Biomedical Imaging & Analysis

Lecture 1, Fall 2014

Part 2:

Review of Basic Math + Linear Operations

*[**Text**: Ch. 1-2 of Machine Vision by Wesley E. Snyder & Hairong Qi]*

Prahlad G Menon, PhD

Assistant Professor

Sun Yat-sen University – Carnegie Mellon University (SYSU-CMU)

Joint Institute of Engineering

Linear Operations

- **What is a linear operation..?** How to represent it as a Matrix operator..?
- Rules for basic matrix multiplication.
- Matrix inverse & unitary matrix.
- What is linear independence..? What is a basis..?
- What is an orthonormal basis set..? Why is it useful..?
- What is function minimization..? Why is it useful..?
- Examples of useful linear operations / operators.
- What is an inverse linear transform..? Example.

Linear Operators

- Wolfram – definition of a linear operator / transformation:

A linear transformation between two vector spaces V and W is a map $T : V \to W$ such that

1. $T(\mathbf{v}_1 + \mathbf{v}_2) = T(\mathbf{v}_1) + T(\mathbf{v}_2)$ for any vectors $\mathbf{v}_1$ and $\mathbf{v}_2$ in V, and
2. $T(\alpha\, \mathbf{v}) = \alpha\, T(\mathbf{v})$ for any scalar α.

- Most image processing operations can be modeled as linear operators.

- Imaging systems can be modeled as linear operators.

Linear algebra

$$\mathbf{v} = [x_1 \ x_2 \ x_3]^{\mathrm{T}} \qquad \mathbf{a}^{\mathrm{T}}\mathbf{b} = \sum_i a_i b_i \qquad |\mathbf{x}| = \sqrt{\mathbf{x}^{\mathrm{T}}\mathbf{x}}$$

- Unit vector: $|\boldsymbol{x}| = 1$
- Orthogonal vectors: $\boldsymbol{x}^{\mathrm{T}}\boldsymbol{y} = 0$

- Orthogonal square matrices: $\boldsymbol{A}^{\mathrm{T}} = \boldsymbol{A}^{-1}$

- Orthonormal: orthogonal unit vectors
- Inner product of continuous functions

$$\langle f(x), g(x) \rangle = \int_a^b f(x) g(x) \, dx$$

  – Orthogonality & orthonormality apply here too

Vector Norms

Definition

The norm of a vector is defined from the inner product as $\|\mathbf{x}\| = \sqrt{\langle \mathbf{x}, \mathbf{x} \rangle}$.

Some useful properties

- Cauchy-Schwarz inequality $|\langle \mathbf{x}, \mathbf{y} \rangle| \leq \|\mathbf{x}\| \|\mathbf{y}\|$.
- Triangle inequality $\|\mathbf{x} + \mathbf{y}\| \leq \|\mathbf{x}\| + \|\mathbf{y}\|$

Linear Operations

- **On the board / derivation:**
  - **Basic definition of a fitting problem**
    - **Requirement of linear independence.**
  - **Steepest descent for solving a fitting problem**
    - **When is this likely to give the right answer..?**
    - **Definition of Positive Definite matrix.**
  - **Fourier decomposition and terminology.**
  - **What is square integrable basis..?**

Linear independence

- No one vector is a linear combination of the others
  - $\boldsymbol{x}_j \neq \Sigma a_i \boldsymbol{x}_i$ for any a_i across all $i \neq j$
- Any linearly independent set of d vectors $\{\boldsymbol{x}_{i=1\ldots d}\}$ is a basis set that spans the space $\Re^d$
  - Any other vector in $\Re^d$ may be written as a linear combination of $\{\boldsymbol{x}_i\}$
- Often convenient to use orthonormal basis sets
- Projection: if $\boldsymbol{y}=\boldsymbol{\Sigma} a_i \boldsymbol{x}_i$ then $a_i=\boldsymbol{y}^T\boldsymbol{x}_i$

Eigen values & Eigen vectors

Let **A** be an n x n matrix.

The number is an eigenvalue of **A** if there exists a non-zero vector **v** such that

$$\mathbf{Av} = \lambda\mathbf{v} \ \textbf{(}\text{The Characteristic equation}\textbf{)}$$

In this case, vector **v** is called an eigen-vector of **A** corresponding to λ.

What does this mean physically..?

A maps **v** onto itself with only a change in length.

