

Convexity: What is it good for?

Yair Weiss

Hebrew University of Jerusalem

Collaborators: Danny Rosenberg, Elad Mezuman, Talya Meltzer

Outline

- Belief Propagation in Computer Vision Applications.
- Convex vs. non-convex BP.
- What should we use?
- Some new theoretical results on ordinary BP

Stereo by Energy Minimization

Input



Output



$$E(x) = \sum_i E_i(x_i) + \sum_{\langle ij \rangle} E_{ij}(x_i, x_j)$$

data term $E_i(x_i)$ and smoothness term $E_{ij}(x_i, x_j)$ are non-quadratic. Optimization is NP Hard (Boykov et al. 04)

Learning Energy Functions for Category-Specific Segmentation



Training Set:



...

...

Novel Input:



(Borenstein and Ullman, 2002)

Energy Functions for Category-Specific Segmentation

Input

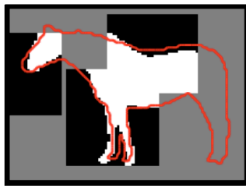


$$x^* = \arg \min_x \sum_i E_i(x_i) + \sum_{\langle ij \rangle} E_{ij}(x_i, x_j)$$

Pairwise term



Data Term



Energy Functions for Category-Specific Segmentation

Input



Output

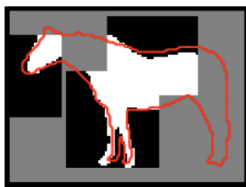


$$x^* = \arg \min_x \sum_i E_i(x_i) + \sum_{\langle ij \rangle} E_{ij}(x_i, x_j)$$

Pairwise term



Data Term

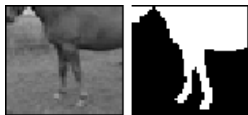


Constructing the Data Term

Input



Fragment



$$x^* = \arg \min_x \sum_i E_i(x_i) + \sum_{\langle ij \rangle} E_{ij}(x_i, x_j)$$

Search Space



Data Term

Constructing the Data Term

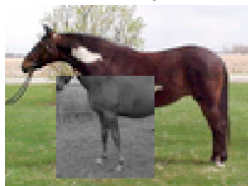
Input



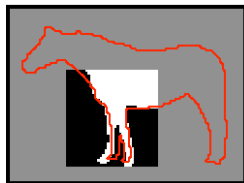
Fragment

$$x^* = \arg \min_x \sum_i E_i(x_i) + \sum_{\langle ij \rangle} E_{ij}(x_i, x_j)$$

Search Space



Data Term

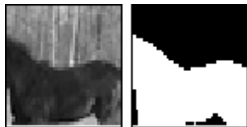


Constructing the Data Term

Input



Fragment

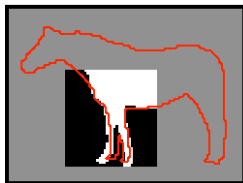


$$x^* = \arg \min_x \sum_i E_i(x_i) + \sum_{\langle ij \rangle} E_{ij}(x_i, x_j)$$

Search Space



Data Term



Constructing the Data Term

Input



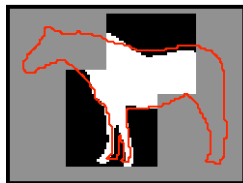
Fragment

$$x^* = \arg \min_x \sum_i E_i(x_i) + \sum_{\langle ij \rangle} E_{ij}(x_i, x_j)$$

Search Space



Data Term



Constructing the Data Term

Input



Fragment

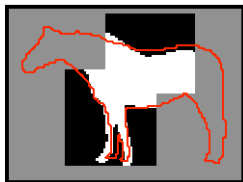


$$x^* = \arg \min_x \sum_i E_i(x_i) + \sum_{\langle ij \rangle} E_{ij}(x_i, x_j)$$

Search Space



Data Term



Constructing the Data Term

Input



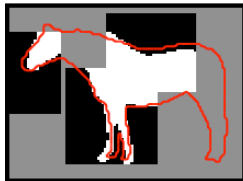
Fragment

$$x^* = \arg \min_x \sum_i E_i(x_i) + \sum_{\langle ij \rangle} E_{ij}(x_i, x_j)$$

Search Space



Data Term



Learning Formulation



...

