

RWTH AACHEN
UNIVERSITY

Machine Learning - Lecture 16

Inference & Applications of MRFs

30.06.2015

Bastian Leibe
RWTH Aachen
<http://www.vision.rwth-aachen.de>
leibe@vision.rwth-aachen.de

Machine Learning, WS 13/14

RWTH AACHEN
UNIVERSITY

Course Outline

- **Fundamentals (2 weeks)**
 - Bayes Decision Theory
 - Probability Density Estimation
- **Discriminative Approaches (5 weeks)**
 - Linear Discriminant Functions
 - Statistical Learning Theory & SVMs
 - Ensemble Methods & Boosting
 - Decision Trees & Randomized Trees
- **Generative Models (4 weeks)**
 - Bayesian Networks
 - Markov Random Fields
 - **Exact Inference**
 - Applications

B. Leibe 2

RWTH AACHEN
UNIVERSITY

Topics of This Lecture

- **Recap: Exact inference**
 - Sum-Product algorithm
 - Max-Sum algorithm
 - Junction Tree algorithm
- **Applications of Markov Random Fields**
 - Application examples from computer vision
 - Interpretation of clique potentials
 - Unary potentials
 - Pairwise potentials
- **Solving MRFs with Graph Cuts**
 - Graph cuts for image segmentation
 - s-t mincut algorithm
 - Extension to non-binary case
 - Applications

B. Leibe 3

RWTH AACHEN
UNIVERSITY

Recap: Factor Graphs

- **Joint probability**
 - Can be expressed as **product of factors**: $p(x) = \frac{1}{Z} \prod_s f_s(x_s)$
 - Factor graphs make this explicit through separate factor nodes.
- **Converting a directed polytree**
 - Conversion to undirected tree creates loops due to moralization!
 - Conversion to a factor graph again results in a tree!

B. Leibe 4
Image source: G. Bishop, 2006

RWTH AACHEN
UNIVERSITY

Recap: Sum-Product Algorithm

- **Objectives**
 - Efficient, **exact inference** algorithm for finding marginals.
- **Procedure:**
 - Pick an **arbitrary node** as root.
 - Compute and propagate messages **from the leaf nodes to the root**, storing received messages at every node.
 - Compute and propagate messages **from the root to the leaf nodes**, storing received messages at every node.
 - Compute the **product of received messages at each node** for which the marginal is required, and normalize if necessary.
$$p(x) \propto \prod_{s \in \text{ne}(x)} \mu_{f_s \rightarrow x}(x)$$
- **Computational effort**
 - Total number of messages = 2 · number of graph edges.

B. Leibe 5
Slide adapted from Chris Bishop

RWTH AACHEN
UNIVERSITY

Recap: Sum-Product Algorithm

- **Two kinds of messages**
 - **Message from factor node to variable nodes:**
 - Sum of factor contributions
$$\mu_{f_s \rightarrow x}(x) \equiv \sum_{X_s} F_s(x, X_s)$$

$$= \sum_{X_s} f_s(x_s) \prod_{m \in \text{ne}(f_s) \setminus x} \mu_{x_m \rightarrow f_s}(x_m)$$
 - **Message from variable node to factor node:**
 - Product of incoming messages
$$\mu_{x_m \rightarrow f_s}(x_m) \equiv \prod_{l \in \text{ne}(x_m) \setminus f_s} \mu_{f_l \rightarrow x_m}(x_m)$$

⇒ Simple propagation scheme.

B. Leibe 6

Machine Learning, WS 13/14

Recap: Sum-Product from Leaves to Root

Message definitions:

$$\mu_{f_s \rightarrow x}(x) \equiv \sum_{X_s} f_s(\mathbf{x}_s) \prod_{m \in \text{ne}(f_s) \setminus x} \mu_{x_m \rightarrow f_s}(x_m)$$

$$\mu_{x_m \rightarrow f_s}(x_m) \equiv \prod_{l \in \text{ne}(x_m) \setminus f_s} \mu_{f_l \rightarrow x_m}(x_m)$$

