Markov Random Fields in Vision

Many slides drawn from presentations by Simon Prince/UCL and Kevin Wayne/Princeton
Image Denoising

Foreground Extraction

Stereo Disparity
Why study MRFs?

• Image denoising is based on modeling what kinds of images are more probable
• Foreground extraction is based on modeling what spatial distribution of foreground pixels is more probable
• Stereo disparity estimation is based on modeling what kinds of disparity fields are more probable
Modeling the joint probability distribution

- Associate a random variable with each pixel

\[ P_r (x_1, x_2, \ldots, x_{16}) \]

For a binary image, this requires specifying 2^{16} numbers for the probability of each configuration.
Conditional independence assumptions necessary for tractability

- Directed graphical models (a.k.a. Bayes nets, Belief networks)
A Markov Random Field is determined by

- a set of sites $S = \{1 \ldots N\}$. These will correspond to the $N$ pixel locations,
- a set of random variables $y = \{y_1 \ldots y_N\}$ associated with each of the sites,
- a set of neighbors $\mathcal{N}_{1\ldots N}$ at each of the $N$ sites. The set $\mathcal{N}_n$ contains the indices of the subset of random variables have an immediate probabilistic connection to variable $y_n$.

To be a Markov random field, the model must obey the Markov property,

$$Pr(y_n|y_{S\setminus n}) = Pr(y_n|y_{N_n}) \quad \forall \quad n \in S,$$
Example: Image with 4-connected pixels

\[ P_r (X_8 | \text{all other nodes}) \]
\[ = P_r (X_8 | X_3, X_7, X_9, X_{13}) \]
Hammersley Clifford Theorem

Any positive distribution that obeys the Markov property

\[ Pr(y_n | y_{S \setminus n}) = Pr(y_n | y_{N_n}) \quad \forall \quad n \in S, \]

can be written in the form

\[ Pr(y) = \frac{1}{Z} \exp \left[ - \sum_{c \in C} \Psi_c(y) \right] \]

Where the \( c \) terms are maximal cliques

Cliques = subsets of variables that all connect to each other.

Maximal = cannot add any more variables and still be a clique
MRF on a line which favors smoothness

\[
Pr(y_{1...5}) = \frac{1}{Z} \phi_{12}(y_1, y_2) \phi_{23}(y_2, y_3) \phi_{34}(y_3, y_4) \phi_{45}(y_4, y_5)
\]

Consider the case where variables are binary, so functions return 4 different values depending on the combination of neighbours. Let's choose

\[
\phi_{nm}(0, 0) = 1.0 \quad \phi_{nm}(0, 1) = 0.1 \\
\phi_{nm}(1, 0) = 0.1 \quad \phi_{nm}(1, 1) = 1.0
\]

<table>
<thead>
<tr>
<th>(y_{1...5})</th>
<th>(Pr(y_{1...5}))</th>
<th>(y_{1...5})</th>
<th>(Pr(y_{1...5}))</th>
<th>(y_{1...5})</th>
<th>(Pr(y_{1...5}))</th>
<th>(y_{1...5})</th>
<th>(Pr(y_{1...5}))</th>
</tr>
</thead>
<tbody>
<tr>
<td>00000</td>
<td>0.09877</td>
<td>01000</td>
<td>0.02469</td>
<td>10000</td>
<td>0.04938</td>
<td>11000</td>
<td>0.04938</td>
</tr>
<tr>
<td>00001</td>
<td>0.04938</td>
<td>01001</td>
<td>0.01235</td>
<td>10001</td>
<td>0.02469</td>
<td>11001</td>
<td>0.02469</td>
</tr>
<tr>
<td>00010</td>
<td>0.02469</td>
<td>01010</td>
<td>0.00617</td>
<td>10010</td>
<td>0.01235</td>
<td>11010</td>
<td>0.01235</td>
</tr>
<tr>
<td>00011</td>
<td>0.04938</td>
<td>01011</td>
<td>0.01235</td>
<td>10011</td>
<td>0.02469</td>
<td>11011</td>
<td>0.02469</td>
</tr>
<tr>
<td>00100</td>
<td>0.02469</td>
<td>01100</td>
<td>0.02469</td>
<td>10100</td>
<td>0.01235</td>
<td>11100</td>
<td>0.04938</td>
</tr>
<tr>
<td>00101</td>
<td>0.01235</td>
<td>01101</td>
<td>0.01235</td>
<td>10101</td>
<td>0.00617</td>
<td>11101</td>
<td>0.02469</td>
</tr>
<tr>
<td>00110</td>
<td>0.02469</td>
<td>01110</td>
<td>0.02469</td>
<td>10110</td>
<td>0.01235</td>
<td>11110</td>
<td>0.04938</td>
</tr>
<tr>
<td>00111</td>
<td>0.04938</td>
<td>01111</td>
<td>0.04938</td>
<td>10111</td>
<td>0.02469</td>
<td>11111</td>
<td>0.09877</td>
</tr>
</tbody>
</table>
Denoising with MRFs

