Graphical models and message-passing
Part I: Basics and MAP computation

Martin Wainwright

UC Berkeley
Departments of Statistics, and EECS

Tutorial materials (slides, monograph, lecture notes) available at:
www.eecs.berkeley.edu/~wainwrig/kyoto12

September 2, 2012
Introduction

- graphical model: * graph $G = (V, E)$ with $N$ vertices
  * random vector: $(X_1, X_2, \ldots, X_N)$

(a) Markov chain (b) Multiscale quadtree (c) Two-dimensional grid

useful in many statistical and computational fields:
  - machine learning, artificial intelligence
  - computational biology, bioinformatics
  - statistical signal/image processing, spatial statistics
  - statistical physics
  - communication and information theory
clique $C$ is a fully connected subset of vertices
compatibility function $\psi_C$ defined on variables $x_C = \{x_s, s \in C\}$
factorization over all cliques

\[ p(x_1, \ldots, x_N) = \frac{1}{Z} \prod_{C \in \mathcal{C}} \psi_C(x_C). \]
Example: Optical digit/character recognition

- **Goal:** correctly label digits/characters based on “noisy” versions
- E.g., mail sorting; document scanning; handwriting recognition systems
Example: Optical digit/character recognition

Goal: correctly label digits/characters based on “noisy” versions.

- strong sequential dependencies captured by (hidden) Markov chain
- “message-passing” spreads information along chain

(Baum & Petrie, 1966; Viterbi, 1967, and many others)
Example: Image processing and denoising

- 8-bit digital image: matrix of intensity values \{0, 1, \ldots 255\}
- enormous redundancy in “typical” images (useful for denoising, compression, etc.)
Example: Image processing and denoising

- 8-bit digital image: matrix of intensity values \{0, 1, \ldots, 255\}
- enormous redundancy in “typical” images (useful for denoising, compression, etc.)
- multiscale tree used to represent coefficients of a multiscale transform (e.g., wavelets, Gabor filters etc.)

(e.g., Willsky, 2002)
Example: Depth estimation in computer vision

Stereo pairs: two images taken from horizontally-offset cameras
Modeling depth with a graphical model

Introduce variable at pixel location \((a, b)\):

\[ x_{ab} \equiv \text{Offset between images in position } (a, b) \]

Use message-passing algorithms to estimate most likely offset/depth map. (Szeliski et al., 2005)
Many other examples

- natural language processing (e.g., parsing, translation)

- computational biology (gene sequences, protein folding, phylogenetic reconstruction)

- social network analysis (e.g., politics, Facebook, terrorism.)

- communication theory and error-control decoding (e.g., turbo codes, LDPC codes)

- satisfiability problems (3-SAT, MAX-XORSAT, graph colouring)

- robotics (path planning, tracking, navigation)

- sensor network deployments (e.g., distributed detection, estimation, fault monitoring)

- ...
Core computational challenges

Given an undirected graphical model (Markov random field):

\[ p(x_1, x_2, \ldots, x_N) = \frac{1}{Z} \prod_{C \in C} \psi_C(x_C) \]

How to efficiently compute?

- most probable configuration (MAP estimate):

  [Math]
  \[ \hat{x} = \arg \max_{x \in \mathcal{X}^N} p(x_1, \ldots, x_N) = \arg \max_{x \in \mathcal{X}^N} \prod_{C \in C} \psi_C(x_C). \]

- the data likelihood or normalization constant

  [Math]
  \[ Z = \sum_{x \in \mathcal{X}^N} \prod_{C \in C} \psi_C(x_C) \]

- marginal distributions at single sites, or subsets:

  [Math]
  \[ p(X_s = x_s) = \frac{1}{Z} \sum_{x_t, t \neq s} \prod_{C \in C} \psi_C(x_C) \]
§1. Max-product message-passing on trees

Goal: Compute most probable configuration (MAP estimate) on a tree:

\[ \hat{x} = \arg \max_{x \in \mathcal{X}^N} \left\{ \prod_{s \in V} \exp(\theta_s(x_s)) \prod_{(s,t) \in E} \exp(\theta_{st}(x_s, x_t)) \right\}. \]

Max-product strategy: “Divide and conquer”: break global maximization into simpler sub-problems. (Lauritzen & Spiegelhalter, 1988)
Max-product on trees

Decompose: 
\[
\max_{x_1,x_2,x_3,x_4,x_5} p(x) = \max_{x_2} \left[ \exp(\theta_1(x_1)) \prod_{t \in N(2)} M_{t2}(x_2) \right].
\]

Update messages:

\[
M_{32}(x_2) = \max_{x_3} \left[ \exp(\theta_3(x_3) + \theta_{23}(x_2,x_3)) \prod_{v \in N(3) \setminus 2} M_{v3}(x_3) \right]
\]
Putting together the pieces

Max-product is an exact algorithm for any tree.