...

Given training set and tens of thousands of fragments, choose a small number of fragments, thresholds and weights.

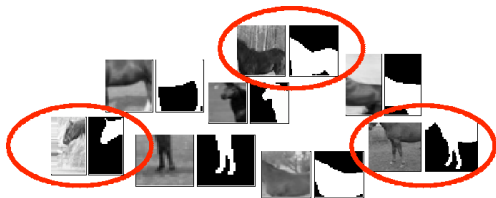
Learning Formulation



Given training set and tens of thousands of fragments, choose a small number of fragments, thresholds and weights.

Equivalent to Feature Induction in Conditional Random Fields (Lafferty et al. 97, Lafferty et al. 2001)

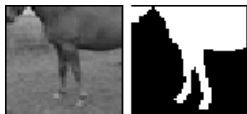
Feature Induction in CRFs



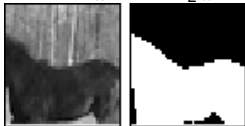
Training set:

$$E(x; I) = \sum_{\langle ij \rangle} w_{ij}(I) |x_i - x_j|$$

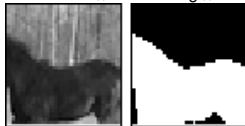
$$+ \lambda_1 \|x - x_{F_1}\|$$



$$+ \lambda_2 \|x - x_{F_2}\|$$



$$+ \lambda_3 \|x - x_{F_3}\|$$



$$\lambda^* = \arg \max \log P(x; \lambda) + \textit{sparsity}$$

Calculating Conditional Likelihood Exactly is Intractable

Iteratively add fragment with highest likelihood gain:

$$\begin{aligned}\Pr(x^*; E) &= \frac{1}{Z(E)} e^{-E(x^*)} \\ &= \frac{1}{\sum_x e^{-E(x)}} e^{-E(x^*)}\end{aligned}$$

- Give low energy to desired segmentations and high energy to all other segmentations.
- “all other” : exponentially many.
- Need to evaluate likelihood gain for tens of thousands of fragments.

Image Segmentation Using Normalized Cut



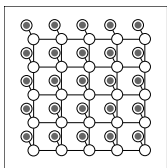
$$\lambda^* = \min_{A,B} \frac{\text{cut}(A, B)}{|A||B|}$$

Linearized problem:

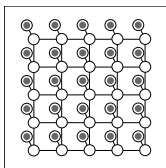
$$\min_{x \in \{0,1\}^n} \sum_{\langle ij \rangle} E_{ij}(x_i, x_j) + \lambda |x| |1 - x|$$

Inference in Graphical Models

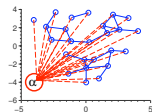
$$\Pr(x) = \frac{1}{Z} \prod_i \psi_i(x_i) \prod_{ij} \psi_{ij}(x_i, x_j)$$



MAP



log Z



MAP

Convex vx. non-convex BP

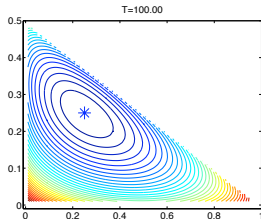
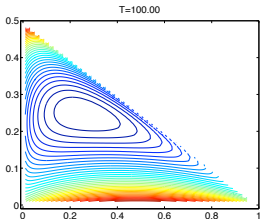
- Input: $\{\Psi_{ij} \propto e^{-E_{ij}}\}$
- Output: beliefs $\{b_{ij}\}, \{b_i\}$

Iterate:

$$m_{ij}(x_j) \leftarrow \sum_{x_i} \prod_{k \neq j} m_{ki}^{\rho_{ki}}(x_i) m_{ji}^{\rho_{ij}-1}(x_i) \Psi_{ij}(x_i, x_j)$$

- $\rho = 1$, standard BP.
- $\rho_{ij} < 1$ “Tree-Reweighted/Fractional/Convex” BP (MPLP, MSD, TRW)

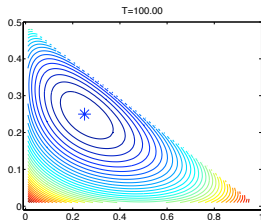
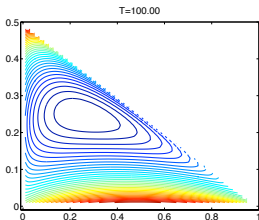
Why is it called “convex” BP?



$$H(\{q_{ij}, q_i\}) = \sum_{ij} c_{ij} H(q_{ij}) + \sum_i c_i q_i$$

Bethe approximation: $c_{ij} = 1, c_i = 1 - d_i$. Usually non-convex.