Derivatives

- Of a scalar function of $\boldsymbol{x}$:
  - Called the gradient
  - Really important!

$$\frac{df}{d\mathbf{x}} = \left[\frac{\partial f}{\partial x_1} \; \frac{\partial f}{\partial x_2} \cdots \frac{\partial f}{\partial x_d} \right]^{\mathrm{T}}$$

- Of a vector function of $\boldsymbol{x}$
  - Called the Jacobian

$$\frac{d\boldsymbol{f}}{d\mathbf{x}} = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \cdots & \frac{\partial f_1}{\partial x_d} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_m}{\partial x_1} & \cdots & \frac{\partial f_m}{\partial x_d} \end{bmatrix}$$

- Hessian = matrix of 2nd derivatives of a scalar function

$$\begin{bmatrix} \frac{\partial^2 f}{\partial x_1^2} & \frac{\partial^2 f}{\partial x_1 \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_1 \partial x_d} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 f}{\partial x_d \partial x_1} & \frac{\partial^2 f}{\partial x_d \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_d^2} \end{bmatrix}$$

Least Squares Fitting
– solving as a linear system

Problem setup

Let us solve $\mathbf{Ax} = \mathbf{b}$ where $\mathbf{A}$ is $M \times N$, with $M > N$. No exact solution may be available.

One solution

Left multiply by $\mathbf{A}^T$. Then $\mathbf{A}^T\mathbf{A}$ will then be square, and the system $\mathbf{A}^T\mathbf{Ax} = \mathbf{A}^T\mathbf{b}$ can be solved. Or, using the inverse notation, $\mathbf{x} = (\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T\mathbf{b}$.

Optimal in least squares sense

$\mathbf{x} = (\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T\mathbf{b}$ solves the following optimization problem

$$\min_{\mathbf{x}} ||\mathbf{Ax} - \mathbf{b}||^2$$

What kind of basis is ideal for A , here..?

Function **Minimization**

- Find the vector $\mathbf{x}$ which produces a minimum of some function $f(\mathbf{x})$
  - $\mathbf{x}$ is a parameter vector
  - $f(\mathbf{x})$ is a scalar function of $\mathbf{x}$
    - The "objective function"
- The minimum value of f is denoted:

$$\overset{\frown}{f}(\mathbf{x}) = \min_{\mathbf{x}} f(\mathbf{x})$$

- The minimizing value of $\mathbf{x}$ is denoted:

$$\hat{\mathbf{x}} = \operatorname{argmin}_{\mathbf{x}} f(\mathbf{x})$$

Useful formulations
– applied to Optimization

- For a matrix, A
- Quadratic form:

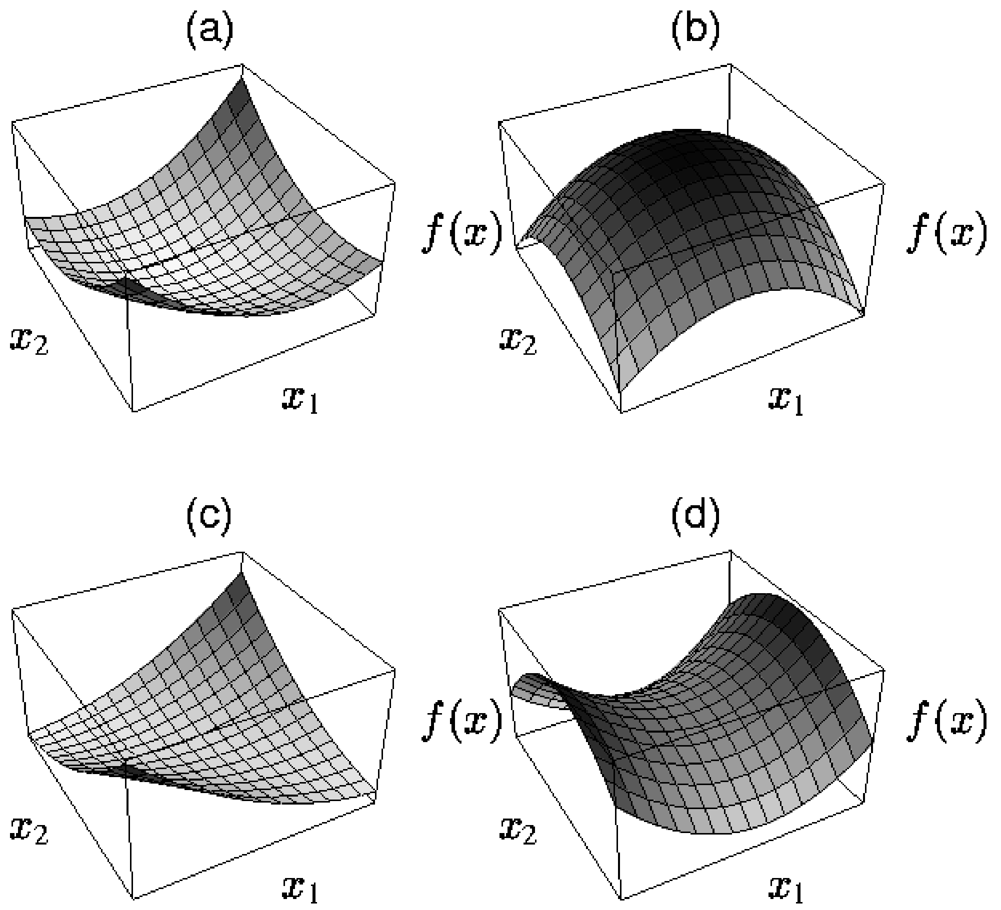
$$\mathbf{x}^{\mathrm{T}} A \mathbf{x}$$

