Introduction to graphical models: Lecture I

Martin Wainwright

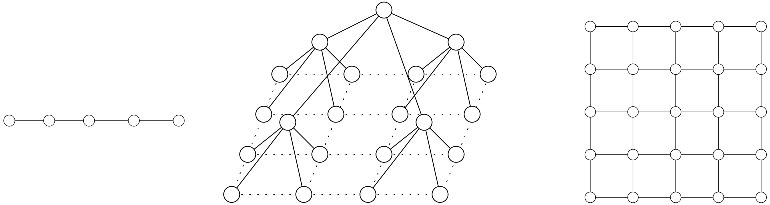
UC Berkeley
Departments of Statistics, and EECS

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

January 28, 2013

Introduction

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

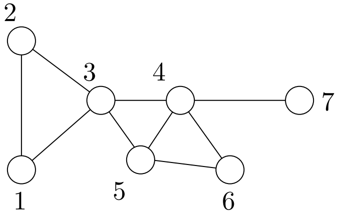


(a) Markov chain (b) Multiscale quadtree (c) Two-dimensional grid

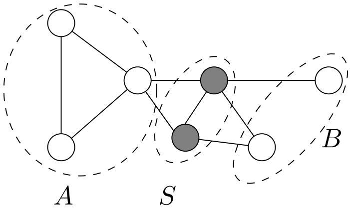
- useful in many statistical and computational fields:
  - spatial statistics
  - statistical physics
  - statistical machine learning, artificial intelligence
  - computational biology, bioinformatics
  - statistical signal/image processing
  - communication and information theory

Graphs and random variables

- associate to each node $s \in V$ a random variable X_s
- for each subset $A \subseteq V$, random vector $X_A := \{X_s, s \in A\}$.



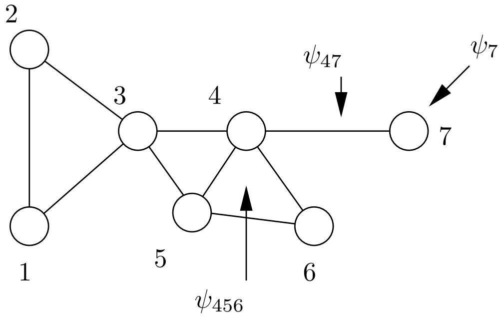
Maximal cliques $(123), (345), (456), (47)$



Vertex cutset S

- a *clique* $C \subseteq V$ is a subset of vertices all joined by edges
- a *vertex cutset* is a subset $S \subset V$ whose removal breaks the graph into two or more pieces

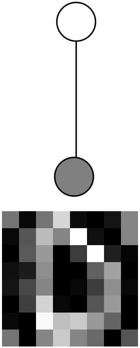
What are undirected graphical models?



- clique C is a fully connected subset of vertices
- non-negative compatibility function ψ_C defined on variables $x_C = \{x_s, s \in C\}$
- associated undirected graphical model is the collection of all distributions that factorize in the form

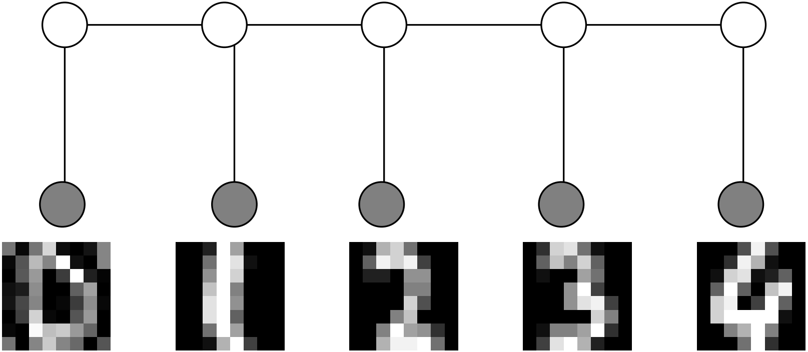
$$p(x_1, \ldots, x_N) = \frac{1}{Z} \prod_{C \in \mathfrak{C}} \psi_C(x_C).$$

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: Social network analysis

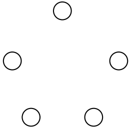
Vote of person s: $$x_s = \begin{cases} +1 & \text{if individual } s \text{ votes "yes"} \\ -1 & \text{if individual } s \text{ votes "no"} \end{cases}$$

