Learning and Inference in Structured Prediction
Tutorial AAAI

Gourab Kundu
IBM Research
Part 3: Amortized Inference

- Overview
- Amortization at Inference Time:
  - Theorems
  - Decomposition
  - Results
- Amortization during Learning:
  - Approximate Inference
  - Results
Inference

After inferring the POS structure for S1, Can we speed up inference for S2? Can we make the k-th inference problem cheaper than the first?

<table>
<thead>
<tr>
<th><strong>S1</strong></th>
<th><strong>S2</strong></th>
<th><strong>POS</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>He</td>
<td>They</td>
<td>PRP</td>
</tr>
<tr>
<td>is</td>
<td>are</td>
<td>VBZ</td>
</tr>
<tr>
<td>reading</td>
<td>watching</td>
<td>VBG</td>
</tr>
<tr>
<td>a</td>
<td>a</td>
<td>DT</td>
</tr>
<tr>
<td>book</td>
<td>movie</td>
<td>NN</td>
</tr>
</tbody>
</table>

S1 & S2 look very different but their output structures are the same

The inference outcomes are the same
**Amortized Inference** [Kundu, Srikumar & Roth, EMNLP-12,ACL-13]

- We formulate the problem of **amortized inference**: reducing inference time over the **lifetime** of an NLP tool.

- We develop conditions under which the solution of a new, previously unseen problem, can be **exactly inferred** from earlier solutions **without invoking a solver**.

- **This results in a family of exact** inference schemes
  - Algorithms are **invariant** to the underlying solver; we simply reduce the **number of calls to the solver**.

- Significant improvements both in terms of **solver calls** and **wall clock time** in several structured prediction tasks.
The goal is to find a consistent assignment of entity types to all entities and relation types to all relations.

- Consistency constraint: A spouse relation can only hold between two person entities and cannot hold between two location entities.
Dole’s wife, Elizabeth, is a native of Champaign, Illinois.
ILP Formulation

<table>
<thead>
<tr>
<th></th>
<th>Dole</th>
<th>Elizabeth</th>
</tr>
</thead>
<tbody>
<tr>
<td>PER</td>
<td>0.5 y1</td>
<td>PER 0.6 y4</td>
</tr>
<tr>
<td>LOC</td>
<td>0.3 y2</td>
<td>LOC 0.1 y5</td>
</tr>
<tr>
<td>ORG</td>
<td>0.2 y3</td>
<td>ORG 0.3 y6</td>
</tr>
</tbody>
</table>

Dole-Elizabeth

<table>
<thead>
<tr>
<th>Relation</th>
<th>Weight</th>
<th>y_i</th>
</tr>
</thead>
<tbody>
<tr>
<td>spouse</td>
<td>0.7</td>
<td>y7</td>
</tr>
<tr>
<td>born_in</td>
<td>0.1</td>
<td>y8</td>
</tr>
<tr>
<td>Located_at</td>
<td>0.1</td>
<td>y9</td>
</tr>
<tr>
<td>No-relation</td>
<td>0.1</td>
<td>y10</td>
</tr>
</tbody>
</table>

maximize

0.5y1 + 0.3y2 + 0.2y3 + 0.6y4 + 0.1y5 + 0.3y6 + 0.7y7 + 0.1y8 + 0.1y9 + 0.1y10

subj to

yi ∈ {0,1}

y1 + y2 + y3 = 1

y4 + y5 + y6 = 1

y7 + y8 + y9 + y10 = 1

2y7 - y1 - y4 ≤ 0

A spouse relation can only hold between two person entities
Amortized Inference for ILP

- We can write the ILP as

  $$\text{arg max}_y \ c y$$

  $$A y \leq b$$

  $$y_i \in \{0, 1\}$$

- Inference problems discussed in previous sections can be represented as 0-1 ILPs.

<table>
<thead>
<tr>
<th>Objective</th>
<th>Constraints</th>
</tr>
</thead>
<tbody>
<tr>
<td>$0.5y_1 + 0.3y_2 + 0.2y_3 + 0.6y_4 + 0.1y_5 + 0.3y_6 + 0.7y_7 + 0.1y_8 + 0.1y_9 + 0.1y_{10}$</td>
<td>$y_1 + y_2 + y_3 = 1$</td>
</tr>
<tr>
<td></td>
<td>$y_4 + y_5 + y_6 = 1$</td>
</tr>
<tr>
<td></td>
<td>$y_7 + y_8 + y_9 + y_{10} = 1$</td>
</tr>
<tr>
<td></td>
<td>$2y_7 - y_1 - y_4 \leq 0$</td>
</tr>
</tbody>
</table>
Preliminary (1)

max \(2y_1 + 3y_2 + 2y_3 + 1.0y_4\)
\(y_1 + y_2 \leq 1\)
\(y_3 + y_4 \leq 1\)

objective vector

\(c_p = \begin{pmatrix} 2 \\ 3 \\ 2 \\ 1 \end{pmatrix}\)
max $2y_1 + 3y_2 + 2y_3 + 1.0y_4$