$\mu_{x \rightarrow f}(x) = 1$ $\mu_{f \rightarrow x}(x) = f(x)$

B. Leibe 7
Image source: C. Bishop, 2006

Machine Learning, WS 13/14

Recap: Sum-Product from Root to Leaves

Message definitions:

$$\mu_{f_s \rightarrow x}(x) \equiv \sum_{X_s} f_s(\mathbf{x}_s) \prod_{m \in \text{ne}(f_s) \setminus x} \mu_{x_m \rightarrow f_s}(x_m)$$

$$\mu_{x_m \rightarrow f_s}(x_m) \equiv \prod_{l \in \text{ne}(x_m) \setminus f_s} \mu_{f_l \rightarrow x_m}(x_m)$$

$\mu_{x \rightarrow f}(x) = 1$ $\mu_{f \rightarrow x}(x) = f(x)$

B. Leibe 8
Image source: C. Bishop, 2006

Machine Learning, WS 13/14

Recap: Max-Sum Algorithm

- Objective: an efficient algorithm for finding**
 - Value \mathbf{x}^{\max} that maximises $p(\mathbf{x})$;
 - Value of $p(\mathbf{x}^{\max})$.
- \Rightarrow Application of dynamic programming in graphical models.
- Key ideas**
 - We are interested in the maximum value of the joint distribution

$$p(\mathbf{x}^{\max}) = \max_{\mathbf{x}} p(\mathbf{x})$$
 - \Rightarrow Maximize the product $p(\mathbf{x})$.
 - For numerical reasons, use the logarithm.

$$\ln \left(\max_{\mathbf{x}} p(\mathbf{x}) \right) = \max_{\mathbf{x}} \ln p(\mathbf{x}).$$
 - \Rightarrow Maximize the sum (of log-probabilities).

Slide adapted from Chris Bishop. 9
B. Leibe

Machine Learning, WS 13/14

Recap: Max-Sum Algorithm

- Initialization (leaf nodes)**

$$\mu_{x \rightarrow f}(x) = 0 \quad \mu_{f \rightarrow x}(x) = \ln f(x)$$
- Recursion**
 - Messages**

$$\mu_{f \rightarrow x}(x) = \max_{x_1, \dots, x_M} \left[\ln f(x, x_1, \dots, x_M) + \sum_{m \in \text{ne}(f_s) \setminus x} \mu_{x_m \rightarrow f}(x_m) \right]$$

$$\mu_{x \rightarrow f}(x) = \sum_{l \in \text{ne}(x) \setminus f} \mu_{f_l \rightarrow x}(x)$$
 - For each node, keep a record of which values of the variables gave rise to the maximum state:

$$\phi(x) = \arg \max_{x_1, \dots, x_M} \left[\ln f(x, x_1, \dots, x_M) + \sum_{m \in \text{ne}(f_s) \setminus x} \mu_{x_m \rightarrow f}(x_m) \right]$$

Slide adapted from Chris Bishop. 10
B. Leibe

Machine Learning, WS 13/14

Recap: Max-Sum Algorithm

- Termination (root node)**
 - Score of maximal configuration

$$p^{\max} = \max_x \left[\sum_{s \in \text{ne}(x)} \mu_{f_s \rightarrow x}(x) \right]$$
 - Value of root node variable giving rise to that maximum

$$x^{\max} = \arg \max_x \left[\sum_{s \in \text{ne}(x)} \mu_{f_s \rightarrow x}(x) \right]$$
 - Back-track to get the remaining variable values

$$x_{n-1}^{\max} = \phi(x_n^{\max})$$

Slide adapted from Chris Bishop. 11
B. Leibe

Machine Learning, WS 13/14

Topics of This Lecture

- Factor graphs**
 - Construction
 - Properties
- Sum-Product Algorithm for computing marginals**
 - Key ideas
 - Derivation
 - Example
- Max-Sum Algorithm for finding most probable value**
 - Key ideas
 - Derivation
 - Example
- Algorithms for loopy graphs**
 - Junction Tree algorithm
 - Loopy Belief Propagation

12
B. Leibe

RWTH AACHEN UNIVERSITY

Junction Tree Algorithm

- **Motivation**
 - **Exact** inference on general graphs.
 - Works by turning the initial graph into a **junction tree** and then running a sum-product-like algorithm.
 - **Intractable** on graphs with large cliques.
- **Main steps**
 1. If starting from directed graph, first convert it to an undirected graph by **moralization**.
 2. Introduce additional links by **triangulation** in order to reduce the size of cycles.
 3. **Find cliques** of the moralized, triangulated graph.
 4. Construct a new graph from the **maximal cliques**.
 5. Remove minimal links to **break cycles** and get a **junction tree**.
⇒ Apply regular **message passing** to perform inference.

