Machine Learning for Computer Vision

18 October, 2013
MVA – ENS Cachan

Lecture 6: Introduction to graphical models

Iasonas Kokkinos

Iasonas.kokkinos@ecp.fr

Center for Visual Computing
Ecole Centrale Paris

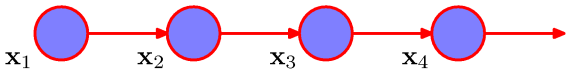
Galen Group
INRIA-Saclay

Lecture outline

- Introduction
  - Graphs and computation
  - Graphical models
- Chain-structured graphical models
- Tree-structured graphical models

Problem 1: measuring the length of a queue

while only complaining to your closest neighbors

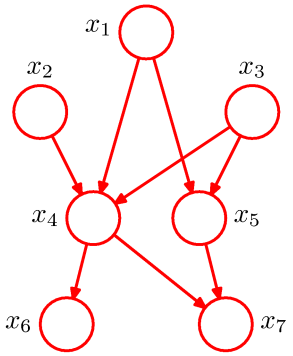


Problem 2: two queues

Problem N: N queues

Lecture outline

- Introduction
  - Computation on graphs
  - Graphical models
- Chain-structured graphical models
- Tree-structured graphical models



Graphical models

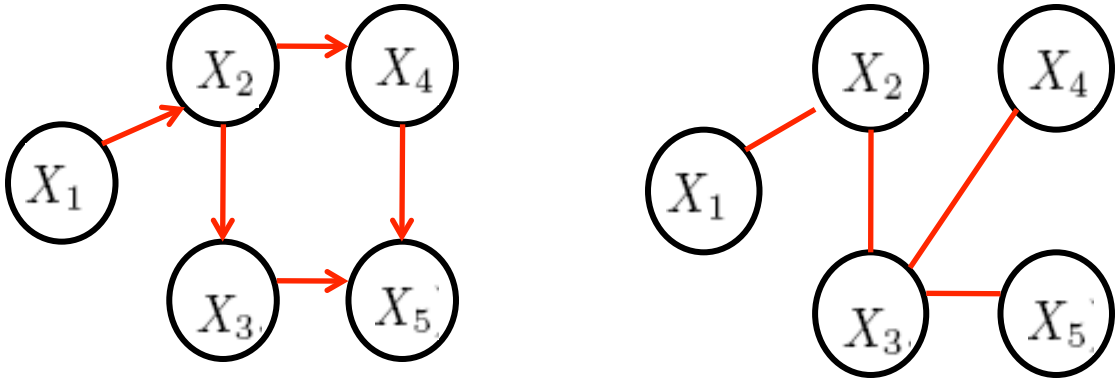
Blend of graph theory and probability

Graph nodes: random variables

Graph edges: relationships

Directed graphs: Bayesian Networks

Undirected graphs: Markov Random Fields



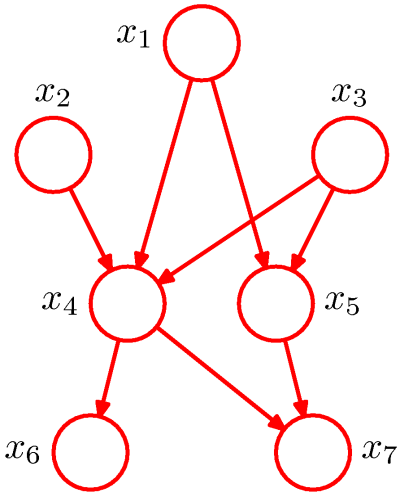
Bayesian Networks

Directed Acyclic Graph $G(V, E)$

Nodes - V : random variables

Edges - E : dependencies

Leave from `parents', arrive at children

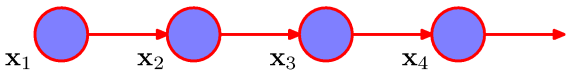


Distribution of each child conditioned on parent: $P(X_v|X_{\pi_v})$

$$P(X_v|\emptyset) = P(X_v)$$

Joint probability distribution: $P(X) = \prod_v P(X_v|X_{\pi_v})$

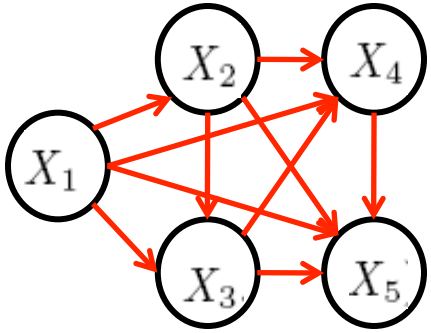
Bayesian network for Brownian motion:



Conditional independencies

We can trivially obtain a bayesian network for a probability distribution

$$P(X_1, X_2, X_3, X_4, X_5) = P(X_1)P(X_2|X_1)P(X_3|X_2, X_1)P(X_4|X_3, X_2, X_1)P(X_5|X_4, X_3, X_2, X_1)$$

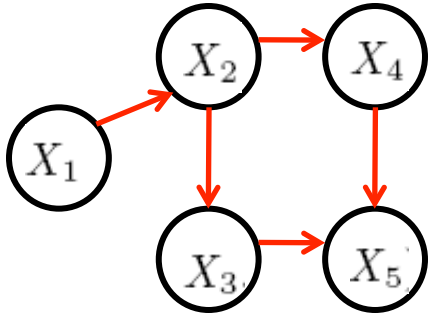


Fully connected graph!

Bayesian net

Non-trivial cases: some conditional independence

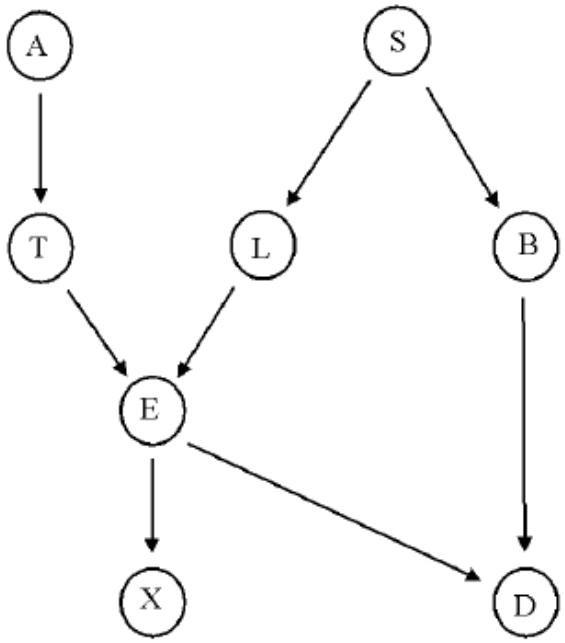
$$\begin{aligned} P(X_3|X_2, X_1) &= P(X_3|X_2) \\ P(X_4|X_3, X_2, X_1) &= P(X_4|X_2) \\ P(X_5|X_1, X_2, X_3, X_4, X_5) &= P(X_5|X_3, X_4) \end{aligned}$$



Main interest: exploit independencies for training & inference

A Bayesian Network-based Expert System

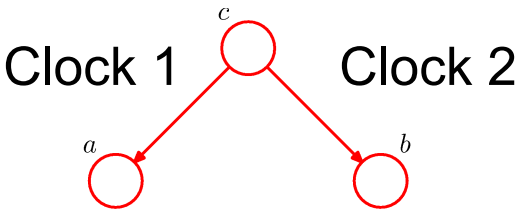
- `Asia Network'
  - A: trip to Asia
  - T: turbeculosis
  - S: smoking
  - L: Lung Cancer
  - B: Bronchitis
  - E: Turbeculosis/Lung Cancer
  - X-ray results
  - D: Dyspnea



- Given: X-rays, Dyspnea, patient went to Asia, patient smokes
- Wanted: posterior probability of Bronchitis

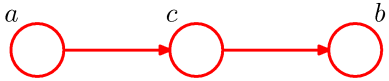
Reading Conditional Independence from a BN Graph

Big Ben



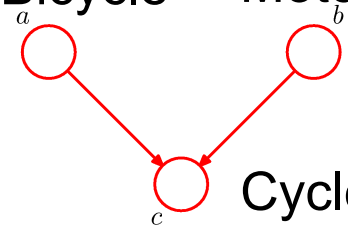
$$P(a,b) \neq P(a)P(b)$$

State 1 State 2 State 3

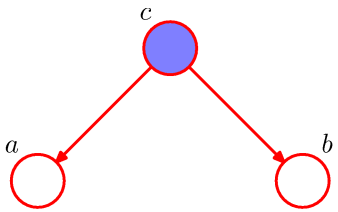


$$P(a,b) \neq P(a)P(b)$$

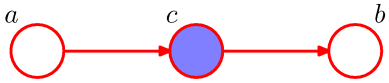
Bicycle Motorcycle



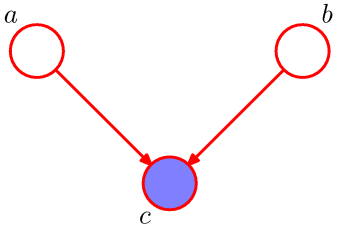
$$P(a,b) = P(a)P(b)$$



$$P(a,b|c) = P(a|c)P(b|c)$$



$$P(a,b|c) = P(a|c)P(b|c)$$



$$P(a,b|c) \neq P(a|c)P(b|c)$$

`Explaining away'

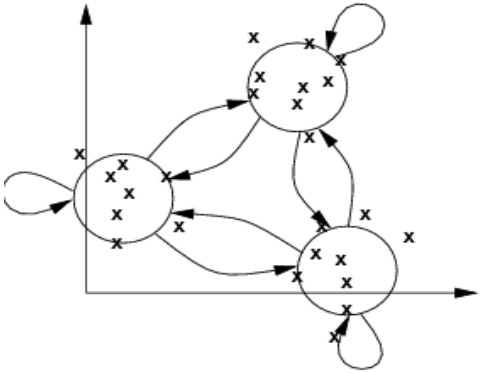
Lecture outline

Introduction

Chain-structured graphical models

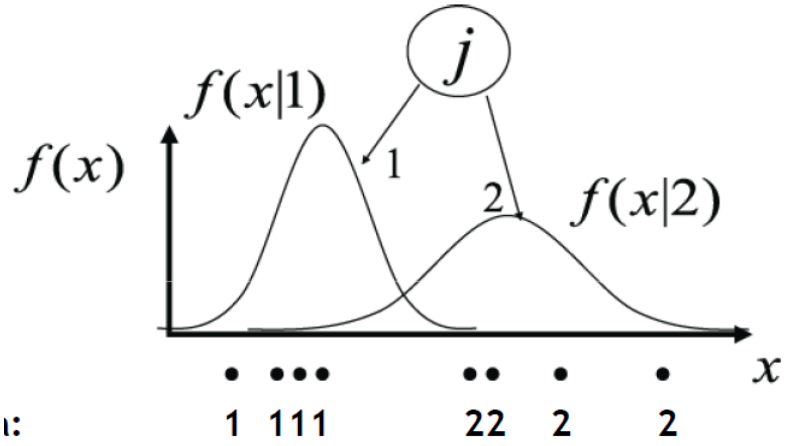
- Hidden Markov Models
- Kalman Filter

Tree-structured graphical models

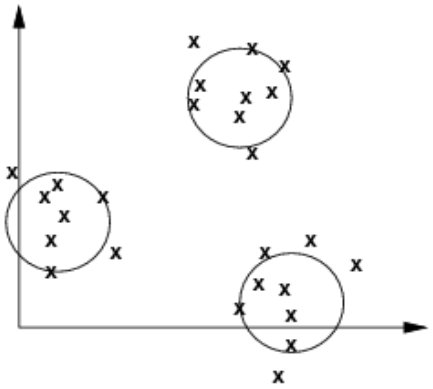


Hidden variables

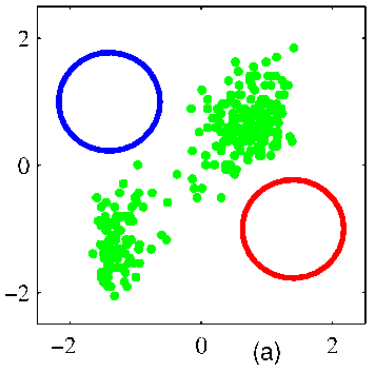
Mixture models: combine simple distributions into complex one

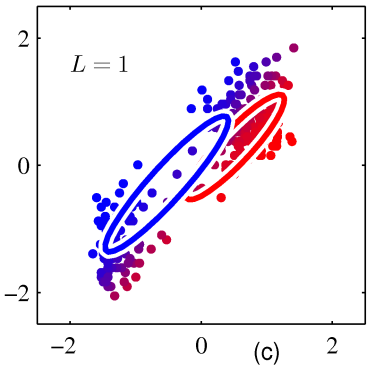


Mixture Models



Data clustering: side-product of parameter estimation





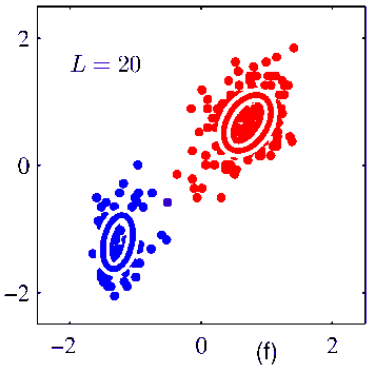
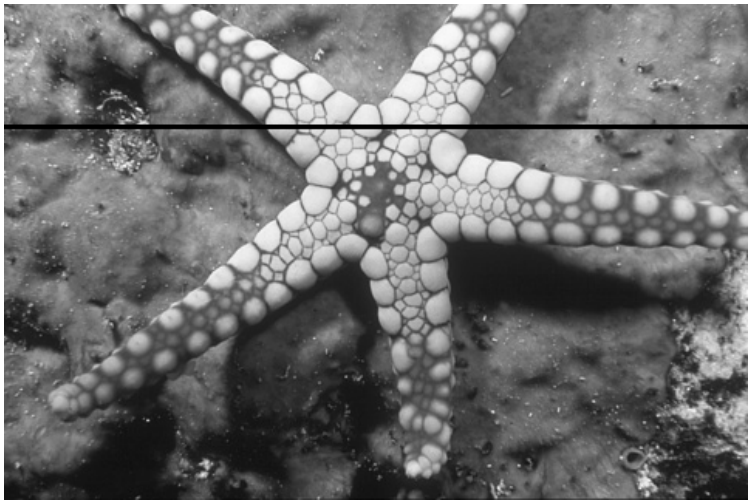
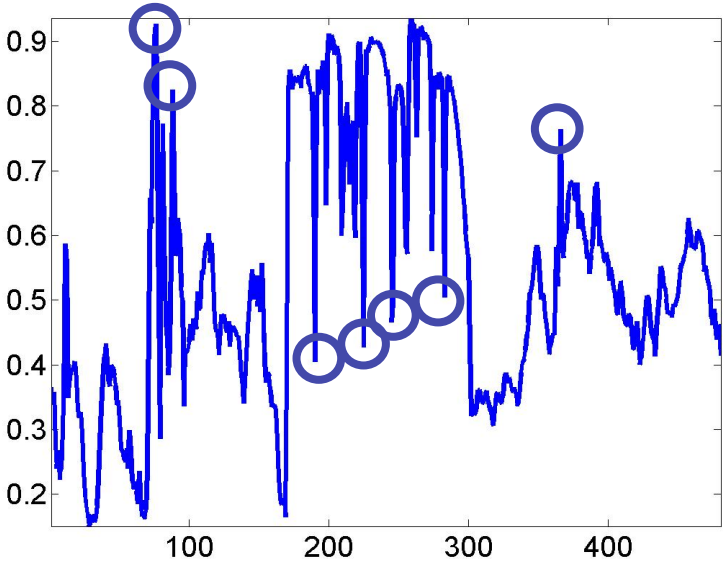