$y_1 + y_2 \leq 1$

$y_3 + y_4 \leq 1$

objective vector

$\mathbf{c}_P = \begin{bmatrix} 2 \\ 3 \\ 2 \\ 1 \end{bmatrix}$

optimal solution

$\mathbf{y}_P^* = \begin{bmatrix} 0 \\ 1 \\ 1 \\ 0 \end{bmatrix}$
Preliminary (3)

\[
\text{max } 2y_1 + 3y_2 + 2y_3 + 1.0y_4 \\
y_1 + y_2 \leq 1 \\
y_3 + y_4 \leq 1
\]

objective vector

\[
c_P = 2 \\
3 \\
2 \\
1
\]

optimal solution

\[
y^*_P = 0 \\
1 \\
1 \\
0
\]

score for optimal solution

\[
c_P \cdot y^*_P = 5
\]
Preliminary (4)

We define an **equivalence class** as the set of ILPs that have:
- the same number of inference variables
- the same feasible set
  (same constraints modulo renaming)

P: \[
\begin{align*}
\text{max } & 2y_1 + 3y_2 + 2y_3 + y_4 \\
y_1 + y_2 & \leq 1 \\
y_3 + y_4 & \leq 1
\end{align*}
\]

Q: \[
\begin{align*}
\text{max } & 2y_1 + 4y_2 + 2y_3 + 0.5y_4 \\
y_1 + y_2 & \leq 1 \\
y_3 + y_4 & \leq 1
\end{align*}
\]

Constraints are same

Same equivalence class

# of variables = 4
Recap: The Recipe

Given:

- A cache of solved ILPs and a new problem

If \text{THEOREM\_SATISFIED}(\text{cache, new problem})
then
\text{SOLUTION}(\text{new problem}) = \text{old solution}
Else
    Call \text{base solver} and update \text{cache}
End

- We will show four different theorems.
Part 3: Amortized Inference

- Overview

- Amortization at Inference Time:
  - Theorems
  - Decomposition

- Amortization during Learning:
  - Approximate Inference
  - Results
Intuition of Theorem I

Two ILPs with same constraints, but different objective coefficients

The objective coefficients of two ILPs

Coefficients for variables that are 1 have not decreased

Coefficients for variables that are 0 have not increased

P and Q have the same solution

Solution to problem P
Theorem I

- Denote: \( \delta c = c_Q - c_P \)

\[
\begin{align*}
 y_{p,i}^* = 0 & \quad \Rightarrow \quad c_{Q,i} \leq c_{P,i} & \quad \Rightarrow \quad \delta c_i \leq 0 & \quad \Rightarrow \quad (2y_{p,i}^* - 1)\delta c_i \geq 0 \\
 y_{p,i}^* = 1 & \quad \Rightarrow \quad c_{Q,i} \geq c_{P,i} & \quad \Rightarrow \quad \delta c_i \geq 0 & \quad \Rightarrow \quad (2y_{p,i}^* - 1)\delta c_i \geq 0
\end{align*}
\]
Full Statement of Theorem I

Theorem:

- Let $y^*_P$ be the optimal solution of an ILP $P$. Assume that an ILP $Q$
  - Is in the same equivalence class as $P$
  - And, For each $i \in \{1, ..., n_p\}$ $(2y^*_p,i - 1)\delta c_i \geq 0$,
    where $\delta c = c_Q - c_P$
- Then, without solving $Q$, we can guarantee that the optimal solution of $Q$ is $y^*_Q = y^*_P$
Intuition of Theorem II (Geometric Interpretation)

All ILPs in the cone will share the maximizer for this feasible region.
Formal Statement of Theorem II

Theorem:

- Assume we have seen \( m \) ILP problems \( \{P_1, P_2, ..., P_m\} \)
  - All are in the same equivalence class
  - All have the same optimal solution
- Let ILP \( Q \) be a new problem s.t.
  - \( Q \) is in the same equivalence class as \( P_1, P_2, ..., P_m \)
  - There exists an \( z \geq 0 \) such that \( c_Q = \sum z_i c_{P_i} \)

Then, without solving \( Q \), we can guarantee that the optimal solution of \( Q \) is \( y^*_Q = y^*_P \)
Proof of Theorem II

Let $y^*$ be the optimal solution of both $P_1$ and $P_2$

- $c_{P_1} \cdot y^* \geq c_{P_1} \cdot y'$ and $c_{P_2} \cdot y^* \geq c_{P_2} \cdot y'$
- $(z_1c_{P_1} + z_2c_{P_2}) \cdot y^* \geq (z_1c_{P_1} + z_2c_{P_2}) \cdot y'$ if $z_1, z_2 \geq 0$