Update: \[ M_{ts}(x_s) \leftarrow \max_{x_t \in \mathcal{X}_t} \left\{ \exp \left[ \theta_{st}(x_s, x_t') + \theta_t(x_t') \right] \prod_{v \in \mathcal{N}(t) \setminus s} M_{vt}(x_t) \right\} \]

Max-marginals: \[ \tilde{p}_s(x_s; \theta) \propto \exp\{\theta_s(x_s)\} \prod_{t \in \mathcal{N}(s)} M_{ts}(x_s). \]
Summary: max-product on trees

- converges in at most graph diameter \# of iterations
- updating a single message is an $O(m^2)$ operation
- overall algorithm requires $O(Nm^2)$ operations

- upon convergence, yields the exact max-marginals:

\[
\tilde{p}_s(x_s) \propto \exp\{\theta_s(x_s)\} \prod_{t \in \mathcal{N}(s)} M_{ts}(x_s).
\]

- when $\text{arg max}_x x_s \tilde{p}_s(x_s) = \{x^s\}$ for all $s \in V$, then $x^* = (x_1^*, \ldots, x_N^*)$ is the unique MAP solution
- otherwise, there are multiple MAP solutions and one can be obtained by back-tracking
max-product can be applied to graphs with cycles (no longer exact)

empirical performance is often very good
Partial guarantees for max-product

- single-cycle graphs and Gaussian models

- local optimality guarantees:
  - “tree-plus-loop” neighborhoods
  - optimality on more general sub-graphs
    (Weiss & Freeman, 2001)
    (Wainwright et al., 2003)

- existence of fixed points for general graphs
  (Wainwright et al., 2003)

- exactness for certain matching problems

- no general optimality results
Partial guarantees for max-product

- single-cycle graphs and Gaussian models

- local optimality guarantees:
  - “tree-plus-loop” neighborhoods
    (Weiss & Freeman, 2001)
  - optimality on more general sub-graphs
    (Wainwright et al., 2003)

- existence of fixed points for general graphs
  (Wainwright et al., 2003)

- exactness for certain matching problems

- no general optimality results

Questions:
- Can max-product return an incorrect answer with high confidence?
- Any connection to classical approaches to integer programs?
Standard analysis via computation tree

- standard tool: computation tree of message-passing updates
  (Gallager, 1963; Weiss, 2001; Richardson & Urbanke, 2001)

- level $t$ of tree: all nodes whose messages reach the root (node 1) after $t$ iterations of message-passing
Example: Inexactness of standard max-product

(Wainwright et al., 2005)

Intuition:

- max-product solves (exactly) a modified problem on computation tree
- nodes *not equally weighted* in computation tree ⇒ max-product can output an incorrect configuration

(a) Diamond graph $G_{dia}$

(b) Computation tree (4 iterations)

- for example: asymptotic node fractions $\omega$ in this computation tree:

$$\begin{bmatrix} \omega(1) & \omega(2) & \omega(3) & \omega(4) \end{bmatrix} = \begin{bmatrix} 0.2393 & 0.2607 & 0.2607 & 0.2393 \end{bmatrix}$$
A whole family of non-exact examples

\[\theta_s(x_s) = \begin{cases} \alpha x_s & \text{if } s = 1 \text{ or } s = 4 \\ \beta x_s & \text{if } s = 2 \text{ or } s = 3 \end{cases}\]

\[\theta_{st}(x_s, x_t) = \begin{cases} -\gamma & \text{if } x_s \neq x_t \\ 0 & \text{otherwise} \end{cases}\]

- for \(\gamma\) sufficiently large, optimal solution is always either 
  \[1^4 = [1 \ 1 \ 1 \ 1] \text{ or } (\!1)^4 = [(-1) \ (-1) \ (-1) \ (-1)]\]

- first-order LP relaxation always exact for this problem

- max-product and LP relaxation give different decision boundaries:

  **Optimal/LP boundary:** 
  \[\hat{x} = \begin{cases} 1^4 & \text{if } 0.25\alpha + 0.25\beta \geq 0 \\ (-1)^4 & \text{otherwise} \end{cases}\]