Example: Social network analysis

Vote of person s: $$x_s = \begin{cases} +1 & \text{if individual } s \text{ votes “yes”} \\ -1 & \text{if individual } s \text{ votes “no”} \end{cases}$$

(1) Independent voting

$$p(x_1, \ldots, x_5) \propto \prod_{s=1}^{5} \exp(\theta_s x_s)$$

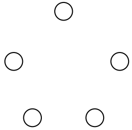


Example: Social network analysis

$$\text{Vote of person } s: \qquad x_s = \begin{cases} +1 & \text{if individual } s \text{ votes "yes"} \\ -1 & \text{if individual } s \text{ votes "no"} \end{cases}$$

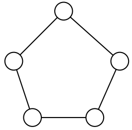
(1) Independent voting

$$p(x_1, \ldots, x_5) \propto \prod_{s=1}^{5} \exp(\theta_s x_s)$$



(2) Cycle-based voting

$$p(x_1, \ldots, x_5) \propto \prod_{s=1}^{5} \exp(\theta_s x_s) \prod_{(s,t) \in C} \exp(\theta_{st} x_s \, x_t)$$

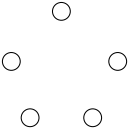


Example: Social network analysis

$$\text{Vote of person } s: \qquad x_s = \begin{cases} +1 & \text{if individual } s \text{ votes "yes"} \\ -1 & \text{if individual } s \text{ votes "no"} \end{cases}$$

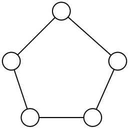
(1) Independent voting

$$p(x_1, \ldots, x_5) \propto \prod_{s=1}^{5} \exp(\theta_s x_s)$$



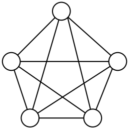
(2) Cycle-based voting

$$p(x_1, \ldots, x_5) \propto \prod_{s=1}^{5} \exp(\theta_s x_s) \prod_{(s,t)\in C} \exp(\theta_{st} x_s\, x_t)$$

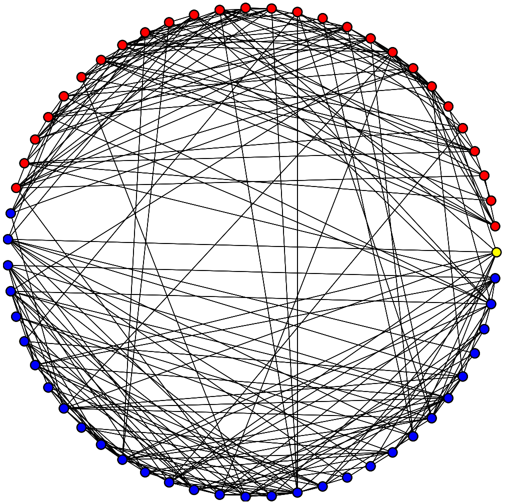


(3) Full clique voting

$$p(x_1, \ldots, x_5) \propto \prod_{s=1}^{5} \exp(\theta_s x_s) \prod_{s\neq t} \exp(\theta_{st} x_s x_t)$$



Graph fit to US politicians



(Ravikumar et al., AOS 2010)

Example: Depth estimation in computer vision

Stereo pairs: two images taken from horizontally-offset cameras

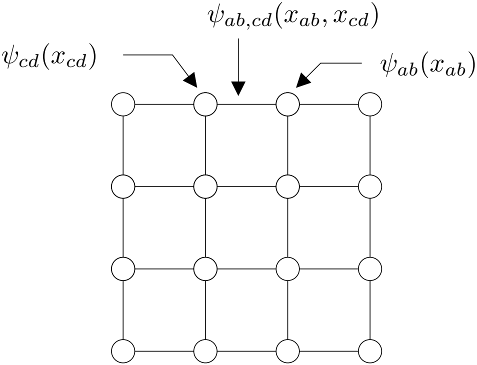
Modeling depth with a graphical model

Introduce variable at pixel location (a, b):

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

Left image

Right image



Use message-passing algorithms to estimate most likely offset/depth map.

(Szeliski et al., 2005)

Example: Communication over noisy channels

Goal: Achieve reliable communication over a noisy channel.