$y^*$ is optimal for any ILP with objective $(z_1c_{P_1} + z_2c_{P_2})$ with $z_1, z_2 \geq 0$ and same constraint set.
Formal Statement of Theorem III

Theorem:

- Assume we have seen $m$ ILP problems \{$P_1, P_2, ..., P_m$\}.
  - All are in the same equivalence class.
  - All have the same optimal solution.
- Let ILP $Q$ be a new problem s.t.
  - $Q$ is in the same equivalence class as $P_1, P_2, ..., P_m$.
  - There exists an $z \geq 0$ such that $\delta c = c_Q - \sum z_i c_{P_i}$ and $(2y^*_{P,i} - 1) \delta c_i \geq 0$.
- Then, without solving $Q$, we can guarantee that the optimal solution of $Q$ is $y^*_Q = y^*_{P_i}$. 

Page 21
Proof of Theorem III

- Let $y^*$ be the optimal solution of both P1 and P2

- Theorem II: P1, P2 => R
  - if $c_R = z_1 c_{P1} + z_2 c_{P2}$, $z_1, z_2 \geq 0$ => $y^*_R = y^*_P1 = y^*_P2$

- Theorem I: R => Q
  - if $(2y^*_{R,i} - 1)\delta c_i \geq 0$, $\delta c = c_Q - c_R$ => $y^*_Q = y^*_R$
  - if $(2y^*_{P1,i} - 1)\delta c_i \geq 0$, $\delta c = c_Q - \sum z_i c_{Pi}$ => $y^*_Q = y^*_P1$
Theorem IV

Decrease in objective value of the solution
\[ A = (C_p - C_Q) y^* \]

Increase in objective value of the competing structures
\[ B = (C_Q - C_p) y \]

Theorem (margin based amortized inference): If \( A + B \) is less than the structured margin, then \( y^* \) is still the optimum for problem P.

Objective values for problem P

Structured Margin \( \delta \)

Increasing objective value

Two competing structures

\( y^* \) the solution to problem P
Formally

Let \( y^* \) be optimal for P with structured margin \( \delta \)

- \( c_p \cdot y^* \geq c_p \cdot y + \delta \) for all \( y \), \( A_1 y \leq b_1, A_2 y \leq b_2 \)

Objective increase for \( y \) from P to Q is \((c_Q - c_P) \cdot y\)

Objective decrease for \( y^* \) from P to Q is \((c_P - c_Q) \cdot y^*\)

\( y^* \) is optimal for Q if

- \((c_Q - c_P) \cdot y + (c_P - c_Q) \cdot y^* \leq \delta \) for all \( y \), \( A_1 y \leq b_1, A_2 y \leq b_2 \)
- \((c_Q - c_P) \cdot y + (c_P - c_Q) \cdot y^* \leq \delta \) for all \( y \), \( A_1 y \leq b_1 \)
Amortized Inference Experiments

Setup

- Verb semantic role labeling
  - Other results also at the end of the section
- Speedup & Accuracy are measured over WSJ test set (Section 23)
- Baseline is solving ILP using Gurobi solver.

For amortization

- Cache 250,000 SRL inference problems from Gigaword
- For each problem in test set, invoke an amortized inference algorithm
Speedup & Accuracy

Speedup = \frac{\text{number of inference calls without amortization}}{\text{number of inference calls with amortization}}

Amortized inference gives a speedup without losing accuracy.

Solve only one in three problems.

Amortization schemes [EMNLP’12, ACL’13]
Amortized Inference

Part 3: Amortized Inference

- Overview

- Amortization at Inference Time:
  - Theorems
  - Decomposition
  - Results

- Amortization during Learning:
  - Approximate Inference
  - Results
So far...

- Amortized inference
  - Making inference faster by re-using previous computations
- Techniques for amortized inference
- But these are not useful if the full structure is not redundant!
Decomposed amortized inference

- Taking advantage of redundancy in components of structures
  - Extend amortization techniques to cases where the full structured output may not be repeated
  - Store partial computations of “components” for use in future inference problems
Entity Relation Extraction task

Dole’s wife, Elizabeth, is a native of Champaign, Illinois.  

Person  Person  Location  Location
  spouse    born_in      Located_at

maximize  
0.5y1 + 0.3y2 + 0.2y3 +  
0.6y4 + 0.1y5 + 0.3y6 +  
0.7y7 + 0.1y8 + 0.1y9 + 0.1y10

subj to yi \in \{0,1\}  
y1 + y2 + y3 = 1  
y4 + y5 + y6 = 1  
y7 + y8 + y9 + y10 = 1  
2y7 - y1 - y4 \leq 0  
+ additional constraints
Decomposed inference for ER task

Dole’s wife, Elizabeth, is a native of Champaign, Illinois.

