Introduction to Graphical Models

lecture 4 - Inference in HMMs, MRFs; Junction Trees

Marc Toussaint
TU Berlin

- outline

  1) examples for inference & BP:
  – HMMs
  – MRFs

  2) additional comments to BP:
  – Junction Trees
  – max-product

Belief Propagation – recap

- general message equations:

$$\mu_{C\to i}(X_i) = \sum_{X_C \setminus X_i} \psi_C(X_C) \prod_{j\in C, j\neq i} \mu_{j\to C}(X_j) \ ,$$

$$\mu_{i\to C}(X_i) = \prod_{D\in\nu(i), D\neq C} \mu_{D\to i}(X_i)$$

- beliefs:

$$b_C(X_C) := \psi_C(X_C) \prod_{i\in C} \mu_{i\to C}(X_i) \ , \quad b_i(X_i) := \prod_{C\in\nu(i)} \mu_{C\to i}(X_i)$$

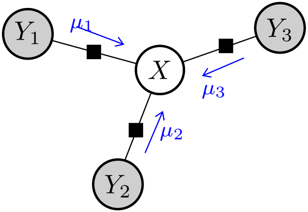
- special case: pair-wise factors $\to$ variable-to-variable messages:

$$\mu_{j\to i}(X_i) = \sum_{X_j} \psi_C(X_i, X_j) \prod_{k:k\neq i} \mu_{k\to j}(X_j) \ ,$$

- special case: separators $\to$ factor-to-factor messages:

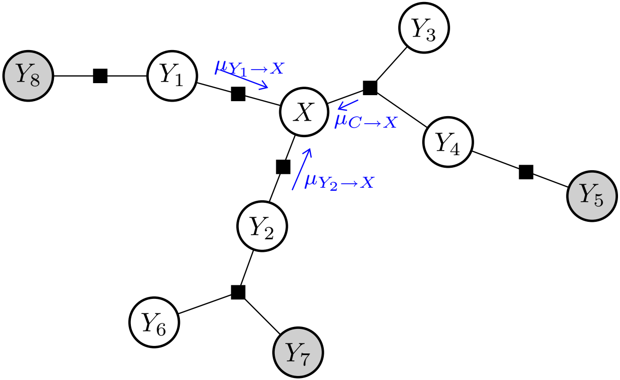
$$\mu_{D\to C}(X_i) = \sum_{X_D \setminus X_i} \psi_D(X_D) \prod_{E:E\neq C} \mu_{E\to D}(X_{E\cap D}) \ ,$$

Belief Propagation – recap



- Naive Bayes:
  $P(X|Y_{1:n}) \propto P(X)\ \prod_{i=1}^{n} \mu_i(X)$
  $\mu_i(X) := P(Y_i = y_i \mid X)$

Belief Propagation – recap



- Belief Propagation $b_i(X) := \prod_{C \in \nu(X)} \mu_{C \to X}(X)$

  – messages represent (indep.) information from branches

  – messages are the temporary terms $t_k(X)$ that arise when eliminating (Elim.Alg.) a branch ($\to$ exactness on trees)

Hidden Markov Models

- used in
  - speech recognition
  - molecular biology sequences
  - linguistic sequences (e.g. part-of-speech tagging)
  - multi-electrode spike-train analysis
  - tracking objects through time

- ... motivate with data ...

Hidden Markov Models

- HMM definition:

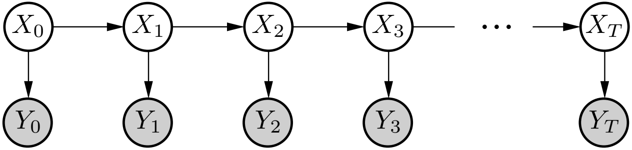
  – a (temporal) sequence of random variables $X_0, .., X_T$, each with the same domain dom(X_t)

  – to each X_t and observation RV associated Y_i, each with same domain dom(Y_t)

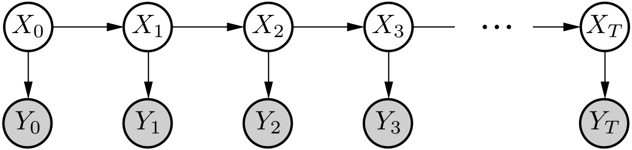
  – the joint distribution

$$P(X_0, .., X_T, Y_0, .., Y_T) = P(X_0) \cdot \prod_{t=1}^{T} P(X_t | X_{t\text{-}1}) \cdot \prod_{t=0}^{T} P(Y_t | X_t) \ .$$