Image clustering - segmentation



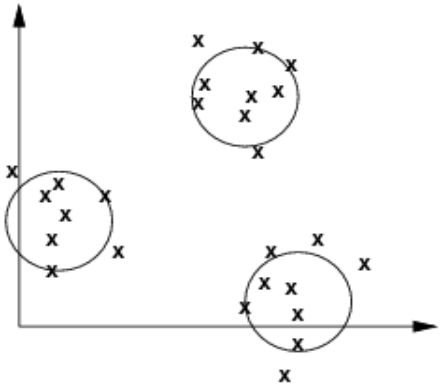
Observations are not independent

(Consecutive points: common region)

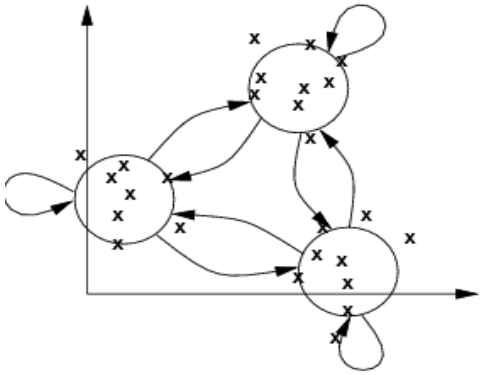


Dependent Hidden variables

Mixture Models

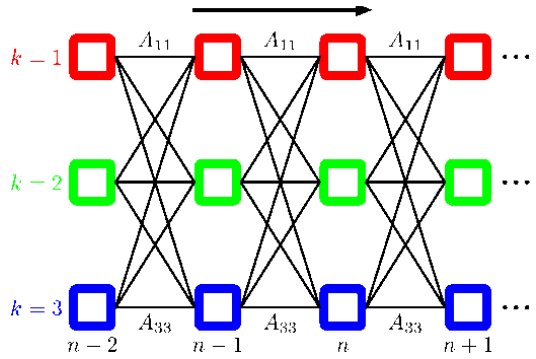


Hiden Markov Models

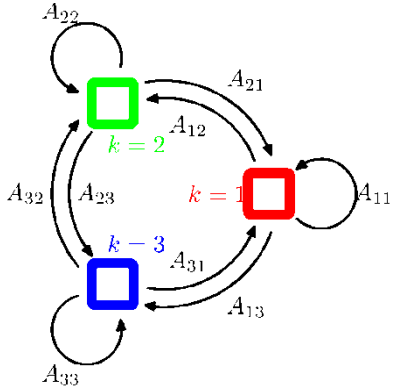


Introduce dependencies between hidden variables

First-order Markov Model



M x M transition matrix

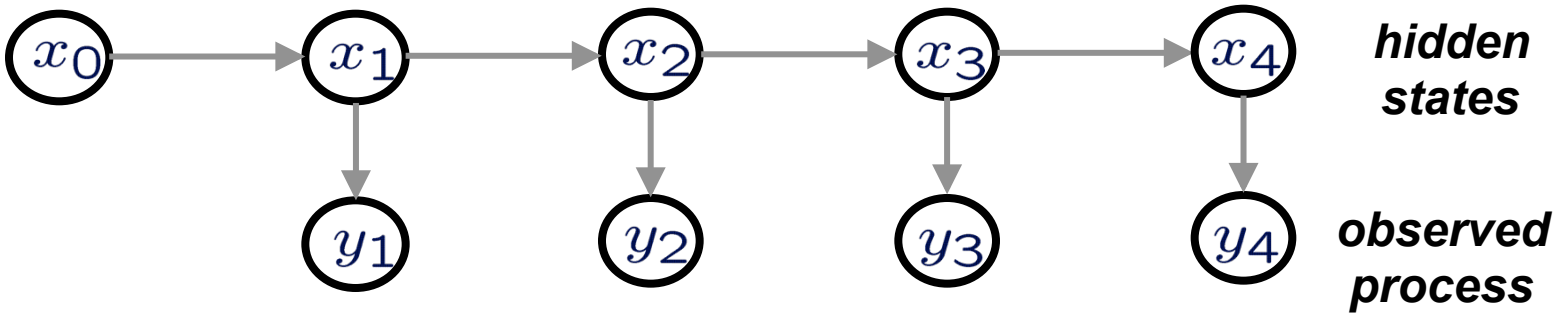


$$A_{i,j} = P(x_{t+1} = i | x_t = j)$$

$$\sum_i A_{i,j} = 1 \quad \forall j$$

$$A_{i,j} \geq 0$$

Hidden Markov Models



- Dependent sequence of observations
- Conditioned on state sequence, observations become independent

Parameters

Observation probabilities $P(y_t|x_t)$

$\pi(y_t = k|x_t = m) = n_{k,m}$ discrete

$P(y_t|x_t = k) = N(y_t; \mu_k, \Sigma_k)$ continuous

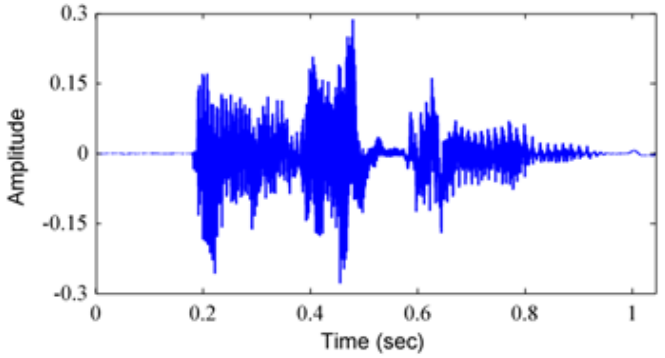
Transition probabilities $\pi(x_{t+1} = l|x_t = m) = \alpha_{l,m}$

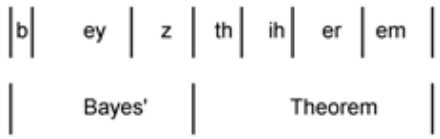
Starting probabilities $\pi(x_0 = m) = \beta_m$

Estimation: Expectation Maximization

HMMs & Speech Recognition

- Input: Air pressure signal

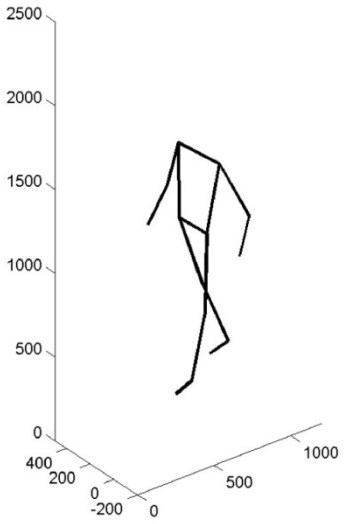




- Output: words/phrases
- Probabilistic model: signal given word
- Hidden Variables:
  phonemes (/ae/,/ey/,/z/…)
  phone subunits
- Hidden variables: not independent
  - Show,shoe,she,sell,sea,
  - Sv?
    (Sven,svelte,..)
  - Sb?

Lecture outline

- Introduction
- Chain-structured graphical models
  - Hidden Markov Models
  - Kalman Filter
- Tree-structured graphical models
- Loopy graphical models



Detection vs. Tracking


t=1


t=2

…

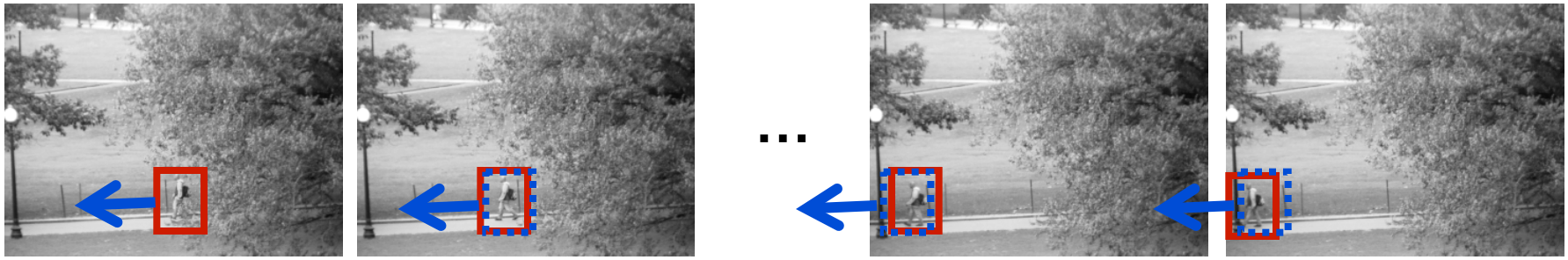

t=20


t=21

Detection vs. Tracking

- Detection: each frame independently

Detection vs. Tracking

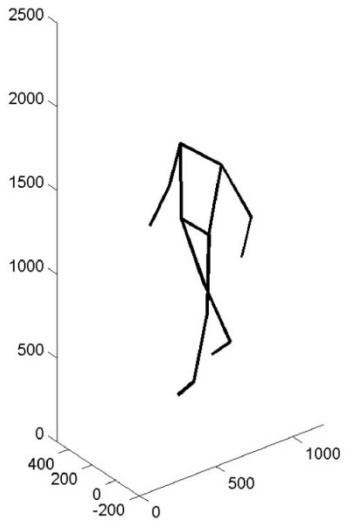


- Tracking with *dynamics*: combine image measurements with our expectation of the object's motion.
  - Restrict search for the object
  - Measurement noise is reduced by trajectory smoothness.

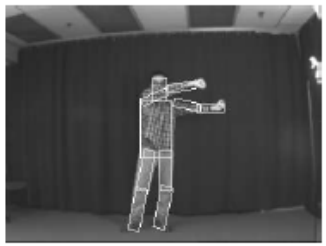
Tracking

- Input: Series of Images
- Output: Estimate of object's motion/action/gesture

- Hidden Variables (`State')
  - Time-dependent
  - Assume linear dynamics



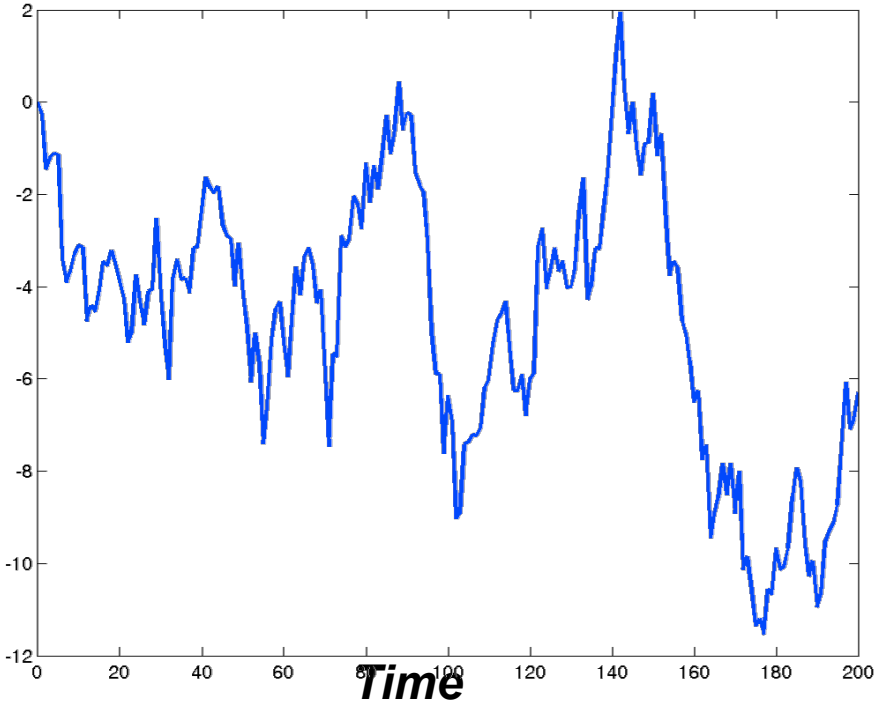
State, X



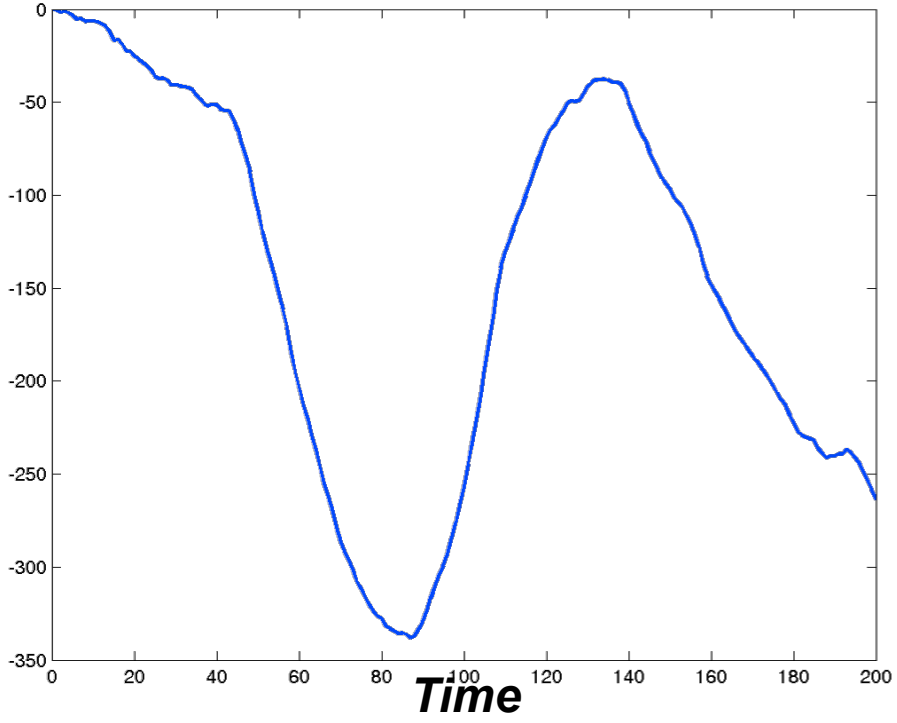
Observations, Y

Simple Linear Dynamics

Brownian Motion



Constant Velocity



$$x_{t+1} = x_t + w_t$$

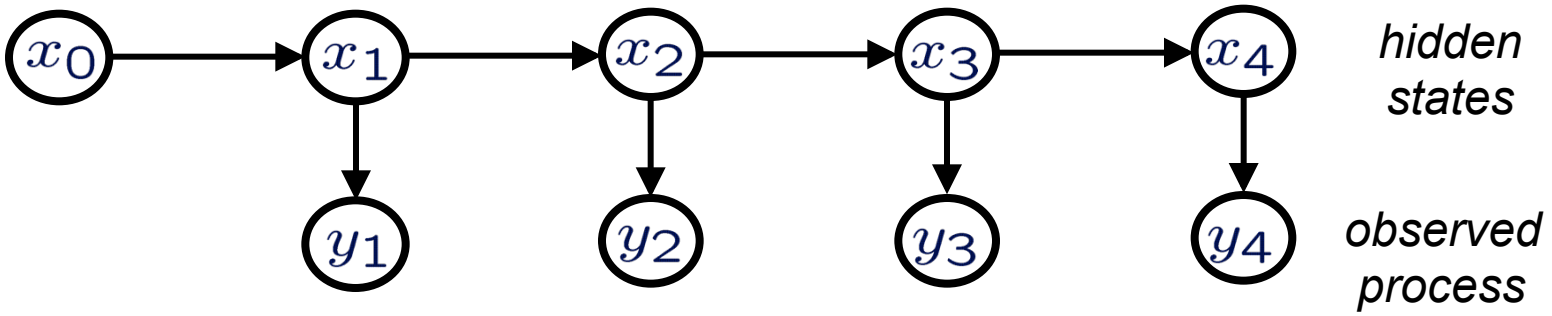
$$\begin{bmatrix} x_{t+1} \\ \delta_{t+1} \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x_t \\ \delta_t \end{bmatrix} + w_t$$

