CS786
Lecture 16: May 26, 2012

MAP inference

[KF Chapter 13]

MAP Inference Techniques

- Variable Elimination (exact)
- Cluster tree propagation (exact)
  – Cluster graph propagation (approximate)
- Integer linear programming (exact)
  – (Relaxed) linear programming (approximate)
- Graph cut (exact in special cases)

Cluster Tree Recap

- Variable elimination:
  - Induces a cluster tree
  - Inference: message propagation on cluster tree
- Cluster tree:
  - Graph is a tree (i.e., no loops)
  - Node: cluster of variables
  - Edge: subset of variables (a.k.a. sepset) that are common to nodes it connects
  - Satisfies running intersection property

Max-Product Cluster Tree Calibration

- $\boldsymbol{C}_i$: variables in the cluster at node i

  $\beta_i(\boldsymbol{C}_i)$: factor at node i
- $\boldsymbol{S}_{ij}$: variables in sepset at edge $i - j$

  $\mu_{ij}(\boldsymbol{S}_{ij})$: factor at edge $i - j$
- Calibrated max-product cluster tree

  For all edges $i - j$: sepset factor is marginal of cluster factors

  $$\mu_{ij}(\boldsymbol{S}_{ij}) = \max_{X \in \boldsymbol{C}_i \backslash \boldsymbol{S}_{ij}} \beta_i(\boldsymbol{C}_i) = \max_{X \in \boldsymbol{C}_j \backslash \boldsymbol{S}_{ij}} \beta_j(\boldsymbol{C}_j)$$

Calibration by Message Passing

- Initialization:
  - Messages: $\delta_{i\to j} \leftarrow 1$ and $\delta_{j\to i} \leftarrow 1 \quad \forall ij$
  - Potentials: $\psi_i \leftarrow \prod$ potentials associated with $\boldsymbol{C}_i$
- Update messages until calibration

$$\delta_{i\to j}(\boldsymbol{S}_{ij}) \leftarrow \max_{X\in \boldsymbol{C}_i\setminus \boldsymbol{S}_{ij}} \left[\psi_i(\boldsymbol{C}_i) \prod_{k\in \boldsymbol{nb}(i)\setminus\{j\}} \delta_{k\to i}(\boldsymbol{S}_{ki})\right]$$

- Return

$$\beta_i(\boldsymbol{C}_i) \leftarrow \psi_i(\boldsymbol{C}_i) \textstyle\prod_{k\in nb(i)} \delta_{k\to i}(\boldsymbol{S}_{ij})$$

$$\mu_{ij}\left(\boldsymbol{S}_{ij}\right) \leftarrow \delta_{i\to j}\left(\boldsymbol{S}_{ij}\right)\delta_{j\to i}(\boldsymbol{S}_{ij})$$

Max-Product Cluster Graph Calibration (a.k.a. Max-Product Loopy Belief Propagation)

- Same as for max-product cluster tree calibration
- Disadvantages:
  - Convergence is not guaranteed
    - Damping techniques may be used to ensure convergence
- Advantages:
  - Approximation is often good in practice and MAP inference scales linearly with the size of the graph

Energy Minimization

- MAP queries:
  $\max_{\boldsymbol{X}} \Pr(\boldsymbol{X}|\boldsymbol{E}) = \max_{\boldsymbol{X}} \prod_i f_i(\boldsymbol{X})$
  Operations: max-product

- Consider the feature based representation
  $$\operatorname*{argmax}_X \prod_i f_i(\boldsymbol{X}) = \operatorname*{argmax}_{\boldsymbol{X}} e^{-\sum_j \lambda_j \phi_j(\boldsymbol{X})} = \operatorname*{argmin}_{\boldsymbol{X}} \sum_j \lambda_j \phi_j(\boldsymbol{X})$$
  where $\sum_j \lambda_j \phi_j(\boldsymbol{X})$ is often called the energy
  Operations: min-sum

Integer Linear Programming

- Since the objective is linear and the domain of each X is discrete, formulate as an integer linear program

- Let $I_{X=x}$ be a binary variable indicating whether $X = x$
- Let $\boldsymbol{X}_j$ be the set of variables associated with feature ϕ_j

$$\min_{\{I_{\boldsymbol{X}_j=\boldsymbol{x}_j}\}} \sum_j \lambda_j \phi_j(\boldsymbol{x}_j)\, I_{\boldsymbol{X}_j=\boldsymbol{x}_j}$$

s.t. $I_{\boldsymbol{X}_j=\boldsymbol{x}_j} \in \{0,1\}$

all $I_{\boldsymbol{X}_j=\boldsymbol{x}_j}$ are consistent

Algorithms for Integer LP

- Exact: branch-and-bound
  - Bounds: relaxed (continuous) linear program
- Approximate:
  - Relaxed (continuous) linear program
  - Simulated annealing
  - Stochastic hill-climbing

Graph-Cut

- Graph-cut: special case where the relaxed (continuous) linear program is exact
  - i.e. solution always consists of integer values even when the integer constraints are not enforced
- Graph-cut problem: find least costly cut in a graph that separates two nodes s and t
  - Each edge is labeled with a cost
  - The cost of a cut is the sum of the cost of all removed edges

Example

Graph-Cut for Pairwise Markov Networks

- Example Image segmentation

Important Note

- Variable Elimination and Cluster Tree Propagation scale exponentially with the size of the largest intermediate factor
- Integer Linear Programming and Graph-Cut can answer some MAP queries in polynomial time even though Variable Elimination and Cluster Tree Propagation would take exponential time
  - i.e., Complexity of ILP and GC do not generate intermediate factors