A Tutorial Introduction to Belief Propagation

James Coughlan

THE SMITH-KETTLEWELL
EYE RESEARCH INSTITUTE

August 2009

Table of Contents

Introduction

This tutorial introduces belief propagation in the context of factor graphs and demonstrates its use in a simple model of stereo matching used in computer vision.

It assumes knowledge of probability and some familiarity with MRFs (Markov random fields), but no familiarity with factor graphs is assumed.

Based on a tutorial presented at Sixth Canadian Conference on Computer and Robot Vision (CRV 2009). Kelowna, British Columbia. May 2009.

Feedback is welcome: please send it to coughlan@ski.org

Updated versions will be available at
http://www.ski.org/Rehab/Coughlan_lab/General/TutorialsandReference.html

What is belief propagation (BP)?

Technique invented in 1982 [Pearl] to calculate marginals in Bayes nets.

Also works with MRFs, graphical models, factor graphs.

Exact in some cases, but approximate for most problems.

Can be used to estimate marginals, or to estimate most likely states (e.g. MAP).

MRFs, graphical models, factor graphs

Common property: joint probability of many variables factors into little pieces.

$$P(x_1, x_2, \ldots, x_N) = \frac{1}{Z} f(x_1, x_2) g(x_2) h(x_2, x_3, x_4)$$

Probability domain

$$P(x_1, x_2, \ldots, x_N) = \frac{1}{Z} e^{F(x_1,x_2)+G(x_2)+H(x_2,x_3,x_4)}$$

Energy (log prob.) domain

The factors (*f, g, F, G,* etc.) are called **potentials.**

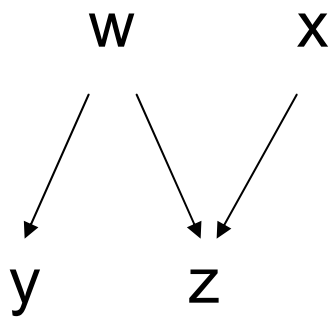
Factors

In general factors are not probabilities themselves – they are functions that *determine* all probabilities.

However, in special cases (Markov chain, Bayes net) they can be interpreted as conditional probabilities.

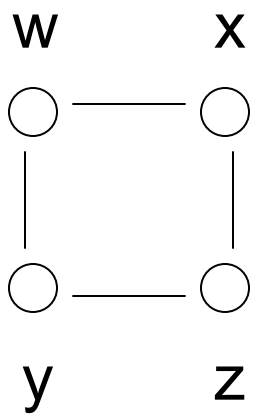
Non-negative (except in log domain), but don’t need to normalize to 1.

Bayes net example



$$P(w, x, y, z) = P(w)P(x)P(y|w)P(z|w, x)$$

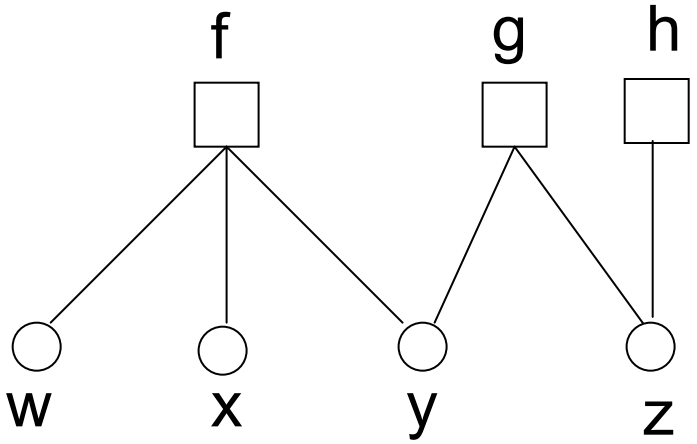
MRF example



$$P(w, x, y, z) = \frac{1}{Z} f_{wx}(w, x) f_{xz}(x, z) f_{yz}(y, z) f_{wy}(w, y)$$

Factor graph example

Each box denotes a factor (interaction) among the variables it connects to:



$$P(w,x,y,z) = \frac{1}{Z} f(w,x,y) g(y,z) h(z)$$

Marginals vs. maximizer

Marginals: find $P(x_1), P(x_2), \ldots, P(x_N)$

Maximizer: find $\arg \max_{x_1, x_2, \ldots x_N} P(x_1, x_2, \ldots, x_N)$

Both are computationally difficult: if state space of all x_i has S possible states, then $O(S^N)$ calculations required (exhaustive addition/ exhaustive search)!

