pseudo-inverse* of matrix Ψ.

If Ψ is a square matrix, i.e. $N = n$ (fully determined case) then

$$(\Psi'\Psi)^{-1}\Psi' = \Psi^{-1}\Psi'^{-1}\Psi' = \Psi^{-1}$$

Thus if Ψ is square matrix, psuedo inverse = standard inverse.

2. Gauss used LS in the 1800s to predict orbits of planets.

3. Least squares estimation of the parameters of a ARMA model (dynamical system) is identical! – notation is more complicated.

4. *Least Absolute Deviations*: minimize sum of absolute errors

$$\theta^* = \operatorname*{argmin}_{\theta} \sum_{k=1}^{N} |y_k - \psi_k'\theta|$$

Robust to outliers; but soln may not be unique.
Note $\min \sum_{k=1}^N |y_k - \psi'\theta| = \min \sum_k \alpha_k : \alpha_k \geq |y_k - \psi'\theta|$.
So equivalent to solving linear program:

$$\min_{\alpha,\theta} \sum_{k=1}^{N} \alpha_k \text{ subject to } 2N \text{ constraints}$$

$$\alpha_k \geq y_k - \psi_k'\theta, \quad \alpha_k \geq -(y_k - \psi_k'\theta), \quad k = 1, \ldots, N$$

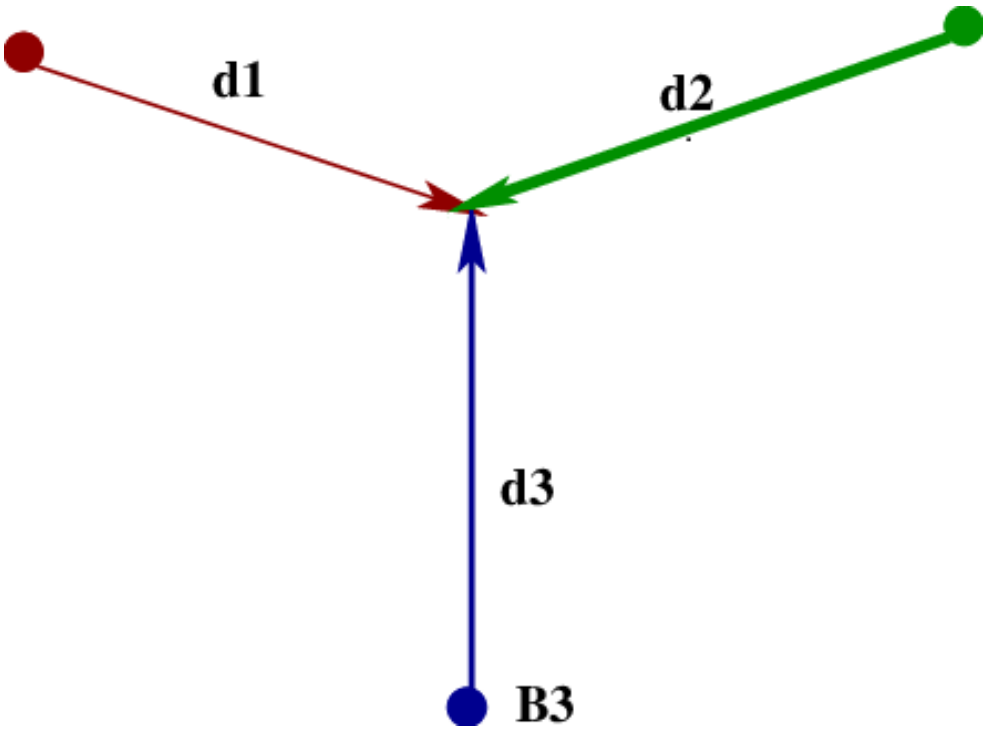
5. Toy example: If $y_1, \ldots, y_N$ are measurements of $\theta \in \mathbb{R}$, then what is least squares estimate? What is least absolute deviation estimate?

6. *Nonlinear Least Squares*: Localization given range measurements.
θ^* denotes unknown location in 3-D.
We measure distance to known points $d_1, \ldots, d_m$:

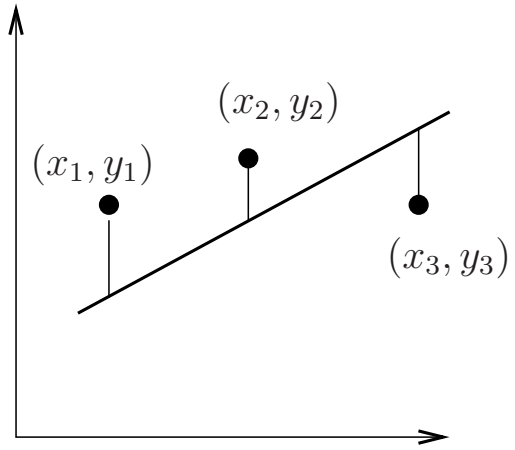
$$y_i = \|\theta^* - d_i\| + \epsilon_i, \quad i = 1, \ldots, m$$

where ϵ_i are measurement errors.

Aim: Compute $\min_\theta \sum_{i=1}^m (\|\theta - d_i\| - y_i)^2$
In general nonlinear LS is a non-convex problem and difficult to solve.

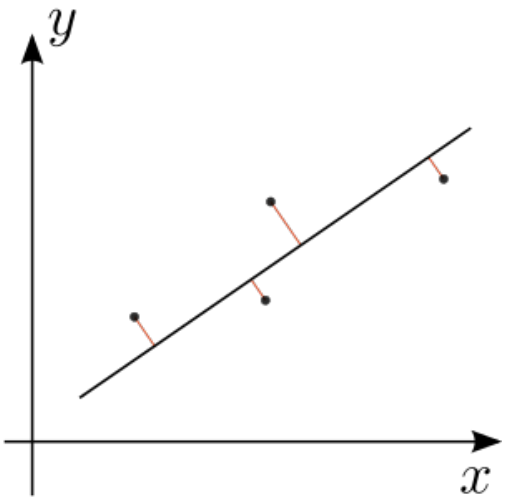


Least Squares vs Total Least Squares



Classical LS: $Y = \Psi\theta + \epsilon$

LS estimate is: $\theta^* = \text{argmin}_\theta \|\epsilon\|^2$ subject to $Y = \Psi\theta + \epsilon$.



Total LS: $Y = [\Psi + \tilde{\Psi}]\theta + \epsilon$

TLS estimate is:

$\theta^* = \text{argmin}_\theta \sum_{i=1}^n \|\tilde{\Psi}_i\|^2 + \|\epsilon\|^2$ subject to $Y = [\Psi + \tilde{\Psi}]\theta + \epsilon$

Convex optimization problem solved via SVD.

Frobenius norm $\|A\|_F = \sum_i \sum_j A_{ij}^2 = \text{trace} AA' = \sum_i \sigma_i^2(A)$.

$$\| \left[\tilde{\Psi}, \epsilon\right] \|_F = \sum_{i=1}^n \|\tilde{\Psi}_i\|^2 + \|\epsilon\|^2$$

Least Squares Identification of ARMA Models

Assume true system is

$$y[k] = \sum_{i=1}^{n} a_i^0 y[k-i] + b_i^0 u[k-i]$$

with true parameter (dimension $2n$)

$$\theta^0 = [a_1^0, \ldots, a_n^0, b_1^0, \ldots, b_n^0]$$

Assume we observe set of inputs and outputs

$$\{u[0], u[1], \ldots, u[N], y[1], \ldots, y[N]\}$$

Aim: Compute θLS As in the static case, follow 3 steps:
Step 1: Error Equation Formulation: For
$k = n, n+1, \ldots, N$,

$$y[k] = \sum_{i=1}^{n} a_i y[k-i] + b_i u[k-i] + e[k;\theta]$$

In matrix vector notation: Define *regression vector*

$$\psi[k] = [y[k-1], y[k-2], \ldots, y[k-n], u[k-1], \ldots, u[k-n]]'$$

Then we have for $k = n, n+1, \ldots, N$

$$y[k] = \psi[k]'\theta + e[k;\theta]$$

Define $Y_N = [y[n], \ldots, y[N]]' \in \mathbb{R}^{N-n+1}$

$$\Psi_N = [\psi[n], \ldots, \psi[N]]' \in \mathbb{R}^{(N-n+1)\times 2n}$$

$$\epsilon_{N;\theta} = [e[n;\theta], \ldots, e[N;\theta]]' \in \mathbb{R}^{N-1+1}$$

Then the error eqn formulation is

$$\boxed{Y_N = \Psi_N \theta + \epsilon_{N;\theta}}$$

which is identical to static case.

Step 2: Least Squares Cost Formulation:

$$\boxed{J(\theta) = \sum_{k=n}^{N} e^2[k;\theta] = \epsilon'_{N;\theta}\epsilon_{N;\theta} = (Y_N - \Psi_N\theta)'(Y_N - \Psi_N\theta)}$$

Aim: Compute $\theta_{\text{LS}} = \arg\min_\theta J(\theta)$

Step 3: Compute Least Squares Estimate:

As in static case setting $\partial J/\partial\theta = 0$ gives

$$\boxed{\Psi'_N\Psi_N\theta_{\text{LS}} = \Psi'_N Y_N} \quad \text{“Normal Equations”}$$

If $(2n \times 2n)$ matrix $\Psi'_N\Psi_N$ is invertible then

$$\boxed{\theta_{\text{LS}} = (\Psi'\Psi)^{-1}\Psi' Y}$$

Note: $\Psi'\Psi = (\sum_{k=n}^{N} \psi_k\psi'_k)^{-1}$ and $\Psi' Y = \sum_{k=n}^{N} \psi_k y_k$.

$$\text{Thus } \theta_{\text{LS}} = \left[\sum_{k=n}^{N} \psi_k\psi'_k\right]^{-1} \left[\sum_{k=n}^{N} \psi_k y_k\right]$$

Persistent Exciting inputs

Recall we need $\Psi'\Psi$ to be invertible. But $\Psi = [\psi[n], \ldots, \psi[N]]'$

$$\psi[k] = [y[k-1], y[k-2], \ldots, y[k-n], u[k-1], \ldots, u[k-n]]'$$

Invertibility of $\Psi'\Psi$ depends on input sequence $u_0, \ldots, u_{N-1}$. What inputs $u[k]$ ensures existence of least squares estimate?

Result: $\Psi'\Psi$ is invertible iff columns of Ψ are linearly indpt.

Recall for linear algebra: The columns of a matrix Ψ are linearly independent if for all vectors $x \neq 0$, $\Psi x \neq 0$

Example: Suppose

$$y[k] = ay[k-1] + b_1 u[k] + b_2 u[k-1] + e[k]$$

If input $u[k] = c$ for all k, then

$$\Psi = \begin{bmatrix} y[0] & u[0] = c & u[1] = c \\ y[1] & u[1] = c & u[2] = c \\ \vdots & \vdots & \vdots \\ y[N-1] & u[N-1] = c & u[N-2] = c \end{bmatrix}$$

Thus $\Psi'\Psi$ is not invertible.

Main Point: For step input it is not possible to estimate b_1, b_2. We need $u[k]$ to vary sufficiently to excite all modes of system.

Persisitently exciting input: $u[k]$ fluctuates enough so that columns of Ψ are linearly indpt.

Proof of Result:

(i) To show: cols of Ψ lin indpt $\implies$ $\Psi'\Psi$ invertible

cols of Ψ lin indpt $\implies$ $\Psi x \neq 0$ for any $x \neq 0$

$\implies$ $(\Psi x)'\Psi x = x'\Psi'\Psi x > 0$ for any $x \neq 0$

$\implies$ $\Psi'\Psi$ is positive definite matrix (by definition)

$\implies$ all eigenvalues are positive (Why?)

$\implies$ $\Psi'\Psi$ invertible

(ii) To show: columns of Ψ dependent $\implies$ $\Psi'\Psi$ not invertible.

Columns of Ψ linearly dependent

$\implies$ $\Psi x = 0$ for some $x \neq 0$

$\implies$ $(\Psi x)'\Psi x = x'\Psi'\Psi x = 0$ for some $x \neq 0$

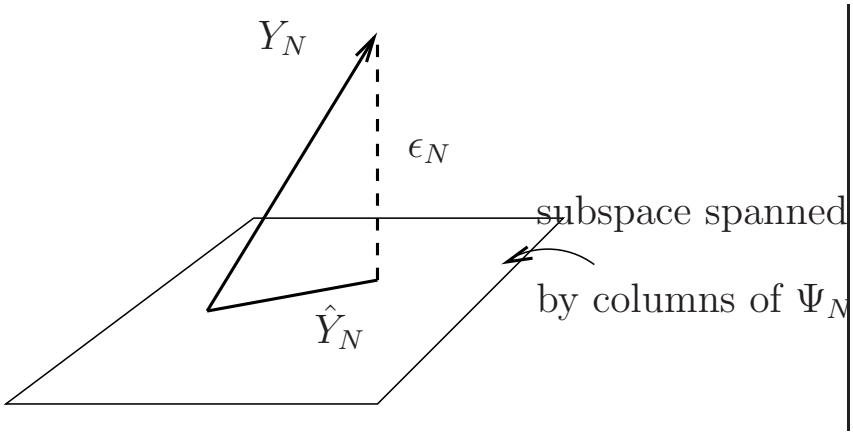
$\implies$ $\Psi'\Psi$ is positive semidefinite

$\implies$ $\Psi'\Psi$ has alteast one eigenvalue $= 0$.

$\implies$ $\Psi'\Psi$ not invertible.

Geometric Interpretation of LS

Define $\hat{Y}_N = \Psi_n \theta_{\text{LS}}$. So $Y_N = \hat{Y}_N + \epsilon_N$



From projection theorem, error of least squares estimate

$$\epsilon_N = Y_N - \hat{Y}_N \perp \text{subspace of columns of } \Psi_N$$

$$\implies (Y_N - \hat{Y}_N)' \Psi_N = 0$$

$$\implies (Y_N - \Psi_N \theta_{\text{LS}})' \Psi_N = 0$$

$$\theta_{\text{LS}} = (\Psi_N' \Psi_N)^{-1} \Psi_N' Y_N$$

The Projection Theorem is a powerful tool from an area of mathematics called "Functional Analysis". It is widely used in advanced optimization

Weighted Least Squares

Recall standard least squares minimizes

$$J(\theta) = \sum_{k=n}^{N} e[k]^2$$

i.e. all errors are equally important.

Sometimes data taken later in an experiment is much more error prone than data taken earlier (or vice versa). Then reasonable to weigh the errors differently.

Weighted Least Squares: Minimize

$$J(\theta) = \sum_{k=n}^{N} \alpha[k] e[k]^2$$

where $\alpha[k] > 0$ are weighting terms. In vector notation

$$J(\theta) = \epsilon' W \epsilon \quad \text{where } W = \text{diag}[\alpha[n], \ldots, \alpha[N]]$$

Setting $\partial J / \partial \theta = 0$ yields WLS normal equations

$$\Psi' W \Psi \theta_{\text{WLS}} = \Psi' W Y$$

If $\Psi' W \Psi$ invertible then

$$\boxed{\theta_{\text{WLS}} = (\Psi' W \Psi)^{-1} \Psi' W Y}$$

Remark: WLS becomes standard LS if $W = I$.

Additional Comments:

1. Choice of weighting coefficients $\alpha[k]$: In many applications, data in the past is less important and is exponentially forgotten:

$$\alpha[k] = \rho^{N-k}, \qquad 0 < \rho < 1$$

Weighs observations close to time N more than past ones.
$\rho \to 1$: longer memory
$\rho << 1$: shorter memory. Very useful in real time computations – e.g. Recursive Least Squares.

2. Note that $\theta_{\text{WLS}} = (\Psi' W \Psi)^{-1} \Psi' W Y$ can be re-written in terms of the regression vectors $\psi[k]$ as:

$$\boxed{\theta_{\text{WLS}} = \left[\sum_{k=n}^{N} \alpha[k]\psi[k]\psi'[k]\right]^{-1} \left[\sum_{k=n}^{N} \alpha[k]\psi[k]y[k]\right]}$$

Computational Least Squares

Given $Y_{N\times 1} = \Psi_{N\times n}\theta_{n\times 1} + \epsilon_{N\times 1}$, $J(\theta) = (Y-\Psi\theta)'(Y-\Psi\theta)$

$$\Psi'\Psi\hat{\theta} = \Psi'Y \text{ or equiv } \hat{\theta} = (\Psi'\Psi)^{-1}\Psi'Y$$

Apart from toy examples, never directly solve above system.

Nice Matrices:

1. Orthonormal: $U_{m\times n}$ s.t. $U'U = I$.

Note: If U is square, then $U^{-1} = U'$ and so $U'U = UU' = I$.

If U is rectangular matrix, then $UU' \neq I$ in general.

2. Diagonal: inverse element wise; elements are eigenvalues.

3. Upper/lower triangular. $|L|$ = prod of diagonal elements

Factorizations that yield nice matrices:

1. Spectral (Eigen) decomposition. $A_{m\times m} = TDT^{-1}$

If A is symmetric, then $T^{-1} = T'$.

2. Singular Value Decomposition (SVD).

$$A_{m\times n} = U_{m\times m}\, D_{m\times n}\, V'_{n\times n}$$

where $U'U = I_{m\times m}$, $V'V = I_{n\times n}$ (orthonormal matrices)

$$\text{If } m \geq n \text{ then } D = \begin{bmatrix} \bar{D}_{n\times n} \\ 0_{(m-n)\times n} \end{bmatrix}, \quad \bar{D} = \begin{bmatrix} \sigma_1 & & \\ & \ddots & \\ & & \sigma_n \end{bmatrix}$$

$\sigma_i \geq 0$ are called singular values of matrix A.

SVD can also be written as $A_{m\times n} = \sum_{i=1}^{\min(m,n)} \sigma_i u_i v_i'$

(i) SVD: more general than eigendecomp. Don't need square matrix.
(ii) If A has rank n, then $\sigma_1, \ldots, \sigma_n > 0$.
(iii) Compared to eigendecomp, need 2 orthonormal matrices U, V.
(iv) For symmetric positive definite matrix A, $\sigma_i = \lambda_i$.
(v) For symmetric matrix A, $\sigma_i = |\lambda_i|$ (show this).
(vi) $A^{-1} = (UDV')^{-1} = VD^{-1}U'$. So $\sigma_i(A^{-1}) = 1/\sigma_i(A)$.

SVD Conceptually. (Matrix Computations, Golub & Van Loan)

$$AA' = UDV'VD'U' = UDD'U' = U \begin{bmatrix} \bar{D}^2 & 0 \\ 0 & 0 \end{bmatrix} U'$$

$$A'A = VD'U'UDV' = VD'DV' = V\bar{D}^2V'$$

U: matrix of eigenvectors of AA'. V: matrix of eigenvectors of $A'A$

Singular values $\sigma_i = \sqrt{\lambda}_i(A'A)$.

Example. `A = rand(3,4), [U,D,V] = svd(A)`

$$A_{3\times 4} = \begin{bmatrix} 0.9040 & 0.2420 & 0.8128 & 0.5896 \\ 0.9409 & 0.9757 & 0.6974 & 0.8330 \\ 0.8025 & 0.3172 & 0.2695 & 0.3638 \end{bmatrix}$$

$$U = \begin{bmatrix} -0.5615 & -0.7340 & -0.3821 \\ -0.7255 & 0.6588 & -0.1991 \\ -0.3978 & -0.1654 & 0.9024 \end{bmatrix}, \; D = \begin{bmatrix} 2.3517 & 0 & 0 & 0 \\ 0 & 0.4873 & 0 & 0 \\ 0 & 0 & 0.2884 & 0 \end{bmatrix}$$

$$V = \begin{bmatrix} -0.6419 & -0.3619 & 0.6639 & -0.1273 \\ -0.4125 & 0.8467 & -0.0017 & -0.3362 \\ -0.4548 & -0.3728 & -0.7153 & -0.3774 \\ -0.4593 & 0.1147 & -0.2180 & 0.8534 \end{bmatrix}.$$

Least Squares using SVD. $\hat{\theta} = (\Psi'\Psi)^{-1}\Psi' Y$

$$\text{SVD: } \Psi_{N\times n} = U_{N\times N} D_{N\times n} V'_{n\times n}, \quad D = \begin{bmatrix} \bar{D}_{n\times n} \\ 0_{(N-n)\times n} \end{bmatrix} \text{ since } N > n$$

$$\begin{aligned} \text{So } \hat{\theta} &= (VD'U'UDV')^{-1} VDU'Y \\ &= (V\bar{D}^2V')^{-1} VD'U'Y = V\bar{D}^{-2}V'\, VD'U'Y \end{aligned}$$

$$\boxed{\hat{\theta}_{n\times 1} = V \begin{bmatrix} \bar{D}^{-1}_{n\times n} & 0_{n\times(N-n)} \end{bmatrix} U'_{N\times N}\, Y_{N\times 1} = \sum_{i=1}^{n} v_i \frac{u_i' Y}{\sigma_i}}$$

assuming Ψ has rank n. If Ψ has rank $r < n$, then

$$\hat{\theta}_{n\times 1} = \sum_{i=1}^{r} v_i \frac{u_i' Y}{\sigma_i}$$

Remark: $\sum_{i=1}^{r} v_i \frac{u_i'}{\sigma_i}$ is the Moore-Penrose inverse of Ψ.

Ill-conditioning/sensitivity of least squares: $Y^{\Delta} = Y + \Delta$. How is least squares estimate affected? Using SVD:

$$\hat{\theta} = \sum_{i=1}^{r} \frac{u_i'(Y+\Delta)}{\sigma_i} v_i$$

For small σ_i, small changes in observation have large effect on least squares estimate.

Linear Algebraic Equations

Moore Penrose Inverse: $A \in \mathbb{R}^{m\times n}$. Then $A^+ \in \mathbb{R}^{n\times m}$ exists, is unique and satisfies

(i) $AA^+A = A$ (ii) $A^+AA^+ = A^+$

(iii) $(AA^+)' = AA^+$ (iv) $(A^+A)' = A^+A$

Limit definition: (Tikhonov Regularization)

$$A^+ = \lim_{\epsilon\to 0}(A'A + \epsilon^2 I)^{-1}A' = \lim_{\epsilon\to 0} A'(AA' + \epsilon^2 I)^{-1}$$

Example: $A = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} \implies A^+ = \begin{bmatrix} 1/4 & 1/4 \\ 1/4 & 1/4 \end{bmatrix}$.

Construction of Moore Penrose inverse: If $A \in \mathbb{R}^{m\times n}$ has rank $r \le \min\{m, n\}$, then using SVD

$$\boxed{A^+ = \sum_{i=1}^{r} v_i \frac{u_i'}{\sigma_i}}$$

Examples: (i) For scalar A, $A^+ = 1/A$ if $A \neq 0$ and 0 otherwise.

(ii) For $A \in \mathbb{R}^n$, $A^+ = A'/(A'A)$ if $A \neq 0$ and 0 otherwise.

Given linear algebraic equation

$$\boxed{A_{m\times n}x_{n\times 1} = b_{m\times 1}}$$

Theorem (Existence & Uniqueness)

1. Existence: A solution exists iff $b \in \mathcal{R}(A)$ (space of linear combinations of cols of A). Equivalently iff $AA^+b = b$.

2. All solutions are of the form

$$\boxed{x = A^+b + (I - A^+A)y, \quad y \in \mathbb{R}^n \text{ is arbitrary}}$$

3. Uniqueness: Iff $A^+A = I$. Equivalently iff $\mathcal{N}(A) = 0$. i.e., $\{x : Ax = 0\}$.

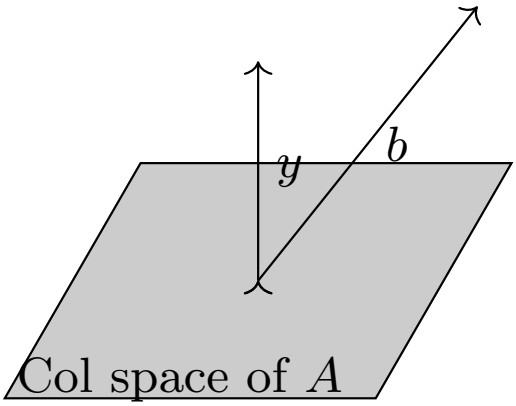
Fundamental Theorem of Linear Algebra

Theorem [Gauss]

$$F = \{x \in \mathbb{R}^n : A_{m\times n}x = b_{m\times 1}\}$$

Exactly one of the following holds (but not both)

- $F \neq \emptyset$ (a solution x exists)
- $\exists y \in \mathbb{R}^m : A'y = 0, b'y \neq 0.$



Farkas Lemma for feasibility of linear inequalities:

$$F = \{x \in \mathbb{R}^n : A_{m\times n}x = b_{m\times 1}, x \geq 0\}$$

Exactly one of the following holds (but not both)

- $F \neq \emptyset$ (a feasible solution x exists)
- $\exists y \in \mathbb{R}^m : A'y \geq 0, b'y < 0.$

Either there is a feasible x, or there is a y that certifies no such x exists.

Motivation. Classical LP: Compute $\min_x c'x$ such that $x \in F$.

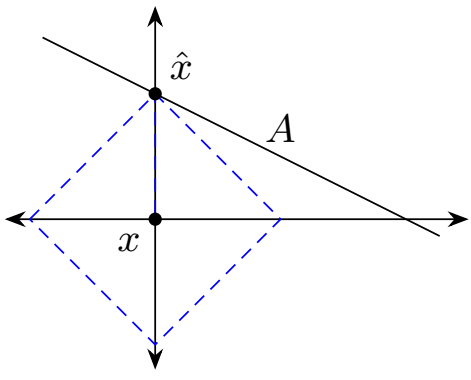
Numerical conditioning of linear equation

Assume A is invertible, solve $A_{n\times n}x_{n\times 1} = b$

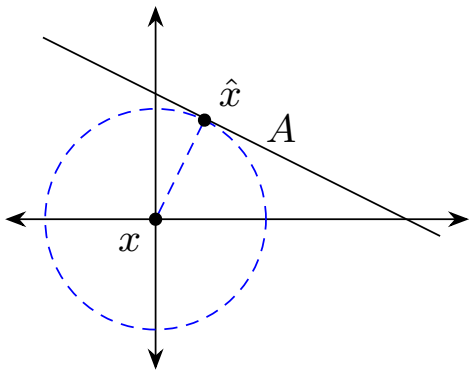
Vector norm: Suppose $x \in \mathbb{R}^n$. Then. $\|\cdot\| : x \to \mathbb{R}^+$ is a vector norm if

1. $\|x\| \geq 0$
2. $\|x\| = 0$ iff $x = 0$.
3. $\|cx\| = |c|\|x\|$ for all scalars c
4. $\|x + y\| \leq \|x\| + \|y\|$ (triangle inequality)

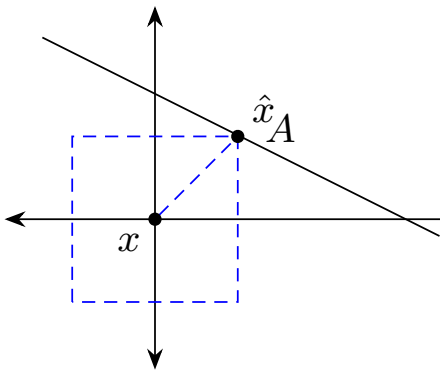
Examples: l_p norm: $\|x\|_p = \left(\sum_{i=1}^n |x_i|^p\right)^{1/p}$, $p > 0$.
l_2 norm: $\|x\|_2 = \left(\sum_{i=1}^n |x_i|^2\right)^{1/2}$
l_∞ norm: $\|x\|_\infty = \max_i |x_i|$.
l_1 norm: $\|x\|_1 = \sum_{i=1}^n |x_i|$



$p = 1$



$p = 2$



$p = \infty$

Unit l_p balls.

Matrix norm: Suppose $A \in \mathbb{R}^{n\times n}$. Then $\|\cdot\| : A \to \mathbb{R}^+$ is a matrix norm if

1. $\|A\| \geq 0$
2. $\|A\| = 0$ iff $A = 0$.
3. $\|cA\| = |c|\|A\|$ for all scalars c
4. $\|A + B\| \leq \|A\| + \|B\|$ (triangle inequality)
5. $\|AB\| \leq \|A\|\,\|B\|$ (sub-multiplicative). Also holds if $B \in \mathbb{R}^n$.

Example: Frobenius norm $\|A\|_F = (\sum_{i=1}^n \sum_{j=1}^n a_{ij}^2)^{1/2}$

Induced matrix norm:

$$\|A\|_p = \max_{x\neq 0} \frac{\|Ax\|_p}{\|x\|_p} = \max_{\|x\|_p=1} \|Ax\|_p$$

Example: $\|A\|_2 = \max_{\|x\|_2=1} \|Ax\|_2 = \max_{x'x=1} \sqrt{x'A'Ax}$

Rayleigh Ritz Theorem. l_2 induced norm is max singular value. (Max singular value is called spectral norm)

$$\|A\|_2 = \sqrt{\lambda_{\max}(A'A)} = \sigma_{\max}(A)$$

Example 1. Suppose $A = U$ (orthonormal). Then $\|A\|_2 = 1$.

Example 2. Suppose A is symmetric. Then $\|A\|_2 = |\lambda_{\max}(A)|$.

Theorem: For a general matrix A, $\|A\|_2 = \sigma_{\max}(A) \geq \lambda_{\max}(A)$, i.e., spectral norm is larger than spectral radius.

Proof: $\|A\|_2 = \max_{x'x=1}(x'A'Ax)^{1/2}$. Choose x as normalized e-vector of $\lambda_{\max}(A)$.

Theorem. Suppose $Ax = b$ and $A(x + \Delta x) = b + \Delta b$. Then

$$\frac{\|\Delta x\|_2}{\|x\|_2} \leq c \frac{\|\Delta b\|_2}{\|b\|_2} \qquad \text{where } c = \frac{\sigma_{\max}(A)}{\sigma_{\min}(A)}$$

c is called the condition number of matrix A.

Proof: $A\Delta x = \Delta b \implies \Delta x = A^{-1}\Delta b \implies \|\Delta x\| \leq \|A^{-1}\|\|\Delta b\|$.

$b = Ax \implies \|b\| \leq \|A\|\|x\|$.

So $\|b\|\|\Delta x\| \leq \|A^{-1}\|\|A\|\|\Delta b\|\|x\|$

Holder's inequality: If $1/p + 1/q = 1$ then

$$\boxed{x'y \leq \|x\|_p \, \|y\|_q}$$

For $p = 2, q = 2$, called Cauchy Schwartz inequality.

Example 1. $|\mathbb{E}\{XY\}|^2 \leq \mathbb{E}\{X^2\}\mathbb{E}\{Y^2\}$.

Proof: Choose $x_i = \sqrt{p}_i X_i$ and $y_i = \sqrt{p}_i Y_i$.

Example 2. $(\sum_i x_i)^2 \leq \sum_i x_i^2$. Proof: Choose $y_i = 1$.

Example 3. Recall $\|x\|_p = (\sum_i x^p)^{1/p}$ and $\|x\|_\infty = \max_i |x_i|$.

Using Cauchy Schwartz, can show that $\lim_{p\to\infty} \|x\|_p = \|x\|_\infty$.

Proof (FYI): $|x_i| \leq \|x\|_p$ by Cauchy Schwartz. This implies

$$\max |x|_i = \|x\|_\infty \leq \|x\|_p, \qquad p \geq 1$$

Also

$$\|x\|_p = \Big(\sum_i |x_i|^{p-q} |x_i|^q\Big)^{1/p} \leq \Big(\sum_i \max_i |x_i|^{p-q} |x_i|^q\Big)^{1/p}$$

$$= \|x\|_\infty^{(p-q)/p} \|x\|_q^{q/p}$$

So $\lim_{p\to\infty} \|x\|_p \leq \|x\|_\infty$.

Principal Component Analysis (PCA)

For $A \in \mathbb{R}^{m\times n}$, Frobenius norm is $\|A\|_F = \sum_{i=1}^m \sum_{j=1}^n a_{ij}^2$.

Theorem[2.5.2 in Golub & Van Loan]. If $A \in \mathbb{R}^{m\times n}$ then

1. $l = \text{rank}(A) \le \min\{m,n\}$ and so A has SVD

$$A_{m\times n} = \sum_{i=1}^{l} \sigma_i u_i v_i', \qquad u_i \in \mathbb{R}^m, v_i \in \mathbb{R}^n$$

2. Find $Z \in \mathbb{R}^{m\times n}$ of rank $r < l$ that minimizes $\|A - Z\|_F$. (Least squares fit of matrix $Z_{m\times n}$ to matrix $A_{m\times n}$.)

$$\text{Soln:} \quad \underset{Z:\text{rank}(Z)=r}{\text{argmin}} \|A - Z\|_F = \sum_{i=1}^{r} \sigma_i u_i v_i' = U_r D_r V_r'$$

and the approximation error is

$$\min_{Z:\text{rank}(Z)=r} \|A - Z\|_F = \Big[\sum_{i=r+1}^{l} \sigma_i^2\Big]^{\frac{1}{2}}, \quad \min_{Z:\text{rank}(Z)=r} \|A - Z\|_2 = \sigma_{r+1}$$

$D_r = \text{diag}(\sigma_1, \ldots, \sigma_r)$ comprising r largest singular values,
$U_r \in \mathbb{R}^{m\times r}$ and $V_r \in \mathbb{R}^{n\times r}$ are first r cols of U and V.

Remarks: Computation of Z is called PCA.

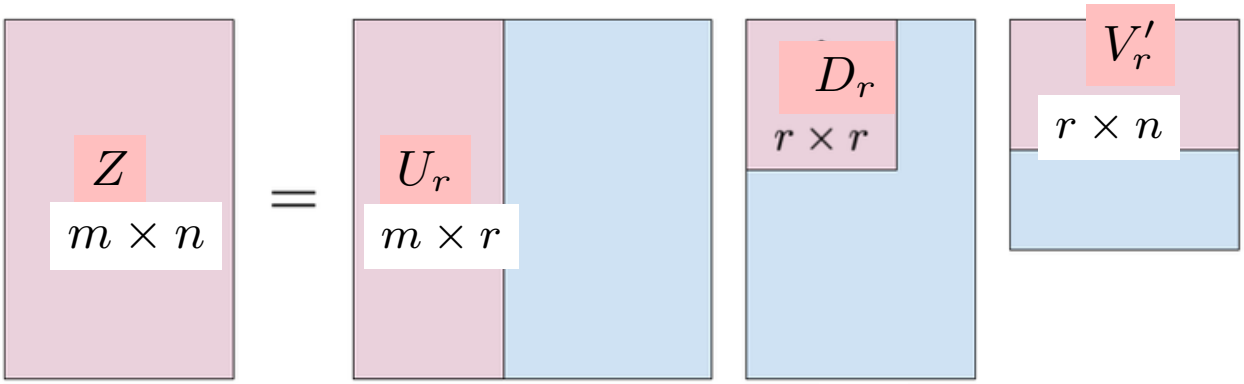
1. Typically n is number of time/data samples and $n > m$.
2. The r vectors $u_i \in \mathbb{R}^m$, $i = 1, \ldots, r$: principal axes.