Slide credit: E. Sudderth

Linear Dynamical Systems

- Noise in State, noise in Observations

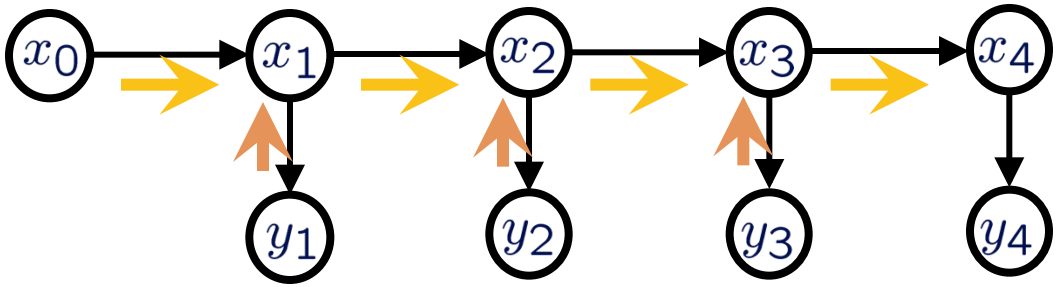
$$
\begin{aligned}
x_t &= Ax_{t-1} + v_t, \quad v_t \propto N(0, \Sigma) \\
y_t &= Cx_t + w_t, \quad w_t \propto N(0, Q) \\
x_1 &= \mu_0 + u, \quad u \propto N(0, V)
\end{aligned}
$$



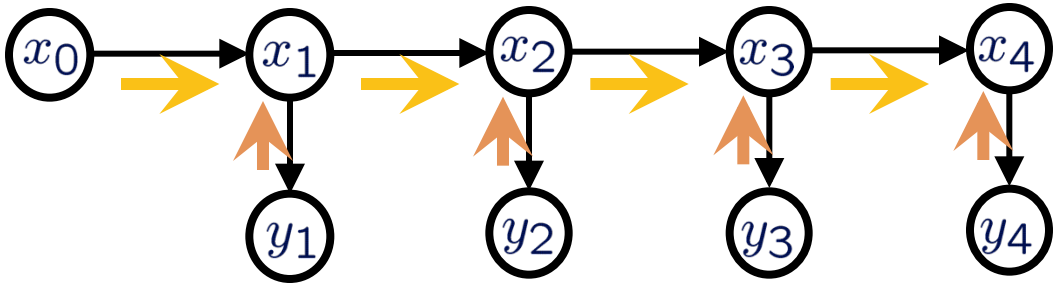
Slide credit: E. Sudderth

Inference Tasks

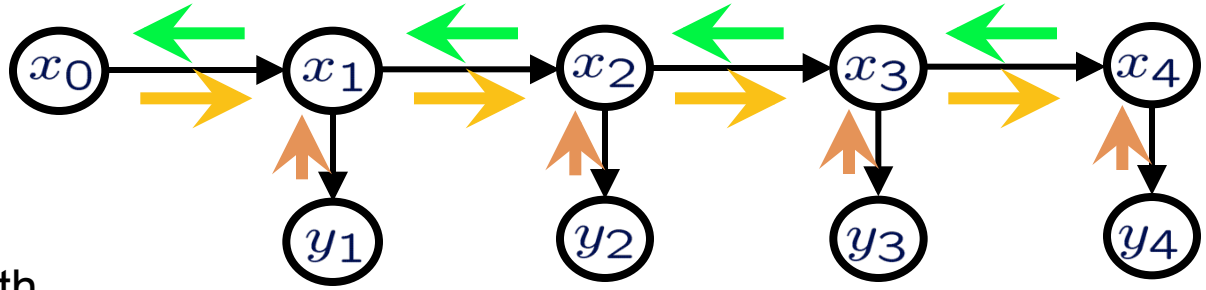
- Prediction



- Filtering
  (Kalman Filter)



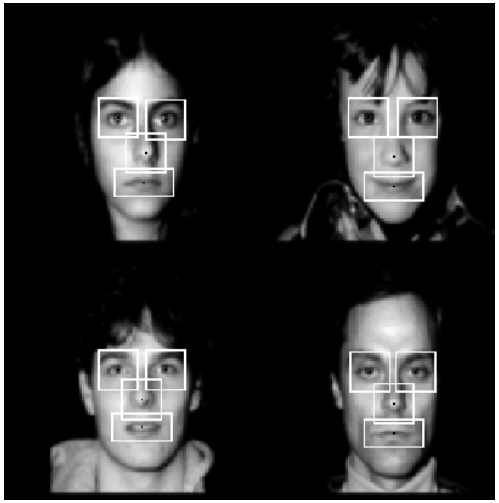
- Smoothing



Slide credit: E. Sudderth

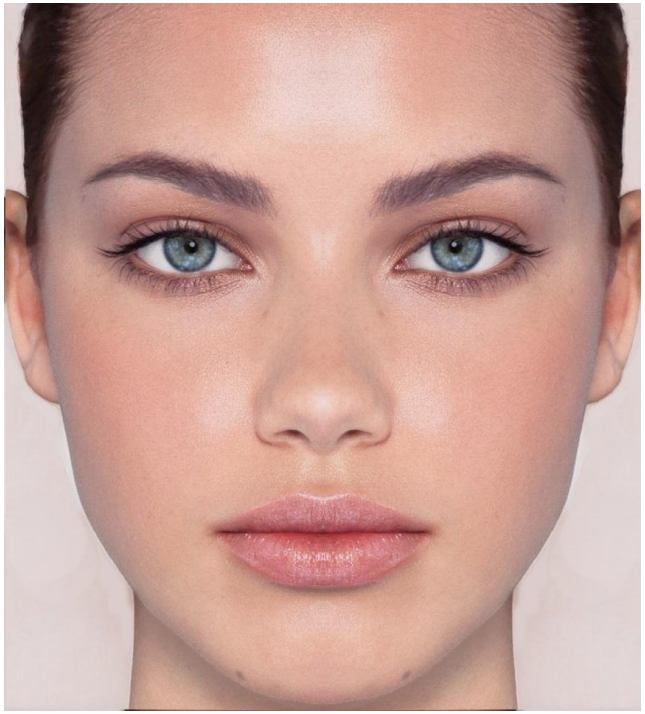
Lecture outline

- Introduction
- Chain-structured graphical models
- Tree-structured graphical models
  - Pictorial Structures
  - Max/Sum-Product algorithm

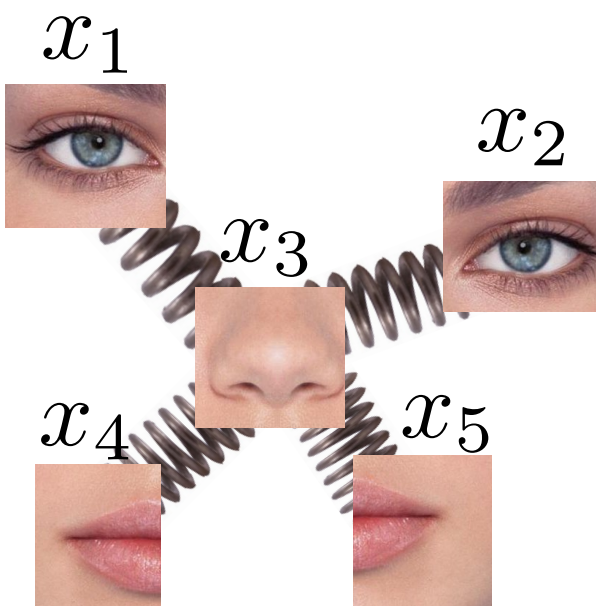


Problem N: N queues

Part-based Models

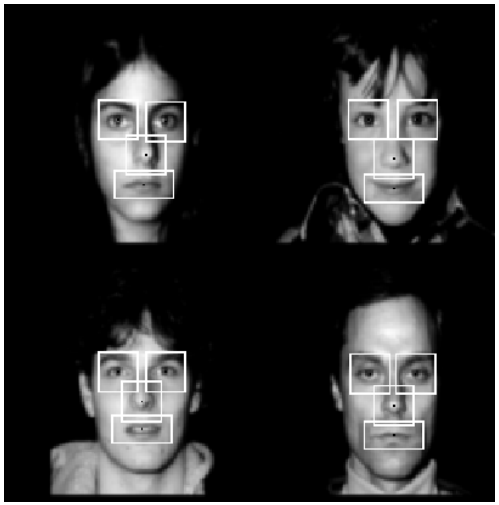


Part-based Models

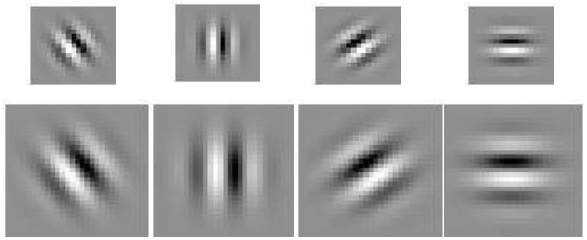


Unary terms (appearance model)

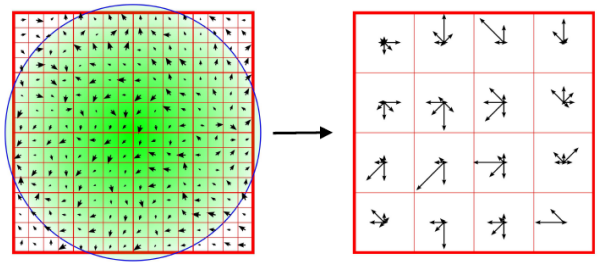
- Consider a patch of the image around the location of the part
- Construct statistical model for appearance at part location



Filterbank responses

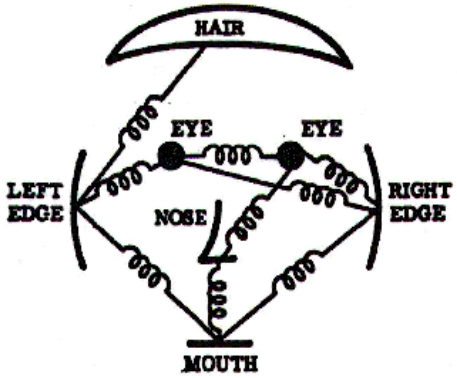


SIFT features

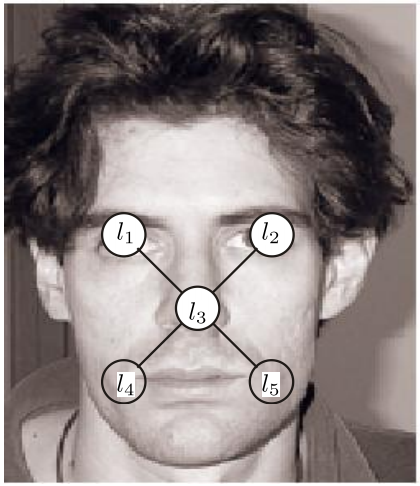


Pairwise terms (deformation model)

- Deformable objects as physical systems
  - Masses: driven towards good image locations
  - Springs: enforce relations among parts



- Star models (tree-structured models)
  - Pick single part as root (e.g. the nose)
  - Model other parts (eyes, mouth corners) w.r.t. root



- Probability of configuration $L = (l_1, l_2, l_3, l_4, l_5)$

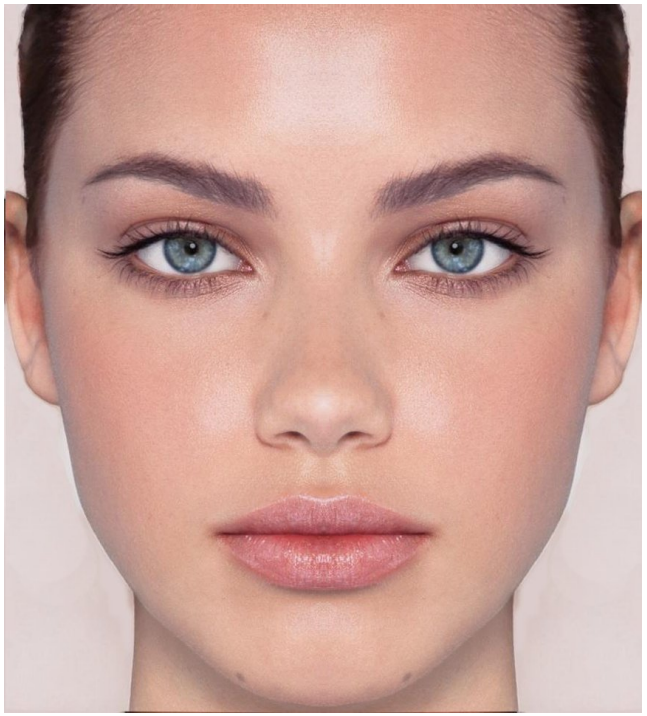
$$P(L|I) \propto P(I|L)P(L) = \prod_i \Phi_i(I_i|l_i) \prod_{(i,j)\in\mathcal{C}} \Psi_{i,j}(l_i, l_j)$$

Unary Terms Pairwise Terms

- How can we maximize over L?