One solution: BP

BP provides *exact* solution when there are no loops in graph! (E.g. chain, tree.)

Equivalent to dynamic programming/ Viterbi in these cases.

Otherwise, “loopy” BP provides *approximate* (but often good) solution.

Alternatives: graph cuts, MCMC/ simulated annealing, etc.

Overview of BP

Overview: iterative process in which neighboring variables "talk" to each other, passing **messages** such as:

"I (variable x_3) think that you (variable x_2) belong in these states with various likelihoods…"

Overview of BP, con't

After enough iterations, this series of conversations is likely to converge to a consensus that determines the marginal probabilities of all the variables.

Estimated marginal probabilities are called **beliefs.**

BP algorithm: update messages until convergence, then calculate beliefs.

Common case: pairwise MRF

Pairwise MRF (graphical model) – has just unary and pairwise factors:

$$P(x_1, x_2, \ldots, x_N) = \frac{1}{Z} \prod_{i=1}^{N} g_i(x_i) \prod_{<ij>} f_{ij}(x_i, x_j)$$

Here $g_i(.)$ are the unary factors,

$f_{ij}(.,.)$ are the pairwise factors,

and the second product is over neighboring pairs of nodes (variables), such that $i<j$ (i.e. don't count a pair of variables twice).

General case: factor graph

BP also formulated for factor graphs, which may have interactions higher-order than pairwise. See [Kschischang et al] for details.

This tutorial will only describe BP for pairwise MRF.

Messages

Message from node *i* to node *j*: $\boldsymbol{m_{ij}(x_j)}$

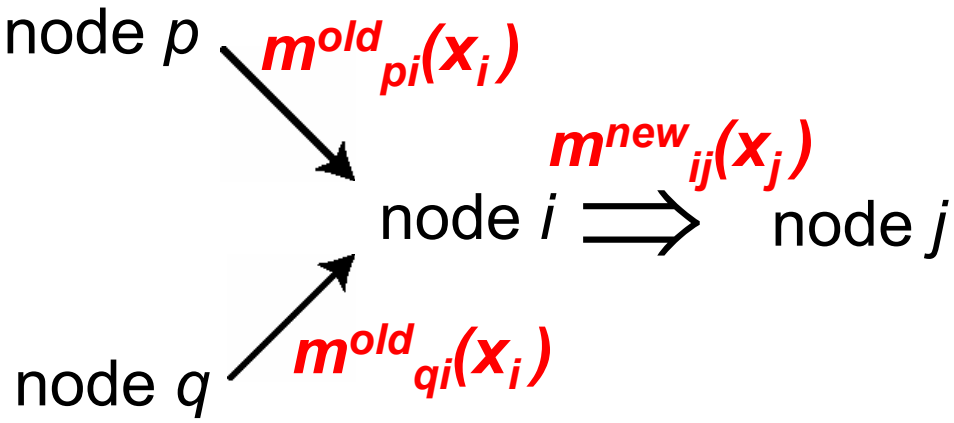
Messages are similar to likelihoods: non-negative, don't have to sum to 1.

A high value of $m_{ij}(x_j)$ means that node *i* "believes" the marginal value $P(x_j)$ to be high.

Usually initialize all messages to 1 (uniform), or random positive values.

Message update

To update message from *i* to *j*, consider all messages flowing into *i* (except for message from *j*):



Message update

The messiest equation in this tutorial:

$$m_{ij}^{new}(x_j) = \sum_{x_i} f_{ij}(x_i, x_j) \underbrace{g_i(x_i) \prod_{k \in Nbd(i) \setminus j} m_{ki}^{old}(x_i)}_{h(x_i)}$$

Messages (and unary) factor on RHS multiply like independent likelihoods

→ update equation has this form:

$$m_{ij}^{new}(x_j) = \sum_{x_i} f_{ij}(x_i, x_j) h(x_i)$$

Message update

Note: given a pair of neighboring nodes, there is only one pairwise interaction but messages flow in *both* directions. Define pairwise potential so that we can use the message update equation in both directions (from *i* to *j* and from *j* to *i*) without problems:

$$f_{ij}(x_i, x_j) = f_{ji}(x_j, x_i)$$

By the way, this isn't the same as assuming a symmetric potential, i.e.

$$f_{ij}(x_i, x_j) = f_{ij}(x_j, x_i)$$

Message normalization

In practice one usually normalizes the messages to sum to 1, so that

$$\sum_{x_j} m_{ij}(x_j) = 1$$

Useful for numerical stability (otherwise overflow/ underflow likely to occur after enough message updates)

Update schedule

Synchronous: update all messages in parallel

Asynchronous: update one message at a time

With luck, messages will converge after enough updates.

