

CS 6347

Lectures 6 & 7

Approximate MAP Inference

Belief Propagation

- Efficient method for inference on a tree
- Represent the variable elimination process as a collection of messages passed between nodes in the tree
 - The messages keep track of the potential functions produced throughout the elimination process

Belief Propagation (for pairwise MRFs)

- $p(x_1, \dots, x_n) = \frac{1}{Z} \prod_{i \in V} \phi_i(x_i) \prod_{(i,j) \in E} \psi_{ij}(x_i, x_j)$

$$m_{i \rightarrow j}(x_j) = \sum_{x_i} \phi_i(x_i) \psi_{ij}(x_i, x_j) \prod_{k \in N(i) \setminus j} m_{k \rightarrow i}(x_i)$$

where $N(i)$ is the set of neighbors of node i in the graph

- Messages are passed in two phases: from the leaves up to the root and then from the root down to the leaves

MAP Inference

- Compute the most likely assignment under the (conditional) joint distribution

$$x^* = \arg \max_x p(x)$$

- Can encode 3-SAT, maximum independent set problem, etc. as a MAP inference problem

Max-Product (for pairwise MRFs)

- $p(x_1, \dots, x_n) = \frac{1}{Z} \prod_{i \in V} \phi_i(x_i) \prod_{(i,j) \in E} \psi_{ij}(x_i, x_j)$

$$m_{i \rightarrow j}(x_j) = \max_{x_i} \left[\phi_i(x_i) \psi_{ij}(x_i, x_j) \prod_{k \in N(i) \setminus j} m_{k \rightarrow i}(x_i) \right]$$

- Guaranteed to produce the correct answer on a tree
- Typical applications do not require computing Z

Max-Product

- To construct the maximizing assignment, we look at the max-marginal produced by the algorithm

$$\mu_i(x_i) = \frac{1}{Z} \phi_i(x_i) \prod_{k \in N(i)} m_{k \rightarrow i}(x_i)$$

- Last time, we argued that, on a tree,

$$\mu_i(x_i) = \max_{x_1, \dots, x_{i-1}, x_{i+1}, \dots, x_n} p(x_1, \dots, x_n)$$

Reparameterization

- The messages passed in max-product can be used to construct a **reparameterization** of the joint distribution

$$p(x_1, \dots, x_n) = \frac{1}{Z} \prod_{i \in V} \phi_i(x_i) \prod_{(i,j) \in E} \psi_{ij}(x_i, x_j)$$

and

$$p(x_1, \dots, x_n) = \frac{1}{Z} \prod_{i \in V} \left[\phi_i(x_i) \prod_{k \in N(i)} m_{k \rightarrow i}(x_i) \right] \prod_{(i,j) \in E} \frac{\psi_{ij}(x_i, x_j)}{m_{i \rightarrow j}(x_j) m_{j \rightarrow i}(x_i)}$$

Reparameterization

$$p(x_1, \dots, x_n) = \frac{1}{Z} \prod_{i \in V} \left[\phi_i(x_i) \prod_{k \in N(i)} m_{k \rightarrow i}(x_i) \right] \prod_{(i,j) \in E} \frac{\psi_{ij}(x_i, x_j)}{m_{i \rightarrow j}(x_j) m_{j \rightarrow i}(x_i)}$$

- Reparameterizations do not change the partition function, the MAP solution, or the factorization of the joint distribution
 - They push "weight" around between the different factors
- Other reparameterizations are possible/useful

Tree Reparameterization

- On a tree, the joint distribution has a special form

$$p(x_1, \dots, x_n) = \frac{1}{Z'} \prod_{i \in V} \mu_i(x_i) \prod_{(i,j) \in E} \frac{\mu_{ij}(x_i, x_j)}{\mu_i(x_i)\mu_j(x_j)}$$

- μ_i is the max-marginal distribution of the i^{th} variable and μ_{ij} is the max-marginal distribution for the edge $(i, j) \in E$
- How to express μ_{ij} as a function of the messages and the potential functions?

MAP in General MRFs

- While max-product solves the MAP problem on trees, the MAP problem in MRFs is, in general, intractable
 - Don't expect to be able to solve the problem exactly
 - Will settle for “good” approximations
 - Can use max-product messages as a starting point
- This is an active area of research
 - Advances are constantly being made

Upper Bounds

$$\max_{x_1, \dots, x_n} p(x_1, \dots, x_n) \leq \frac{1}{Z} \prod_{i \in V} \max_{x_i} \phi_i(x_i) \prod_{(i,j) \in E} \max_{x_i, x_j} \psi_{ij}(x_i, x_j)$$

- This provides an upper bound on the optimization problem
 - Do other reparameterizations provide better bounds?

Duality

$$L(m) = \frac{1}{Z} \prod_{i \in V} \max_{x_i} \left[\phi_i(x_i) \prod_{k \in N(i)} m_{k \rightarrow i}(x_i) \right] \prod_{(i,j) \in E} \max_{x_i, x_j} \left[\frac{\psi_{ij}(x_i, x_j)}{m_{i \rightarrow j}(x_j) m_{j \rightarrow i}(x_i)} \right]$$

- We construct a dual optimization problem

$$\min_m L(m) \geq \max_x p(x)$$

- The dual problem is log-convex in the messages

$$L(m)^\delta L(m')^{1-\delta} \geq L(\delta m + (1-\delta)m')$$

Equivalently, $\log L(m)$ is a convex function

Optimizing the Dual

- Minimizing $L(m)$
 - **Block coordinate descent:** improve the bound by changing only a small subset of the messages at a time (usually look like message-passing algorithms)
 - **Subgradient descent:** variant of gradient descent for non-differentiable functions
 - Many more methods...