13

Machine Learning, WS 13/14 B. Leibe

RWTH AACHEN UNIVERSITY

Junction Tree Algorithm

- Starting from an directed graph...

14

Machine Learning, WS 13/14 Slide adapted from Zoubin Gharahmani B. Leibe Image source: Z. Gharahmani

RWTH AACHEN UNIVERSITY

Junction Tree Algorithm

1. Convert to an undirected graph through **moralization**.
 - Marry the parents of each node.
 - Remove edge directions.

15

Machine Learning, WS 13/14 Slide adapted from Zoubin Gharahmani B. Leibe Image source: Z. Gharahmani

RWTH AACHEN UNIVERSITY

Junction Tree Algorithm

2. **Triangulate**
 - Such that there is **no loop of length > 3** without a chord.
 - This is necessary so that the final junction tree satisfies the **"running intersection"** property (explained later).

16

Machine Learning, WS 13/14 Slide adapted from Zoubin Gharahmani B. Leibe Image source: Z. Gharahmani

RWTH AACHEN UNIVERSITY

Junction Tree Algorithm

3. **Find cliques** of the moralized, triangulated graph.

17

Machine Learning, WS 13/14 Slide adapted from Zoubin Gharahmani B. Leibe Image source: Z. Gharahmani

RWTH AACHEN UNIVERSITY

Junction Tree Algorithm

4. Construct a new **junction graph** from maximal cliques.
 - Create a node from each clique.
 - Each link carries a list of all variables in the intersection.
 - Drawn in a **"separator"** box.

18

Machine Learning, WS 13/14 B. Leibe Image source: Z. Gharahmani

RWTH AACHEN UNIVERSITY

Junction Tree Algorithm

5. Remove links to break cycles \Rightarrow junction tree.

- For each cycle, remove a link with the minimal number of shared nodes until all cycles are broken.
- Result is a maximal spanning tree, the **junction tree**.

19
B. Leibe
Image source: Z. Gharahmani

RWTH AACHEN UNIVERSITY

Junction Tree - Properties

- Running intersection property**
 - "If a variable appears in more than one clique, it also appears in all intermediate cliques in the tree".
 - This ensures that neighboring cliques have consistent probability distributions.
 - Local consistency \rightarrow global consistency

20
B. Leibe
Slide adapted from Zoubin Gharahmani

RWTH AACHEN UNIVERSITY

Interpretation of the Junction Tree

- Undirected graphical model

- Junction tree

Clique Separator Clique
 $P(U) = \prod P(\text{Clique}) / \prod P(\text{Separator})$
 $P(A, B, C) = P(A, B) P(B, C) / P(B)$

21
B. Leibe
Slide adapted from Pawan Kumar

RWTH AACHEN UNIVERSITY

Junction Tree: Example 1

- Algorithm**
 - Moralization
 - Triangulation (not necessary here)

22
B. Leibe
Image source: J. Pearl, 1988

RWTH AACHEN UNIVERSITY

Junction Tree: Example 1

- Algorithm**
 - Moralization
 - Triangulation (not necessary here)
 - Find cliques
 - Construct junction graph

23
B. Leibe
Image source: J. Pearl, 1988

RWTH AACHEN UNIVERSITY

Junction Tree: Example 1

- Algorithm**
 - Moralization
 - Triangulation (not necessary here)
 - Find cliques
 - Construct junction graph
 - Break links to get junction tree

24
B. Leibe
Image source: J. Pearl, 1988

Machine Learning, WS 13/14

Junction Tree: Example 2

- Without triangulation step
 - The final graph will contain cycles that we cannot break without losing the running intersection property!

