Probabilistic Graphical Models

David Sontag

New York University

Lecture 6, March 1, 2012

Today's lecture

1. Running-time analysis of elimination algorithm (treewidth)
   Done on blackboard
2. Sum-product Belief Propagation (BP)
   Done on blackboard
3. MAP inference as an ILP (integer linear program)

MAP inference

- Recall the MAP inference task,

$$\arg\max_{\mathbf{x}} p(\mathbf{x}), \qquad p(\mathbf{x}) = \frac{1}{Z} \prod_{c \in C} \phi_c(\mathbf{x}_c)$$

  (we assume any evidence has been subsumed into the potentials, as discussed in the last lecture)

- Since the normalization term is simply a constant, this is equivalent to

$$\arg\max_{\mathbf{x}} \prod_{c \in C} \phi_c(\mathbf{x}_c)$$

  (called the *max-product* inference task)

- Furthermore, since log is monotonic, letting $\theta_c(\mathbf{x_c}) = \lg \phi_c(\mathbf{x_c})$, we have that this is equivalent to

$$\arg\max_{\mathbf{x}} \sum_{c \in C} \theta_c(\mathbf{x}_c)$$

  (called *max-sum*)

Can apply variable elimination!

- Suppose we have a simple chain, $A - B - C - D$, and we want to find the MAP assignment,

$$\max_{a,b,c,d} \phi_{AB}(a,b)\phi_{BC}(b,c)\phi_{CD}(c,d)$$

- Just as we did before, we can push the maximizations inside to obtain:

$$\max_{a,b} \phi_{AB}(a,b) \max_{c} \phi_{BC}(b,c) \max_{d} \phi_{CD}(c,d)$$

or, equivalently,

$$\max_{a,b} \theta_{AB}(a,b) + \max_{c} \theta_{BC}(b,c) + \max_{d} \theta_{CD}(c,d)$$

- To find the actual maximizing assignment, we do a traceback (or keep back pointers)

Procedure Max-Product-VE (
  Φ, // Set of factors over $\boldsymbol{X}$
  $\prec$ // Ordering on $\boldsymbol{X}$
)
  Let $X_1, \ldots, X_k$ be an ordering of $\boldsymbol{X}$ such that
    $X_i \prec X_j$ iff $i < j$
    for $i = 1, \ldots, k$
      $(\Phi, \phi_{X_i}) \leftarrow$ Max-Product-Eliminate-Var(Φ, X_i)
    $\boldsymbol{x}^* \leftarrow$ Traceback-MAP$(\{\phi_{X_i} \; : \; i = 1, \ldots, k\})$
    return $\boldsymbol{x}^*, \Phi$ // Φ contains the probability of the MAP

Procedure Max-Product-Eliminate-Var (
  Φ, // Set of factors
  Z // Variable to be eliminated
)
  $\Phi' \leftarrow \{\phi \in \Phi \; : \; Z \in \mathit{Scope}[\phi]\}$
  $\Phi'' \leftarrow \Phi - \Phi'$
  $\psi \leftarrow \prod_{\phi \in \Phi'} \phi$
  $\tau \leftarrow \max_Z \psi$
  return $(\Phi'' \cup \{\tau\}, \psi)$

Procedure Traceback-MAP (
  $\{\phi_{X_i} \; : \; i = 1, \ldots, k\}$
)
  for $i = k, \ldots, 1$
    $\boldsymbol{u}_i \leftarrow (x^*_{i+1}, \ldots, x^*_k)\langle \mathit{Scope}[\phi_{X_i}] - \{X_i\}\rangle$
      // The maximizing assignment to the variables eliminated after X_i
    $x^*_i \leftarrow \arg\max_{x_i} \phi_{X_i}(x_i, \boldsymbol{u}_i)$
      // x^*_i is chosen so as to maximize the corresponding entry in the factor, relative to the previous choices $\boldsymbol{u}_i$
  return $\boldsymbol{x}^*$

Max-product belief propagation (for tree-structured MRFs)

- Same as sum-product belief propagation except that the messages are now:

$$m_{j\to i}(x_i) = \max_{x_j} \phi_j(x_j)\phi_{ij}(x_i, x_j) \prod_{k\in N(j)\setminus i} m_{k\to j}(x_j)$$

- After passing all messages, can compute single node *max-marginals*,

$$m_i(x_i) = \phi_i(x_i) \prod_{j\in N(i)} m_{j\to i}(x_i) \quad \propto \max_{\mathbf{x}_{V\setminus i}} p(\mathbf{x}_{V\setminus i}, x_i)$$

- If the MAP assignment $\mathbf{x}^*$ is **unique**, can find it by locally decoding each of the single node max-marginals, i.e.

$$x_i^* = \arg\max_{x_i} m_i(x_i)$$

- If the MAP assignment is not unique, can either:
  1. Randomly perturb the objective functions to make MAP unique, or
  2. Use the traceback method shown on previous slide

Exactly solving MAP, beyond trees

- MAP as a discrete optimization problem is

$$\arg\max_{\mathbf{x}} \sum_{i \in V} \theta_i(x_i) + \sum_{ij \in E} \theta_{ij}(x_i, x_j)$$

- Very general discrete optimization problem – many hard combinatorial optimization problems can be written as this (e.g., 3-SAT)
- Studied in operations research communities, theoretical computer science, AI (constraint satisfaction, weighted SAT), etc.
- Very fast moving field, both for theory and heuristics
- For some special cases of θ, polynomial-time algorithms known:
  - Trees; binary variables with *sub-modular* edge potentials; planar Ising model with $\mathbf{u} = \mathbf{0}$; matching problems; ...