- graphically:



Properties of HMMs



- Markov property:

  for all $a \leq t$ and $b > t$

  – X_a conditionally independent from X_b given X_t

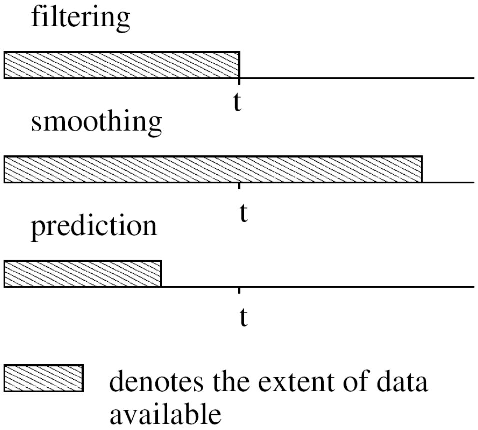
  – Y_a conditionally independent from Y_b given X_t

  – the future is independent of the past given the present

  – note that conditioning on Y_t does not yield any conditional independences

different inference problems in HMMs

- $P(x_{0:T} \mid y_{0:T})$ inferring hidden state given $y_{0:T}$
- $P(x_t \mid y_{0:T})$ marginal of above
- $P(x_t \mid y_{0:t})$ filtering
- $P(x_t \mid y_{0:a})$, $t > a$ prediction
- $P(x_t \mid y_{0:b})$, $t < b$ smoothing
- $P(y_{0:T})$ likelihood calculation
- Find sequence $x^*_{0:T}$ that maximizes $P(x_{0:T} \mid y_{0:T})$ [Viterbi alignment]


filtering
t
smoothing
t
prediction
t
denotes the extent of data available

Inference in HMMs

- 1) classical derivation
  – good to know, typical notation found in the literature

- 2) derivation from Belief Propagation equations

Inference in HMMs

- classical derivation

$$
\begin{aligned}
P(x_t \,|\, y_{0:T}) &= \frac{P(y_{0:T} \,|\, x_t)\; P(x_t)}{P(y_{0:T})} \\
&= \frac{P(y_{0:t} \,|\, x_t)\; P(y_{t+1:T} \,|\, x_t)\; P(x_t)}{P(y_{0:T})} \\
&= \frac{P(y_{0:t}, x_t)\; P(y_{t+1:T} \,|\, x_t)}{P(y_{0:T})} \\
&= \frac{\alpha(x_t)\; \beta(x_t)}{P(y_{0:T})} =: \gamma(x_t)
\end{aligned}
$$

$$
\begin{aligned}
\alpha(x_t) &= P(y_{0:t}, x_t) = \phi(y_t)\; P(y_{0:t\text{-}1}, x_t)\;, \quad \phi(x_t) \equiv P(y_t | x_t) \\
&= \phi(y_t)\; \sum_{x_{t\text{-}1}} P(x_t \,|\, x_{t\text{-}1})\; \alpha(x_{t\text{-}1}) \\
\beta(x_t) &= P(y_{t+1:T} \,|\, x_t) = \sum_{x+1} P(y_{t+1:T} \,|\, x_{t+1})\; P(x_{t+1} \,|\, x_t) \\
&= \sum_{x+1} \Big[\beta(x_{t+1})\; \phi(y_{t+1})\Big]\; P(x_{t+1} \,|\, x_t)
\end{aligned}
$$

Inference in HMMs

- derivation from BP equations – variable-to-variable message:

$$\mu_{j\to i}(X_i) = \sum_{X_j} \psi_C(X_i, X_j) \prod_{k:k\neq i} \mu_{k\to j}(X_j) \ ,$$

- message in the HMM case

$$\mu_{t\text{-}1\to t}(x_t) = \sum_{x_{t\text{-}1}} P(x_t|x_{t\text{-}1})\ \mu_{t\text{-}2\to t\text{-}1}(x_{t\text{-}1})\ \phi(x_{t\text{-}1})$$

$$\mu_{t+1\to t}(x_t) = \sum_{x_{t+1}} P(x_{t+1}|x_t)\ \mu_{t+2\to t+1}(x_{t+1})\ \phi(x_{t+1})$$

$$b(x_t) = \mu_{t\text{-}1\to t}(x_t)\ \phi(x_t)\ \mu_{t+1\to t}(x_t)$$

belief = product of message from past, future, and cur. observation!