25
B. Leibe Image source: J. Pearl, 1988

Machine Learning, WS 13/14

Junction Tree: Example 2

- When applying the triangulation
 - Only small cycles remain that are easy to break.
 - Running intersection property is maintained.

26
B. Leibe Image source: J. Pearl, 1988

Machine Learning, WS 13/14

Junction Tree Algorithm

- Good news
 - The junction tree algorithm is efficient in the sense that for a given graph there does not exist a computationally cheaper approach.
- Bad news
 - This may still be too costly.
 - Effort determined by number of variables in the largest clique.
 - Grows exponentially with this number (for discrete variables).

⇒ Algorithm becomes impractical if the graph contains large cliques!

27
B. Leibe

Machine Learning, WS 13/14

Loopy Belief Propagation

- Alternative algorithm for loopy graphs
 - Sum-Product on general graphs.
 - Strategy: **simply ignore the problem.**
 - Initial unit messages passed across all links, after which messages are passed around until convergence
 - Convergence is not guaranteed!
 - Typically break off after fixed number of iterations.
 - **Approximate** but **tractable** for large graphs.
 - Sometime works well, sometimes not at all.

28
B. Leibe

Machine Learning, WS 13/14

Topics of This Lecture

- Recap: Exact inference
 - Sum-Product algorithm
 - Max-Sum algorithm
 - Junction Tree algorithm
- Applications of Markov Random Fields
 - Application examples from computer vision
 - Interpretation of clique potentials
 - Unary potentials
 - Pairwise potentials
- Solving MRFs with Graph Cuts
 - Graph cuts for image segmentation
 - s-t mincut algorithm
 - Extension to non-binary case
 - Applications

29
B. Leibe

Machine Learning, WS 13/14

Markov Random Fields (MRFs)

- What we've learned so far...
 - We know they are **undirected graphical models**.
 - Their joint probability factorizes into **clique potentials**,

$$p(\mathbf{x}) = \frac{1}{Z} \prod_C \psi_C(\mathbf{x}_C)$$

which are conveniently expressed as **energy functions**.

$$\psi_C(\mathbf{x}_C) = \exp\{-E(\mathbf{x}_C)\}$$

- We know how to perform inference for them.
 - **Sum/Max-Product BP** for exact inference in tree-shaped MRFs.
 - **Loopy BP** for approximate inference in arbitrary MRFs.
 - **Junction Tree** algorithm for converting arbitrary MRFs into trees.
- But what are they actually good for?
 - And how do we apply them in practice?

30
B. Leibe Image source: C. Bishop, 2006

Machine Learning, WS 13/14

Markov Random Fields

- Allow rich probabilistic models.
 - But built in a local, modular way.
 - Learn local effects, get global effects out.
- Very powerful when applied to regular structures.
 - Such as images...

Observed evidence

Hidden "true states"

Neighborhood relations

Slide adapted from William Freeman. B. Leibe

31

Machine Learning, WS 13/14

Applications of MRFs

- Movie "No Way Out" (1987)

B. Leibe

32

Machine Learning, WS 13/14

Applications of MRFs

- Many applications for low-level vision tasks
 - Image denoising

B. Leibe

Results by [Roth & Black, CVPR'05]

33

Machine Learning, WS 13/14

Applications of MRFs

- Many applications for low-level vision tasks
 - Image denoising

Noisy observations

"True" image content

"Smoothness constraints"

B. Leibe

Results by [Roth & Black, CVPR'05]

34

Machine Learning, WS 13/14

Applications of MRFs

- Many applications for low-level vision tasks
 - Image denoising
 - Inpainting

Since 1870, when French settlers landed at the mouth of the Mississippi River and celebrated the first Mardi Gras in New Orleans, has brewed a fascinating melange of cultures. It was French, then Spanish, then French again, then Italian, the British, through all the years, but after the 1900s, came another mix everywhere: African (Cajun), African, Indian.