Person  
spouse

Person  
born_in

Location  
Located_at

Consistent assignment of entity types to first two entities (Dole, Elizabeth) and relation types to relations among these entities

Consistent assignment of entity types to last two entities (Champaign, Illinois) and relation types to relations among these entities

Re-introduce constraints using Lagrangian Relaxation

Rush & Collins, A Tutorial on Dual Decomposition and Lagrangian Relaxation for Inference in Natural Language Processing, JAIR, 2011.
Amortized Inference

Part 3: Amortized Inference

- Overview

- Amortization at Inference Time:
  - Theorems
  - Decomposition
  - Results

- Amortization during Learning:
  - Approximate Inference
  - Results
Amortized inference gives a speedup without losing accuracy.
Reduction in inference calls (SRL)

- **Inference Engine**
  - Num. inference calls: 100

- **Theorem 1**
  - Num. inference calls: 41
  - +decomposition: 24.4

- **Margin based inference**
  - Num. inference calls: 32.7
  - +decomposition: 16.6

Solve only one in six problems
Reduction in inference calls (Entity-relation extraction)

- Num. inference calls
- +decomposition

<table>
<thead>
<tr>
<th></th>
<th>Infrence Engine</th>
<th>Theorem 1</th>
<th>Margin based inference</th>
</tr>
</thead>
<tbody>
<tr>
<td>Num. inference calls</td>
<td>100</td>
<td>59.5</td>
<td>28.2</td>
</tr>
<tr>
<td>+decomposition</td>
<td></td>
<td>57</td>
<td>24.4</td>
</tr>
</tbody>
</table>

Solve only one in four problems
So far...

- We have given theorems that allow **savings of 5/6 of the calls** to your favorite inference engine.

- But, there is some cost in
  - Checking the conditions of the theorems
  - Accessing the cache

- Our implementations are clearly not state-of-the-art but....
Reduction in wall-clock time (SRL)

- ILP Solver
- Theorem 1
- Margin based inference

- Num. inference calls
- +decomposition

Solve only one in 2.6 problems

- ILP Solver: 100
- Theorem 1: 54.8, 40
- Margin based inference: 45.9, 38.1
Part 3: Amortized Inference

- Overview
- Amortization at Inference Time:
  - Theorems
  - Decomposition
  - Results
- Amortization during Learning:
  - Approximate Inference
  - Results
Redundancy in Learning Phase

- [AAAI 15]: Structural Learning with Amortized Inference

![Graph showing the relationship between the number of training rounds and the counts of inference problems and distinct solutions.](image)
Amortization during Learning w/ Theorem I

- We can apply Theorem I to amortize inference calls during learning.

- Recall: Condition of Theorem 1:
  - For each $i \in \{1, \ldots, n_p\}$ \((2y^*_{p,i} - 1)\delta c_i \geq 0\), where $\delta c = c_Q - c_p$

- Guarantee of exactness: $y^*_Q = y^*_p$
Amortization during Learning w/ Approximate Solution

- Approximate solutions to inference problems can be good enough to guide learning.

- New Condition:
  - For each $i \in \{1, ..., n_p \}$ \((2y^*_p,i - 1)\delta c_i \geq -\varepsilon |c_{Q,i}|\), where \(\delta c = c_Q - c_p\)

- Guarantee of Approximation
  - \(y^*_p\) is a \(1 / (1 + M \varepsilon)\) approximate solution to Q.
Learning with Approximate Amortized Inference

- Learning Structured SVM with approximate amortized inference gives a model with **bounded empirical risk**
  - Finley, T., and Joachims, T. 2008. *Training structural SVMs when exact inference is intractable*. In ICML 2008
  - our formulation is an under-generating approximation with approximation ratio \( \frac{1}{1 + M\varepsilon} \)

- Dual coordinate descent for structured SVM can still return **an exact model** even if approx. amortized inference is used.
  - call exact inference after every \( \tau \) iterations
Amortized Inference

Part 3: Amortized Inference

- Overview
- Amortization at Inference Time:
  - Theorems
  - Decomposition
  - Results

- Amortization during Learning:
  - Approximate Inference
  - Results
# Solver Calls (Entity-Relation Extraction)

<table>
<thead>
<tr>
<th>% Solver Calls</th>
<th>Baseline</th>
<th>Our</th>
<th>Our-approx.</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>20</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>40</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>60</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>80</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>100</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Exact

Ent F1: 87.7
Rel F1: 47.6

Better

Ent F1: 87.3
Rel F1: 47.8
Part 3: Amortized Inference

- Amortization at Inference Time:
  - Theorems
  - Decomposition
  - Results

- Amortization during Learning:
  - Approximate Inference
  - Results