On Amortizing Inference Cost for Structured Prediction

Vivek Srikumar   Gourab Kundu   Dan Roth

Presenter: Zhan Xiong Chin

Overview

- Background and motivation
- Theorems and approximations
- Experimental results in Semantic Role Labeling

Background and Motivation

Background and Motivation

If we solve a lot of inference problems, can we sometimes reuse the results for new problems instead of starting from scratch?

Background and Motivation

- Integer linear programming (ILP) can be used to do inference on any structured prediction problem
    - Parts of speech tagging
    - Dependency parsing
    - Semantic role labeling

Example from Semantic Role Labeling

- From (Punyakanok et al., 2008)
- Given a **sentence** and a **verb**, label the corresponding **arguments** of the verb:

[$_{\text{A0}}$ I] [$_{\text{V}}$ *left*] [$_{\text{A1}}$ my pearls] [$_{\text{A2}}$ to my daughter-in-law] [$_{\text{AM-LOC}}$ in my will].

Example from Semantic Role Labeling

- Train classifiers that score how well each label fits each word/phrase
- Use ILP to maximize overall sum of scores given restrictions:

$$\underset{\mathbf{u}\in\{0,1\}^{|\mathbf{u}|}}{\operatorname{argmax}} \sum_{i=1}^{M} \sum_{c\in\mathcal{P}} p_{ic} u_{ic},$$

$$\sum_{c\in\mathcal{P}} u_{ic} = 1 \qquad \forall i,$$

$$\sum_{i=1}^{M} u_{i\mathrm{A0}} \leq 1$$

Background and Motivation

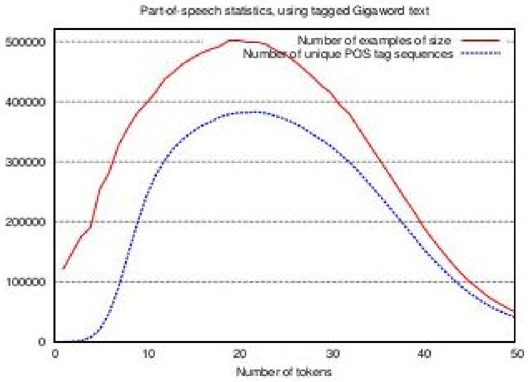
- Sometimes, there are many theoretically possible structures for a problem, but only a handful of commonly-seen ones:

[I] [left] [my pearls] [to my daughter-in-law] [in my will].

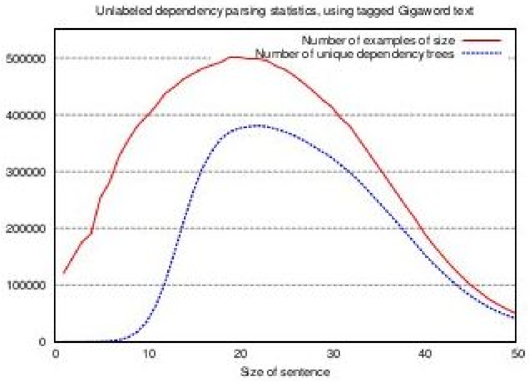
[He] [left] [his house] [to his son] [in a letter].

Background and Motivation

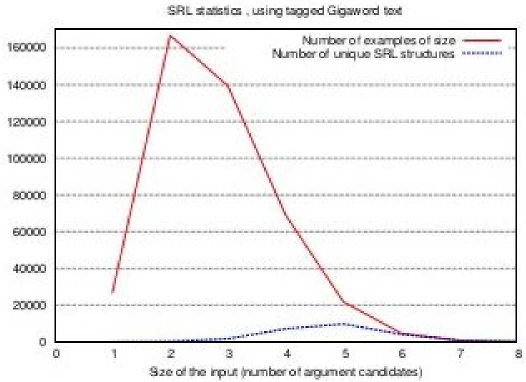
- Sometimes, there are many possible structures for a problem, but only a handful of commonly-seen ones



(a) Part-of-speech tagging



(b) Unlabeled dependency parsing



(c) Semantic role labeling

Background and Motivation

If we solve a lot of inference problems, can we sometimes reuse the results for new problems instead of starting from scratch?

