# Graphical Models for Graph Matching

Tibério S. Caetano,Terry Caelli,
Dante A. C. Barone

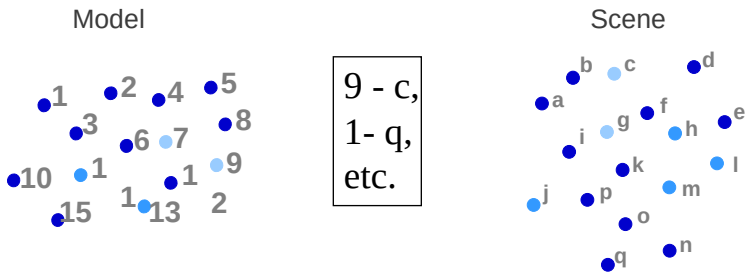
CVPR 2004

Presented by Robin Hewitt
for CSE 252C, Fall 2004

Paper is available online at
http://www.cs.ualberta.ca/~tcaetano/cvpr2004.pdf

## Problem Statement

Find the best node-to-node matching between two attributed graphs.



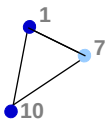

May have missing or extra nodes. Graphs may be rotated relative to one another. Both graphs are fully connected.

# Attributed-Graph Description, cont.

*Unary* attributes apply to single nodes.
Examples: line length, color, intensity.

*Binary* attributes apply to node pairs.
Example: angle between two line segments.

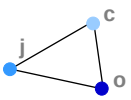
Model




$L_1$, $L_{10}$, $L_o$: 8px.
$L_7$, $L_c$: 4 px.

$\theta_{17}$: 12°.
$\theta_{oc}$: 68°.

Scene






# Authors' Claim

Their approach supports exact probabilistic inference.

In contrast, Probabilistic Relaxation Labeling is less formal. It's a heuristic method.

Their approach is more robust than PRL.

# HMRF Method(s)

Model the graph as a Hidden Markov Random Field.

**Method 1** (single-path dynamic programming):
traverse all graph nodes with a single chain.
Use dynamic programming to find an optimal match.

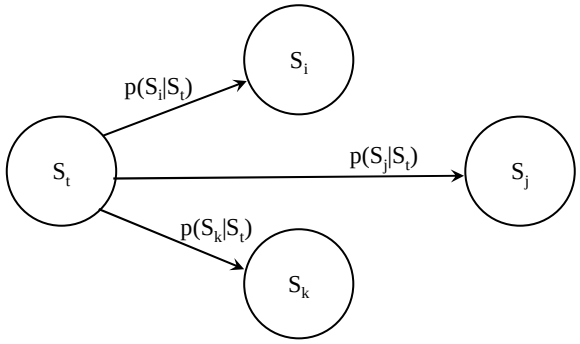
**Method 2** (Junction Trees):
Create Junction Trees with maximum cluster sizes of
2a) four (JT4)
2b) three (JT3).
Use Hugin algorithm to compute labeling (i.e., node-to-node matching) probabilities.



# Hidden Markov Random Field

**Markov chain:**

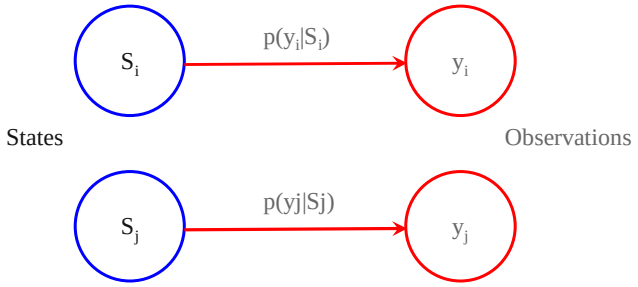
- A finite state machine with probabilities for each transition.
- $P(S_{t+1})$ depends only on $S_t$.



# Hidden Markov Random Field, cont.

**Hidden Markov Model:**

- A probability model that assumes a Markov chain.
- State values can't be observed directly. They're "hidden."
- Instead we have observations and we infer an underlying state sequence. Adds uncertainty to the state estimates.






# Hidden Markov Random Field, cont.

**HMRF:**

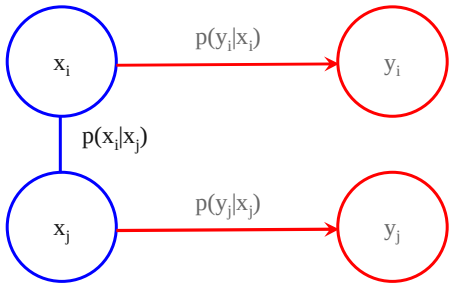
- In a Markov chain, new state probability depends only on a single, previous state. It's one dimensional.
- The field generalization allows new state probability to depend on influences from a surrounding neighborhood.

# Graphs as HMRF

The "real" graph, G, is unknown.