Deformable Part Models (DPMs)

$$\mathbf{S}(\mathbf{x}) = \sum_{p=1}^{P} U_p(x_p) + B_p(x, x_p)$$



Local appearance

$$U_p(x_p) = \langle w_p, H(x_p) \rangle$$

Pairwise compatibility

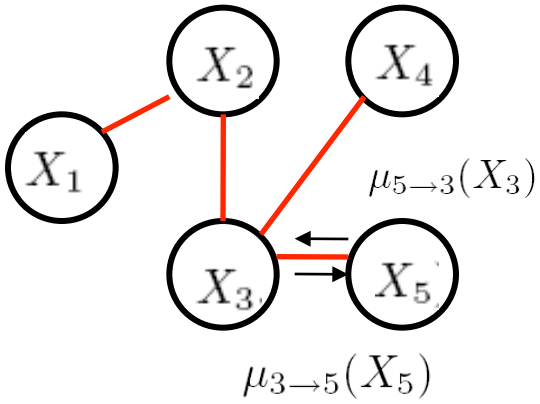
$$B_p(x, x_p) = -(h - h_p - \hat{h}_p)^2 \eta - (v - v_p - \hat{v}_p)^2 \nu$$

Object score

$$S(x) = \sum_{p=1}^{P} \max_{x'} \left[U_p(x') + B_p(x, x') \right]$$

Lecture outline

- Introduction
- Chain-structured graphical models
- Tree-structured graphical models
  - Pictorial Structures
  - Max/Sum-Product algorithm

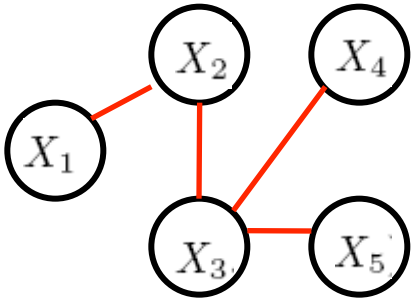


Inference on tree-structured Graphs

- Assume we want to find the most likely value of X_1

$$P(X) = \frac{1}{Z}\prod_{i=1}^{5}\Phi_i(X_i)\prod_{(i,j)\in\mathcal{C}}\Psi_{i,j}(X_i,X_j)$$

$$\mathcal{C} = \{(1,2),(2,3),(3,4),(3,5)\}$$



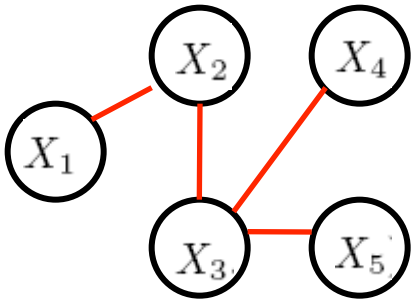
- Brute-force maximization:

$$\max P(X_1) = \max_{X_2,X_3,X_4,X_5} P(X_1,X_2,X_3,X_4,X_5)$$

$$|X_1| = K \rightarrow K^4$$

- Exploit factorization

Exploiting the factorization

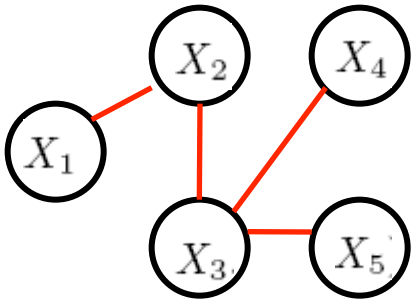


$$P(X) = \frac{1}{Z}\prod_{i=1}^{5}\Phi_i(X_i)\prod_{(i,j)\in\mathcal{C}}\Psi_{i,j}(X_i,X_j)$$

$$\mathcal{C} = \{(1,2),(2,3),(3,4),(3,5)\}$$

$$\max P(X_1)$$

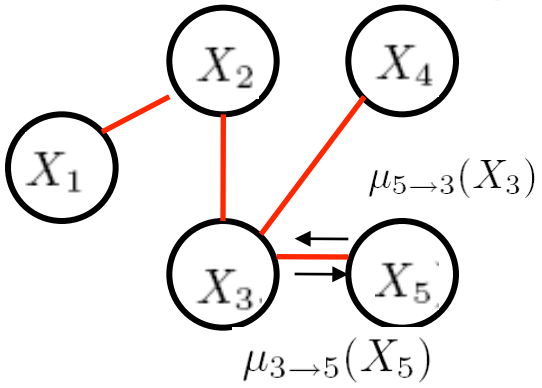
Exploiting the factorization



$$P(X) = \frac{1}{Z}\prod_{i=1}^{5}\Phi_i(X_i)\prod_{(i,j)\in\mathcal{C}}\Psi_{i,j}(X_i,X_j)$$
$$\mathcal{C} = \{(1,2),(2,3),(3,4),(3,5)\}$$

$$\begin{aligned}
& \max P(X_1) \\
= \; & \max_{X_2,X_3,X_4,X_5}\Phi(X_1)\Phi(X_2)\Phi(X_3)\Phi(X_4)\Phi(X_5)\Psi(X_1,X_2)\Psi(X_2,X_3)\Psi(X_3,X_4)\Psi(X_3,X_5) \\
= \; & \Phi(X_1)\max_{X_2}\Phi(X_2)\Psi(X_1,X_2)\max_{X_3,X_4}\Phi(X_3)\Psi(X_2,X_3)\Phi(X_4)\Psi(X_3,X_4)\underbrace{\max_{X_5}\Phi(X_5)\Psi(X_3,X_5)}_{\mu_5(X_3)} \\
= \; & \Phi(X_1)\max_{X_2}\Phi(X_2)\Psi(X_1,X_2)\max_{X_3}\Phi(X_3)\Psi(X_2,X_3)\mu_5(X_3)\underbrace{\max_{X_4}\Phi(X_4)\Psi(X_3,X_4)}_{\mu_4(X_3)} \\
= \; & \Phi(X_1)\max_{X_2}\Phi(X_2)\Psi(X_1,X_2)\underbrace{\max_{X_3}\Phi(X_3)\Psi(X_2,X_3)\mu_5(X_3)\mu_4(X_3)}_{\mu_3(X_2)} \\
= \; & \Phi(X_1)\underbrace{\max_{X_2}\Phi(X_2)\Psi(X_1,X_2)\mu_3(X_2)}_{\mu_2(X_1)}
\end{aligned}$$

Max-Product algorithm



- Distributed computation on a graph
  - `message-passing'

- Message from node i to node j

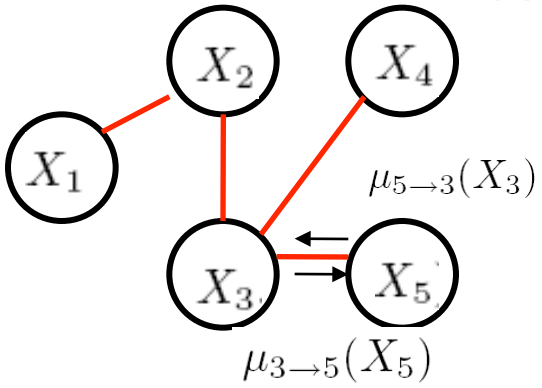
$$\mu_{i\to j}(X_j) \quad = \quad \max_{X_i} \Phi_i(X_i)\Psi_{i,j}(X_i,X_j) \prod_{k\in\mathcal{N}(i)\setminus j} \mu_{k\to j}(X_i)$$

  - `Given all I know from the rest, this is where you should be'

- Beliefs: $$B_j(X_j) \quad = \quad \Phi_j(X_j) \prod_{i\in\mathcal{N}(j)} \mu_{i\to j}(X_j)$$

- At convergence $$B_j(X_j) = \max P(X_j)$$

Sum-Product algorithm



- Distributed computation on a graph
  - `message-passing'

- Message from node i to node j

$$\mu_{i\to j}(X_j) = \sum_{X_i} \psi_i(X_i)\psi_{i,j}(X_i, X_j) \prod_{k\in\{\mathcal{N}(i)\setminus j\}} \mu_{k\to i}(X_i)$$

  - `Given all I know from the rest, this is where you should be'

- Beliefs: $$B_j(X_j) = \Phi_j(X_j) \prod_{i\in\mathcal{N}(j)} \mu_{i\to j}(X_j)$$

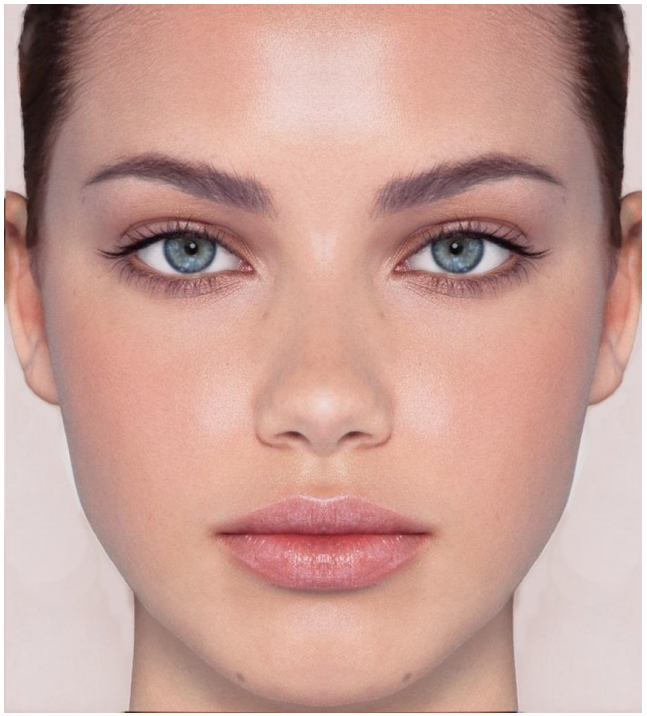
- At convergence $$P(X_i) = B(X_i)$$

Max/Sum-Product algorithms

- Asychronous inference
- Local computations
- Guaranteed optimality for tree-structured graphs
- Equivalent algorithms:
  - Tracking: Kalman Filtering (Sum-Product)
  - HMMs: Alpha-Beta (Sum-Product)
  - Coding/HMMs: Viterbi Algorithm (Max-Product)
  - Bayesian Nets: Belief Propagation (Sum-Product)
- Max Product: Dynamic Programming

Deformable Part Models (DPMs)

$$\mathbf{S}(\mathbf{x}) = \sum_{p=1}^{P} U_p(x_p) + B_p(x, x_p)$$



Local appearance

$$U_p(x_p) = \langle w_p, H(x_p) \rangle$$

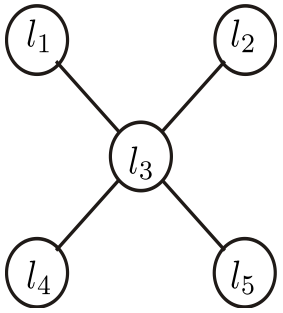
Pairwise compatibility

$$B_p(x, x_p) = -(h - h_p - \hat{h}_p)^2 \eta - (v - v_p - \hat{v}_p)^2 \nu$$

Object score

$$S(x) = \sum_{p=1}^{P} \max_{x'} \left[U_p(x') + B_p(x, x') \right]$$

Pictorial Structures



- Graphical model
  - Nodes: part locations
  - Edges: dependencies - relative locations

- Object localization: maximization with respect to $L = (l_1, l_2, l_3, l_4, l_5)$

$$P(L|I) \propto P(I|L)P(L) = \prod_i \Phi_i(I_i|l_i) \prod_{(i,j)\in\mathcal{C}} \Psi_{i,j}(l_i, l_j)$$

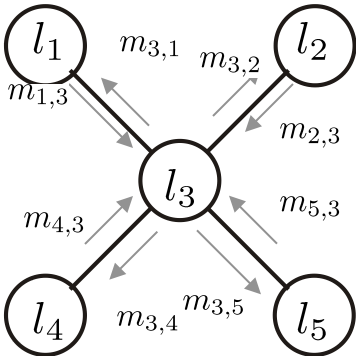
Unary terms Pairwise terms

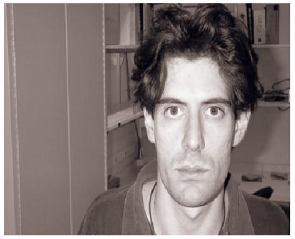
- Message Passing: $\mu_{i\to j}(X_j) = \max_{X_i} \Phi_i(X_i)\Psi_{i,j}(X_i, X_j) \prod_{k\in\mathcal{N}(i)\setminus j} \mu_{k\to j}(X_i)$

- Efficient Implementation: Felzenszwalb & Huttenlocher, CVPR 2001/IJCV 05

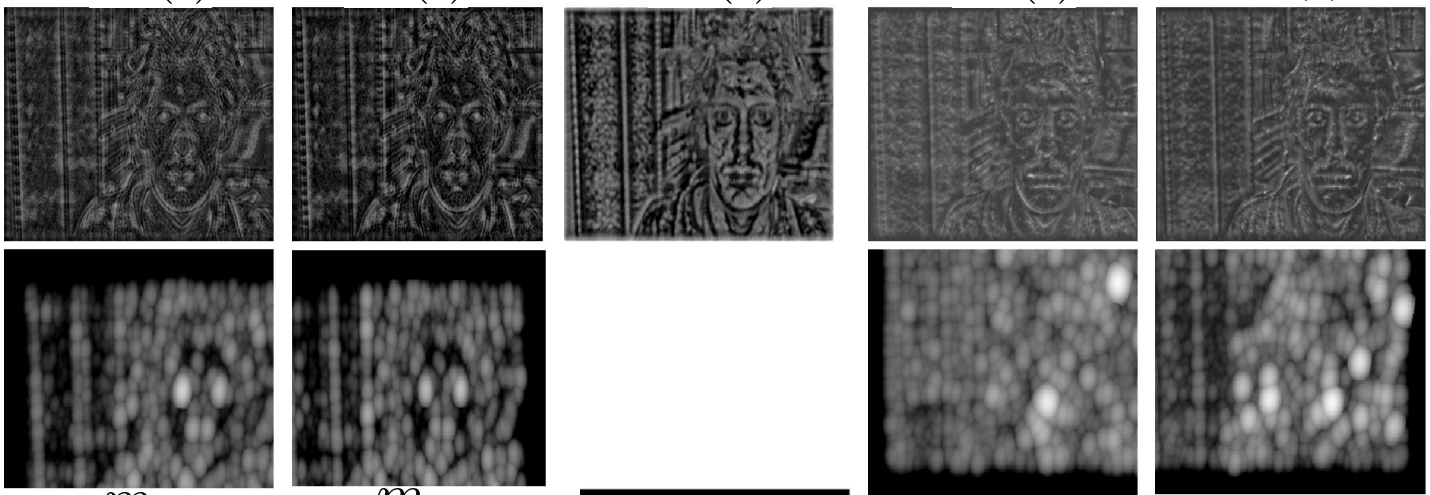
$$\mu_{i\to j}(X_j) = \max_{X_i} \Phi_i(X_i)\Psi_{i,j}(X_i, X_j) \prod_{k\in\mathcal{N}(i)\setminus j} \mu_{k\to j}(X_i)$$