B. Leibe

Results by [Roth & Black, CVPR'05]

35

Machine Learning, WS 13/14

Applications of MRFs

- Many applications for low-level vision tasks
 - Image denoising
 - Inpainting
 - Image restoration

B. Leibe

Results by [Roth & Black, CVPR'05]



36

Machine Learning, WS 13/14

RWTH AACHEN UNIVERSITY

Applications of MRFs

- Many applications for low-level vision tasks
 - Image denoising
 - Inpainting
 - Image restoration
 - Image segmentation

B. LeibeImage source: Pawan M. Kumar


37

Machine Learning, WS 13/14


RWTH AACHEN UNIVERSITY

Applications of MRFs

- Many applications for low-level vision tasks
 - Image denoising
 - Inpainting
 - Image restoration
 - Image segmentation
- Super-resolution
 - Convert a low-res image into a high-res image!



upsampling



B. LeibeImage source: [Freeman et al., CG&A'03]

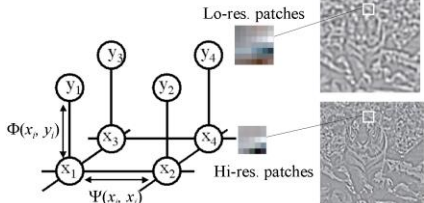
38

Machine Learning, WS 13/14


RWTH AACHEN UNIVERSITY

Applications of MRFs


- Many applications for low-level vision tasks
 - Image denoising
 - Inpainting
 - Image restoration
 - Image segmentation
- Super-resolution
 - Convert a low-res image into a high-res image!



Lo-res. patches



Hi-res. patches



B. LeibeImage source: [Freeman et al., CG&A'03]

39

Machine Learning, WS 13/14

RWTH AACHEN UNIVERSITY

Applications of MRFs

- Many applications for low-level vision tasks
 - Image denoising
 - Inpainting
 - Image restoration
 - Image segmentation
- Super-resolution
 - Convert a low-res image into a high-res image!
- Optical flow
 - Estimate motion between image pairs.

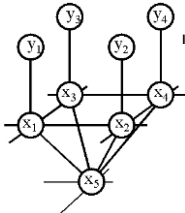


Image patches

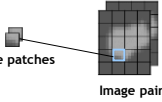
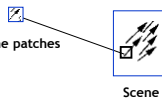
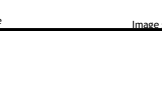


Image pair



Scene patches



Scene

B. LeibeImage source: William Freeman



40

Machine Learning, WS 13/14

RWTH AACHEN UNIVERSITY

Applications of MRFs

- Many applications for low-level vision tasks
 - Image denoising
 - Inpainting
 - Image restoration
 - Image segmentation
- Super-resolution
 - Convert a low-res image into a high-res image!
- Optical flow
 - Estimate motion between image pairs.
- Stereo depth estimation
 - Estimate depth from stereo image pairs.

B. Leibe

42

Machine Learning, WS 13/14

RWTH AACHEN UNIVERSITY

Applications of MRFs

- Many applications for low-level vision tasks
 - Image denoising
 - Inpainting
 - Image restoration
 - Image segmentation
- Super-resolution
 - Convert a low-res image into a high-res image!
- Optical flow
 - Estimate motion between image pairs.
- Stereo depth estimation
 - Estimate depth from stereo image pairs.

- MRFs have become a standard tool for such tasks.
 - Let's look at how they are applied in detail...

B. Leibe

43

Machine Learning, WS 13/14

MRF Structure for Images

- Basic structure
 - Noisy observations
 - "True" image content
- Two components
 - Observation model
 - How likely is it that node x_i has label L_i given observation y_i ?
 - This relationship is usually learned from training data.
 - Neighborhood relations
 - Simplest case: 4-neighborhood
 - Serve as smoothing terms.
 - ⇒ Discourage neighboring pixels to have different labels.
 - This can either be learned or be set to fixed "penalties".

B. Leibe 44

Machine Learning, WS 13/14

MRF Nodes as Pixels

Original image Degraded image Reconstruction from MRF modeling pixel neighborhood statistics

These neighborhood statistics can be learned from training data!

Slide adapted from William Freeman B. Leibe 45

Machine Learning, WS 13/14

MRF Nodes as Patches

Image patches Scene patches Image Scene

More general relationships expressed by potential functions Φ and Ψ .

