Machine Learning - Lecture 14

MRF Applications & Graph Cuts

30.06.2009

Bastian Leibe
RWTH Aachen
http://www.umic.rwth-aachen.de/multimedia
leibe@umic.rwth-aachen.de

Topics of This Lecture

• Recap: Exact inference
  - Factor Graphs
  - Sum-Product/Max-Sum Belief Propagation
  - 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
  - x-t minimax algorithm
  - Extension to non-binary case
  - Applications

Recap: Factor Graphs

• Joint probability
  - Can be expressed as product of factors: \( p(x) = \frac{1}{Z} \prod f_i(x_i) \)
  - 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!

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.
  - Computational effort
    - Total number of messages = 2 * number of graph edges.

Course Outline

• Fundamentals (2 weeks)
  - Bayes Decision Theory
  - Probability Density Estimation
• Discriminative Approaches (5 weeks)
  - Lin. Discriminants, SVMs, Boosting
• Graphical Models (5 weeks)
  - Bayesian Networks & Applications
  - Markov Random Fields & Applications
  - Exact Inference
  - Approximate Inference
• Regression Problems (2 weeks)
  - Gaussian Processes

Slide adapted from Chris Bishop, 2006.

Recap: Sum-Product Algorithm

• Two kinds of messages
  - Message from factor node to variable nodes:
    - Sum of factor contributions
    - Message from variable node to factor node:
      - Product of incoming messages
      - Simple propagation scheme.
Recap: 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.
Recap: Junction Tree Example

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

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

Topics of This Lecture

- Recap: Exact inference
  - Factor Graphs
  - Sum-Product/Max-Sum Belief Propagation
  - 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
  - a theoretical algorithm
  - Extensions to non-binary case
  - Applications

Markov Random Fields (MRFs)

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

\[ p(x) = \frac{1}{Z} \prod C \psi_C(x_C) \]

which are conveniently expressed as energy functions.

\[ \psi_C(x_C) = \exp(-E(x_C)) \]

- 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?

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...

Applications of MRFs

- Movie “No Way Out” (1987)
Applications of MRFs

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

“True” image content
Noisy observations
“Smoothness constraints”

Applications of MRFs

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

Convert a low-res image into a high-res image!
Applications of MRFs

- Many applications for low-level vision tasks
  - Image denoising
  - Inpainting
  - Image restoration
  - Image segmentation
  - Super-resolution
  - Optical flow
  - Texture synthesis
  - Stereo depth estimation

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".

MRFs have become a standard tool for such tasks.

Let’s look at how they are applied in detail...
**Perceptual and Sensory Augmented Computing**

**How to Set the Potentials? Some Examples**

- **Unary potentials**
  - E.g. color model, modeled with a Mixture of Gaussians
  \[ \phi(x_i, y_l; \beta) = \log \sum_k \beta_k p(k|x_i) N(y_l; \gamma_k, \Sigma_k) \]
  - Learn color distributions for each label

- **Pairwise potentials**
  - 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)

**Simple Binary Image Denoising Model**

- **MRF Structure**
  - Observation process
    \[ E(x, y) = h \sum x_i + \beta \sum \delta(x_i \neq y_i) - \eta \sum x_i y_i \]
  - Example: simple energy function ("Potts model")
    \[ E(x, y) = \sum x_i + \beta \sum \delta(x_i \neq y_i) \]
  - Prior
  - Smoothness
  - Observation

- **Smoothness term**: fixed penalty \( \beta \) if neighboring labels disagree.
- **Observation term**: fixed penalty \( \eta \) if label and observation disagree.

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

**Energy Formulation**

- **Energy function**
  \[ E(x, y) = \sum \phi(x_i, y_i) + \sum \psi(x_i, x_j) \]
  - Single-node potentials \( \phi \)
  - Pairwise potentials \( \psi \)
  - How likely is a pixel/patch to belong to a certain class (e.g. foreground/background)?
  - 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)
How to Set the Potentials? Some Examples

- **Pairwise potentials**
  - **Potts Model**
    \[ \psi(x_i, x_j; \theta) = \theta \delta(x_i \neq x_j) \]
    - 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”
    \[ \psi(x_i, x_j, y_i, y_j; \theta, \beta) = \theta \delta(x_i \neq x_j) \]
    - Discourages label changes except in places where there is also a large change in the observations.

Example: MRF for Image Segmentation

- **MRF structure**
  - Unary potential: \( \phi(D|x_i) \)
  - Pairwise potential: \( \phi(D|x_i, x_j) \)
  - Energy formulation:
    \[
    E(x) = \sum_{i \in X} \phi(D|x_i) + \sum_{i \neq j} \phi(D|x_i, x_j) + \text{const}
    \]
  - **Goal:** Infer the optimal labeling of the MRF.
  - Many inference algorithms are available, e.g.
    - Simulated annealing
    - Iterated conditional modes (ICM)
    - Belief propagation
    - Graph cuts
    - Variational methods
    - 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).