$$B_j(X_j) = \max P(X_j)$$

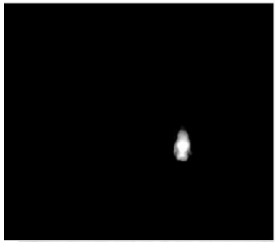




$\Phi_1(l)$ $\Phi_2(l)$ $\Phi_3(l)$ $\Phi_4(l)$ $\Phi_5(l)$

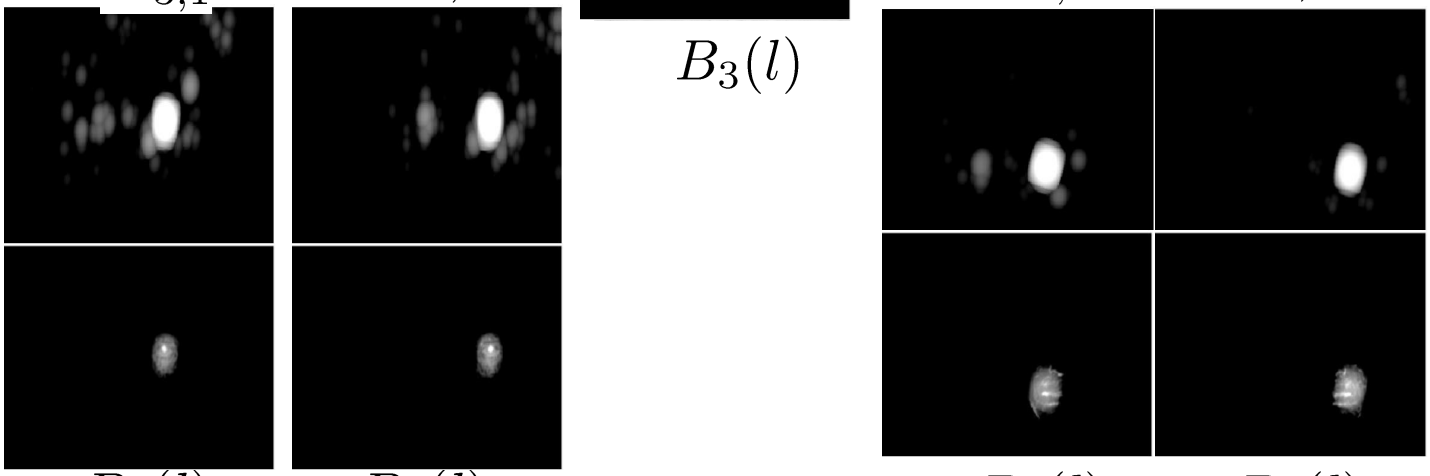


$m_{1,3}$ $m_{2,3}$ $m_{4,3}$ $m_{5,3}$



$B_3(l)$

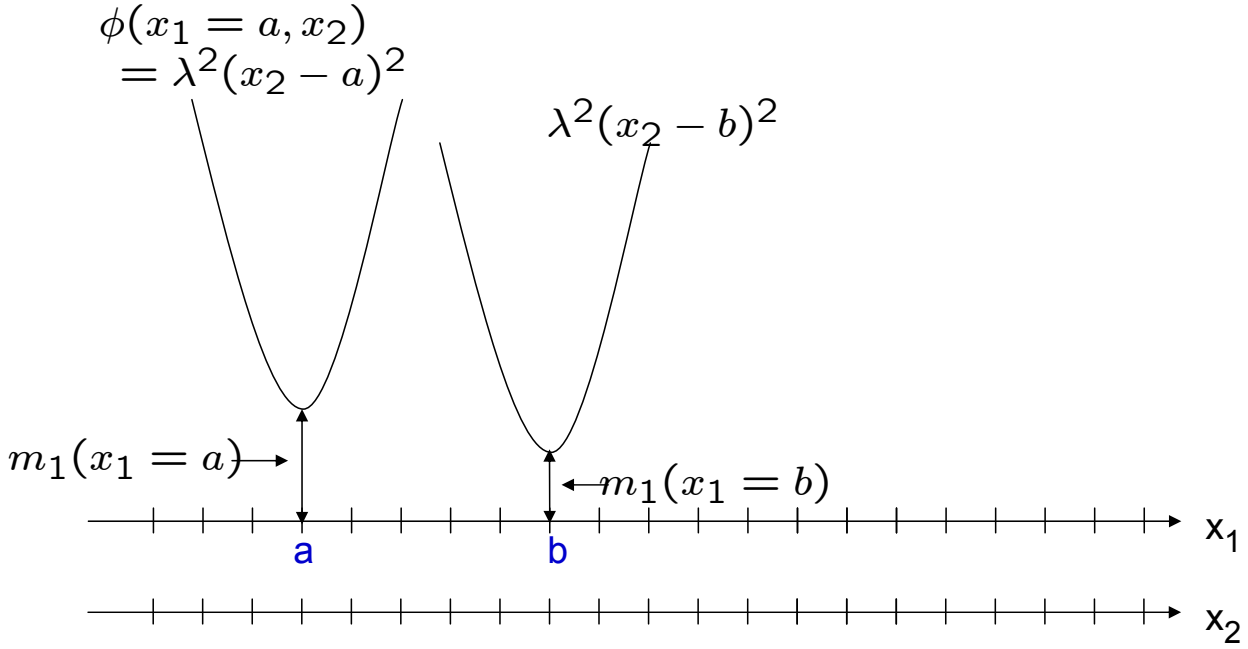
$m_{3,1}$ $m_{3,2}$ $m_{3,4}$ $m_{3,5}$



$B_1(l)$ $B_2(l)$ $B_4(l)$ $B_5(l)$

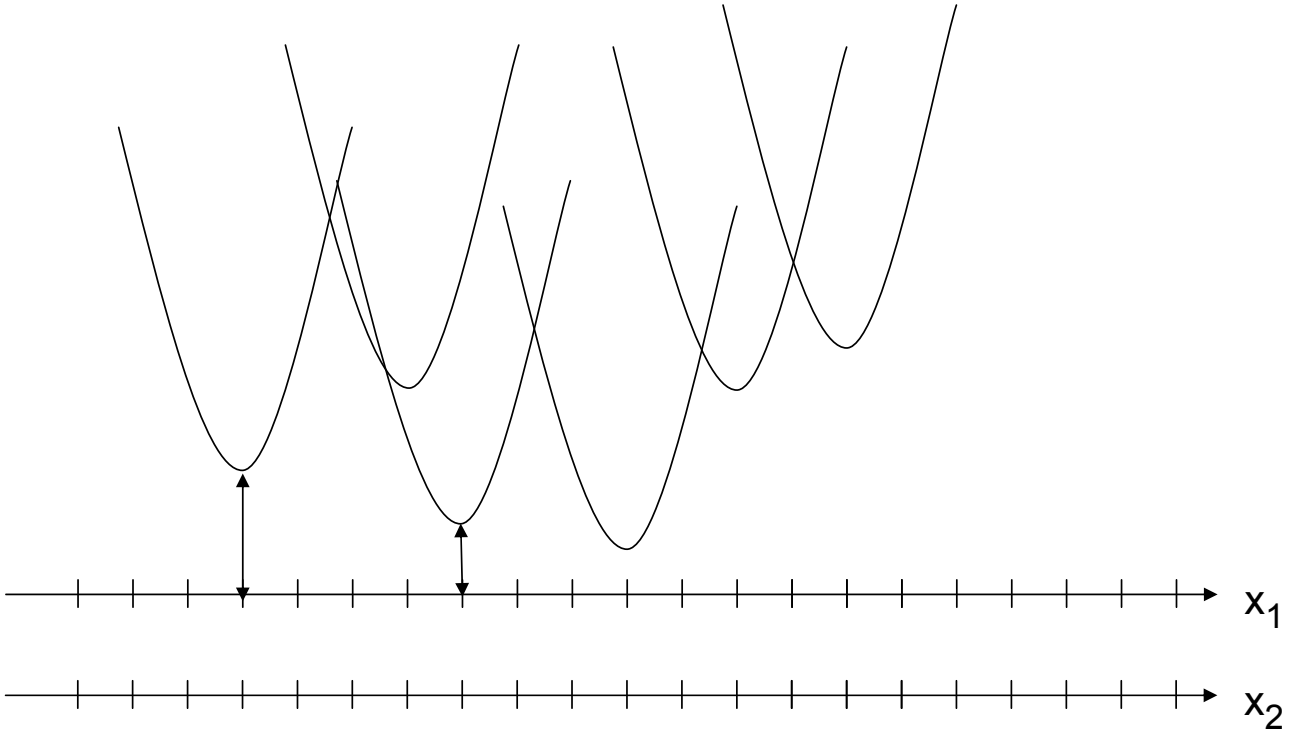
Generalized Distance Transforms- 1D

Plot $\min_{x_1}\{m_1(x_1) + \phi(x_1, x_2)\}$ as function of x_2



GDT- 1D

Plot $\min_{x_1}\{m_1(x_1) + \phi(x_1, x_2)\}$ as function of x_2

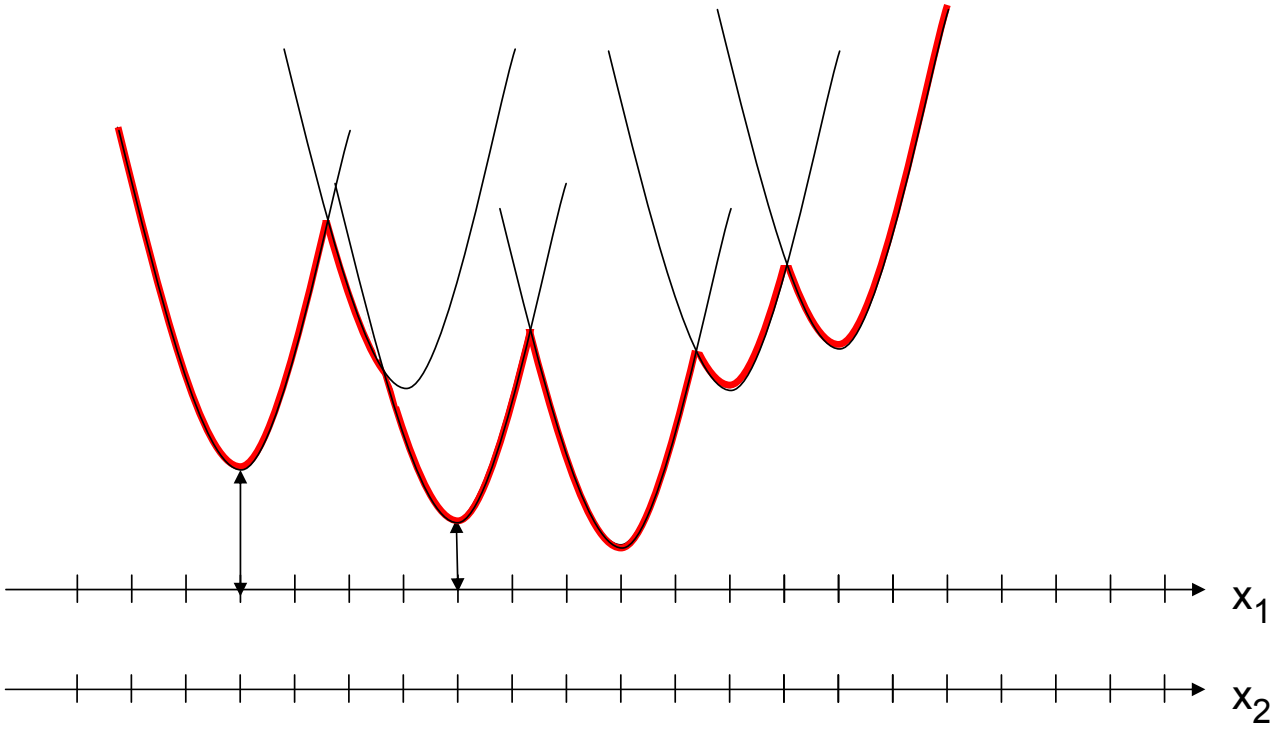


For each x_2

- Finding min over x_1 is equivalent finding minimum over set of offset parabolas
- Lower envelope computed in O(h) rather than $O(h^2)$ via gen. distance transform

GDT- 1D

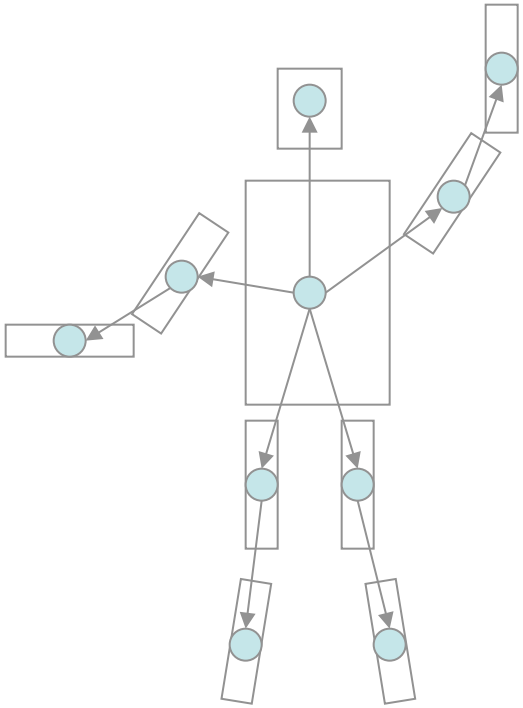
Plot $\min_{x_1}\{m_1(x_1) + \phi(x_1, x_2)\}$ as function of x_2

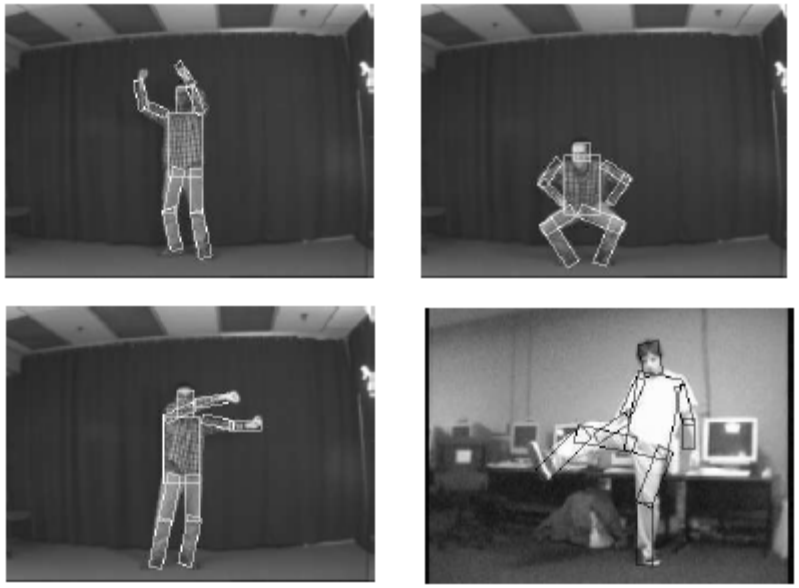


For each x_2

- Finding min over x_1 is equivalent finding minimum over set of offset parabolas
- Lower envelope computed in O(h) rather than $O(h^2)$ via gen. distance transform

Detection Results

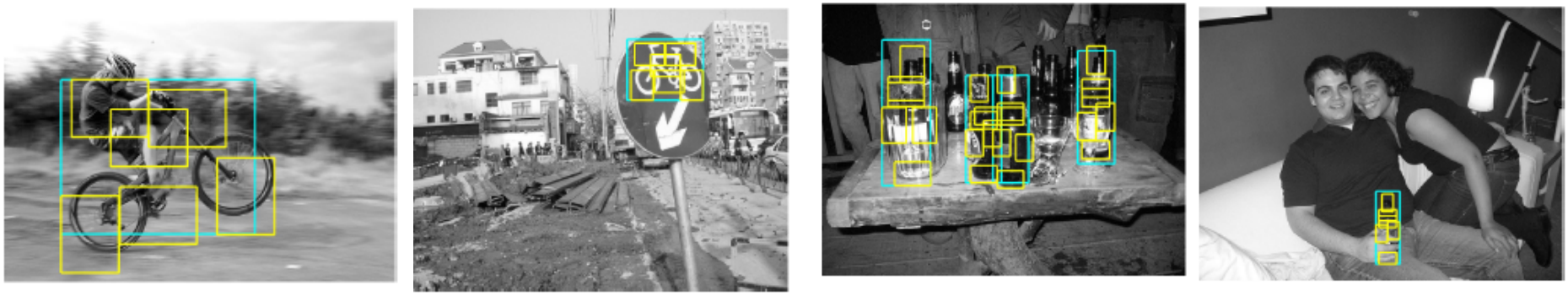




Part-based models + discriminative framework

P. Felzenszwalb, D. McAllester, D. Ramanan`A Discriminatively Trained, Multiscale, Deformable Part Model' CVPR 2008

- Linear classifier gives individual part cost
- Gaussian-like cost for spatial offsets
- Learn cost for parts and deformations
  - Discriminative framework

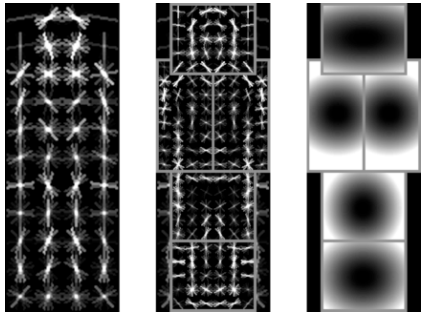


- State-of-the-art, 2008

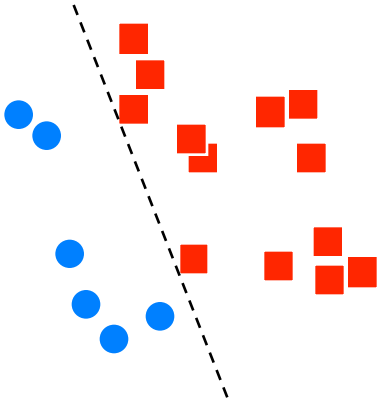
Discriminative Training of DPMs

- Training data consists of images with labeled bounding boxes.
- Need to learn the model structure, filters and deformation costs.

