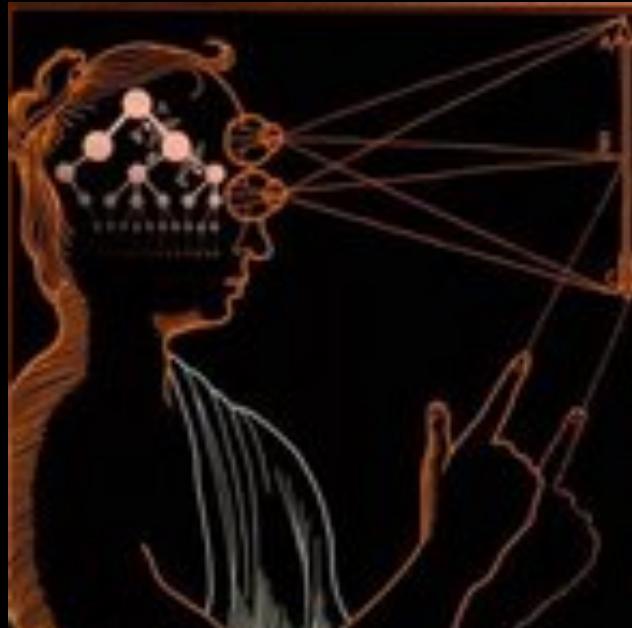


Machine Learning for Computer Vision

26 October, 2012
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



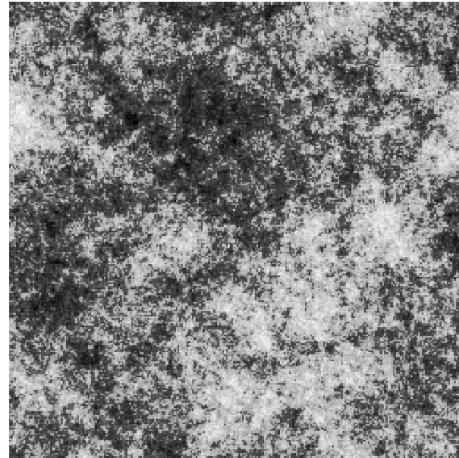
Introduction

Brownian motion

Graphical models

Chain-structured graphical models

Tree-structured graphical models



1D Discrete-time Brownian motion

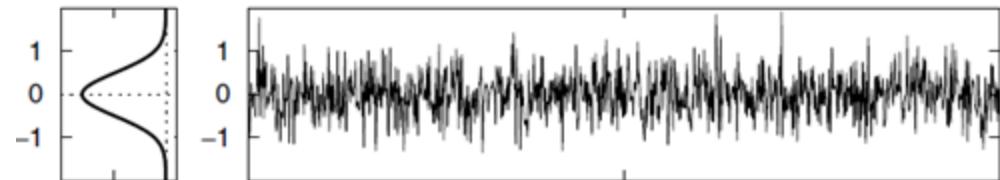
Dynamical model: $x_{t+1} = x_t + w_t$

Deterministic system driven by random excitation

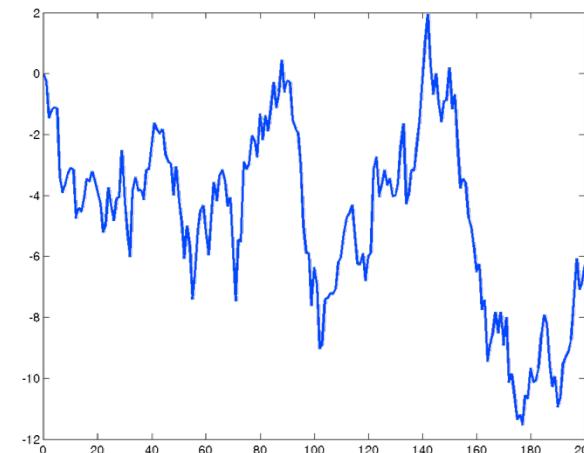
Input: White Gaussian Noise $w_t \propto N(0, \sigma)$

Output: Brownian motion

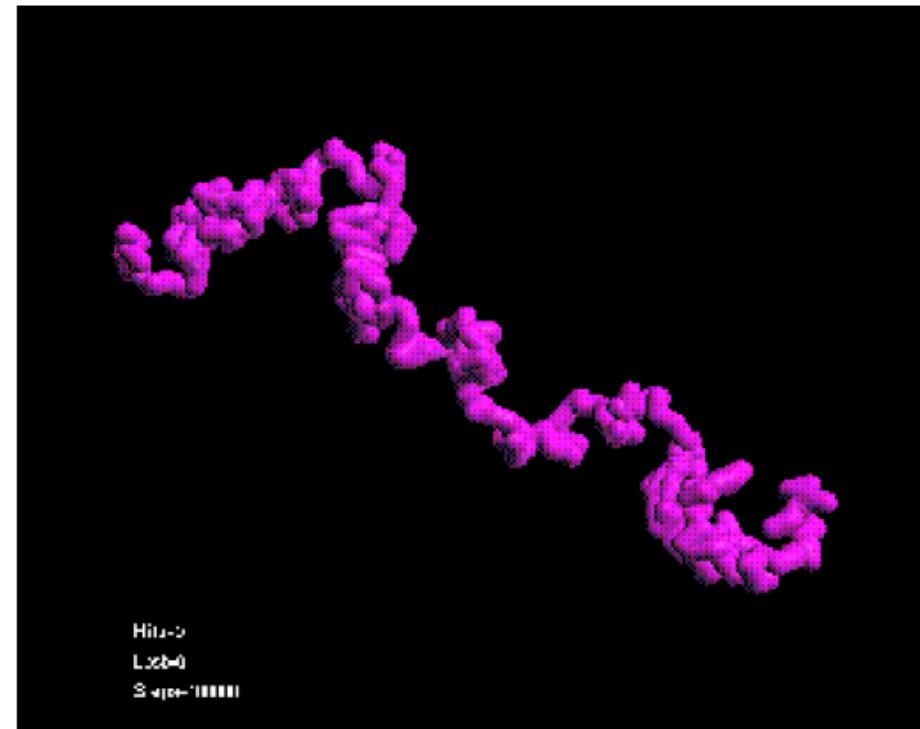
Realization of input:



Realization of output:



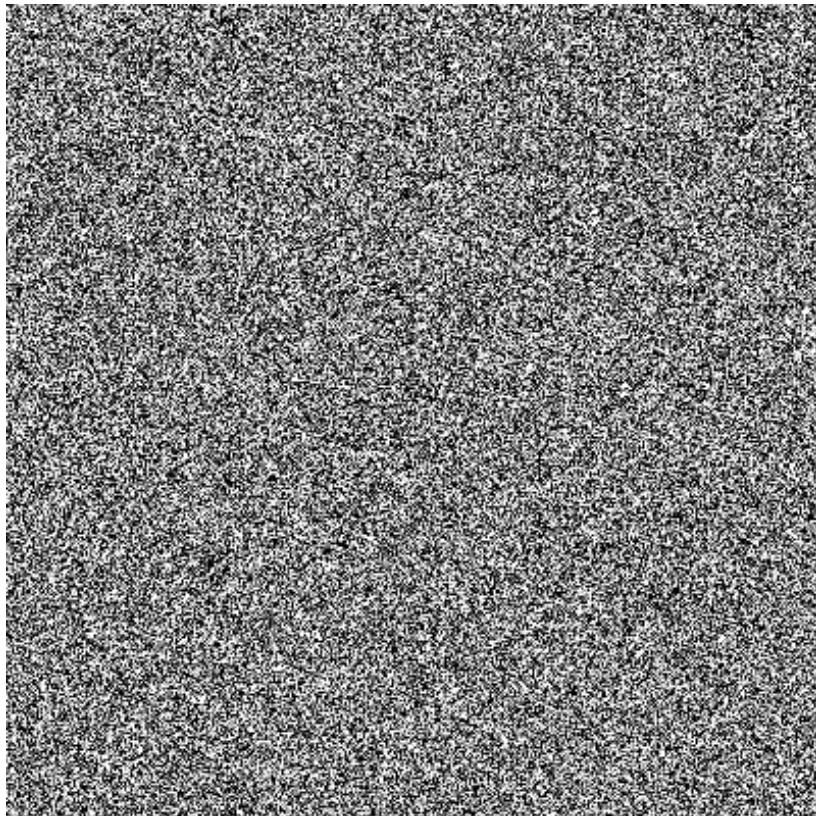
Example: Randomly Drifting Points



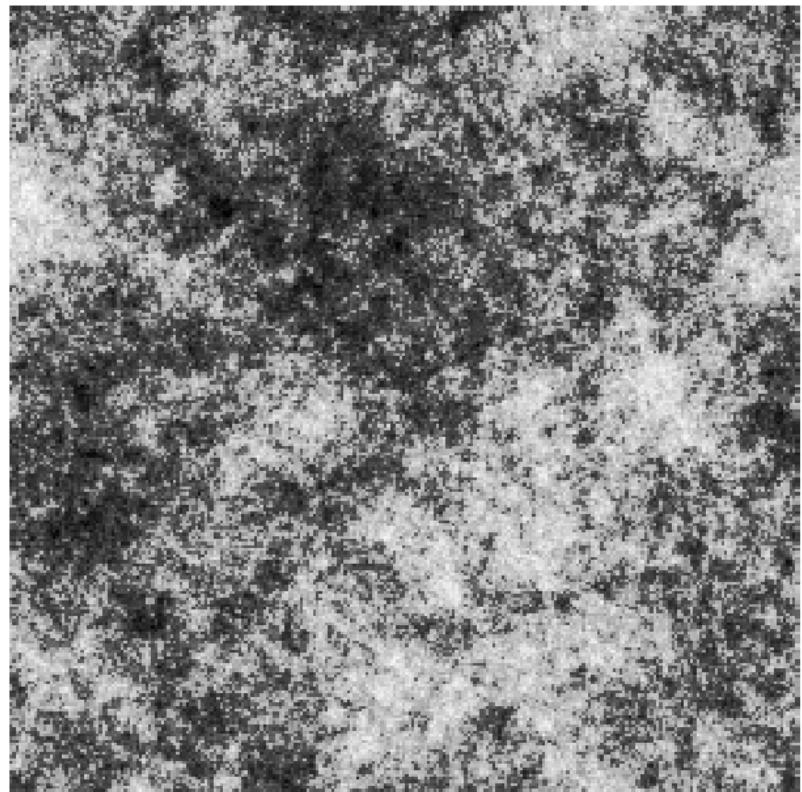
Hit=0
Loc=0
Step=10000

cic.nist.gov/lipman/sciviz/images/random3.gif
<http://www.grunch.net/synergetics/images/random3.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
c_k	$\cos(2\pi(kt) + \phi_k)$
$c_{k,m}$	$\cos(2\pi(kx + my) + \phi_{k,m})$

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 sinusoids as signal basis

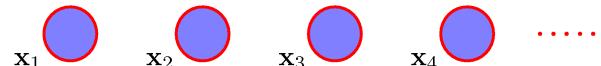
PCA: ‘Customized’ basis for stochastic process (Karhunen-Loeve)

PCA on Brownian motion: Sinusoidal basis

Random Variable Dependencies

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

Independent RVs



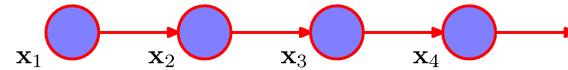
(WGN)

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

Dependent RVs

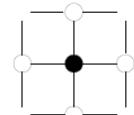
$$x_{t+1} = x_t + w_t$$

(Brownian Motion)



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

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





Lecture outline

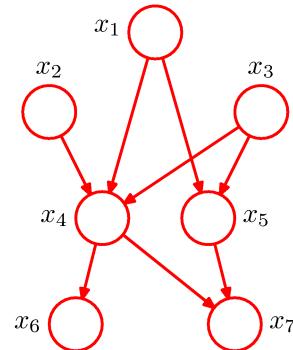
Introduction

Brownian motion

Graphical models

Chain-structured graphical models

Tree-structured graphical models



Graphical models

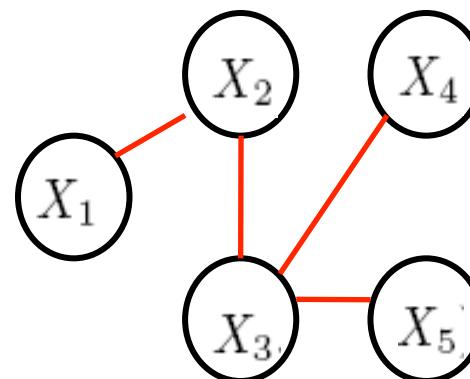
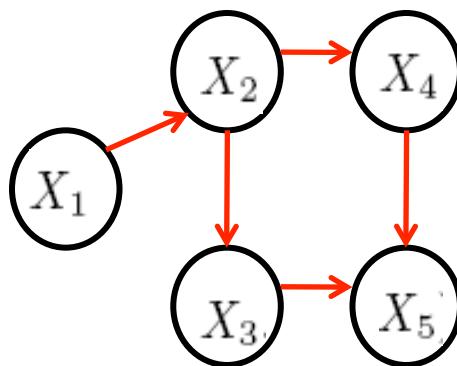
Blend of graph theory and probability

Graph nodes: random variables

Graph edges: relationships

Directed graphs: Bayesian Networks

Undirected graphs: Markov Random Fields



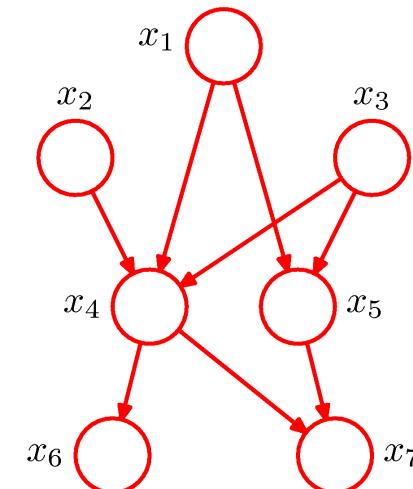
Bayesian Networks

Directed Acyclic Graph $G(V, E)$

Nodes - V : random variables

Edges - E : dependencies

Leave from ‘parents’, arrive at children

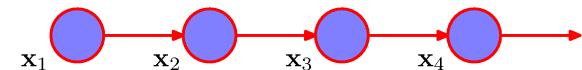


Distribution of each child conditioned on parent: $P(X_v | X_{\pi_v})$

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

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

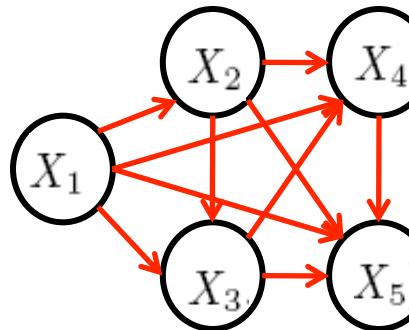
Bayesian network for Brownian motion:



Conditional independencies

We can trivially obtain a bayesian network for a probability distribution

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

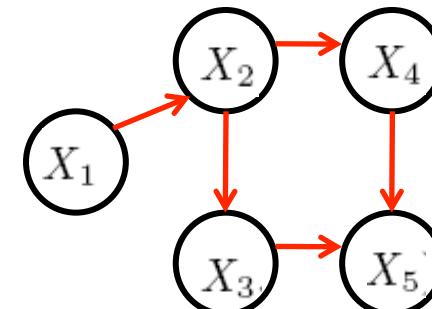


Fully connected graph!

