Reparameterization: a Universal Tool for Optimization and Counting

George Katsirelos

10/05/2017

WCSP/MRF

- A set of discrete variables $\mathbf{X}$, each with a domain D
- We define a joint function on all variables $f : D^{\mathbf{X}} \rightarrow S$
- By decomposing the joint function to a set $\mathcal{C}$ of functions of small arity (*factors*)
- Concise way of describing complicated functions

Function Aggregation – WCSP

$S \equiv \mathbb{R}^+ \cup \{0, \infty\}$

$$f(\mathbf{x}) = \sum_{c \in \mathcal{C}} c(\mathbf{x})$$

- f represents a cost or energy or potential
- Each c is a cost function

Function Aggregation – MRF

$S \equiv \mathbb{R}^+ \cup \{0\}$

$$f(\mathbf{x}) = \prod_{c \in \mathcal{C}} c(\mathbf{x})$$

- Each c is a probability table

$$P(\mathbf{x}) = \frac{f(\mathbf{x})}{Z}$$

$$Z = \frac{1}{\sum_{\mathbf{x}'} \prod_{c \in \mathcal{C}} c(\mathbf{x}')}$$

WCSP/MRF Equivalence

- Given MRF P, a WCSP P' has

$$c'(\mathbf{x}) = -\log c(\mathbf{x})$$

Then

$$exp(-f'(\mathbf{x})) \propto P(\mathbf{x})$$

$$Z = \frac{1}{\sum_{\mathbf{x}'} \prod_{c \in \mathcal{C}} exp(-c(\mathbf{x}'))}$$

- So we deal with costs only

MAP

- Maximum a posteriori estimation
- Compute assignment with maximum probability in MRF
  - By equivalence to WCSP, *same problem* as cost minimization
- Optimization of an NP-hard set, hence FP^{NP}
- Generalizes Boolean satisfiability, constraint satisfaction

Partition Function

- Compute Z, the normalization constant (probability mass of the function)
- P^{PP}-complete
- By Toda's theorem, this is Beyond PH

Marginal MAP

- Partition $\mathbf{X}$ into variable sets A, B
- Compute assignment $\mathbf{x}_A$ that maximizes probability mass of $f|_{\mathbf{x}_A}$
- NP^{PP}

Aside: WCSP as COP

- WCSP combines crisp CSP with arbitrary polynomial objective
  - Clever dual bounds
- *Small* arity is not necessary
- Can use the machinery developed in CSP for more expressiveness
  - Higher level language
  - Propagators
    - Global Cost Functions an underexplored area
- New scenarios
  - MAP: What's the most likely to succeed schedule
  - Marginal MAP: What choices can I make that make schedules more likely to succeed

Reparameterization

- Use a naive way to compute a bound
- Local transformation that leaves the problem unchanged
  - but improves naive bound
  - If we touch factors S, require

$$\forall \mathbf{x} \sum_{c \in S} c(\mathbf{x}) = \sum_{c \in S} c'(\mathbf{x})$$

- Dates back to at least the Held-Karp lower bound for TSP

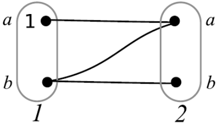
WCSP reparameterization

$\textsc{Move}(c_1, c_2, \mathbf{x}, \alpha)$

- Shifts α units of cost between c_1 and c_2 on the common assignment $\mathbf{x}$
- Shift direction: sign of α.
- α constrained: no negative costs!
- Commonly restricted to $scope(c_1) \subset scope(c_2)$ and in particular $|scope(c_1)| = 1$:

$$\textsc{Project}(\{i\}, \{i, j\}, a, \alpha)$$

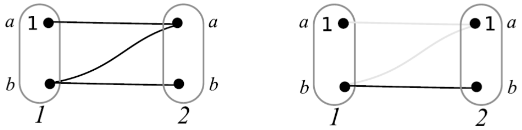
Example



Example

$$\textsc{Project}(\{1,2\},\{2\},a,1)$$

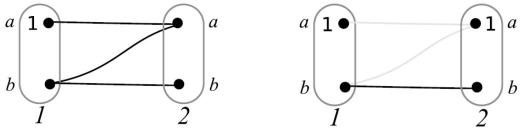
$$\rightarrow$$



Example

$$\textsc{Project}(\{1,2\},\{2\},a,1)$$

$$\rightarrow$$



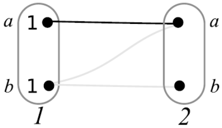
$$\leftarrow$$

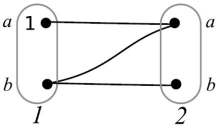
$$\textsc{Project}(\{1,2\},\{2\},a,-1)$$