Training

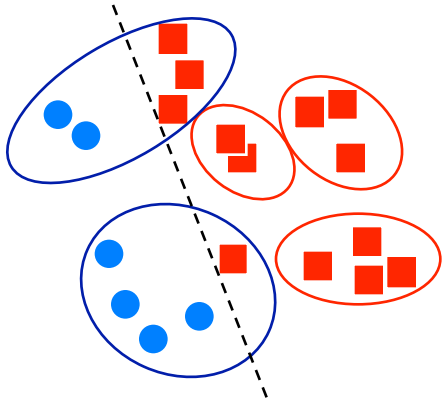


Multiple Instance Learning



Typical Learning

$$S = \{(x^i, y^i)\}$$



Multiple Instance Learning

$$S = \{(\underbrace{\{x^{i,1}, \ldots, x^{i,|B_i|}\}}_{B_i}, y^i)\}$$

$$y^i \in \{0, 1\} \; x^i \in \mathcal{X}$$

$$F : \mathcal{X} \rightarrow \{0, 1\}$$

Positive bag: at least one instance should be positive

Negative bag: no instance should be positive

$$\max_{x \in B_i} F(x) = y^i$$

Slide Credit: K. Grauman

Score function for DPM detection

$$\text{score}(p_0, \ldots, p_n) = \underbrace{\sum_{i=0}^{n} F_i \cdot \phi(H, p_i)}_{\text{“data term”}} - \underbrace{\sum_{i=1}^{n} d_i \cdot (dx_i^2, dy_i^2)}_{\text{“spatial prior”}}$$

filters

displacements

deformation parameters

$$\text{score}(z) = \beta \cdot \Psi(H, z)$$

concatenation filters and deformation parameters

concatenation of HOG features and part displacement features

SVM training for DPMs

Classifiers that score an example x using

$$f_\beta(x) = \max_{z \in Z(x)} \beta \cdot \Phi(x, z)$$

β are model parameters
z are latent values

Training data $D = (\langle x_1, y_1 \rangle, \ldots, \langle x_n, y_n \rangle) \qquad y_i \in \{-1, 1\}$

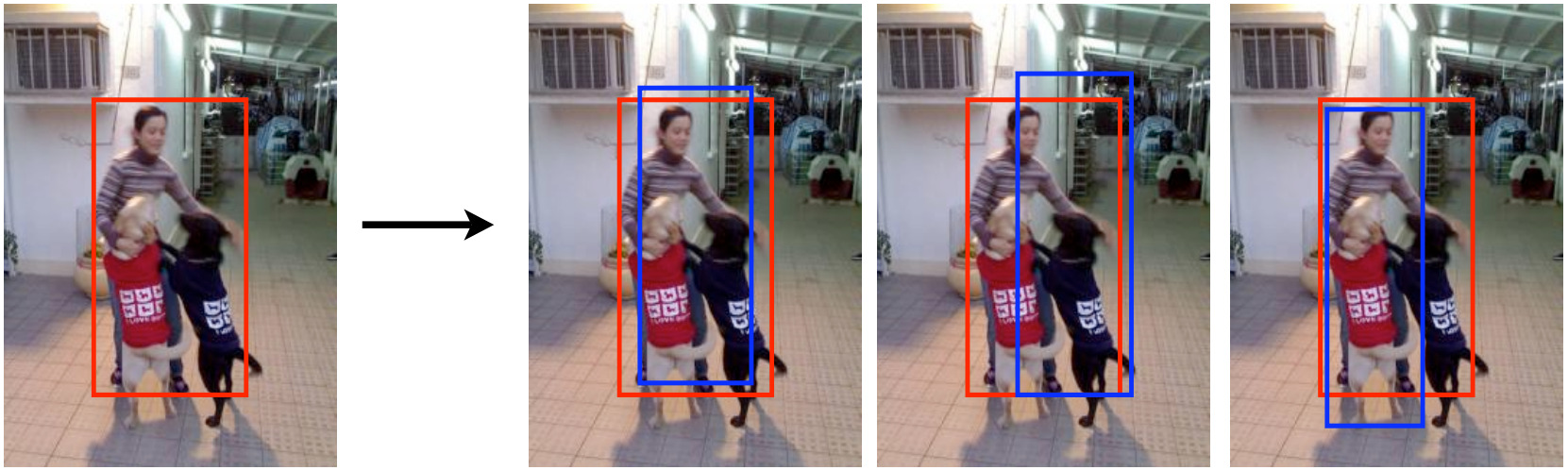
We would like to find β such that: $y_i f_\beta(x_i) > 0$

Minimize

$$L_D(\beta) = \frac{1}{2}||\beta||^2 + C \sum_{i=1}^{n} \max(0, 1 - y_i f_\beta(x_i))$$

Annotation & Latent Variables

- Positive example specifies some z should have high score
- Bounding box defines range of root locations
  - Parts can be anywhere
  - This defines $Z(x)$



P.F. Felzenszwalb, R. B. Girshick, D. Ramanan and D. A. McAllester, a discriminatively trained multiscale deformable part model, IEEE PAMI, 2010

Latent SVM training

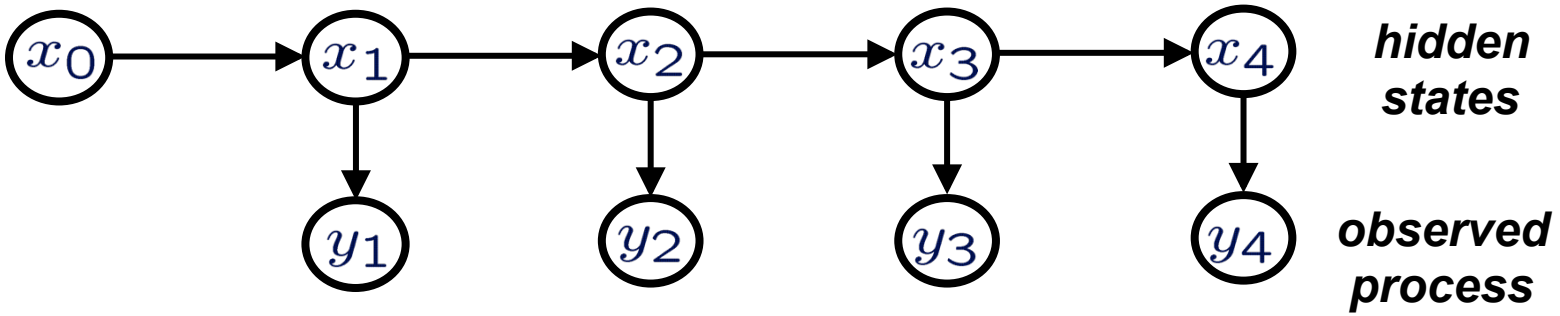
$$L_D(\beta) = \frac{1}{2}||\beta||^2 + C\sum_{i=1}^{n}\max(0, 1 - y_i f_\beta(x_i))$$

- Convex if we fix z for positive examples
- Optimization:
  - Initialize β and iterate:
    - Pick best z for each positive example
    - Optimize β via gradient descent with data-mining

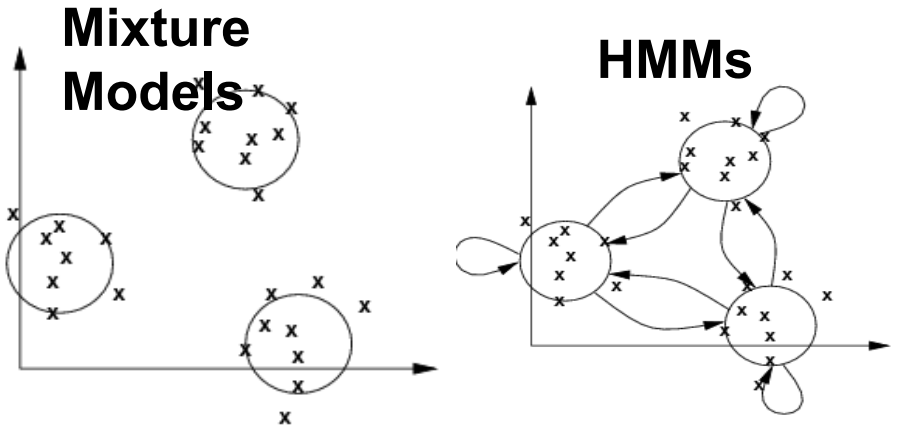
Appendix: Inference & Learning for Chain-structured GMs

- Discrete State: Hidden Markov Models
  - **Forward-Backward**
  - Viterbi
  - Learning
- Continuous State
  - Kalman Filter
  - Particle Filters

Hidden Markov Models



- Dynamics take simple form based on hidden states
  - Process: word utterance
  - Hidden States: phonemes
  - Observed: speech signal



Three Main problems

- Likelihood Evaluation

$$P(Y)$$

- Inference

$$\arg\max_X P(X|Y)$$

- Parameter Estimation

$$\arg\max_\theta P(Y|\theta) \qquad \theta = \{A, \mu_{1\ldots N}, \Sigma_{1\ldots N}, \pi\}$$

Three Algorithms

- Likelihood Evaluation (Forward-Backward)

$$P(Y)$$

- Inference (Viterbi)

$$\arg\max_X P(X|Y)$$

- Parameter Estimation (EM/Baum-Welch)

$$\arg\max_\theta P(Y|\theta) \quad \theta = \{A, \mu_{1\ldots N}, \Sigma_{1\ldots N}, \pi\}$$

Likelihood Evaluation

- Using Marginalization

$$
\begin{aligned}
P(Y) &= \sum_{X} P(X,Y) \\
&= \sum_{X} P(Y|X)P(X) \\
&= \sum_{x_1}\sum_{x_2}\cdots\sum_{x_T} \underbrace{\prod_{t=1}^{T} P(y_t|x_t)}_{P(Y|X)} \underbrace{P(x_1)\prod_{t=1}^{T-1} P(x_i|x_{i-1})}_{P(X)}
\end{aligned}
$$

- Complexity: TN^T

  where N is the number of states

Exploiting the independencies

- Rewrite likelihood as

$$
\begin{aligned}
P(Y) &= \sum_{x_1}\sum_{x_2}\dots\sum_{x_T}\prod_{t=1}^{T} P(y_t|x_t)P(x_1)\prod_{t=1}^{T-1} P(x_i|x_{i-1}) \\
&= \sum_{x_1} P(y_1|x_1)P(x_1)\sum_{x_2} P(y_2|x_2)P(x_2|x_1)\dots\sum_{x_T} P(y_T|x_T)P(x_T|x_{T-1})
\end{aligned}
$$

$$
\begin{aligned}
\alpha_1(j) &= P(x_1 = j, y_1) \\
\alpha_2(j) &= P(x_2 = j, y_1, y_2) \\
\alpha_t(j) &= P(x_t = j, y_1, \dots, y_t)
\end{aligned}
$$

- Observe that $\sum_j \alpha_T(j) = \sum_j P(x_T = j, y_1, \dots, y_T) = P(Y)$

Recursion for α

$$
\begin{aligned}
\alpha_t(j) &= P(x_t, y_1, \ldots, y_t) \\
&= P(y_1, \ldots, y_t | x_t) P(x_t) \\
&= P(y_1, \ldots, y_{t-1} | x_t, y_t) P(y_t | x_t) P(x_t) \\
&= P(y_1, \ldots, y_{t-1} | x_t) P(y_t | x_t) P(x_t) \\
&= \sum_{x_{t-1}} P(y_1, \ldots, y_{t-1}, x_{t-1} | x_t) P(y_t | x_t) P(x_t) \\
&= \sum_{x_{t-1}} P(y_1, \ldots, y_{t-1} | x_{t-1}, x_t) P(x_{t-1} | x_t) P(x_t) P(x_{t-1}, x_t) P(y_t | x_t) \\
&= \sum_{x_{t-1}} P(y_1, \ldots, y_{t-1} | x_{t-1}) P(x_t | x_{t-1}) P(x_{t-1}) P(y_t | x_t) \\
&= \sum_{x_{t-1}} P(y_1, \ldots, y_{t-1}, x_{t-1}) P(x_t | x_{t-1}) P(y_t | x_t) \\
&= \sum_{x_{t-1}} \alpha_{t-1}(x_{t-1}) P(x_t | x_{t-1}) P(y_t | x_t)
\end{aligned}
$$

$$P(a, b|c) = P(a|b, c)P(b|c)$$

$$\alpha_t(x_t) = \sum_{x_{t-1}} \alpha_{t-1}(x_{t-1}) P(x_t | x_{t-1}) P(y_t | x_t)$$

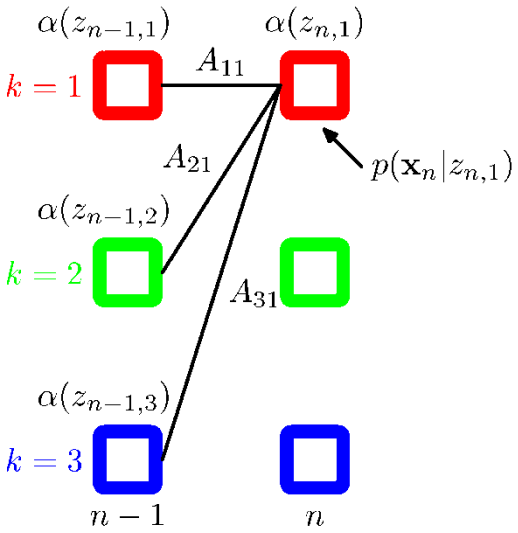
$$\alpha_1(x_1) = P(x_1, y_1) = P(y_1 | x_1) P(x_1)$$