Graph Cuts for Binary Problems

- **Idea:** convert MRF into source-sink graph
  - Minimum cost cut can be computed in polynomial time
    \[
    w_{ij} = \exp\left( -\frac{\psi(x_i, x_j)}{2\sigma^2} \right)
    \]
  - s-t mincut algorithm
  - Applications

Topics of This Lecture

- Recap: Exact Inference
  - Factor graphs
  - Sum-Product/Max-product Belief Propagation
  - Junction Tree algorithm
- Applications of Markov Random Fields
  - Application examples from computer vision
    - interpretation of cinque pawns
      - Unary potentials
      - Pairwise potentials
- Solving MRFs with Graph Cuts
  - Graph cuts for image segmentation
  - s-t mincut algorithm
  - Extension to non-binary case
  - Applications
Simple Example of Energy

\[ E(L) = \sum_p D_p(L_p) + \sum_{pq} W_{pq} \delta(L_p \neq L_q) \]

Regional term \( D_p(L_p) \)

Boundary term \( \delta(L_p \neq L_q) \)

\( L_p \in \{S, I\} \)

(binary object segmentation)

Adding Regional Properties

Regional bias example

Suppose \( I' \) and \( I'' \) are given “expected” intensities of object and background

\[ D_p(L_p) = -\log p(I_p|L_p) \]

Object and background color distributions

Topics of This Lecture

- Recap: Exact Inference
  - Factor graphs
  - Sum-Product/Max-Product Belief Propagation
  - Junction Tree algorithm
- Applications of Markov Random Fields
  - Application examples from computer vision
  - Interpretation of clique potentials
  - Unary potentials
  - Pairwise potentials
- More generally, regional bias can be based on any intensity/color models of object and background.

How Does it Work? The s-t-Mincut Problem

Graph (\( V, E, C \))

Vertices \( V = \{v_1, v_2, v_3\} \)

Edges \( E = \{(v_1, v_2), (v_1, v_3)\} \)

Costs \( C = (c_{12}, c_{13}) \)
**The s-t-Mincut Problem**

**What is an st-cut?**
An st-cut \((S, T)\) divides the nodes between source and sink.

**What is the cost of a st-cut?**
Sum of cost of all edges going from \(S\) to \(T\).

**Sum of cost of all edges going from \(S\) to \(T\):**
\[5 + 2 + 9 = 16\]

**What is the st-mincut?**
**st-cut with the minimum cost**

**How to Compute the s-t-Mincut?**

- Solve the dual maximum flow problem
- Compute the maximum flow between Source and Sink

**Constraints**
- Edges: Flow < Capacity
- Nodes: Flow in = Flow out

**Min-cut/Max-flow Theorem**
In every network, the maximum flow equals the cost of the st-mincut.

**History of Maxflow Algorithms**

**Augmenting Path and Push-Relabel**

<table>
<thead>
<tr>
<th>Year</th>
<th>Discoverer(s)</th>
<th>Based on</th>
</tr>
</thead>
<tbody>
<tr>
<td>1956</td>
<td>Dinic's</td>
<td>Dinic's algorithm</td>
</tr>
<tr>
<td>1957</td>
<td>Edmonds</td>
<td>Edmonds's algorithm</td>
</tr>
<tr>
<td>1958</td>
<td>Edmonds &amp; Karp</td>
<td>Edmonds-Karp algorithm</td>
</tr>
<tr>
<td>1962</td>
<td>Goldberg &amp; Tarjan</td>
<td>Goldberg-Tarjan algorithm</td>
</tr>
<tr>
<td>1974</td>
<td>Karzanov</td>
<td>Karzanov's algorithm</td>
</tr>
<tr>
<td>1976</td>
<td>Cherkassky</td>
<td>Cherkassky's algorithm</td>
</tr>
<tr>
<td>1990</td>
<td>Gold &amp; Shearer</td>
<td>Gold-Shearer algorithm</td>
</tr>
<tr>
<td>1993</td>
<td>Sleator &amp; Tarjan</td>
<td>Sleator-Tarjan algorithm</td>
</tr>
<tr>
<td>1986</td>
<td>Goldberg &amp; Tarjan</td>
<td>Goldberg-Tarjan algorithm</td>
</tr>
<tr>
<td>1997</td>
<td>Ahuja &amp; Orlin</td>
<td>Ahuja-Orlin algorithm</td>
</tr>
<tr>
<td>1999</td>
<td>Ahuja et al.</td>
<td>Ahuja et al.'s algorithm</td>
</tr>
<tr>
<td>1992</td>
<td>Acar</td>
<td>Acar's algorithm</td>
</tr>
<tr>
<td>1994</td>
<td>King</td>
<td>King's algorithm</td>
</tr>
<tr>
<td>1997</td>
<td>Goldberg &amp; Rao</td>
<td>Goldberg-Rao algorithm</td>
</tr>
</tbody>
</table>