The r vectors $\sigma_i v_i \in \mathbb{R}^n$, $i = 1, \ldots, r$: principal components.

$[\sigma_1 v_1(j), \ldots, \sigma_r v_r(j)]' \in \mathbb{R}^r$: feature vector of j-th data sample, $j = 1, \ldots, n$.

3. Z has $(m + n + 1)r$ parameters; A has $m \times n$ parameters.

PCA (Interpretation 2)



$$A = \underset{m\times m}{U} \quad \underset{m\times n}{D} \quad \underset{n\times n}{V'}$$

Given $A \in \mathbb{R}^{m\times n}$, find matrix $Z \in \mathbb{R}^{m\times n}$ of rank $r < \min(m,n)$ that minimizes square error:

$$\boxed{\text{Aim: minimize } \|A-Z\|_F^2 \quad \text{subject to } \operatorname{rank}(Z) \le r}$$

$\operatorname{rank}(Z) \le r$ is equiv to $Z = XY$ where $X \in \mathbb{R}^{m\times r}$, $Y \in \mathbb{R}^{r\times n}$.

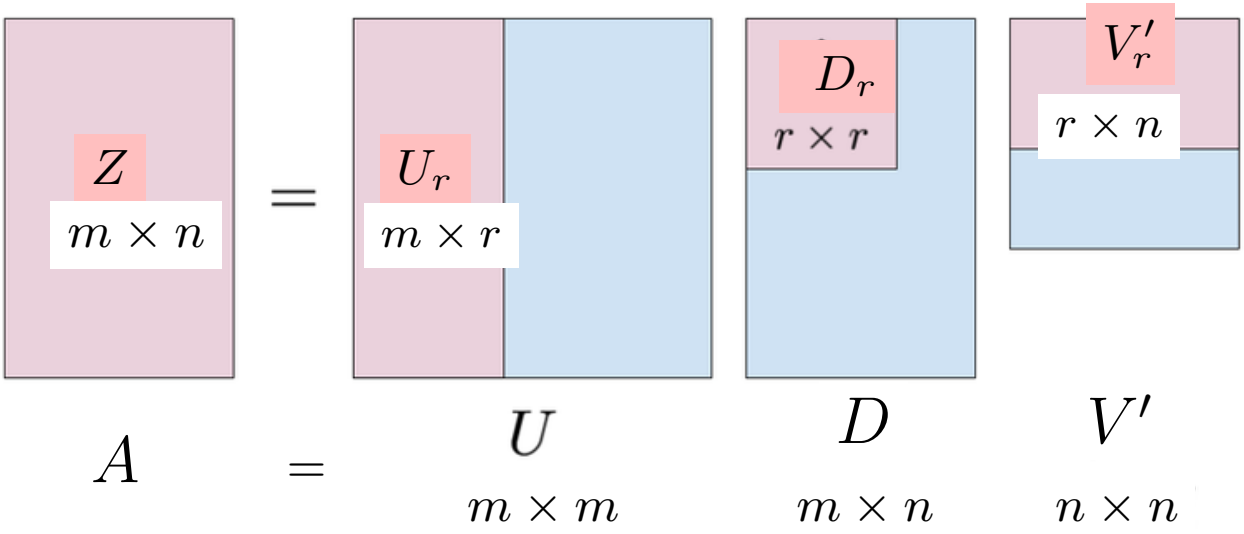
$$\text{Aim: minimize } \|A-XY\|_F^2 \quad \text{where } X \in \mathbb{R}^{m\times r}, Y \in \mathbb{R}^{r\times n}$$

Let $X = \begin{bmatrix} x_1 \\ \vdots \\ x_m \end{bmatrix}$, $Y = [y_1, \ldots, y_n]$. $\boxed{\text{Aim: } \min \sum_{i=1}^{m}\sum_{j=1}^{n}(A_{ij} - x_i y_j)^2}$.

PCA is a least squares fit of a matrix Z to a matrix A (of same dimension) when Z is constrained to have rank r.

Solution: (i) Compute SVD: $A = U_{m\times m} D_{m\times n} V'_{n\times n}$ where $U'U = I$, $V'V = I$ and $D = [\text{diag}(\sigma_1, \ldots, \sigma_m) | 0_{m\times(n-m)}]$ if $m < n$.
(ii) Set $D_r = \text{diag}(\sigma_1, \ldots, \sigma_r)$ with r largest singular values.
Set $U_r \in \mathbb{R}^{m\times r}$ and $V_r \in \mathbb{R}^{n\times r}$ as first r cols of U and V.
(iii) Choose $X = U_r D_r$, $Y = V'_r$. Set $Z = XY = \sum_{i=1}^{r} \sigma_i u_i v'_i$.

The m-dim vectors $u_1, u_2, \ldots, u_r$ are principal axes/directions.
n-dim vectors $\sigma_1 v_i, \ldots, \sigma_r v_r$: principal components/scores/features



Example. `A = rand(3,4), [U,D,V] = svd(A)`

$$A_{3\times 4} = \begin{bmatrix} 0.9040 & 0.2420 & 0.8128 & 0.5896 \\ 0.9409 & 0.9757 & 0.6974 & 0.8330 \\ 0.8025 & 0.3172 & 0.2695 & 0.3638 \end{bmatrix}$$

$$U = \begin{bmatrix} -0.5615 & -0.7340 & -0.3821 \\ -0.7255 & 0.6588 & -0.1991 \\ -0.3978 & -0.1654 & 0.9024 \end{bmatrix}, \; D = \begin{bmatrix} 2.3517 & 0 & 0 & 0 \\ 0 & 0.4873 & 0 & 0 \\ 0 & 0 & 0.2884 & 0 \end{bmatrix}$$

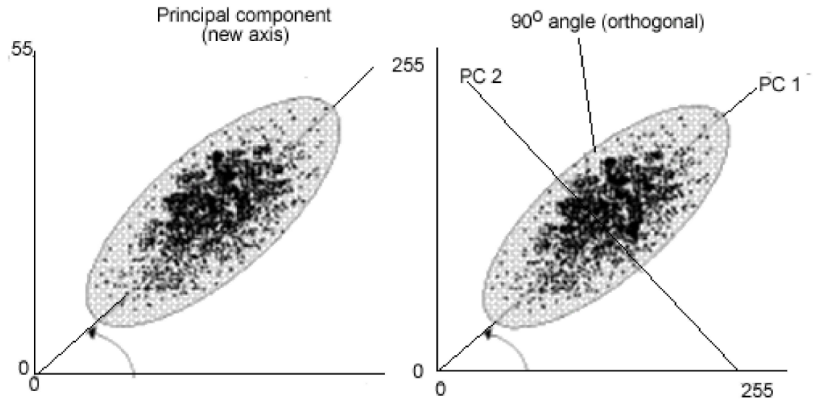
$$V = \begin{bmatrix} -0.6419 & -0.3619 & 0.6639 & -0.1273 \\ -0.4125 & 0.8467 & -0.0017 & -0.3362 \\ -0.4548 & -0.3728 & -0.7153 & -0.3774 \\ -0.4593 & 0.1147 & -0.2180 & 0.8534 \end{bmatrix}.$$

Rank 1 approx to A: $U_1 D_1 V_1 = \begin{bmatrix} 0.8477 & 0.5447 & 0.6006 & 0.6066 \\ 1.0953 & 0.7038 & 0.7760 & 0.7837 \\ 0.6006 & 0.3859 & 0.4255 & 0.4298 \end{bmatrix}$.

Rank 2 approx to A: $U_2 D_2 V_2 = \begin{bmatrix} 0.9771 & 0.2419 & 0.7340 & 0.5656 \\ 0.9791 & 0.9756 & 0.6564 & 0.8205 \\ 0.6298 & 0.3177 & 0.4556 & 0.4205 \end{bmatrix}$

Of course rank 3 approx of A is A!

PCA is called Karhunen Loeve transformation in image proc.



Applications of PCA

1. Efficient approx matrix multiplication with stream of vectors.
$A_{m\times n}x_{n\times 1}$: $O(m\times n)$ computations.
PCA based $Z_{m\times n}x_{n\times 1}$ only needs $O((m+n)r)$ computations.
Why? $Zx=\sum_{i=1}^{r}\sigma_i u_i v_i' x$ where $u_i\in\mathbb{R}^{m\times 1}$, $v_i\in\mathbb{R}^{n\times 1}$.
$v_i'x$: n multiplications; $u_i\times(v_i'x)$ m multiplications.
Total: $(m+n)$ multiplications for each i.

2. Min error Feature Compression and Storage: Given big data $A=[a_1,a_2,\ldots,a_n]$ where $a_i\in\mathbb{R}^m$, $m<n$ and m is large.
SVD: $A=UDV'=\sum_{j=1}^{m}\sigma_j u_j v_j'$. storage $=m\times n$
PCA: $Z=U_rD_rV_r'=\sum_{j=1}^{r}\sigma_j u_j v_j'$. storage $=(n+m)r$.

Example Image Compression.
Approximate $A_{m\times m}$ by $\text{Acomp}_{m\times m}$ of rank r.

```
>> A=rgb2gray(imread('basket.jpg'));
>> imshow(A);
>> [U,S,V]=svd(double(A));
>> r=25; % Rank-r approximation
>> Acomp=U(:,1:r)*S(1:r,1:r)*(V(:,1:r))';
>> imshow(uint8(Acomp));
```

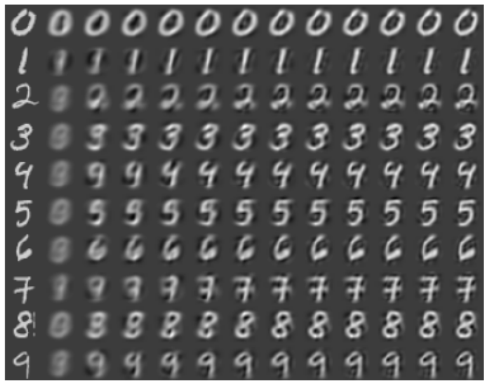
$m=400$. Choosing $r=25$ gives the RHS figure.

3. PCA on MNIST data:

1. Each image $28 \times 28 = 784$ pixels `https://cseweb.ucsd.edu/classes/wi17/cse258-a/reports/a059.pdf`
2. Each image stacked into col vector of dim $m = 784$.
3. Each element: pixel intensity $\in \{0, \ldots, 255\}$
4. Given $n = 60,000$ examples (training set) define $A \in \mathbb{R}^{m \times n}$.

How to reduce dimension of training set? $m = 784$, $n = 60,000$.

Use PCA to compute $Z_{m \times n}$ = rank r approx of $A_{m \times n}$.



Results.

1st column: original image $A_{:,1}$.

2nd col: $Z_{,:1}$ rank $r = 1$.

3rd col: $Z_{,:1}$ rank $r = 6$,

4th col: $Z_{,:1}$ rank $r = 11$,

Last (13th) col: $Z_{:,1}$ rank $r = 56$.

Conclusions:

1. No significant improvement after rank $r = 36$ (9-th column in picture).
2. Dimension reduction: 784 to 36.
3. Can then train a neural net on lower dimensional data.

   Recall SVD: $A = UDV' = \sum_{j=1}^{m} \sigma_j u_j v_j'$. storage $= m \times n$

   PCA: $Z = U_r D_r V_r' = \sum_{j=1}^{r} \sigma_j u_j v_j'$. storage $= (n + m)r$.

4. Eigenface Face Recognition in Computer Vision: Given n face images, and a new image, whose face is it?

Brute force:

1. Denote each $N \times N$ image i as $m = N^2$ dim vector A_i, $i = 1, 2 \ldots, n$.
2. Subtract mean: $a_i = A_i - \frac{1}{n}\sum_{j=1}^{n} A_j$. Construct matrix $A_{m\times n} = [a_1, \ldots, a_n]$
3. Given new image a^{new}, closest face $= \text{argmin}_i \|a_i - a^{\text{new}}\|$

PCA Approximation: Start with $A = UDV'$.

1. Image $i = i$th col of $A \implies a_i = UDV'e_i$.
   PCA: $Z = U_r D_r V_r' \implies z_i = \underbrace{U_r}_{m\times r} \quad \underbrace{D_r V_r' e_i}_{\substack{\text{feature of image } i \\ r \times 1}}$
2. Given a^{new}, closest approx face $= \text{argmin}_i \|z_i - a^{\text{new}}\|$. Simplify the computation as:

$$\begin{aligned}
\operatorname*{argmin}_i \|z_i - a^{\text{new}}\| &= \operatorname*{argmin}_i \|U_r D_r V_r' e_i - a^{\text{new}}\| \\
&= \operatorname*{argmin}_i \|U_r' U_r D_r V_r' e_i - U_r' a^{\text{new}}\| \\
&= \operatorname*{argmin}_i \| \underbrace{D_r V_r' e_i}_{\text{feature of } i} - U_r' a^{\text{new}}\|
\end{aligned}$$

since U_r is orthonormal matrix (so preserves norm).

See "Eigenfaces for Recognition" by Turk & Pentland, 1991.

Much more serious applications of SVD in dynamical systems involve model order reduction of balanced linear realizations.

5. Reduced Dimension Least Squares: $Y_{p\times 1} = \Psi_{p\times q}\theta_q$, $p > q$.

$$\theta_* = (\Psi'\Psi)^{-1}\Psi' Y$$

Choose $r < q$. Then PCA on Ψ' yields $\bar{\Psi}'$ of rank r:

$$\bar{\Psi}' = U_r D_r V_r' \implies \bar{\Psi}'\bar{\Psi} = U_r D_r^2 U_r' \in \mathbb{R}^{r\times r}$$

So reduced dimensional least squares estimate $\bar{\theta}_* \in \mathbb{R}^r$ is

$$\bar{\theta}_* = (\bar{\Psi}'\bar{\Psi})^{-1}\bar{\Psi}' Y = (U_r D_r^2 U_r')^{-1} U_r D_r V_r' Y$$

6. Max variance interpretation of PCA: Given $A \in \mathbb{R}^{m\times n}$, find orthonormal matrix $Q \in \mathbb{R}^{m\times r}$

$$Q_* = \operatorname*{argmin}_Q \sum_{i=1}^n \| a_i - \underbrace{QQ'a_i}_{Z_i} \|_2^2, \qquad \text{subject to } Q'Q = I$$

where $A = [a_1, \ldots, a_n]$.
Solution: Choose $Q_* = U_r$ from PCA (truncated SVD).
Maximum variance interpretation:

$$Q_* = \operatorname*{argmin}_Q \sum_{i=1}^n \| a_i - \underbrace{QQ'a_i}_{Z_i} \|_2^2 = \operatorname*{argmax}_Q \operatorname{tr}(Q'AA'Q)$$

$$= \operatorname*{argmax}_Q \operatorname{var}(Q'A)$$

So PCA reduces dimension while retaining max possible variance.

Work thru the example on `https://www.projectrhea.org/rhea/index.php/PCA_Theory_Examples`

Recursive Least Squares

There are 2 types of parameter estimation algorithms:
(i) **Off-line: (Batch-processing algorithms)**
Example: Least Squares $\theta_{\text{WLS}} = (\Psi' W \Psi)^{-1} \Psi' W Y$

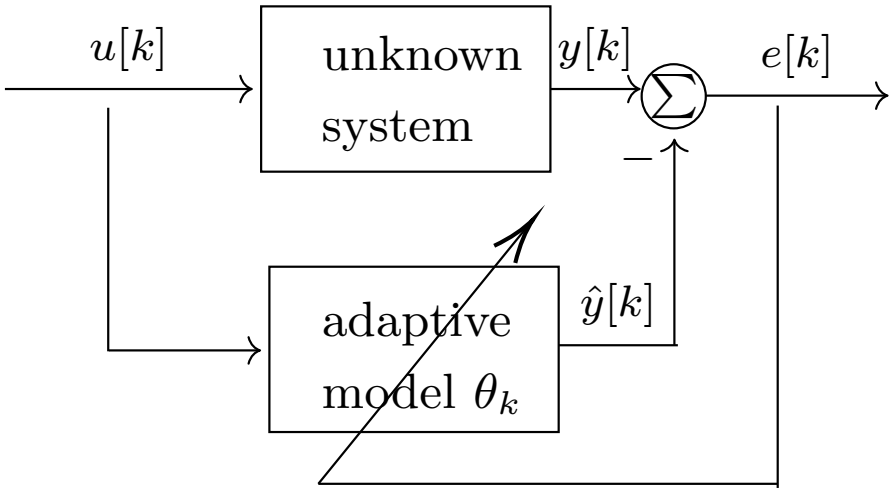
- Operates on a fixed length of data N
- Assumes constant true model parameter θ^0
- Algorithm processes all data and then yields θ_{WLS}.

2. On-line: (Recursive algorithms)
Recursive Least Squares (RLS), Least Mean Squares (LMS)

- Updated model estimate θ_k with each new data point.

$$\theta_{k+1} = \theta_k + f(y[k+1], u[k+1], \theta_k)$$

- Real time update of parameters
- Can track slowly varying parameters



Recursive Least Squares (RLS)

$$\text{True model: } y_k = \sum_{i=1}^{n} a_i^0 y_{k-i} + b_i^0 u_{k-i}$$

$$\theta^0 = [a_1^0, \ldots, a_n^0, b_1^0, \ldots, b_n^0]' \in \mathbb{R}^{2n}$$

Weighted least squares estimate with $w_{k,N} = c\rho^{N-k}$, $0 < \rho \leq 1$:

$$\theta_N = \operatorname*{argmin}_{\theta} \sum_{k=n}^{N} w_{k,N} \left(y_k - \psi_k'\theta\right)^2 = (\Psi_N' W_N \Psi_N)^{-1} \Psi_N' W_N Y_N,$$

$$\psi_k = [y_{k-1}, \ldots, y_{k-n}, u_{k-1}, \ldots, u_{k-n}]', \Psi_N = [\psi_n, \ldots, \psi_N]',$$

$$Y_N = [y_n, \ldots, y_N]', \quad W_N = \operatorname{diag}(w_{n,N}, \ldots, w_{N,N}).$$

Given new data u_{N+1}, y_{N+1} we know

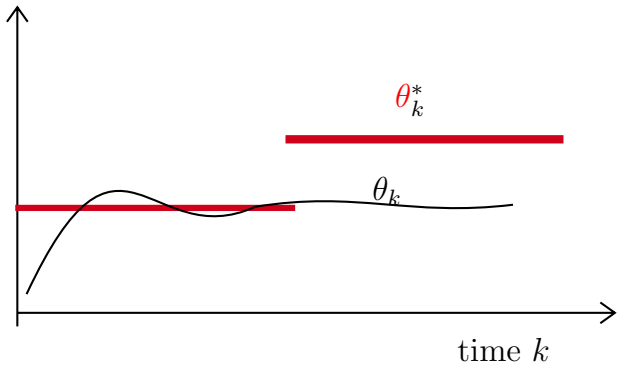
$$\theta_{N+1} = (\Psi_{N+1}' W_{N+1} \Psi_{N+1})^{-1} \Psi_{N+1}' W_{N+1} Y_{N+1}$$

How to compute θ_{N+1} in terms of θ_N?

Result: The RLS equations are: Pick θ_1 arbitrarily. Pick P_1 as some positive definite matrix. Then for $N = 1, 2, \ldots$

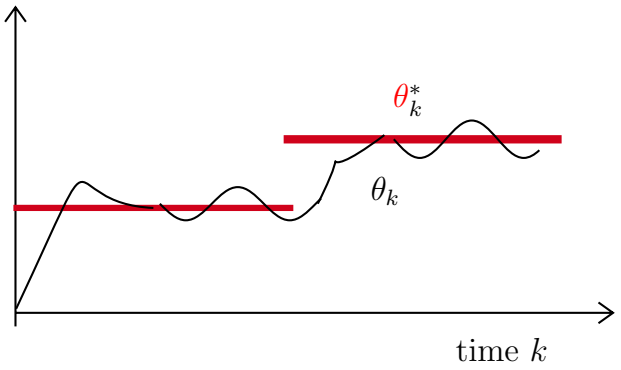
$$\theta_{N+1} = \theta_N + \frac{P_N \psi_{N+1}}{\frac{\rho}{c} + \psi_{N+1}' P_N \psi_{N+1}} \left(y_{N+1} - \psi_{N+1}' \theta_N\right)$$

$$P_{N+1} = \frac{1}{\rho}\left[P_N - \frac{P_N \psi_{N+1} \psi_{N+1}' P_N}{\frac{\rho}{c} + \psi_{N+1}' P_N \psi_{N+1}}\right]$$



No forgetting factor $c = \rho = 1$.

Cannot track time varying θ_k^*



Expo forgetting: $\rho < 1,\ c = 1-\rho$.

Tracks time varying θ_k^*.

Remarks:

1. The equation for update of θ_N is of form

$$\theta_{N+1} = \theta_N + L_{N+1}\psi_{N+1}(y_{N+1} - \psi'_{N+1}\theta_N)$$

$$\text{new estimate} = \text{old estimate} + \text{gain} \times \text{error}$$

where $L_{N+1} = \frac{P_N}{\rho/c + \psi'_{N+1}P_N\psi_{N+1}}$ is matrix-valued step size

2. Unlike least squares, in RLS no matrix inversion is required! So only matrix vector multiplications $O(n^2)$.

3. RLS is applicable when parameter θ is slowly time varying.

4. Least Mean Square (LMS) algorithm. Replace matrix step size L_N with constant scalar step size $\epsilon = 1 - \rho$:

$$\boxed{\theta_{N+1} = \theta_N + \epsilon\,\psi_{N+1}(y_{N+1} - \psi'_{N+1}\theta_N)}$$

5. RLS and LMS are *adaptive filtering algorithms*. Used to estimate slowly time evolving model θ_k given y_k, ψ_k, where

$$y_k = \psi'_k\theta_k + e_k.$$

If model (state) $\theta_{k+1} = A\theta_k + v_k$. Then Kalman filter is the optimal estimator.

Examples of RLS.

Example 1. Given $y_k = \theta + e_k$, $k = 1, \ldots, N$, compute RLS. Least squares estimator is $\theta_N = \frac{1}{N}\sum_{k=1}^{N} y_k$.

$$\begin{aligned}\theta_{N+1} &= \frac{1}{N+1}\sum_{k=1}^{N+1} y_k = \frac{N}{N+1}\frac{1}{N}\sum_{k=1}^{N} y_k + \frac{1}{N+1}y_{N+1} \\ &= \frac{N}{N+1}\theta_N + \frac{1}{N+1}y_{N+1} \\ &= \theta_N + \frac{1}{N+1}(y_{N+1} - \theta_N)\end{aligned}$$

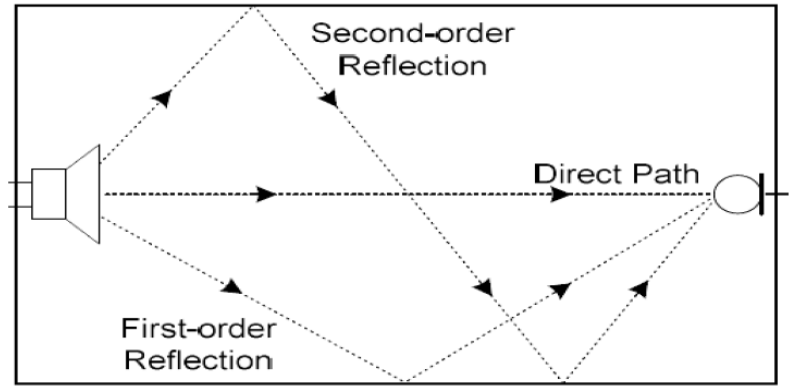
Let us check that RLS gives the same estimator. Choose $P_1 = 1$, $\rho = c = 1$, $\psi = 1$ in RLS eqns gives

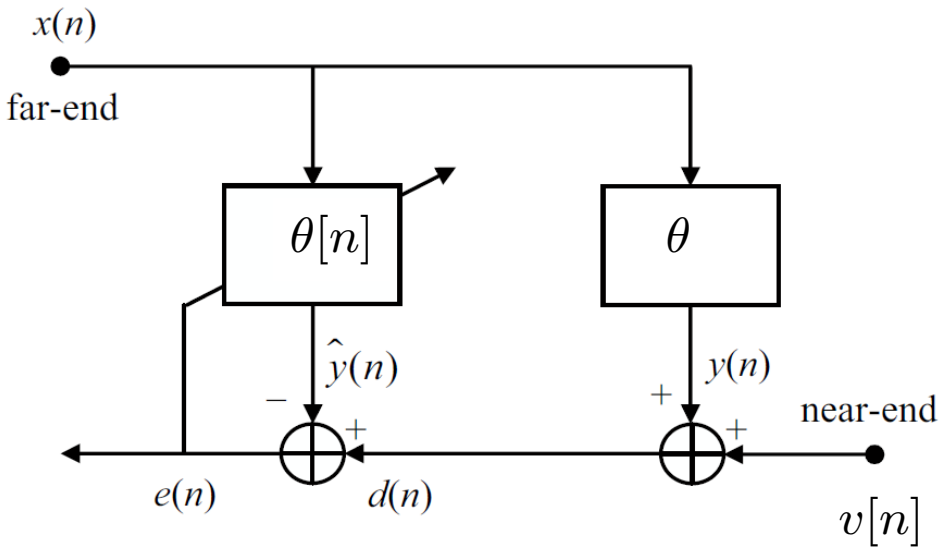
$$P_{N+1} = P_N - \frac{P_N^2}{1+P_N} = \frac{P_N}{1+P_N} \implies P_{N+1} = \frac{1}{N+1}$$

$$\theta_{N+1} = \theta_N + \frac{P_N}{1+P_N}(Y_{N+1} - \theta_N) = \theta_N + \frac{1}{N+1}(y_{N+1} - \theta_N)$$

Example 2. Homework: Read and implement the example in `https://www.mathworks.com/help/ident/examples/online-recursive-least-squares-estimation.html` Using RLS estimate the parameters of the nonlinear model for an internal combustion engine and also detect changes in the engine inertia.

Example 3. Echo Cancellation.





Input to microphone: $y[n] = \psi'[n]\,\theta + v[n]$
where $v[n]$ is iid noise (in the absence of near end speech),

$$\psi[n] = [x[n-1], \ldots, x[n-L]]'$$

is vector of L most recent speech samples
$\theta \in \mathbb{R}^L$: impulse response from loudspeaker to microphone.
θ is not known. Estimate θ to cancel echo signal $\psi'[n]\theta$.

Example 4. Noise Reducing Headphones `http://www-personal.umich.edu/~gowtham/bellala_EECS452report.pdf`

Derivation of RLS Equations

Step 1: Compute $(\Psi'_{N+1} W_{N+1} \Psi_{N+1})^{-1}$ recursively.

$$\begin{aligned}\Psi'_{N+1} W_{N+1} \Psi_{N+1} &= \sum_{k=n}^{N+1} \psi_k w_{k,N+1} \psi'_k \\ &= \sum_{k=n}^{N} \psi_k w_{k,N+1} \psi'_k + \psi_{N+1} w_{N+1,N+1} \psi'_{N+1}\end{aligned}$$

Recall exponential forgetting factor $w_{k,N+1} = c\rho^{N+1-k}$. So

$$\Psi'_{N+1} W_{N+1} \Psi_{N+1} = \rho\, \Psi'_N W_N \Psi_N + \psi_{N+1}\, c\, \psi'_{N+1}$$

For convenience define $P_{N+1} = (\Psi'_{N+1} W_{N+1} \Psi_{N+1})^{-1}$. Then

$$\boxed{P_{N+1} = \left[\rho\, P_N^{-1} + \psi_{N+1}\, c\, \psi'_{N+1}\right]^{-1}}$$

We now use matrix inversion lemma: Let

$$A = \rho P_N^{-1},\ B = \psi_{N+1},\ C = c,\ D = \psi'_{N+1}$$

Matrix inversion lemma states that

$$(A + BCD)^{-1} = A^{-1} - A^{-1}B(C^{-1} + DA^{-1}B)^{-1}DA^{-1}$$

$$\begin{aligned}P_{N+1} &= \left[\rho\, P_N^{-1} + \psi_{N+1}\, c\, \psi'_{N+1}\right]^{-1} \\ &= \frac{P_N}{\rho} - \frac{\frac{P_N}{\rho}\psi_{N+1}\psi'_{N+1}\frac{P_N}{\rho}}{\frac{1}{c} + \psi'_{N+1}\frac{P_N}{\rho}\psi_{N+1}}\end{aligned}$$

Step 2: Compute $\Psi'_{N+1}W_{N+1}Y_{N+1}$ recursively

$$\Psi'_{N+1}W_{N+1}Y_{N+1} = \rho\,\Psi'_N W_N Y_N + \psi_{N+1} c y_{N+1}$$

Step 3: Combining Step 1 and 2

$$\theta_{N+1} = P_{N+1}\Psi'_{N+1}W_{N+1}Y_{N+1}$$

Expanding out the above expression and noting that

$$\theta_N = P_N \Psi'_N W_N Y_N$$

we obtain RLS equations.

Other examples: Channel equalization, deconvolution, almost identical to Kalman filter (covered later).

Stochastic Least Squares

Model $y[k] = b_1^0 u[k-1] + b_2^0 u[k-2] + \ldots + b_n^0 u[k-n] + v[k]$

1. $u[k]$ are known inputs
2. $v[k]$ is zero mean noise – further assumptions later.

$$\boxed{Y_N = \Psi_N \theta^0 + V}$$

$Y_n = (y[n], \ldots, y[N])'$, $\Psi_N = (\psi[n], \ldots, \psi[N])'$,
$\theta^0 = (b_1^0, \ldots, b_n^0)'$, $\psi[k] = [u[k-1], \ldots, u[k-n]]'$,

Remarks: 1. Note $y[k]$ stochastic.
2. $\psi[k]$ deterministic
3. In an AR model

$$y[k] = a_1 y[k-1] + b_1 u[k-1] + e[k]$$

$\psi[k] = (y[k-1], u[k-1])'$ is stochastic.
We wont consider models where $\psi[k]$ is stochastic.

Stochastic Least Squares Estimate:
$\theta_{\mathrm{LS}}(Y_N) = \arg\min_\theta J(\theta) = (\Psi_N' \Psi_N)^{-1} \Psi_N' Y_N$ where

$$J(\theta) = (Y_N - \Psi_N'\theta)'(Y_N - \Psi_N'\theta) = \sum_{k=n}^{N} e^2[k;\theta]$$

where $e[k;\theta]$ depends on noise and choice of θ. Note $e[k;\theta^0] = v[k]$.

Our aim is to analyse how good is the LS estimate

$$\theta_N = (\Psi_N' \Psi_N)^{-1} \Psi_N' Y_N$$

We will consider 2 measures of performance
1. Unbiasedness (finite sample property)
2. Consistency (asymptotic property).

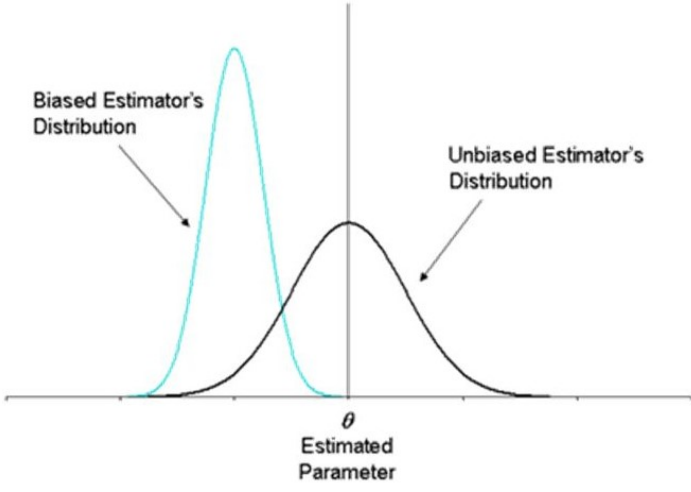
Unbiasedness

Definition: Let θ^0 denote the true model. An estimator θ_N is unbiased if $\mathbb{E}\{\theta_N\} = \theta^0$

Remarks: Finite sample property. Difficult to ensure in general. Many estimators are biased

Theorem: [Thm 1.1 in math stat] Under assumptions

(i) $Y_N = \Psi_N \theta^0 + V$, $\theta^0 < \infty$
(ii) Ψ_N is non-stochastic
(iii) $\Psi_N' \Psi_N$ is nonsingular
(iv) $\mathbb{E}\{V\} = 0$
(v) $V \sim N(0, \sigma^2 I)$, $\sigma^2 < \infty$.
Then following results hold:



(a) Existence: Given (i) to (iii), θ_N exists and is unique.
(b) Unbiased: Given (i) to (iv): θ_N is unbiased.
(c) Normality: Given (i) to (v), $\theta_N \sim N(\theta^0, \sigma^2 (\Psi_N' \Psi_N)^{-1})$
(d) Efficiency: Under (i) to (v) θ_N is ML estimator and attains CR bound.

Proof of (b).

$$\theta_N - \theta^0 = (\Psi_N'\Psi_N)^{-1}\Psi_N' Y_N - \theta^0$$
$$= (\Psi_N'\Psi_N)^{-1}\Psi_N'(\Psi_N\theta^0 + V) - \theta^0 = (\Psi_N'\Psi_N)^{-1}\Psi_N' V$$

Proof of (c). Due to (v), θ_N is Gaussian.

$$\begin{aligned}\text{cov}(\theta_N) &= \mathbb{E}\{(\Psi_N'\Psi_N)^{-1}\Psi_N' VV'\Psi_N(\Psi_N'\Psi_N)^{-1}\} \\ &= (\Psi_N'\Psi_N)^{-1}\Psi_N'\mathbb{E}\{VV'\}\Psi_N(\Psi_N'\Psi_N)^{-1} \\ &= (\Psi_N'\Psi_N)^{-1}\Psi_N'\sigma^2 I\Psi_N(\Psi_N'\Psi_N)^{-1} \\ &= \sigma^2(\Psi_N'\Psi_N)^{-1}\end{aligned}$$

Remark: Smaller covariance, better the estimator.