Which schedule to choose?

For a chain, asynchronous is most efficient for a serial computer (up and down chain once guarantees convergence). Similar procedure for a tree.

For a grid (e.g. stereo on pixel lattice), people often sweep in an “up-down-left-right fashion” [Tappen & Freeman].

Choosing a good schedule requires some experimentation

Belief read-out

Once messages have converged, use belief read-out equation:

$$b_i(x_i) \propto g_i(x_i) \prod_{k \in Nbd(i)} m_{ki}(x_i)$$

If you normalize belief then it approximates the marginal probability. (Approximation exact when no loops.)

Note: another belief equation available for pairwise beliefs, i.e. estimates of pairwise marginal distributions. See [Bishop 2006].

Computational complexity

The main cost is the message update equation, which is $O(S^2)$ for each pair of variables.

Much better than brute-force complexity $O(S^N)$.

Sum-product vs. max-product

The standard BP we just described is called **sum-product** (from message update equation), and is used to estimate marginals.

A simple variant, called **max-product** (or **max-sum** in log domain), is used to estimate the state configuration with *maximum probability*.

Max-product

Message update same as before, except that sum is replaced by max:

$$m_{ij}^{new}(x_j) = \max_{x_i} f_{ij}(x_i, x_j) g_i(x_i) \prod_{k \in Nbd(i) \setminus j} m_{ki}^{old}(x_i)$$

Belief equation same as before, but beliefs no longer estimate marginals. Instead, they are scoring functions whose maxima point to **most likely states**.

Max-sum

This is what max-product becomes in log domain: products become sums, and messages can be positive or negative.

Note: in practice, beliefs are often "normalized" to avoid underflow/ overflow, e.g. by uniformly shifting them so that lowest belief value is 0.

Example: MRF stereo

Let *r* or *s* denote 2D image coordinates *(x,y)*.

Unknown disparity field *D(x,y)=D(r)*.

Smoothness prior: $P(D) = \frac{1}{Z} e^{-\beta V(D)}$

where $V(D) = \sum_{<rs>} |D_r - D_s|$ sums over all neighboring pixels.

Often the penalty $\min(|D_r - D_s|, \tau)$ is used instead of $|D_r - D_s|$ for greater robustness.

Likelihood function

Let *m* denote the matching error across the entire left and right images, i.e.

$$\mathrm{m}_r(D_r) = |L(x + D_r, y) - R(x, y)|$$

Simple likelihood function:

$$P(m_r(D_r)|D_r) = \frac{1}{Z'} e^{-\mu m_r(D_r)}$$

Assume conditional independence across image:

$$P(m|D) = \prod_r P(m_r(D_r)|D_r)$$

Posterior and inference

Posterior: $P(D \mid m) = P(D)\, P(m|D) / P(m)$

Posterior has unary and pairwise factors.

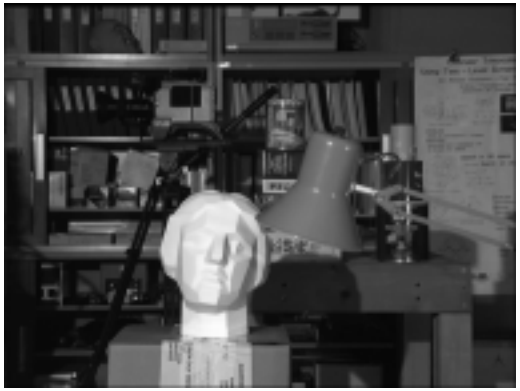
Sum-product BP → estimates marginals $P(D_r \mid m)$ at each pixel r.

Sample results

Tsukuba images from Middlebury stereo database (http://vision.middlebury.edu/stereo/)

Left

Right



Original images in color, but our simple model uses grayscale versions

Sample results

Message update schedule: “left-right-up-down”

“Left” means an entire sweep that updates messages from all pixels to their left neighbors, etc.

One iteration consists of a sweep left, then right, then up, then down.