$$\frac{d}{d\mathbf{x}}\left(\mathbf{x}^{\mathrm{T}} A \mathbf{x}\right) = \left(A + A^{\mathrm{T}}\right)\mathbf{x}$$

- Positive definite: Always has a global minima; good for optimization
  - – Applies to A if

$$\mathbf{x}^{\mathrm{T}} A \mathbf{x} > 0 \;\; \forall \mathbf{x} \in \Re^{d}, \mathbf{x} \neq 0$$

Visualizing Positive Definiteness…



a) Positive definite matrix b) negative-definite matrix
c) Singular matrix d) positive indefinite matrix

Numerical minimization (iterative) techniques…

- Gradient descent:
  - The derivative points away from the minimum
  - Take small steps, each one in the “down-hill” direction
- Local vs. global optimization…
- Combinatorial optimization (Global search):
  - Simulated annealing (SA eg: D. Piao et al. 2014)
  - At each step, the SA heuristic considers some neighbouring state *s′* of the current state *s*, andprobabilistically decides between moving the system to state *s′* or staying in state *s*. The neighbours of a state are produced after altering a given state in some well-defined way.
  - Mean field annealing - deterministic approximation to SA.

Iterative Methods

Stationary:

$$x^{(k+1)} = Gx^{(k)} + c$$

where G and c do not depend on iteration count (k)

Non Stationary:

$$x^{(k+1)} = x^{(k)} + a_k p^{(k)}$$

where computation involves information that change at each iteration

Stationary: Jacobi Method

In the *i-th* equation solve for the value of x_i while assuming the other entries of *x* remain fixed:

$$\sum_{j=1}^{N} m_{ij} x_j = b_i \rightarrow x_i = \frac{b_i - \sum_{j \neq i} m_{ij} x_j}{m_{ii}} \qquad x_i^{(k)} = \frac{b_i - \sum_{j \neq i} m_{ij} x_j^{(k-1)}}{m_{ii}}$$

In matrix terms the method becomes:

$$x^{(k)} = D^{-1}(L+U)x^{(k-1)} + D^{-1}b$$

where D, -L and -U represent the diagonal, the strictly lower-trg and strictly upper-trg parts of M

M=D-L-U

Stationary-Gause-Seidel

Like Jacobi, but now assume that previously computed results are used as soon as they are available:

$$\sum_{j=1}^{N} m_{ij} x_j = b_i \rightarrow x_i = \frac{b_i - \sum_{j \neq i} m_{ij} x_j}{m_{ii}} \quad \Rightarrow \quad x_i^{(k)} = \frac{b_i - \sum_{j<i} m_{ij} x_j^{(k)} - \sum_{j>i} m_{ij} x_j^{(k-1)}}{m_{ii}}$$

In matrix terms the method becomes:

$$x^{(k)} = \left(D - L\right)^{-1} \left(U x^{(k-1)} + b\right)$$

where D, -L and -U represent the diagonal, the strictly lower-trg and strictly upper-trg parts of M

M=D-L-U

Stationary: Successive Overrelaxation (SOR)

Devised by extrapolation applied to Gauss-Seidel in the form of weighted average:

$$x_i^{(k)} = w\bar{x}_i^{(k)} + (1-w)x_i^{(k-1)}$$

$$x_i^{(k)} = \frac{b_i - \sum_{j<i} m_{ij} x_j^{(k)} - \sum_{j>i} m_{ij} x_j^{(k-1)}}{m_{ii}}$$