(ii)**Mean square consistency**

Definition: θ_N is mean square consistent if

$$\lim_{N\to\infty} \mathbb{E}\{(\theta_N - \theta^0)'(\theta_N - \theta^0)\} = 0$$

or equivalently if

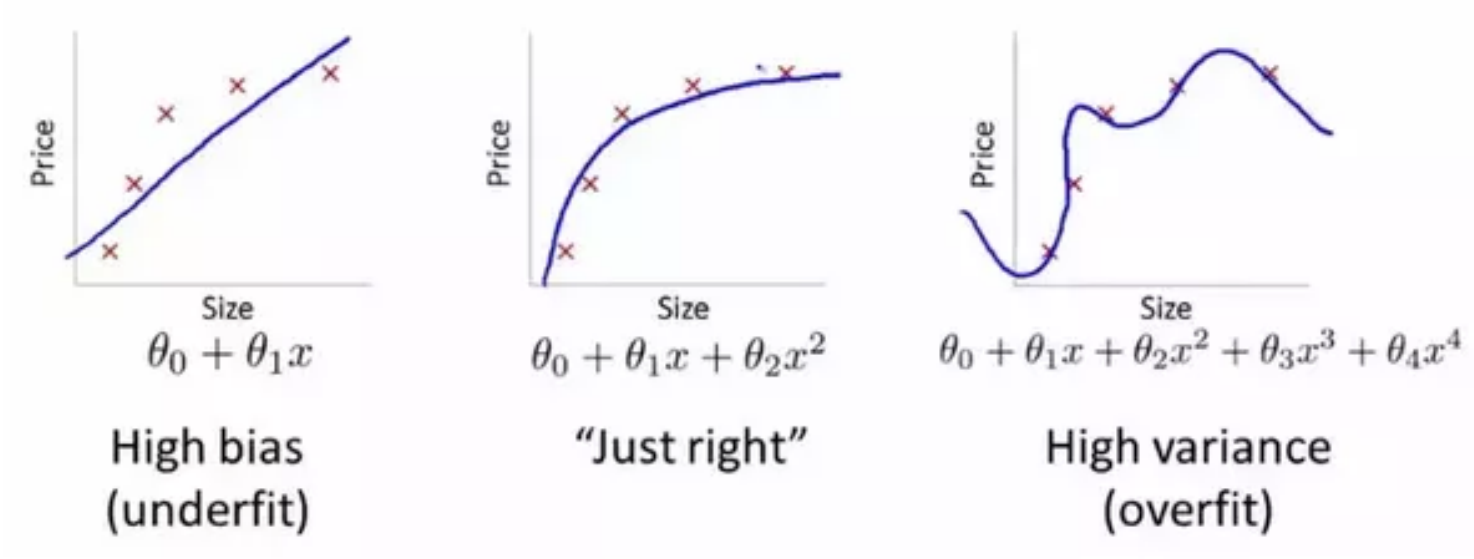
$$\lim_{N\to\infty} \text{tr } \mathbb{E}\{(\theta_N - \theta^0)(\theta_N - \theta^0)'\} = \text{tr cov}(\theta_N) = 0$$

Theorem: Under (i) to (iv) with $\text{cov}(V) = \sigma^2 I$, least square

estimate θ_N is ms consistent if

$$\lim_{N\to\infty} \sigma^2 \text{trace}(\Psi_N \Psi_N')^{-1} = 0$$

Bias Variance Tradeoff & Model Order Estimation



Model: $Y = \theta^o + \epsilon$, where $\mathbb{E}\{\epsilon\} = 0_n$, $\text{cov}\{\epsilon\} = \sigma^2 I$.

If no constraint on dimension, least square estimate $\theta^* \in \mathbb{R}^n$ is $\theta^* = Y$. Often, this "overfits" the data.

If θ^* constrained to dimension $d < n$, what is bias and variance? This gives a principled approach for selecting the model dimension given a real dataset.

We can encode model dimension as a subspace constraint.

Aim: Given $Y = \theta^o + \epsilon$, where $\mathbb{E}\{\epsilon\} = 0_n$, $\text{cov}\{\epsilon\} = \sigma^2 I$. Compute

$$\boxed{\theta^* = \underset{\theta}{\text{argmin}} \frac{1}{2}\|Y - \theta\|^2 \text{ such that } A_{m\times n}\theta_{n\times 1} = 0_m}$$

The m constraints imply θ lives in $n - m$ dim subspace.

Ex1: If $n = 3$ and $\theta(1) = 0$ (so $m = 1$ constraint), then θ lives in $d = n - m = 2$ dimension subspace. Constraint matrix

$$A = [1, 0, 0]$$

Ex2: If $n = 3$, $\theta(1) = \theta(2) = 0$ (so $m = 2$ constraints), then θ lives in $d = n - m = 1$ dimension subspace. Constraint matrix

$$A = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}$$

Ex 3: Here is another type of constraint: $n = 3$ and $\theta(1) + \theta(2) + \theta(3) = 0$ (so $m = 1$ constraint), then θ lives in 2 dimension subspace. Constraint matrix

$$A = [1, 1, 1]$$

Model: $Y = \theta^o + \epsilon$, $\mathbb{E}\{\epsilon\} = 0_n$, $\text{cov}\{\epsilon\} = \sigma^2 I$, $\theta^o \in \mathbb{R}^n$.
Constraint: $A_{m\times n}\theta_{n\times 1} = 0_m \implies d = n - m$ free parameters

Model order estimation proceeds in 3 steps:

Step 1: Compute Dimension Constrained Least Squares Estimator (Result 1 below)

Step 2: Compute Risk of Dimension Constrained Least Squares Estimator (Result 2 below)

Step 3: Give Model order estimation algorithm.

Overview of Main Result.

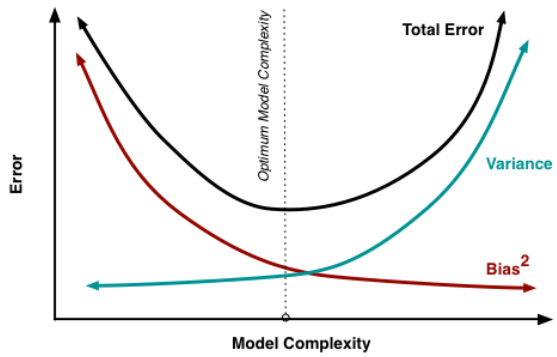
1. Let $d = n - m$ denote effective dim of θ_d^*. Then

$$\theta_d^* = (I - A'(AA')^{-1}A)Y = (I - P)Y \text{ where } P = A'(AA')^{-1}A$$

2. Akaike Information Criterion (AIC): Estimate model dimension d as:

$$\hat{d} = \underset{d}{\text{argmin}} \, \|Y - \theta_d^*\|^2 + 2d\,\sigma^2$$

First term (bias) $\downarrow d$. Second term (variance) $\uparrow d$ penalizes choosing high model dimension. (Penalized likelihood method)



Model: $Y = \theta^o + \epsilon$, where $\mathbb{E}\{\epsilon\} = 0_n$, $\text{cov}\{\epsilon\} = \sigma^2 I$, $\theta^o \in \mathbb{R}^n$.

Dimension Constrained Least Squares Estimator

Aim: Compute $\theta^* = \text{argmin}_\theta \frac{1}{2}\|Y - \theta\|^2$ s.t. $A_{m\times n}\theta_{n\times 1} = 0_m$

The m constraints imply θ lives in $n - m$ dim subspace.

Result 1: Let $d = n - m$ denote effective dim of θ_d^*. Then

$$\boxed{\theta_d^* = (I - A'(AA')^{-1}A)Y = (I - P)Y \text{ where } P = A'(AA')^{-1}A}$$

Remarks:

(i) If no constraint, then $A = P = 0_{n\times n}$. So $d = n$ and $\theta_d^* = Y$.

(ii) $\theta_d^* \in \mathbb{R}^n$ and satisfies constraint $A_{m\times n}\theta_{n\times 1} = 0_m$

(iii) $P = A'(AA')^{-1}A$ is the $n \times n$ orthogonal projection matrix to $d = n - m$ dimension subspace.

Verify $P = P' = P^2$ and $\text{trace}(P) = m$ since

$\text{trace}(AB) = \text{trace}(BA)$

Example: If $A_{2\times 3} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}$ then $P_{3\times 3} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$.

Proof of Result 1. Lagrangian is $L_\theta = \frac{1}{2}\|Y - \theta\|^2 + \lambda' A\theta$.

$$\nabla L_\theta = 0 \implies Y - \theta_d^* + A'\lambda = 0 \implies \theta_d^* = Y + A'\lambda$$

Constraint yields $A\theta_d^* = AY + AA'\lambda = 0 \implies \lambda = -(AA')^{-1}Y$.

$$\theta_d^* = (I - A'(AA')^{-1}A)Y$$

Result 2. Model: $Y = \theta^o + \epsilon_k$, $\mathbb{E}\{\epsilon\} = 0_n$, $\text{cov}\{\epsilon\} = \sigma^2 I$, $\theta^o \in \mathbb{R}^n$

How to estimate model dimension d***?***

$$\boxed{\begin{aligned}\text{Risk}_d = \mathbb{E}\{\|\theta_d^* - \theta^o\|^2\} &= \underbrace{\|P\theta^o\|^2}_{(\text{bias})^2} + \underbrace{\sigma^2 d}_{\text{variance}} \\ &= \mathbb{E}\{\|Y - \theta_d^*\|^2\} + \sigma^2(2d - n)\end{aligned}}$$

Clearly, variance $\uparrow$ as model dimension $d = n - m$ increases.

Bias $\downarrow$ as model dimension $d \uparrow$ as we now show.

Suppose m constraints on $\theta \in \mathbb{R}^n$ are $\theta(1) = \cdots = \theta(m) = 0$.

(i) $d = n$, $m = 0$, then $A = P = 0$, bias $= 0$. (overfitting data)

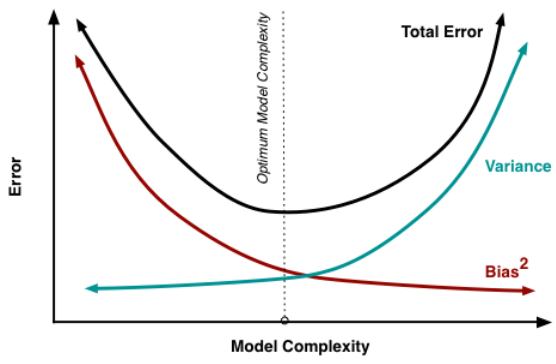
(ii) $d = n - 1$, $m = 1$, $A_{1\times n} = [1, 0, \ldots, 0]$, then $\text{bias}^2 = \theta^2(1)$.

(iii) $d = n - 2$, $m = 2$, $A_{2\times n} = \begin{bmatrix} 1 & 0 & \ldots 0 \\ 0 & 1 & \ldots 0 \end{bmatrix}$, $\text{bias}^2 = \theta^2(1) + \theta^2(2)$.

Result 3. (Model order estimation) Akaike Information Criterion (AIC): Estimate model dimension d as:

$$\boxed{\hat{d} = \underset{d}{\text{argmin}} \, \|Y - \theta_d^*\|^2 + 2d\,\sigma^2}$$

First term (bias) $\downarrow d$. Second term (variance) $\uparrow d$ penalizes choosing high model dimension. (Penalized likelihood method)



Proof of Result 2. Recall $P = A'(AA')^{-1}A$ is $n \times n$ matrix;
A is $m \times n$ matrix if there are m constraints.
Recall $P = P' = P^2$ and $\text{trace}(P) = m$. (since $\text{trace}(AB) = \text{trace}(BA)$)

$$\theta_d^* - \theta^o = (I-P)Y - \theta^o = (I-P)(\theta^o + \epsilon) - \theta^o = -P\theta^o + (I-P)\epsilon$$

$$\mathbb{E}\{\epsilon\} = 0 \implies \mathbb{E}\{\|\theta_d^* - \theta^o\|^2\} = \|P\theta^o\|^2 + \sigma^2 \text{trace}(I-P)(I-P)'$$

$$\text{trace}(I-P)(I-P)' = \text{trace}(I - 2P + P^2) = \text{trace}(I-P)$$
$$= n - m = d.$$

Proof of Result 3. AIC derivation:

$$Y - \theta_d^* = \theta^o + \epsilon - (I-P)(\theta^o + \epsilon) = P(\theta^o + \epsilon)$$

$$\implies \mathbb{E}\{\|Y - \theta_d^*\|^2\} = \|P\theta^o\|^2 + \sigma^2 \text{trace}(P) = \|P\theta^o\|^2 + \sigma^2\, m$$

Recall Result 2 that $\text{Risk}_d = \|P\theta^o\|^2 + \sigma^2 d$ where $d = n - m$. So

$$\text{Risk}_d = \mathbb{E}\{\|Y - \theta_d^*\|^2\} - \sigma^2\, m + \sigma^2 d = \mathbb{E}\{\|Y - \theta_d^*\|^2\} + \sigma^2(2d - n)$$

But we dont know $\mathbb{E}$ since we dont know density of Y.
Unbiased estimate of risk is

$$\hat{R}_d = \|Y - \theta_d^*\|^2 + \sigma^2(2d - n)$$

AIC estimates model dimension as

$$d^* = \underset{d}{\text{argmin}}\, \hat{R}_d = \underset{d}{\text{argmin}} \left[\|Y - \theta_d^*\|^2 + 2d\,\sigma^2\right]$$

since the term $-\sigma^2 n$ is irrelevant to optimization wrt d.

Several other model order estimators: BIC, MDL. For $Y \in \mathbb{R}^n$,
Bayesian Information criterion: $d^* = \text{argmin}_d \left[\|Y - \theta_d^*\|^2 + (\ln n) d\,\sigma^2\right]$

Regularized Least Squares - Main Ideas

$$Y_{N\times 1} = \Psi_{N\times n}\theta_{n\times 1}$$

Classical least squares: For $N \gg n$ (overdetermined case),

$$\underset{\theta}{\operatorname{argmin}} \|Y_{N\times 1} - \Psi_{N\times n}\theta\|^2$$

Minimum Norm Soln: For $n \gg N$ (underdetermined case),

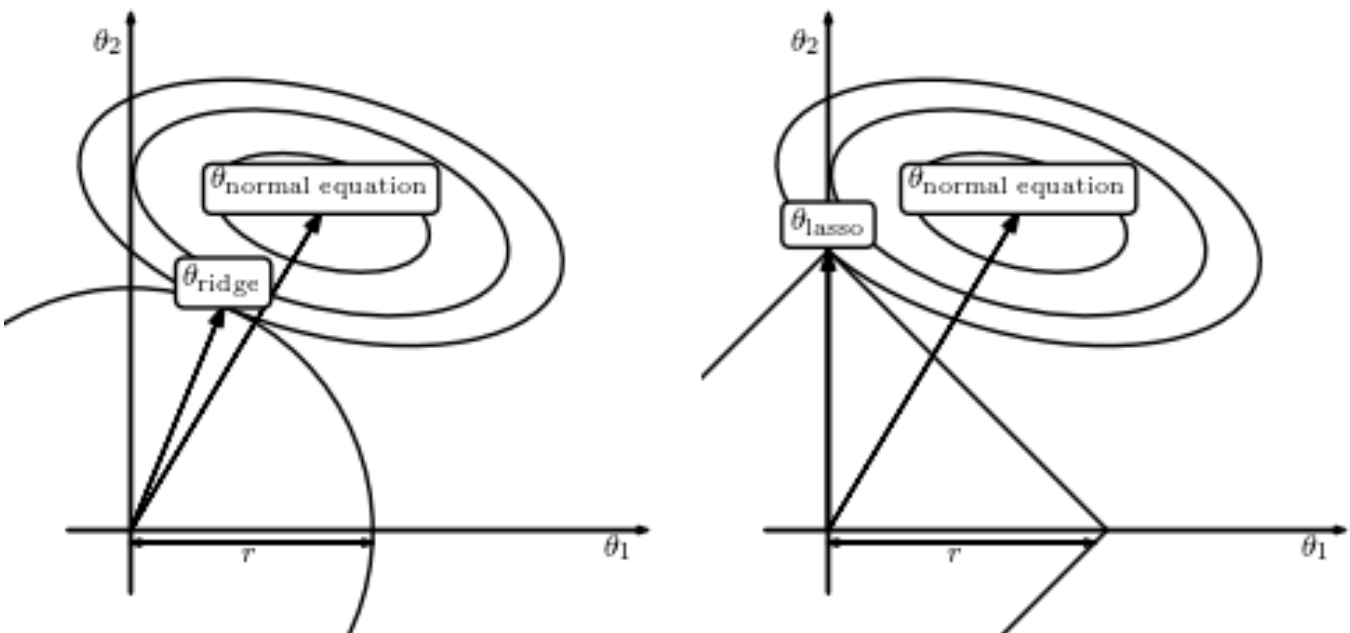
$$\underset{\theta}{\operatorname{argmin}} \|\theta\|^2 \text{ subject to } Y = \Psi\theta$$

Ridge Regression: For $\lambda \in \mathbb{R}_+$, $\operatorname{argmin}_\theta \|Y - \Psi\theta\|^2 + \lambda\|\theta\|_2^2$

$$\boxed{\theta_R = (\Psi'\Psi + \lambda I)^{-1}\Psi' Y} \quad \text{(c.f. Moore Penrose limit)}$$

Least Absolute Shrinkage & Selection Operator LASSO

$$\underset{\theta}{\operatorname{argmin}} \|Y - \Psi\theta\|^2 + \lambda\|\theta\|_1$$



Sparsity & Compressed Sensing

Compressed Sensing. Recover sparse vector: $n \gg N$,

$$\min \|\theta\|_0 = \#\{l : \theta_l \neq 0\} \text{ s.t. } Y_{N\times 1} = \Psi_{N\times n}\theta_{n\times 1}$$

Combinatorial optimization problem: search $\binom{n}{N}$ spaces S^m:

$$\theta^* = \min_m \arg\min_{\theta\in S^m} \|Y - \Psi\theta\|^2 + \lambda\|\theta\|_0$$

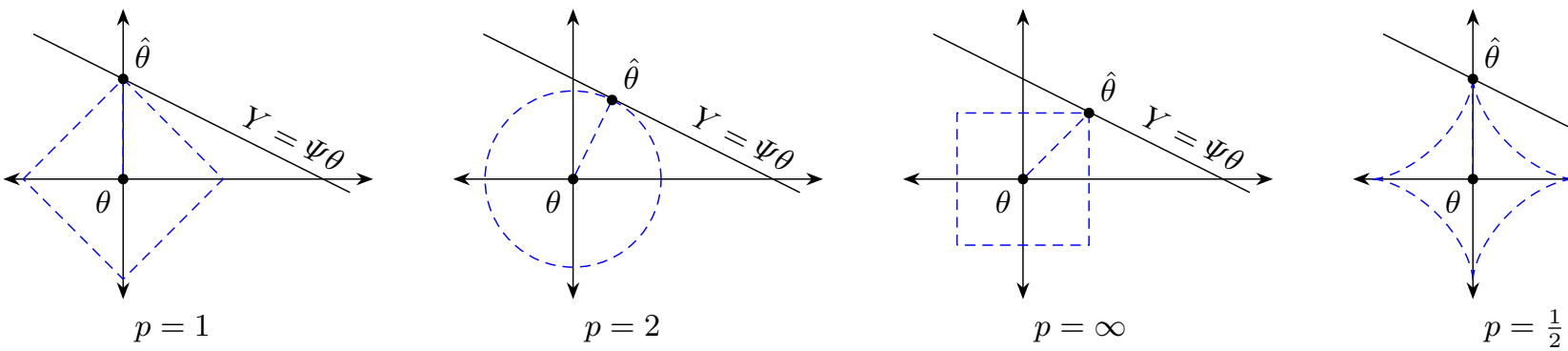
where $\|\theta\|_0 = \text{support}(\theta)$. Non-convex non-smooth optimization. LASSO replaces $\|\theta\|_0$ with convex $\|\theta\|_1$:

$$\theta^* = \arg\min_{\theta\in S^m} \|Y - \Psi\theta\|^2 + \lambda\|\theta\|_1$$

Optimization wrt l_1 norm yields l_0 soln with high probability

Main Idea: l_1-norm promotes sparsity.

$$\min \|\theta\|_p = (\sum_{i=1}^{n} |\theta_i|^p)^{1/p} \text{ s.t. } Y_{N\times 1} = \Psi_{N\times n}\theta_{n\times 1}$$



For $p \leq 1$ solns are sparse. But for $p < 1$, optimization problem is non-convex (hard to solve). Hence $p = 1$ is useful choice.

https://web.stanford.edu/~hastie/StatLearnSparsity_files/SLS_corrected_1.4.16.pdf

Big Data: Types of Sparsity.

$$Y = \Psi\theta^o + \epsilon, \quad \theta \in \mathbb{R}^n$$

1. Coordinate Sparsity: Only a few elements of θ^o are non-zero. LASSO:

$$\theta^* = \underset{\theta}{\operatorname{argmin}} \|Y - \Psi\theta|^2 + \lambda \sum_{i=1}^{n} |\theta_i|$$

2. Variation Sparsity: Only a few $\theta^o_{i+1} - \theta^o_i$, $i = 1, 2, \ldots, n-1$ are non-zero. (e.g step functions) Fused-LASSO

$$\theta^* = \underset{\theta}{\operatorname{argmin}} \|Y - \Psi\theta|^2 + \lambda \sum_{i=1}^{n-1} |\theta_{i+1} - \theta_i|$$

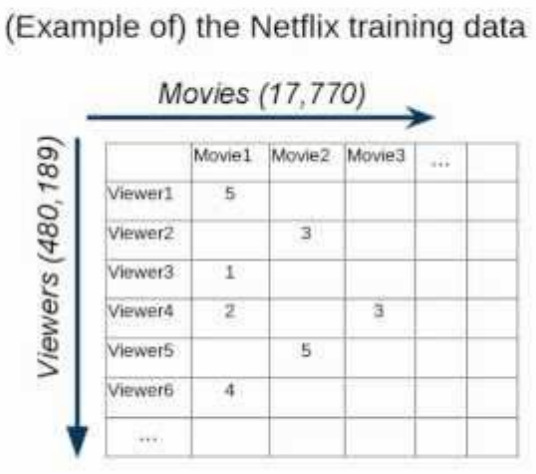
3. Group Sparsity: Suppose we can partition $(1, \ldots, n) = \cup_{k=1}^{M} G_k$ and only a few vectors $(\theta_i, i \in G_k)$ are non-zero. Assume we know the groups $G_k, k = 1, \ldots, M$. Group LASSO

$$\theta^* = \underset{\theta}{\operatorname{argmin}} \|Y - \Psi\theta|^2 + \sum_{k=1}^{M} \lambda_k \|\theta_{G_k}\|_1$$

4. Sparse Group Sparsity: Group sparse and the vectors $\theta_i, i \in G_k$ themselves are sparse. Sparse Group LASSO

$$\theta^* = \underset{\theta}{\operatorname{argmin}} \|Y - \Psi\theta|^2 + \sum_{k=1}^{M} \lambda_k \|\theta_{G_k}\|_1 + \|\theta\|_1$$

Matrix Completion problem. Rank Sparsity



99% of ratings missing. How to fill in missing entries to design recommender system?

Assume rank of matrix X is small.

`https://statweb.stanford.edu/~candes/papers/MatrixCompletion.pdf`

Find $Z_{n\times m}$ of small rank r such that training error is small:

$$\min_{Z} \sum_{i,j\in \text{observed}} |X_{ij} - Z_{ij}|^2 \quad \text{s.t.} \quad \text{rank}(Z) = r$$

Soln: PCA.

A more general approach: If additional row sparsity constraints, then cannot use PCA.

Rank constraint is non convex; so difficult to solve.

Main idea: replace rank with nuclear norm which is convex.

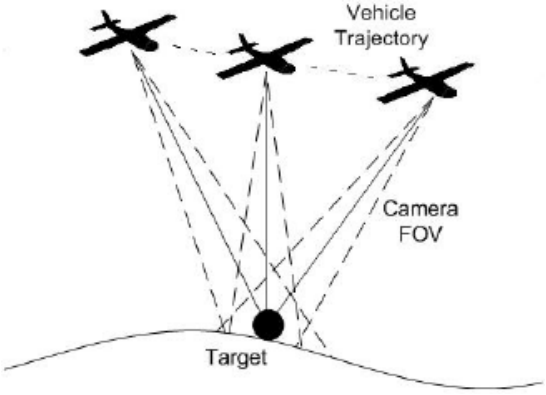
$\|Z\|_* = \sum_i \sigma_i(Z)$ = sum of singular values of matrix.

Aim. Solve following convex optimization problem:

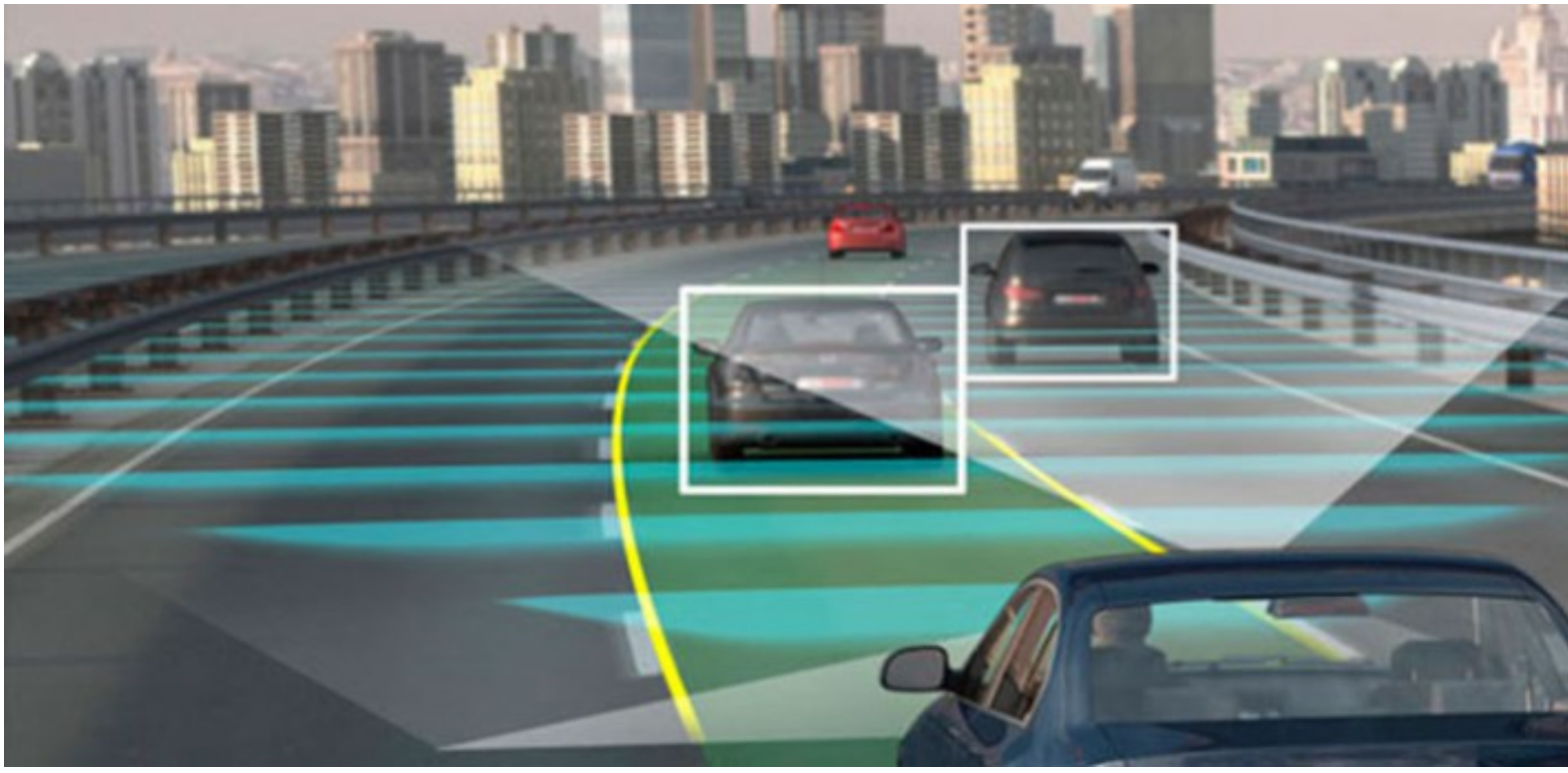
$$\boxed{\min_{Z} \sum_{i,j\in \text{observed}} |X_{ij} - Z_{ij}|^2 + \lambda \|Z\|_*, \quad \lambda > 0}$$

Part III. Bayesian Estimation

Localization: Given noisy measurements of a random variable x, how to estimate the location of x?



Tracking. Given noisy measurements of a random process x_k, how to estimate x_k? Self driving cars [Optimal Filtering]



Classical Bayesian Estimation

Model: obs y is a probabilistic function of random state x.

1. state $x \sim \pi_0$ where π_0 is a prior pdf/pmf
2. observation $y \sim p(y|x)$ (observation/sensor likelihood).

Aim: Given model π_0 and $p(y|x)$, estimate x given obs y.

4 Main Results:

1. Optimal estimate of state x given obs y is conditional mean:

$$\hat{x} \stackrel{\text{defn}}{=} \mathbb{E}\{x|y\}$$

CM is optimal in the sense of minimizing mean square error:

$$\mathbb{E}\{(\hat{x} - x)^2\} \leq \mathbb{E}\{(g(y) - x)^2\}$$

for any other estimator $g(y)$ of the state x.

2. Bayes rule:

$$\boxed{p(x|y) = \frac{\pi_0(x)p(y|x)}{\int_z \pi_0(z)p(y|z)dz}}$$

$$\boxed{\text{posterior} \ \propto \ \text{prior} \ \times \ \text{likelihood}}$$

3. The conditional mean estimate of x given observation y is

$$\boxed{\hat{x} = \mathbb{E}\{x|y\} = \int x\, p(x|y)dx}$$

4. The maximum aposteriori (MAP) estimate of x given y is:

$$x^{\text{MAP}} = \underset{x}{\text{argmax}}\, p(x|y) \qquad \text{(Bayesian classifier)}$$

x^{MAP} minimizes mis-classification prob $P(g(y) \neq x|y)$.

Example 1: Discrete rv case. Suppose $x \in \{1,2\}$, $y \in \{1,2\}$;

$\pi_0 = [0.2, 0.8]'$,

$P(y\|x)$	$y=1$	$y=2$
$x=1$	0.6	0.4
$x=2$	0.3	0.7

Suppose we observe $y = 1$. Compute Bayesian estimate of x.

$$P(x=1|y=1) = \frac{\pi_0(1)P(y=1|x=1)}{\pi_0(1)P(y=1|x=1)+\pi_0(2)P(y=1|x=2)} = \frac{1}{3}$$

$$P(x=2|y=1) = \frac{\pi_0(2)P(y=1|x=2)}{\pi_0(1)P(y=1|x=1)+\pi_0(2)P(y=1|x=2)} = \frac{2}{3}$$

MAP estimate is $x^{\text{MAP}} = \text{argmax}_x\, P(x|y=1) = 2$.
CM estimate is $\hat{x} = 1 \times 1/3 + 2 \times 2/3 = 1.667$.

Remarks: CM = MMSE = min var estimator

1. Conditional mean: soft estimate. MAP: hard estimate. MAP estimator = Bayes Classifier in machine learning.
2. $p(y|x)$ is "likelihood" - plausibility y was generated from x. Maximum likelihood estimate (MLE) is

$$\boxed{x^{\text{MLE}} = \underset{x}{\text{argmax}}\, p(y|x)}$$

   MLE does not use prior info. For above example $x^{\text{MLE}} = 1$.
3. If prior is non-informative i.e., $\pi_0(x)$ is constant wrt x, then $p(x|y) = p(y|x)$ and so MAP = MLE.
4. If likelihood is non-informative, i.e., $p(y|x)$ is indpt of x, then $p(x|y) = \pi_0(x)$. MLE is useless; $x^{\text{MAP}} = \text{argmax}_x\, \pi_0(x)$.

Proof of Optimality of Conditional Mean Estimator

Theorem: Suppose $\hat{x} \stackrel{\text{defn}}{=} \mathbb{E}\{x|y\}$. Then for any estimator $g(y)$,

$$\mathbb{E}\{(\hat{x}-x)^2\} \leq \mathbb{E}\{(g(y)-x)^2\} \iff \operatorname*{argmin}_g \mathbb{E}\{(g(y)-x)^2\} = \hat{x}.$$

Proof:

$$\operatorname{argmin}_g \mathbb{E}\{(g(y)-x)^2\} = \operatorname{argmin}_g \mathbb{E}\{(g(y)-\hat{x}+\hat{x}-x)^2\} =$$
$$\operatorname{argmin}_g \mathbb{E}\{(g(y)-\hat{x})^2\} + \underbrace{\mathbb{E}\{(\hat{x}-x)^2\}}_{\text{irrelevant}} + 2\underbrace{\mathbb{E}\{(g(y)-\hat{x})'(\hat{x}-x)\}}_{0}.$$

Last term is zero follows from law of iterated expectations.

$$\mathbb{E}\{(g(y)-\hat{x})'(\hat{x}-x)\} = \mathbb{E}\{\mathbb{E}\{(g(y)-\hat{x})'(\hat{x}-x)|Y\}\}$$
$$= \mathbb{E}\big\{(g(y)-\hat{x})'\mathbb{E}\{(\hat{x}-x)|Y\}\big\} = 0$$

Example. Linear Discriminant Analysis

LDA = MAP estimator = $\operatorname{argmax}_x p(x|y)$

Has applications in

1. Face recognition
2. Pattern recognition

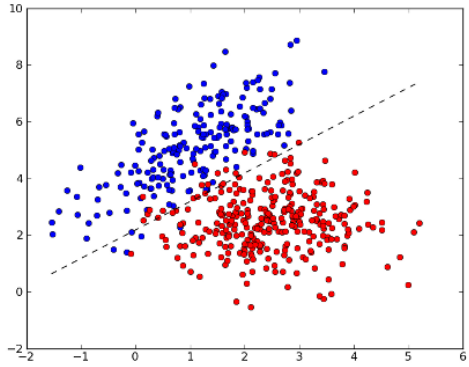
Example 1. Bayesian Classifier and Linear Discriminant Analysis (LDA)

Given data $y_1, \ldots, y_n$, classify into $x_1, \ldots, x_n$ where $x_i \in \{\text{spam} = 1, \text{ not spam} = 2\}$. So y is noisy observation of x.