Bayesian net

Non-trivial cases: some conditional independence

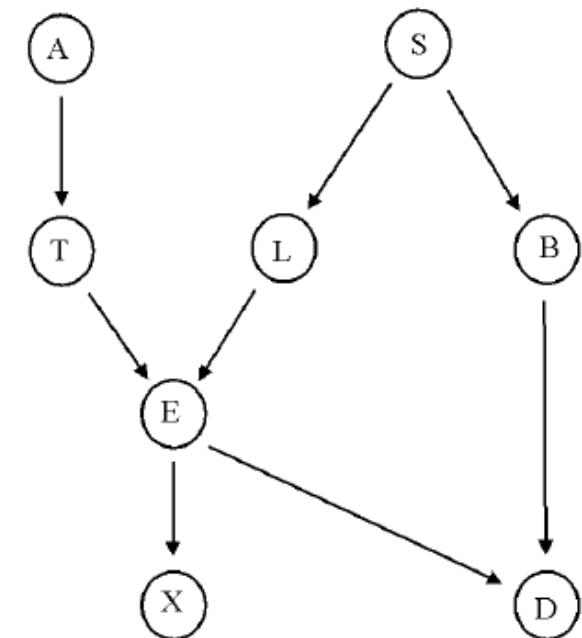
$$\begin{aligned} P(X_3|X_2, X_1) &= P(X_3|X_2) \\ P(X_4|X_3, X_2, X_1) &= P(X_4|X_2) \\ P(X_5|X_1, X_2, X_3, X_4, X_5) &= P(X_5|X_3, X_4) \end{aligned}$$



Main interest: exploit independencies for training & inference

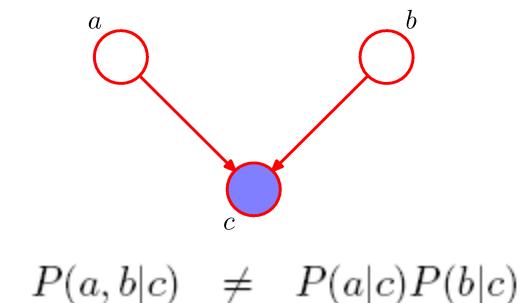
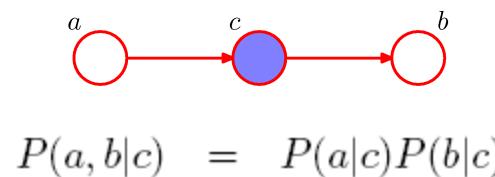
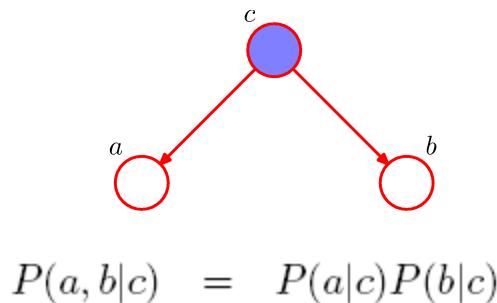
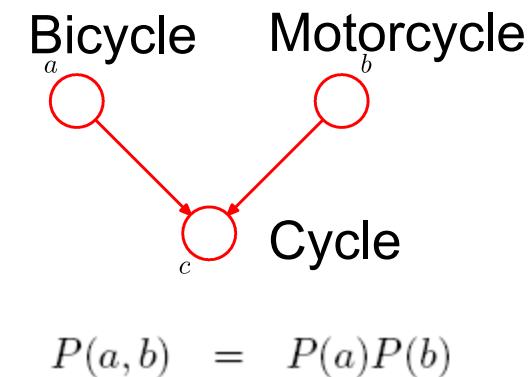
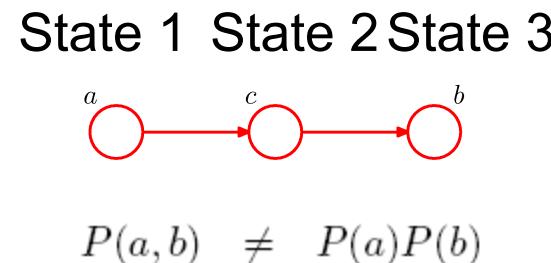
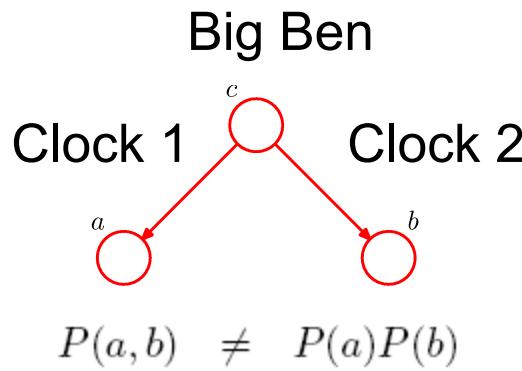
A Bayesian Network-based Expert System

- ‘Asia Network’
 - A: trip to Asia
 - T: tuberculosis
 - S: smoking
 - L: Lung Cancer
 - B: Bronchitis
 - E: Tuberculosis/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



‘Explaining away’



Lecture outline

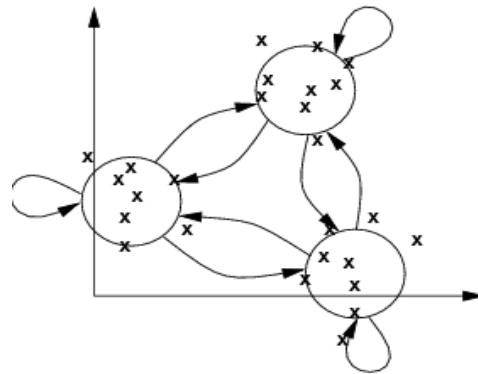
Introduction

Chain-structured graphical models

Hidden Markov Models

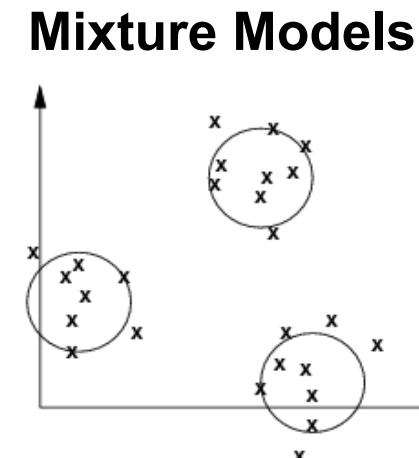
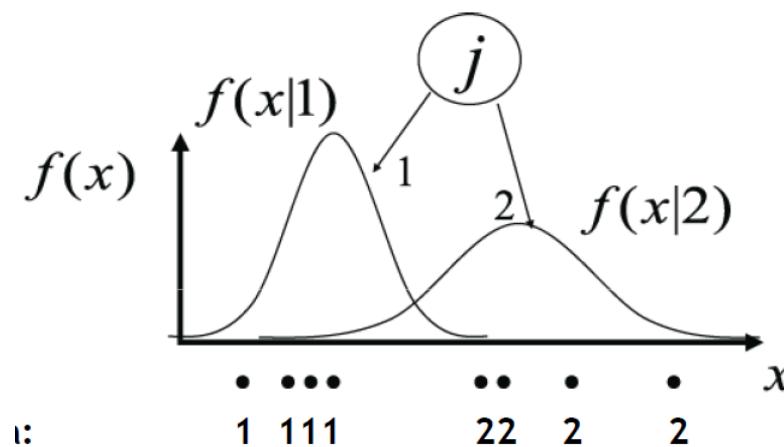
Kalman Filter

Tree-structured graphical models



Hidden variables

Mixture models: combine simple distributions into complex one



Data clustering: side-product of parameter estimation

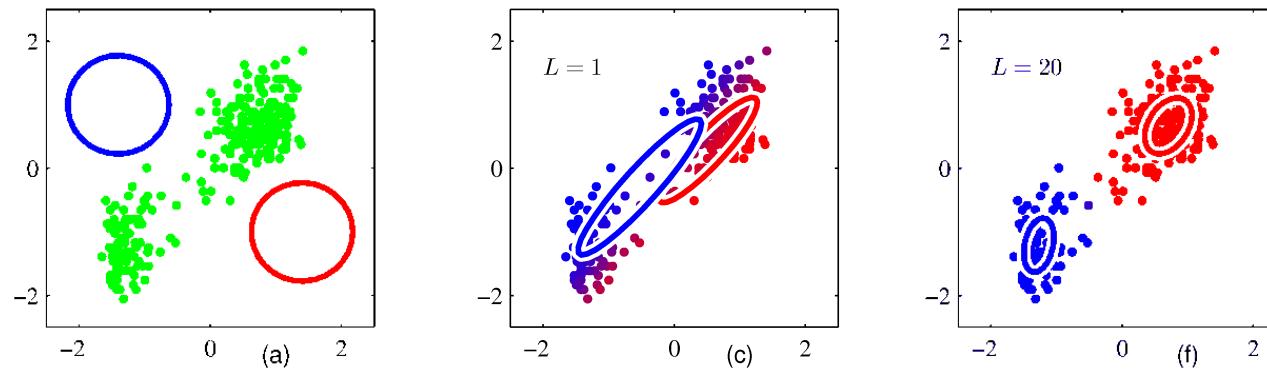
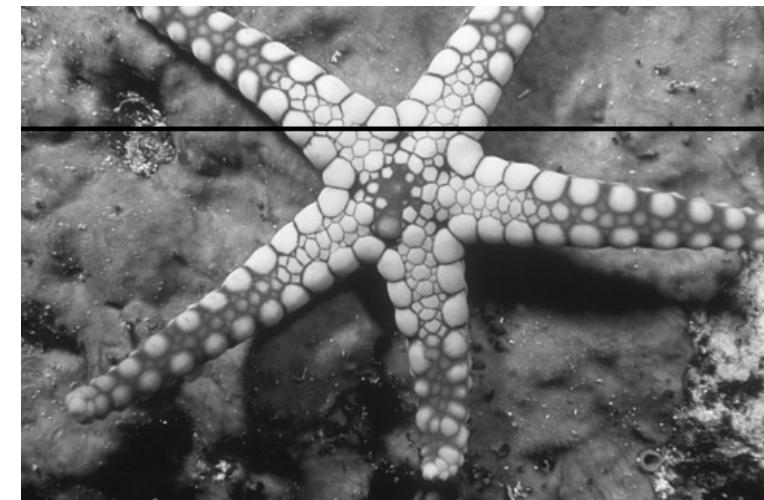
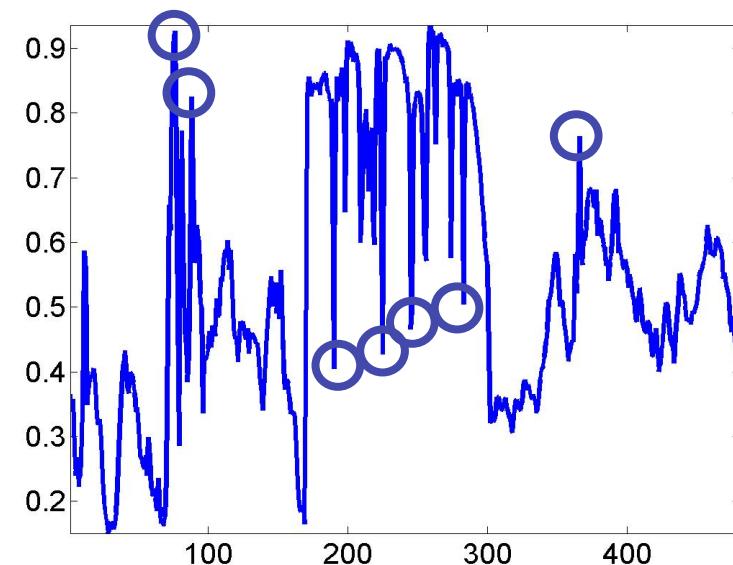


Image clustering - segmentation

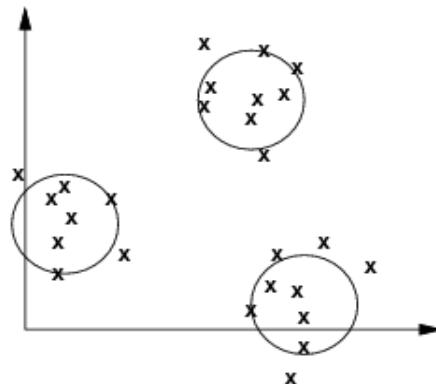


Observations are not independent
(Consecutive points: common region)

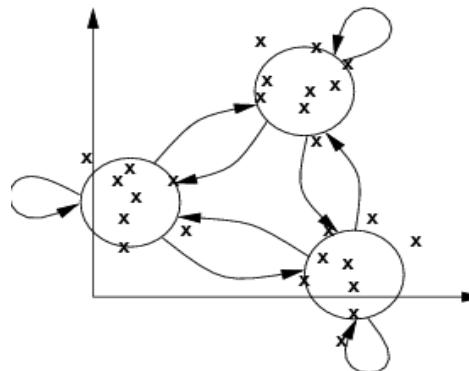


Dependent Hidden variables

Mixture Models

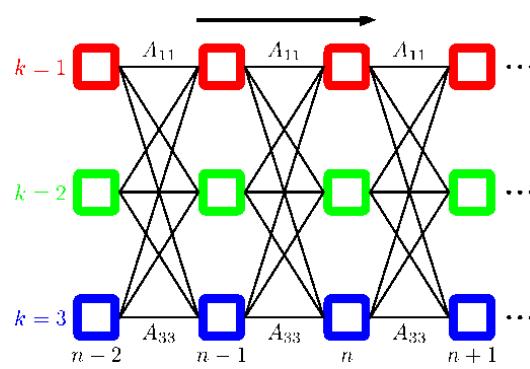


Hiden Markov Models

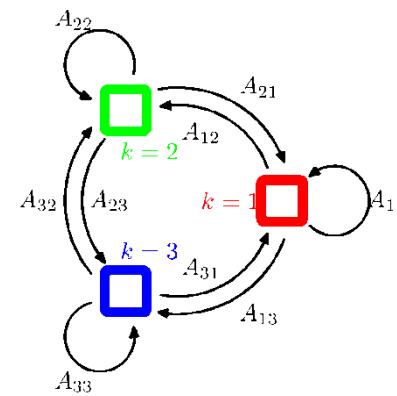


Introduce dependencies between hidden variables

First-order Markov Model



$M \times M$ transition matrix

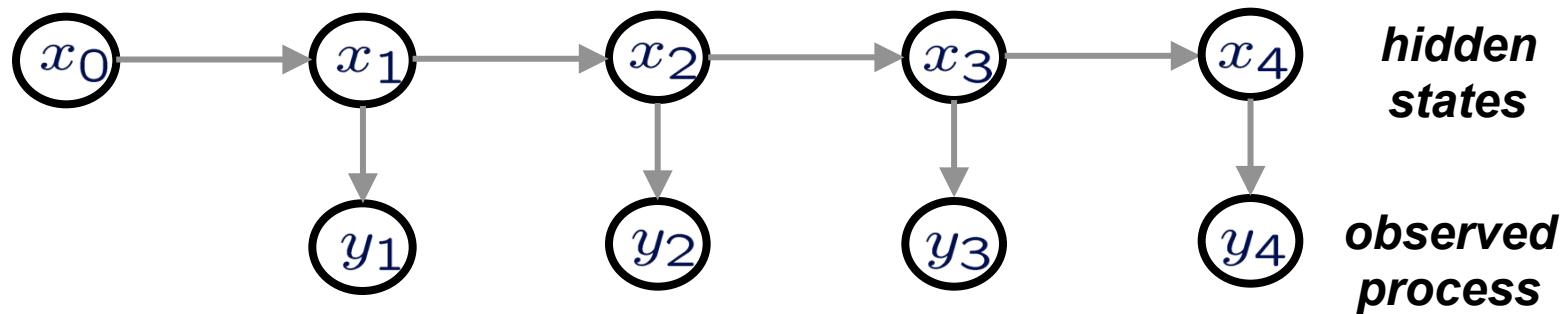


$$A_{i,j} = P(x_{t+1} = i | x_t = j)$$

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

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

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
---	---

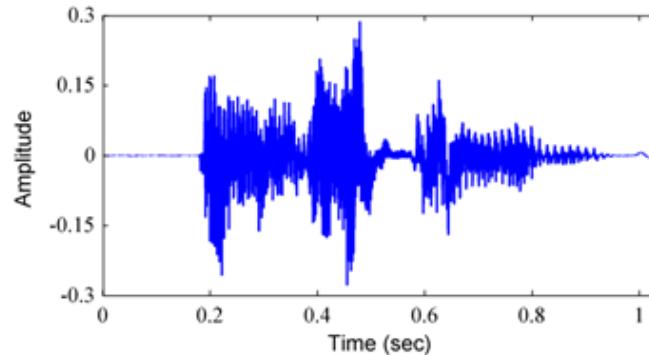
Transition probabilities $\pi(x_{t+1} = l | x_t = m) = \alpha_{l,m}$