MAP as an integer linear program (ILP)

- MAP as a discrete optimization problem is

$$\arg\max_{\mathbf{x}} \sum_{i \in V} \theta_i(x_i) + \sum_{ij \in E} \theta_{ij}(x_i, x_j).$$

- To turn this into an integer linear program, we introduce variables
  1. $\mu_i(x_i)$, one for each $i \in V$ and state x_i
  2. $\mu_{ij}(x_i, x_j)$, one for each edge $ij \in E$ and pair of states x_i, x_j
- The objective function is then

$$\max_{\mu} \sum_{i \in V} \sum_{x_i} \theta_i(x_i)\mu_i(x_i) + \sum_{ij \in E} \sum_{x_i, x_j} \theta_{ij}(x_i, x_j)\mu_{ij}(x_i, x_j)$$

- What do the constraints need to be? We want to enforce that every choice of μ corresponds one-to-one with an assignment $\mathbf{x}$

MAP as an integer linear program (ILP)

$$\max_{\mu} \sum_{i \in V} \sum_{x_i} \theta_i(x_i)\mu_i(x_i) + \sum_{ij \in E} \sum_{x_i, x_j} \theta_{ij}(x_i, x_j)\mu_{ij}(x_i, x_j)$$

- Subject to

$$\begin{aligned} \mu_i(x_i) &\in \{0,1\} \quad \forall i \in V, x_i \\ \mu_{ij}(x_i, x_j) &\in \{0,1\} \quad \forall ij \in E, x_i, x_j \end{aligned}$$

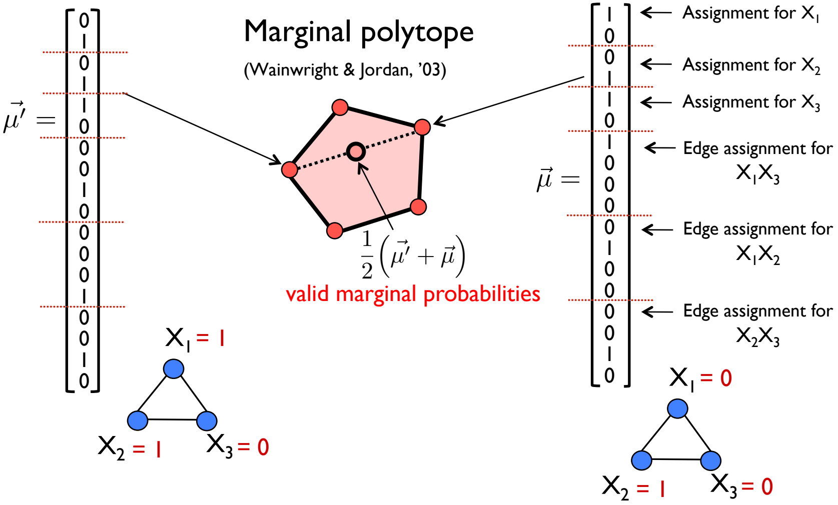
  and

$$\begin{aligned} \sum_{x_i} \mu_i(x_i) &= 1 \quad \forall i \in V \\ \sum_{x_i, x_j} \mu_{ij}(x_i, x_j) &= 1 \quad \forall ij \in E \end{aligned}$$

- We also need the edge variables μ_{ij} to be consistent with the node variables:

$$\begin{aligned} \mu_i(x_i) &= \sum_{x_j} \mu_{ij}(x_i, x_j) \quad \forall ij \in E, x_i \\ \mu_j(x_j) &= \sum_{x_i} \mu_{ij}(x_i, x_j) \quad \forall ij \in E, x_j \end{aligned}$$

Visualization of feasible μ vectors



MAP as an integer linear program (ILP)

$$\mathrm{MAP}(\theta) = \max_{\mu} \sum_{i \in V} \sum_{x_i} \theta_i(x_i)\mu_i(x_i) + \sum_{ij \in E} \sum_{x_i, x_j} \theta_{ij}(x_i, x_j)\mu_{ij}(x_i, x_j)$$

subject to:

$$\begin{aligned}
\mu_i(x_i) &\in \{0,1\} \quad \forall i \in V, x_i \\
\sum_{x_i} \mu_i(x_i) &= 1 \quad \forall i \in V \\
\mu_i(x_i) &= \sum_{x_j} \mu_{ij}(x_i, x_j) \quad \forall ij \in E, x_i \\
\mu_j(x_j) &= \sum_{x_i} \mu_{ij}(x_i, x_j) \quad \forall ij \in E, x_j
\end{aligned}$$

- Many extremely good off-the-shelf solvers, such as CPLEX and Gurobi