Aim: Using training data construct optimal classifier $g(y)$ to minimize

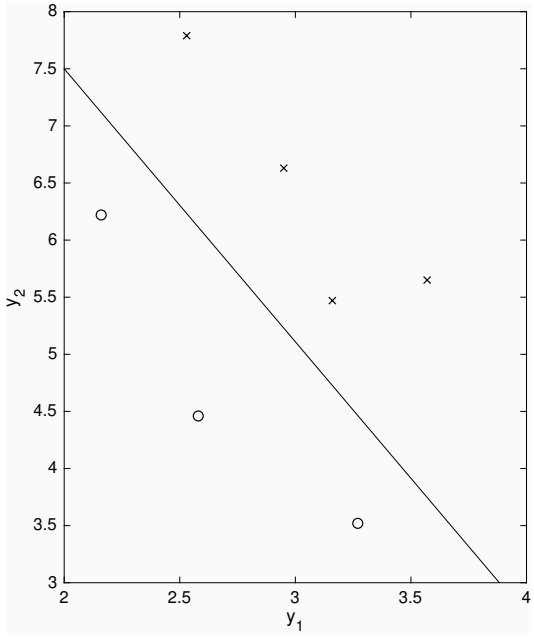
$$P(x \neq g(y)|y) = \text{prob of miss-classification}$$

Fig shows $y_i \in \mathbb{R}^2$ and linear classifier (in y)



Example: A factory produces optical lenses. Quality measured in terms of curvature and diameter. From quality control data:

Curvature $y(1)$	Diameter $y(2)$	Result	x
2.95	6.63	Passed	1
2.53	7.79	Passed	1
3.57	5.65	Passed	1
3.16	5.47	Passed	1
2.58	4.46	Not Passed	2
2.16	6.22	Not Passed	2
3.27	3.52	Not Passed	2



Aim: Use this training data to construct a Bayesian classifier.

Result: Bayes classifier that minimizes $P(x \neq g(y)|y)$ is MAP estimate

$$\boxed{g(y) = \arg\max_x P(x|y) = \arg\max_x \frac{P(y|x)P(x)}{\cancel{P(y)}}}$$

Proof: Let $u = g(y) \in \{1, 2\}$ = classification decision.

$$\text{Compute } u^* = \arg\min_u \mathbb{E}\{L(x,u)|y\} \text{ where } L(x,u) = \begin{cases} 1 & x \neq u \\ 0 & \text{otherwise} \end{cases}$$

$$\arg\min_u \mathbb{E}\{L(x,u)|y\} = \arg\min_u \sum_x L(x,u)P(x|y)$$

$$= \arg\min_u \sum_{x \neq u} P(x|y) = \arg\min_u (1 - P(u|y)) = \arg\max_u P(u|y)$$

Since $X \in \{1, 2\}$, Bayes classifier is

$$g(y) = \begin{cases} 1 & P(x=1|y) > P(x=2|y) \\ 2 & \text{otherwise} \end{cases} = \begin{cases} 1 & \log \frac{P(y|1)}{P(y|2)} + \log \frac{P(1)}{P(2)} > 0 \\ 2 & \text{otherwise} \end{cases}$$

In machine learning, often likelihood $P(y|x)$ and prior $P(x)$ not known. Given training data $(x_1, \ldots, x_n, y_1, \ldots, y_n)$ how to build classifier?

1. Parameteric Classifier: assume Gaussian $P(y|x)$ and estimate parameters from data – linear discriminant anaylsis
2. Semi-parametric Classifier: assume a logistic model for posterior and estimate parameters from data

Method 1. Parametric Model. Assume Gaussian likelihood $P(Y|X) \sim N(\mu_X, \Sigma)$. Denote prior as $\pi(i)$. Use training data to estimate $\hat{\pi}, \hat{\Sigma}, \hat{\mu}_X$. Then Bayes classifier is linear y:

$$g(y) = \arg \max_{i\in\{1,2\}} \{\log \pi(i) - \frac{1}{2}(y-\mu_i)' \Sigma^{-1}(y-\mu_i)\}$$

$$= \arg \max_{i\in\{1,2\}} \{\log \pi(i) + \frac{1}{2}(2\mu_i' \Sigma^{-1} y - \mu_i' \Sigma^{-1} \mu_i)\}$$

$$\boxed{\text{LDA} : \hat{g}(y) = \arg \max_{i\in\{1,2\}} \{\log \hat{\pi}(i) + \frac{1}{2}(2\hat{\mu}_i' \Sigma^{-1} y - \hat{\mu}_i' \hat{\Sigma}^{-1} \hat{\mu}_i)\}}$$

LDA Decision threshold is straight line in y:

$$\log \hat{\pi}(1) + \frac{1}{2}(2\hat{\mu}_1' \Sigma^{-1} y - \hat{\mu}_1' \hat{\Sigma}^{-1} \hat{\mu}_1) = \log \hat{\pi}(2) + \frac{1}{2}(2\hat{\mu}_2' \Sigma^{-1} y - \hat{\mu}_2' \hat{\Sigma}^{-1} \hat{\mu}_2)$$

Example (cont): From data $\hat{\mu}_1 = [3.05, 6.38]'$; $\hat{\mu}_2 = [2.67, 4.73]'$, $\Sigma^{-1} = \begin{bmatrix} 5.745 & 0.791 \\ 0.791 & 0.701 \end{bmatrix}$, $\hat{\pi}(1) = 4/7, \hat{\pi}(2) = 3/7$.
LDA decision threshold is $3.49y_1 + 1.46y_2 = 17.78$ (see figure).

Method 2. Semi-Parameteric Model. Assume posterior

$$P(X = 1|Y) = \frac{\exp(\alpha + \beta' y)}{1 + \exp(\alpha + \beta' y)} \text{ (called a logistic model)}$$

where parameters α, β are estimated from training data. Then

$$P(X = 1|Y) > 1/2 \iff \exp(\alpha + \beta' y) > 1 \iff \alpha + \beta y > 0$$

$$\boxed{\text{Semi-parametric classifier is } \hat{g}(y) = \begin{cases} 1 & \alpha + \beta y > 0 \\ 2 & \text{otherwise} \end{cases}}$$

Ex 2. Bayesian Interpretation of LASSO

Suppose $x \in \mathbb{R}^X$ is a random variable with Laplace (double exponential) prior pdf

$$p(x) = \prod_{j=1}^{X} \frac{\lambda}{2} \exp\left(-\lambda |x_j|\right).$$

Suppose x is observed via the observation equation

$$y = Ax + v, \qquad v \sim \boldsymbol{N}(0, I)$$

where A is a known $n \times X$ matrix. Then

$$p(x|y) \propto \exp\Big(-\frac{1}{2} \operatorname{Lasso}(x, y, \mu) \Big)$$

where $\mu > 0$ and

$$\operatorname{Lasso}(x, y, \mu) = \|y - Ax\|^2 + \mu \|x\|_1.$$

Therefore computing the maximum aposteriori estimate $\hat{x}$ is equivalent to computing the minimizer $\hat{x}$ of $\operatorname{Lasso}(x, y, \mu)$.

Likelihood, Prior, Bayesian Inference

1. How to compute likelihood $p(y|x)$?
2. How to specify prior $p(x)$?
3. How to compute $p(x|y) = \frac{p(y|x)p(x)}{\int p(y|\zeta)p(\zeta)d\zeta}$?

1. Likelihood Formula to compute $p(y|x)$.

Suppose $Y = \phi(X) + W$, and W has cdf F_W and pdf f_W. Then $F_{Y|X}(y|x) = P(Y \le y|X = x) = P(\phi(X) + W \le y|X = x) = P(W \le y - \phi(x)) = F_W(y - \phi(x))$. Take derivative wrt y:

$$\boxed{f_{Y|X}(y|x) = \frac{dF}{dy} F_{Y|X}(y|x) = f_W(y - \phi(x))}$$

Example 1: $Y = X + W$ where $W \sim N(0, \sigma^2)$. Then

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

Example 2: $Y = AX + DW$ where $A, D \in \mathbb{R}^{n\times n}$ and D is positive definite. Show that $p(y|x) = \frac{1}{|D|} p_W(y - Ax)$

2. How to specify prior? Philosophical question; not based on observations

1. Single prior (classical)
2. Family of priors indexed by hyper parameter θ.
   (i) Hierarchical Bayes: $p(y|x)$, $p(x|\theta)$, $p(\theta)$.
   (ii) Empirical Bayes: $p(y|x)$, $p(x|\theta)$; estimate θ from data.
   (iii) Robust Bayes - choose non-informative prior which treats all parameters equally.

Two types of priors: Non-informative and informative.

Non-informative Priors. also called diffuse or flat prior.

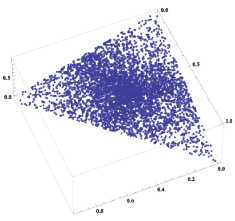
When nothing is known about x choose $p(x)$ as uniform pdf.

Example. Hierarchical Bayes Model and Dirichlet prior: used in Latent Dirichlet Allocation (LDA) in NLP.

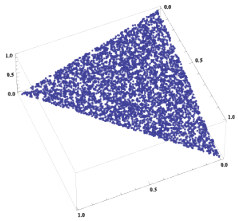
1. θ is an X-dim pmf chosen uniformly from all X-dim pmfs. $\theta \sim U(\Pi(X))$ where $\Pi(X) = \{\pi : \sum_{i=1}^{X} \pi(i) = 1, \pi(i) \geq 0\}$? $\Pi(X)$ is space of X-dim pmfs = unit $X-1$-dim simplex. (Example: $\theta = [0.1, 0.5, 0.4]'$.)
2. $x \sim \theta$. (Example. $x = 2$ with prob 0.5)
3. observation $y \sim p(y|x)$. (Example. $y = x +$ noise)

How to sample uniformly from space of X-dimensional pdfs?
$\Pi(X) = \{\pi : \sum_{i=1}^{X} \pi(i) = 1, \pi(i) \geq 0\}$? $\Pi(X)$ is space of X-dim pmfs = unit $X-1$-dimensional simplex.
Ans: Dirichlet distribution.

Algorithm for sampling uniformly from $\Pi(X)$.
(a) Sample $\zeta(i) \sim e^{-\zeta}, \zeta \geq 0, i = 1, \ldots, X$ (unit expo density)
(b) Set $\theta(i) = \zeta(i) / \sum_j \zeta(j)$.



Normalized Uniform Distr.



Normalized Exp. Distr.

Note: Generating $\zeta(i) \sim U[0, 1]$ in step (a) does not work.

Jeffreys Prior (Advanced)

For discrete rv, discrete uniform prior is non-informative.
Monotone fn of disc uniform is disc uniform.
For continuous rv, uniform prior not invariant to transformation. No info about x implies no info about x^2; but uniform prior for x is not equivalent to uniform prior for x^2.
Jeffreys prior: pdf that is invariant to monotone transformation.
What is invariance of prior $p(x)$? If $z = h(x)$, then pdf of z is

$$q(z) = \frac{p(h^{-1}(z))}{|h'(h^{-1}(z))|}$$

Invariance to transformation $h(x)$ means pdf p satisfies

$$p(z) = q(z) \equiv p(z) = \frac{p(h^{-1}(z))}{|h'(h^{-1}(z))|} \implies \boxed{p(h(z)) = \frac{p(z)}{|h'(z)|}}$$

Fisher information: measures curvature of log-likelihood

$$I(x) = \mathbb{E}_y\big\{\big(\nabla_x \log p(y|x)\big)^2\big\} = -\mathbb{E}_y\{\nabla_x^2 \log p(y|x)\}$$

High curvature implies deeper valley: easier to estimate x.
Cramer Rao bound: For unbiased estimator $\hat{x}$, $\mathrm{Var}\{\hat{x}\} \geq 1/I(x)$.

Jeffreys prior: $\boxed{p(x) \propto \sqrt{I(x)}}$ Jeffreys prior is scale invariant: For any monotone function $h(x)$ show that

$$I(x) = I(h(x))(\nabla_x h(x))^2 \implies p(x) \propto \underbrace{\sqrt{I(h(x))}}_{p(h(x))} |\nabla_x h(x)|.$$

Example 1. Gaussian obs with prior on mean μ:

$$p(y|\mu) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp(-\frac{1}{2}\frac{(y-\mu)^2}{\sigma^2})$$

$I(x) = -\mathbb{E}_y\{\nabla^2_\mu \log p(y|\mu)\} = -\mathbb{E}_y\{\nabla^2_\mu \frac{(-y-\mu)^2}{2\sigma^2}\} = \frac{1}{\sigma^2}$.
Jeffreys prior for mean is $p(\mu) =$ constant indpt of μ, i.e. uniform prior. Improper prior since it cannot be a valid density.

Example 2. Gaussian obs with prior on standard deviation σ:

$$p(y|\sigma) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp(-\frac{1}{2}\frac{(y-\mu)^2}{\sigma^2})$$

Jeffreys prior for std deviation is $p(\sigma) = \sqrt{I(\sigma)} \propto \frac{1}{\sigma}$.
Jeffreys prior for variance is $p(\sigma^2) = \sqrt{I(\sigma^2)} \propto (\sigma^2)^{-3/2}$.

Improper Priors. Since $p(x|y) = \frac{p(y|x)p(x)}{\int p(y|\zeta)p(\zeta)d\zeta}$, it follows that $p(x)$ does not have to be a valid pdf for $p(x|y)$ to be a valid pdf.

Example: $p(x) =$ positive constant for $x \in \mathbb{R}$ is an improper prior. Results in proper posterior if $p(y|x)$ is valid pdf.

3. How to compute posterior distribution?

Bayes rule: $p(x|y) = \frac{p(y|x)p(x)}{\int p(y|\zeta)p(\zeta)d\zeta}$
In general, obtaining $p(x|y)$ in closed form is intractable due to denominator involving multidim integral.

Two ways:

1. Analytically: Conjugate Priors (if nice structure) gives closed form for posterior
2. Numerically: MCMC sampling from posterior for large scale problems - computational Bayesian (hot area in machine learning and signal processing)

Conjugate Priors

Bayes: posterior $\propto$ likelihood $\times$ prior

Suppose prior & likelihood chosen s.t. posterior has same pdf family as prior. Then prior & likelihood form *conjugate pair*.

Example 1. Gaussian prior and likelihood are conjugate. Prior $x \sim N(\mu, \sigma_w^2)$; likelihood $y \sim N(x, \sigma_v^2)$, then Gaussian posterior.
Notation. X-variate Gaussian with mean μ and cov P:

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

Theorem 5 (Swiss-Army-Knife for Gaussians)**.** *Consider Gaussian likelihood* $\boldsymbol{N}(y; Cx, R)$ *and prior* $\boldsymbol{N}(x; \mu, P)$. *Then*

$$\begin{aligned}
&\boldsymbol{N}(y; Cx, R)\, \boldsymbol{N}(x; \mu, P) \\
&\qquad = \boldsymbol{N}(y; C\mu, CPC' + R)\ \boldsymbol{N}(x; m, P - \bar{K}CP) \\
&\qquad \textit{where } \bar{K} = PC'(CPC' + R)^{-1}, \quad m = \mu + \bar{K}(y - C\mu).
\end{aligned}$$

As a result, the following hold:

$$\int_{\mathcal{X}} \boldsymbol{N}(y; Cx, R)\boldsymbol{N}(x; \mu, P)dx = \boldsymbol{N}(y; C\mu, CPC' + R) \tag{6}$$

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

Main point: first Gaussian on RHS only involves y and not x. To marginalize product of Gaussians by integrating over x, we can take the first term on RHS outside integral and second term on RHS integrates to 1 – this yields (6).

There are several examples of conjugate priors.

1. Prior $x \sim N(\mu, \sigma_w^2)$; likelihood $y \sim N(x, \sigma_v^2)$ (above example).
2. Prior $\sigma^{-2} \sim \text{Gamma}(\alpha, \beta)$, likelihood $y \sim N(\mu, \sigma^2)$

See wikipedia for several additional examples.

Next: Computational Bayesian using MCMC. Key idea:

> *A cloud of samples from a distribution is equivalent to knowing the distribution*

Real life examples of offline Bayesian Inference:

1. Latent Dirichlet Allocation in NLP for topic modeling (in text mining, social media analysis, etc).
2. Optimal Search and Resource Allocation (dynamic spectrum allocation, defense applications)
3. Clinical Medical Trials (survival analysis for censored data)
4. Target localization (signal processing, defense)
5. Linear Discriminant Analysis (pattern recognition, face recognition)

I.4. Simulation Part 2: Markov Chain Monte-Carlo (MCMC)

Outline

1. Cont-state Markov Processes
2. Reversible Markov Chains
3. Metropolis Hasting Algorithm
4. MCMC for Bayesian inference
5. Gibbs Sampling for Bayesian inference

Aside: Continuous state Markov Chains

Consider stochastic difference equation:

$$x_{k+1} = \phi_k(x_k, w_k), \quad x_0 \sim \pi_0$$

State x_k lies in the state space $\mathcal{X} = \mathbb{R}^X$.
State noise $\{w_k\}$: iid sequence. Assume $\{w_k\}$, and x_0 are indpt.
Then x_k is a continuous state Markov process on $\mathbb{R}^X$:

$$\mathbb{P}(x_{n+1} \in S | x_1, x_2, \ldots, x_n) = \mathbb{P}(x_{n+1} \in S | x_n)$$

for any set $S \subseteq \mathbb{R}^X$.

Define *transition density* $P_{xy} = p(x_{k+1} = y | x_k = x)$: For $S \subseteq \mathcal{X}$,

$$\mathbb{P}(x_{k+1} \in S | x_k) = \int_S p(x_{k+1} = x | x_k)\, dx, \quad \int_{\mathcal{X}} p(x_{k+1} = x | x_k)\, dx = 1.$$

How to specify transition density? Use likelihood formula:
Suppose additive noise

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

Assuming $\Gamma_k(x_k)$ is square invertible matrix,

$$\boxed{p(x_{k+1}|x_k) = |\Gamma_k^{-1}(x_k)| \;\; p_w\left(\Gamma_k^{-1}(x_k)\; [x_{k+1} - A_k(x_k)]\right)}$$

Example: If $x_{k+1} = a x_k + w_k$, $w_k \sim N(0,1)$, transition density

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

Markov Chain Monte-Carlo (MCMC)

For irreducible aperiodic Markov process, stationary distribution of the transition probability matrix (transition density) satisfies

Finite state Markov chain: $\pi_\infty(j) = \sum_i P_{ij}\pi_\infty(i) \iff \pi_\infty = P'\pi_\infty$

Cont. state Markov chain: $\pi_\infty(y) = \int P_{xy}\pi_\infty(x)dx$

---

1. We know how to simulate a Markov chain $\{x_k\}$ with transition matrix (density)
2. Sample path $\{x_k\}$ has stationary distribution π_∞.
   SLLN implies: (i) empirical cdf of x_k converges to cdf π_∞
   (ii) $\frac{1}{N}\sum_{k=1}^N h(x_k) \to \sum_i h(i)\pi_\infty(i)$ as $N \to \infty$.

MCMC Key idea:

1. Given a target distribution π_∞, find a transition prob P with stationary distribution π_∞.
2. Simulate Markov process $\{x_k\}$ with this transition prob P.

How to construct transition matrix P so that π_∞ is stationary distribution? Metropolis-Hastings algorithm constructs a *reversible* Markov chain

---

Why sample from a Markov chain rather than simulate i.i.d.?
(i) impossible to simulate i.i.d. samples from multidim cdf.
(ii) often π_∞ known up to a normalizing constant. Computing normalizing constant is intractable. e.g. Bayes rule

Reversible Markov Chains

A MC with transition prob P and stationary distribution π_∞ is *reversible* if it satisfies "balanced equation"

$$\boxed{P_{ij}\,\pi_\infty(i) = P_{ji}\,\pi_\infty(j) \quad \text{OR} \quad P_{xy}\pi_\infty(x) = P_{yx}\pi_\infty(y)}$$

Remarks: (i) If there exists a probability vector π_∞ that satisfies above, then π_∞ must be stationary distribution (Why?)
(ii) Suppose $\{x_k\}$ is irreducible aperiodic Markov chain. Then

$$P(x_{n-1} = j | x_n = i, x_{n+1}, \ldots, x_{n+k}) = P(x_{n-1} = j | x_n = i) = \frac{\pi(j)P_{ji}}{\pi(i)}$$

So time-reversed process is Markov with transition probability

$$Q_{ij} = P(x_{n-1} = j | x_n = i) = \frac{\pi(j)P_{ji}}{\pi(i)}$$

In reversible Markov chain: $Q_{ij} = P_{ij}$, i.e, forward and backward transition probabilities are identical. So

$$P_{ij} = \frac{\pi(j)P_{ji}}{\pi(i)} \iff P_{ij}\pi(i) = P_{ji}\pi(j)$$

Running process forward or backward are statistically identical.
(iii) All eigenvalues of P are real. Reason: P is *similar* to a symmetric matrix, i.e., $\exists T$s.t. $TPT^{-1} = S$ for symmetric S.
Proof: Define $\Pi_\infty \stackrel{\text{defn}}{=} \text{diag}(\pi_\infty(1), \cdots, \pi_\infty(X))$.

Reversibility: $\Pi_\infty P = P'\Pi_\infty \implies \Pi_\infty^{1/2} P \Pi_\infty^{-1/2} = (\Pi_\infty^{1/2} P \Pi_\infty^{-1/2})'$

If $TPT^{-1} = S$, clearly P and S have identical eigenvalues since
$\det(\lambda I - TPT^{-1}) = \det(T^{-1}T)\det(\lambda I - TPT^{-1})$

Why reversible Markov chains? Given π_∞, much easier to construct reversible transition matrix that satisfies componentwise relationship, than to construct transition matrix that admits π_∞ as an eigenvector. That is,

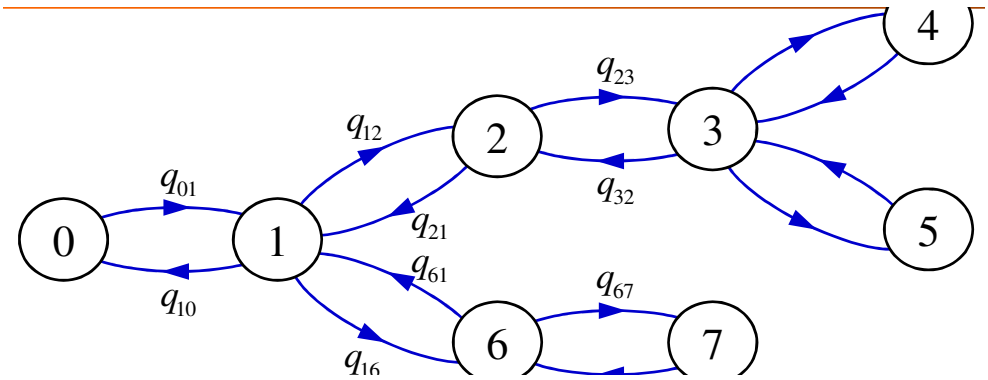
$$\text{Reversible MC:} \quad P_{ij}\,\pi_\infty(i) = P_{ji}\,\pi_\infty(j)$$

is simpler than $\pi_\infty(j) = \sum_i P_{ij}\pi_\infty(i)$ for general MC.

Also, optimizing the second largest eigenvalue modulus of a reversible transition matrix is a convex optimization problem.

How to tell if a Markov chain is reversible?
Result Sufficient condition: If the graph is a tree (contains no loops), then the Markov chain is reversible.

A Markov chain is a tree process if: (i) $P_{ij} > 0$ iff $P_{ji} > 0$.
(ii) For every $i, j, i \neq j.$, there is a unique sequence of distinct states $i = i_0, i_1, i_2, \ldots, i_{n-1}, i_n = j$ such that $P_{i_k, i_{k+1}} > 0$.



(i) Any 2 state Markov chain is reversible.

(ii) Any tri-diagonal transition matrix is reversible.

(iii) Any symmetric transition matrix is reversible.

(iv) Not reversible if non-symmetric zero element.

Kolmogorov conditions. A Markov chain is reversible iff any path starting from state i and back to state i, has the same probability as going in the reverse direction;

$$P_{i,i_1} P_{i_1,i_2} \cdots P_{i_k,i} = P_{i,i_k} P_{i_k,i_{k-1}} \cdots P_{i_1,i} \qquad \forall k$$

Example (i): 3×3 transition matrix is reversible if

$$P_{12} P_{23} P_{31} = P_{13} P_{32} P_{21}$$

(ii) Any 2 state Markov chain is reversible.

Result. A transition matrix is not reversible if it has a non-symmetric zero element.

Outline

1. Cont-state Markov Processes
2. Reversible Markov Chains
3. Metropolis Hasting Algorithm
4. MCMC for Bayesian inference
5. Gibbs Sampling for Bayesian inference

Metropolis Hastings Algorithm

Aim: Simulate from *target distribution* π_∞ by constructing a reversible Markov chain that has π_∞ as stationary distribution.

Metropolis-Hastings (discrete rv)

- Choose any irreducible transition matrix Q *(proposal distribution)*.
- Define acceptance probability

$$\alpha_{ij} = \min\left(1, \frac{\pi_\infty(j)Q_{ji}}{\pi_\infty(i)Q_{ij}}\right)$$

- Given state $x_k = i$ at time k:
  1. Simulate next state of the Markov chain $j \sim Q_{ij}$.
  2. Generate $u \sim U[0,1]$.
  3. If $u < \alpha_{ij}$ then set $x_{k+1} = j$ (accept); otherwise set $x_{k+1} = x_k = i$ (reject).

Metropolis-Hastings (continuous rv)

- Choose any transition kernel $q(y|x) > 0$ for all $x, y \in \mathcal{X}$. $q(y|x)$ is called *proposal distribution*.
- Define

$$\alpha_{xy} = \min\left(1, \frac{\pi_\infty(y)q(x|y)}{\pi_\infty(x)q(y|x)}\right)$$

- Given state $x_k = x$ at time k:
  1. Simulate next state of the Markov chain $y \sim q_{y|x}$.
  2. Generate $u \sim U[0,1]$.
  3. If $u < \alpha_{xy}$ then set $x_{k+1} = y$ (accept); otherwise set $x_{k+1} = x_k = x$ (reject).

Theorem: $\{x_k\}$ generated by Metropolis Hastings is a reversible Markov chain with stationary distribution π_∞.

Remarks:

1. Step 3 accepts $x_{k+1} = j$ with probability α_{ij} and remains in the state $x_k = i$ with probability $1 - \alpha_{ij}$.
2. Target distribution π_∞ only needs to be known up to a normalizing constant, since the acceptance probabilities of step 3 depend on the ratio $\pi_\infty(j)/\pi_\infty(i)$. This is crucially important in Bayesian inference (sampling from posterior).
3. If proposal distribution Q is chosen symmetric $Q_{ij} = Q_{ji}$

$$\alpha_{ij} = \min(1, \frac{\pi_\infty(j)}{\pi_\infty(i)})$$

   This is called random-walk Metropolis algorithm.
4. *Proof that Metropolis Hastings algorithm works*:
   Markov chain $\{x_k\}$ has transition probabilities

$$P_{ij} = \alpha_{ij} Q_{ij}, \quad j \neq i \qquad \text{where } \alpha_{ij} = \min\left(1, \frac{\pi_\infty(j) Q_{ji}}{\pi_\infty(i) Q_{ij}}\right) \tag{7}$$

   We prove x_k is a reversible MC, i.e., $P_{ij}\pi_\infty(i) = P_{ji}\pi_\infty(j)$.

$$P_{ij}\pi_\infty(i) = \alpha_{ij} Q_{ij} \pi_\infty(i) = \min(Q_{ij}\pi_\infty(i), Q_{ji}\pi_\infty(j))$$

   Suppose $\alpha_{ij} < 1$. Then (7) implies
   (a) $Q_{ji}\pi_\infty(j) < Q_{ij}\pi_\infty(i)$ so $P_{ij}\pi_\infty(i) = Q_{ji}\pi_\infty(j)$.
   (b) $\alpha_{ji} = 1$. So $P_{ji}\pi_\infty(j) = \alpha_{ji} Q_{ji}\pi_\infty(j) = Q_{ji}\pi_\infty(j)$.
   Therefore $P_{ij}\pi_\infty(i) = P_{ji}\pi_\infty(j)$.
   Hence x_k is reversible MC with stationary distribution π_∞.

Example 1. MCMC for n-dimensional numerical integration

$$E\{h(X)\} = \int_{\mathbb{R}^n} h(x)p(x)dx \leftarrow \frac{1}{N}\sum_{k=1}^{N} h(x_k)$$

Suppose $p(x)$ is difficult to simulate from - so use MCMC.
Soln. Choose symmetric $q(y|x) = q(x|y)$ e.g. multidim Gaussian

$$q(y|x) \propto \exp\big(-\frac{1}{2}(x-y)'\Sigma^{-1}(x-y)\big)$$

This is the random walk MH algorithm: $\alpha(x,y) = \min(\frac{p(y)}{p(x)}, 1)$.

Markov process $\{x_k\}$ generated by MH has stationary dist $p(x)$.

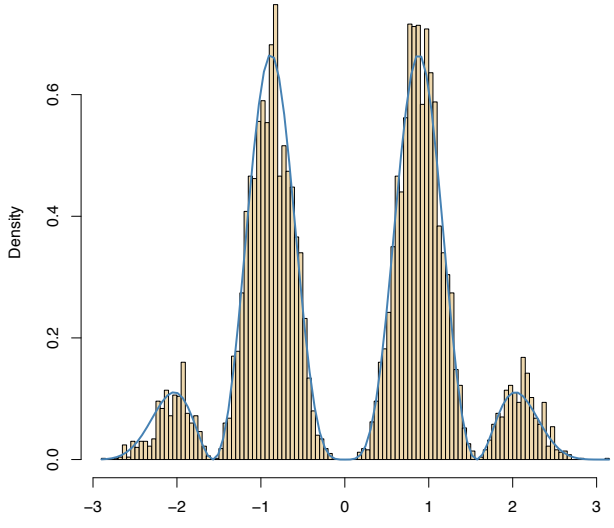
Example 2. Let $\psi(x) = N(0,1)$. Simulate from

$$\pi(x) \propto \sin^2(x) \times \sin^2(2x) \times \psi(x)$$

Proposal: $q(y|x) = U(x-\alpha, x+\alpha) = \frac{1}{2\alpha} I(x-\alpha \le y \le x+\alpha)$.
Clearly $q(y|x)$ is symmetric: $q(x|y) = q(y|x)$.

$$q(x|y) = \frac{1}{2\alpha} I(y-\alpha \le x \le y+\alpha) = \frac{1}{2\alpha} I(x-\alpha \le y \le x+\alpha)$$

So $\alpha_{xy} = \min(1, \pi(y)/\pi(x))$.



Metropolis Hastings for 10^4 iterations $\alpha = 1$ starting at $x_0 = 3.14$.

MCMC for Computational Bayesian

How to numerically implement Bayes rule?

$$p(x|y) = \frac{p(y|x)p(x)}{\int_{\mathbb{R}^X} p(y|\zeta)p(\zeta)d\zeta}$$

Denominator: high dimensional integral; difficult to evaluate.

Key idea: Given observation y, use Metropolis Hastings to simulate Markov chain x_k with target (stationary) distribution

$$\pi(x) \propto p(y|x)p(x), \quad k = 1, 2, \ldots$$

For convenience, use a random walk Metropolis Hastings:

1. Simulate x_k with symmetric transition $q(\bar{x}|x) = q(x|\bar{x})$
2. acceptance probability $\alpha_{x,\bar{x}} = \min\left(1, \frac{p(y|\bar{x})p(\bar{x})}{p(y|x)p(x)}\right)$

A cloud of samples $\{x_k\}$ is equivalent to knowing pdf or pmf.

We can then use MCMC simulated samples $\{x_k\}$ to:

1. estimate posterior cdf or pdf. Empirical pdf of posterior is

$$\hat{p}_n(x|y) = \frac{1}{n}\sum_{k=1}^{n} I(x_k \in [x - \Delta, x + \Delta))$$

   Matlab: Given $x = [x_1, \ldots, x_n]$, use `hist(x, nbins, 1)`
2. estimate conditional mean: by law of large numbers

$$\frac{1}{n}\sum_{k=1}^{n} x_k \to \mathbb{E}\{x|y\} = \int x p(x|y) dx$$

Remark. Marginalization: Given samples $\alpha_k, \beta_k \sim p(\alpha, \beta)$ then samples $\alpha_k \sim p(\alpha)$.

Example 1- Binomial with improper prior.

We observe number of wins S_n in n indpt trials.

Prior probability of winning in each trial is rv $X \sim p(X)$.

1. Estimate posterior distribution $p(X|S_n)$.
2. Estimate conditional mean $\mathbb{E}\{X|S_n\}$.

Model. Prior: $p(X) \propto 2\cos^2(4\pi X)$ (improper prior), $x \in [0,1]$.

$P(Y_i = 1 \text{ (win)}) = X$ for $i = 1, 2, \ldots, n$ (Bernoulli rv).

Observation: number of wins $S_n = \sum_{i=1}^n Y_i$.

So likelihood is: $p(S_n = s|X) = \binom{n}{s} X^s (1-X)^{n-s}$ (binomial)

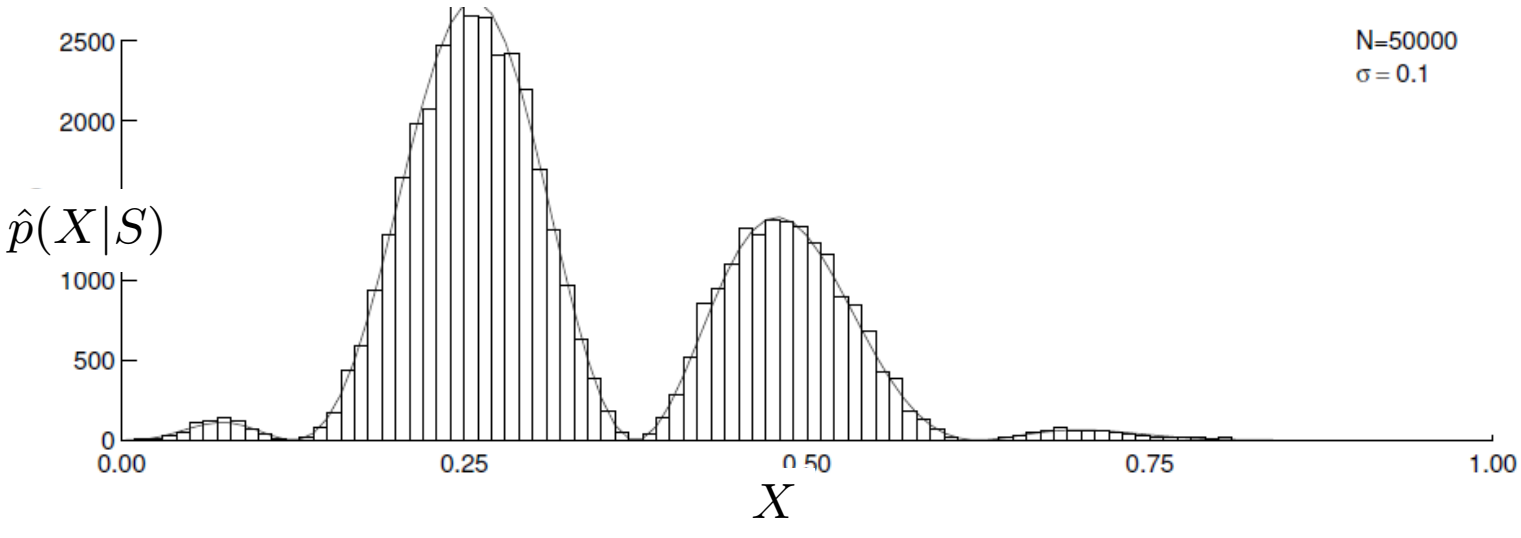
Procedure: Given S_n, use MH to simulate samples from posterior

$$p(X|S_n) \propto X^{S_n}(1-X)^{n-S_n} \cos^2(4\pi X)$$

Soln: Choose symmetric proposal in Metropolis Hastings:

$$\text{simulate } x_k \sim q(\bar{X}|X) \propto \exp\Big(-\frac{1}{2\sigma^2}(X-\bar{X})^2 \Big)$$

$$\alpha_{X,\bar{X}} = \min\{1, \frac{\bar{X}^{S_n}(1-\bar{X})^{n-S_n} \cos^2(4\pi\bar{X})}{X^{S_n}(1-X)^{n-S_n} \cos^2(4\pi X)}$$



Example 2 - Covariation for Hierarchical Bayes Model

Given two data streams of observations $x_{1:N}$ and $y_{1:N}$.