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

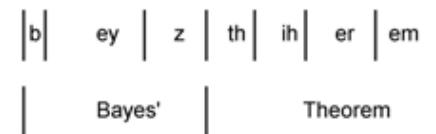
Estimation: Expectation Maximization

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

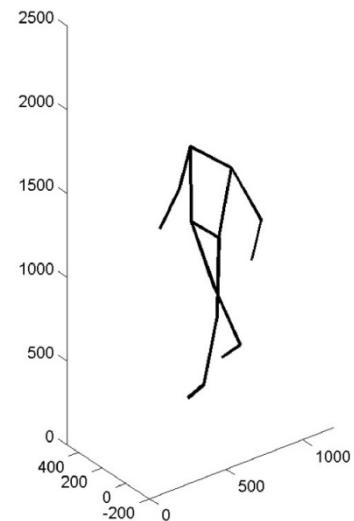
Chain-structured graphical models

Hidden Markov Models

Kalman Filter

Tree-structured graphical models

Loopy graphical models



Detection vs. Tracking



t=1



t=2

...



t=20



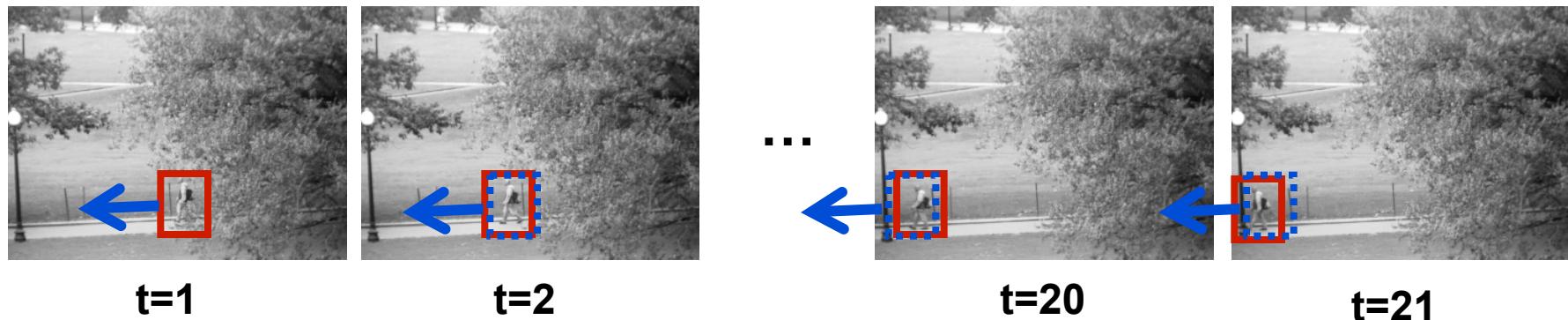
t=21

Detection vs. Tracking



- Detection: each frame independently

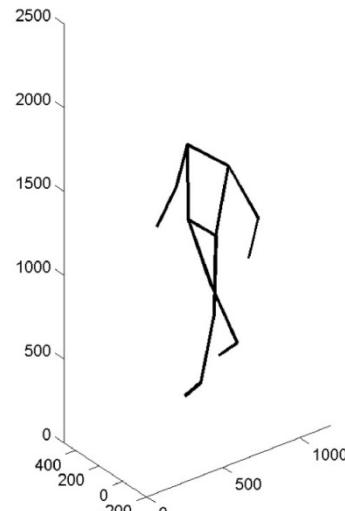
Detection vs. Tracking



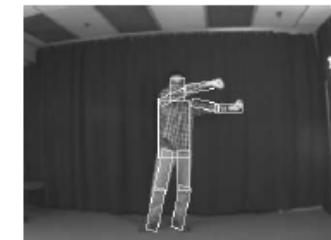
- Tracking with *dynamics*: combine image measurements with our expectation of the object's motion.
 - Restrict search for the object
 - Measurement noise is reduced by trajectory smoothness.

Tracking

- Input: Series of Images
- Output: Estimate of object's motion/action/gesture
- Hidden Variables ('State')
 - Time-dependent
 - Assume linear dynamics



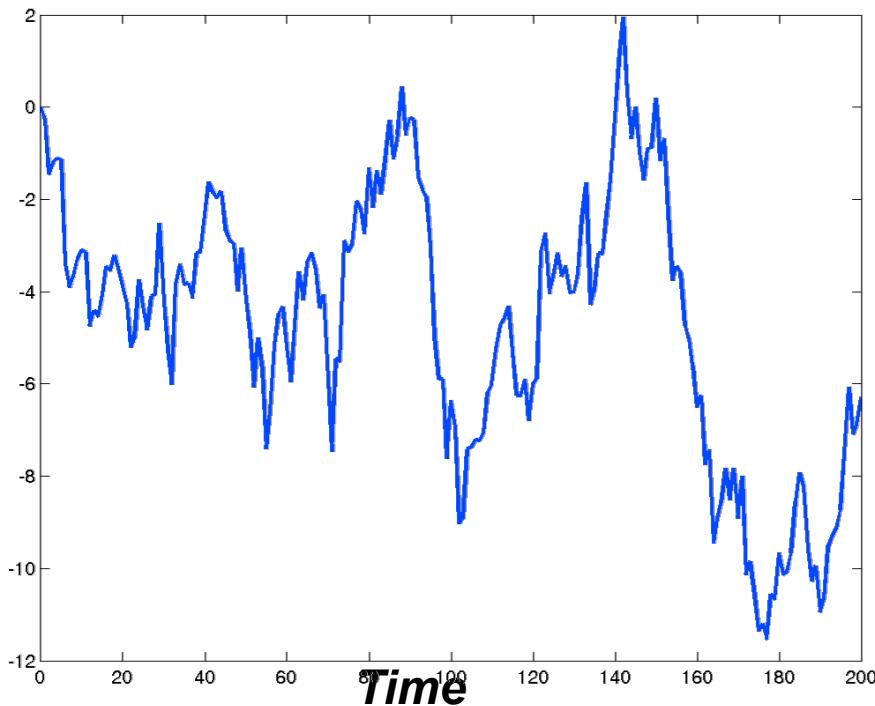
State, X



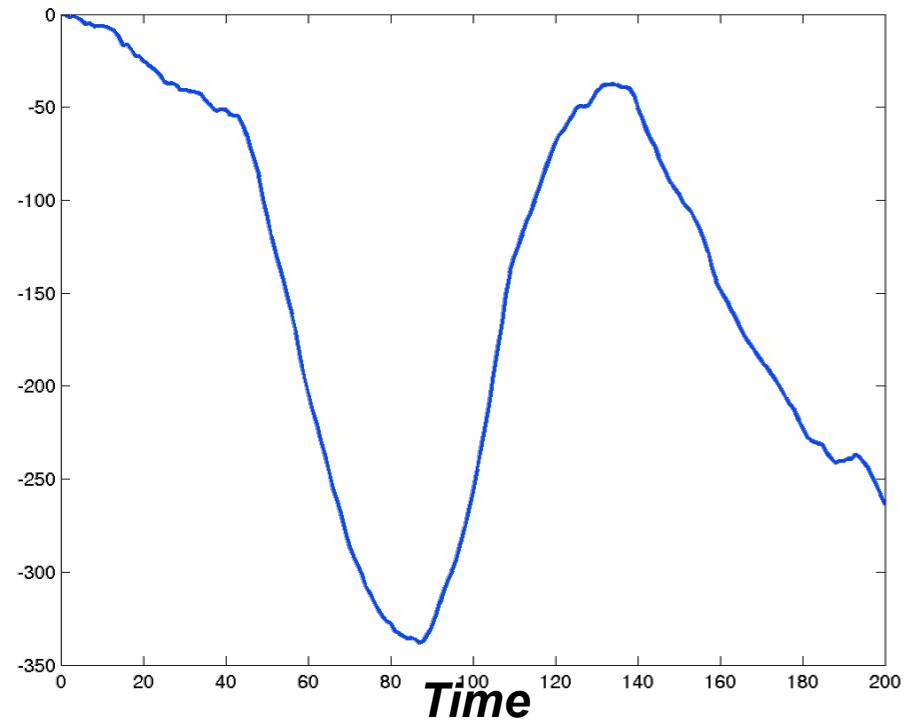
Observations, Y

Simple Linear Dynamics

Brownian Motion



Constant Velocity



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

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

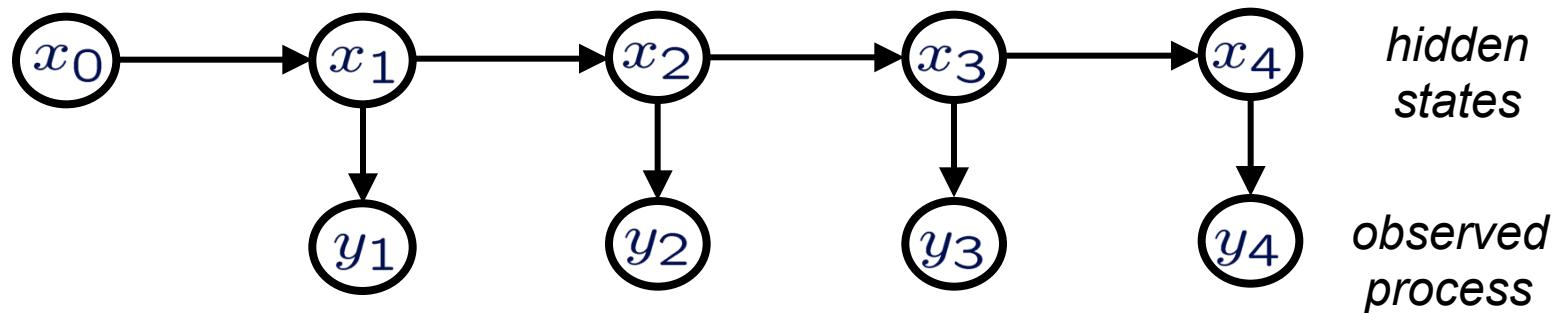
Linear Dynamical Systems

- Noise in State, noise in Observations

$$x_t = Ax_{t-1} + v_t, \quad v_t \sim N(0, \Sigma)$$

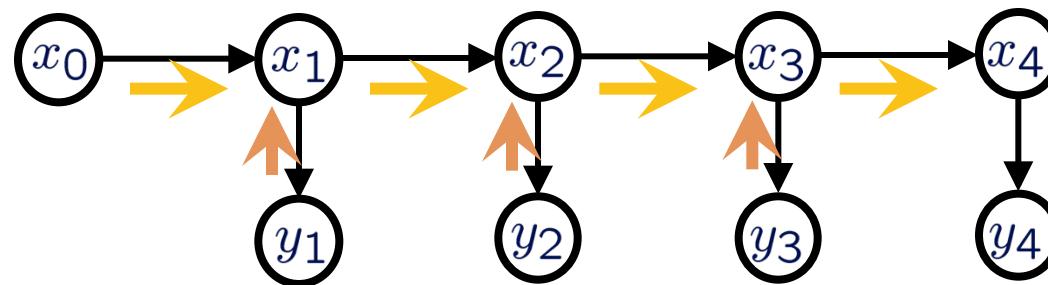
$$y_t = Cx_t + w_t, \quad w_t \sim N(0, Q)$$

$$x_1 = \mu_0 + u, \quad u \sim N(0, V)$$

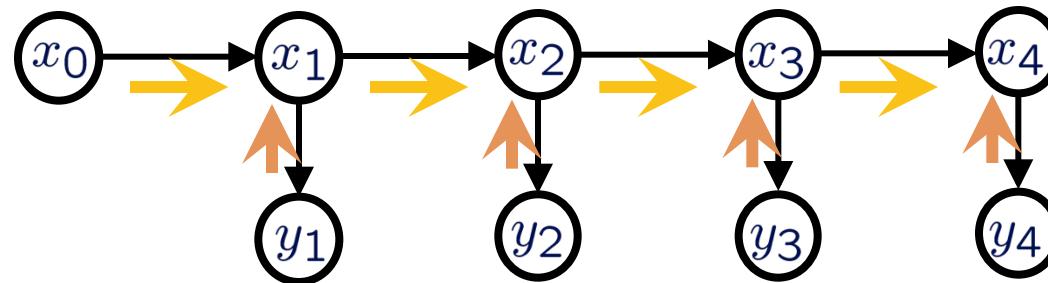


Inference Tasks

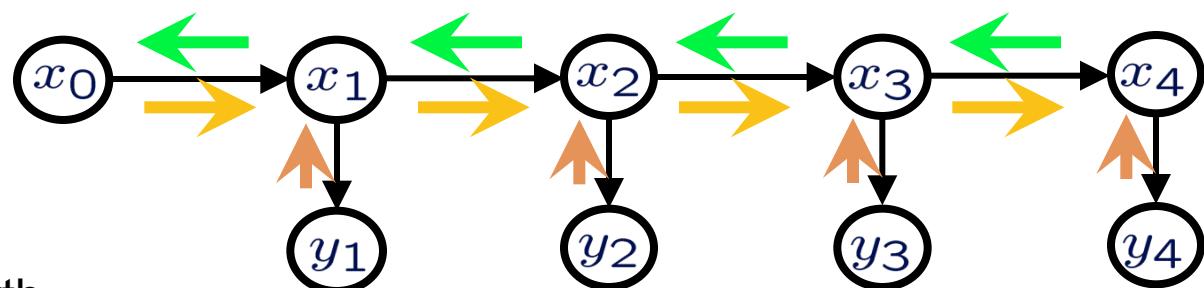
- Prediction



- Filtering
(Kalman Filter)