Recursion for α

$$\alpha_t(x_t) = \sum_{x_{t-1}} \alpha_{t-1}(x_{t-1}) P(x_t|x_{t-1}) P(y_t|x_t)$$

$$\alpha_1(x_1) = P(x_1, y_1) = P(y_1|x_1) P(x_1)$$

Forward in time



Rewrite likelihood

$$P(Y) = \sum_{x_1} \underbrace{P(y_1|x_1)P(x_1)}_{\alpha_1(x_1)} \sum_{x_2} P(y_2|x_2)P(x_2|x_1) \dots \sum_{x_T} P(y_T|x_T)P(x_T|x_{T-1})$$

$$= \sum_{x_2} \underbrace{\sum_{x_1} \alpha_1(x_1)P(y_2|x_2)P(x_2|x_1)}_{\alpha_2(x_2)} \sum_{x_3} P(y_3|x_3)P(x_3|x_2) \dots \sum_{x_T} P(y_T|x_T)P(x_T|x_{T-1})$$

$$T = 50, \quad N = 10$$

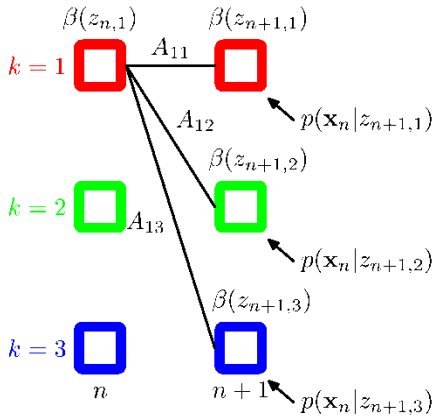
- Brute force: $T \times N^T \quad 50 \times 10^{50}$
- Recursive: $T \times N^2 \quad 50 \times 100$

Backward Variables β

- Similar recursion holds: $\beta_t(j) = P(y_{t+1}, y_{t+2}, \ldots, y_T | x_t = j)$

$$\beta_T(i) = 1$$

- Backward in time: $\beta_t(x_t) = \sum_{x_{t+1}} P(x_{t+1}|x_t) P(y_{t+1}|x_{t+1}) \beta_{t+1}(x_{t+1})$



- Discrete State: Hidden Markov Models
  - Forward-Backward
  - **Viterbi**
  - Learning
- Continuous State
  - Kalman Filter
  - Particle Filters

Three Main problems

- Likelihood Evaluation (Forward-backward)

$$P(Y)$$

- **Inference** **(Viterbi)**

$$\arg\max_X P(X|Y)$$

- Parameter Estimation (EM)

$$\arg\max_\theta P(Y|\theta) \qquad \theta = \{A, \mu_{1\ldots N}, \Sigma_{1\ldots N}, \pi\}$$

Posterior for a single state

- Bayes' optimal decision.. $$x_t^* = \arg\max P(x_t|Y)$$
- Introduce $$\gamma_t(i) = P(x_t = i|y_1, \ldots, y_T).$$

- Based on Bayes' rule

$$\begin{aligned} P(x_t|y_1, \ldots, y_T) &= \frac{P(x_t, y_1, \ldots, y_T)}{P(y_1, \ldots, y_T)} \\ &= \frac{P(x_t, y_1, \ldots, y_T)}{\sum_{x_t} P(x_t, y_1, \ldots, y_T)} \end{aligned}$$

- Can rewrite using α,β

γ=αβ

$$
\begin{aligned}
P(x_t|y_1,\ldots,y_T) &= \frac{P(x_t,y_1,\ldots,y_T)}{P(y_1,\ldots,y_T)} \\
&= \frac{P(x_t,y_1,\ldots,y_T)}{\sum_{x_t} P(x_t,y_1,\ldots,y_T)}
\end{aligned}
$$

$$
\begin{aligned}
P(x_t,y_1,\ldots,y_T) &= P(y_1,\ldots,y_T|x_t)P(x_t) \\
&= P(y_1,\ldots,y_t|x_t)P(y_{t+1},\ldots,y_T|x_t)P(x_t) \\
&= P(y_1,\ldots,y_t,x_t)P(y_{t+1},\ldots,y_T|x_t) \\
&= \alpha_t(x_t)\beta_t(x_t)
\end{aligned}
$$

- So

$$
\begin{aligned}
\gamma_t(x_t) &= \frac{P(x_t,y_1,\ldots,y_T)}{\sum_{x_t} P(x_t,y_1,\ldots,y_T)} \\
&= \frac{\alpha_t(x_t)\beta_t(x_t)}{\sum_{x_t} \alpha_t(x_t)\beta_t(x_t)}
\end{aligned}
$$

Optimal State Sequence

- Different from concatenation of optimal states

$$x_t^* = \arg\max_x P(x_t|Y) \qquad X^* = \arg\max_X P(X|Y)$$

$$X^* \neq (x_1^*, \ldots, x_T^*)$$

- Viterbi Algorithm
  - Exploit Decomposable cost function
  - Dynamic Programming

Exploit conditional independence

$$
\begin{aligned}
& P(x_1, \ldots, x_{t-2}, x_{t-1}, x_t = i, y_1, \ldots, y_t) \\
= \quad & P(x_1, \ldots, x_{t-2}, y_1, \ldots, y_t | x_t = i, x_{t-1}) P(x_t, x_{t-1}) \\
= \quad & P(x_1, \ldots, x_{t-2}, y_1, \ldots, y_{t-1}, y_t | x_t = i, x_{t-1}) P(x_t | x_{t-1}) P(x_{t-1}) \\
= \quad & P(x_1, \ldots, x_{t-2}, y_1, \ldots, y_{t-1} | x_{t-1}) P(y_t | x_t) P(x_t | x_{t-1}) P(x_{t-1}) \\
= \quad & P(x_1, \ldots, x_{t-2}, x_{t-1}, y_1, \ldots, y_{t-1}) P(y_t | x_t) P(x_t | x_{t-1})
\end{aligned}
$$

Viterbi Algorithm: recursion

- Introduce

$$\delta_t(i) = \max_{x_1,\ldots,x_{t-1}} P(x_1,\ldots,x_{t-1},x_t=i,y_1,\ldots,y_t)$$

- Express recursively:

$$\begin{aligned}
\delta_t(i) &= \max_{x_1,\ldots,x_{t-1}} P(x_1,\ldots,x_{t-1},x_t=i,y_1,\ldots,y_t) \\
&= \max_{x_1,\ldots,x_{t-1}} P(x_1,\ldots,x_{t-2},x_{t-1},y_1,\ldots,y_{t-1})P(y_t|x_t=i)P(x_t=i|x_{t-1}) \\
&= \max_{x_1,\ldots,x_{t-2}} \max_j P(x_1,\ldots,x_{t-2},x_{t-1}=j,y_1,\ldots,y_{t-1})P(y_t|x_t=i)P(x_t=i|x_{t-1}=j) \\
&= \max_j \left[\delta_{t-1}(j)P(y_t|x_t=i)P(x_t=i|x_{t-1}=j)\right] P(x_t=i|x_{t-1}=j)
\end{aligned}$$

Viterbi Algorithm

- Initialize

$$\delta_1(i) = \pi_i P(y_1|x_1 = i) \qquad \psi_1(i) = 0$$

- Recurse

$$\delta_k(i) = \max_j \left[\delta_{t-1}(j) P(y_t|x_t = i) P(x_t = i|x_{t-1} = j)\right] P(x_t = i|x_{t-1} = j)$$

$$\psi_k(i) = \arg\max_j \left[\delta_{t-1}(j) P(y_t|x_t = i) P(x_t = i|x_{t-1} = j)\right]$$

- Backtrack

$$P^* = \max_j \delta_T(j)$$

$$q_T^* = \arg\max_j \delta_T(j)$$

$$q_t^* = \psi_{t+1}(q_t^* + 1)$$

- Discrete State: Hidden Markov Models
  - Forward-Backward
  - Viterbi
  - **Learning**
- Continuous State
  - Kalman Filter
  - Particle Filters

Three Main problems

- Likelihood Evaluation (Forward-backward)

$$P(Y)$$

- Inference (Viterbi)

$$\arg\max_X P(X|Y)$$

- **Parameter Estimation (EM)**

$$\arg\max_\theta P(Y|\theta) \qquad \theta = \{A, \mu_{1\ldots N}, \Sigma_{1\ldots N}, \pi\}$$

Full observation log-likelihood

- Hidden Variables: State Sequence
- Full observations: Observations + HVs.
- Full observation likelihood

$$
\begin{aligned}
P(Y, X|\theta) &= P(Y|X, \theta)P(X|\theta) \\
&= \prod_{t=1}^{T} P(y_t|x_t, \theta)P(x_1) \prod_{i=1}^{M} \prod_{t=1}^{T-1} P(x_{t+1}|x_t, \theta) \\
&= \prod_{t=1}^{T} \prod_{i=1}^{M} P(y_t|\mu_i, \Sigma_i)^{[x_t=i]} \pi_i^{[x_1=i]} \prod_{t=1}^{T-1} \prod_{i=1}^{M} \prod_{j=1}^{M} a_{i,j}^{[x_{t+1}=i, x_t=j]}
\end{aligned}
$$

- Full Observation log-likelihood:

$$
\begin{aligned}
\log P(Y, X|\theta) &= \sum_{i=1}^{T} \sum_{i=1}^{M} [x_t = i] \log P(y_t|\mu_i, \Sigma_i) + \sum_{i=1}^{M} [x_1 = 1] \log(\pi_i) \\
&\quad + \sum_{t=1}^{T-1} \sum_{i=1}^{M} \sum_{j=1}^{M} [x_{t+1} = i, x_t = j] \log(a_{i,j})
\end{aligned}
$$

Parameter Estimates for Full Observations

$$
\begin{aligned}
\mu_i^* &= \frac{\sum_{t=1}^{T} [x_t = i] y_t}{\sum_{t=1}^{T} [x_t = i]} \\
\Sigma_i^* &= \frac{\sum_{t=1}^{T} [x_t = i] (y_t - \mu_i^*) (y_t - \mu_i^*)^T}{\sum_{t=1}^{T} [x_t = i]} \\
a_{i,j} &= \frac{\sum_{t=1}^{T-1} [x_{t+1} = i, x_t = j]}{\sum_{t=1}^{T-1} 1}
\end{aligned}
$$

Expectation Maximization

- E-step:
  - Expectation of Full-observation log-likelihood
- M-step:
  - Maximization w.r.t parameters

E-step

- Use current parameter estimate
- Run forward-backward algorithm
- Form γ
- Plug in full-observation log-likelihood

$$< \log P(Y, X|\theta) > = \sum_{i=1}^{T}\sum_{i=1}^{M} < [x_t = i] > \log P(y_t|\mu_i, \Sigma_i) + \sum_{i=1}^{M} < [x_1 = i] > \log(\pi_i) + \sum_{t=1}^{T-1}\sum_{i=1}^{M}\sum_{i=1}^{M} < [x_{t+1} = i, x_t = j] > a_{i,j}$$

$$\begin{aligned} q_t(i) &\equiv < [x_t = i] > \\ &= P(x_t = i|Y)) \\ &= \gamma_t(x_t) = \frac{\alpha_t(x_t)\beta_t(x_t)}{\sum_{x_t} \alpha_t(x_t)\beta_t(x_t)} \end{aligned}$$

Posterior on joint states

$$
\begin{aligned}
q_t(i,j) &\equiv < [x_{t+1} = i, x_t = j] > \\
&= P(x_{t+1} = i, x_t = j | Y) \\
&= \frac{P(Y | x_{t+1} = i, x_t = j) P(x_{t+1} = i, x_t = j)}{P(Y, \theta)} \\
&= \frac{P(y_1, \ldots, y_t | x_t = j) P(y_{t+1}, \ldots, y_T | x_{t+1} = i) P(x_{t+1} = i | x_t = j) P(x_t = j)}{P(Y, \theta)} \\
&= \frac{P(y_1, \ldots, y_t, x_t = j) P(y_{t+1} | x_{t+1} = i) P(y_{t+2}, \ldots, y_T | x_{t+1} = i) P(x_{t+1} = i | x_t = j)}{P(Y, \theta)} \\
&= \frac{\alpha_t(j) P(y_{t+1} | x_{t+1} = i) \beta_{t+1}(i) P(x_{t+1} = i | x_t = j)}{P(Y, \theta)}
\end{aligned}
$$

M-step

- Replace hidden variables with their expectations

$$\mu_i^* = \frac{\sum_{t=1}^{T} q_t(i) y_t}{\sum_{t=1}^{T} q_t(i)}$$

$$\Sigma_i^* = \frac{\sum_{t=1}^{T} q_t(i) \left(y_t - \mu_i^*\right) \left(y_t - \mu_i^*\right)^T}{\sum_{t=1}^{T} q_t(i)}$$

$$a_{i,j} = \frac{\sum_{t=1}^{T-1} q_t(i,j)}{\sum_{t=1}^{T-1} 1}$$

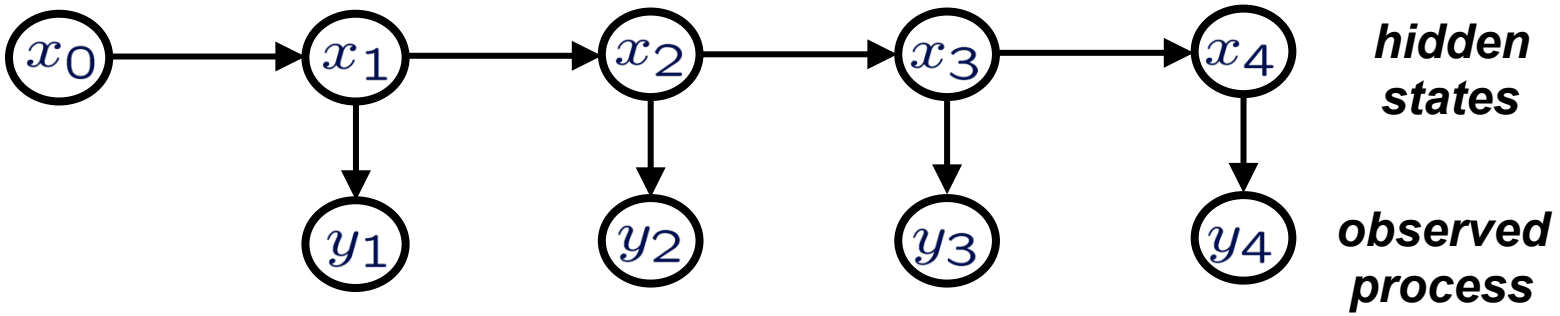
5th Lecture Layout

- Discrete State: Hidden Markov Models
  - Forward-Backward
  - Viterbi
  - Learning
- **Continuous State**
  - Kalman Filter
  - Particle Filters

Linear Dynamical Systems

- Noise in State, noise in Observations

$$
\begin{aligned}
x_t &= Ax_{t-1} + v_t, \quad v_t \propto N(0, \Sigma) \\
y_t &= Cx_t + w_t, \quad w_t \propto N(0, Q) \\
x_1 &= \mu_0 + u, \quad u \propto N(0, V)
\end{aligned}
$$



- Identical dependencies among random variables
  - But now integrals instead of summations
  - All distributions are Gaussian

Kalman Filter Recursion

- State estimate

$$\begin{aligned}\alpha_t(x_t) &= P(x_t|y_1,\ldots,y_t) \\ &= \int_{x_1}\ldots\int_{x_{t-1}} P(x_1,\ldots,x_t|y_1,\ldots,y_t)dx_1\ldots dx_{t-1}\end{aligned}$$

- As in HMMs, come up with a recursion

$$c_t\alpha_t(x_t) = P(y_t|x_t)\underbrace{\int \alpha_{t-1}(x_{t-1})P(x_t|x_{t-1})dx_{t-1}}_{P(x_t|y_1,\ldots,y_{t-1})}$$

- State estimate = prediction x correction
- All distributions are Gaussian: analytic expressions