  **Max-product boundary:** 
  \[\hat{x} = \begin{cases} 1^4 & \text{if } 0.2393\alpha + 0.2607\beta \geq 0 \\ (-1)^4 & \text{otherwise} \end{cases}\]
§3. A more general class of algorithms

- by introducing weights on edges, obtain a more general family of reweighted max-product algorithms
- with suitable edge weights, connected to linear programming relaxations
- many variants of these algorithms:
  - sequential TRMP (Kolmogorov, 2005)
  - convex message-passing (Weiss et al., 2007)
  - dual updating schemes (e.g., Globerson & Jaakkola, 2007)
Tree-reweighted max-product algorithms

(Wainwright, Jaakkola & Willsky, 2002)

Message update from node $t$ to node $s$:

$$M_{ts}(x_s) \leftarrow \kappa \max_{x'_t \in \mathcal{X}_t} \left\{ \exp \left[ \frac{\theta_{st}(x_s, x'_t)}{\rho_{st}} \right] + \theta_t(x'_t) \right\} \frac{\prod_{v \in \mathcal{N}(t) \setminus s} \left[ M_{vt}(x_t) \right]^{\rho_{vt}}}{\left[ M_{st}(x_t) \right]^{1 - \rho_{ts}}}.$$

Properties:

1. Modified updates remain distributed and purely local over the graph.
   - Messages are reweighted with $\rho_{st} \in [0, 1]$.
2. Key differences:
   - Potential on edge $(s, t)$ is rescaled by $\rho_{st} \in [0, 1]$.
   - Update involves the reverse direction edge.
3. The choice $\rho_{st} = 1$ for all edges $(s, t)$ recovers standard update.
**Edge appearance probabilities**

**Experiment:** What is the probability $\rho_e$ that a given edge $e \in E$ belongs to a tree $T$ drawn randomly under $\rho$?

1. Original
2. $\rho(T^1) = \frac{1}{3}$
3. $\rho(T^2) = \frac{1}{3}$
4. $\rho(T^3) = \frac{1}{3}$

In this example: $\rho_b = 1; \quad \rho_e = \frac{2}{3}; \quad \rho_f = \frac{1}{3}$.

The vector $\rho_e = \{ \rho_e \mid e \in E \}$ must belong to the *spanning tree polytope*. (Edmonds, 1971)
§4. Reweighted max-product and linear programming

- **MAP as integer program:** \( f^* = \max_{x \in \mathcal{X}^N} \{ \sum_{s \in V} \theta_s(x_s) + \sum_{(s,t) \in E} \theta_{st}(x_s, x_t) \} \)

- **Define local marginal distributions** (e.g., for \( m = 3 \) states):

  \[
  \mu_s(x_s) = \begin{bmatrix} \mu_s(0) \\ \mu_s(1) \\ \mu_s(2) \end{bmatrix} \quad \mu_{st}(x_s, x_t) = \begin{bmatrix} \mu_{st}(0,0) & \mu_{st}(0,1) & \mu_{st}(0,2) \\ \mu_{st}(1,0) & \mu_{st}(1,1) & \mu_{st}(1,2) \\ \mu_{st}(2,0) & \mu_{st}(2,1) & \mu_{st}(2,2) \end{bmatrix}
  \]
§4. Rewighted max-product and linear programming

- MAP as integer program: \( f^* = \max_{x \in \mathcal{X}^N} \{ \sum_{s \in V} \theta_s(x_s) + \sum_{(s,t) \in E} \theta_{st}(x_s, x_t) \} \)

- define local marginal distributions (e.g., for \( m = 3 \) states):
  \[
  \mu_s(x_s) = \begin{bmatrix} \mu_s(0) \\ \mu_s(1) \\ \mu_s(2) \end{bmatrix} \quad \mu_{st}(x_s, x_t) = \begin{bmatrix} \mu_{st}(0,0) & \mu_{st}(0,1) & \mu_{st}(0,2) \\ \mu_{st}(1,0) & \mu_{st}(1,1) & \mu_{st}(1,2) \\ \mu_{st}(2,0) & \mu_{st}(2,1) & \mu_{st}(2,2) \end{bmatrix}
  \]

- alternative formulation of MAP as linear program?
  \[
g^* = \max_{(\mu_s, \mu_{st}) \in \mathbb{M}(G)} \{ \sum_{s \in V} E_{\mu_s}[\theta_s(x_s)] + \sum_{(s,t) \in E} E_{\mu_{st}}[\theta_{st}(x_s, x_t)] \}
  \]

  Local expectations: \( E_{\mu_s}[\theta_s(x_s)] := \sum_{x_s} \mu_s(x_s) \theta_s(x_s) \).
§4. Reweighted max-product and linear programming