Sample results

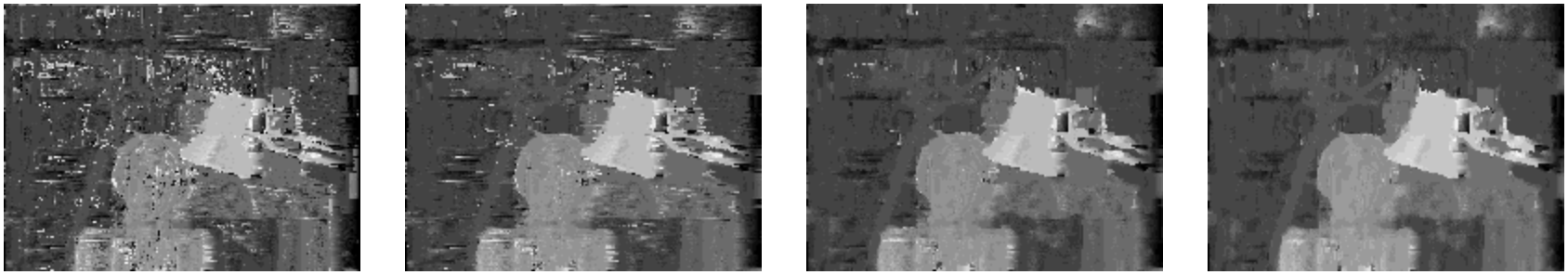
Winning disparities shown by grayscale levels (lighter pixels have higher estimated disparity)

Before BP (i.e. disparities estimated solely by unary potentials):

Sample results

First iteration: disparities shown after

left, right, up, down sweeps

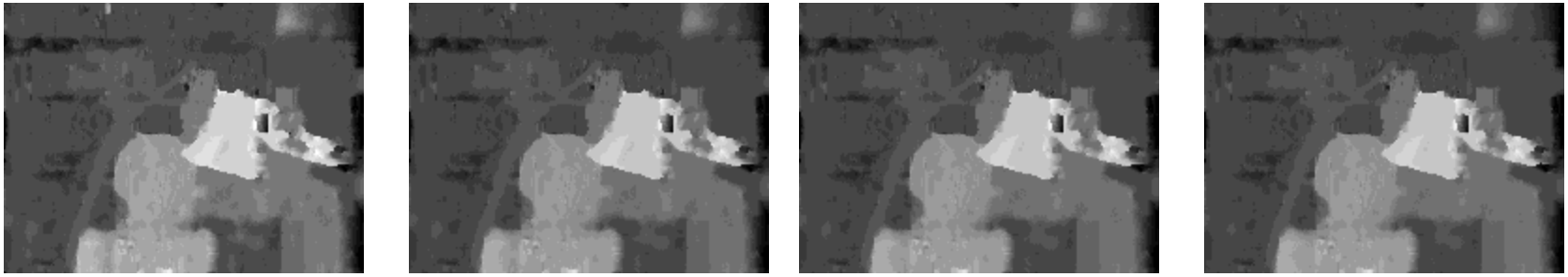


Noticeable “streaking” after left and right sweeps, mostly erased by up sweep.

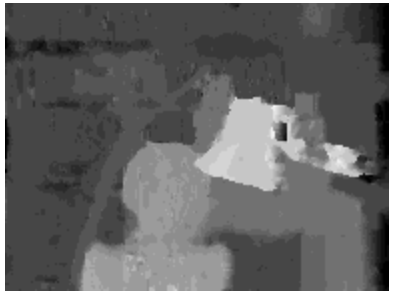
Sample results

Subsequent iterations:

2 3 4 5



… 20



Note:

Little change after first few iterations.

Model can be improved to give better results

-- this is just a simple example to illustrate BP.

Complications and "gotchas"

1.) Ties: suppose there are two state configurations that are equally probable. BP beliefs will show ties between the two solutions; how to recover both globally consistent solutions?

Solution: back-tracking [Bishop]

2.) If messages oscillate instead of converge → damp them with "momentum" [Murphy et al]

Speed-ups

Binary variables: use log ratios [Mackay]

Distance transform and multi-scale [Felzenszwalb & Huttenlocher]

Sparse forward-backward [Pal et al]

Dynamic quantization of state space [Coughlan & Shen]

Higher-order factors with linear interactions [Potetz and Lee]

GPU [Brunton et al]

Extensions/variations

Factor graph BP: higher-order factors (cliques) [Kschiang et al]

Particles for continuous variables: non-parametric BP [Sudderth et al]

Top m solutions [Yanover & Weiss]

Tree reweighted BP [Wainwright et al]

Connections

Variational formulation/ connection with statistical physics: Bethe free energy, Kikuchi, comparison with mean field [Yedidia]

Model of neurodynamics [Ott & Stoop]

Advantages

Extremely general: you can apply BP to any graphical model with any form of potentials – even higher-order than pairwise!