Integer Programming

- We can also express the MAP problem as a 0,1 integer programming problem
 - Convert a maximum of a product into a maximum of a sum by taking logs
 - Introduce indicator variables, τ , to represent the chosen assignment

Integer Programming

- Introduce variables
 - $\tau_i(x_i) \in \{0,1\}$ for each $i \in V$ and x_i
 - $\tau_{ij}(x_i, x_j) \in \{0,1\}$ for each $(i, j) \in E$ and x_i, x_j
- The linear objective function is then

$$\max_{\tau} \sum_{i \in V} \sum_{x_i} \tau_i(x_i) \log \phi_i(x_i) + \sum_{(i,j) \in E} \sum_{x_i, x_j} \tau_{ij}(x_i, x_j) \log \psi_{ij}(x_i, x_j)$$

where the τ 's are required to satisfy certain marginalization conditions

Integer Programming

$$\max_{\tau} \sum_{i \in V} \sum_{x_i} \tau_i(x_i) \log \phi_i(x_i) + \sum_{(i,j) \in E} \sum_{x_i, x_j} \tau_{ij}(x_i, x_j) \log \psi_{ij}(x_i, x_j)$$

such that

$$\sum_{x_i} \tau_i(x_i) = 1$$

For all $i \in V$

$$\sum_{x_j} \tau_{ij}(x_i, x_j) = \tau_i(x_i)$$

For all $(i, j) \in E, x_i$

$$\tau_i(x_i) \in \{0, 1\}$$

For all $i \in V, x_i$

$$\tau_{ij}(x_i, x_j) \in \{0, 1\}$$

For all $(i, j) \in E, x_i, x_j$

Integer Programming

$$\max_{\tau} \sum_{i \in V} \sum_{x_i} \tau_i(x_i) \log \phi_i(x_i) + \sum_{(i,j) \in E} \sum_{x_i, x_j} \tau_{ij}(x_i, x_j) \log \psi_{ij}(x_i, x_j)$$

such that

These constraints define the vertices of the **marginal polytope** (set of all valid marginal distributions)

$$\sum_{x_i} \tau_i(x_i) = 1$$

For all $i \in V$

$$\sum_{x_j} \tau_{ij}(x_i, x_j) = \tau_i(x_i)$$

For all $(i, j) \in E, x_i$

$$\tau_i(x_i) \in \{0, 1\}$$

For all $i \in V, x_i$

$$\tau_{ij}(x_i, x_j) \in \{0, 1\}$$

For all $(i, j) \in E, x_i, x_j$

An Example: Independent Sets

- What is the integer programming problem corresponding to the uniform distribution over independent sets of a graph $G = (V, E)$?

$$p(x_V) = \frac{1}{Z} \prod_{(i,j) \in E} 1_{x_i + x_j \leq 1}$$

(worked out on the board)

Linear Relaxation

- The integer program can be relaxed into a linear program by replacing the 0,1 integrality constraints with linear constraints
 - This relaxed set of constraints forms the **local marginal polytope**
 - The τ 's no longer correspond to an achievable marginal distribution, so we call them pseudo-marginals
 - We call it a relaxation because the constraints have been relaxed: all solutions to the IP are contained as solutions of the LP
- Linear programming problems can be solved in polynomial time

Linear Relaxation

$$\max_{\tau} \sum_{i \in V} \sum_{x_i} \tau_i(x_i) \log \phi_i(x_i) + \sum_{(i,j) \in E} \sum_{x_i, x_j} \tau_{ij}(x_i, x_j) \log \psi_{ij}(x_i, x_j)$$

such that

$$\sum_{x_i} \tau_i(x_i) = 1$$

For all $i \in V$

$$\sum_{x_j} \tau_{ij}(x_i, x_j) = \tau_i(x_i)$$

For all $(i, j) \in E, x_i$

$$\tau_i(x_i) \in [0, 1]$$

For all $i \in V, x_i$

$$\tau_{ij}(x_i, x_j) \in [0, 1]$$

For all $(i, j) \in E, x_i, x_j$

An Example: Independent Sets

- What is the **linear** programming problem corresponding to the uniform distribution over independent sets of a graph $G = (V, E)$?

$$p(x_V) = \frac{1}{Z} \prod_{(i,j) \in E} 1_{x_i + x_j \leq 1}$$

- The MAP LP is a relaxation of the integer programming problem
 - MAP LP could have a better solution... (example in class)

LP vs. Dual

- Both the LP relaxation and the dual $L(m)$ provide an upper bound on the MAP objective function
 - That is, finding an optimal collection of messages is equivalent to finding the best **pseudo-marginals**
- In fact, they are equivalent optimization problems: this seems quite surprising because the problems look so different
 - The proof uses the method of Lagrange multipliers (a standard mathematical technique to construct dual optimization problems)

Tightness of the MAP LP

- When is it that solving the MAP LP (or equivalently, the dual optimization) is the same as solving the integer programming problem?
 - We say that there is no duality gap (or that the dual is tight) when this is the case
 - The answer can be expressed as a structural property of the graph