- Smoothing





Lecture outline

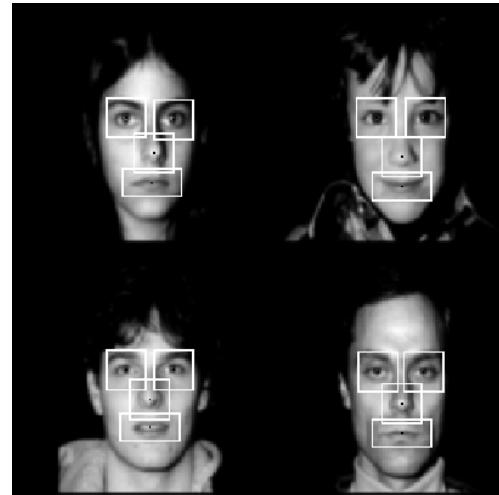
Introduction

Chain-structured graphical models

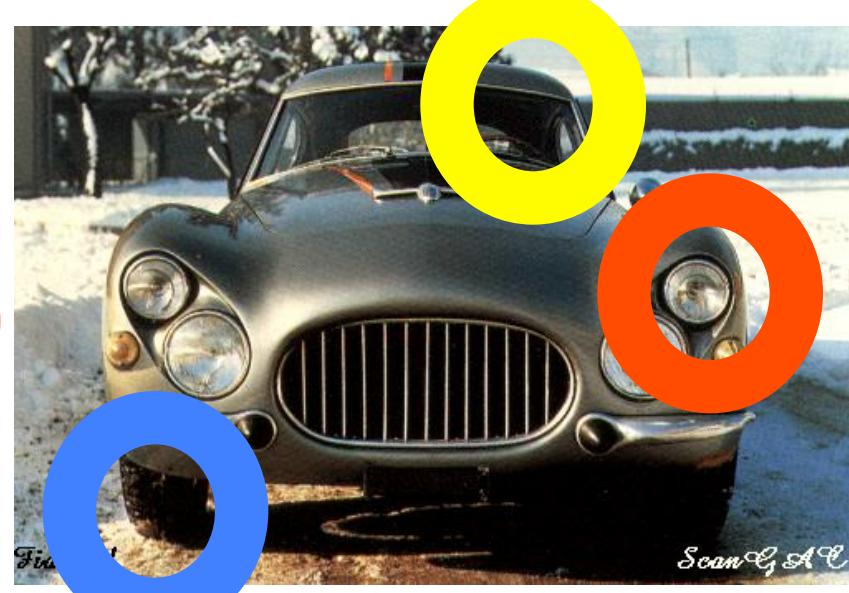
Tree-structured graphical models

Pictorial Structures

Max/Sum-Product algorithm



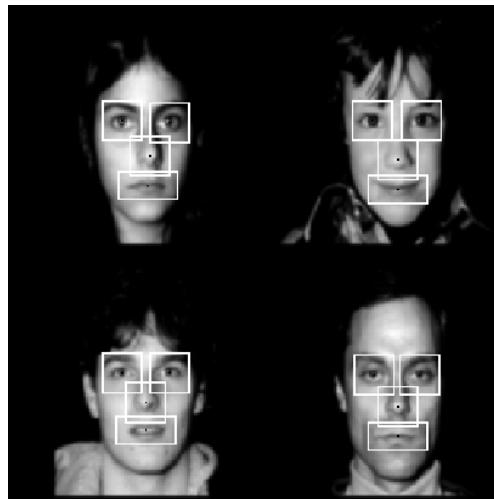
Part-based object models



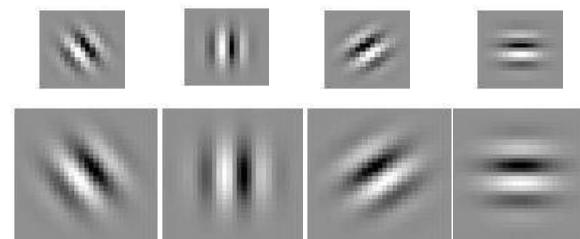
Slide credit: R. Fergus

Unary terms (appearance model)

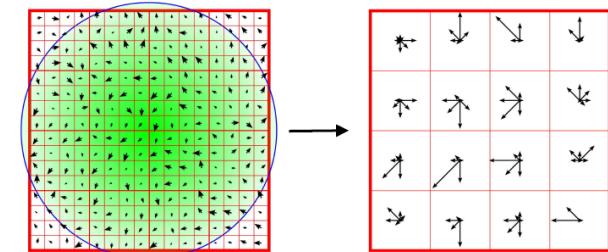
- Consider a patch of the image around the location of the part
- Construct statistical model for appearance at part location



Filterbank responses

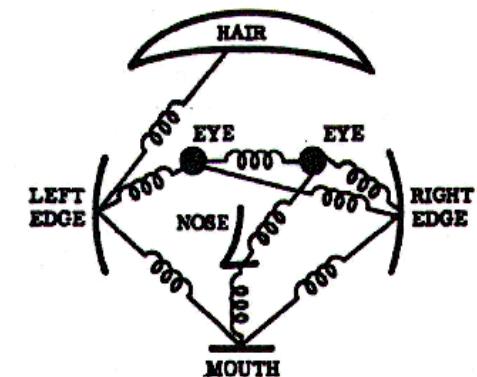


SIFT features



Pairwise terms (deformation model)

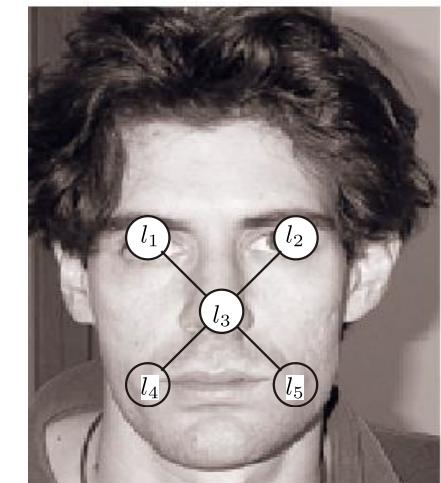
- Deformable objects as physical systems
 - Masses: driven towards good image locations
 - Springs: enforce relations among parts



- 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?



Lecture outline

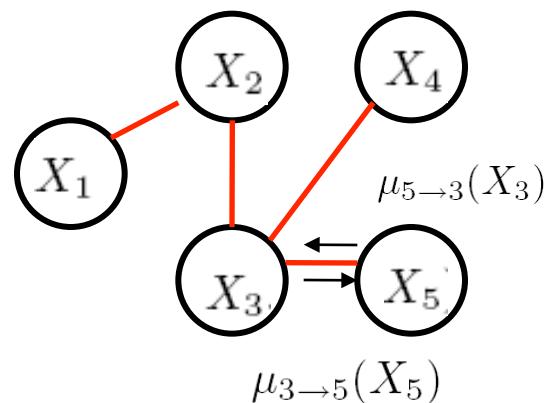
Introduction

Chain-structured graphical models

Tree-structured graphical models

Pictorial Structures

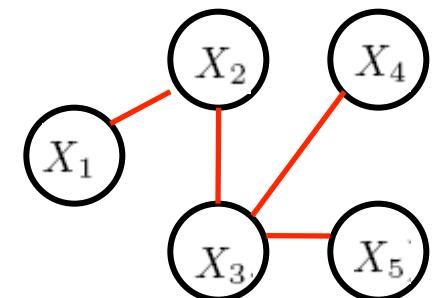
Max/Sum-Product algorithm



Inference on tree-structured Graphs

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

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



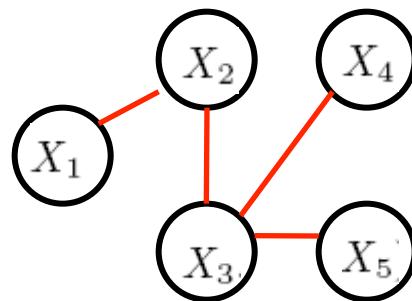
- Brute-force maximization:

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

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

- Exploit factorization

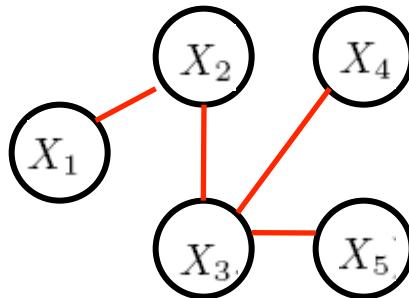
Exploiting the factorization



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

$\max P(X_1)$

Exploiting the factorization



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

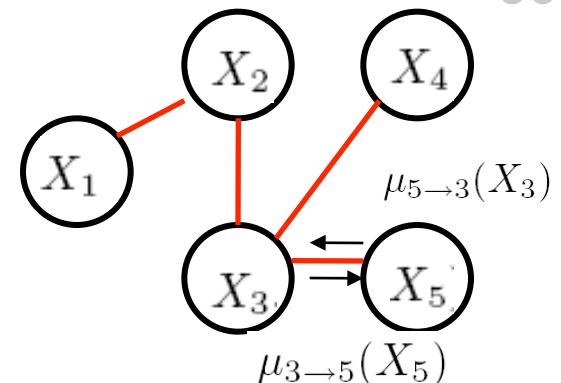
$$\begin{aligned}
 &\max P(X_1) \\
 &= \max_{X_2, X_3, X_4, X_5} \Phi(X_1) \Phi(X_2) \Phi(X_3) \Phi(X_4) \Phi(X_5) \Psi(X_1, X_2) \Psi(X_2, X_3) \Psi(X_3, X_4) \Psi(X_3, X_5) \\
 &= \Phi(X_1) \max_{X_2} \Phi(X_2) \Psi(X_1, X_2) \max_{X_3, X_4} \Phi(X_3) \Psi(X_2, X_3) \Phi(X_4) \Psi(X_3, X_4) \underbrace{\max_{X_5}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

Max-Product algorithm

- Distributed computation on a graph
 - ‘message-passing’
- Message from node i to node j

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

- ‘Given all I know from the rest, this is where you should be’
- Beliefs: $B_j(X_j) = \Phi_j(X_j) \prod_{i \in \mathcal{N}(j)} \mu_{i \rightarrow j}(X_j)$
- At convergence $B_j(X_j) = \max P(X_j)$



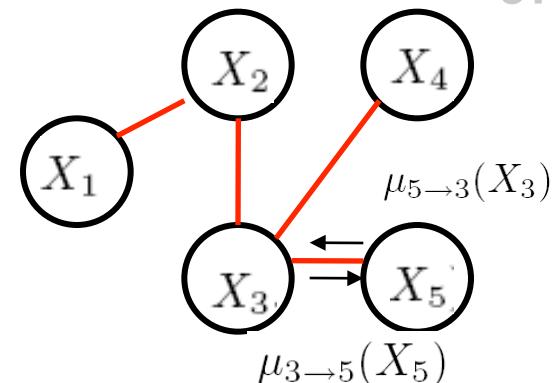
Sum-Product algorithm

- Distributed computation on a graph
 - ‘message-passing’
- Message from node i to node j

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

– ‘Given all I know from the rest, this is where you should be’

- Beliefs: $B_j(X_j) = \Phi_j(X_j) \prod_{i \in \mathcal{N}(j)} \mu_{i \rightarrow j}(X_j)$
- At convergence $P(X_i) = B(X_i)$

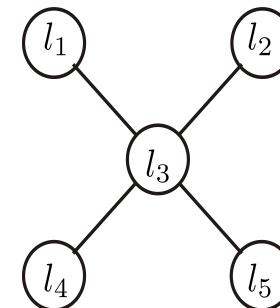


Max/Sum-Product algorithms

- Asynchronous 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

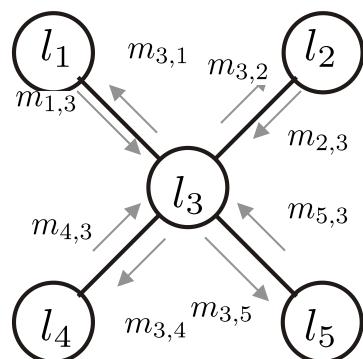
Pictorial Structures

- 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: $\mu_{i \rightarrow j}(X_j) = \max_{X_i} \Phi_i(X_i) \Psi_{i,j}(X_i, X_j) \prod_{k \in \mathcal{N}(i) \setminus j} \mu_{k \rightarrow j}(X_i)$
 - Efficient Implementation: Felzenszwalb & Huttenlocher, CVPR 2001/IJCV 05

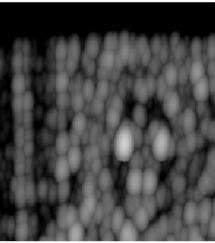


$$\mu_{i \rightarrow j}(X_j) = \max_{X_i} \Phi_i(X_i) \Psi_{i,j}(X_i, X_j) \prod_{k \in \mathcal{N}(i) \setminus j} \mu_{k \rightarrow j}(X_i)$$

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



$$\Phi_1(l)$$



$$\Phi_2(l)$$



$$\Phi_3(l)$$



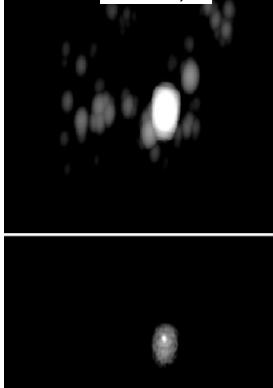
$$\Phi_4(l)$$



$$\Phi_5(l)$$



$$m_{1,3}$$



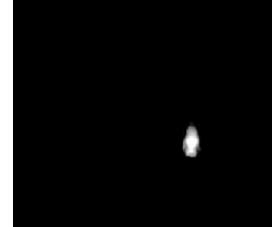
$$B_1(l)$$

$$m_{2,3}$$

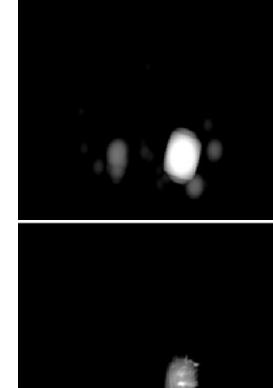


$$B_2(l)$$

$$B_3(l)$$

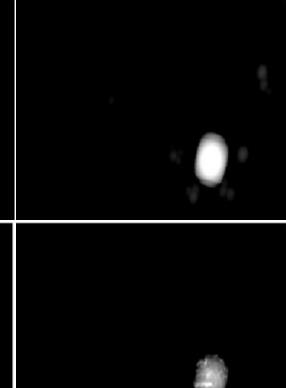


$$m_{3,4}$$



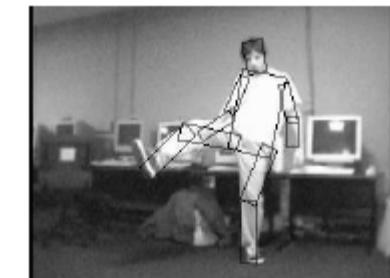
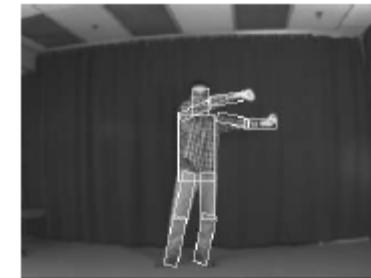
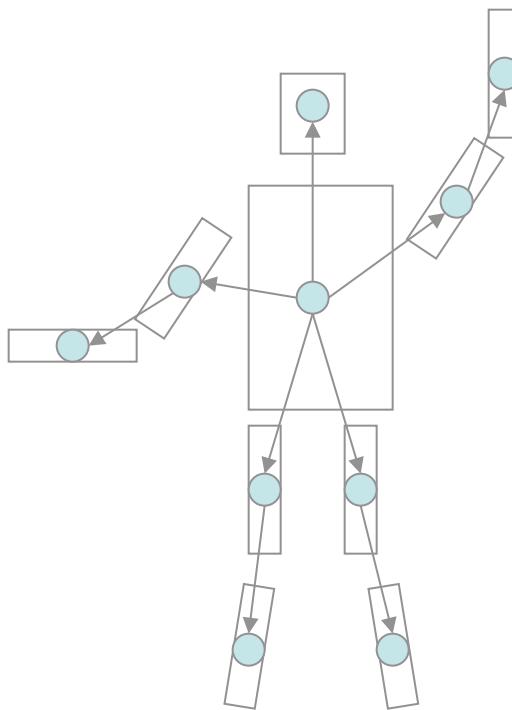
$$B_4(l)$$

$$m_{3,5}$$



$$B_5(l)$$

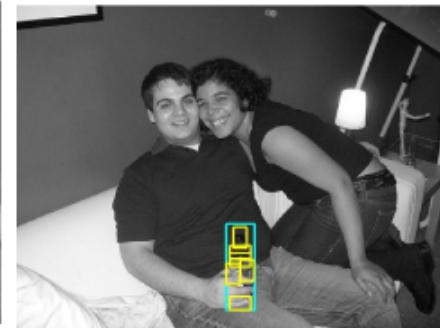
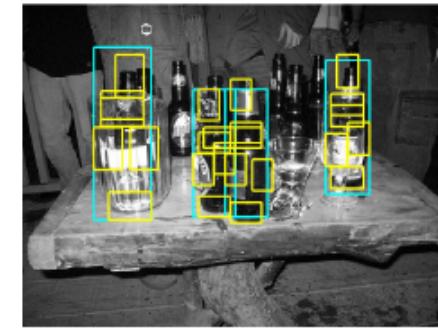
Detection Results