In matrix terms the method becomes:

$$x^{(k)} = \left(D - wL\right)^{-1}(wU + (1-w)D)x^{(k-1)} + w(D - wL)^{-1}b$$

where D, -L and -U represent the diagonal, the strictly lower-trg and strictly upper-trg parts of M

M=D-L-U

SOR

- Choose w to accelerate the convergence

$$x_i^{(k)} = w\overline{x}_i^{(k)} + (1-w)x_i^{(k-1)}$$

  - W =1 : Jacobi / Gauss-Seidel
  - 2>W>1: Over-Relaxation
  - W < 1: Under-Relaxation

Non-stationary Iterative Method

- State from initial guess x0, adjust it until close enough to the exact solution

$$x_{(i+1)} = x_{(i)} + a_{(i)} p_{(i)}$$ i=0,1,2,3,……

$p_{(i)}$ Adjustment Direction

$a_{(i)}$ Step Size

- How to choose direction and step size?

Steepest Descent Method (1)

- Choose the direction in which f decrease most quickly: the direction opposite of $f'(x_{(i)})$

$$-f'(x_{(i)}) = b - Ax_{(i)} = r_{(i)}$$

- Which is also the direction of residue

$$x_{(i+1)} = x_{(i)} + a_{(i)} r_{(i)}$$

Steepest Descent Method (2)

- How to choose step size ?
  - Line Search

    $a_{(i)}$ should minimize f, along the direction of $r_{(i)}$, which means $\frac{d}{da} f(x_{(i+1)}) = 0$

$$\Rightarrow a_{(i)} = \frac{r_{(i)}{}^T r_{(i)}}{r_{(i)}{}^T A r_{(i)}}$$

$$r_{(i)} = b - Ax_{(i)}$$

$$x_{(i+1)} = x_{(i)} + a_{(i)} r_{(i)}$$

Now, starting with x0, iterate until residue is smaller than error tolerance .

Examples of linear operators

- Rotation, translation, scaling – spatial transforms
- Convolution

- Wolfram – definition of a linear operator / transformation:

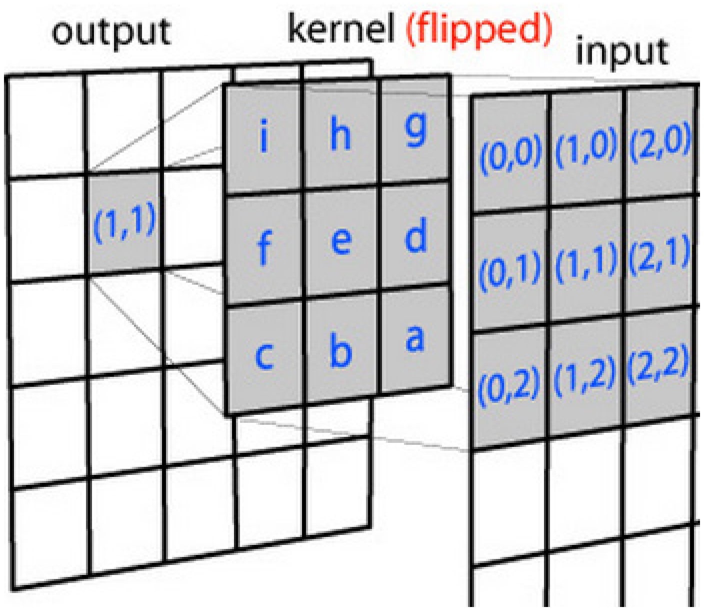
A linear transformation between two vector spaces V and W is a map $T : V \to W$ such that

1. $T(\mathbf{v}_1 + \mathbf{v}_2) = T(\mathbf{v}_1) + T(\mathbf{v}_2)$ for any vectors $\mathbf{v}_1$ and $\mathbf{v}_2$ in V, and

2. $T(\alpha\, \mathbf{v}) = \alpha\, T(\mathbf{v})$ for any scalar α.