Prediction distribution

- Write in terms of previous state posterior

$$P(x_t|y_1,\ldots,y_{t-1}) = N(x_t|\mu_{t|t-1},\Sigma_{t|t-1})$$

$$\begin{aligned}\mu_{t|t-1} &= E(x_t|y_1,\ldots,y_{t-1}) \\ \Sigma_{t|t-1} &= E((x_t-\mu_{t|t-1})(x_t-\mu_{t|t-1})^T|y_1,\ldots,y_{t-1})\end{aligned}$$

$$\begin{aligned}x_{t+1} &= Ax_t + v_t, v_t \propto N(0,\Sigma) \\ \mu_{t|t-1} &= A\mu_{t-1|t-1} \\ \Sigma_{t|t-1} &= A\Sigma_{t-1|t-1}A^T + \Sigma\end{aligned}$$

Posterior distribution

- Posterior distribution

$$\begin{aligned} P(x_t|y_1,\ldots,y_t) &\propto P(x_t|y_1,\ldots,y_{t-1})P(y_t|x_t) \\ &= N(x_t|\mu_{t|t},\Sigma_{t|t}) \end{aligned}$$

$$\begin{aligned} \mu_{t|t} &= E(x_t|y_1,\ldots,y_t) \\ \Sigma_{t|t} &= E((x_t-\mu_{t|t})(x_t-\mu_{t|t})^T|y_1,\ldots,y_t) \end{aligned}$$

- Consider joint distribution of state and observation

$$P(x_t,y_t|y_1,\ldots,y_{t-1})$$

Joint distribution of state and observation

- We have distribution for x, need to consider y

$$\begin{aligned}\hat{y}_t &= E(y_t|y_1,\ldots,y_{t-1}) \\ &= E(Cx_t + w_t|y_1,\ldots,y_{t-1}) \\ &= C\mu_{t|t-1}\end{aligned}$$

$$\begin{aligned}E((y_t-\hat{y}_t)(y_t-\hat{y}_t)^T|y_1,\ldots,y_{t-1}) &= E((Cx_t+w_t-C\mu_{t|t-1})(Cx_t+w_t-C\mu_{t|t-1})^T|y_1,\ldots,y_{t-1}) \\ &= C\Sigma_{t|t-1}C^T + Q\end{aligned}$$

- So

$$E((y_t-\hat{y}_t)(x_t-\hat{x}_t)^T|y_1,\ldots,y_{t-1}) = C\Sigma_{t|t-1}$$

$$E\left(\begin{bmatrix} x_t \\ y_t \end{bmatrix}\right) = \begin{bmatrix} \mu_{t|t-1} \\ C\mu_{t|t-1} \end{bmatrix}$$

$$Cov\left(\begin{bmatrix} x_t \\ y_t \end{bmatrix}\right) = \begin{bmatrix} \Sigma_{t|t-1} & \Sigma_{t|t-1}C^T \\ C\Sigma_{t|t-1} & C\Sigma_{t|t-1}C^T + Q \end{bmatrix}$$

Posterior distribution on state

- Condition on new observation y

$$
\begin{aligned}
K &= \Sigma_{t|t-1} C^T \left(C \Sigma_{t|t-1} C^T + Q \right)^{-1} \\
\mu_{t|t} &= \mu_{t|t-1} + K(y_t - C\mu_{t|t-1}) \\
\Sigma_{t|t} &= \Sigma_{t|t-1} - KC\Sigma_{t|t-1}
\end{aligned}
$$

1D Discrete-time Brownian motion

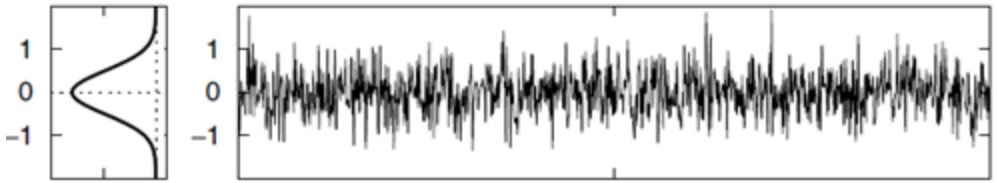
Dynamical model: $x_{t+1} = x_t + w_t$

Deterministic system driven by random excitation

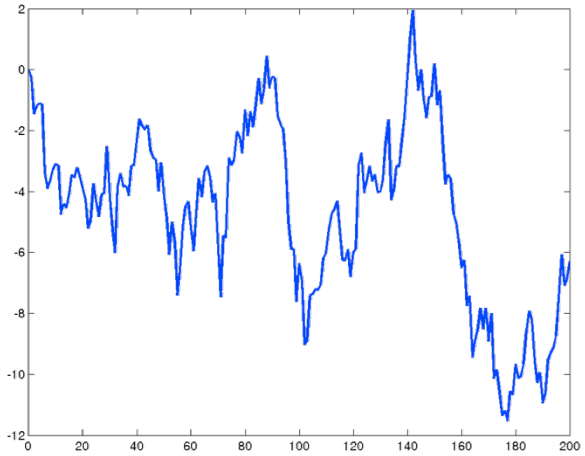
Input: White Gaussian Noise $w_t \propto N(0, \sigma)$

Output: Brownian motion

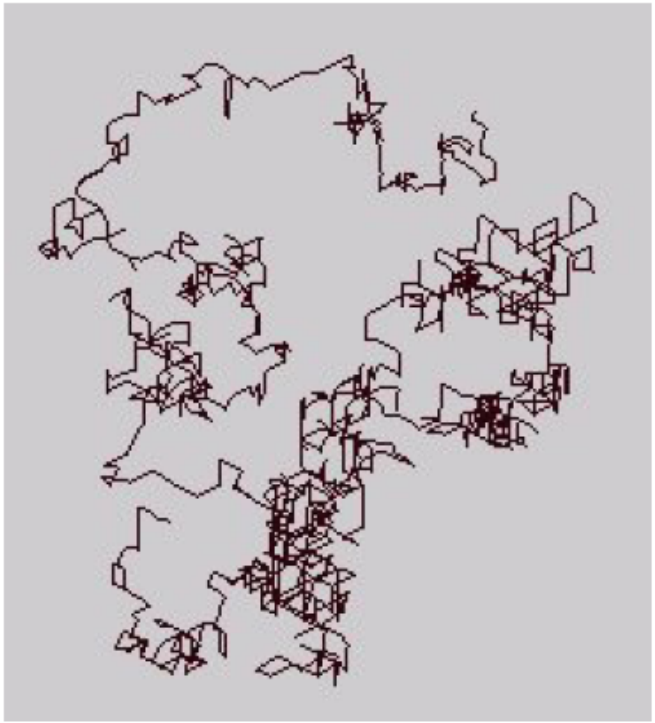
Realization of input:

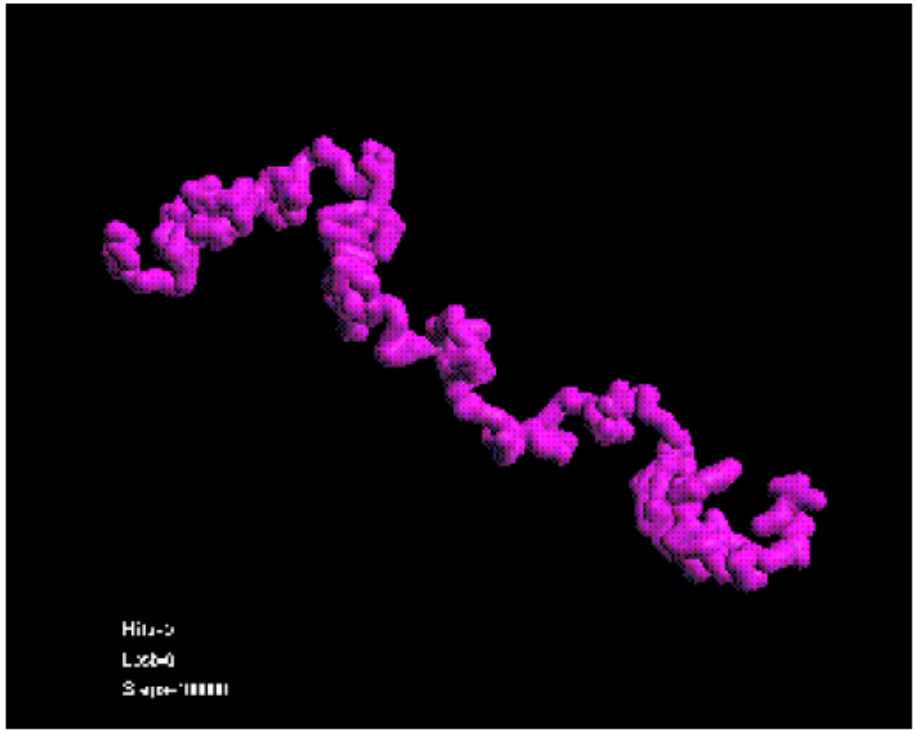


Realization of output:



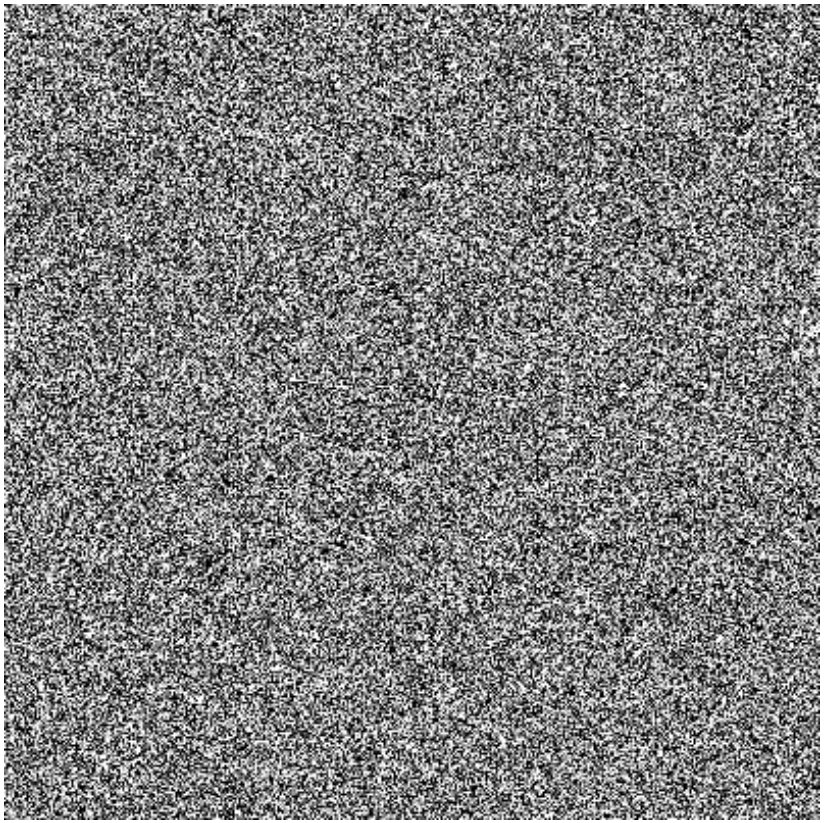
Example: Randomly Drifting Points



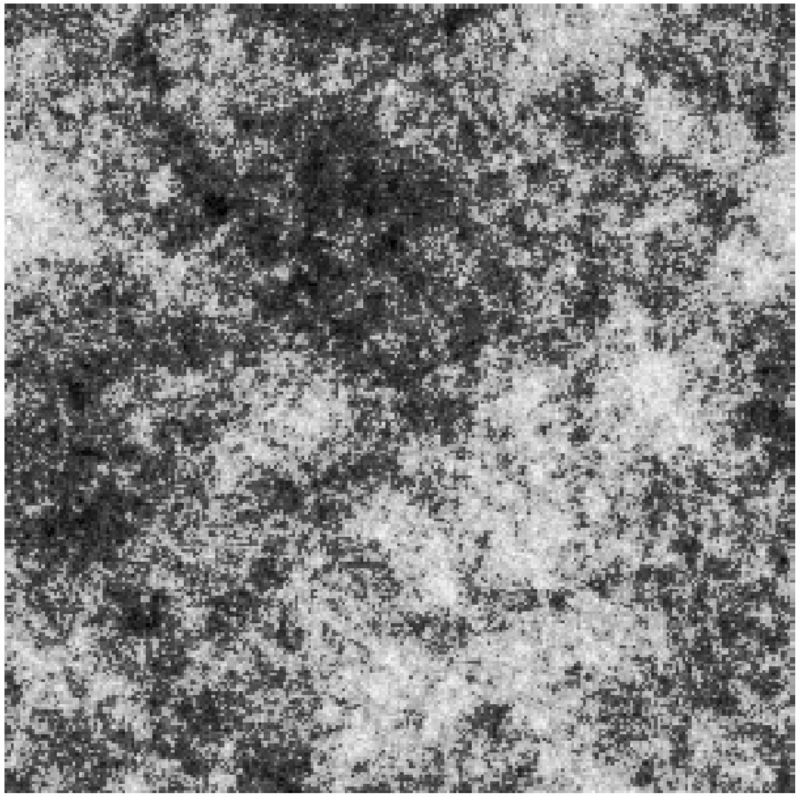


eic.nist.gov/lipman/sciviz/images/random3.gif
http://www.grunch.net/synergetics/images/rand
3.jpg

2D Brownian motion as a model for natural images



2D White Gaussian Noise



2D BM (stochastic fractal)

Generative model for Brownian motion

Synthesis equation: $I(t) = \sum_k \frac{\sigma}{k} c_k \cos(2\pi(kt) + \phi_k)$

$$c_k \sim N(0,1) \quad \phi_k \sim U[0,1]$$

Coefficients Basis

2D: $I(x,y) = \sum_k \sum_m \frac{1}{\sqrt{2c(k^2+m^2)}} c_{k,m} \cos(2\pi(kx+my) + \phi_{k,m})$

$$c_{k,m} \sim N(0,1) \quad \phi_{k,m} \sim U[0,1]$$

Fourier Synthesis Equation: use sinusoidals as signal basis

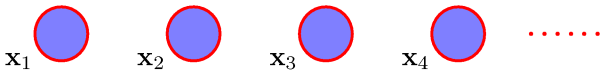
PCA: `Customized' basis for stochastic process (Karhunen-Loeve)

PCA on Brownian motion: Sinusoidal basis

Random Variable Dependencies

Stochastic process: sequence of RVs $X = (x_1, x_2, \ldots, x_T)$

Independent RVs



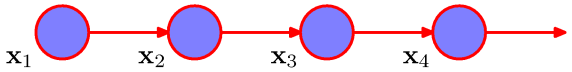
(WGN)

$$P(x_1, x_2, \ldots, x_T) = \prod_{i=1}^{T} P(x_i)$$

Dependent RVs

$$x_{t+1} = x_t + w_t$$

(Brownian Motion)



$$P(x_{t+1}|x_t, x_{t-1}, x_{t-2}, \ldots x_1) = P(x_{t+1}|x_t)$$ (1st-order Markov)

$$= P(x_{t+1}|x_t)P(x_t|x_{t-1}, \ldots x_1)P(x_{t-1}, \ldots x_1)$$

$$= \left(\prod_{i=1}^{t} P(x_{i+1}|x_i)\right) P(x_1)$$

2D: $P(x_0, x_1, \ldots, x_N) = \prod_{(i,j)\in\mathcal{C}} \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(x_i - x_j)^2}{2\sigma^2}\right)$