Example

$\textsc{Project}(\{1,2\},\{1\},b,1)$

$\leftarrow$

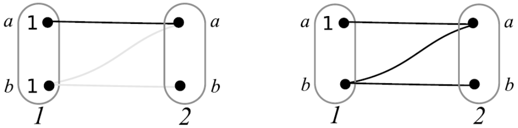




Example

$\textsc{Project}(\{1,2\},\{1\},b,1)$

$\leftarrow$



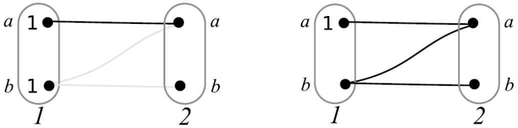
$\rightarrow$

$\textsc{Project}(\{1,2\},\{1\},b,-1)$

Example

$\textsc{Project}(\{1,2\},\{1\},b,1)$

$\leftarrow$

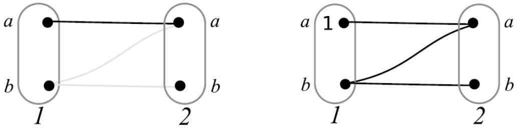


$\Downarrow \qquad \textsc{Project}()(\{1\},\varnothing,[],1)$

Example

$\textsc{Project}(\{1,2\},\{1\},b,1)$

$\leftarrow$



$\Downarrow \qquad \textsc{Project}()(\{1\},\varnothing,[],1)$

$c_{\varnothing} = 1$

Lower bounds for cost minimization

- The sum of the lower bound of each function $\min_{\mathbf{x}} \sum_c c(\mathbf{x}) \leq \sum_c \min_{\mathbf{x} \in c} c(\mathbf{x})$

Min Sum Diffusion

1. Choose overlapping factors c_1, c_2
2. For every $\mathbf{x}$ in the intersection, choose α so that $c_1(\mathbf{x}) = c_2(\mathbf{x})$
3. Repeat until convergence

- Averages factors
- Will converge as number of iterations goes to infinity, as long as each pair of factors is chosen infinitely often
- Will converge to arc consistent state

Block Coordinate Descent

- Min Sum Diffusion is a Block Coordinate Descent algorithm
- Differentiate on subproblem, order of updates
- At best will converge to optimum of linear relaxation
- Perform pruning

Branch-and-bound

1. Start with root node, corresponding to initial problem
2. Pick an *open* node
3. Compute dual bound
   1. If the primal bound is violated, *close* node; else
   2. Make a binary choice, replace by two new nodes
4. Go to step 2

Upper bound for Partition Function

- Product of mass of all factors

$$Z = \sum_{\mathbf{x}} \prod_{c} exp(-c(\mathbf{x})) \leq \prod_{c} \sum_{\mathbf{x}} exp(-c(\mathbf{x}))$$

- Proof: by distributing the product over the sum

Approximate Z

- Branch and bound
- Ignore subtrees as long as the contribution is small enough

1. Start with root node, corresponding to initial problem, $U = 0$
2. Pick an *open* node
3. Compute Z upper bound u
    1. If $u < \varepsilon U$, *close* node; else
    2. If full assignment, add its weight to U; else
    3. Make a binary choice, replace by two new nodes
4. Go to step 2

Marginal MAP

- Prune subtree as soon as upper bound for $Z(f_{\mathbf{x}_A})$ is lower than incumbent

1. Start with root node, corresponding to initial problem
2. Pick an *open* node
3. Compute $Z(f|_{\mathbf{x}_A})$ upper bound u
   1. If $u < \varepsilon U$, *close* node; else
   2. If all A variables have been assigned, compute $Z(f_{\mathbf{x}_A})$, replacing incumbent if needed; else
   3. Make a binary choice on variables in A, replace by two new nodes
4. Go to step 2

Conclusions

- Reparameterization is a universal tool
  - Maintains cost/probability of all assignments, so always applicable
  - Non-trivial improvement of trivial bounds
- Precise connection to linear programming in cost minimization
- Hierarchies of strengthening reparameterizations which change network
- Linear programming cuts

Q?