Why is it called “convex” BP?

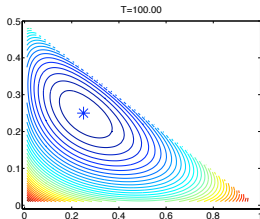
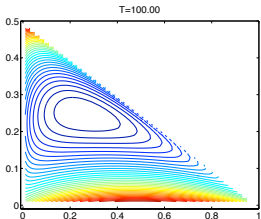


$$H(\{q_{ij}, q_i\}) = \sum_{ij} c_{ij} H(q_{ij}) + \sum_i c_i q_i$$

$$\rho_i = \frac{1}{c_i + \sum_{j \in N_i} c_{ij}}$$

$$\rho_{ij} = \rho_j c_{ij}$$

Why is it called “convex” BP?



$$H(\{q_{ij}, q_i\}) = \sum_{ij} c_{ij} H(q_{ij}) + \sum_i c_i q_i$$

$$\rho_i = \frac{1}{c_i + \sum_{j \in N_i} c_{ij}}$$

$$\rho_{ij} = \rho_j c_{ij}$$

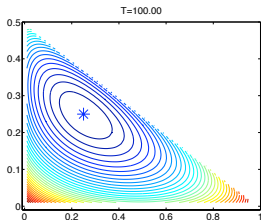
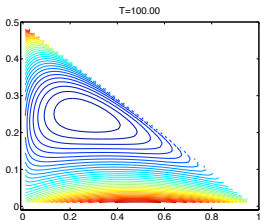
$$m_{ij}(x_j) \leftarrow \sum_{x_i} \prod_{k \neq j} m_{ki}^{\rho_{ki}}(x_i) m_{ji}^{\rho_{ij}-1}(x_i) \Psi_{ij}(x_i, x_j)$$

So what should we use?



For stereo we successfully used max-product convex BP and for segmentation sum-product convex BP. Mostly because of cleaner theory.

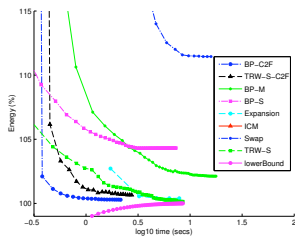
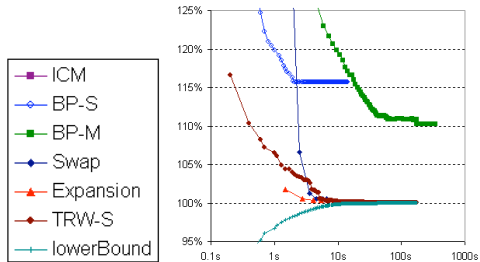
Theory of convex BP:



- no local minima (sum).
- bound on $\log Z$ (sum).
- connection to linear programming relaxation (max).
- bound on optimal assignment (max).
- certificate of optimality (max)

(Wainwright et al. 01, Vontobel and Koetter 06, Weiss et al. 07, Koller and Friedman 09)

Sometimes excellent results with convex BP



but sometimes not as good as BP...
(Szeliski et al. 08, Rosenberg and Weiss 11)

Other comparisons

Overall, the outcome of the experiments is that across all settings, AP obtains better results than either MPLP or DD, at a better run time. This is somewhat disappointing, as both MPLP and DD come with theoretical justification and convergence guarantees.

(Givoni et al. 11)

Despite these merits, in terms of quality of the approximation, convex free energies are still often not competitive with Bethe and in fact result in poorer performance over a wide range of parameter settings.

(Meshi et al. 08)

Q: What should we use?