The Convolution Operator

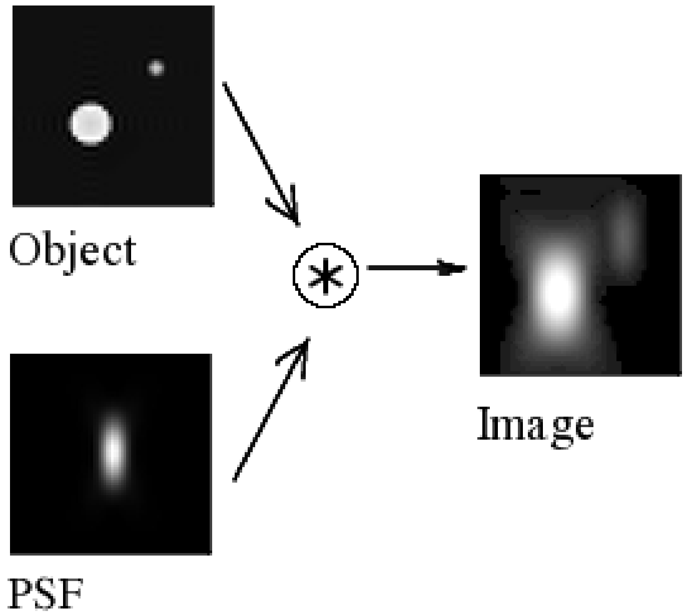
- Simple example of convolution of input image (matrix, S) and impulse response (kernel) in 2D spatial. *Y[n] = (S * h) [n].*
- Notice that the kernel matrix is flipped both horizontal and vertical direction before multiplying the overlapped input data.



$$\begin{array}{c|ccc} n \backslash m & -1 & 0 & 1 \\ \hline -1 & a & b & c \\ 0 & d & e & f \\ 1 & g & h & i \end{array}$$

$$\begin{aligned} y[1,1] &= \sum_{j=-\infty}^{\infty}\sum_{i=-\infty}^{\infty} x[i,j]\cdot h[1-i,1-j] \\ &= x[0,0]\cdot h[1,1] + x[1,0]\cdot h[0,1] + x[2,0]\cdot h[-1,1] \\ &\quad + x[0,1]\cdot h[1,0] + x[1,1]\cdot h[0,0] + x[2,1]\cdot h[-1,0] \\ &\quad + x[0,2]\cdot h[1,-1] + x[1,2]\cdot h[0,-1] + x[2,2]\cdot h[-1,-1] \end{aligned}$$

Linear **INVERSE** problems – de-blurring an image



Given measured data, **g**, and information about the system **H** and noise model, recover an estimate ~f that closely approximates f .

When H is square, solve **g = Hf** via $\mathbf{f} = \mathbf{H}^{-1}\mathbf{g}$ directly (exactly) via Fourier Transform methods.

Eg: **Lucy–Richardson** **deconvolution**, is an iterative procedure for recovering a latent image that has been blurred by a known point spread function .

Orthonormal **Expansion**

- Relate the problem of expanding these approximations to solving **Ax = b**.
- Now **x** are the expansion coefficients, b is an image, and **A** = [$\Phi 1, \Phi 2$, …], where each Φi is a column vector representing an orthonormal basis.
- When columns of A are orthonormal: **(A*)(A) = I** , so **x =** $\mathbf{A}^T\mathbf{b}$.
- $b[p] = \Sigma_k x_k A_{p,k}$, or, $s[p] = \Sigma_k c_k \Phi_k[p]$, where c_k may be termed Fourier cofficients, if $\Phi_k[p]$ refers to the Fourier basis.
- $k[n] = e^{-j2kn/N} = \cos(2kn/N) + j \sin(2kn/N)$. $<\Phi_k, \Phi_k> = N$ for Fourier.

Fourier Orthonormal Basis

- Many imaging problems easier to understand if we choose building blocks (k) carefully.
- $<\Phi_k, \Phi_k> = \Sigma_n \Phi^*_k[n] \Phi_l[n] = 0$ if $k != l$,

(1/N) **Ac = s** , is simply **c = A*s**:

- $s[n] = \frac{1}{N} \sum_{k=0}^{N-1} c[k] \phi_k[n]$ with
- $c[k] = \sum_{n=0}^{N-1} s[n] \phi_k^*[n]$

The coefficients c[k] of are the Discrete Fourier Transform of the data s[n].

Useful properties of a Fourier basis…

Convolution:

Let $$v[n] = (s * h)[n]$$

Then $\hat{v}[k] = \hat{s}[k]\hat{h}[k]$

Interesting fact:

k-space → MRI data is acquired in frequency space (or Fourier space) and then converted back to image space.