- wide variety of applications: satellite communication, sensor networks, computer memory, neural communication
- error-control codes based on careful addition of redundancy, with their fundamental limits determined by Shannon theory
- very active area of contemporary research: *graphical codes* (e.g., turbo codes, LDPC) and message-passing algorithms (e.g., Gallager, 1963; Berroux et al., 1993; MacKay, 1999; Richardson & Urbanke, 2007)

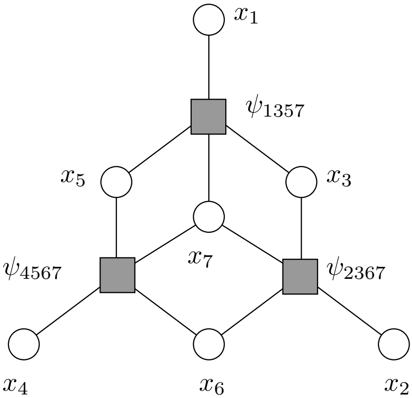
Graphical codes and decoding

Parity check matrix

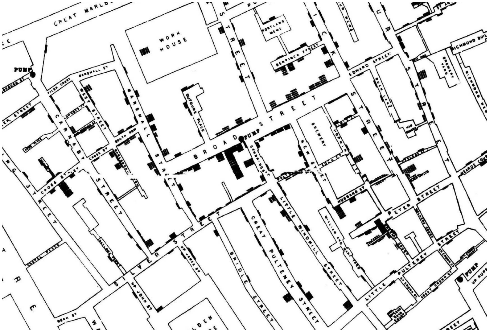
$$H = \begin{bmatrix} 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix}$$

Codeword: [0 1 0 1 0 1 0]
Non-codeword: [0 0 0 0 0 1 1]

Factor graph



Example: Epidemiological networks

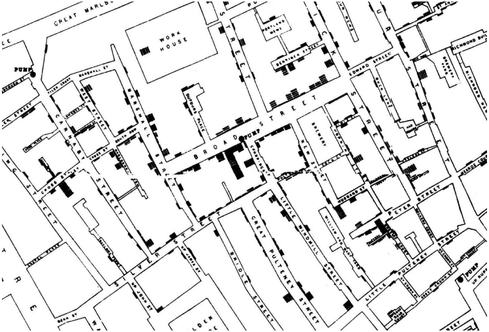


(a) Cholera epidemic (London, 1854)

Snow, 1855

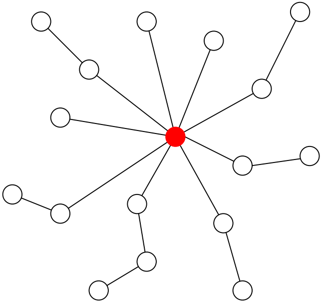
- network structure associated with spread of disease

Example: Epidemiological networks



(a) Cholera epidemic (London, 1854)

Snow, 1855



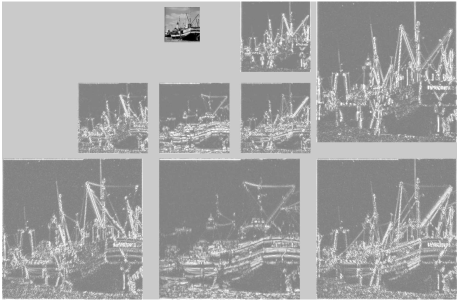
(b) “Spoke-hub” network

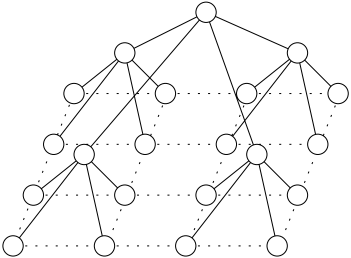
- network structure associated with spread of disease
- useful diagnostic information: contaminated water from Broad Street pump

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)

Many other examples

- natural language processing (e.g., parsing, translation)
- computational biology (gene sequences, protein folding, phylogenetic reconstruction)
- transportation and commodity networks
- data compression and source coding
- satisfiability problems (3-SAT, MAX-XORSAT, graph colouring)
- robotics (path planning, tracking, navigation)
- sensor network deployments (e.g., distributed detection, estimation, fault monitoring)
- ...