- compare to classical:

$$\alpha(x_t) \equiv \mu_{t\text{-}1\to t}(x_t)\ \phi(x_t) \quad \Rightarrow \quad \mu_{t\text{-}1\to t}(x_t) \equiv P(y_{0:t\text{-}1}, x_t)$$

$$\beta(x_t) \equiv \mu_{t+1\to t}(x_t) \equiv P(y_{t+1:T}\,|\,x_t)$$

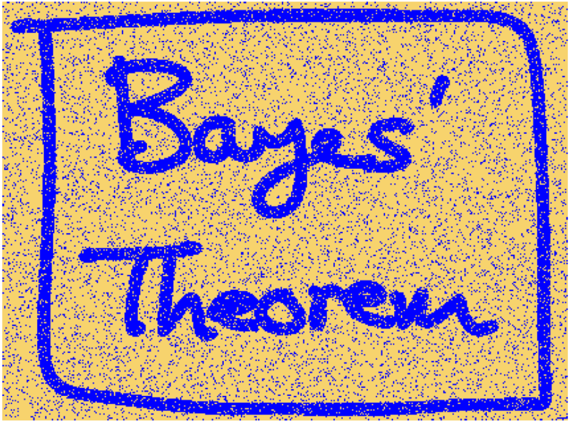
(asymmetry w.r.t. $|x_t$ stems from asymmetry of the factors $P(x_t|x_{t\text{-}1})$)

HMMs

- ... demo on binary data ...

Markov Random Fields

- image denoising example:



Markov Random Fields

- assume every pixel is a binary (black/white) random variable
  Let $I = \{0, .., W\} \times \{0, .., H\}$ be be the index set (height$\times$width)
  we have binary random variables X_i for all $i \in I$ representing the pixels of the true image
  we have binary randon variable Y_i representing the observations (camera snapshot)

- assume neighboring pixels are coupled

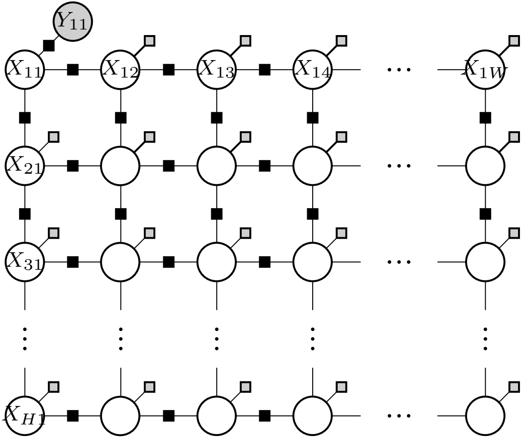
$$P(x_I, y_I) \propto \prod_{(ij)} \psi(x_i, x_j) \cdot \prod_{i \in I} \phi(x_i, y_i)$$

  with couplings

$$\psi(x_i, x_j) = \begin{cases} \varrho & x_i = x_j \\ 1 - \varrho & \text{else} \end{cases} \quad \phi(x_i, y_i) = \begin{cases} \epsilon & x_i = y_i \\ 1 - \epsilon & \text{else} \end{cases}$$

Markov Random Fields

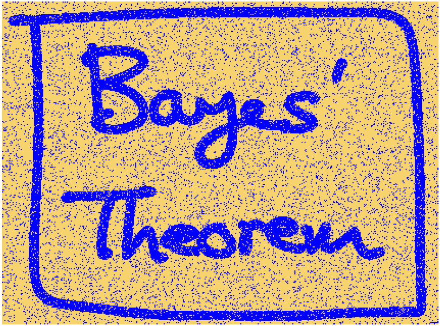
- as a factor graph

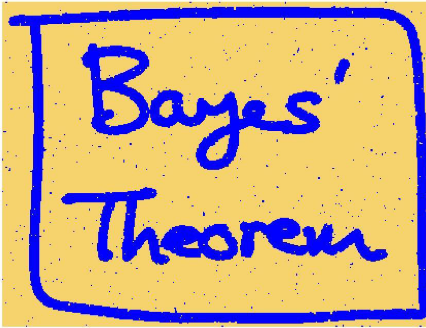


Markov Random Fields

- image denoising is an inference problem
  - for given camera image y_I compute the most probable true image