Useful for marginals or maximum probability solution

Exact when there are no loops

Easy to program

Easy to parallelize

Disadvantages

Other methods may be more accurate, faster and/or less memory-intensive in some domains [Szeliski et al].

For instance, graph cuts are faster than BP for stereo, and give slightly better results.

Perspective

But [Meltzer et al]: better to improve model than to improve the optimization technique:

> "As can be seen, the global minimum of the energy function does not solve many of the problems in the BP or Graph Cuts solutions. This suggests that the problem is not in the optimization algorithm but rather in the energy function. A promising problem for future research is to learn better energy functions from ground truth data."

Thanks to

Dr. Ender Tekin and Dr. Huiying Shen

View of Lake Okanagan (about 10 miles from Kelowna)

References

Introductory material:

C. M. Bishop. **Pattern Recognition and Machine Learning**. Springer. 2006. http://research.microsoft.com/en-us/um/people/cmbishop/PRML/index.htm

D. Mackay. **Information Theory, Inference, and Learning Algorithms**. Cambridge University Press. 2003.

J. Pearl. “Reverend Bayes on inference engines: A distributed hierarchical approach.” AAAI-82: Pittsburgh, PA. Second National Conference on Artificial Intelligence. Menlo Park, California: AAAI Press. pp. 133–136. 1982.

Performance studies:

T. Meltzer, C. Yanover and Y. Weiss. “Globally Optimal Solutions for Energy Minimization in Stereo Vision using Reweighted Belief Propagation.” ICCV 2005.

K.P. Murphy,Y. Weiss and M.I. Jordan. “Loopy belief propagation for approximate inference: an empirical study.” Uncertainty in AI. 1999.

R. Szeliski, R. Zabih, D. Scharstein, O. Veksler, V. Kolmogorov, A. Agarwala, M. Tappen, and C. Rother. “A comparative study of energy minimization methods for Markov random fields with smoothness-based priors.” IEEE Transactions on Pattern Analysis and Machine Intelligence, 30(6):1068-1080. June 2008.

M. F. Tappen and W. T. Freeman. “Comparison of Graph Cuts with Belief Propagation for Stereo, using Identical MRF Parameters.” International Conference on Computer Vision (ICCV). 2003.

Factor graphs:

F.R. Kschischang, B.J. Frey and H.-A. Loeliger. “Factor graphs and the sum-product algorithm.” IEEETIT: IEEE Transactions on Information Theory 47. 2001.

References, con't

Extensions:

E. Sudderth, A. Ihler, W. Freeman, and A. Willsky. "Nonparametric Belief Propagation." Conference on Computer Vision & Pattern Recognition (CVPR). June 2003.

M. Wainwright, T. Jaakkola, and A. Willsky, "Map Estimation via Agreement on Trees: Message-Passing and Linear Programming." IEEE Trans. Information Theory, vol. 51, no. 11, pp. 3697-3717. 2005.

C. Yanover and Y. Weiss. "Finding the M Most Probable Configurations using Loopy Belief Propagation." NIPS 2003.

Speed-ups:

B. Potetz and T.S. Lee. "Efficient belief propagation for higher-order cliques using linear constraint nodes." Comput. Vis. Image Understanding, 112(1):39-54. 2008.

A. Brunton, C. Shu, and G. Roth. "Belief propagation on the gpu for stereo vision." 3rd Canadian Conference on Computer and Robot Vision. 2006.

J. Coughlan and H. Shen. "Dynamic Quantization for Belief Propagation in Sparse Spaces." Computer Vision and Image Understanding (CVIU) Special issue on Generative-Model Based Vision. Volume 106, Issue 1, pp. 47-58. April 2007.

P. F. Felzenszwalb and D. P. Huttenlocher. "Efficient belief propagation for early vision." Int. J. Comput. Vision, 70(1):41–54. 2006.

C. Pal, C. Sutton, and A. McCallum. "Sparse forwardbackward using minimum divergence beams for fast training of conditional random fields." In Proc. Intl. Conf. on Acoustics, Speech, and Signal Processing, volume 5, pages 581–584, 2006.

Variational formulation:

J. Yedidia. "Characterization of belief propagation and its generalizations." TR2001-015. 2001.

J. Yedidia. "Bethe free energy, Kikuchi approximations, and belief propagation algorithms." TR2001-016. 2001.

Neuroscience:

T. Ott and R. Stoop. "The neurodynamics of belief propagation on binary markov random fields." NIPS 2006.