Part-based models + discriminative framework

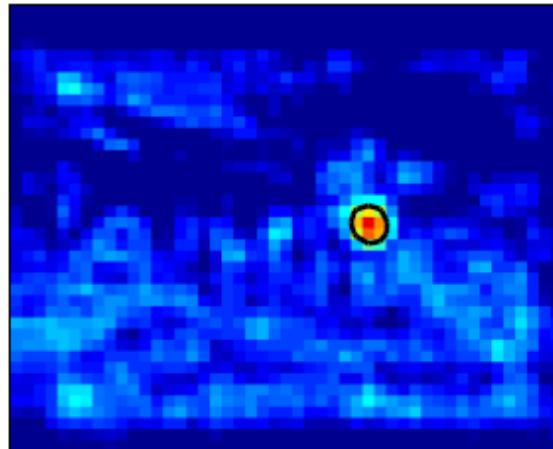
P. Felzenszwalb, D. McAllester, D. Ramanan`A Discriminatively Trained, Multiscale, Deformable Part Model' CVPR 2008

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

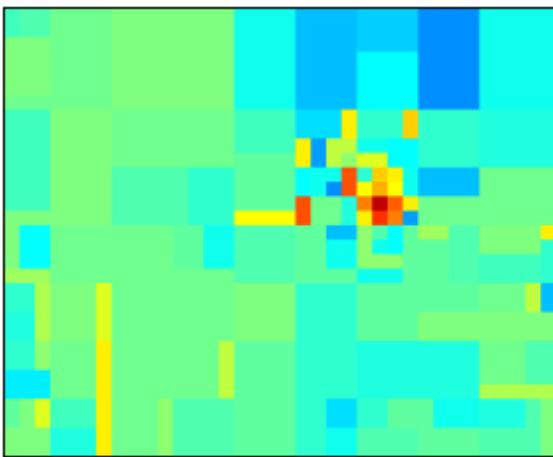
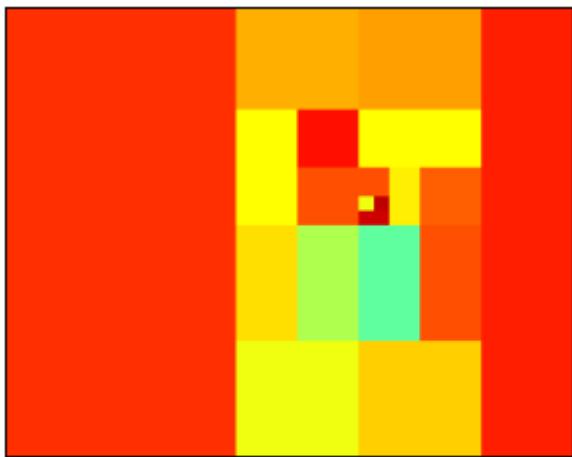


- State-of-the-art, 2008

Efficient object detection with dense features



(a) Input & Detection result (b) Detector score $S(x)$



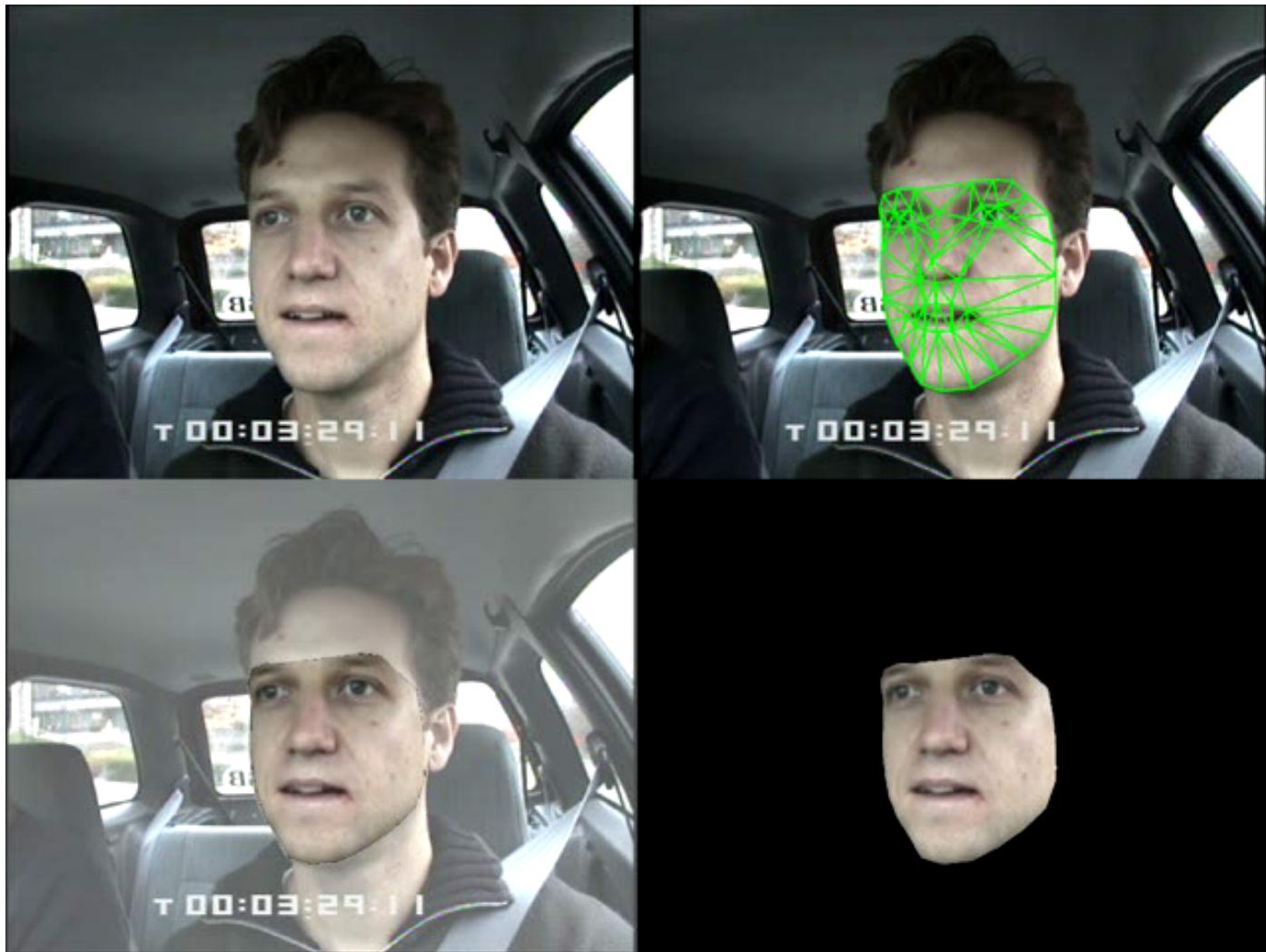
(c) BB for $\arg \max_x S(x)$ (d) BB for $S(x) \geq -1$.

Next class: Rapid Deformable Object Detection using Dual-Tree Branch & Bound,
I.Kokkinos, NIPS 2011

Class projects

- Programming: Matlab, C, C++, OpenCV,...
- Content: Two options
 - Implement a current state-of-the-art/classic paper
 - ~1 week
 - Harder papers: 2 persons
 - Do a small research project

3D AAM for face tracking



Current State-of-the-art



Face Detection, Pose Estimation, and Landmark Localization
in the Wild, X. Zhu and D. Ramanan, CVPR 2012

Project proposal (2 persons / Stage)

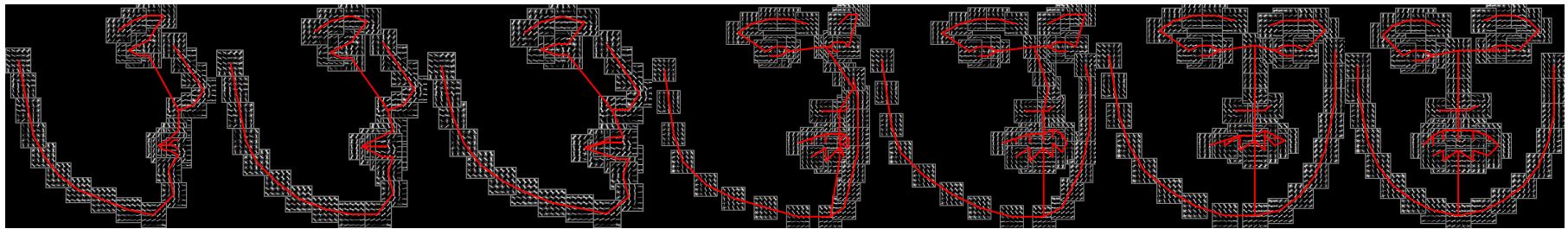


Figure 2: Our mixture-of-trees model encodes topological changes due to viewpoint. Red lines denote springs between pairs of parts; note there are no closed loops, maintaining the tree property. All trees make use of a common, shared pool of part templates, which makes learning and inference efficient.

Computation: Our 4 models tend to have consistent relative performance across our datasets and evaluations, with *Independent-1050* performing the best, taking roughly 40 seconds per image, while *Share-99* performs slightly worse but roughly 10× faster. We present quantitative results later in Fig. 12. With parallel/cascaded implementations, we believe our models could be real-time. Due to space restrictions, we mainly present results for these two “extreme” models in this section.

Project: exact acceleration using Branch-and-Bound (lectures 7)

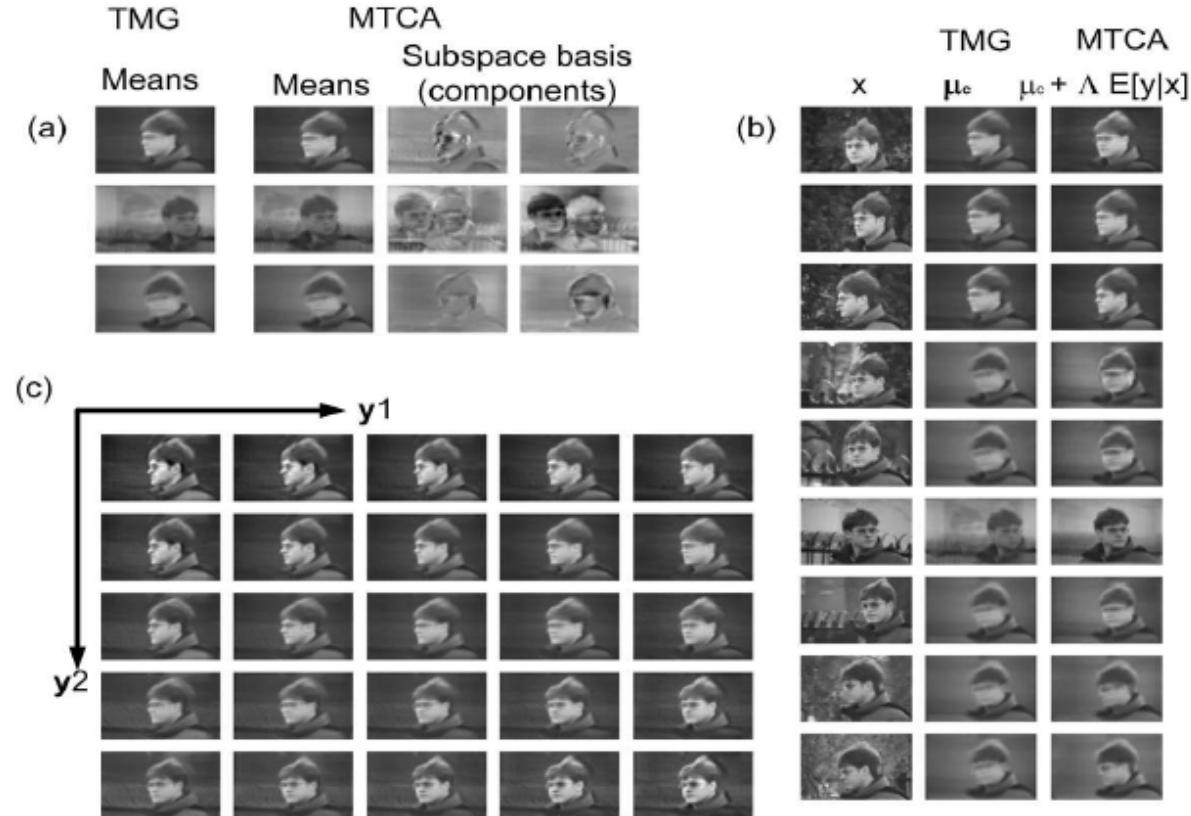
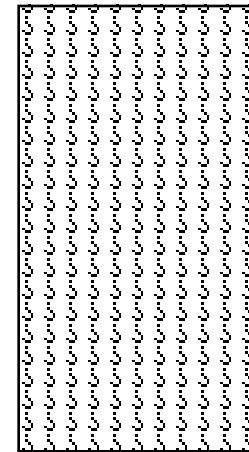
- I. Kokkinos. Rapid Deformable Object Detection using Dual Tree Branch and Bound. In NIPS 2011.
- I. Kokkinos. Bounding Part Scores for Rapid Detection with DPMs, In PnA Workshop, ECCV 2012
<http://vision.mas.ecp.fr/Personnel/iasonas/dpms.html>

Continuous + Discrete Hidden Variables: Mixture of Transformed Components

Latent variables for cluster

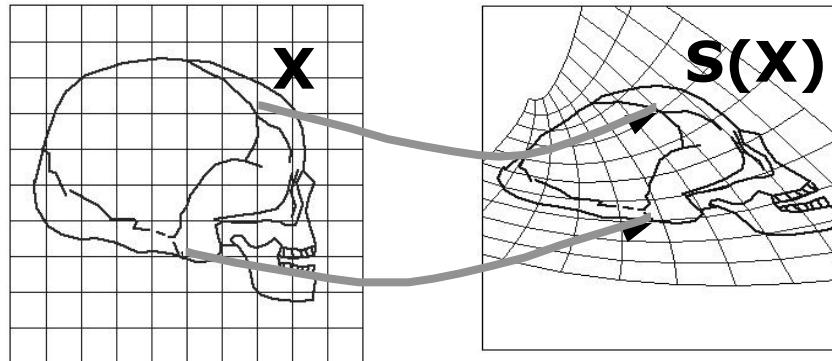
Latent variables for components

Latent variables for shift



Nonrigid deformations: AAMS

$$T(\mathbf{x}) = I(\mathcal{S}(\mathbf{x}))$$



- Active Appearance Models

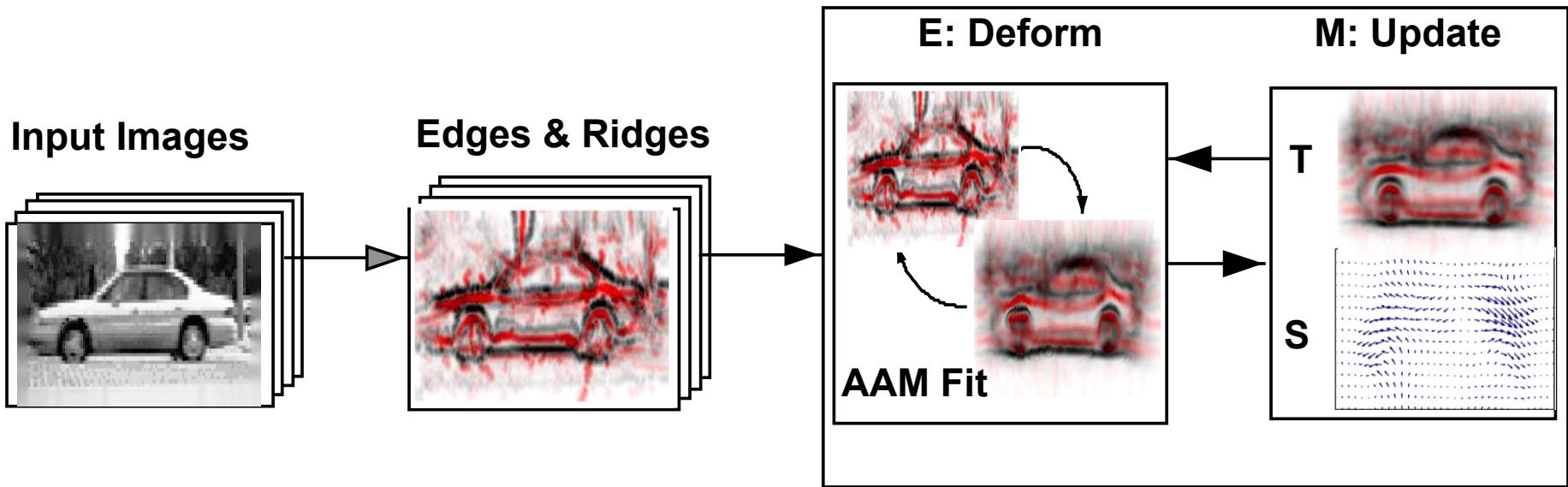
$$\mathcal{S}(\mathbf{x}; \mathbf{s}) = \sum_i \mathbf{s}_i S_i(\mathbf{x}) \quad T(\mathbf{x}; \mathbf{t}) = \sum_i \mathbf{t}_i T_i(\mathbf{x})$$