Main theorems

Main contributions of paper

- 3 "exact" theorems
    - If conditions hold, the solution to an older ILP problem is necessarily also the optimal solution to a new ILP problem
- Approximations for the theorems
- Experimental results for Semantic Role Labeling (SRL)
    - Makes ½ the number of ILP calls while getting exact optimal solutions
    - Makes ⅓ the number of ILP calls when using approximations, with minimal loss of accuracy

Review of ILP

- For the purposes of this paper, we restrict ourselves to 0-1 ILP
    - Given some linear inequalities...
    - … find a solution that maximizes some linear objective function...
    - … with each component of the solution either 0 or 1
- ILP (unlike the non-integer version) is NP-hard

$$\arg\max c \cdot y$$
$$\forall i \sum_j A_{ij} y_j \leq b_j$$
$$\forall j, y_j \in \{0, 1\}$$

Equivalence classes of ILPs

- Two ILPs are in the same equivalence class if they have:
  a. the same **number of inference variables**
  b. the same **feasible set**

Equivalence classes of ILPs

- Two ILPs are in the same equivalence class if they have:
  a. the same **number of inference variables**
  b. the same **feasible set**

$$\begin{gathered} \arg\max 5y_1 + 3y_2 + 7y_3 \\ 2y_1 + 4y_2 + 6y_3 \leq 9 \\ 4y_1 + 8y_2 + y_3 \leq 7 \\ y_1, y_2, y_3 \in \{0, 1\} \end{gathered}$$

$$\begin{gathered} \arg\max \mathbf{12}y_1 + \mathbf{3}y_2 + \mathbf{2}y_3 \\ 2y_1 + 4y_2 + 6y_3 \leq 9 \\ 4y_1 + 8y_2 + y_3 \leq 7 \\ y_1, y_2, y_3 \in \{0, 1\} \end{gathered}$$

Theorem 1

- Increasing weights on "active" variables (variables that are set to 1) or decreasing weights on "inactive" variables (variables that are set to 0) doesn't change the optimal solution

$$\arg\max c \cdot y$$
$$\mathrm{c} = (1,\ 2,\ 2)$$
$$\mathrm{y} = (1,\ 0,\ 1)$$

Theorem 1

- Increasing weights on "active" variables (variables that are set to 1) or decreasing weights on "inactive" variables (variables that are set to 0) doesn't change the optimal solution

$$\arg\max c \cdot y$$
$$\mathrm{c} = (\mathbf{4},\ 2,\ 2)$$
$$\mathrm{y} = (1,\ 0,\ 1)$$

Theorem 1

- Increasing weights on "active" variables (variables that are set to 1) or decreasing weights on "inactive" variables (variables that are set to 0) doesn't change the optimal solution

$$\arg\max c \cdot y$$

$$\mathrm{c} = (4, \mathbf{-1}, 2)$$

$$\mathrm{y} = (1, 0, 1)$$

Proof of Theorem 1

- Let c be the original weights, with y* the optimal solution for the original problem. WLOG, assume c' increases the first component of c by k, and that the first component of y* is 1.

$$
\begin{aligned}
c'y^* &= cy^* + (c' - c)y^* \\
&= cy^* + ky_1^* \\
&= cy^* + k \\
&\geq cy + ky_1 \\
&= cy + (c' - c)y = cy
\end{aligned}
$$

Proof of Theorem 1

- For the "inactive" case, the argument is similar
- Apply both cases repeatedly to all components of c

Theorem 1

- Another way of putting it: take the difference between weights, compare positive and negatives of the difference to the solution found

$$\mathrm{c}_1 = (1, 2, 2)$$
$$\mathrm{c}_2 = (4, -1, 2)$$
$$\delta c = c_2 - c_1 = (3, -3, 0)$$
$$\mathrm{y} = (1,\ 0,\ 1)$$

Theorem 1

Theorem 1. *Let* $\mathbf{p}$ *denote an inference problem posed as an integer linear program belonging to an equivalence class* $[P]$. *Let* $\mathbf{q} \sim [P]$ *be another inference instance in the same equivalence class. Define* $\delta\mathbf{c} = \mathbf{c_q} - \mathbf{c_p}$ *to be the difference of the objective coefficients of the ILPs. Then,* $\mathbf{y_p}$ *is the solution of the problem* $\mathbf{q}$ *if for each* $i \in \{1, \cdots, n_{\mathbf{p}}\}$*, we have*

$$(2\mathbf{y}_{\mathbf{p},i} - 1)\delta\mathbf{c}_i \geq 0 \tag{3}$$

Theorem 2

- If a solution works for two different vectors of weights, it works for a (nonnegative) linear combination of them too.

$$
\begin{gathered}
\mathrm{y}^* = \arg\max c_1 \cdot y \\
\mathrm{y}^* = \arg\max c_2 \cdot y \\
\Downarrow \\
\mathrm{y}^* = \arg\max (5c_1 + 7c_2) \cdot y
\end{gathered}
$$

Theorem 2

Theorem 2. *Let* P *denote a collection* $\{\mathbf{p}^1, \mathbf{p}^2, \cdots, \mathbf{p}^m\}$ *of* m *inference problems in the same equivalence class* $[P]$ *and suppose that all the problems have the same solution,* $\mathbf{y_p}$. *Let* $\mathbf{q} \sim [P]$ *be a new inference program whose optimal solution is* $\mathbf{y}$. *Then* $\mathbf{y} = \mathbf{y_p}$ *if there is some* $\mathbf{x} \in \Re^m$ *such that* $\mathbf{x} \geq \mathbf{0}$ *and*

$$\mathbf{c_q} = \sum_j \mathbf{x}_j \mathbf{c}^j_{\mathbf{p}}. \tag{4}$$

Theorem 3

- Can we combine Theorem 1 and 2?