Probability

- Probability of an event a occurring:
  - $Pr(a)$
- Independence
  - $Pr(a)$ does not depend on the outcome of event b, and vice-versa
- Joint probability
  - $Pr(a,b)$ = Prob. of both a and b occurring
- Conditional probability
  - $Pr(a|b)$ = Prob. of a if we already know the outcome of event b
  - Read "probability of a given b"

Probability for continuously-valued functions

- Probability distribution function:

$$P(x) = Pr(z<x)$$

- Probability density function:

$$p(x) = \frac{d}{dx}P(x)$$

$$\int_{-\infty}^{\infty} p(x)\,dx = 1$$

Variance:

$$\begin{aligned}\operatorname{Var}(X) &= \operatorname{E}\left[(X - \operatorname{E}[X])^2\right] \\ &= \operatorname{E}\left[X^2 - 2X\operatorname{E}[X] + (\operatorname{E}[X])^2\right]\end{aligned}$$

- Expected value $\operatorname{E}[X] = \int_{-\infty}^{\infty} x f(x)\,\mathrm{d}x$

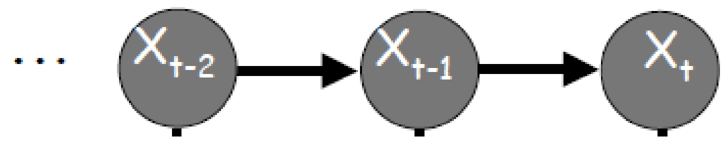
$$\operatorname{E}[X] = \frac{x_1p_1 + x_2p_2 + \cdots + x_kp_k}{1} = \frac{x_1p_1 + x_2p_2 + \cdots + x_kp_k}{p_1 + p_2 + \cdots + p_k}.$$

Markov models

- For temporal processes:
  - The probability of something happening is dependent on a thing that just recently happened.
- For spatial processes
  - The probability of something being in a certain state is dependent on the state of something nearby.
  - Example: The value of a pixel is dependent on the values of its neighboring pixels.

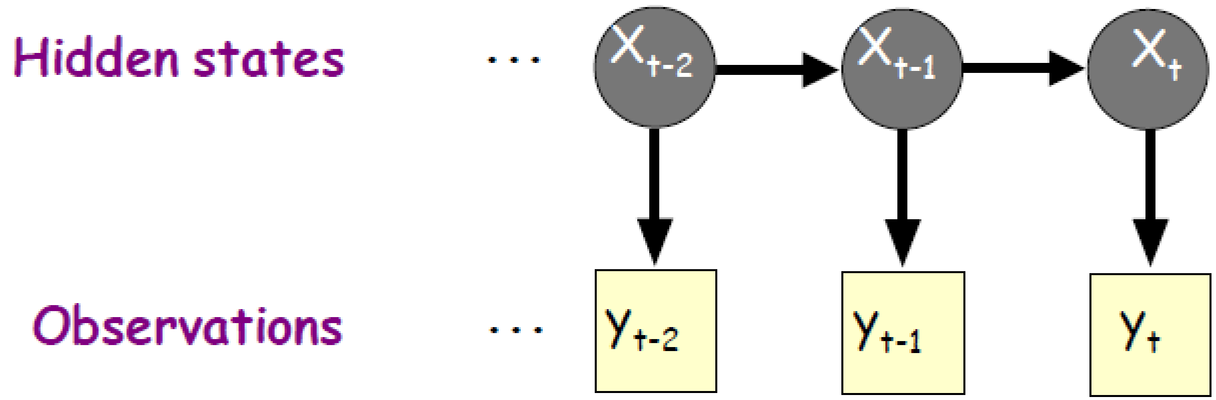
Markov chain

- Simplest Markov process – automatic random transition between finite states with determinable:
  - *Initial probabilities*
  - *Transition probabilities*



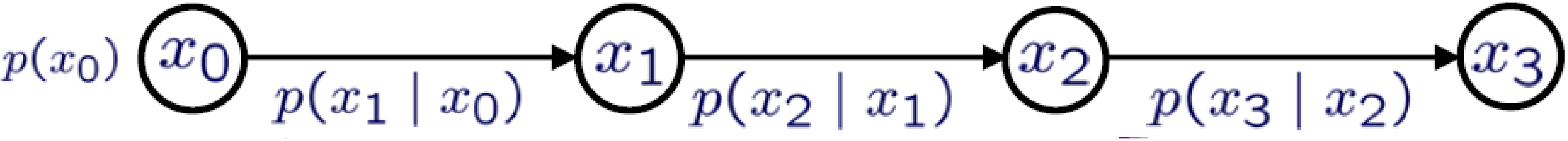
- Example: symbols transmitted one at a time
  - What is the probability that the next symbol will be w?
- For a Markov chain:
  - "The probability conditioned on all of history is identical to the probability conditioned on the last symbol received."