EM-based AAM learning

Training criterion:

$$\min_{\mathbf{s}_k, S_i, \mathcal{T}} \sum_{k=1}^K \sum_{\mathbf{x}} [I_k(\mathcal{S}(\mathbf{x}; \mathbf{s}_k)) - \mathcal{T}(\mathbf{x})]^2$$

$$\mathcal{S}(\mathbf{x}; \mathbf{s}_k) = \sum_i \mathbf{s}_{i,k} S_i(\mathbf{x})$$



Bottle models

Observations (x)



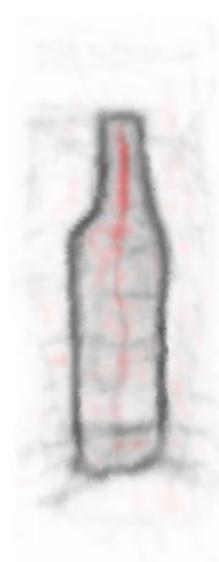
Template



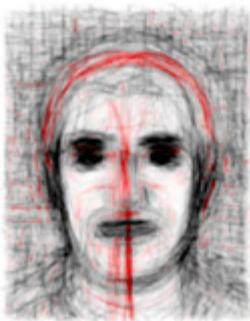
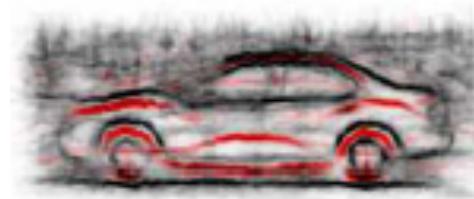
1st basis element



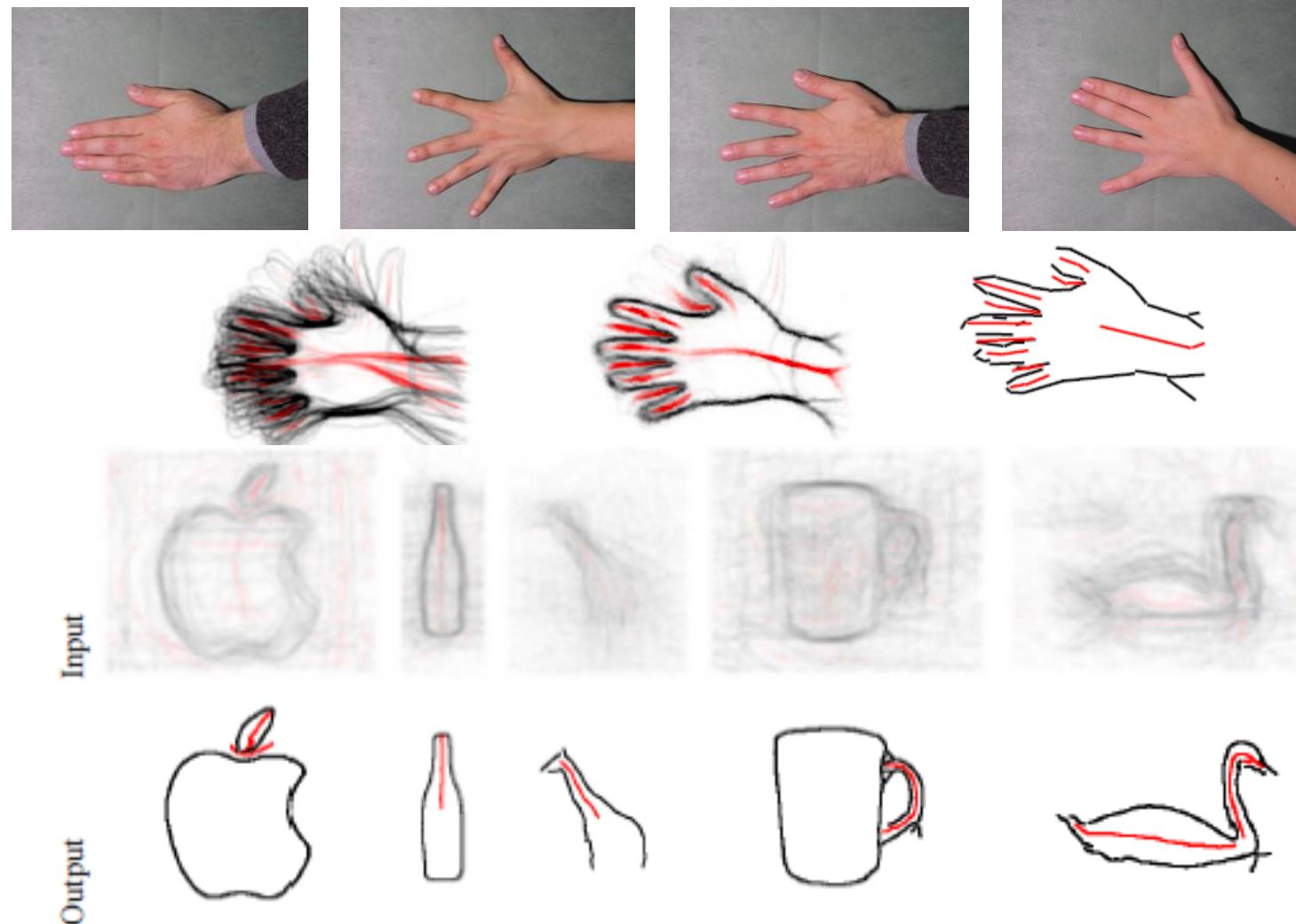
2nd basis element



Recovering Object Contours (2007)



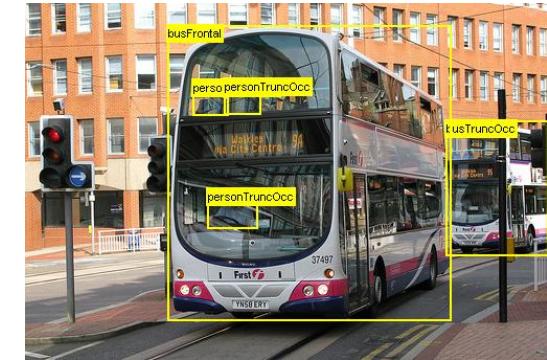
Hand, apple, giraffe, mug, swan models (2008)



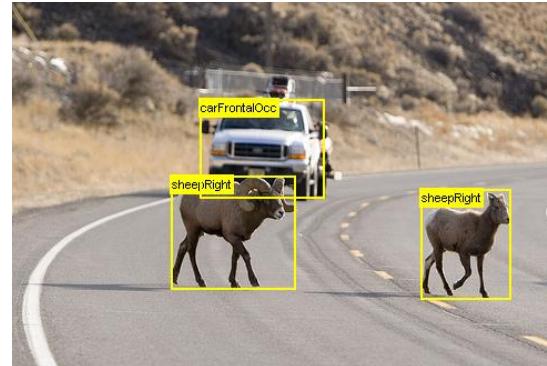
Pascal Dataset

20 Categories, 25000 images

'Bus'



'Car'

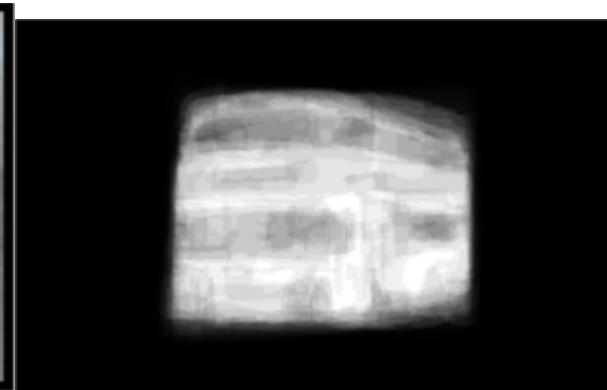
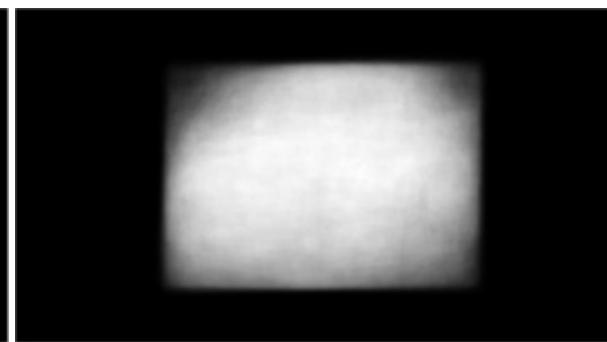
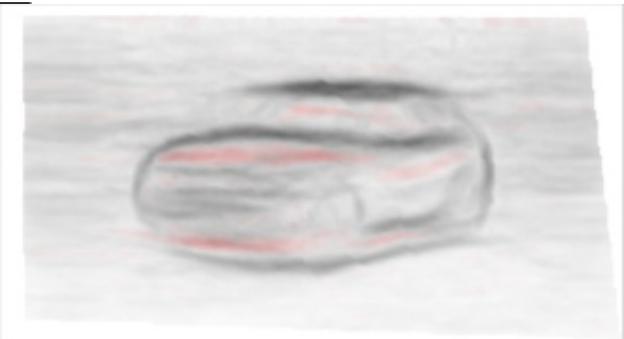
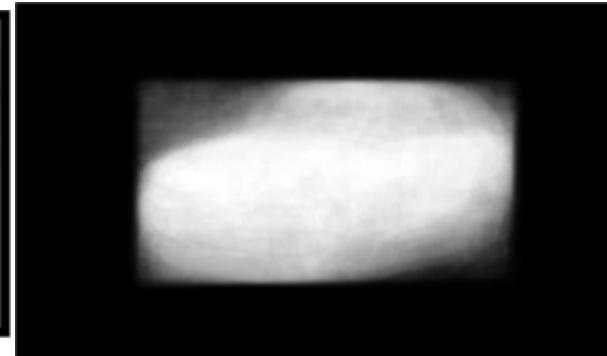


17.03.2008

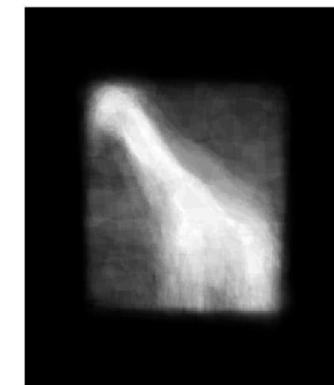
Multi-view models (2011)



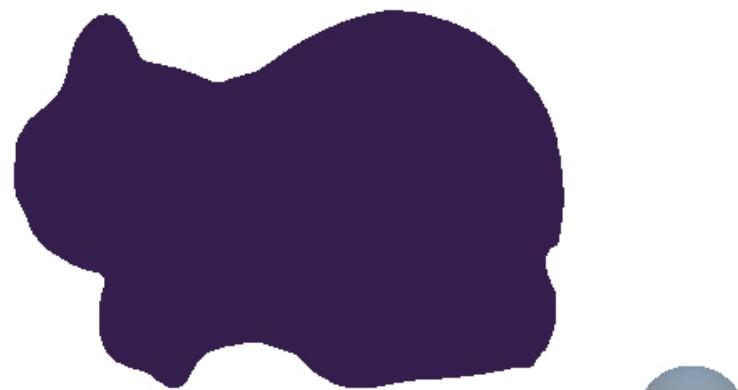
Average segmentation/boundary masks



What's next? Segmentation

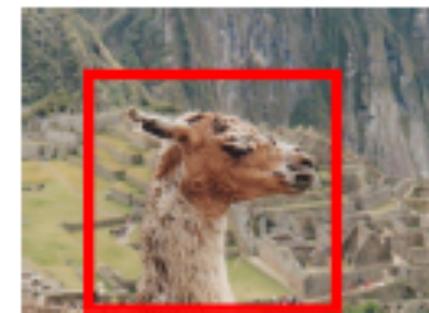
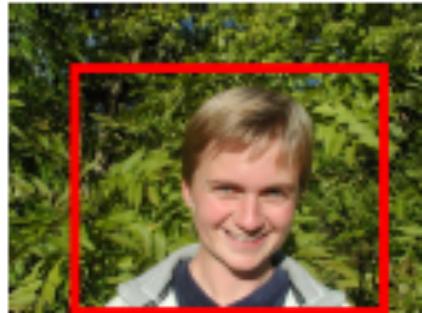


Project: Shape-from-shading on category data



Color Constancy, Intrinsic Images, and Shape Estimation
Jonathan T. Barron, Jitendra Malik
European Conference on Computer Vision (ECCV), 2012

Interactive image segmentation (1 Person)

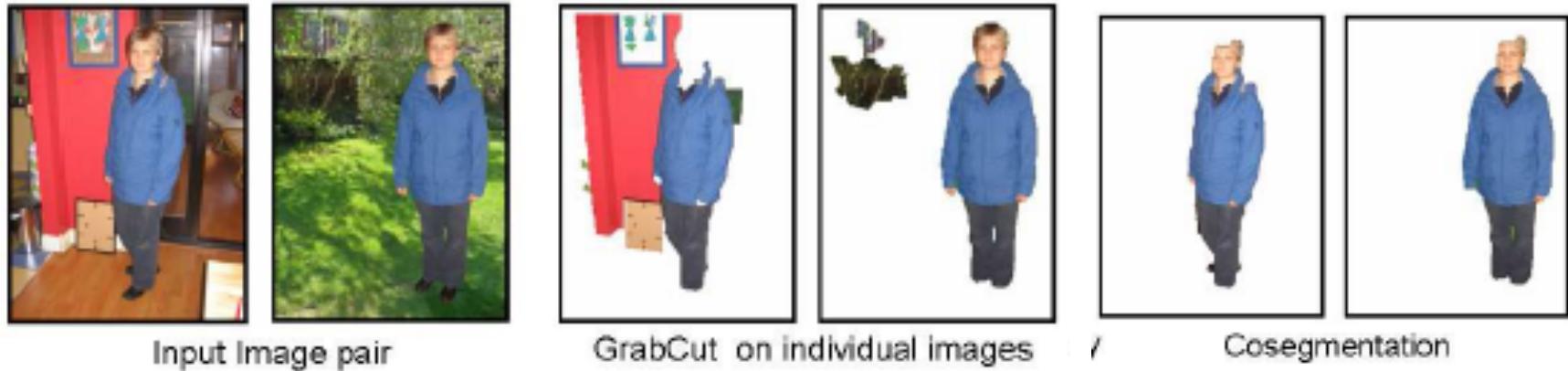


- Involves:
 - Mixture-of-Gaussians training
 - Graph-cuts segmentation
 - Color, intensity and texture features