Aim: Compute posterior estimate of correlation ρ between data streams, i.e. compute $\mathbb{E}\{\rho|x_{1:N}, y_{1:N}\}$. Assume

$$x_i, y_i|\rho \sim N(\mu, \Sigma(\rho)), \quad \mu \in \mathbb{R}^2, \Sigma(\rho) = \begin{bmatrix} 1 & \rho \\ \rho & 1 \end{bmatrix}$$

for $i = 1, \ldots, N$. So likelihood

$$p(x_{1:N}, y_{1:N}|\rho) = \prod_{i=1}^{N} \frac{1}{2\pi\sqrt{1-\rho^2}} \exp\left(-\frac{1}{2(1-\rho^2)}[x_i^2 - 2\rho x_i y_i + y_i^2]\right)$$

Assume prior is $p(\rho) \propto (1-\rho^2)^{-3/2}$.

So from Bayes rule, the posterior

$$p(\rho|x_{1:N}, y_{1:N}) \propto \text{ prior } \times \text{ likelihood}$$

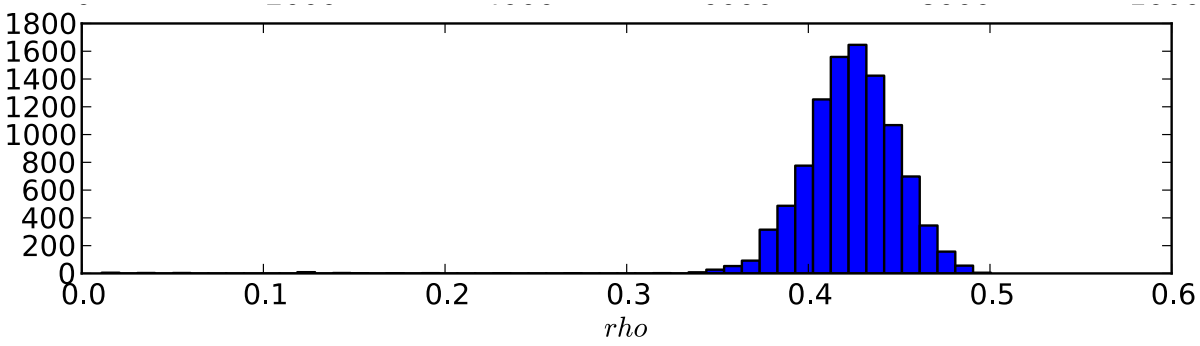
We use MH to sample from posterior $p(\rho|x_{1:N}, y_{1:N})$ and then estimate conditional mean $\mathbb{E}\{\rho|x_{1:N}, y_{1:N}\}$.

Choose a symmetric kernel $q(\rho_{k+1}|\rho_k) = U(\rho_k - 0.07, \rho_k + 0.07)$.

Then acceptance probability is

$$\alpha_{\rho_{k-1}, \rho_k} = \min\{1, \frac{p(\rho_k|x_{1:N}, y_{1:N})}{p(\rho_{k-1}|x_{1:N}, y_{1:N})}\}$$

Simulation for $N = 10000$ time points. Posterior distribution histogram based on posterior samples:



Estimate of $\mathbb{E}\{\rho|x_{1:10000}, y_{1:10000}\} = 0.42$, std dev $= 0.03$.

Example 3. Hierarchical Bayesian Models.

$$\Theta \sim p(\theta) \quad \text{(hyper-parameter)}$$
$$X|\Theta \sim p(x|\theta)$$
$$Y|X \sim p(y|x)$$

Aim: Given y, sample from posterior $p(x|y)$.

Step 1. Use Metropolis Hastings to generate samples $(\theta, x)_i$, $i = 1, 2, \ldots$ from

$$p(\theta, x|y) \propto p(y|x)\, p(x|\theta)\, p(\theta)$$

(i) Simulate (θ_k, x_k) with symmetric transition density $q(\bar{x}, \bar{\theta}|x, \theta) = q(x, \theta|\bar{x}, \bar{\theta})$

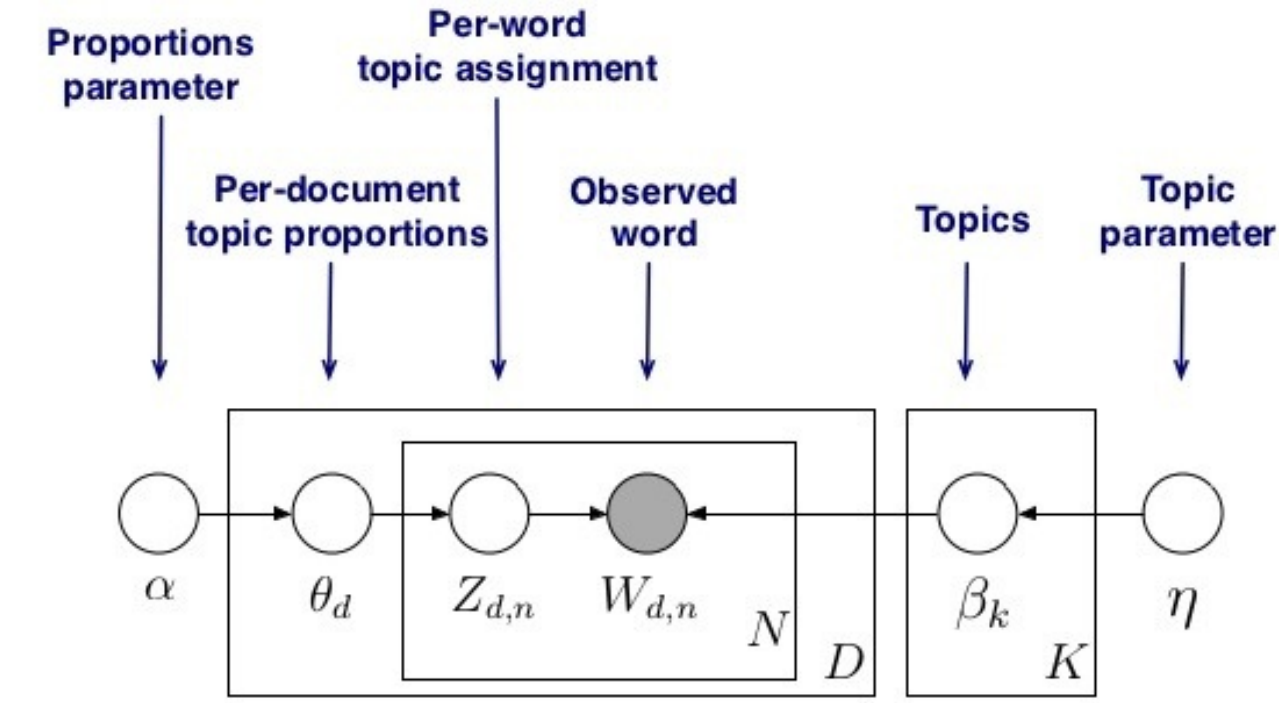
(ii) acceptance probability

$$\alpha_{x,\theta,\bar{x},\bar{\theta}} = \min\left(1, \frac{p(y|\bar{x})p(\bar{x}|\bar{\theta})p(\bar{\theta})}{p(y|x)p(x|\theta)p(\theta)}\right)$$

Step 2. Then samples x_i, $i = 1, \ldots$ are from posterior $p(x|y)$.

Example: **Latent Dirichlet Allocation** (LDA) for *topic modeling* in natural language processing: understand what document describes and retrieve documents that describe topic.

LDA as a graphical model



$$\prod_{i=1}^{K} p(\beta_i \mid \eta) \prod_{d=1}^{D} p(\theta_d \mid \alpha) \left(\prod_{n=1}^{N} p(z_{d,n} \mid \theta_d) p(w_{d,n} \mid \beta_{1:K}, z_{d,n}) \right)$$

See the following famous paper

`http://ai.stanford.edu/~ang/papers/jair03-lda.pdf`

Gibbs Sampling

Special case of the Metropolis-Hastings algorithm.
Used in 2 related contexts:
(i) To sample from multivariate $p(x_1, x_2, \ldots, x_L)$ when it is easy to sample from conditional univariate distributions,

$$p(x_1|x_2, \ldots, x_L), \quad p(x_2|x_1, x_3, \ldots, x_L), \ldots$$

(ii) (Data augmentation) Difficult to sample from $p(x)$ but there exists latent variable y such that $p(x, y)$ can be Gibbs sampled via $p(x|y)$ and $p(y|x)$ Then samples of x yields marginal $p(x)$.
Example: censored data models in survival analysis

Gibbs Sampling Algorithm from $\pi_\infty(x) = p(x_1, \ldots, x_L)$
Each iteration n has L stages.

- Given samples $x_1^{(n)}, x_2^{(n)}, \ldots, x_L^{(n)}$ from iteration n.
  At $(n+1)$th iteration generate
  1. $x_1^{(n+1)} \sim p(x_1|x_2^{(n)}, x_3^{(n)}, \ldots, x_L^{(n)})$
  2. $x_2^{(n+1)} \sim p(x_2|x_1^{(n)}, x_3^{(n)}, \ldots, x_L^{(n)})$
  $\vdots$
  L. $x_L^{(n+1)} \sim p(x_L|x_1^{(n)}, x_2^{(n)}, \ldots, x_{L-1}^{(n)})$

Remarks:

1. The sequence $(x_1^{(n)}, x_2^{(n)}, \ldots, x_L^{(n)})$, $n = 1, 2, \ldots$ has empirical pdf that converges to $\pi_\infty(x) = p(x_1, \ldots, x_L)$.
2. Gibbs sampling is a special case of Metropolis Hastings.
3. Order of updating the L coordinates can be chosen randomly.

Gibbs sampling = composition of L Metropolis-Hastings algorithms each with acceptance probability $\alpha = 1$.
Denote $x = (x_1, x_2, x_3)$, $y = (x_1, \bar{x}_2, x_3)$, $\pi_\infty(x) = p(x)$, $\pi_\infty(y) = p(y)$.

$$Q_{xy} = \frac{1}{3}p(y|x) = \frac{1}{3}p(\bar{x}_2|x_1, x_3) = \frac{1}{3}\frac{p(x_1, \bar{x}_2, x_3)}{P(x_1, x_3)} = \frac{1}{3}\frac{p(y)}{p(x_1, x_3)}$$

$$Q_{yx} = \frac{1}{3}\frac{p(x)}{p(x_1, x_3)}$$

So acceptance probabilities in Metropolis-Hastings algorithm are

$$\frac{\pi_\infty(y)Q_{yx}}{\pi_\infty(x)Q_{xy}} = \frac{p(y)Q_{yx}}{p(x)Q_{xy}} = 1 \quad \text{i.e. no rejections}$$

Example 1: Bivariate Normal. see movie at
`http://gorayni.blogspot.com/2013/08/gibbs-sampling.html`

Aim: Simulate $(x, y) \sim \boldsymbol{N}(0, \Sigma)$, $\Sigma = \begin{bmatrix} 1 & \rho \\ \rho & 1 \end{bmatrix}$, $|\rho| < 1$.

Verify using Swiss-army knife formula that

$$p(x|y) = \boldsymbol{N}(\rho y, 1 - \rho^2), \qquad p(y|x) = \boldsymbol{N}(\rho x, 1 - \rho^2)$$

Gibbs Sampler Simulate for $n = 1, 2, \ldots$

1. $x^{(n+1)} \sim p(x|y^{(n)})$,
2. $y^{(n+1)} \sim p(y|x^{(n+1)})$

This converges to stationary distribution $\boldsymbol{N}(0, \Sigma)$.

Example 2. Bayesian Inference for Censored Data Models

Motivation. (i) Survival analysis: How to determine effect of drug on average life-span; in a fixed-time study? People who survive fixed-time study length yield no info on life span.
Let y_k denote date of death.

Model.
$$z_k = \begin{cases} y_k & \text{if } y_k \le c \\ * & \text{otherwise} \end{cases}, \quad y_k \sim p(y_k|x), \quad x \sim p(x)$$

Assume observations are $z_{1:m} = y_{1:m}$ and $z_{m+1:n} = *$.
$*$ is a *right censored* observation (data right of c is censored)
Aim: How to sample from posterior $p(x|z_1, \ldots, z_n)$? Evaluating posterior in closed form is impossible in general.

Motivation. (ii) Quality Control: Test a product over c years to estimate time to failure. If product still functions after c years, then censored observation.

Trick. Use data augmentation.

1. Augment posterior with fictitious observations $y_{m+1:n}$; then use Gibbs sampling on $p(x, y_{m+1:n}|z_{1:n})$.
2. Then marginal $p(x|z_{1:n})$ is obtained from the empirical pdf of simulated samples $x^{(i)}, i = 1, 2, \ldots$.

Details. The conditional pdfs for Gibbs sampling are:

$$p(x|y_{1:n}, z_{1:n}) = p(x|y_{1:n}) \propto \prod_{k=1}^{n} p(y_k|x)$$

$$p(y_{m+1}|x, z_{1:n}, y_{m+2:n}) \propto p(y_{m+1}|x) I(y_{m+1} \geq c)$$

$$p(y_n|x, z_{1:n}, y_{m+1:n-1}) \propto p(y_n|x) I(y_n \geq c)$$

Use the above pdfs to Gibbs sample $x^{(i)}$, $y^{(i)}_{m+1:n}, i = 1, 2, \ldots$ from augmented posterior $p(x, y_{m+1:n}|z_{1:n})$.
Then marginal $p(x|z_{1:n})$ is obtained from the empirical pdf of simulated samples $x^{(i)}, i = 1, 2, \ldots$.

Example. Survival Analysis. Suppose number of years to live for each individual is Poisson with rate x: So

$$\text{Prob(live } i \text{ years } |x) = p(i|x) = \frac{e^{-x} x^i}{i!}$$

A study was conducted for 3 years.
y_i: number of individuals who live for i years after given drug.
$z_i = \min\{y_i, 3\}$: right censored version of y_i.

The following is right censored data for 360 individuals in 3 year trial.

year i	0	1	2	3	≥ 4
# people z_i	139	128	55	25	13

Aim. Given above data for 360 individuals in a 3 year trial, estimate survival rate x.
Estimate posterior $p(x|z_{1:4})$ where x has prior $p(x) \propto 1/x$.

Posterior $p(x|z_{1:4}) \propto p(z_{1:4}|x)p(x)$ where $p(x) \propto 1/x$.

Step 1. Let us work out likelihood $p(z_{1:4}|x)$:

$$p(z_i|x) = \Big(\frac{e^{-x}x^i}{i!}\Big)^{z_i} \propto e^{-xz_i}x^{iz_i}, i = 0,1,2,3$$

$$p(i \geq 4|x) = 1 - \sum_{i=0}^{3} p(i|x) = 1 - \sum_{i=0}^{3} \frac{e^{-x}x^i}{i!}$$

$$= \text{ prob individual lives for } \geq 4 \text{ years}$$

$$p(z_4|x) = (1 - \sum_{i=0}^{3} \frac{e^{-x}x^i}{i!})^{z_4}$$

$$\Longrightarrow p(z_{1:4}|x) \propto \prod_{i=0}^{3} e^{-xz_i}x^{iz_i} \, (1 - \sum_{i=0}^{3} \frac{e^{-x}x^i}{i!})^{z_4}$$

Step 2. Gibbs sampling of posterior $p(x|z_{1:4})$ is as follows:
Consider augmented posterior $p(x, y_{1:13}|z_{1:4})$ where $y_{1:13}$ are fictitious lifetimes for $i \geq 4$. Then

$$p(x|y_{1:13}, z) \propto p(y, z|x) = \prod_{i=1}^{13} e^{-x}x^{y_i} \prod_{i=0}^{3} e^{-xz_i}x^{(i-1)z_i}$$

$$p(y_i|x, z) \propto e^{-\lambda}\frac{\lambda^{y_i}}{y_i!} I(y_i \geq 4), \quad i = 1\ldots, 13$$

Summary: Above two pdfs are used to Gibbs sample from augmented posterior $p(x, y_{1:13}|z_{1:4})$.
Then empirical pdf of sampled $x^{(i)}, i = 1, 2, \ldots$ converges to pdf $p(x|z_{1:4})$. Can compute $\mathbb{E}\{x|z_{1:4}\}$, conditional variance, etc.

Outline

- **Part I. Things to know**
- **Part II. Least Squares**
- **Part III. Bayesian Inference and MCMC**
  1. Stochastic State Space Models & Optimal Predictors
  2. Bayesian Filtering and Smoothing
- **Part IV. Maximum Likelihood and EM**

- **Part V. Stochastic Optimization**

III.2. Optimal Bayesian 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, machine learning, robotics

- Stochastic State Space Models & Optimal Predictors (2)
- Kalman Filter, HMM filter
- Briefly describe sequential MCMC based particle filters.
- Describe how these filters can be used for ML parameter estimation (if time permits)

Stochastic Difference Equation to Transition Density

$$\begin{aligned} x_{k+1} &= A_k(x_k) + \Gamma_k(x_k) w_k, \quad x_0 \sim \pi_0 \\ y_k &= C_k(x_k) + D_k(x_k) v_k. \end{aligned} \tag{8}$$

Assume $\Gamma_k(x_k)$ and $D_k(x_k)$ are square invertible matrices.
Then transition density and observation likelihood are:

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

where $|\cdot|$ denotes determinant.

Example: Moving Autonomous Vehicle Linear Gaussian state space model

$$x_{k+1} = Ax_k + Bu_k + v_k$$

$x_k \stackrel{\text{defn}}{=} [r_x[k], \dot{r}_x[k], r_y[k], \dot{r}_y[k]]'$: state vector at time kT
T is the sampling interval.
$r_x(k)$ is x-coordinate position, $\dot{r}_x(k)$ is velocity in x direction.

$$A = \begin{pmatrix} 1 & T & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & T \\ 0 & 0 & 0 & 1 \end{pmatrix}, \qquad B = \begin{pmatrix} T^2/2 & 0 \\ T & 0 \\ 0 & T^2/2 \\ 0 & T \end{pmatrix}$$

$u_k = [u_k(1), u_k(2)]' \in \mathbb{R}^2$ models acceleration (maneuver) of the target. u_k changes infrequently (e.g. when target turns).

v_k iid noise process – models uncertainly in target dynamics

Model is obtained by discretizing continuous-time model:

$$\frac{dx_1}{dt} = x_2 \text{ (velocity)}, \quad \frac{dx_2}{dt} = u_1 \text{ (acceleration)}$$

Discretizing with sampling interval T yields:

$$x_{k+1}(1) = x_k(1) + \frac{T}{2}(x_{k+1}(2) + x_k(2))$$
$$x_{k+1}(2) = x_k(2) + Tu_k(1)$$

So first eqn becomes: $x_{k+1}(1) = x_k(1) + Tx_k(2) + \frac{T^2}{2}u_k(1)$.

Measurement. E.g. using radars. In the simplest setting,

$$y_k = Cx_k + v_k, \quad v_k \sim \boldsymbol{N}(0, R).$$

where

$$x_k = [r_x[k], \dot{r}_x[k], r_y[k], \dot{r}_y[k]]'$$

Example 1. If only position is observed, then $y_k \in \mathbb{R}^2$,

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

Example 2. If the position and velocity observed (Doppler radar), $C = I_{4\times 4}$.

Optimal Prediction: Chapman Kolmogorov Equation

Aim: How to optimally predict future state given current state probability?
Given Markov process with transition density $p(x_{k+1}|x_k)$ and initial condition π_0, compute state pdf $\pi_k(x) = p(x_k = x)$ at time k.

We call π_k the *predicted* density.

Chapman Kolmogorov: From total probability rule

$$\boxed{\pi_k(x) = \int_{\mathcal{X}} p(x_k = x|x_{k-1})\, \pi_{k-1}(x_{k-1}) dx_{k-1}, \quad \text{initialized by } \pi_0.}$$

Therefore predicted state and covariance at time k are

$$\boxed{\begin{aligned} \hat{x}_k = \mathbb{E}\{x_k\} &= \int_{\mathcal{X}} x\, \pi_k(x) dx, \\ \text{cov}(x_k) = \mathbb{E}\{(x_k - \hat{x}_k)(x_k - \hat{x}_k)'\} &= \mathbb{E}\{x_k x_k'\} - \hat{x}_k \hat{x}_k'. \end{aligned}}$$

Predicted mean is $\hat{x}_k$ is optimal in the minimum mean square error sense:

$$\mathbb{E}\{(x_k - \hat{x}_k)^2\} \leq \mathbb{E}\{(x_k - \phi(\pi_0))^2\}.$$

Hence called "optimal predictor".

Ex 1. Linear Gaussian State Space Model

Notation:

$$\boldsymbol{N}(\zeta;\mu,\Sigma) = (2\,\pi)^{-l/2}\,|\Sigma|^{-1/2}\exp\left[-\frac{1}{2}(\zeta-\mu)'\Sigma^{-1}(\zeta-\mu)\right].$$

Sometimes shorter notation $\boldsymbol{N}(\mu,\Sigma)$ will be used.

The linear Gaussian state space model

$$\boxed{\begin{aligned} x_{k+1} &= A_k x_k + w_k, \quad x_0 \sim \pi_0 = \boldsymbol{N}(\hat{x}_0,\Sigma_0),\ w_k \sim \boldsymbol{N}(0,Q_k) \\ y_k &= C_k x_k + v_k, \quad v_k \sim \boldsymbol{N}(0,R_k). \end{aligned}}$$

$x_k \in \mathcal{X} = \mathbb{R}^X$, $y_k \in \mathcal{Y} = \mathbb{R}^Y$, $A_k \in \mathbb{R}^{X\times X}$, $C_k \in \mathbb{R}^{Y\times X}$.

Transition density form:

$$p(x_{k+1}|x_k) = p_w\left(x_{k+1} - A_k(x_k)\right) = \boldsymbol{N}(x_{k+1}; A_k x_k, Q_k)$$
$$p(y_k|x_k) = p_v\left(y_k - C_k(x_k)\right) = \boldsymbol{N}(y_k; C_k x_k, R_k).$$

Optimal Predictor Using the Chapman Kolmogorov equation $\pi_{k+1} = \boldsymbol{N}(\hat{x}_{k+1},\Sigma_{k+1})$ where

$$\boxed{\begin{aligned} \hat{x}_{k+1} &= \mathbb{E}\{x_{k+1}\} = A_k\hat{x}_k \\ \Sigma_{k+1} &= \mathrm{cov}\{x_{k+1}\} = A_k\Sigma_k A_k' + Q_k. \end{aligned}}$$

Covariance update is called *Lyapunov equation.*
Same mean and covariance recursions also hold for non-gaussian case

Ex 2. Hidden Markov Model

Markov chain measured via a noisy observation process.

(i) *Markov chain*: $\{x_k\} \in \mathcal{X} = \{1, 2, \ldots, X\}$.
Transition probability matrix P with elements

$$P_{ij} = \mathbb{P}(x_{k+1} = j | x_k = i), \quad 0 \le P_{ij} \le 1, \quad \sum_{j=1}^{X} P_{ij} = 1 \equiv P\mathbf{1} = \mathbf{1}$$

Initial: $\mathbb{P}(x_0 = i) = \pi_0(i)$, $i = 1, \ldots, X$ with $\sum_{i=1}^{X} \pi_0(i) = 1$.
State level vector (physical states) $C = [C(1), C(2), \ldots, C(X)]'$.

(ii) *Noisy Observations*: Define *observation probability* density

$$y_k \sim B_{xy} = p(y_k = y | x_k = x), \quad x \in \mathcal{X}, y \in \mathcal{Y}.$$

Note $\sum_y B_{xy} = 1$ or $\int B_{xy} dy = 1$

Example 1: Additive noise HMM. $\mathcal{Y} = \mathbb{R}$,

$$y_k = C(x_k) + D(x_k)\, v_k, \quad v_k \sim p_v$$

Then using likelihood formula

$$B_{xy} = p(y_k = y | x_k = x) = \frac{1}{D(x)}\, p_v\Big(\frac{y - C(x)}{D(x)}\Big)$$

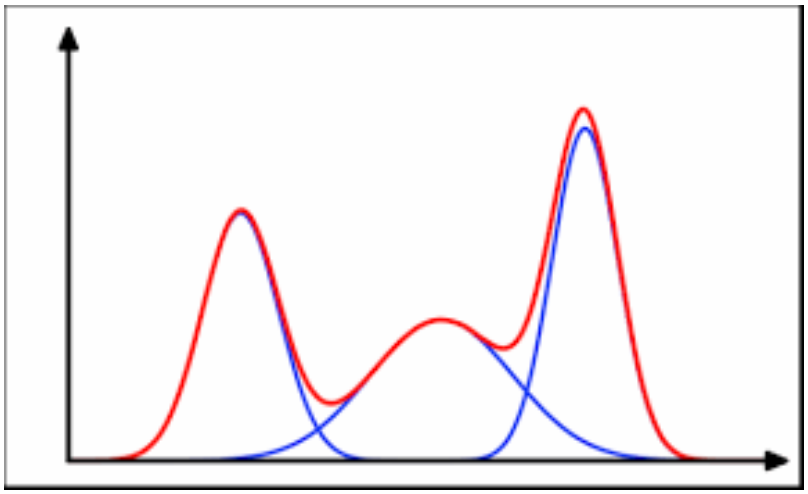
Example 2: Finite Observation Space HMM. $\mathcal{Y} = \{1, \ldots, Y\}$.
Then $B_{xy} = \mathbb{P}(y_k = y | x_k = x)$ are called "symbol" probabilities.

$$B_{X \times Y} = \begin{bmatrix} 0.8 & 0.2 & 0 \\ 0.1 & 0.3 & 0.6 \end{bmatrix}$$

HMM= *dynamic mixture model.*

If $P = \text{iid}$, then HMM specializes to classical mixture model.

E.g. if $P = \text{iid}$, B_{xy} Gaussian, then Gaussian mixture model.



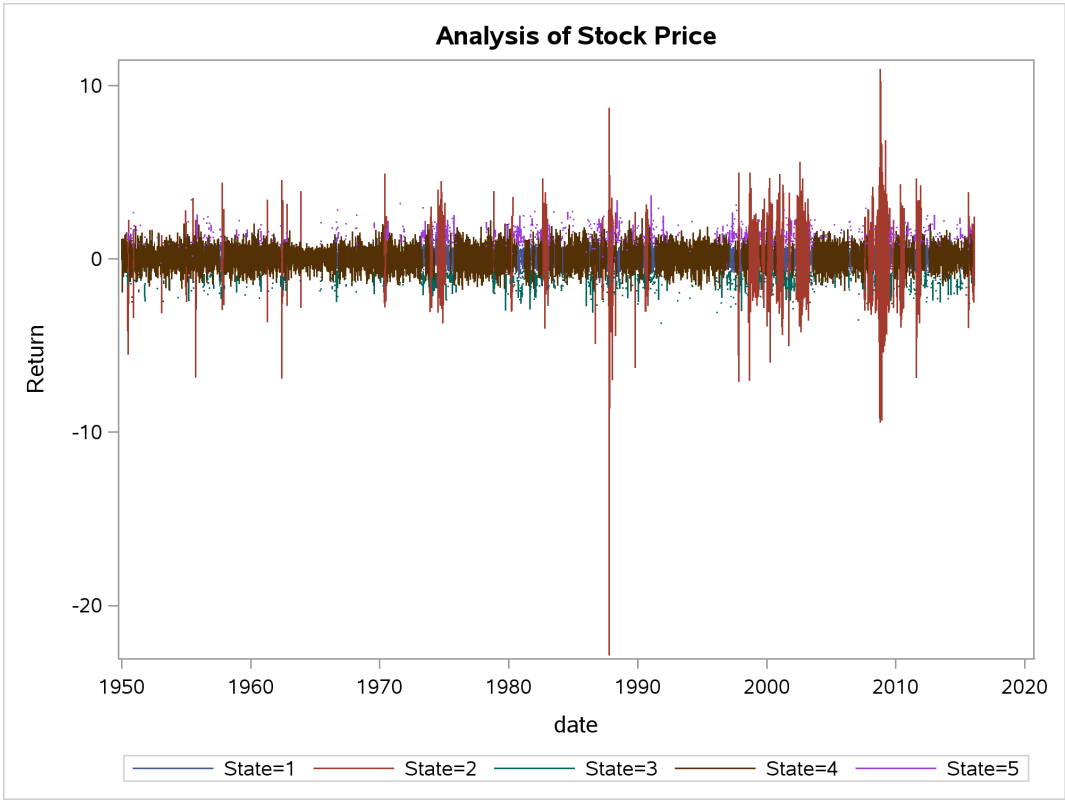
$X = 3$ and

$$p(y) = \sum_{i=1}^{X} \pi(i) B_{iy}$$

where $B_{iy} = p(y|x = i)$ is Gaussian.

HMM (dynamic mixture): $p(y) = \sum_{i=1}^{X} \pi_k(i) B_{iy}$.

HMMs and mixture models used in: Topic analysis in documents (LDA), speech recognition, neurobiology, bioinformatics, financial models, econometrics, wireless comms, target tracking, handwriting recognition, image segmentation



Optimal Predictor Define state probability vector at time k

$$\pi_k = \begin{bmatrix} \mathbb{P}(x_k = 1) & \dots & \mathbb{P}(x_k = X) \end{bmatrix}'. \tag{9}$$

Then the Chapman-Kolmogorov equation reads

$$\pi_{k+1} = P'\pi_k \quad \text{initialized by } \pi_0. \tag{10}$$

So (10) is the optimal predictor for an HMM. Then

$$\mathbb{E}\{C(x_{k+1})\} = \sum_{i=1}^{X} C(i)\pi_{k+1}(i) = C'\pi_{k+1}.$$

Limiting Distribution *Limiting distribution* is

$$\lim_{k\to\infty} \pi_k = \lim_{k\to\infty} {P'}^k \pi_0.$$

This limiting distribution may not exist. For example if

$$P = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \pi_0 = \begin{bmatrix} \pi_0(1) \\ \pi_0(2) \end{bmatrix}, \quad \text{then } \pi_k = \begin{cases} \begin{bmatrix} \pi_0(2) & \pi_0(1) \end{bmatrix}' & k \text{ odd} \\ \begin{bmatrix} \pi_0(1) & \pi_0(2) \end{bmatrix}' & k \text{ even} \end{cases}$$

and so $\lim_{k\to\infty} \pi_k$ does not exist unless $\pi_0(1) = \pi_0(2) = 1/2$.

Stationary Distribution X-dimensional vector π_∞

$$\pi_\infty = P'\pi_\infty, \quad \mathbf{1}'\pi_\infty = 1$$

So π_∞ is normalized right eigenvector of P' corresponding to the unit eigenvalue. Equivalently, choosing $\pi_0 = \pi_\infty$ implies $\pi_k = \pi_\infty$ for all k. The stationary distribution is also called the *invariant, equilibrium or steady-state distribution.*

Limiting distributions are a subset of stationary distributions. For example, given P in (181), $\pi_\infty = \begin{bmatrix}0.5 & 0.5\end{bmatrix}'$ is a stationary distribution but there is no limiting distribution.

Theorem 6 (Perron-Frobenius). *Consider a finite-state Markov chain with regular transition matrix P. Then:*

1. *The eigenvalue 1 has algebraic & geometric multiplicity of one.*
2. *All remaining eigenvalues of P have modulus strictly smaller than 1.*
3. *The eigenvector of P' corresponding to eigenvalue of 1 can be chosen with non-negative elements.*
4. *$P^k = \mathbf{1}\pi'_\infty + O(|\lambda_2|^k)$ where λ_2 is the second largest eigenvalue modulus.*
5. *The limiting distribution and stationary distribution coincide.*

Statement 4 says if the transition matrix P is regular, state probability vector π_k forgets initial condition geometrically fast.

$$\pi_k = {P'}^k \pi_0 = \pi_\infty \mathbf{1}'\pi_0 + O(|\lambda_2|^k)\pi_0 = \pi_\infty + O(|\lambda_2|^k)\pi_0.$$

So k-step ahead predictor of a Markov chain forgets initial condition geometrically fast in terms of the second largest eigenvalue modulus, $|\lambda_2|$.
Equivalently, π_k converges geometrically fast to π_∞.

Ex 3. Jump Markov Linear Systems

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

where r_k is finite state Markov chain, u_k is known input.

1. JMLS (Switched Markov Linear System) widely used stochastic state space models to model maneuvering moving objects, deconvolution, finance.
2. Models hybrid systems: continuous state real life system interaction with finite state automata.
3. If r_k is one state, then classical linear system.
   If $A = 1$, $\Gamma = f = 0$, then HMM.

Augmented state process $x_k = (z_k, r_k)$ is a Markov process in state space $\mathcal{X} = \mathbb{R}^z \times \{1, \ldots, X\}$.