Slide credit: William Freeman B. Leibe 47

Machine Learning, WS 13/14

Network Joint Probability

- Interpretation of the factorized joint probability

$$P(x, y) = \prod_i \Phi(x_i, y_i) \prod_{i,j} \Psi(x_i, x_j)$$

Scene Image Image-scene compatibility function Local observations Scene-scene compatibility function Neighboring scene nodes

Slide credit: William Freeman B. Leibe 48

Machine Learning, WS 13/14

Energy Formulation

- Energy function

$$E(x, y) = \sum_i \underbrace{\phi(x_i, y_i)}_{\text{Single-node potentials}} + \sum_{i,j} \underbrace{\psi(x_i, x_j)}_{\text{Pairwise potentials}}$$
- Single-node (unary) potentials ϕ
 - Encode local information about the given pixel/patch.
 - How likely is a pixel/patch to belong to a certain class (e.g. foreground/background)?
- Pairwise potentials ψ
 - Encode neighborhood information.
 - How different is a pixel/patch's label from that of its neighbor? (e.g. based on intensity/color/texture difference, edges)

B. Leibe 49

Machine Learning, WS 13/14

How to Set the Potentials? Some Examples

- Unary potentials
 - E.g., color model, modeled with a Mixture of Gaussians

$$\phi(x_i, y_i; \theta_\phi) = \log \sum_k \theta_\phi(x_i, k) p(k|x_i) \mathcal{N}(y_i; \bar{y}_k, \Sigma_k)$$
 - ⇒ Learn color distributions for each label

B. Leibe 50

RWTH AACHEN UNIVERSITY

How to Set the Potentials? Some Examples

- **Pairwise potentials**
 - **Potts Model**

$$\psi(x_i, x_j; \theta_\psi) = \theta_\psi \delta(x_i \neq x_j)$$
 - Simplest discontinuity preserving model.
 - Discontinuities between any pair of labels are penalized equally.
 - Useful when labels are unordered or number of labels is small.
 - Extension: “contrast sensitive Potts model”

$$\psi(x_i, x_j, g_{ij}(y); \theta_\psi) = \theta_\psi g_{ij}(y) \delta(x_i \neq x_j)$$
 where

$$g_{ij}(y) = e^{-\beta \|y - y_j\|^2} \quad \beta = 2 \cdot \text{avg} \left(\|y_i - y_j\|^2 \right)$$
 - Discourages label changes except in places where there is also a large change in the observations.

51

RWTH AACHEN UNIVERSITY

Extension: Conditional Random Fields (CRF)

- **Idea: Model conditional instead of joint probability**

- **Energy formulation**

$$E(\mathbf{x}) = \sum_{i \in S} \left(\phi(\mathbf{D}|\mathbf{x}_i) + \sum_{j \in \mathcal{N}_i} (\phi(\mathbf{D}|\mathbf{x}_i, \mathbf{x}_j) + \psi(\mathbf{x}_i, \mathbf{x}_j)) \right) + \text{const}$$

Unary likelihood Contrast Term Uniform Prior (Potts Model)

52

RWTH AACHEN UNIVERSITY

Example: MRF for Image Segmentation

- **MRF structure**

53

RWTH AACHEN UNIVERSITY

Energy Minimization

- **Goal:**
 - Infer the optimal labeling of the MRF.
- **Many inference algorithms are available, e.g.**
 - Simulated annealing ← *What you saw in the movie.*
 - Iterated conditional modes (ICM) ← *Too simple.*
 - Belief propagation ← *Last lecture*
 - Graph cuts ← *Next Lecture!*
 - Variational methods } ← *For more complex problems*
 - Monte Carlo sampling }
- **Recently, Graph Cuts have become a popular tool**
 - Only suitable for a certain class of energy functions.
 - But the solution can be obtained very fast for typical vision problems (~1MPixel/sec).

54

RWTH AACHEN UNIVERSITY

References and Further Reading

- A gentle introduction to Graph Cuts can be found in the following paper:
 - Y. Boykov, O. Veksler, Graph Cuts in Vision and Graphics: Theories and Applications. In *Handbook of Mathematical Models in Computer Vision*, edited by N. Paragios, Y. Chen and O. Faugeras, Springer, 2006.
- Try the GraphCut implementation at <http://www.cs.ucl.ac.uk/staff/V.Kolmogorov/software.html>

96