Nodes in G are the hidden states, $x_i$.

Node attributes, $y_i$, are the observations.






# Graphs as HMRF

Potentials for unary observations:

$$\phi(x_i) = p(y^i_x \mid x_i) = [S(y^i_x, y^1_s), \ldots S(y^i_x, y^S_s)]^t$$

Potentials for binary observations:

$$\psi(x_i, x_j) = p(x_j \mid x_i) = \begin{bmatrix} S(y^{ij}_x, y^{11}_s) & \cdots & S(y^{ij}_x, y^{1S}_s) \\ \vdots & \ddots & \vdots \\ S(y^{ij}_x, y^{S1}_s) & \cdots & S(y^{ij}_x, y^{SS}_s) \end{bmatrix}$$

Where $S(a,b) = N_a(b,cov)$.

# Single Path Dynamic Programming

Use a single Markov chain.

Create a path that traverses each vertex once.

Compute most likely label at each node based on
- Unary features of current node.
- Binary features of current node and consecutive nodes.

Optimization uses dynamic programming.

Look at one node, then two, etc. First optimize the unary, then the binary potentials.



# SPDP, cont.

Define the function, $\delta$ as

$$\delta_{\mathrm{i}}(a) = \max_{\mathrm{x}_1,\ldots\mathrm{x}_{\mathrm{i-1}}} \mathrm{P}(\mathrm{x}_1,\ldots\mathrm{x}_{\mathrm{i-1}}, \mathrm{x}_{\mathrm{i}} = \mathrm{s}_a)$$

$$\delta_{\mathrm{i+1}}(b) = \max_{a} [\delta_i(a)\psi(\mathrm{x}_{\mathrm{i}} = \mathrm{s}_a, \mathrm{x}_{\mathrm{i+1}} = \mathrm{s}_b)] \times \phi(\mathrm{x}_{\mathrm{i+1}} = \mathrm{s}_b)$$

where $\phi$ = the unary potentials and $\psi$ = the binary potentials.

# SPDP, cont.

Algorithm ($\xi$ tracks the maximum for each step):

Initialization

For $1 \leq a \leq N$,

$\delta_1(a) = \phi(x_1 = s_a)$

$\xi_1(a) = 0.$

Recursion

For $2 \leq i \leq M$, $1 \leq b \leq N$,

$\delta_i(b) = \max_a [\delta_{i-1}(a)\psi(x_i = s_a, x_{i+1} = s_b)] \times \phi(x_{i+1} = s_b)$

$\xi_i(b) = \arg\max_a [\delta_{i-1}(a)\psi(x_i = s_a, x_{i+1} = s_b)]$

Where M = number of model nodes, N = number of scene nodes.



# Junction Trees

Used for inference in belief networks.

Form node clusters and optimize within each cluster.

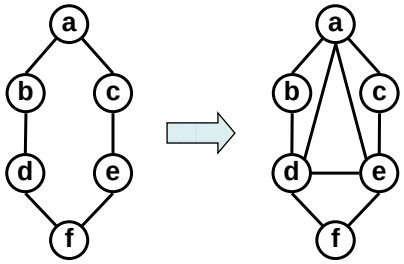
Alternately optimize within and between clusters.

# Junction Trees, cont.

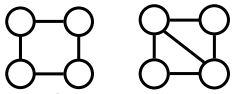
Forming clusters:

1. Triangulation – add edges.

Graphics based on ppt by
Y. Hamo and M. Silberstein.




No cycle of length > 3 that does not contain a chord.



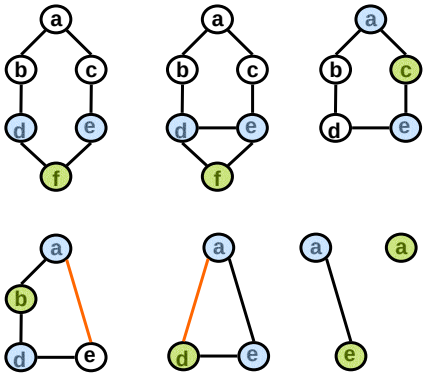



# Junction Trees, cont.

Forming clusters:

2. Create an elimination ordering.

Ordering eliminates the vertex with fewest added edges at each step.

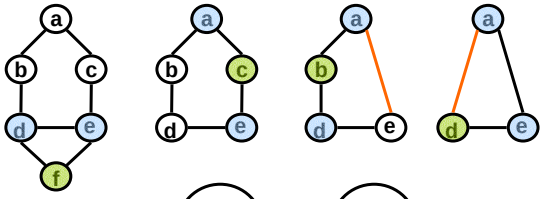


# Junction Trees, cont.

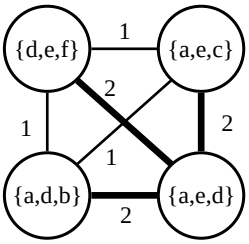
Linking clusters:

Form complete cluster graph and find max-weight spanning tree.