Factorization and Markov properties

The graph G can be used to impose constraints on the random vector $X = X_V$ (or on the distribution p) in different ways.

Markov property: X is *Markov w.r.t G* if X_A and X_B are conditionally indpt. given X_S whenever S separates A and B.

Factorization: The distribution p *factorizes according to G* if it can be expressed as a product over cliques:

$$p(x_1, x_2, \ldots, x_N) = \underbrace{\frac{1}{Z}}_{\text{Normalization}} \prod_{C \in \mathcal{C}} \underbrace{\psi_C(x_C)}_{\text{compatibility function on clique } C}$$

Theorem: (Hammersley & Clifford, 1973) For strictly positive $p(\cdot)$, the Markov property and the Factorization property are equivalent.

Core computational challenges

Given an undirected graphical model (Markov random field):

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

How to efficiently compute?

- most probable configuration (MAP estimate):

$$\textbf{Maximize}: \qquad \widehat{x} = \arg \max_{\mathbf{x} \in \mathcal{X}^N} p(x_1, \ldots, x_N) \; = \; \arg \max_{\mathbf{x} \in \mathcal{X}^N} \prod_{C \in \mathcal{C}} \psi_C(x_C).$$

- the data likelihood or normalization constant

$$\textbf{Sum/integrate}: \qquad Z \; = \; \sum_{x \in \mathcal{X}^N} \prod_{C \in \mathcal{C}} \psi_C(x_C)$$

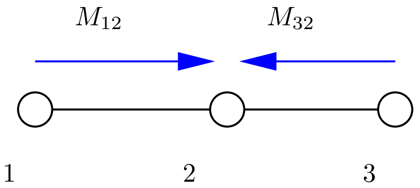
- marginal distributions at single sites, or subsets:

$$\textbf{Sum/integrate}: \qquad p(X_s = x_s) = \frac{1}{Z} \sum_{x_t,\, t \neq s} \prod_{C \in \mathcal{C}} \psi_C(x_C)$$

§1. Max-product message-passing on trees

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

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

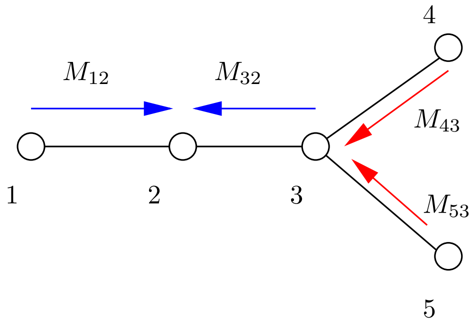


$$\max_{x_1,x_2,x_3} p(\mathbf{x}) \;=\; \max_{x_2} \left[\exp(\theta_2(x_2)) \prod_{t\in 1,3} \left\{ \max_{x_t} \exp[\theta_t(x_t) + \theta_{2t}(x_2,x_t)] \right\} \right]$$

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

Max-product on trees

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

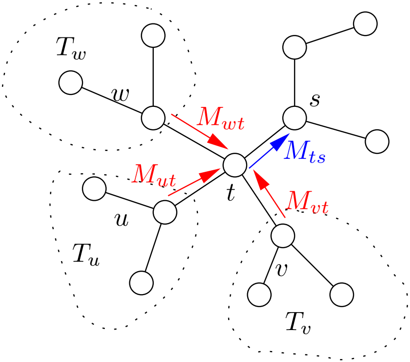


Update messages:

$$M_{32}(x_2) \;=\; \max_{x_3} \left[\exp(\theta_3(x_3) + \theta_{23}(x_2, x_3)) \prod_{v \in N(3)\backslash 2} M_{v3}(x_3)\right]$$

Putting together the pieces

Max-product is an exact algorithm for any tree.



M_{ts}	$\equiv$	message from node t to s
$\mathcal{N}(t)$	$\equiv$	neighbors of node t

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

Max-marginals: $\widetilde{p}_s(x_s; \theta) \propto \exp\{\theta_s(x_s)\} \prod_{t \in \mathcal{N}(s)} M_{ts}(x_s)$.

Summary: max-product on trees

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

- upon convergence, yields the exact *max-marginals*:

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

- when $\arg max_{x_s} \widetilde{p}_s(x_s) = \{x^{s}\}$ 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