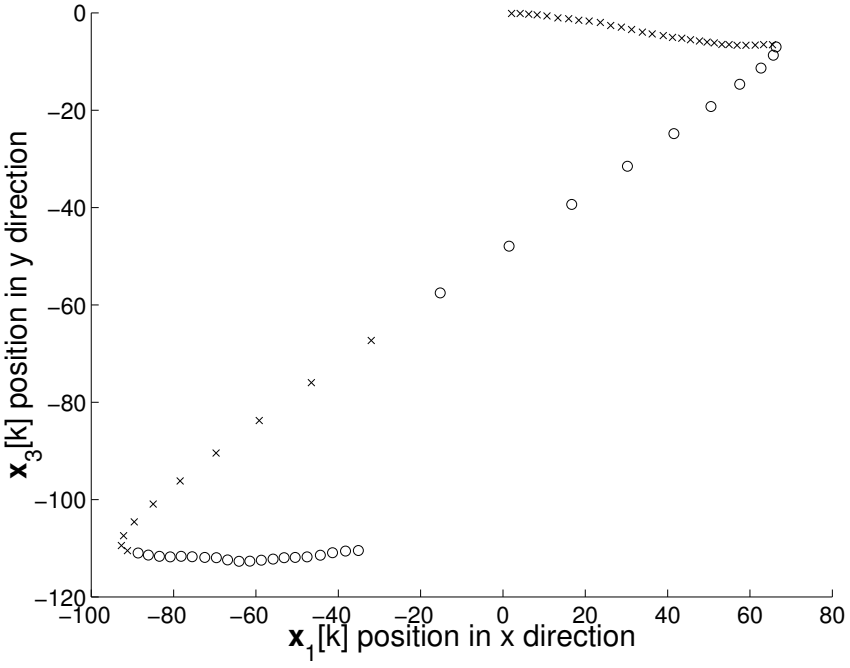
The transition density of the JMLS state x_k is

$$\begin{aligned} & p\left(x_{k+1} = (\bar{z}, j) | x_k = (z, i)\right) \\ &= p(r_{k+1} = j | r_k = i) p\left(z_{k+1} = \bar{z} | r_{k+1} = j, z_k = z\right) \\ &= P_{ij}\, |\Gamma^{-1}(j)|\, p_w\left(\Gamma^{-1}(j)\left[\bar{z} - A(j)z - f(j)u_{k+1}\right]\right). \end{aligned}$$

The observation likelihood is evaluated as

$$p(y_k | x_k = (z, i)) = |D^{-1}(i)|\, p_v\left(D^{-1}(i)\left[y_k - C(i)z - g(i)u_k\right]\right).$$

Example: Modeling Maneuvering Target



Consider a target moving in $\mathbb{R}^2$. $\Delta > 0$: sampling interval.

In time interval $[k\Delta, (k+1)\Delta)$, target undergoes acceleration $q_k^{(1)}$ and $q_k^{(2)}$ in x and y coordinates.

$p_t^{(1)}$ and $\dot{p}_t^{(1)}$: position and velocity in x-direction,

$p_t^{(2)}$ and $\dot{p}_t^{(2)}$: position and velocity in y-direction.

Then

$$\frac{d}{dt}\dot{p}_t^{(1)} = q_k^{(1)}, \quad \frac{d}{dt}\dot{p}_t^{(2)} = q_k^{(2)}, \quad t \in [k\Delta, (k+1)\Delta)$$

As a result, in discrete time for $i = 1, 2$,

$$\dot{p}_{k+1}^{(i)} = \dot{p}_k^{(i)} + \Delta\, q_k^{(i)}, \quad p_{k+1}^{(i)} = p_k^{(i)} + \Delta \dot{p}_k^{(i)} + \frac{\Delta^2}{2}\, q_k^{(i)}, \quad k = 1, 2, \ldots$$

Define target state $z_k = (p_k^{(1)}, \dot{p}_k^{(1)}, p_k^{(2)}, \dot{p}_k^{(2)})'$. Then

$$z_{k+1} = A\, z_k + f\, r_{k+1} + w_k, \quad k = 0, 1, \ldots, N$$

$$A = \begin{bmatrix} 1 & \Delta & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & \Delta \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad f = \begin{bmatrix} \Delta^2/2 & 0 \\ \Delta & 0 \\ 0 & \Delta^2/2 \\ 0 & \Delta \end{bmatrix}, \quad r_{k+1} = \begin{bmatrix} q_k^{(1)} \\ q_k^{(2)} \end{bmatrix}.$$

Here r_{k+1} denotes the constant acceleration (maneuvers) during the k-th sampling period.

The maneuver process $\{r_k\}$ is modeled as a finite-state Markov chain. Target motion is subject to small random perturbations $w_k \sim \boldsymbol{N}(0, Q)$ occurring every Δ seconds.

Summary. Target moves as a jump Markov linear system where the 4-dimensional state $\{z_k\}$ evolves as

$$z_{k+1} = Az_k + fr_{k+1} + w_k, \quad k = 0, 1, \ldots. \tag{11}$$

Here $\{r_k\}$ is a finite-state Markov chain, and $w_k \sim \boldsymbol{N}(0, Q)$.

Measurement. In the simplest setting,

$$y_k = Cz_k + v_k, \quad v_k \sim \boldsymbol{N}(0, R).$$

E.g.: If only position is observed, then $y_k \in \mathbb{R}^2$, and C is a 2×4 matrix with $C_{1,1} = C_{2,3} = 1$ and the remaining elements are zero. If the position and velocity are observed, then $C = I$.

```
% matlab program to simulate maneuvering  target
 Delta=1
 A=[1 Delta 0 0;0 1 0 0;0 0 1 Delta; 0 0 0 1];
 f=[Delta*Delta/2 0; Delta 0;0 Delta^2/2; 0 Delta]  ;
x(:,1)=[0 2 0 0]'     ;

for k=2:30,
  x(:,k) = A* x(:,k-1) + f * [0 0]' + .1*randn(4,1);
  plot(x(1,k),x(3,k),'x');
end;

for k=31:40,
  x(:,k) = A* x(:,k-1) + f * [-2 -1]' + .1*randn(4,1);
  plot(x(1,k),x(3,k),'o');
end;

for k=41:50,
  x(:,k) = A* x(:,k-1) + f * [2 1]' + .1*randn(4,1);
  plot(x(1,k),x(3,k),'x');
end;

for k=51:70,
  x(:,k) = A* x(:,k-1) + f * [0 0]' + .1*randn(4,1);
  plot(x(1,k),x(3,k),'o');
end;
```

Outline

- **Part I. Bayesian Inference and MCMC** (6 classes)
  1. Things to know (1)
  2. Simulation-1 (1)
  3. Bayes Inference of Random Variables (1)
  4. Simulation-2: MCMC (3)
- **Part II. Bayesian Filtering and Graphical Models** (12 classes)
  1. Stochastic State Space Models & Optimal Predictors (2)
  2. Bayesian Filtering and Smoothing (4)
  3. Graphical Models and Social Learning (2)
- **Part III Maximum Likelihood and EM** (4 hours)
  MLE for Hidden Markov Models
- **Part IV. Bayesian Decision Making** (9 classes)
  1. Markov Decision Processes
     (a) Stochastic Dynamic Programming
  2. Partially Observed Markov Decision Processes
     (a) POMDP Model and applications
     (b) Stochastic Dynamic Programming
     (c) Bayesian Stopping Time Problems

Bayesian Filtering

Wiener Filter: Nobert Wiener 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.

In continuous-time stochastic filtering theory involves stochastic calculus – widely used in mathematical finance. Not covered here.

Perspective

Given a partially observed stochastic dynamical system

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

$\{v_k\}$ and $\{w_k\}$ are iid. In transition density form

$$\begin{aligned} p(x_{k+1}|x_k) &\propto p_w \left(\Gamma_k^{-1}(x_k) \left[x_{k+1} - A_k(x_k) \right] \right) \\ p(y_k|x_k) &\propto p_v \left(D_k^{-1}(x_k) \left[y_k - C_k(x_k) \right] \right). \end{aligned}$$

Assume known model.
Aim: Compute state estimate $\hat{x}_k = \mathbb{E}\{x_k | y_1, \ldots, y_k\}$.

State estimation has two broad philosophies

- **Bayesian State Estimation:** Model based **optimal filtering** such as Kalman Filters, Hidden Markov Model filters, particle filters. (This part)
- **Adaptive filtering:** e.g. LMS, RLS. x_k is assumed to vary slowly with unknown dynamics.

Given a stochastic dynamical system

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

Assume model and parameters are known.

Aim: Compute the min-variance state estimate $\hat{x}_{l|k}$ given the sequence of observations $Y_k = y_1, \ldots, y_k$.
As shown $\hat{x}_{l|k} = \mathbb{E}\{x_l|Y_k\}$; and is unbiased estimator.

There are 3 problems of interest:

- **Filtering**. $k = l$ (real time): compute $p(x_k|Y_k)$.
- **Prediction**. $l > k$ (real time): compute $p(x_{k+\Delta}|Y_k)$.
- **Smoothing**: $l < k$. (off-line): compute $p(x_{k-\Delta}|Y_k)$.

We focus here on filtering.

Predictor = filter with non-information observations:

$p(y|x)$ indpt of x.

Smoother: forward and backward filter discussed later

Filtering Recursion

Model : $p(x_{k+1}|x_k) \propto p_w\left(\Gamma_k^{-1}(x_k)\,[x_{k+1} - A_k(x_k)]\right)$,

$p(y_k|x_k) \propto p_v\left(D_k^{-1}(x_k)\,[y_k - C_k(x_k)]\right)$, $\pi_0(x) = p(x_0 = x)$.

Aim: For $k = 1, 2\ldots$, recursively compute filtered density

$$\pi_k(x) = p(x_k = x|y_{1:k})$$

Then conditional mean (optimal) state estimate is

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

Main Result: Starting with $\pi_0(x)$, for $k = 0, 1, \ldots$

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

$$\hat{x}_{k+1} = \mathbb{E}\{x_k|y_{1:k}\} = \int_{\mathcal{X}} x\, \pi_{k+1}(x)dx$$

Remarks:

1. With only 2 exceptions (Kalman & HMM filter) posteriors $\{\pi_k\}$ are not finite dimensional computable.
2. Predictor is special case of filter with $p(y|x)$ indpt of x:

$$\pi_{k+1}(x_{k+1}) = \int_{\mathcal{X}} p(x_{k+1}|x_k)\, \pi_k(x_k)\, dx_k \qquad \text{(CK eqn)}$$

Prediction & Measurement Form of Optimal Filter

Starting with $\pi_0(x)$, for $k = 0, 1, \ldots$

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

1. First step is Chapman Kolomogorov eqn for predictor
2. Second step is Bayesian update

Un-normalized Filter Update

Recall $\pi_k(x) = p(x_k = x|y_{1:k})$ is conditional density
It is often convenient to compute un-normalized density:

$$q_k(x) = p(x_k = x, y_{1:k}).$$

$$\text{Clearly } \pi_k(x) = \frac{q_k(x)}{\int_{\mathcal{X}} q_k(x)dx}$$

Filtering recursion for un-normalized density is:
Start with $q_0(x) = \pi_0(x)$. Then for $k = 0, 1, \ldots$.

$$\boxed{q_{k+1}(x) = p(y_{k+1}|x_{k+1} = x) \int_{\mathcal{X}} p(x_{k+1} = x|x_k)q_k(x_k)dx_k.}$$

$$\hat{x}_{k+1} = \mathbb{E}\{x_{k+1}|y_{1:k+1}\} = \frac{\int_{\mathcal{X}} x\, q_{k+1}(x)dx}{\int_{\mathcal{X}} q_{k+1}(x)dx}.$$

Example. Toy Kalman Filter

Consider scalar linear Gaussian model

$$\begin{aligned} x_{k+1} &= x_k + w_k, \quad w_k \sim N(0,1); \quad \pi_0(x) \sim \boldsymbol{N}(\hat{x}_0, \Sigma_0). \\ y_k &= x_k + v_k, \quad v_k \sim N(0,1) \end{aligned}$$

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

$$p(x_{k+1}|x_k) = \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{1}{2}(x_{k+1} - x_k)^2\right), \ \pi_0(x) \sim N(\hat{x}_0, \Sigma_0).$$

Filtering recursion for $q_k(x) = p(x, y_{1:k})$ is: $q_0(x) \sim \boldsymbol{N}(\hat{x}_0, \Sigma_0)$

$$\begin{aligned} q_{k+1}(x_{k+1}) &= \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{1}{2}(y_{k+1} - x_{k+1})^2\right) \\ &\quad \times \int_{\mathbb{R}} \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{1}{2}(x_{k+1} - x_k)^2\right) q_k(x_k) dx_k \\ \pi_{k+1}(x_{k+1}) &= \frac{q_{k+1}(x_{k+1})}{\int_{\mathbb{R}} q_{k+1}(z) dz} \end{aligned}$$

How to solve for q_{k+1}? Suppose $q_k \sim \boldsymbol{N}(\hat{x}_k, \Sigma_k)$.

1. Since $q_k(x)$ is Gaussian, and convolution of Gaussians is Gaussian, the integral is Gaussian.
2. The integral is a Gaussian prior, the likelihood is Gaussian; since Gaussians are conjugate priors, $q_{k+1}(x)$ is Gaussian.

Key point. $q_k(x) \sim \boldsymbol{N}(\hat{x}_k, \Sigma_k) \implies q_{k+1}(x) \sim \boldsymbol{N}(\hat{x}_{k+1}, \Sigma_{k+1})$

Kalman filter: $\hat{x}_{k+1} = \phi_1(\hat{x}_k, \Sigma_k)$, $\Sigma_{k+1} = \phi_2(\Sigma_k)$.

Optimal Filter 1. Kalman Filter

Model: Linear Gaussian State Space

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

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

Kalman filter equations: (for conditional mean and covariance)

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

Note that the posteriors are Gaussian:

$$p(x_k|y_{1:k}) = \boldsymbol{N}(\hat{x}_k, \Sigma_k),$$
where $\hat{x}_k = \mathbb{E}\{x_k|y_{1:k}\}, \quad \Sigma_k = \mathbb{E}\{(\hat{x}_k - x_k)(\hat{x}_k - x_k)'\}$
Also, $p(x_{k+1}|y_{1:k}) = \boldsymbol{N}(\hat{x}_{k+1|k}, \Sigma_{k+1|k})$ where
$$\hat{x}_{k+1|k} = \mathbb{E}\{x_{k+1}|y_{1:k}\},$$
$$\Sigma_{k+1|k} = \mathbb{E}\{(\hat{x}_{k+1} - x_{k+1})(\hat{x}_{k+1} - x_{k+1})'\}$$

Prediction form of Kalman filter equations:

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

K_k is called the Kalman filter gain.

Update for covariance $\Sigma_{k+1|k}$ is called Riccati equation. If $R_k \to \infty$ becomes Liapunov equation for optimal linear predictor.

---

1. Kalman filter is optimal filter for linear Gaussian model - it yields the conditional mean estimate.

2. In general filtering recursion, we can only compute *conditional* covariance of the state estimate (conditioned on $y_{1:k}$)

$$\mathrm{cov}(x_k|y_{1:k}) = \int x^2 \pi_k(x)dx - \left(\int x\pi_k(x)dx\right)^2$$

Evaluating unconditional covariance requires integration over all possible observation trajectories is impossible.
Remarkably Kalman covariance Σ_k is unconditional:

$$\Sigma_k = \mathrm{cov}(x_k) = \int \left[\int x^2 p(x_k|y_{1:k}) - \left(\int x p(x_k|y_{1:k})\right)^2\right] p(y_{1:k})dy_{1:k}$$

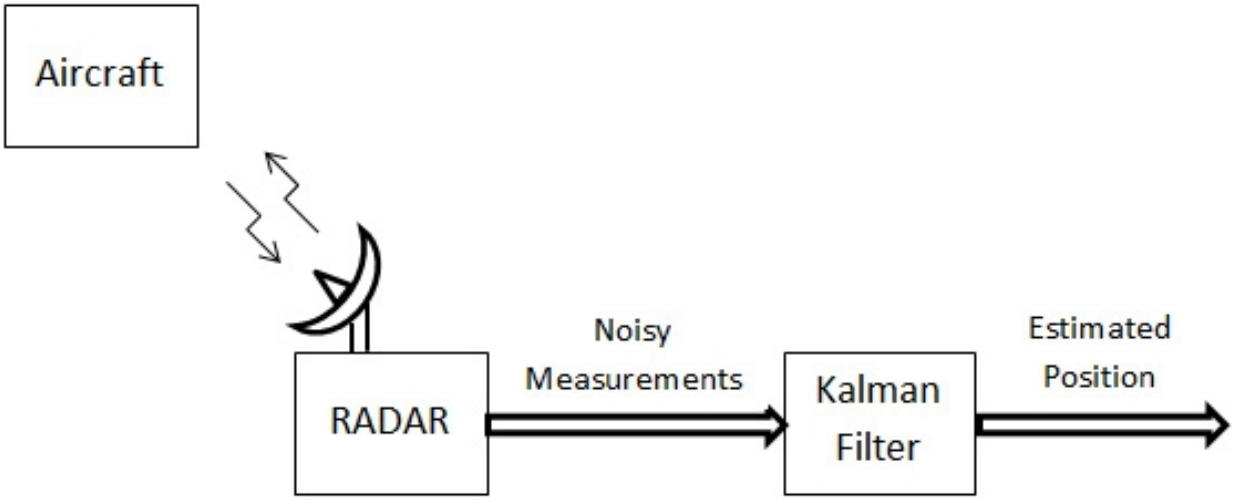
So KF automatically computes quality of state estimate.

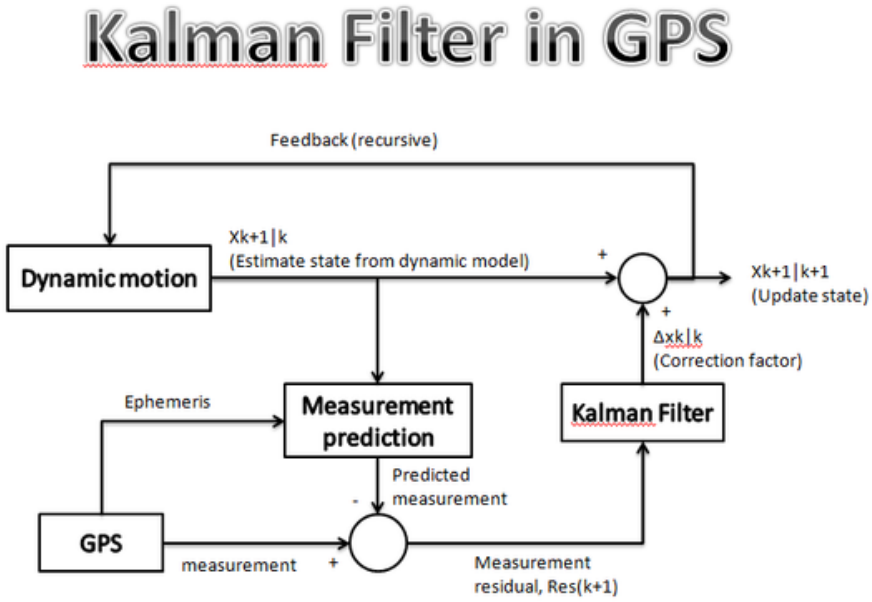
Kalman Predictor

$$\hat{x}_{k+i+1|k} = A_{k+i}\hat{x}_{k+i}, \quad \Sigma_{k+i+1|k} = A_{k+i}\Sigma_{k+i|k}A_{k+i}' + Q_{k+i}.$$

Predictor covariance update is called Liapunov equation

Examples of Kalman filters





`https://plus.maths.org/content/understanding-unseen`

How NASA used Kalman filter for spacecraft navigation

Properties of the Kalman Filter

- Kalman Filter is linear, discrete-time, finite dimensional system with 2 sufficient statistics: $\hat{x}_k$, Σ_k.
- Covariance Σ_k can be precomputed since it is independent of the data. (unconditional covariance).
- 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. Widely used in industrial control.
- In non-Gaussian noise, Kalman filter is linear minimum variance estimator: Amongst the class of linear estimators, Kalman filter is the minimum variance estimator.
- Above derivation is algebraic. Another method is basd on projection theorem (Hilbert space approach to linear functionals).
- Derivation of Kalman filter follows straightforwardly from Theorem 5

For details on Kalman filters see "Optimal Filtering" by B.D.O Anderson and J.B. Moore, Prentice Hall, 1979.

`https://www.mathworks.com/videos/understanding-kalman-filters-part-1-why-use-kalman-filters-html`

Optimal Filter 2. HMM Filter

Recall HMM is (P, B, π_0).

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

HMM Filter: Since $\mathcal{X} = \{1, \ldots, X\}$, so

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

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

In matrix-vector notation:

$$B_{y_k} = \text{diag}\begin{bmatrix} p(y_k|x_k = 1) & \cdots & p(y_k|x_k = X) \end{bmatrix}.$$

$$\pi_k = \begin{bmatrix} \pi_k(1) & \cdots & \pi_k(X) \end{bmatrix}'$$

$$\boxed{\pi_{k+1} = \frac{B_{y_{k+1}} P' \pi_k}{\mathbf{1}' B_{y_{k+1}} P' \pi_k}}$$

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

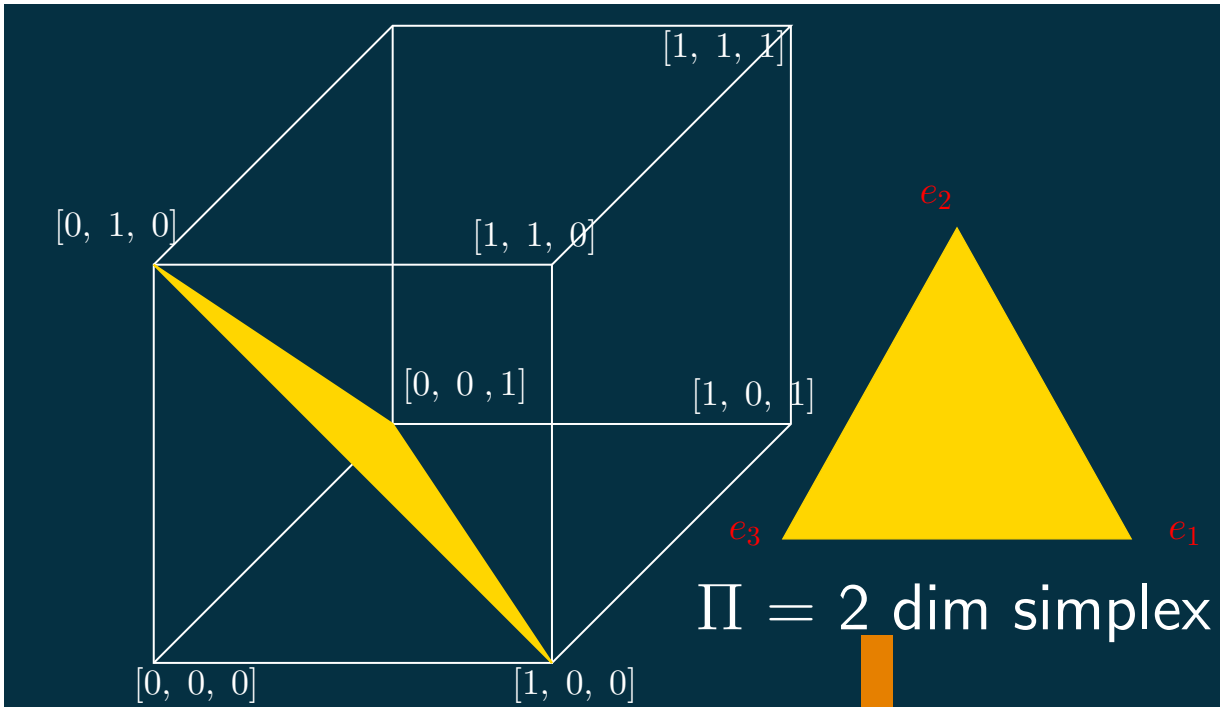
$$C'(\hat{x}_{k+1}) = \mathbb{E}\{C(x_{k+1}) | y_{1:k+1}\} = C' \pi_{k+1}.$$

HMM filter requires $O(X^2)$ multiplications at each time instant.
Note: If $P = I$, then HMM is called mixture-model. HMM filter specializes to recursive Bayesian estimation of rv

Belief Space: Geometric interpretation

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

Unit vectors $e_1, e_2, \ldots, e_X$ are vertices of this simplex Π.



Un-normalized HMM filter and Forward algorithm

$$q_{k+1} = p(x_{k+1}, y_{1:k+1}) = B_{y_{k+1}} P' q_k.$$

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

Called *forward* algorithm.

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

Fixed interval HMM Smoother

Aim: Given data $y_{1:N} = (y_1, \ldots, y_N)$, compute smoothed pmf

$$\pi_{k|N}(i) = P(x_k = i | y_{1:N}), \quad \text{for } k = 1, \ldots, N$$

Forward variable: $q_k(i) = P(x_k = i, y_{1:k})$. Backward variable:

$$\beta_{k|N}(i) = p(y_{k+1:N} | x_k = i), \ \beta_{k|N} = \Big[\beta_{k|N}(1), \ldots, \beta_{k|N}(X)\Big]'.$$

$$\implies p(x_k = i, y_{1:N}) = p(x_k = i, y_{1:k}) p(y_{k+1:N} | x_k = i) = q_k(i)\beta_{k|N}(i).$$

$$\implies \pi_{k|N}(i) = \frac{\frac{q_k(i)}{\mathbf{1}'q_k}\beta_{k|N}(i)}{\sum_{j=1}^{X} \frac{q_k(j)}{\mathbf{1}'q_k}\beta_{k|N}(j)} = \frac{\pi_k(i)\beta_{k|N}(i)}{\sum_{j=1}^{X} \pi_k(j)\beta_{k|N}(j)}$$

Backward vector $\beta_{k|N}$ is computed via backward recursion:

$$\boxed{\beta_{k|N} = PB_{y_{k+1}}\beta_{k+1|N}, \quad k = N-1, \ldots, 1, \quad \beta_{N|N} = \mathbf{1}_X}$$

Summary: Smoothed estimate $\pi_{k|N}(i) = p(x_k = i | y_{1:N})$, $i = 1, \ldots, X$ computed via *forward backward algorithm*:

$$\pi_k = \frac{B_{y_k} P' \pi_{k-1}}{\mathbf{1}' B_{y_k} P' \pi_{k-1}} \quad k = 1, 2, \ldots, N, \quad \text{(forward filter)}$$

$$\beta_{k|N} = PB_{y_{k+1}}\beta_{k+1|N}, \quad k = N-1, \ldots, 1, \ \beta_{N|N} = \mathbf{1} \quad \text{(backward)}$$

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

Scaling: Must scale $\beta_{k|N}$ during the backward recursion to avoid numerical underflow. Scale factor cancels out in $\pi_{k|N}$.

Remarks about forward-backward algorithm:

1. Derivation of backward recursion:

$$\begin{aligned}\beta_{k|N}(i) &= p(y_{k+1:N}|x_k = i) = p(y_{k+1}, y_{k+2:N}|x_k = i) \\ &= \sum_{j=1}^{X} p(y_{k+2:N}|x_{k+1} = j, y_{k+1}, x_k = i)p(x_{k+1} = j, y_{k+1}|x_k = i) \\ &= \sum_{j=1}^{X} \beta_{k+1|N}(j)\, B_{j,y_{k+1}}\, P_{ij}\end{aligned}$$

2. Initialization of $\beta_{N|N}$: At time N, foward backward algorithm yields:

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

But $\pi_N(i) = \pi_{N|N}(i)$. Therefore $\beta_{N|N}(i) = 1$ for all i.

3. Memory and Computational Cost: Forward backward algorithm needs

1. $O(XN)$ memory to store the X vectors π_k and $\beta_{k|N}$, $k = 1, \ldots, N$.
2. $O(X^2N)$ multiplications.

Smoothers for Functionals of State

Aim: Compute smoothed estimate of functional

$$\mathbb{E}\{\sum_{k=1}^{N} \phi(x_k)|y_{1:N}\}$$

Why? Need this in ML estimation algorithms for HMMs.

Ex1: $D_i(N)$ denote duration time in state i until time N. Then

$$\mathbb{E}\{D_i(N)|y_{1:N}\} = \mathbb{E}\{\sum_{k=1}^{N} I(x_k = i)|y_{1:N}\}$$

$$= \sum_{k=1}^{N} P(x_k = i|y_{1:N}) = \sum_{k=1}^{N} \pi_{k|N}(i)$$

Ex2: $J_{ij}(N)$: number of jumps from i to j until time N. Then

$$\mathbb{E}\{J_{ij}(N)|y_{1:N}\} = \mathbb{E}\{\sum_{k=1}^{N} I(x_{k-1} = i, x_k = j)|y_{1:N}\}$$

$$= \sum_{k=1}^{N} P(x_{k-1} = i, x_k = j|y_{1:N}) = \sum_{k=1}^{N} \frac{\pi_k(i) P_{ij} B_{jy_{k+1}} \beta_{k+1|N}(j)}{\sum_{l=1}^{X} \pi_k(l) P_{lj} B_{jy_{k+1}} \beta_{k+1|N}(j)}$$

Markov Modulated Auto-regressive Time Series

Used in econometric modelling and maneuvering targets.

Example: Let x_k be finite state Markov chain. Consider

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

Identical to HMM filter with observation density

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

Viterbi Algorithm for HMMs

Recall HMM smoother computes conditional mean estimate

$$\mathbb{E}\{x_k|y_{1:N}\} = \operatorname*{argmin}_g \mathbb{E}\{g(y_{1:N}) - x_k\}^2$$

Given HMM obs sequence $y_{1:N} = (y_1, \ldots, y_N)$, Viterbi algorithm computes Maximum likelihood sequence estimate (MLSE):

$$\boxed{\hat{x}_{1:N} = \arg\max_{x_{1:N}} p(y_{1:N}, x_{1:N})}, \quad x_{1:N} = (x_1, \ldots, x_N).$$

$$\hat{x}_{1:N} = \arg\max_{x_{1:N}} p(y_{1:N}, x_{1:N}) = \arg\max_{x_{1:N}} \prod_{k=1}^{N} p(y_k|x_k)p(x_k|x_{k-1})p(x_1)$$

$$= \arg\max_{x_{1:N}} \sum_{k=1}^{N} \log p(y_k|x_k) + \log p(x_k|x_{k-1}) + \log p(x_1)$$

Viterbi algorithm computes MLSE $\hat{x}_{1:N}$ via forward dynamic programming given HMM obs sequence $y_{1:N}$. For $k = 1, 2, \ldots, N$

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

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

Terminate. $\hat{x}_N = \arg\max_i \delta_N(i)$

Backtrack. $\hat{x}_k = u_{k+1}(\hat{x}_{k+1}),\ k = N-1, \ldots, 1.$

Comparison of Kalman and HMM filter

(i) KF is linear filter: conditional mean is linear in observation. 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 iid noise v_k

(iv) Both HMM and KF are geometrically ergodic, i.e. they forget their initial condition exponentially fast under reasonable conditions on the model.

(v) **Martingale formulation of HMM**: Let $x_k \in \{e_1, \ldots, e_X\}$ denote states of Markov chain. HMM can be represented as

$$x_{k+1} = P' x_k + w_k$$
$$y_k = C x_k + v_k$$

where w_k is a martingale difference: $\mathbb{E}\{w_k | x_0, x_1, \ldots, x_k\} = 0$.

Since w is not Gaussian, Kalman filter is optimal linear estimator for state x_k of above HMM. HMM filter is optimal (nonlinear) estimator for x_k.

Approximate Filters

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

1. Grid approximation: HMM approximation

$$q_{k+1}(x_{k+1}) = p(y_{k+1}|x_{k+1}) \int_{\mathbb{R}} p(x_{k+1}|x_k) q_k(x_k) dx_k$$

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

$$q_{k+1}(r_j) = p(y_{k+1}|x_{k+1} = r_j) \sum_{i=1}^{M} p(x_{k+1} = r_j | x_k = r_i) q_k(r_i)$$

$O(M^2)$ computations at each time for M grid points. Approx error: $O(M^{-1/X})$ where X = state dim – suffers from curse of dimensionality.

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

3. MLSE estimators: Compute state sequence estimate $\arg\max_{x_1,\ldots,x_T} p(Y_T, x_1, \ldots, x_T)$. (e.g. Viterbi algorithm)

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

Particle filters. Sequential MCMC

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

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

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

Remarks: 1. Cloud of samples approximates a pdf; particle filter uses samples (positions) and scalar weights.

2. *Deterministic Grid.* For $\int_{\mathbb{R}^X}$, given N grid points, approx. error

$$\epsilon = O(N^{-1/X}) \implies N = O(\epsilon^{-X}) = \frac{c}{\epsilon^X}$$

Curse of dimensionality

3. *Particle filter*: Mean square error from CLT is $O(N^{-1})$. Randomization breaks the curse of dimensionality! e.g. if state dim $X = 10$, and N is # grid points:

N	10	100	1000
$N^{-1/X}$	0.794	0.631	0.501
N^{-1}	0.1	0.01	0.001

4. *Bootstrap particle filter:* D.Q Mayne, 1968, Automatica. It was re-invented in 1996. Particle filters are a class of *sequential Markov Chain Monte Carlo (MCMC) algorithms.*

Simple version. Bootstrap particle filter

Given samples $x_k^{(i)} \sim p(x_k|y_{1:k})$, sample $x_{k+1}^{(i)} \sim p(x_{k+1}|y_{1:k+1})$, $i = 1, \ldots, N$.

Step (i) *Predictor Update.* Given samples $x_k^{(i)} \sim p(x_k|y_{1:k})$ simulate from predicted density (Chapman Kolmogorov)

$$p(x_{k+1}|y_{1:k}) = \int p(x_{k+1}|x_k)p(x_k|y_{1:k})dx_k$$

Soln. For each particle $x_k^{(i)}$ simulate

$$\boxed{x_{k+1|k}^{(i)} \sim p(x_{k+1}|x_k = x_k^{(i)})}$$

Then composition method implies $x_{k+1|k}^{(i)} \sim p(x_{k+1}|y_{1:k})$.

Step (ii) *Bayes Update.* Given samples $x_{k+1|k}^{(i)} \sim p(x_{k+1}|y_{1:k})$, simulate

$$x_{k+1}^{(i)} \sim p(x_{k+1}|y_{1:k+1}) \propto p(y_{k+1}|x_{k+1})\, p(x_{k+1}|y_{1:k})$$

Soln. Use Sampling Importance Resampling (SIR) algorithm.

SIR Algorithm: Given samples $x^{(i)} \sim f(x)$ how to sample from $g(x)$? Both f and g can be un-normalized.

Step 1: Compute $w^{(i)} = \frac{g(x^{(i)})}{f(x^{(i)})}$, $i = 1, \ldots, N$.

Compute normalized weights $\tilde{\omega}^{(i)} = \frac{w^{(i)}}{\sum_{j=1}^{N} w^{(j)}}$.

Step 2: Sample integers $I^{(i)} \sim \tilde{\omega}^{(i)}$, $i = 1, \ldots, N$.

Step 3 (resampling): Choose samples $x^{(I^{(i)})}$, $i = 1, \ldots, N$.