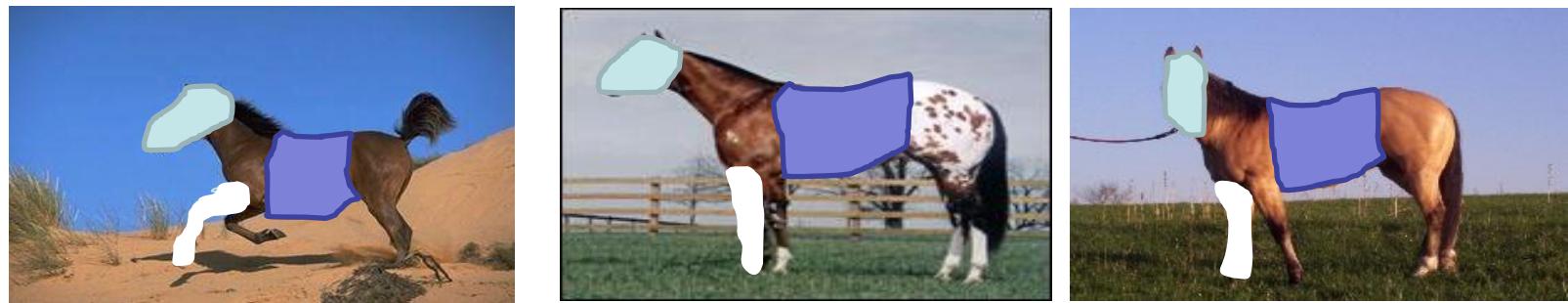
‘“GrabCut” — Interactive Foreground Extraction using Iterated Graph Cuts’ Carsten Rother, Vladimir Kolmogorov, Andrew Blake, Siggraph 2004

Image Cosegmentation (1-2 Persons)

- Simultaneously segment two images



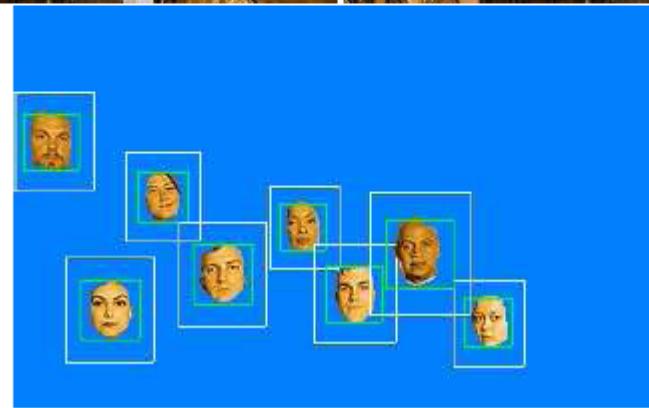
- Open problem: simultaneously segment multiple images



Cosegmentation of Image Pairs by Histogram Matching — Incorporating a Global Constraint into MRFs. Rother, C. Minka, T. Blake, A. Kolmogorov

Combined segmentation and recognition (1-2 Persons)

- Combination of graph-cuts with object detection



OBJCUT for Face Detection

Jonathan Rihan, Pushmeet Kohli, and Philip H.S. Torr

BiCoS: Bi-level Co-Segmentation (2 Persons)

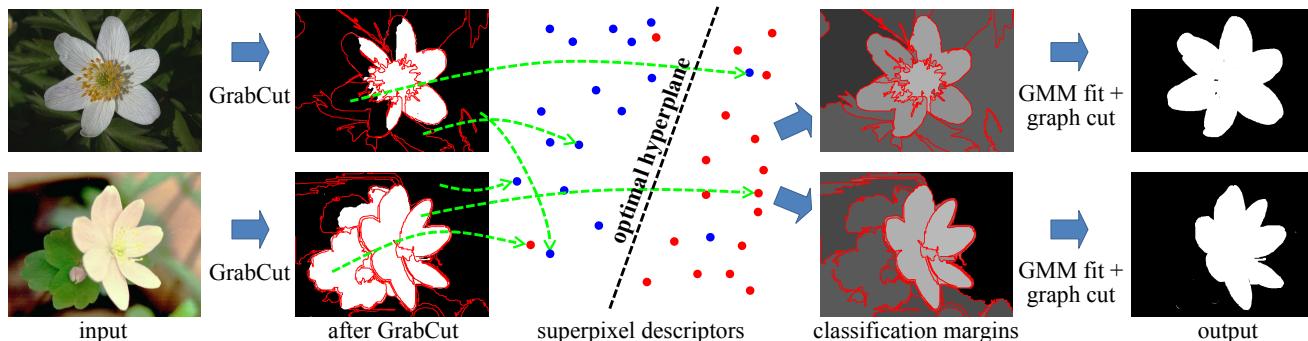


Figure 1. The diagram of our method (BiCoS) shown for 2 (out of 15) training images from one class of the Oxford Flowers-17 dataset. BiCoS starts with GrabCut at the pixel-level applied to each image independently. Each superpixel (superimposed) is then assigned to foreground or background and mapped to a descriptor space, where the optimal separating hyperplane is found. The pixels in each image are then assigned values according to the margins of the respective superpixels, and such soft segmentation map is then used to fit color Gaussian mixtures and apply pixel-level graph cut to each image.



Figure 3. Flower segmentations: (top) original images, (middle) segmentations by Joulin *et al.* [19] and (bottom) the segmentations of BiCoS-MT. The results of BiCoS are very similar to BiCoS-MT for these particular images.

TriCoS: A Tri-level Class-Discriminative Co-segmentation Method for Image Classification, Chai, Rahtu, Lempitsky, V. Gool, Zisserman ,ECCV 2012

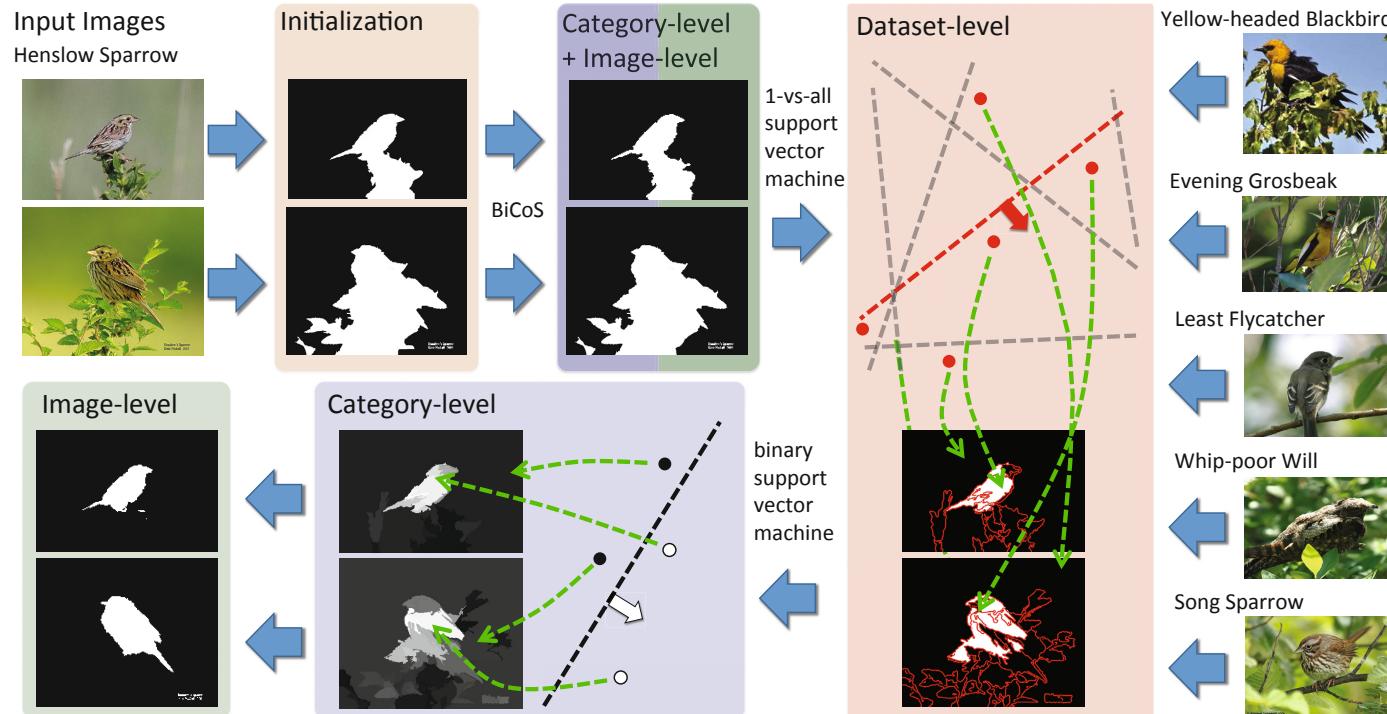
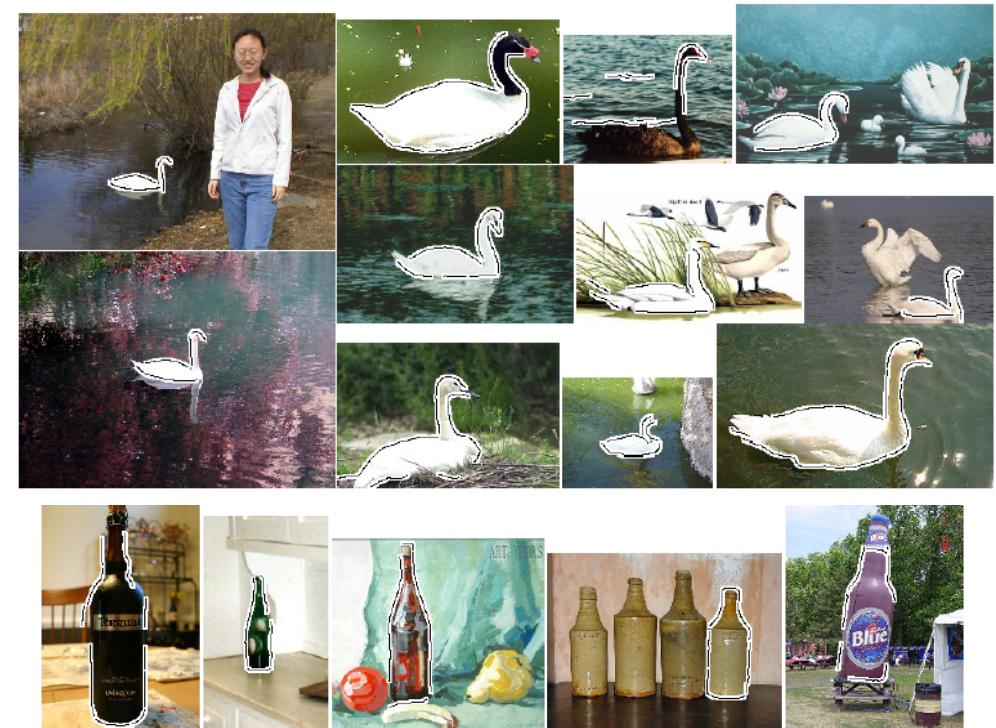
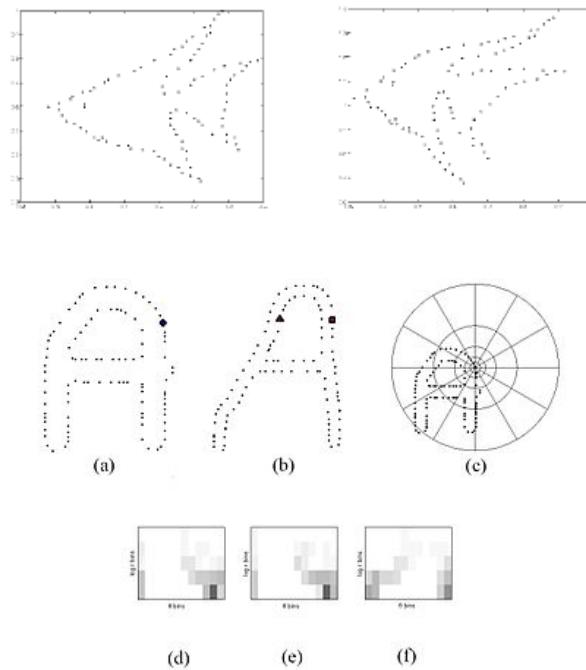


Fig. 1. An overview of the flow of the discriminative co-segmentation pipeline TriCoS. We start with the input images and their initialization segmentation masks on the top left. They are then processed by the BiCoS [6]. Although BiCoS often improves the segmentation quality, in this example, it preserves both the birds and the tree branches as foreground because they appear multiple times across the images from the same category. The new dataset-level step finds leaves to be less discriminative since most birds fly around trees and therefore discards the leaves in the segmentation.

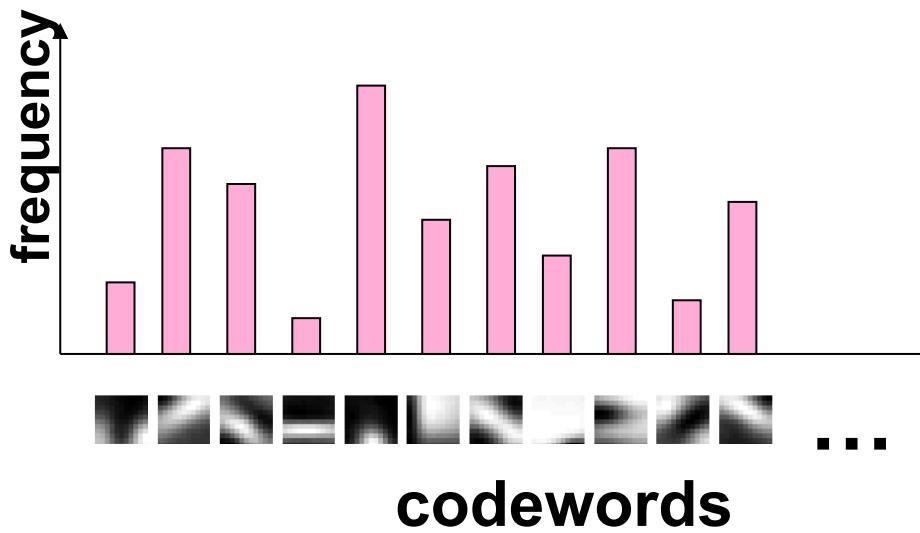
Shape matching & recognition (1 Person)

- "Shape Matching and Object Recognition Using Shape Contexts" [S. Belongie](#), [J. Malik](#), and J. Puzicha. PAMI 2002
- Hierarchical Matching of Deformable Shapes, P. Felzenszwalb 2007

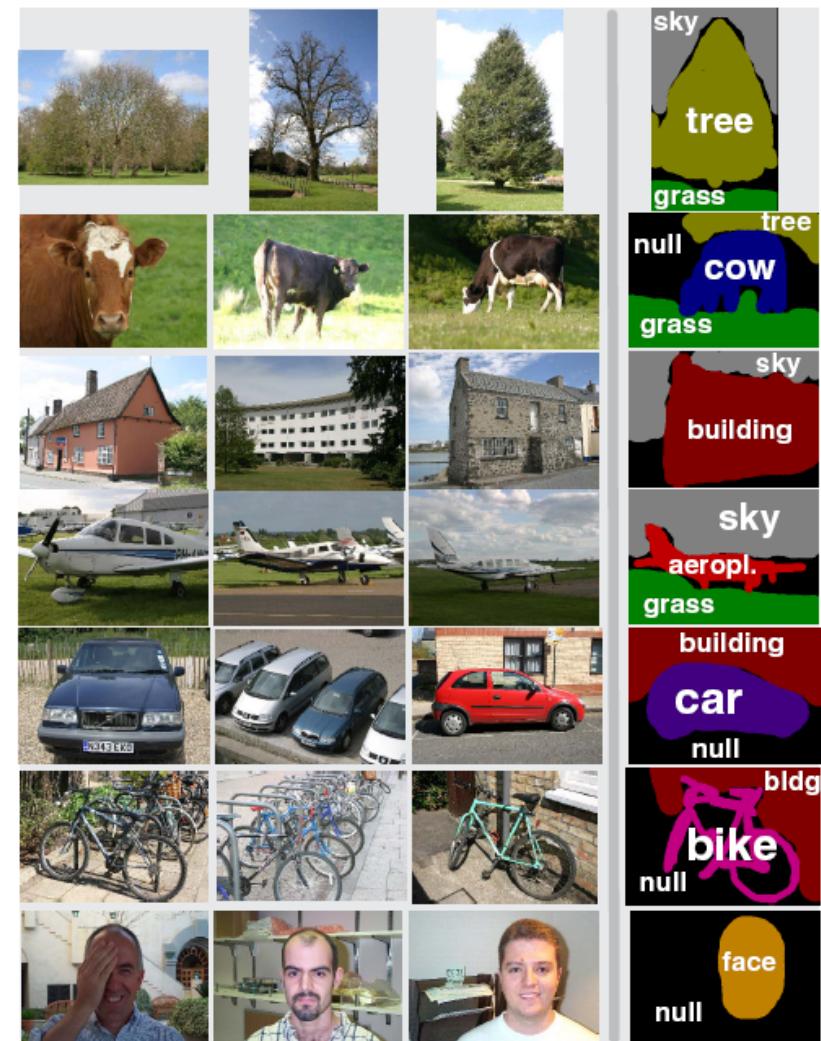


Object recognition as texture recognition (1 Person)