Hidden Markov models (HMMs)



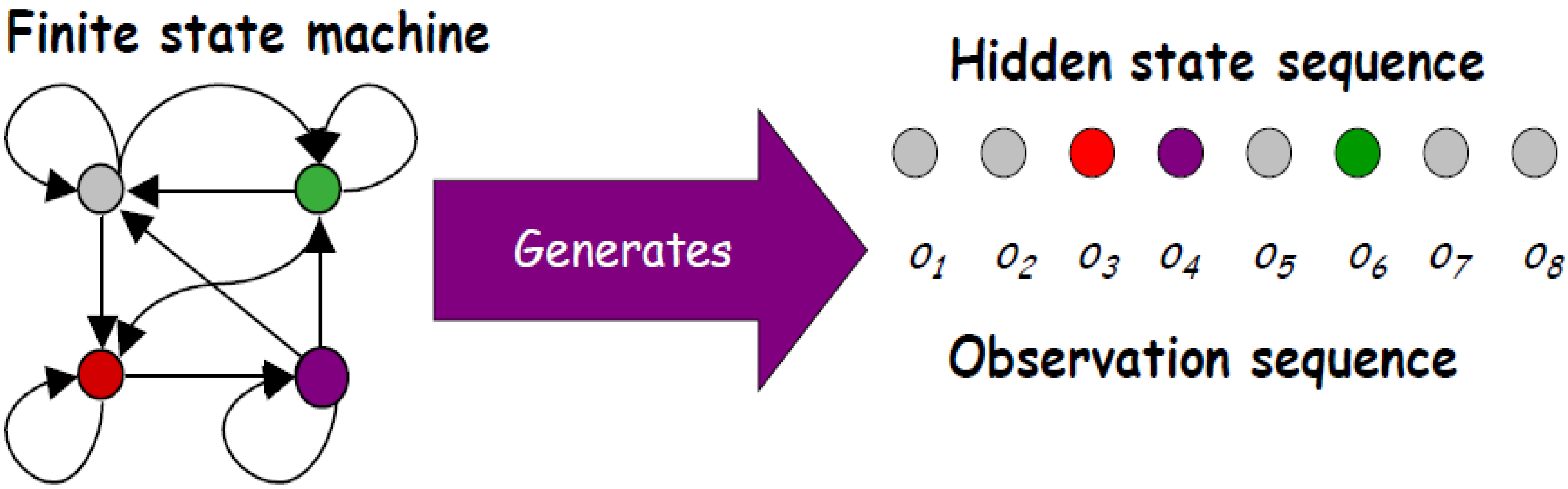
Markov Chains: Graphical Models

$$p(x_0, x_1, \ldots, x_T) = p(x_0) \prod_{t=1}^{T} p(x_t \mid x_{t-1})$$



HMM switching

- Governed by a finite state machine (FSM)



HMM generates observation sequence

The HMM Task

- Given only the output $f(t)$, determine:

1. The most likely state sequence of the switching FSM
   - Use the Viterbi algorithm
   - Computational complexity =

     (# state changes) * (# state values)^2
   - Much better than brute force, which =

     (# state values)$^{\text{(\# state changes)}}$

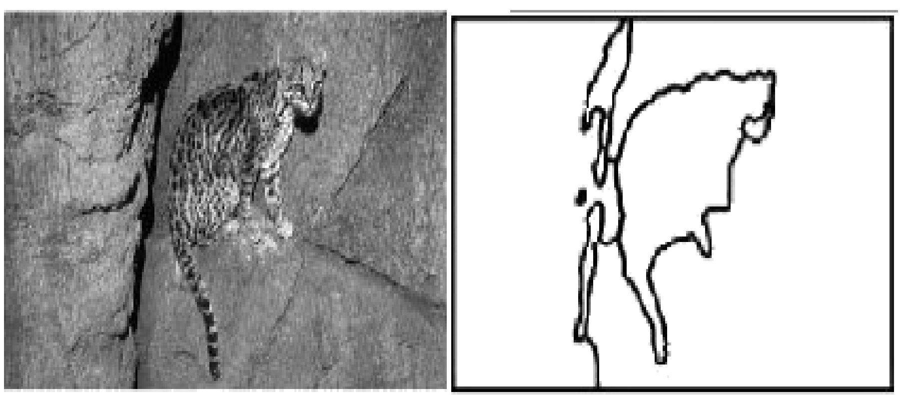
2. The parameters of each hidden Markov model
   - Use the iterative process in the book
   - Better, use someone else's debugged code that they've shared

Example: Markov models for **Image Segmentation**

IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, VOL. 24, NO. 5, MAY 2002 657