Then these resampled samples $x^{(I^{(i)})} \sim g(x)$

Bayes update for bootstrap filter: In SIR algorithm, choose

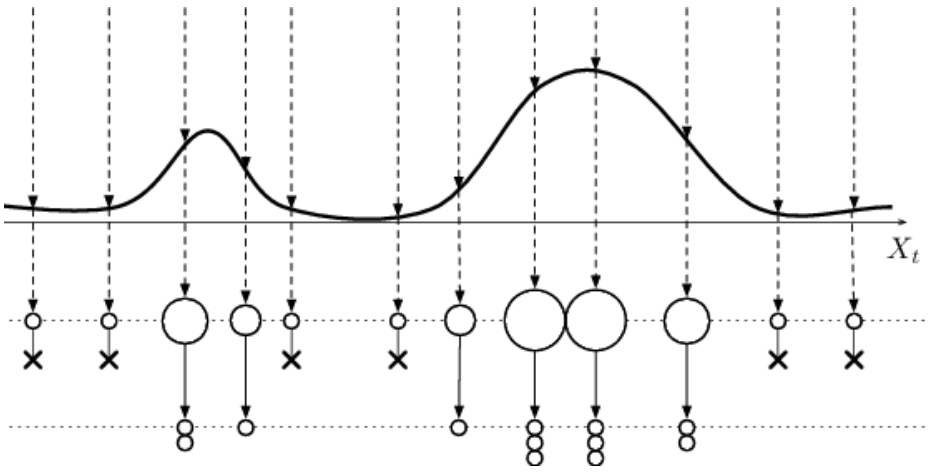
$$f(x) = p(x_{k+1}|y_{1:k})$$

$$g(x) = p(y_{k+1}|x_{k+1}) f(x) \propto p(x_{k+1}|y_{1:k+1})$$

$$\Longrightarrow w^{(i)} = p(y_{k+1}|x^{(i)}_{k+1|k})$$

Given samples $x_k^{(i)} \sim p(x_k|y_{1:k})$, $i = 1, \ldots, N$,

1. sample integers $I^{(i)} \sim \tilde{\omega}^{(i)} = \frac{w^{(i)}}{\sum_{j=1}^{N} w^{(j)}}$.
2. Set $\boxed{x^{(i)}_{k+1} = x^{(I^{(i)})}_{k+1|k},\ i = 1, \ldots, N}$.



weights $\tilde{\omega}^{(i)}$

resampled unit weight particles

Summary of Bootstrap Particle Filter

Given samples $x_k^{(i)} \sim p(x_k|y_{1:k})$, $i = 1, \ldots, N$

1. (Composition method) For each particle $x_k^{(i)}$ simulate $x_{k+1|k}^{(i)} \sim p(x_{k+1}|x_k)$.
2. Compute weight $w^{(i)} = p(y_{k+1}|x_{k+1|k}^{(i)})$ and therefore normalized weight $\tilde{\omega}^{(i)} = \frac{w^{(i)}}{\sum_{j=1}^N w^{(j)}}$.
3. (Resampling) Sample integers $I^{(i)} \sim \tilde{\omega}^{(i)}$. Choose $x_{k+1}^{(i)} = x_{k+1|k}^{(I^{(i)})}$, $i = 1, \ldots, N$. Resampling step prevents degeneracy (else algorithm will crash)

Remarks: 1. Bootstrap particle filter approx is

$$\sum_{i=1}^N \tilde{\omega}^{(i)} \delta(x - x_{k+1|k}^{(i)}) \approx p(x_{k+1} = x|y_{1:k+1})$$

2. Estimate conditional mean as either

$$\boxed{\mathbb{E}\{x_{k+1}|y_{1:k+1}\} \approx \sum_{i=1}^N \tilde{\omega}^{(i)}\, x_{k+1|k}^{(i)}} \quad \text{(lower variance)}$$

$$\text{OR} \quad \boxed{\mathbb{E}\{x_{k+1}|y_{1:k+1}\} \approx \frac{1}{N} \sum_{i=1}^N x_{k+1}^{(i)}} \quad \text{(higher variance)}$$

since $\mathbb{E}\{\frac{1}{N}\sum_{i=1}^N x_{k+1}^{(i)} | x_{k+1|k}^{(i)}, i = 1, \ldots, N\} = \sum_{i=1}^N \tilde{\omega}^{(i)}\, x_{k+1|k}^{(i)}$

General Particle Filter Algorithm

Consider general model:

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

Can have long range dependency in state and observation compared to $p(x_{k+1}|x_k)$ and $p(y_k|x_k)$.

Aim: Estimate $\mathbb{E}\{\phi(x_{0:k})|y_{1:k}\}$ via sequential MCMC.
If we choose $\phi(x_{0:k}) = x_k$, then yields $\mathbb{E}\{x_k|y_{1:k}\}$.

Main idea – Sequential Importance Sampling.
Given samples (trajectories) $x^{(i)}_{0:k-1} \sim p(x_{0:k-1}|y_{1:k-1})$, simulate samples (trajectories) $x^{(i)}_{0:k} \sim p(x_{0:k}|y_{1:k})$ by *re-using* samples $x^{(i)}_{0:k-1}$, $i = 1, \ldots, N$.

1. Simulate $x^{(i)}_k \sim \pi(x_k|x_{0:k-1}, y_{1:k})$.
2. Set $x^{(i)}_{0:k} = (x^{(i)}_{0:k-1}, x^{(i)}_k)$.

General Particle Filter

1. **Generalized Model**: $p(x_{k+1}|x_{0:k}), \quad p(y_k|y_{1:k-1}, x_{1:k})$.
Aim: Simulate trajectories $x_{0:k}^{(i)}$, $i = 1, \ldots, N$ from $p(x_{0:k}|y_{1:k})$ for $k = 1, 2 \ldots$ via sequential MCMC.
Clearly $x_k^{(i)} \sim p(x_k|y_{1:k})$. (no integration needed to marginalize!)

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 from importance density: $x_{0:k}^{(i)} \sim \pi(x_{0:k}|y_{1:k})$. From SLLN

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

So density estimate $p(x_{0:k} = x_{0:k}^{(i)}|y_{1:k}) \approx \sum_{i=1}^{N} \delta(x_{0:k}^{(i)}) \frac{w_k^{(i)}}{\sum_{j=1}^{N} w_k^{(j)}}$

3. **Sequential Importance sampling**: (real time)
Aim: Given samples $x_{0:k-1}^{(i)} \sim p(x_{0:k-1}|y_{1:k-1})$, simulate samples $x_{0:k}^{(i)} \sim p(x_{0:k}|y_{1:k})$ by *re-using* samples $x_{0:k-1}^{(i)}$, $i = 1, \ldots, N$.
Note $\pi(x_{0:k}|y_{1:k}) = \pi(x_0|y_{1:k}) \prod_{t=1}^{k} \pi(x_t|x_{0:t-1}, y_{1:k})$
Real time: $\pi(x_{0:k}|y_{1:k}) = \pi(x_0) \prod_{t=1}^{k} \pi(x_t|x_{0:t-1}, y_{1:t})$.

Step (i). Simulate $x_k^{(i)} \sim \pi(x_k|x_{0:k-1}, y_{1:k})$. Set $x_{0:k}^{(i)} = (x_{0:k-1}^{(i)}, x_k^{(i)})$.
Step (ii). Update importance weights in real time: For $i = 1, \ldots, N$,

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

Summary: General 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 $\tilde{\omega}_k^{(i)}$ of particles

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

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

Selection step: (optional - to combat degeneracy).
Effective number of particles: $\hat{N} = \frac{1}{\sum_{i=1}^N [\tilde{\omega}_k^{(i)}]^2}$. 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 $\tilde{\omega}_k^{(i)}$ to obtain N new particles $x_{0:k}^{(i)}$, $i = 1, \ldots, N$ with unit weight.

Summary: The particle filter approximation is:

$$p(x_{0:k}|y_{1:k}) \approx \sum_{i=1}^N \tilde{\omega}_k^{(i)} \delta(x_{0:k}^{(i)}), \quad p(x_k|y_{1:k}) \approx \sum_{i=1}^N \tilde{\omega}_k^{(i)} \delta(x_k^{(i)}),$$

$$\mathbb{E}\{x_n|y_{0:n}\} \approx \sum_{i=1}^N \tilde{\omega}_k^{(i)} x_n^{(i)}$$

Example. Bootstrap Particle Filter (revisited)

Suppose model is: $p(x_{k+1}|x_k), \quad p(y_k|x_k)$
Choose importance density $\pi(x_k|x^{(i)}_{0:k-1}, y_{1:k}) = p(x_k|x^{(i)}_{k-1})$

- Sample N particles $\tilde{x}^{(i)}_k \sim p(x_k|x^{(i)}_{k-1})$.
  Set $\tilde{x}^{(i)}_{0:k} = (x^{(i)}_{0:k-1}, \tilde{x}^{(i)}_k)$.
- Importance weight update:

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

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

- Selection step: (optional) Use SIR to resample.
  Sample integers $I^{(i)} \sim \tilde{\omega}^{(i)}_k$
  Then $x^{(i)}_k = \tilde{x}^{(I^{(i)})}_k$, $i = 1, \ldots, N$.
  Each resampled particle has weight $w_k(x^{(i)}_{0:k}) = 1$.

Bootstrap particle filter discussed earlier is special case where selection step is done at each iteration.
Resampling increases variance but prevents degeneracy.

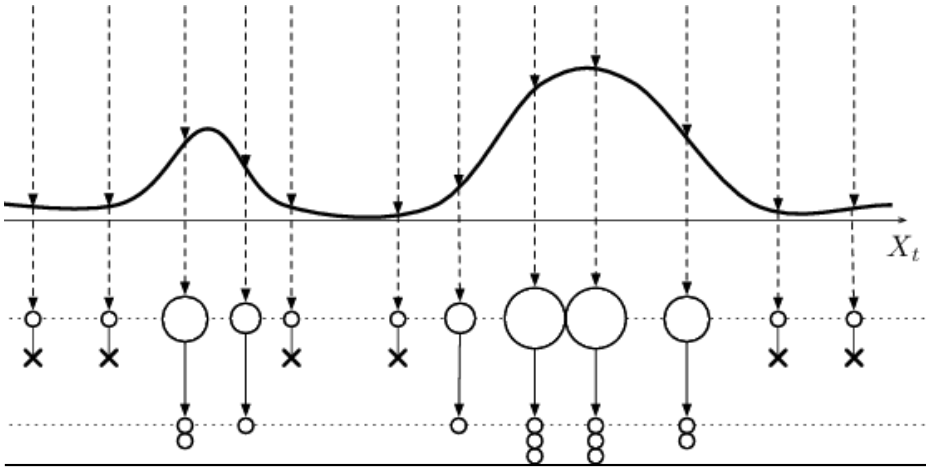
Implementation Issues

1. **Degeneracy**: Variance of importance weights $w_k^{(i)}$ grows with time. Most 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.



Main idea: Sampling Importance Resampling.

Suppose $x^{(i)} \sim \pi(x|y)$ and $w^{(i)} = \frac{p(x^{(i)}|y)}{\pi(x^{(i)}|y)}$, $i = 1, \ldots, N$.

Define normalized weights $\tilde{\omega}^{(i)} = \frac{w^{(i)}}{\sum_{j=1}^N w^{(j)}}$.

Sample integers $I^{(i)} \sim \tilde{\omega}^{(i)}$, $i = 1, \ldots, N$.

Then $x^{(I^{(i)})} \sim p(x|y)$ and each $x^{(I^{(i)})}$ has unit weight.

Before selection: $p(x_{0:k}|y_{0:k}) \propto \sum_i w_k^{(i)} \delta(x_{0:k}^{(i)})$

After selection: $p(x_{0:k}|y_{0:k}) \propto \sum_i \delta(x_{0:k}^{I^{(i)}})$

Selection (resampling) scheme increases variance - so compromise between degeneracy and variance.

2. **Choice of importance function**: Many possibilities:

(i) Bootstrap filter: Choose $\pi(x_k|x_{0:k-1}, y_{0:k}) = p(x_k|x_{k-1})$.

Then $\boxed{w_k^{(i)} = w_{k-1}^{(i)} p(y_k|x_k^{(i)})}$.

If SIR step applied every time, then $w_k^{(i)} = p(y_k|x_k^{(i)})$.

Sensitive to outliers. Particles evolve indpt of obs.

(ii) Fixed importance function: $\pi(x_k|x_{0:k-1}, y_{0:k}) = p(x_k)$

(iii) *Optimal Choice*: Minimizes

$\text{Var}(w_k|y_{1:k}, x_{0:k-1}) = \text{Var}(\frac{p(x_{0:k}|y_{1:k})}{\pi(x_{0:k}|y_{1:k})}|y_{1:k}, x_{0:k-1})$ wrt x_k.

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

Since $p(x_k|x_{k-1}, y_k) = \frac{p(y_k|x_k)p(x_k|x_{k-1})}{p(y_k|x_{k-1})}$, it follows that

$$\boxed{w_k^{(i)} = w_{k-1}^{(i)}\, p(y_k|x_{k-1}^{(i)})}$$

Compute $w_k^{(i)}$, sample $x_n^{(i)}$ in parallel since indpt of $x_k^{(i)}$!

(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:

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

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

$$\Sigma^{-1} = \Sigma_v^{-1} + C'\Sigma_w^{-1}C; \quad m_k = \Sigma(\Sigma_v^{-1} f(x_{k-1})'_C \Sigma_w^{-1} y_k)$$

$$p(y_k|x_{k-1}) = N\left(Cf(x_{k-1}), (\Sigma_w + C\Sigma_v C')\right)$$

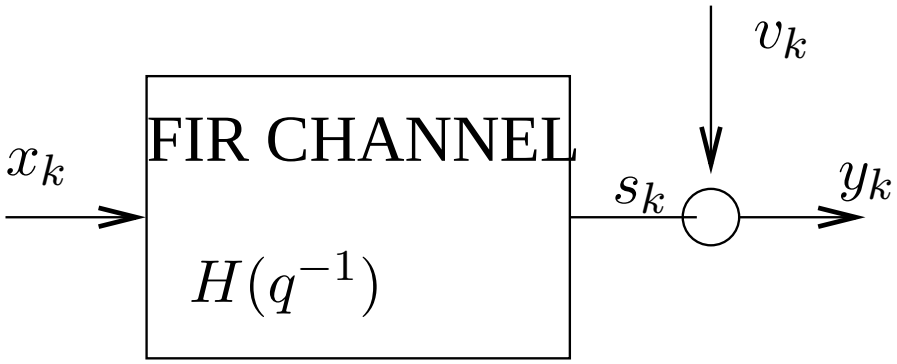
Part IV: ML Parameter Estimation

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
  - 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).

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.

ML Estimation

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

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

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 (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 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.

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 chain. Parameter is

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

Note the parameter constraints:

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

The likelihood and log likelihood functions are

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

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

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

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

$J_{ij} = \#$ 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

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

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

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

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

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 $Y_N = (y_1, \ldots, y_N)$ and initial parameter estimate $\theta^{(0)} \in \Theta$.
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) Run un-normalized HMM filter α^θ_k, $k = 1, \ldots, N$

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

  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 $R^\theta_k(i) = \nabla_\theta \alpha^\theta_k(i)$, $k = 1, \ldots, N$ is

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

- Update parameter estimate via Newton Raphson as:

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

`fmincon` in Matlab is general purpose optimization algorithm.

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 $\theta = (P, B)$

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

$$\beta^\theta_k(i) = p(Y_{k+1:N}|x_k = q_i, \theta) = \sum_{j=1}^{X} \beta^\theta_{k+1}(j) P_{ij} b_j(y_{k+1})$$

$$\gamma^\theta_k(i) = P(x_k = q_i|Y_N, \theta) = \frac{\alpha^\theta_k(i)\beta^\theta_k(i)}{\sum_{i=1}^{X} \alpha^\theta_k(i)\beta^\theta_k(i)}$$

$$\gamma^\theta_k(i,j) = P(x_k = q_i, x_{k+1} = q_j|Y_N, \theta)$$

$$= \frac{\alpha^\theta_k(i) P_{ij} b_j(y_{k+1}) \beta^\theta_{k+1}(j)}{\sum_{i=1}^{X}\sum_{j=1}^{X} \alpha^\theta_k(i) P_{ij} b_j(y_{k+1}) \beta^\theta_{k+1}(j)}$$

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 α_k^θ and backward filter β_k^θ are X dimensional vectors. Their computation is called *forward backward algorithm.*

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

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 $\alpha_k^\theta(i)$, $\beta_k^\theta(i)$, $\gamma_k^\theta(i)$, $k = 1, \ldots, N$.

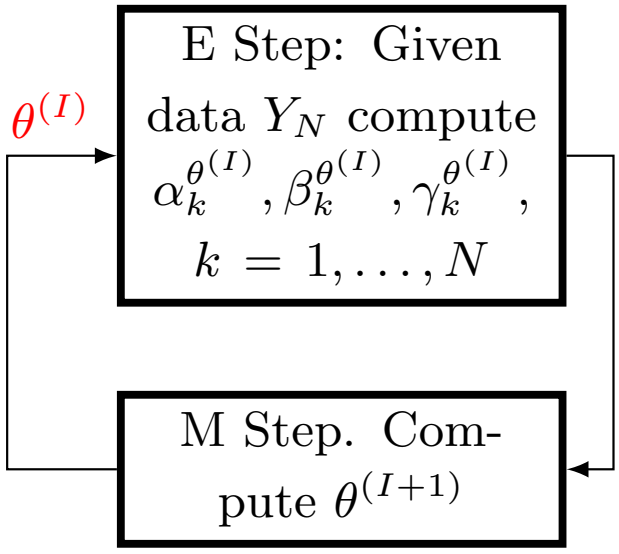
Compute expected duration time $\hat{D}_N^\theta(i) = \sum_{k=1}^N \gamma_k^\theta(i)$, and expected number of jumps $\hat{J}_N^\theta(i,j) = \sum_{k=1}^N \gamma_k^\theta(i,j)$.

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

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

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 $X_N = (x_1,\ldots,x_N)$, $Y_N = (y_1,\ldots,y_N)$.
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 $\theta^{(I)}$, $I = 1,2,\ldots$ as follows:

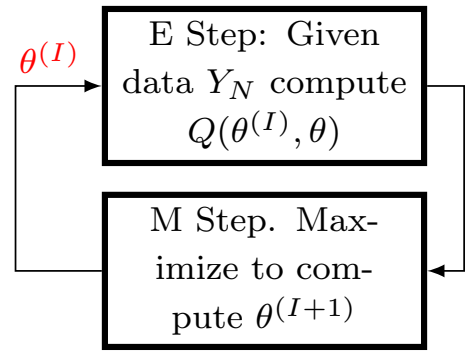
Each iteration consists of 2 steps:

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

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

- *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 $Q(\theta^{(I)}, \theta) = E\{\ln p(Y_N, X_N|\theta)|Y_N, \theta^{(I)}\}$
Result: The auxiliary likelihood $Q(\theta^{(I)}, \theta)$ is:

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

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 $0 \le P_{ij} < 1$, $\sum_j P_{ij} = 1$ 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.

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

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

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

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

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

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

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

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

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

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

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

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:

$$Q(\theta^{(I)}, \theta) = -\frac{N}{2}\ln\sigma_v^2 - \frac{1}{2\sigma_v^2}\sum_{t=1}^{N}\mathbb{E}\{(y_t - x_t)^2|Y_N, \theta^{(I)}\}$$

$$- \frac{N}{2}\ln\sigma_w^2 - \frac{1}{2\sigma_w^2}\sum_{t=1}^{N}\mathbb{E}\{(x_t - a\, x_{t-1})^2|Y_N, \theta^{(I)}\}$$

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 σ_v and σ_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

$$\begin{aligned} X|\Theta &\sim p(x|\theta) \\ Y|X &\sim p(y|x) \end{aligned} \tag{12}$$

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

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

Estimate θ^* 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 θ, Q fn increases slower than log likelihood in terms of θ. That is:

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

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

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

$\implies$ LHS (A) $> 0 \implies$ RHS (A) $> 0 \implies P(Y_N|\theta^{(I+1)}) \geq P(Y_N|\theta^{(I)})$
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:

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

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

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

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

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

Jensen's inequality:

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

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

Dempster, Laird and Rubin invented EM algorithm, 1977.

Consistency of MLE (advanced)

Suppose $y_1, \ldots, y_N$ is an iid sequence of observations. $\theta^* \in \Theta$ true parameter. MLE θ_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 Θ 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,

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

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

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

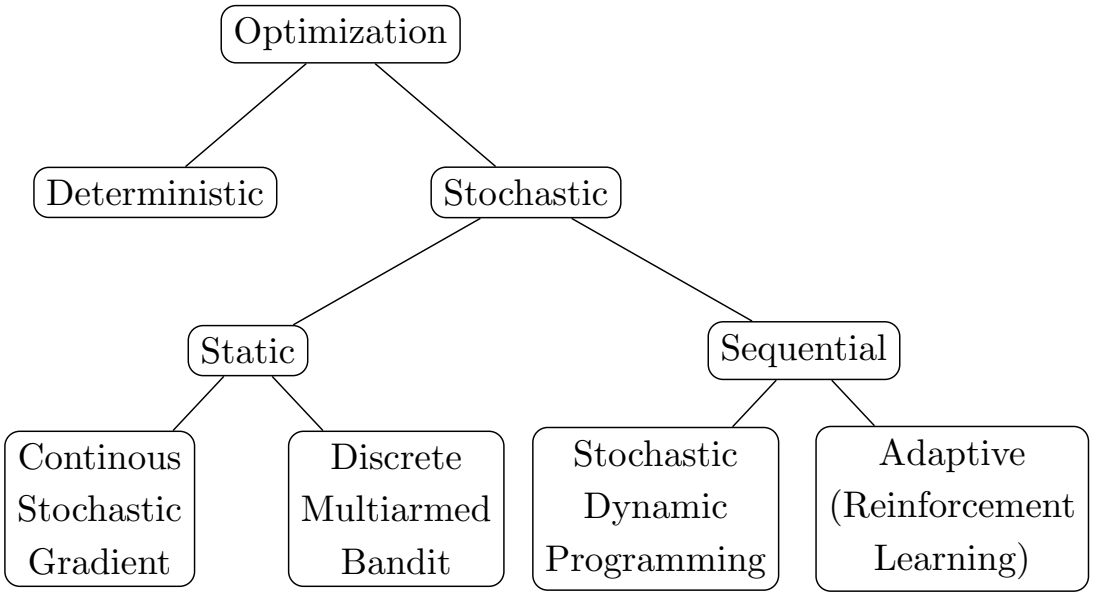
$$\lim_{N\to\infty} \sup_{\theta\in\Theta} \frac{1}{N} \log p(y_1, \ldots, y_N|\theta) \overset{\text{w.p.1}}{\to} K(\theta, \theta^*) \text{ uniform convergence}$$

Requires stochastic equicontinuity of $\{l_n(\theta)\}$:

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

Part V. Stochastic Optimization

Optimization. Big Picture



Deterministic Optimization.

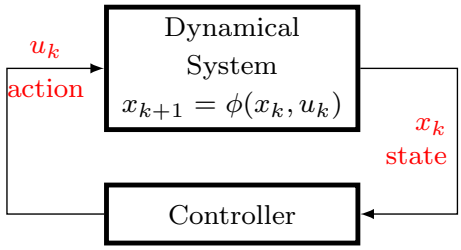
1. *Deterministic Static Continuous Optimization*
$\min_{x\in\mathbb{R}^d} F(x)$ with constraints $H(x)=0$, $G(x)\leq 0$.
(i) Convex (easy) (ii) Non-convex (hard)

2. *Deterministic Static Discrete Optimization*: $\min_{x\in\mathcal{Z}^d} F(x)$ Integer programming problems are typically NP complete. We will do the stochastic version (multi-armed bandits).

3. *Deterministic Sequential Optimization*: Optimal control.

$$\min_{u_0,\ldots,u_{N-1}} F(x_1,\ldots,x_N,u_0,\ldots,u_{N-1})$$

$$\text{subject to } x_{k+1}=\phi(x_k,u_k)$$



Soln: stochastic dynamic programming

Reinforcement Learning

> Reinforcement learning is learning what to do–how to map situations to actions–so as to maximize a numerical reward signal. The learner is not told which actions to take, . . . but instead must discover which actions yield the most reward by trying them. . . Actions may affect not only the immediate reward but also the next situation and, through that, all subsequent rewards. These two characteristics–trial-and-error search and delayed reward–are the two most important distinguishing features of reinforcement learning.

R. Sutton and A. Barto. Reinforcement learning: An introduction. MIT Press, 1998

Example 1. Social network media

1. User comes to Google (with history of previous visits, IP addresses, data related to google account)
2. Google chooses info to present (urls, ads, news stories)
3. User reacts to presented info (clicks, comes back, clicks again, etc)

Google wants to interactively choose content using observed feedback to improve future content choices.

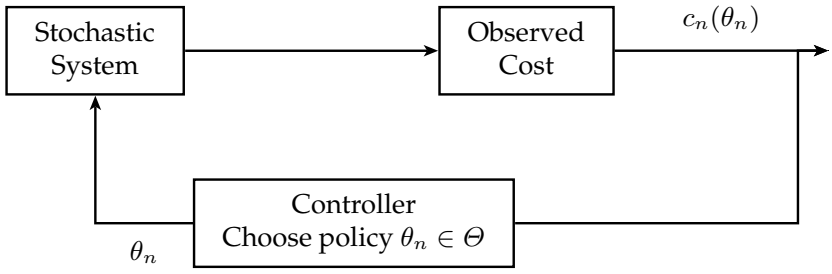
Example 2. Clinical Decision Making

1. Patient visits doctor (with history of symptoms, medical history, test results)
2. Doctor chooses treatment
3. Patient responds to treatment

Doctor wants to determine strategy for choosing targeted treatment for individual patient.

We cover 3 types of reinforcement learning:

1. Bandit problems: optimal tradeoff btw exploration and exploitation
2. Stochastic Gradient algorithms: how to learn and optimize a cont-valued objective.
3. Markov Decision processes: learn & control Markov process



RL for Discrete Stochastic Optimization

Bandit Setting. For $n = 1, \ldots, N$:

1. World produces context $\theta \in \Theta = \{1, 2, \ldots, S\}$
2. Learner chooses action $\theta_n \in \Theta$
3. World reacts with reward $c_n(\theta_n)$

Learn good policy for choosing actions given context.

Aim. Compute $\theta^* = \underset{\theta \in \Theta}{\operatorname{argmax}} C(\theta)$, where $C(\theta) = \mathbb{E}\{c_n(\theta)\}$.

- $\Theta = \{1, 2, \ldots, S\}$ denotes the finite (discrete) set.
- $c_n(\theta)$ is the observed reward incurred when using θ

For each $\theta \in \Theta$, $c_n(\theta) \sim p_\theta(c)$ are iid.
But pdf $p_\theta(c)$ is not known. So

$$C(\theta) = \int c_n(\theta) p_\theta(c) dc$$

cant be evaluated; otherwise deterministic integer optimization.

Applications: Gambling, experimental design, biostatistics, reinforcement learning (exploration vs exploitation), adaptive modulation, optimal search, ad placement, website optimization, packet routing.
`https://arxiv.org/pdf/1204.5721.pdf`

Some basic observations

- We don't know reward of actions not taken
- Exploration is required
- observations are independent over time (otherwise problem becomes a dependent bandit which is harder to solve)
- Compared to Bayesian and MLE, we are now doing non-parameteric estimation. We dont know the noise distributions.

Approach 1. Brute force: For each $\theta \in \Theta$, compute time average

$$\hat{c}_N(\theta) = \frac{1}{N}\sum_{n=1}^{N} c_n(\theta), \text{ for large } N$$

Then choose $\hat{\theta} = \text{argmax}_{\theta\in\Theta}\, \hat{c}_N(\theta)$.
Since $\{c_n(\theta)\}$ are i.i.d.

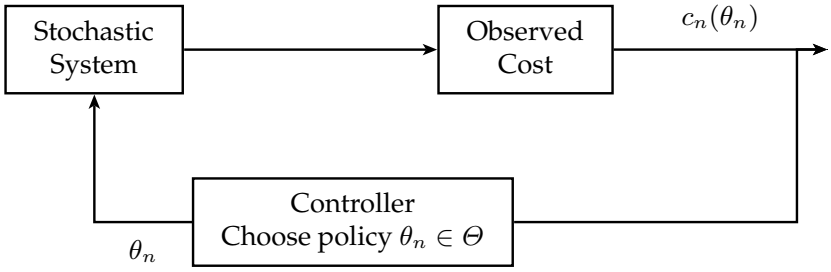
$$\hat{c}_N(\theta) \to \mathbb{E}\{c_n(\theta)\} \text{ w.p.1 as } N \to \infty$$

Highly inefficient since $\hat{c}_N(\theta)$ needs to be evaluated for each $\theta \in \Theta$. Evaluating $\hat{c}_N(\theta)$ for $\theta \notin \theta^*$ contributes nothing to estimating $\hat{c}_N(\theta^*)$.

Homework: read thru

`https://blogs.mathworks.com/loren/2016/10/10/multi-armed-bandit-problem-and-exploration-vs-exploita`

Discrete stochastic opt: Finding θ^* is easy by brute force.
Aim: Find efficient algorithm that learns from past decisions (feedback system) and spends more time obtaining observations $c_n(\theta)$ near θ^* (exploitation) and less in other areas.



How to quantify efficiency of discrete stoch opt algorithm? Let

$$\pi_N(\theta) = \frac{1}{N}\sum_{k=1}^{N} I(\theta_k = \theta)$$

= fraction of times θ is chosen by algorithm until time N.
Option 1: Design algorithm s.t. as $N \to \infty$, $\pi_N(\theta^*) > \pi_N(\theta)$. (Chapter 17.4 of my book; also see problem sheet)
Option 2: Design algorithm s.t. regret is minimized. Choose θ_n at time $n = 1, \ldots, N$ so as to minimize regret at time N:

$$\boxed{R_N = \mathbb{E}\Big\{N c_n(\theta^*) - \sum_{n=1}^{N} c_n(\theta_n)\Big\} = NC(\theta^*) - N\sum_{\theta\in\Theta}\pi_N(\theta)C(\theta)}$$

$$= N\sum_{\theta\in\Theta}\pi_N(\theta)\big[C(\theta^*) - C(\theta)\big]$$

Regret: How much do I lose by choosing suboptimal arms?
If I only played optimal arm, regret is 0.

Result [Lai & Robbins 1985]. As $N \to \infty$, smallest possible regret is achieved by any algorithm is $O(\log N)$ and

$$\mathbb{E}\{\pi_N(\theta)\} = \left[\frac{1}{D(p_\theta||p^*)}\right] \frac{\log N}{N}, \text{ where}$$

$$D(p_\theta||p^*) = \int p_\theta(x) \log \frac{p_\theta(x)}{p^*(x)}\, dx. \text{ (Kullback Leibler divergence)}$$

Over time horizon N, must explore suboptimal candidates at least $\log N$ times to determine optimal candidate.

T. Lai and H. Robbins. *Asymptotically efficient adaptive allocation rules.* Advances in Applied Math 6.1 (1985): 4-22.

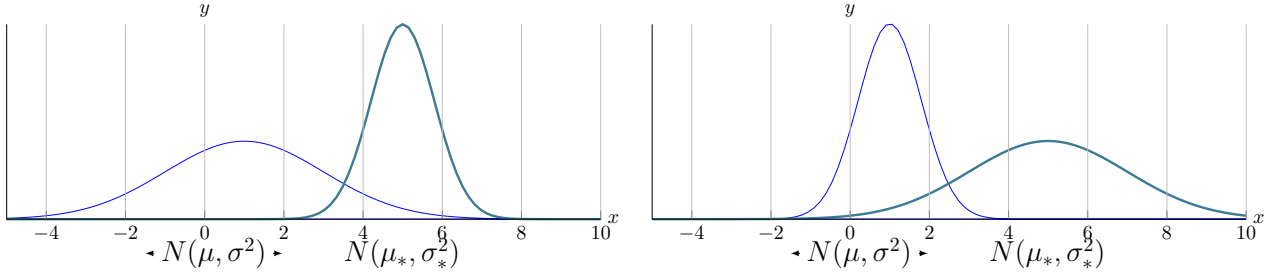
Example 1: Suppose $p^* \sim N(\mu_*, \sigma_*^2)$, $p_\theta \sim N(\mu, \sigma^2)$. Then

$$D(p_\theta||p^*) = \int p_\theta(x) \log \frac{p_\theta(x)}{p^*(x)}\, dx = \log \frac{\sigma_*}{\sigma} + \frac{\sigma^2 + (\mu - \mu_*)^2}{2\sigma_*^2} - \frac{1}{2}$$

If $\mu_* \gg \mu$, then $D(p_\theta||p^*)$ is large. So $\mathbb{E}\{\pi_N(\theta)\}$ & regret small.

If σ_* is small, $D(p_\theta||p^*)$ is large. So $\mathbb{E}\{\pi_N(\theta)\}$ & regret small.

Less obvious: To reduce regret: small σ_* better than small σ



Upper Confidence Bound (UCB) Algorithm

Last 15 years: Substantial progress in designing $\log N$ regret algorithms for finite time horizon N. UCB is famous example.

B = upper bound: $\max_{\theta\in\Theta} c_n(\theta) \le B$. Exploration factor $\xi > 1$.

$$\text{visit}_{\theta,n} = n\pi_n(\theta) : \#\text{ of visits to candidate } \theta \text{ until time } n.$$

Step 0. **Initialization.** Simulate each $\theta \in \Theta = \{1, 2, \ldots, S\}$ once to obtain $c_1(\theta)$.
Set $\widehat{c}_{\theta,S} = c_1(\theta)$ and $\text{visit}_{\theta,S} = 1$. Set $n = S + 1$.

Step 1a. **Sampling.** At time n sample candidate solution

$$\theta_n = \operatorname*{argmax}_{\theta\in\Theta} \left[\widehat{c}_{\theta,n-1} + B\sqrt{\frac{\xi \log n}{\text{visit}_{\theta,n-1}}} \right]$$

$$\text{where } \widehat{c}_{\theta,n-1} = \frac{1}{\text{visit}_{\theta,n-1}} \sum_{\tau=1}^{n-1} c_\tau(\theta_\tau) I(\theta_\tau = \theta)$$

Step 1b. **Evaluation.** Simulate arm θ_n to obtain $c_n(\theta_n)$.

Step 2. **Global Optimum Estimate.** $\theta_n^* \in \operatorname{argmax}_{\theta\in\Theta} \widehat{c}_{\theta,n}$.

Step 3. **Recursion.** Set $n \leftarrow n + 1$ and go to Step 1.

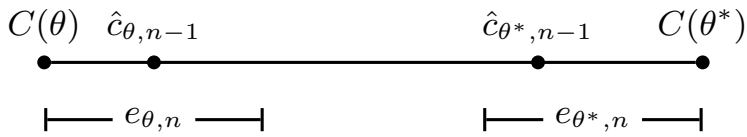
Theorem: Consider UCB Algorithm with exploration constant $\xi > 1$. Expected number of times suboptimal θ is sampled in n time points is

$$\mathbb{E}\{\text{visit}_{\theta,n}\} \le \frac{4B^2}{(C(\theta^*) - C(\theta))^2} \xi \log n + \frac{3}{2} + \frac{1}{2(\xi - 1)}$$

UCB achieves log regret for finite N. But scaling term $\neq$ KL distance.