A: if you really need a bound, use convex BP.

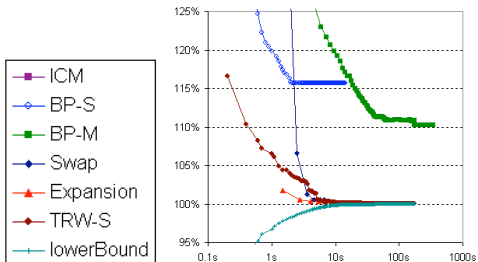


Image Segmentation Using Normalized Cut



$$\lambda^* = \min_{A,B} \frac{\text{cut}(A, B)}{|A||B|}$$

Linearized problem:

$$\min_{x \in \{0,1\}^n} \sum_{\langle ij \rangle} E_{ij}(x_i, x_j) + \lambda |x| |1 - x|$$

The λ question

$$E^* = \min_x \frac{f(x)}{g(x)} \text{ s.t. } g(x) > 0$$

$$\min_x f(x) - \lambda g(x) \text{ s.t. } g(x) > 0 \text{ ???}$$

< 0

$$E^* < \lambda$$

$= 0$

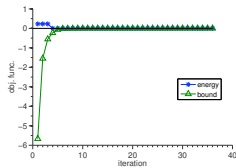
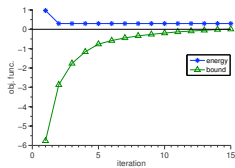
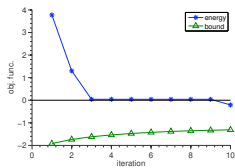
$$E^* = \lambda \text{ and}$$

$$\arg \min_x f(x) - \lambda g(x) = \arg \min_x \frac{f(x)}{g(x)}$$

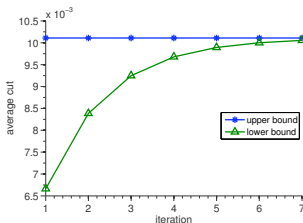
> 0

$$E^* > \lambda$$

Solving the λ question using convex BP



Solving average cut on a benchmark



For 92% of images we find the global optimum of average cut, up to tolerance 0.1.

Interim Summary

- Sometimes convex BP is better, sometimes worse.
- If you really need a bound, use convex BP.
- More theory needed

Bounds from ordinary BP

Using reparametrization property of BP (Wainwright et al. 01).
Let b_i, b_{ij} be the BP beliefs at any iteration:

$$\begin{aligned}\Pr(x) &= \frac{1}{Z} \prod_i \psi_i(x_i) \prod_{ij} \psi_{ij}(x_i, x_j) \\ &= \frac{1}{Z_2} \prod_i b_i(x_i) \prod_{ij} \frac{b_{ij}(x_i, x_j)}{b_i(x_i)b_j(x_j)}\end{aligned}$$

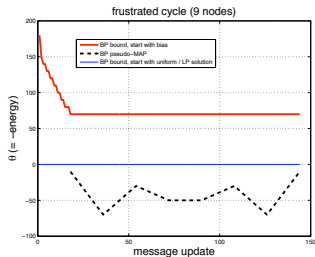
\Rightarrow

$$\max_x \Pr(x) \leq \frac{1}{Z_2} \prod_i \max_{x_i} b_i(x_i) \prod_{ij} \max_{x_i, x_j} \frac{b_{ij}(x_i, x_j)}{b_i(x_i)b_j(x_j)}$$

sometimes, we can show the bound is tight.

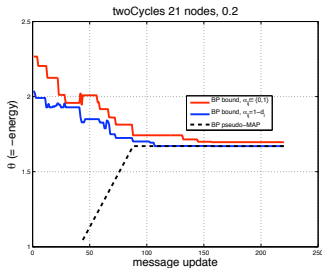
(Meltzer and Weiss, in preparation)

BP as coordinate descent on the bound



- When the graph contains at most one cycle, BP is coordinate descent on the bound.
- Even when BP oscillates, the bound converges.

Locally tree-like graphs



BP tends to improve the bound.

Discussion

- Many successful applications of both convex and ordinary BP.
- More theory needed.
- Bounds from ordinary BP.