Image Segmentation by Data-Driven Markov Chain Monte Carlo

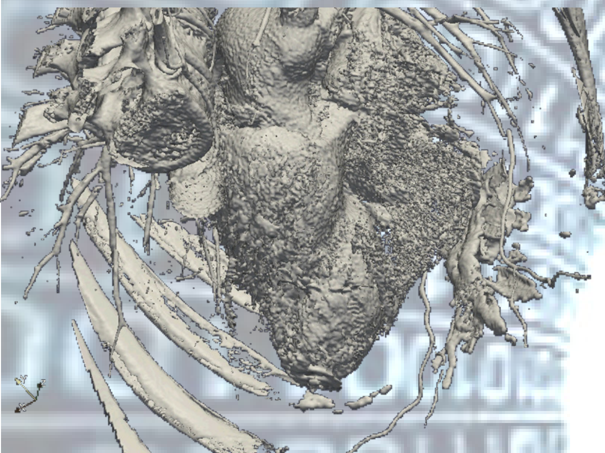
Zhuowen Tu and Song-Chun Zhu

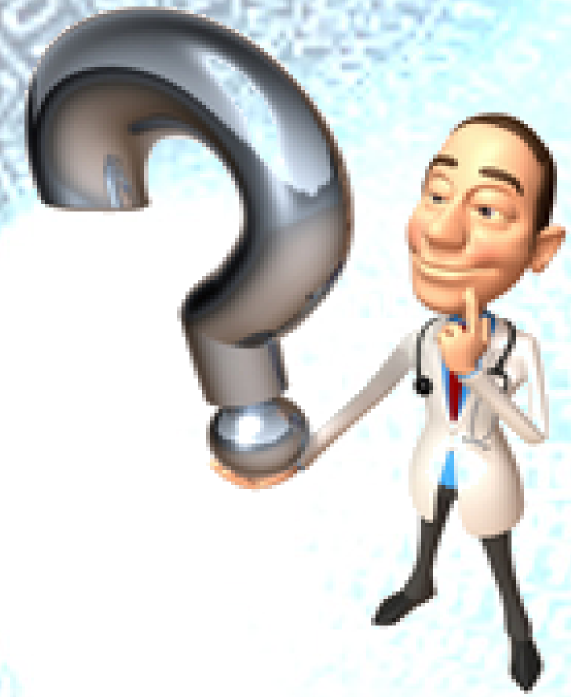


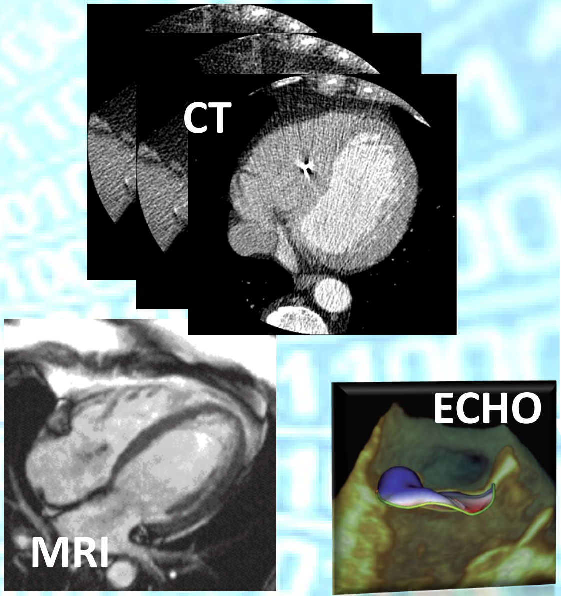
http://www.stat.ucla.edu/~sczhu/papers/DDMCMC_reprint.pdf

Code: http://pages.ucsd.edu/~ztu/Download_ddmcmc.htm

BIA 2014





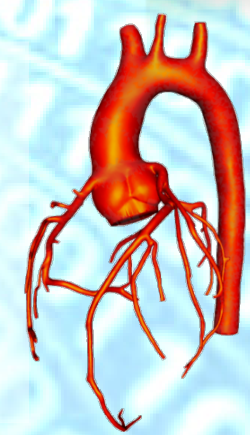


Prahlad G Menon, PhD

www.justcallharry.com

+1 412-259-3031

pgmenon@andrew.cmu.edu



What's next?

QUIZ on Ch 1-2, Snyder & Qi.

Assignment 2

Use your triangle normal calculation function from Assignment 1 (Part 2, Q2), to find the normals of a coronary artery model with known points and triangulation.

Visualize results in Matlab.