Elimination Order is f, c, b, d.



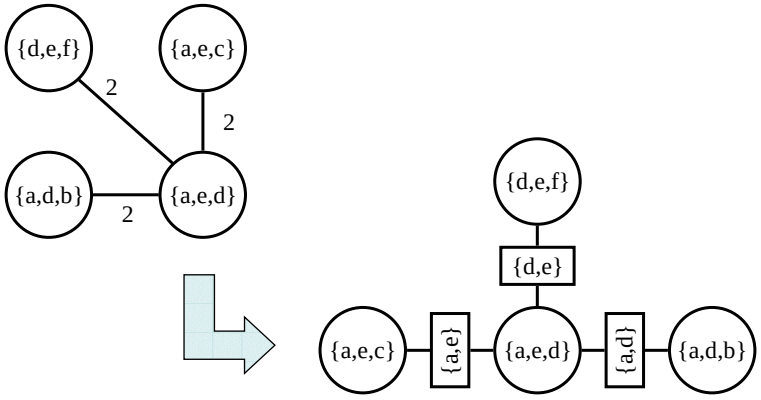

Weight of edge linking nodes U and V is U∩V. Example: {d,e,f}∩{a,e,f} = 2.






# Junction Trees, cont.

Linking clusters, cont.



# Hugin JT Algorithm

Give each cluster, C, and each separator, S, a potential function over its variables. Initialize:

$\varphi_C(\mathbf{x}) = \psi_C(\mathbf{x})$

$\varphi_S(\mathbf{x}) = 1$.

To pass a message from cluster B to cluster C over separator S, update

$$\phi_S^* = \max_{B \setminus S} \psi_B$$

$$\psi_C^* = \frac{\phi_S^*}{\phi_S} \psi_C$$



# JT Algorithm, cont.

Message passing protocol: Cluster B is allowed to send a message to neighbor C only after it has received messages from all neighbors except C.

Designate a root cluster and make the Junction Tree a directed tree.

After all messages have been passed, the potential in each cluster and separator is equal to the maximium a posteriori probability for the nodes in that cluster.

To get node potentials, use maximum probability from singleton separators.

# Results and Complexity

| Rank | Method | Complexity |
|---|---|---|
| 1 | JT4 | $O(N^4M)$ |
| 2 | JT3 | $O(N^3M)$ |
| 3 | SPDP | $O(N^2M)$ |
| 4 | PRL | $O(N^3M^2)$ |



# References and Resources

Tibério S. Caetano,Terry Caelli, Dante A. C. Barone. *Graphical Models for Graph Matching*. CVPR 2004
http://www.cs.ualberta.ca/~tcaetano/cvpr2004.pdf

**HMRF:**

F. Forbes and N. Peyrard. *Hidden Markov Random Field Model Selection Criteria Based on Mean Field-Like Approximations*. IEEE Trans. PAMI 25(9):1089-1101, September 2003.

# References and Resources

**Junction Trees:**

Mark A. Paskin. *A Short Course on Graphical Models*.
http://www.stanford.edu/~paskin/gm-short-course

Y. Hamo and M. Silberstein. *From Variable Elimination to Junction Trees*.
www.cs.technion.ac.il/~fmaayan/cs236633/jt.ppt



Graphics based on Y. Hamo and M. Silberstein.

## References and Resources

**PRL:**

W. J. Christmas, J. Kittler, and M. Petrou. *Structural matching in computer vision using probabilistic relaxation*. IEEE Trans. PAMI, 17(8):749–764, 1994.



# Extra Slides

# Probabilistic Relaxation Labeling

Define a compatibility measure $C_{ij}(L_m,L_n)$ between each pair of model nodes:

$$C_{ij}(L_m, L_n) = \frac{[p_i(L_m) - \bar{p}(L_m)] \cdot [p_j(L_n) - \bar{p}(L_n)]}{[p\ max(L_m) - \bar{p}(L_m)] \cdot [p\ max(L_n) - \bar{p}(L_n)]}$$

where

$L_m$ and $L_n$ are the labels assigned nodes i and j

$p_i(L)$ is the probability associated with label L for node i

$$\bar{p}(L) = (1/N)\sum_{1}^{N} p_i(L)$$

$p\ max(L) = Max\{p_i(L)\}$, $i = 1,\ldots N$

N = number of model nodes.



# Probabilistic Relaxation Labeling

The $p_i(L)$ values are based on compatibilities between both unary and binary features.

Relaxation is iterative. Start with an initial guess for the probability of each label assignment. Each iteration, update that probability based on neighboring assignments.

Stop when probabilities are close to 1's and 0's, or the amount of improvement is small, or after a maximum number of iterations.