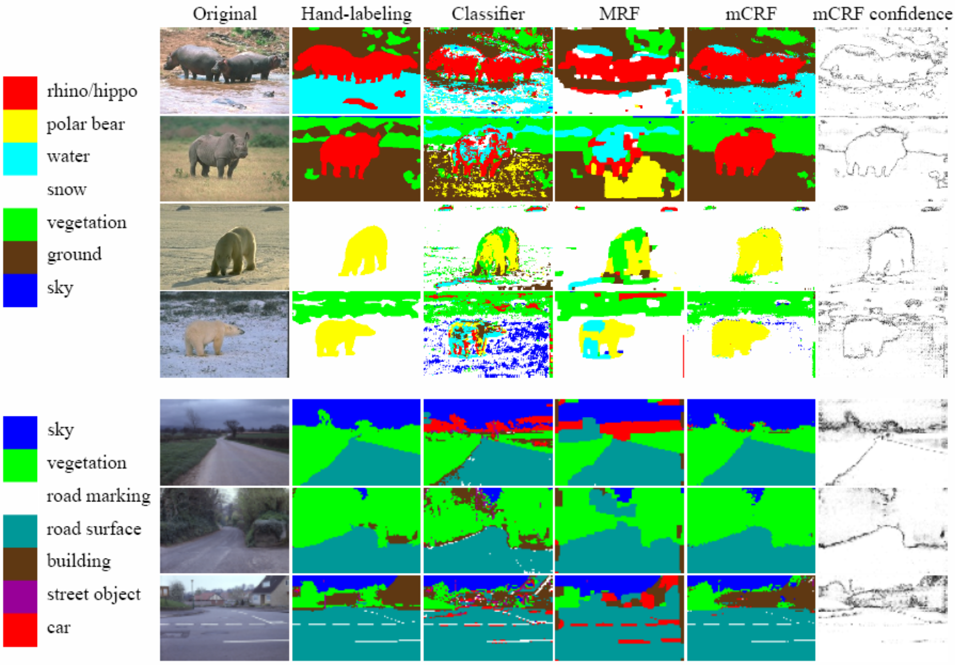
$$\underset{x_I}{\operatorname{argmax}} P(x_I, y_I)$$





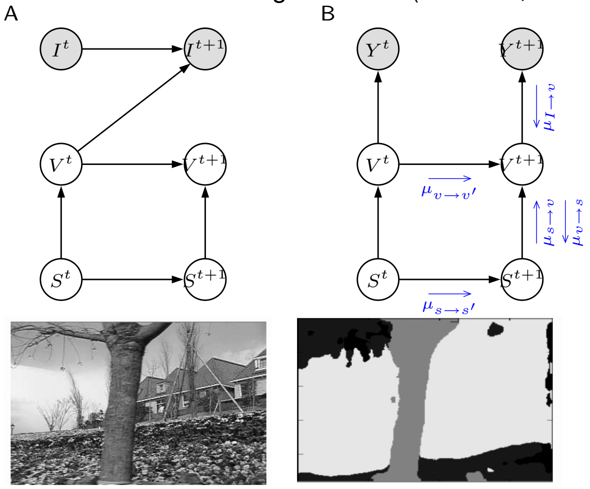
other applications

- google “conditional random field image”
  - Multiscale Conditional Random Fields for Image Labeling (CVPR 2004)
  - Scale-Invariant Contour Completion Using Conditional Random Fields (ICCV 2005)
  - Conditional Random Fields for Object Recognition (NIPS 2004)
  - Image Modeling using Tree Structured Conditional Random Fields (IJCAI 2007)
  - A Conditional Random Field Model for Video Super-resolution (ICPR 2006)


Original
Hand-labeling
Classifier
MRF
mCRF
mCRF confidence
rhino/hippo
polar bear
water
snow
vegetation
ground
sky
sky
vegetation
road marking
road surface
building
street object
car

other applications

- inference for motion segmentation (Toussaint, Willert, BMVC 2007)



- outline

  1) examples for inference & BP:

  – HMMs

  – MRFs

  2) additional comments to BP:

  – Junction Trees

  – max-product

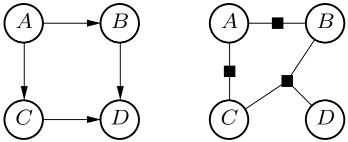
Junction Trees

- so far all messages have beed defined over *single* variables
  $\mu_{C \to i}(X_i)$
  $\mu_{i \to C}(X_i)$

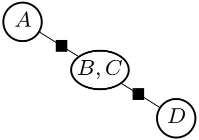
- loops can be resolved by defining larger variable groups (separators) on which messages are defined

Junction Trees – example

- example:



- joint variable B and C to a single "separator"



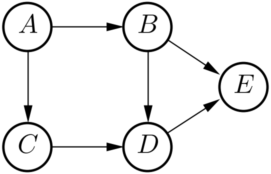
– mathematically: a variable substitution: rename the tuple (B, C) as a single random variable

$\psi_1(A, B, C) = P(B|A)\ P(A)\ P(C|A)$

$\psi_2(B, C, D) = P(D|B, C)$

this still reprents *the same old joint distribution* $P(A, B, C, D)$ – only *factored in a different way*

Junction Trees – example



...