**Maxflow Algorithms**

**Augmenting Path Based Algorithms**

1. Find path from source to sink with positive capacity
2. Push maximum possible flow through this path
3. Repeat until no path can be found
Maxflow Algorithms

Augmenting Path Based Algorithms

1. Find path from source to sink with positive capacity
2. Push maximum possible flow through this path
3. Repeat until no path can be found

Algorithms assume non-negative capacity

Flow = 0 + 2

Source

1

2-2

V1

V2

9

5-2

Sink

Flow = 2

Source

1

2

V1

V2

3

2

4

Sink

Flow = 2 + 4

Source

5

1

V1

V2

3

2

0

Sink

Flow = 6

Source

5

1

V1

V2

3

2

0

Sink

Flow = 2

Source

9

1

0

V1

V2

3

2

4

Sink

Flow = 2

Source

9

1

0

V1

V2

3

2

4

Sink
Maxflow Algorithms

Flow = 6

Augmenting Path Based Algorithms

1. Find path from source to sink with positive capacity
2. Push maximum possible flow through this path
3. Repeat until no path can be found

Algorithms assume non-negative capacity

Flow = 6 + 1

Augmenting Path Based Algorithms

1. Find path from source to sink with positive capacity
2. Push maximum possible flow through this path
3. Repeat until no path can be found

Algorithms assume non-negative capacity

Augmenting Path Based Algorithms

1. Find path from source to sink with positive capacity
2. Push maximum possible flow through this path
3. Repeat until no path can be found

Algorithms assume non-negative capacity

Applications: Maxflow in Computer Vision

- Specialized algorithms for vision problems
  - Grid graphs
  - Low connectivity (m - O(n))
- Dual search tree augmenting path algorithm
  [Boykov and Kolmogorov PAMI 2004]
  - Finds approximate shortest augmenting paths efficiently.
  - Empirically outperforms other algorithms on vision problems.
  - Efficient code available on the web
  http://www.cs.ucl.ac.uk/staff/V.Kolmogorov/software.html

When Can s-t Graph Cuts Be Applied?

\[
E(L) = \sum \{ E_p(L_p) \} \quad \text{Regional term}
\]
\[
\sum \{ E(L_p, L_n) \} \quad \text{Boundary term}
\]

\[
L_{t,n} \in \{ s,t \}
\]

- s-t graph cuts can only globally minimize binary energies that are submodular.
- Submodularity is the discrete equivalent to convexity.
  Implies that every local energy minimum is a global minimum.
  Solution will be globally optimal.
Topics of This Lecture

- Recap: Exact inference
  - Factor graphs
  - Sum-Product/Max-Sum Belief Propagation
  - 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

Dealing with Non-Binary Cases

- Limitation to binary energies is often a nuisance. ⇒ E.g., binary segmentation only...
- We would like to solve also multi-label problems. The bad news: Problem is NP-hard with 3 or more labels!
- There exist some approximation algorithms which extend graph cuts to the multi-label case:
  - α-Expansion
  - αβ-Swap
- They are no longer guaranteed to return the globally optimal result.
  - But α-Expansion has a guaranteed approximation quality (2-approx) and converges in a few iterations.

α-Expansion Move

- Basic idea:
  - Break multi-way cut computation into a sequence of binary s-t cuts.

Example: Stereo Vision

α-Expansion Moves

- In each α-expansion a given label “α” grabs space from other labels

For each move, we choose the expansion that gives the largest decrease in the energy: binary optimization problem
Topics of This Lecture

- Recap: Exact inference
  - Factor graphs
  - Sum-Product/Max-Sum Belief Propagation
  - 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

GraphCut Applications: “GrabCut”

- Interactive Image Segmentation [Boykov & Jolly, ICCV'01]
  - Rough region cues sufficient
  - Segmentation boundary can be extracted from edges
- Procedure
  - User marks foreground and background regions with a brush.
  - This is used to create an initial segmentation which can then be corrected by additional brush strokes.

GrabCut: Data Model

- Obtained from interactive user input
  - User marks foreground and background regions with a brush
  - Alternatively, user can specify a bounding box

GrabCut: Coherence Model

- An object is a coherent set of pixels:

\[ \psi(x, y) = \gamma \sum_{(m,n) \in C} \delta[x_m \neq x_n] c^{1 - \delta(x_m, x_n)} \]

Graph Cuts for Image Segmentation

- Segmentation by s-t mincut
  - Cut: separating source and sink
  - Min Cut: Global minimal energy in polynomial time (1MPixel/sec)
  - s-t MinCut Problem is equal to MaxFlow Problem [Fulkerson 56]

Iterated Graph Cuts

- Energy after each iteration
References and Further Reading

- A gentle introduction to Graph Cuts can be found in the following paper:

- Try the GraphCut implementation at http://www.cs.ucl.ac.uk/staff/V.Kolmogorov/software.html