Theorem 3

1. Look for a linear combination of vectors (Theorem 2) such that…
2. … the difference between this combination and the ILP problem we are solving fulfills Theorem 1

$$\delta c = c - (5c_1 + 7c_2) = (3, -3, 0)$$
$$\mathrm{y} = (1,\ 0,\ 1)$$

Theorem 3

Theorem 3. *Let* P *denote a collection* $\{\mathbf{p}^1, \mathbf{p}^2, \cdots, \mathbf{p}^m\}$ *of* m *inference problems belonging to the same equivalence class* $[P]$. *Furthermore, suppose all the programs have the same solution* $\mathbf{y_p}$. *Let* $\mathbf{q} \sim [P]$ *be a new inference program in the equivalence class. For any* $\mathbf{x} \in \Re^m$, *define* $\Delta\mathbf{c}(\mathbf{x}) = \mathbf{c_q} - \sum_j \mathbf{x}_j \mathbf{c}^j_{\mathbf{p}}$. *The assignment* $\mathbf{y_p}$ *is the optimal solution of the problem* $\mathbf{q}$ *if there is some* $\mathbf{x} \in \Re^m$ *such that* $\mathbf{x} \geq \mathbf{0}$ *and for each* $i \in \{1, n_{\mathbf{p}}\}$, *we have*

$$(2\mathbf{y}_{\mathbf{p},i} - 1)\Delta\mathbf{c}_i \geq 0 \tag{5}$$

Implementation

- Theorem 1: loop through all previously solved problems in the same equivalence class, check weights to see if theorem fulfilled
- Theorem 2 and 3: solve a linear (non-integer!) program for combining the existing ILP instances
    - Can also optimize further by only selectively including ILP instances

Approximations

- Top-1/Top-K (approximation "baseline")
  - Since many similar instances tend to have the same structure, just cache the most frequently seen instances for each equivalence class, and use those as our guesses

Approximations

- Approximate Theorem 1/Theorem 3
    - Allow inequalities to be violated by some epsilon
    - Can reuse solutions more often, even if not necessarily the optimal solution

$$(2\mathbf{y} - 1)\ \delta \boldsymbol{c} \geq 0$$
$$\Downarrow$$
$$(2\mathrm{y} - 1)\ \delta \boldsymbol{c} \geq -\boldsymbol{\epsilon}$$

Experimental results

Recap of Semantic Role Labeling

- From (Punyakanok et al., 2008)
- Given a **sentence** and a **verb**, label the corresponding **arguments** of the verb:

[A0 I] [V *left*] [A1 my pearls] [A2 to my daughter-in-law] [AM-LOC in my will].

Experimental results

Type	Algorithm	# instances	# solver calls	Speedup	Clock speedup	F1
Exact	Baseline	5127	5217	1.0	1.0	75.85
Exact	Theorem 1	5127	2134	2.44	1.54	75.90
Exact	Theorem 2	5127	2390	2.18	1.14	75.79
Exact	Theorem 3	5127	2089	2.50	1.36	75.77
Approx.	Most frequent (Support = 50)	5127	2812	1.86	1.57	62.00
Approx.	Top-10 solutions (Support = 50)	5127	2812	1.86	1.58	70.06
Approx.	Theorem 1 (approx, $\epsilon = 0.3$)	5127	1634	3.19	1.81	75.76
Approx.	Theorem 3 (approx, $\epsilon = 0.3$)	5127	1607	3.25	1.50	75.46

Extensions

- Kundu, Srikumar, Roth (2013)
    - Margin based generalization of Theorem 1
    - Also decompose each inference problem into parts, try to use technique on smaller subproblems rather than on large problems
    - Further improvements: e.g. only makes 16% of inference calls (vs 41%) in Semantic Role Labeling
- Chang, Upadhyay, Kundu, Roth (2015)
    - Another extension of Theorem 1
    - Further improvements to learning using amortized inference
    - Only makes 10%-24% of inference calls in Entity Relation Extraction

References

V. Srikumar, G. Kundu, and D. Roth. 2012. On amortizing inference cost for structured prediction. In *EMNLP*.

V. Punyakanok, D. Roth, and W. Yih. 2008. The importance of syntactic parsing and inference in semantic role labeling. *Computational Linguistics.*

Questions?