- Bag-of-words model

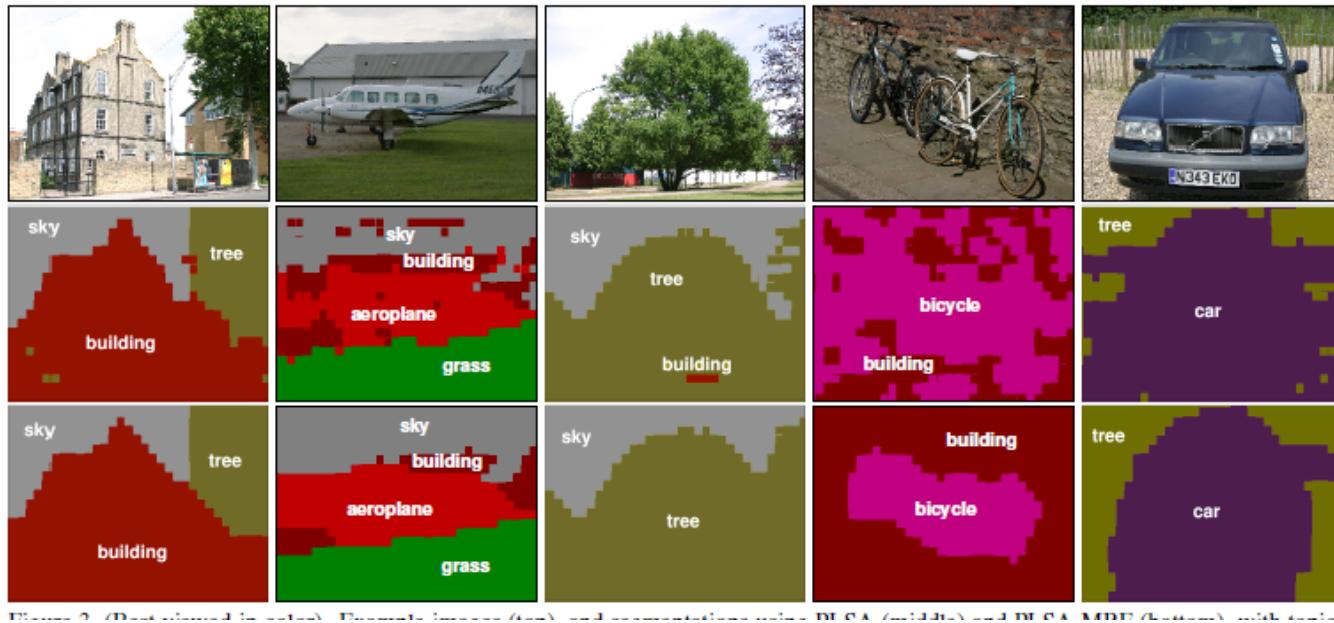


- MSRC dataset



Object recognition as texture recognition

- Topic models + Markov Random Fields

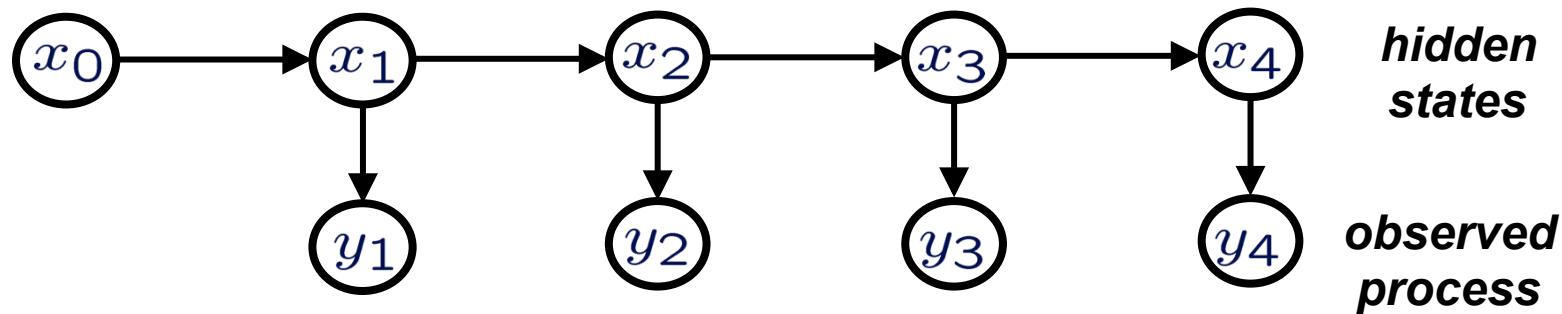


Region classification with Markov field aspect models. Jakob Verbeek and William Triggs, CVPR 2007

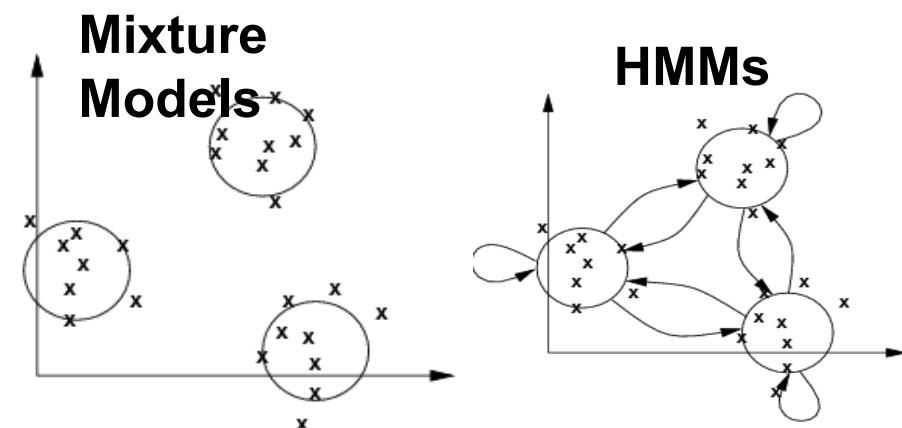
Appendix: Inference & Learning for Chain-structured GMs

- 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

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

Three Algorithms

- Likelihood Evaluation (Forward-Backward)

$$P(Y)$$

- Inference (Viterbi)

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

- Parameter Estimation (EM/Baum-Welch)

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

Likelihood Evaluation

- Using Marginalization

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

- Complexity:
where N is the number of states

$$TN^T$$

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

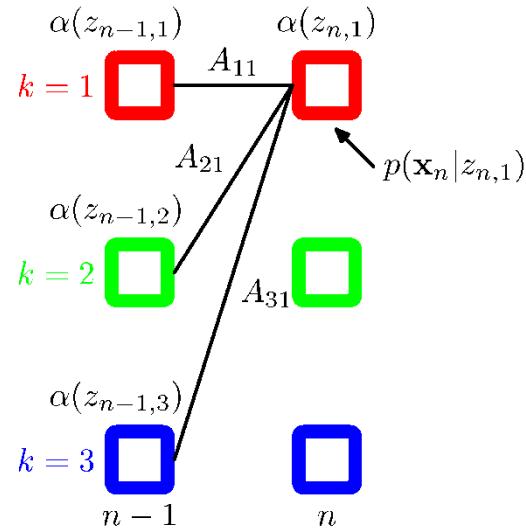
$$\begin{aligned}
 \alpha_t(j) &= P(x_t, y_1, \dots, y_t) \\
 &= P(y_1, \dots, y_t | x_t) P(x_t) \\
 &= P(y_1, \dots, y_{t-1} | x_t, y_t) P(y_t | x_t) P(x_t) && P(a, b | c) = P(a | b, c) P(b | c) \\
 &= P(y_1, \dots, y_{t-1} | x_t) P(y_t | x_t) P(x_t) \\
 &= \sum_{x_{t-1}} P(y_1, \dots, y_{t-1}, x_{t-1} | x_t) P(y_t | x_t) P(x_t) \\
 &= \sum_{x_{t-1}} P(y_1, \dots, y_{t-1} | x_{t-1}, x_t) P(x_{t-1} | x_t) P(x_t) P(x_{t-1}, x_t) P(y_t | x_t) \\
 &= \sum_{x_{t-1}} P(y_1, \dots, y_{t-1} | x_{t-1}) P(x_t | x_{t-1}) P(x_{t-1}) P(y_t | x_t) \\
 &= \sum_{x_{t-1}} P(y_1, \dots, y_{t-1}, x_{t-1}) P(x_t | x_{t-1}) P(y_t | x_t) \\
 &= \sum_{x_{t-1}} \alpha_{t-1}(x_{t-1}) P(x_t | x_{t-1}) P(y_t | x_t) \\
 \alpha_t(x_t) &= \sum_{x_{t-1}} \alpha_{t-1}(x_{t-1}) P(x_t | x_{t-1}) P(y_t | x_t) \\
 \alpha_1(x_1) &= P(x_1, y_1) = P(y_1 | x_1) P(x_1)
 \end{aligned}$$

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

$$\begin{aligned} P(Y) &= \sum_{x_1} \underbrace{P(y_1|x_1)P(x_1)}_{\alpha_1(x_1)} \sum_{x_2} P(y_2|x_2)P(x_2|x_1) \dots \sum_{x_T} P(y_T|x_T)P(x_T|x_{T-1}) \\ &= \sum_{x_2} \underbrace{\sum_{x_1} \alpha_1(x_1)P(y_2|x_2)P(x_2|x_1)}_{\alpha_2(x_2)} \sum_{x_3} P(y_3|x_3)P(x_3|x_2) \dots \sum_{x_T} P(y_T|x_T)P(x_T|x_{T-1}) \end{aligned}$$

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

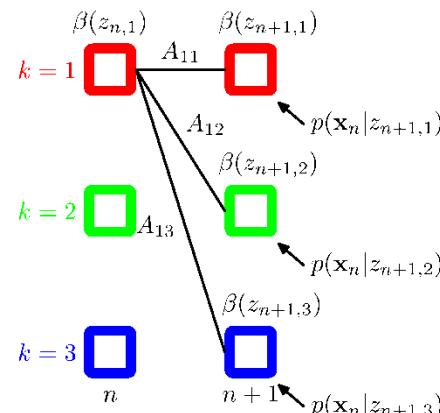
- Brute force: $T \times N^T \quad 50 \times 10^{50}$
- Recursive: $T \times N^2 \quad 50 \times 100$

Backward Variables β

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

$$\beta_T(i) = 1$$

- 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)

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

Posterior for a single state

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

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

- Can rewrite using α, β

$$\gamma = \alpha \beta$$

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

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

- So

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

Optimal State Sequence

- Different from concatenation of optimal states

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

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

- Viterbi Algorithm
 - Exploit Decomposable cost function
 - Dynamic Programming

Exploit conditional independence

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

Viterbi Algorithm: recursion

- Introduce

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

- Express recursively:

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

Viterbi Algorithm

- Initialize

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

- Recurse

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

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

- Backtrack $P^* = \max_i \delta_T(i)$
 $q_T^* = \arg \max_j \delta_T(j)$
 $q_t^* = \psi_{t+1}(q_t^* + 1)$

- 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)**

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

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:

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

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

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

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

Posterior on joint states

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

M-step

- Replace hidden variables with their expectations

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

5th Lecture Layout

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

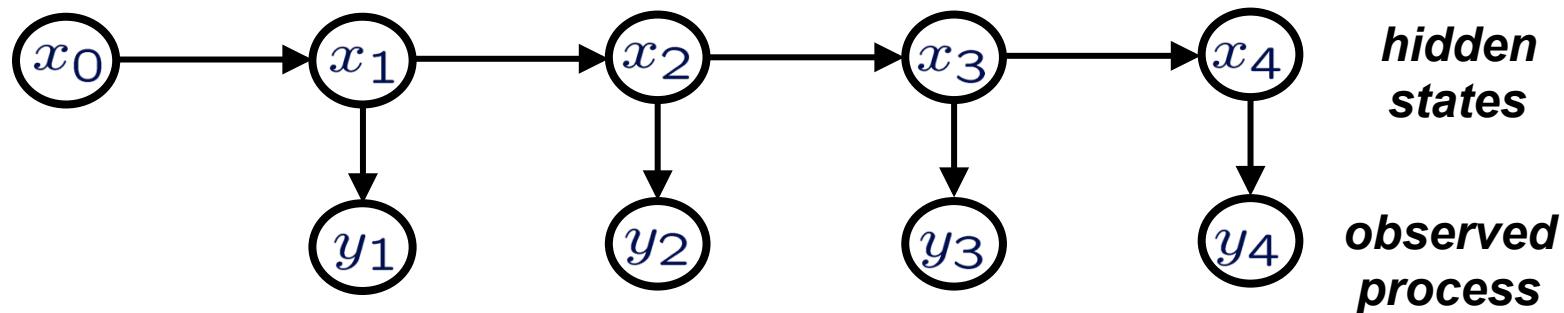
Linear Dynamical Systems

- Noise in State, noise in Observations

$$x_t = Ax_{t-1} + v_t, \quad v_t \sim N(0, \Sigma)$$

$$y_t = Cx_t + w_t, \quad w_t \sim N(0, Q)$$

$$x_1 = \mu_0 + u, \quad u \sim N(0, V)$$



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

Kalman Filter Recursion

- State estimate

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

- As in HMMs, come up with a recursion

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

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

Prediction distribution

- Write in terms of previous state posterior

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

$$\mu_{t|t-1} = E(x_t | y_1, \dots, y_{t-1})$$

$$\Sigma_{t|t-1} = E((x_t - \mu_{t|t-1})(x_t - \mu_{t|t-1})^T | y_1, \dots, y_{t-1})$$

$$x_{t+1} = Ax_t + v_t, v_t \sim N(0, \Sigma)$$

$$\mu_{t|t-1} = A\mu_{t-1|t-1}$$

$$\Sigma_{t|t-1} = A\Sigma_{t-1|t-1}A^T + \Sigma$$

Posterior distribution

- Posterior distribution

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

$$\mu_{t|t} = E(x_t | y_1, \dots, y_t)$$

$$\Sigma_{t|t} = E((x_t - \mu_{t|t})(x_t - \mu_{t|t})^T | y_1, \dots, y_t)$$

- Consider joint distribution of state and observation

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

Joint distribution of state and observation

- We have distribution for x , need to consider y

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

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

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

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

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

Posterior distribution on state

- Condition on new observation y

$$K = \Sigma_{t|t-1} C^T (C \Sigma_{t|t-1} C^T + Q)^{-1}$$

$$\mu_{t|t} = \mu_{t|t-1} + K(y_t - C\mu_{t|t-1})$$

$$\Sigma_{t|t} = \Sigma_{t|t-1} - K C \Sigma_{t|t-1}$$