Original image, \( y \)

Observed image, \( x \)

MRF Prior (pairwise cliques)

\[
Pr(y) = \frac{1}{Z} \exp \left[ - \sum_{c \in C} \Psi_c(y) \right]
\]

Likelihoods

\[
Pr(x_n | y_n = 0) = \text{Bern}_{x_n}[\rho]
\]
\[
Pr(x_n | y_n = 1) = \text{Bern}_{x_n}[1 - \rho]
\]

Inference via Bayes’ rule:

\[
Pr(y_1 ... N | x_1 ... N) = \frac{\prod_{n=1}^{N} Pr(x_n | y_n) Pr(y_1 ... N)}{Pr(x_1 ... N)}
\]
MAP Inference

\[
\hat{y}_{1...N} = \arg \max_{y_{1...N}} Pr(y_{1...N} | x_{1...N})
\]

\[
= \arg \max_{y_{1...N}} \prod_{n=1}^{N} Pr(x_n | y_n) Pr(y_{1...N})
\]

\[
= \arg \max_{y_{1...N}} \sum_{n=1}^{N} \log[Pr(x_n | y_n)] + \log[Pr(y_{1...N})]
\]

\[
= \arg \min_{y_{1...N}} \sum_{n=1}^{N} U_n(y_n) + \sum_{(m,n) \in \mathcal{C}} P_{m,n}(y_m, y_n)
\]

**Unary terms**
(compatibility of data with label y)

**Pairwise terms**
(compatibility of neighboring labels)
Graph Cuts Overview

Graph cuts used to optimise this cost function:

\[
\arg \min_{y_1 \ldots N} \sum_{n=1}^{N} U_n(y_n) + \sum_{(m,n) \in C} P_{m,n}(y_m, y_n)
\]

Unary terms
(compatibility of data with label \(y\))

Pairwise terms
(compatibility of neighboring labels)

Three main cases:

- binary MRFs (i.e. \(y_i \in \{0, 1\}\)) where the costs for different combinations of adjacent labels are “submodular”. Exact MAP inference is tractable here.

- multi-label MRFs (i.e. \(y_i \in \{1, 2 \ldots, K\}\)) where the costs are “submodular”. Once more, exact MAP inference is possible.

- multi-label MRFs where the costs are more general. Exact MAP inference is intractable, but good approximate solutions can be found in some cases.
Graph Cuts Overview

Graph cuts used to optimise this cost function:

$$\arg \min_{y_1...N} \sum_{n=1}^{N} U_n(y_n) + \sum_{(m,n) \in C} P_{m,n}(y_m, y_n)$$

**Unary terms**
(compatibility of data with label $y$)

**Pairwise terms**
(compatibility of neighboring labels)

Approach:

Convert minimization into the form of a standard CS problem,

MAXIMUM FLOW or MINIMUM CUT ON A GRAPH

Low order polynomial methods for solving this problem are known
Minimum Cut Problem

Flow network.
- Abstraction for material flowing through the edges.
- \( G = (V, E) \) = directed graph, no parallel edges.
- Two distinguished nodes: \( s = \text{source}, t = \text{sink} \).
- \( c(e) \) = capacity of edge \( e \).
Def. An \textit{s-t cut} is a partition \((A, B)\) of \(V\) with \(s \in A\) and \(t \in B\).

Def. The \textbf{capacity} of a cut \((A, B)\) is: \[ \text{cap}(A, B) = \sum_{e \text{ out of } A} c(e) \]

\[
\begin{array}{c}
\text{Def.}\ \text{An s-t cut is a partition (A, B) of V with s } \in A \text{ and t } \in B. \\
\text{Def. The capacity of a cut (A, B) is: } \text{cap}(A, B) = \sum_{e \text{ out of } A} c(e) \\
\end{array}
\]

\[
\text{Capacity} = 10 + 5 + 15 = 30
\]
**Cuts**

**Def.** An **s-t cut** is a partition \((A, B)\) of \(V\) with \(s \in A\) and \(t \in B\).

**Def.** The **capacity** of a cut \((A, B)\) is: 

\[
\text{cap}(A, B) = \sum_{e \text{ out of } A} c(e)
\]

\[
\text{Capacity} = 9 + 15 + 8 + 30 = 62
\]
Min $s$-$t$ cut problem. Find an $s$-$t$ cut of minimum capacity.
Flows

Def. **An s-t flow** is a function that satisfies:
- For each \( e \in E \): \( 0 \leq f(e) \leq c(e) \) (capacity)
- For each \( v \in V - \{s, t\} \): \( \sum_{e \text{ in to } v} f(e) = \sum_{e \text{ out of } v} f(e) \) (conservation)

Def. **The value** of a flow \( f \) is: \( v(f) = \sum_{e \text{ out of } s} f(e) \).
**Def.** An s-t flow is a function that satisfies:
- For each $e \in E$: $0 \leq f(e) \leq c(e)$ (capacity)
- For each $v \in V - \{s, t\}$: $\sum_{e \text{ in } v} f(e) = \sum_{e \text{ out of } v} f(e)$ (conservation)

**Def.** The value of a flow $f$ is: $\nu(f) = \sum_{e \text{ out of } s} f(e)$.

Flows

![Graph showing an s-t flow with specified capacities and flows. The flow from s to t is 11, and the total value is 24.]
Max flow problem. Find s-t flow of maximum value.