Deterministic Intuition for UCB. Define $e_{\theta,n} = B\sqrt{\frac{\xi \log(n)}{\text{visit}_{\theta,n-1}}}$.
$\hat{c}_{\theta,n-1} + e_{\theta,n}$ in Step 1a is upper bound of confidence interval.
If θ and θ^* have been sampled many times, then with high probability

$$\text{(a): } \hat{c}_{\theta^*,n-1} \geq C(\theta^*) - e_{\theta^*,n} \quad \text{and (b): } \hat{c}_{\theta,n-1} \leq C(\theta) + e_{\theta,n}.$$



Step 1 of UCB is: $\theta_n = \text{argmax}_\theta[\hat{c}_{\theta,n-1} + e_{\theta,n}]$.
So with high probability, θ is no longer sampled if

$$\text{(c): } \hat{c}_{\theta,n-1} + e_{\theta,n} < \hat{c}_{\theta^*,n-1} + e_{\theta^*,n}.$$

(b) $\iff$ $\hat{c}_{\theta,n-1} + e_{\theta,n} \leq C(\theta) + 2e_{\theta,n}$
(a) $\iff$ $\hat{c}_{\theta^*,n-1} + e_{\theta^*,n} > C(\theta^*)$.
So a sufficient condition for (c) is

$$C(\theta) + 2e_{\theta,n} < C(\theta^*) \iff \text{visit}_{\theta,n-1} \geq \frac{4B^2}{(C(\theta^*) - C(\theta))^2}\, \xi \log n$$

UCB achieves logarithmic regret. But scaling term $\neq$ KL distance.

Stochastic Analysis. What does "with high probability" mean?
Recall *Hoeffding inequality*: For iid rv $X_n \in [0, B]$, with $\mu = \mathbb{E}\{X_n\}$ and $\hat{\mu}_n = \frac{1}{n}\sum_{k=1}^n X_k$,

$$\boxed{\begin{aligned} P(\hat{\mu}_n - \mu > \epsilon) &\leq \exp(-\frac{2n^2\epsilon^2}{B^2}) \\ P(|\hat{\mu}_n - \mu| > \epsilon) &\leq 2\exp(-\frac{2n^2\epsilon^2}{B^2}) \end{aligned}}$$

Complement of event (b) has small probability (similarly for (a)):

$$P(\hat{c}_{\theta,n-1} - C(\theta) > e_{\theta,n}) \leq \exp(-\frac{2\,\text{visit}^2_{\theta,n-1}\, e^2_{\theta,n}}{B^2}) = n^{-2\xi\,\text{visit}_{\theta,n-1}} \leq n^{-2\xi}$$

For complete proof see
https://link.springer.com/content/pdf/10.1023%2FA%3A1013689704352.pdf

Context. Multi-armed Bandit Problems

1. *Bayesian*: Given pdfs $f_1, \ldots, f_S$, (S iid arms) where f is known but unknown parameters (e.g. mean). Determine strategy μ to choose which arm to play at each time to maximize cumulative reward over an infinite horizon:

$$\max_{\mu} \mathbb{E}_{\mu}\{\sum_{k=0}^{\infty} \rho^k c_k(\theta_k)\}, \qquad \text{where } 0 \leq \rho < 1.$$

Note: strategy $\mu : \{\theta_1, \ldots, \theta_{k-1}, c_1(\theta_1), c_{k-1}(\theta_{k-1})\} \to \{1, \ldots, S\}$.

2. *Non-Bayesian*: Given S unknown pdfs, determine strategy μ to minimize regret for finite horizon N (UCB achieves $\log N$ regret)

$$R_N = \mathbb{E}_{\mu}\big\{N c_n(\theta^*) - \textstyle\sum_{k=1}^{N} c_n(\theta_n)\big\}$$

Closing Remarks. Bandits are "hot" in machine learning.

1. Thompson Sampling performs better than UCB in some cases.
http://proceedings.mlr.press/v23/agrawal12/agrawal12.pdf

2. Finite time analysis of UCB: P. Auer, N. Cesa-Bianchi, P. Fischer, Finite-time Analysis of the Multiarmed Bandit Problem, Machine Learning 2002.

3. https://arxiv.org/pdf/1204.5721.pdf has recent details on bandits and algorithms

4. Apart from regret minimization algorithms, several other types of algorithms, see Chapter 17.4 of my book and internet supplement.

5. https://www.stat.berkeley.edu/~bartlett/courses/2014fall-cs294stat260/lectures/bandit-ucb-notes.pdf has some nice notes on UCB.

Outline

1. **Probability for Signal Proc (Review)**:
   Uniform, Exponential, Gaussian Random variables
   Stochastic Simulation of Random Variables
2. **Probabilistic Models: DSP, Comms, Control**
   Random Processes: IID, Statistical Inference, Markov
   Applications: Digital Comms, Google Page Rank algorithm, Channel Models.
3. **Linear Time Series - Signal Processing**: Linear Stochastic Difference Eqns, ARMA Models, Least Squares, PCA, Stochastic Least Squares, Linear Prediction
4. **RL for Stochastic Optimization**
   - RL for Discrete Stochastic Optimization: Multiarmed Bandits
   - *Primer on Continuous Unconstrained Optimization*
   - Averaging Theory
   - RL for Continuous Stochastic Optimization: Stochastic Gradient Algorithms
   - Constrained Optimization

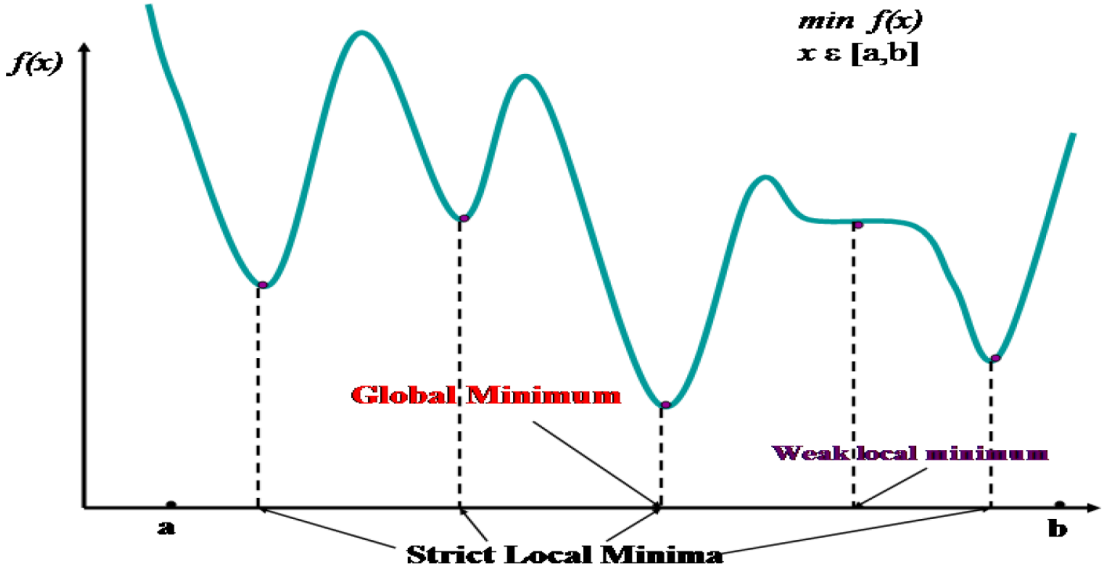
Unconstrained Optimization

$$\text{Compute } x^* = \operatorname*{argmin}_{x \in \mathbb{R}^d} F(x)$$

Result: Suppose x^* is a stationary point: i.e., it satisfies first order necessary condition $\nabla_x F(x^*) = 0$. Then x^* is a

1. local min if Hessian matrix $\nabla^2_{xx} F(x^*)$ is positive definite.
2. local max if $\nabla^2_{xx} F(x^*)$ is negative definite.
3. saddle point if Hessian has positive & negative eigenvalues.
4. otherwise inconclusive

Example 1: $F(x) = (x-2)^2$. Then $\nabla_x F(x) = 0 \implies x^* = 2$.
$\nabla^2_{xx} F(2) = 2$; so local minimum.



Example 2: $F(x_1, x_2) = (x_1 + x_2)(x_1 x_2 + x_1 x_2^2)$
$\nabla_x F(x) = 0$: stationary points $(0,0)$, $(0,-1)$, $(1,-1)$, $(\frac{3}{8}, -\frac{3}{4})$.
Let $D = \det(\nabla^2 F)$. $D(0,0) = 0$ inconclusive.
$(0,1), (1-1)$ are saddle points. $(3/8 - 3/4)$ is local max.

Numerical Algorithms for Unconstrained Optimization

Why? (i) Solving $\nabla_x F(x^*) = 0$ in closed form is usually impossible. Need a numerical algorithm.
(ii) Stochastic Opt Alg $\xrightarrow[\text{theory}]{\text{averaging}}$ Deterministic Opt Alg

Motivation: Solve algebraic equation numerically:

$$\text{Solve } \; Q(x) = [q_1(x), q_2(x), \ldots, q_d(x)]' = 0_d \text{ for } x \in \mathbb{R}^d$$

Algorithm 1: $x^{(n+1)} = x^{(n)} - \epsilon_n Q(x^{(n)})$
Algorithm 2 (Newton): $x^{(n+1)} = x^{(n)} - [\nabla Q(x^{(n)})]^{-1} Q(x^{(n)})$
Both algorithms converge to a solution (soln may not be unique); and interested in real-valued solns.

Necessary condition for minimum: $\nabla F(x) = 0$.
So set $Q(x) = \nabla F(x)$ above. Yields
Algorithm 1 (Gradient algorithm):

$$\boxed{x^{(n+1)} = x^{(n)} - \epsilon_n \nabla F(x^{(n)})}$$

Algorithm 2 (Newton):

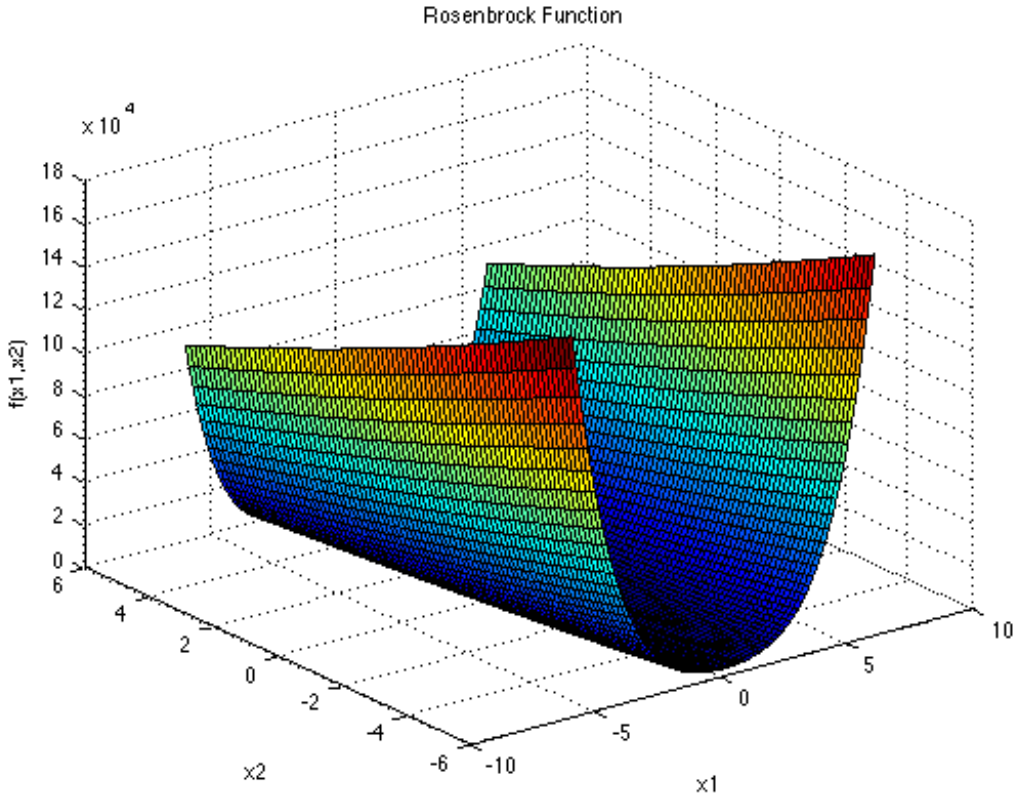
$$\boxed{x^{(n+1)} = x^{(n)} - [\nabla^2 F(x^{(n)})]^{-1} \nabla F(x^{(n)})}$$

$\nabla^2 F$ is called the Hessian matrix ($d \times d$ matrix)

Both algorithms will converge to a local stationary point.

Benchmark Example – Rosenbrock Function.

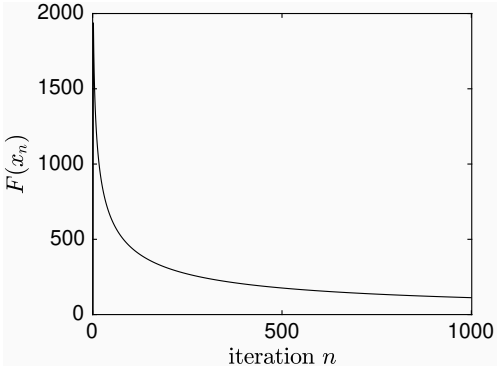
$\min_{x\in\mathbb{R}^2} F(x_1,x_2) = 100\big(x_2 - x_1^2\big)^2 + \big(x_1-1\big)^2.$



Although minimum $\big(x_1,x_2\big) = (1,1)$ is known, numerical algorithms can struggle to compute minimum for this function.

$$\nabla_x F(x) = \begin{bmatrix} -200\big(x_2 - x_1^2\big) \times 2x_1 + 2\big(x_1-1\big) \\ 200\big(x_2 - x_1^2\big) \end{bmatrix}$$

Gradient alg: $x^{(n+1)} = x^{(n)} - \epsilon_n \nabla_x F(x^{(n)})$
with $\epsilon_n = 10^{-4}/n$, $x_0 = [5,5]'$.
Convergence is very slow.
Homework: Simulate Newton Raphson.



Algorithms for Unconstrained Optimization

Consider unconstrained optimization problem: $\min_{x\in\mathbb{R}^d} F(x)$

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

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

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

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

Remarks:

1. Intuition: Numerically solve *algebraic eqn* for $x \in \mathbb{R}^d$:

$$\text{Solve } \; Q(x) = [q_1(x), q_2(x), \ldots, q_d(x)]' = 0_d$$

Algorithm 1: $x_{n+1} = x_n - \epsilon_n Q(x_n)$ will converge to a soln.
Algorithm 2 (Newton): $x_{n+1} = x_n - [\nabla Q(x_n)]^{-1} Q(x_n)$ will converge to a soln.
For optimization, $\nabla F(x) = 0$. So set $Q(x) = \nabla F(x)$ above.

2. Gradient algorithm converges with linear order:
As $n \to \infty$, $\|x_{n+1} - x^*\| < \alpha \|x_n - x^*\|$ where $\alpha < 1$.
3. Newton Raphson converges with quadratic order:
As $n \to \infty$, $\|x_{n+1} - x^*\| < \beta \|x_n - x^*\|^2$ for some scalar β.
4. In stochastic optimization, computing and inverting Hessian is difficult; so gradient algorithms are mainly used.

Proof of Gradient Algorithm

Result: Consider problem $\min_{x\in\mathbb{R}^d} F(x)$ and algorithm

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

Assume $F(x)$ is bounded from below and $|\nabla^2 F(x)| \leq M$. Then $F(x_n)$ converges and $\nabla F(x_n) \to 0$.

Proof: From Taylor series expansion

$$F(x_{n+1}) = F(x_n) + (x_{n+1} - x_n)\nabla F(x_n) + \frac{1}{2}(x_{n+1} - x_n)^2 \underbrace{\nabla^2 F(\bar{x})}_{\leq M}$$

where $\bar{x} = \alpha x_n + (1-\alpha)x_{n+1}$ for some $\alpha \in [0,1]$. Substituting $x_{n+1} = x_n - \epsilon_n \nabla F(x_n)$,

$$F(x_{n+1}) \leq F(x_n) - \left[\epsilon_n - \frac{\epsilon_n^2}{2} M\right] \left[\nabla F(x_n)\right]^2$$

Since $\epsilon_n \to 0$, for sufficiently large N, $F(x_{n+1}) \leq F(x_n)$ for all $n > N$. So algorithm yields a monotonic decreasing sequence. Any bounded monotone sequence converges (basic calculus). Next, we show that as $n \to \infty$, $\nabla F(x_n) \to 0$. Note

$$F(x_{n+1}) \leq F(x_0) - \sum_{k=1}^{n} \left[\epsilon_k - \frac{\epsilon_k^2}{2} M\right] \left[\nabla F(x_k)\right]^2$$

$\sum_k \epsilon_k = \infty \implies \sum_{k=1}^n \left[\epsilon_k - \frac{\epsilon_k^2}{2} M\right] \to \infty$. So $\left[\nabla F(x_k)\right]^2 \to 0$.

Note: $\sum_k a_k b_k$ bounded and $\sum_k a_k = \infty$ implies $b_k \to 0$.

Convex Functions

Definition (i) $f : \mathbb{R}^d \to \mathbb{R}$ is convex if for all $x_1, x_2 \in \mathbb{R}^d$,

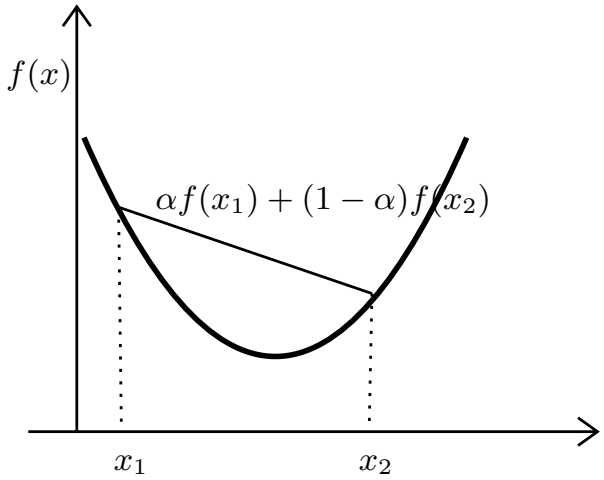
$$f(\alpha x_1 + (1-\alpha)x_2) \leq \alpha f(x_1) + (1-\alpha) f(x_2) \text{ for } \alpha \in [0,1]$$

(ii) Differentiable function f is convex if

$$f(y) \geq f(x) + (y-x)' \nabla f(x), \quad \text{for all } x, y \in \mathbb{R}^d$$

(iii) Twice differentiable function f is convex if Hessian matrix $\nabla^2 f(x)$ is positive semidefinite for all $x \in \mathbb{R}^d$.

Examples of convex functions: Ax, e^x, $-\log(x)$, $x'Qx$ where Q is positive definite, etc.



Result: For a convex function, the stationary point $x^* = \{x : \nabla f(x) = 0\}$ is the global minimum.

Proof: $f(y) \geq f(x) + (y-x)' \nabla f(x)$ for all x, y. At stationary point $x = x^*$, $\nabla f(x^*) = 0$. So $f(y) \geq f(x^*)$ for all y.

Therefore for a convex function, gradient descent and Newton Raphson converge to the global minimum.

Stochastic Gradient Algorithms for Stochastic Optimization

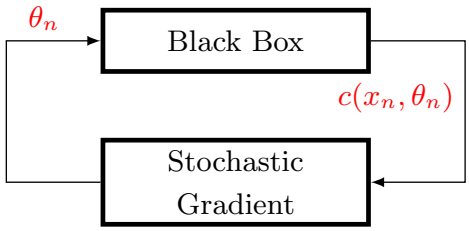
Stochastic Optimization Problem 1.
Compute $\text{argmin}_\theta \mathbb{E}\{c(x_n, \theta)\}$. Assume
(i) pdf of x_n is not known and does not depend on θ.
(ii) We know cost function $c(x, \theta)$.

If pdf of x_n is known then deterministic optimization problem: $\text{argmin}_\theta C(\theta) = \int c(x, \theta)\, p(x) dx$. Gradient algorithm is

$$\theta^{(I+1)} = \theta^{(I)} - \epsilon_I \nabla_\theta C(\theta)|_{\theta=\theta^{(I)}}$$

Stochastic Gradient Algorithm.

$$\boxed{\theta_{n+1} = \theta_n - \epsilon_n \nabla_\theta c(x_n, \theta_n)}$$



Step size: either $\epsilon_n = 1/n$ (strong convg) or positive constant (weak convg).

Example LMS algorithm. Min $\mathbb{E}\{c(x_n, \theta)\} = \mathbb{E}\{(x_n - \psi_n' \theta)^2\}$.

$$\theta_{n+1} = \theta_n + \epsilon \psi_n (x_n - \psi_n' \theta)$$

Convergence of stochastic gradient. Use Averaging Theory

Suppose a system has two-time scales: fast and slow. Assume the fast system is ergodic (satisfies law of large numbers). Then on the slow time scale, the fast system can be approximated by its average (expected value).

Convg of Stochastic Gradient using Averaging Theory

Step 1: Averaged System: For small step size ϵ, θ_k is a slow signal. x_k is fast signal. Averaged system is:

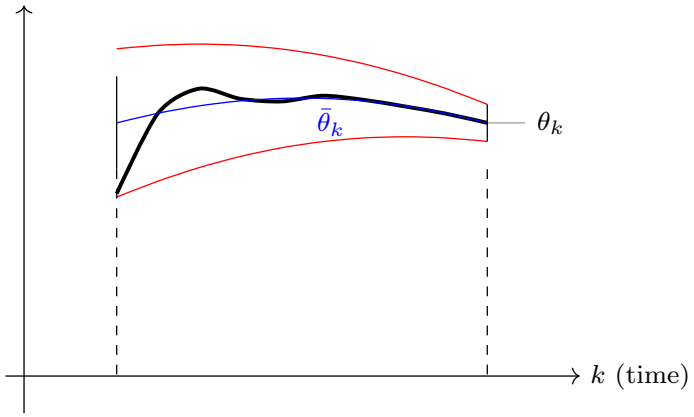
$$\begin{aligned}\bar{\theta}_{k+1} &= \bar{\theta}_k - \epsilon_k \mathbb{E}\{\nabla_\theta c(x_k, \theta_k)\} \\ &= \bar{\theta}_k - \epsilon_k \int \nabla_\theta c(x_k, \theta_k) p(x_k) dx_k \\ &= \bar{\theta}_k - \epsilon_k \nabla_\theta \mathbb{E}\{c(x_k, \theta)\} = \bar{\theta}_k - \epsilon_k \nabla_\theta C(\theta)|_{\theta_k}\end{aligned}$$

So averaged stochastic gradient alg = deterministic gradient alg.

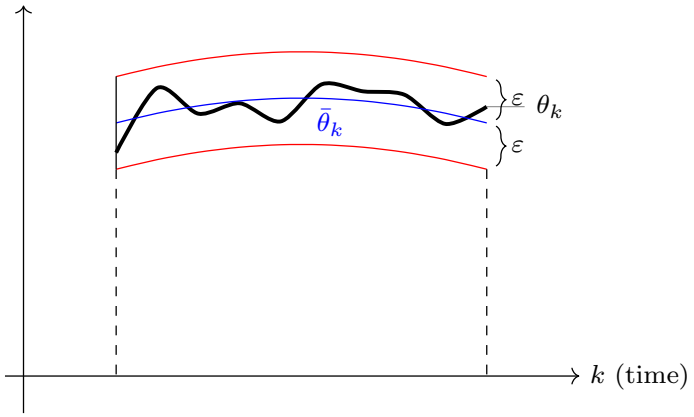
Step 2: Stochastic System converges to Averaged System:

1. Decreasing step size algorithm $\epsilon_k = 1/k$. As $k \to \infty$, $\theta_k \to \bar{\theta}_k$ strongly (wp1)

2. Constant step size algorithm ϵ constant. As $k \to \infty$, $\theta_k \to \bar{\theta}_k$ weakly (in distribution)



Decreasing step size algorithm



Constant step size algorithm

Step 3: Limit of stochastic algorithm = limit of averaged system.

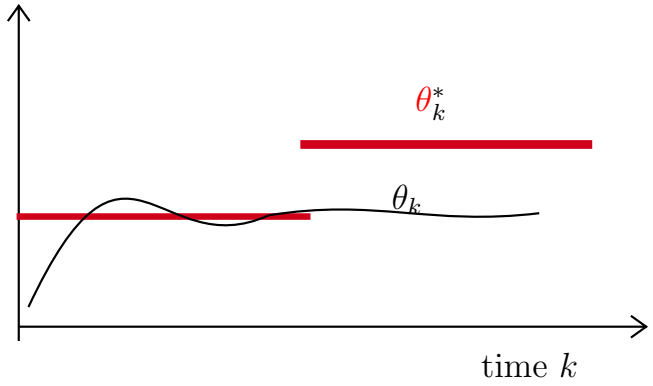
Averaged system converges to $\nabla_\theta \mathbb{E}\{c(x_k, \bar{\theta}_k)\} = 0$ as $k \to \infty$ (see deterministic case discussed earlier).

Therefore θ_k converges to this solution (strongly or weakly).

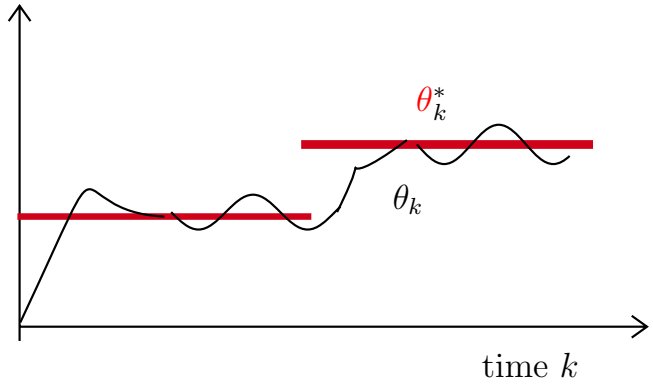
Averaging analysis involves sophisticated math. But intuition is clear

Tracking Algorithms: Constant Step Size

In most ECE and many machine learning applications the optimal value θ_* itself evolves over time (the communication channel changes, the environment changes, etc).



Decreasing step size algorithm cannot track time varying optimum θ_k^*



Constant step size algorithm can track time varying optimum θ_k^*.

$$\theta_{k+1} = \theta_k - \epsilon_k \hat{\nabla} C_k(\theta_k)$$

1. If step size $\epsilon_k = \epsilon$ is a small positive constant, then estimates θ_k bounce around but algorithm tracks time varying optimum (weak convergence)
2. If $\epsilon_k = 1/k$, then estimates θ_k converge to a constant (wp1 convergence)

Constant step size algorithm = unsupervised learning in non-stationary environment

Examples of Stoch Gradient Algorithm

Example 1. Least Mean Square (LMS) algorithm:
Adaptive filtering algorithm used widely in signal processing. Echo cancellation, channel equalization, neural networks, etc.
Aim: Minimize

$$C(\theta) = \mathbb{E}_\pi\{(y_k - \psi_k'\theta)^2\}$$

where y_k and ψ_k are observed at each time k. The LMS is a constant step size stochastic gradient algorithm:

$$\boxed{\theta_{k+1} = \theta_k + \epsilon\psi_k(y_k - \psi_k'\theta_k)}$$

Note $c(x,\theta) = (y_k - \psi_k'\theta)^2$ is known explicitly as a function of $x = (y,\psi)$ and θ. Also pdf π does not depend explicitly on θ.

Averaging Analysis: Observations generated by true model

$$y_k = \psi_k'\theta^o + v_k, \quad v_k \text{ iid }, \ \mathbb{E}\{v_k\} = 0$$

Then averaged system is the deterministic system

$$\bar\theta_{k+1} = \bar\theta_k + \epsilon\mathbb{E}\{\psi_k y_k\} - \epsilon\mathbb{E}\{\psi_k\psi_k'\}\bar\theta_k$$

Suppose $\mathbb{E}\{\psi_k\psi_k'\} = R$ positive definite matrix. Then

$$\bar\theta_{k+1} = \bar\theta_k - \epsilon R(\bar\theta_k - \theta^o)$$

This deterministic algorithm minimizes convex objective $(\theta - \theta^o)'R(\theta - \theta^o)$. So $\bar\theta_k$ converges to true model θ^0.
Therefore θ_k (LMS algorithm) converges weakly to true model.

Example 2. Logistic Regression with Supervised Learning

Linear regression $y_k = \psi_k'\theta + e_k$. Not useful model when $y_k \in \{0,1\}$
Logistic regression model: Generate $y_k \in \{0,1\}$ as

$$P(y=1|\theta) = \sigma(\psi'\theta) = \frac{1}{1+\exp(-\psi'\theta)}$$

where $\psi_k \in \mathbb{R}^n$ is input vector at time n, $\theta \in \mathbb{R}^n$,
$\sigma(\cdot)$: S-shaped sigmoidal (logistic function) with range $[0,1]$.

Aim. Given N training samples $(\psi_k, y_k), k = 1,\ldots,N$, compute maximum likelihood estimate θ^*.

$$\max_\theta L(\theta) = \log P(y_1,\ldots,y_N|\theta) = \sum_{k=1}^N \log P(y_k|\theta)$$

$$= \sum_{k=1}^N \log P(y_k=1|\theta)I(y_k=1) + \log P(y_k=0|\theta)I(y_k=0)$$

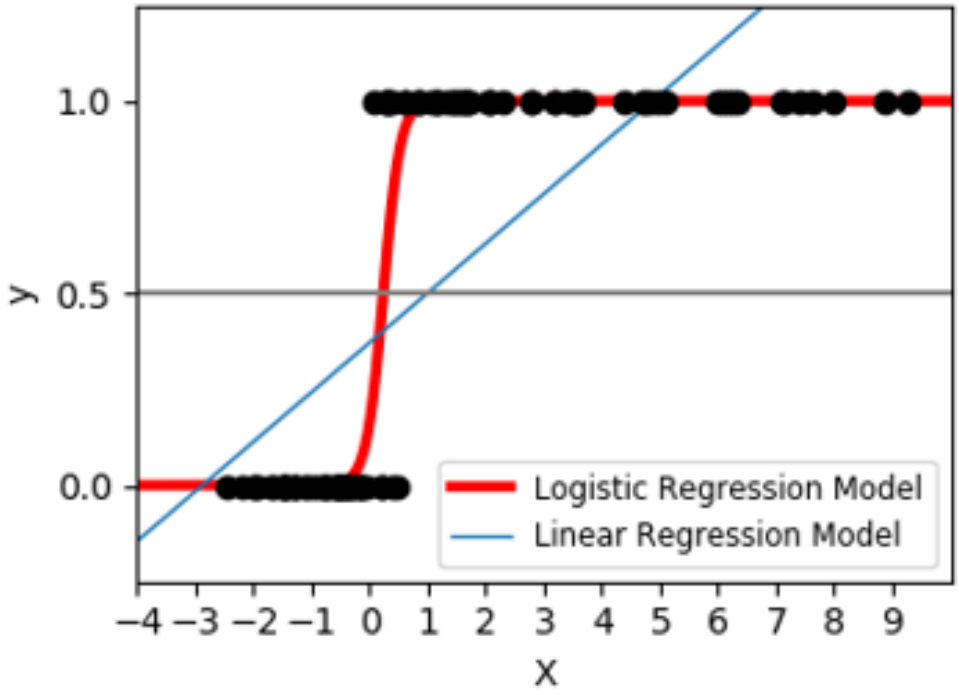
$$= \sum_{k=1}^N y_k \log\sigma(\psi_k'\theta) + (1-y_k)\log\big(1-\sigma(\psi_k'\theta)\big)$$

Off line gradient algorithm to compute MLE is:

$$\theta^{(I+1)} = \theta^{(I)} + \epsilon_k \nabla_\theta L(\theta)|_{\theta^{(I)}} = \theta^{(I)} + \epsilon_k \sum_{k=1}^N \big(\sigma(\psi_k'\theta) - y_k\big)\psi_k$$

Stochastic gradient algorithm is:

$$\theta_{k+1} = \theta_k + \epsilon_k\big(\sigma(\psi_k'\theta) - y_k\big)\psi_k$$



Homework: Apply logistic regression to classify images of digits

`http://ufldl.stanford.edu/tutorial/supervised/LogisticRegression/`

MNIST dataset

Stochastic Optimization Problem 2

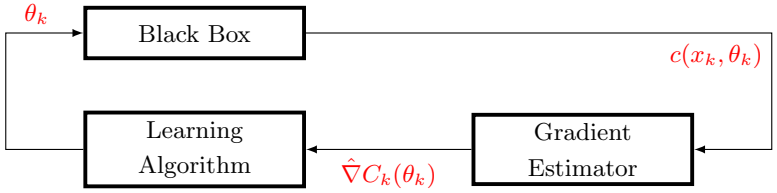
Aim. Compute $\theta^* = \text{argmin}_\theta C(\theta) = \mathbb{E}_{\pi_\theta}\{c(x_k, \theta)\}$ assuming:
(i) pdf of x_n is not known but depends on θ.
(ii) cost function $c(x, \theta)$ not known but we can observe random sample $c(x_k, \theta)$ where $x_k \sim \pi_\theta$ for any choice of θ.
Stochastic Gradient Algorithm: If $\hat{\nabla} C_k(\theta_k)$ is unbiased estimate of $\nabla C(\theta_k)$ then for step size ϵ_k

$$\theta_{k+1} = \theta_k - \epsilon_k \hat{\nabla} C_k(\theta_k)$$

How to compute unbiased gradient estimate $\hat{\nabla} C_k(\theta_k)$ *?* Note

$$\nabla_\theta C(\theta) = \nabla_\theta \mathbb{E}_{\pi_\theta}\{c(x, \theta)\} \neq \int_{\mathcal{X}} \nabla_\theta c(x, \theta) \pi_\theta(x) dx.$$

Also we dont know $c(x, \theta)$ explicitly. Unlike problem 1, need additional step: stochastic simulation to estimate gradient.
(i) Finite difference Estimator
(ii) Score Function Estimator (iii) Weak derivative Estimator.



Example: RL for MDP

Convergence. Assuming $\hat{\nabla} C_k(\theta_k)$ is an unbiased gradient estimate then using averaging theory, averaged system is:

$$\bar{\theta}_{k+1} = \bar{\theta}_k - \epsilon_k \nabla C(\bar{\theta}_k)$$

This deterministic system converges to local minimum (or global minimum for convex $C(\theta)$). So gradient algorithm converges.