- MAP as integer program: \( f^* = \max_{x \in \mathcal{X}^N} \{ \sum_{s \in V} \theta_s(x_s) + \sum_{(s,t) \in E} \theta_{st}(x_s, x_t) \} \)

- define local marginal distributions (e.g., for \( m = 3 \) states):
  \[
  \mu_s(x_s) = \begin{bmatrix} \mu_s(0) \\ \mu_s(1) \\ \mu_s(2) \end{bmatrix} \quad \mu_{st}(x_s,x_t) = \begin{bmatrix} \mu_{st}(0,0) & \mu_{st}(0,1) & \mu_{st}(0,2) \\ \mu_{st}(1,0) & \mu_{st}(1,1) & \mu_{st}(1,2) \\ \mu_{st}(2,0) & \mu_{st}(2,1) & \mu_{st}(2,2) \end{bmatrix}
  \]

- alternative formulation of MAP as linear program?
  \[
  g^* = \max_{(\mu_s,\mu_{st}) \in \mathcal{M}(G)} \{ \sum_{s \in V} \mathbb{E}_{\mu_s}[\theta_s(x_s)] + \sum_{(s,t) \in E} \mathbb{E}_{\mu_{st}}[\theta_{st}(x_s, x_t)] \}
  \]

  Local expectations: \( \mathbb{E}_{\mu_s}[\theta_s(x_s)] := \sum_{x_s} \mu_s(x_s) \theta_s(x_s) \).

**Key question:** What constraints must local marginals \( \{\mu_s, \mu_{st}\} \) satisfy?
Marginal polytopes for general undirected models

- $\mathcal{M}(G) \equiv \text{set of all } \textit{globally realizable} \text{ marginals } \{\mu_s, \mu_{st}\}$:

\[
\left\{ \bar{\mu} \in \mathbb{R}^d \mid \mu_s(x_s) = \sum_{x_t, t \neq s} p_\mu(x), \text{ and } \mu_{st}(x_s, x_t) = \sum_{x_u, u \neq s, t} p_\mu(x) \right\}
\]

for some $p_\mu(\cdot)$ over $(X_1, \ldots, X_N) \in \{0, 1, \ldots, m - 1\}^N$.

- polytope in $d = m|V| + m^2|E|$ dimensions ($m$ per vertex, $m^2$ per edge)
- with $m^N$ vertices
- number of facets?
Marginal polytope for trees

- \( \mathbb{M}(T) \equiv \text{special case of marginal polytope for tree } T \)
- local marginal distributions on nodes/edges (e.g., \( m = 3 \))

\[
\mu_s(x_s) = \begin{bmatrix} \mu_s(0) \\ \mu_s(1) \\ \mu_s(2) \end{bmatrix} \quad \mu_{st}(x_s, x_t) = \begin{bmatrix} \mu_{st}(0, 0) & \mu_{st}(0, 1) & \mu_{st}(0, 2) \\ \mu_{st}(1, 0) & \mu_{st}(1, 1) & \mu_{st}(1, 2) \\ \mu_{st}(2, 0) & \mu_{st}(2, 1) & \mu_{st}(2, 2) \end{bmatrix}
\]

Deep fact about tree-structured models: If \( \{\mu_s, \mu_{st}\} \) are non-negative and locally consistent:

**Normalization:** \[ \sum_{x_s} \mu_s(x_s) = 1 \]

**Marginalization:** \[ \sum_{x'_t} \mu_{st}(x_s, x'_t) = \mu_s(x_s), \]

then on any tree-structured graph \( T \), they are globally consistent.

Follows from junction tree theorem (Lauritzen & Spiegelhalter, 1988).
Max-product on trees: Linear program solver

- MAP problem as a simple linear program:

\[
    f(\hat{x}) = \arg \max_{\vec{\mu} \in \mathcal{M}(T)} \left\{ \sum_{s \in V} \mathbb{E}_{\mu_s}[\theta_s(x_s)] + \sum_{(s,t) \in E} \mathbb{E}_{\mu_{st}}[\theta_{st}(x_s, x_t)] \right\}
\]

subject to \(\vec{\mu}\) in tree marginal polytope:

\[
\mathcal{M}(T) = \left\{ \vec{\mu} \geq 0, \sum_{x_s} \mu_s(x_s) = 1, \sum_{x'_t} \mu_{st}(x_s, x'_t) = \mu_s(x_s) \right\}.
\]

Max-product and LP solving:

- on tree-structured graphs, max-product is a dual algorithm for solving the tree LP. (Wai. & Jordan, 2003)

  max-product message \(M_{ts}(x_s) \equiv\) Lagrange multiplier for enforcing the constraint \(\sum_{x'_t} \mu_{st}(x_s, x'_t) = \mu_s(x_s)\).
Tree-based relaxation for graphs with cycles

Set of locally consistent pseudomarginals for general graph $G$:

$$\mathcal{L}(G) = \left\{ \vec{\tau} \in \mathbb{R}^d \mid \vec{\tau} \geq 0, \sum_{x_s} \tau_s(x_s) = 1, \sum_{x_t} \tau_{st}(x_s, x_t') = \tau_s(x_s) \right\}.$$

**Key:** For a general graph, $\mathcal{L}(G)$ is an outer bound on $\mathcal{M}(G)$, and yields a linear-programming relaxation of the MAP problem:

$$f(\hat{x}) = \max_{\bar{\mu} \in \mathcal{M}(G)} \theta^T \bar{\mu} \leq \max_{\vec{\tau} \in \mathcal{L}(G)} \theta^T \vec{\tau}.$$
Looseness of $\mathbb{L}(G)$ with graphs with cycles

Locally consistent (pseudo)marginals

Pseudomarginals satisfy the “obvious” local constraints:

**Normalization:** $\sum_{x'_s} \tau_s(x'_s) = 1$ for all $s \in V$.

**Marginalization:** $\sum_{x'_s} \tau_s(x'_s, x_t) = \tau_t(x_t)$ for all edges $(s,t)$. 
First-order (tree-based) LP relaxation:

\[
f(\hat{x}) \leq \max_{\tau \in \mathcal{L}(G)} \left\{ \sum_{s \in V} E_{\tau_s} [\theta_s(x_s)] + \sum_{(s,t) \in E} E_{\tau_{st}} [\theta_{st}(x_s, x_t)] \right\}
\]

Results: (Wainwright et al., 2005; Kolmogorov & Wainwright, 2005):

(a) **Strong tree agreement** Any TRW fixed-point that satisfies the strong tree agreement condition specifies an optimal LP solution.

(b) **LP solving:** For any binary pairwise problem, TRW max-product solves the first-order LP relaxation.

(c) **Persistence for binary problems:** Let \( S \subseteq V \) be the subset of vertices for which there exists a single point \( x^*_{s} \in \arg \max_{x_s} \nu^*_s(x_s) \). Then for any optimal solution, it holds that \( y_s = x^*_{s} \).
On-going work on LPs and conic relaxations

- tree-reweighted max-product solves first-order LP for any binary pairwise problem  
  (Kolmogorov & Wainwright, 2005)

- convergent dual ascent scheme; LP-optimal for binary pairwise problems  
  (Globerson & Jaakkola, 2007)

- convex free energies and zero-temperature limits  
  (Wainwright et al., 2005, Weiss et al., 2006; Johnson et al., 2007)

- coding problems: adaptive cutting-plane methods  
  (Taghavi & Siegel, 2006; Dimakis et al., 2006)

- dual decomposition and sub-gradient methods:  
  (Feldman et al., 2003; Komodakis et al., 2007, Duchi et al., 2007)

- solving higher-order relaxations; rounding schemes  
  (e.g., Sontag et al., 2008; Ravikumar et al., 2008)
Hierarchies of conic programming relaxations


- hierarchies of SDP relaxations for polynomial programming (Lasserre, 2001; Parrilo, 2002)

- intermediate between LP and SDP: second-order cone programming (SOCP) relaxations (Ravikumar & Lafferty, 2006; Kumar et al., 2008)

- all relaxations: particular outer bounds on the marginal polyope

Key questions:
- when are particular relaxations tight?
- when does more computation (e.g., LP $\rightarrow$ SOCP $\rightarrow$ SDP) yield performance gains?
Stereo computation: Middlebury stereo benchmark set

- standard set of benchmarked examples for stereo algorithms (Scharstein & Szeliski, 2002)
- Tsukuba data set: Image sizes $384 \times 288 \times 16$ ($W \times H \times D$)

(a) Original image

(b) Ground truth disparity
Comparison of different methods

(a) Scanline dynamic programming
(b) Graph cuts
(c) Ordinary belief propagation
(d) Tree-reweighted max-product

(a), (b): Scharstein & Szeliski, 2002; (c): Sun et al., 2002 (d): Weiss, et al., 2005;
Ordinary belief propagation
Tree-reweighted max-product
Ground truth