Max Flow Problem

Value = 28
Flow value lemma. Let $f$ be any flow, and let $(A, B)$ be any $s$-$t$ cut. Then, the net flow sent across the cut is equal to the amount leaving $s$.

$$\sum_{e \text{ out of } A} f(e) - \sum_{e \text{ in to } A} f(e) = v(f)$$
Flow value lemma. Let $f$ be any flow, and let $(A, B)$ be any $s$-$t$ cut. Then, the net flow sent across the cut is equal to the amount leaving $s$.

$$\sum_{e \text{ out of } A} f(e) - \sum_{e \text{ in to } A} f(e) = v(f)$$

Value = 6 + 0 + 8 - 1 + 11 = 24
Flow value lemma. Let $f$ be any flow, and let $(A, B)$ be any s-t cut. Then, the net flow sent across the cut is equal to the amount leaving $s$.

$$
\sum_{e \text{ out of } A} f(e) - \sum_{e \text{ in to } A} f(e) = v(f)
$$
Max-flow min-cut theorem. [Ford-Fulkerson 1956] The value of the max flow is equal to the value of the min cut.

There are low order polynomial time algorithms known for determining these, and associated software is available.
Graph Cuts: Binary MRF

Graph cuts used to optimise this cost function:

\[
\arg \min_{y_1 \ldots N} \sum_{n=1}^{N} U_n(y_n) + \sum_{(m,n) \in C} P_{m,n}(y_m, y_n)
\]

**Unary terms**
(compatibility of data with label \( y \))

**Pairwise terms**
(compatibility of neighboring labels)

First work with binary case (i.e. True label \( y \) is 0 or 1)

Constrain pairwise costs so that they are “zero-diagonal”

\[
\begin{align*}
P_{m,n}(0, 0) &= 0 \\
P_{m,n}(0, 1) &= \theta_{01} \\
P_{m,n}(1, 0) &= \theta_{10} \\
P_{m,n}(1, 1) &= 0,
\end{align*}
\]
Graph Construction

- One node per pixel (here a 3x3 image)
- Edge from source to every pixel node
- Edge from every pixel node to sink
- Reciprocal edges between neighbours

Note that in the minimum cut EITHER the edge connecting to the source will be cut, OR the edge connecting to the sink, but NOT BOTH (unnecessary).

Which determines whether we give that pixel label 1 or label 0.

Now a 1 to 1 mapping between possible labelling and possible minimum cuts
Graph Construction

Now add capacities so that minimum cut, minimizes our cost function

Unary costs $U(0)$, $U(1)$ attached to links to source and sink.

- Either one or the other is paid.

Pairwise costs between pixel nodes as shown.

- Why? Easiest to understand with some worked examples.
Example 1

Solution

0 0 0

Cost

\[ U_a(0) + U_b(0) + U_c(0) \]
Example 2

Solution

Cost

\[ U_a(1) + U_b(0) + U_c(0) + P_{ab}(1, 0) \]
Graph Cuts: Binary MRF

Graph cuts used to optimise this cost function:

$$\arg \min_{y_1 \ldots N} \sum_{n=1}^{N} U_n(y_n) + \sum_{(m,n) \in C} P_{m,n}(y_m, y_n)$$

- **Unary terms** (compatability of data with label $y$)
- **Pairwise terms** (compatability of neighboring labels)

Summary of approach

- Associate each possible solution with a minimum cut on a graph
- Set capacities on graph, so cost of cut matches the cost function
- This minimizes the cost function and finds the MAP solution
Image Denoising

Foreground Extraction

Stereo Disparity
The connection between graph cuts and MRF inference was first made in this paper.

Exact Maximum A Posteriori Estimation for Binary Images

By D. M. GREIG, B. T. PORTEOUS and A. H. SEHEULT†

University of Durham, UK

[Received June 1987. Final revision September 1988]

SUMMARY

In this paper, for a degraded two-colour or binary scene, we show how the image with maximum a posteriori (MAP) probability, the MAP estimate, can be evaluated exactly using efficient variants of the Ford–Fulkerson algorithm for finding the maximum flow in a certain capacitated network. Availability of exact estimates allows an assessment of the performance of simulated annealing and of MAP estimation itself in this restricted setting. Unfortunately, the simple network flow algorithm does not extend in any obvious way to multicolour scenes. However, the results of experiments on two-colour images suggest that, in general, simulated annealing, according to practicable ‘temperature’ schedules, can produce poor approximations to the MAP estimate to which it converges.