- a variable can be contained in multiple separators – but only along a *running intersection*

Junction Tree Algorithm

- Algorithm to automatically find separators and coupling factors (=junctions) to form a tree

- graph theoretical formulation:
  – moralize a Bayes Net (= form the factor graph)
  – triangulate the graph (= insert additional links/combine variables to separators)
  – generate tree of maximal cliques (maximal spanning tree algorithm)

- here: use Elimination Algorithm to find the Junction Tree

Elimination Algorithm

$$
\begin{aligned}
&P(x_1, x_6) \\
&= \sum_{x_2} \sum_{x_3} \sum_{x_4} \sum_{x_5} \psi(x_1, x_2)\, \psi(x_3, x_1)\, \psi(x_2, x_4)\, \psi(x_3, x_5)\, \psi(x_2, x_5, x_6) \\
&= \sum_{x_2} \sum_{x_3} \sum_{x_4} \psi(x_1, x_2)\, \psi(x_3, x_1)\, \psi(x_2, x_4) \sum_{x_5} \psi(x_3, x_5)\, \psi(x_2, x_5, x_6) \\
&= \sum_{x_2} \sum_{x_3} \sum_{x_4} \psi(x_1, x_2)\, \psi(x_3, x_1)\, \psi(x_2, x_4)\, t_1(x_2, x_3, x_6) \\
&= \sum_{x_2} \sum_{x_3} \psi(x_1, x_2)\, \psi(x_3, x_1)\, t_1(x_2, x_3, x_6) \sum_{x_4} \psi(x_2, x_4) \\
&= \sum_{x_2} \sum_{x_3} \psi(x_1, x_2)\, \psi(x_3, x_1)\, t_1(x_2, x_3, x_6)\, t_2(x_2) \\
&= \sum_{x_2} \psi(x_1, x_2)\, t_2(x_2) \sum_{x_3} \psi(x_3, x_1)\, t_1(x_2, x_3, x_6) \\
&= \sum_{x_2} \psi(x_1, x_2)\, t_2(x_2)\, t_3(x_1, x_2, x_6) \\
&= t_4(x_1, x_6)
\end{aligned}
$$

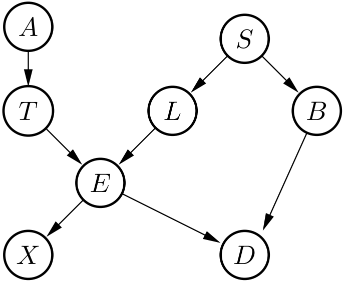
Elimination → Junction Tree

Elimination Algorithm:

- determine an elimination order
- "simulate" the iterative process of
  - eliminating a variable
  - adding a new temporary factor $t_k(\dots)$ to the factor list
- keep track of the terms!

$$\sum_{\text{variable}} \text{terms}(\underbrace{\text{some vars}}_{\text{clique}}) = t_k(\underbrace{\text{remaining vars}}_{\text{separator}})$$

example



– eliminate in order D, B, S, L, A, T, X, E

– eliminate in order E,... (not good)

– eliminate in order D, X, A, S, B, L, T, E

- on the Junction Tree, we can use BP (the special case factor-to-factor message equations) to do exact inference.

comments

- naming conventions

here	other places
exact BP on trees	sum-product algorithm, message passing algorithm, inward-outward
loopy BP	BP

- finding max configurations of random variables:

$$\underset{x_{1:n}}{\operatorname{argmax}} P(X_{1:n} = x_{1:n})$$

– max-product algorithm: replace $\sum$ by $\max$ in the message equations!
– (numerical stability: transfer to log scale and replace $\prod$ by $\sum$,
max-sum algorithm)

- read Bishop's chapter 8 (course webpage)

summary

- so far:
  - Bayes Nets & Factor Graphs
  - inference (Elimination, Belief Propagation)

- next big topic:
  - learning!