Constraints as Prior Knowledge

Ming-Wei Chang, Lev Ratinov, Dan Roth
Department of Computer Science
University of Illinois at Urbana-Champaign

July 2008
ICML Workshop
on
Prior Knowledge for Text and Language
Tasks of Interest

- Global decisions in which several local decisions play a role but there are mutual dependencies on their outcome.
  - E.g. Structured Output Problems – multiple dependent output variables

- (Learned) models/classifiers for different sub-problems
  - In some cases, not all models are available to be learned simultaneously
  - Key examples in NLP are Textual Entailment and QA
  - In these cases, constraints may appear only at evaluation time

- Incorporate models’ information, along with prior knowledge/constraints, in making coherent decisions
  - decisions that respect the learned models as well as domain & context specific knowledge/constraints.
Task of Interests: Structured Output

- For each instance, assign values to a set of variables
- Output variables depend on each other

- **Common** tasks in
  - Natural language processing
    - Parsing; Semantic Parsing; Summarization; co-reference,…
  - Information extraction
    - Entities, Relations,…

- **Many pure** machine learning approaches exist
  - Hidden Markov Model (HMM)
  - Perceptrons…

- However, …
Information Extraction via Hidden Markov Models


Prediction result of a trained HMM

Unsatisfactory results!
Strategies for Improving the Results

- *Pure* Machine Learning Approaches
  - Higher Order HMM?
  - Increasing the window size?
  - Adding a lot of new features
    - Requires a lot of labeled examples
  - What if we only have a few labeled examples?

  Increasing the model complexity

- Any other options?
  - Humans can *immediately* tell bad outputs
  - The output *does not make sense*

Can we keep the learned model simple and still make expressive decisions?
Information extraction without Prior Knowledge


Prediction result of a trained HMM

[AUTHOR] Lars Ole Andersen
[TITLE] Program analysis and specialization for the C Programming language
[EDITOR] 
[BOOKTITLE] 
[TECH-REPORT] PhD thesis
[INSTITUTION] DIKU, University of Copenhagen
[DATE] May 1994

Violates lots of natural constraints!
Examples of Constraints

- Each field must be a **consecutive list of words and can appear at most once in a citation**.

- State transitions must occur on **punctuation marks**.

- The citation can only start with **AUTHOR or EDITOR**.

- The words *pp.*, *pages* correspond to **PAGE**.

- Four digits starting with *20xx* and *19xx* are **DATE**.

- **Quotations** can appear only in **TITLE**.

- Easy to express pieces of “knowledge”

- Non Propositional; May use Quantifiers
Adding constraints, we get **correct** results!

- **Without** changing the model

- **[AUTHOR]** Lars Ole Andersen
- **[TITLE]** Program analysis and specialization for the C Programming language
- **[TECH-REPORT]** PhD thesis
- **[INSTITUTION]** DIKU, University of Copenhagen
- **[DATE]** May, 1994
This Talk

- Present Constrained Conditional Models
  - A general framework that combines
    - Learning models and using expressive constraints
    - Within a constrained optimization framework
  - Has been shown useful in the context of many NLP problems
    - SRL, Summarization; Co-reference; Information Extraction
    - [Roth&Yih04,07; Punyakanok et al. 05,08; Chang et al. 07,08; Clarke&Lapata06,07; Denise&Baldridge07]

- Here: focus on semi-supervised learning scenarios
  - Result: 20 labeled ex’s + constraints is competitive with 300 labeled ex’s
- Investigate ways for training models and combining constraints
  - Joint Learning and Inference vs. decoupling Learning & Inference
    - Learning constraints’ weight
  - Training Discriminatively vs. ML
Outline

- Constrained Conditional Model
  - Feature vs Constraints
  - Inference
  - Training
  - Semi-supervised Learning

- Results

- Discussion
Constrained Conditional Models

\[ f_{\Phi, C}(x, y) = \sum w_i \phi_i(x, y) \]

Weight Vector for “local” models

A collection of local features

\[ y^* = \arg\max_y f_{\Phi, C}(x, y) \]

- How to solve?
  - This is an Integer linear Programming Problems
  - Use ILP packages or search techniques

- How to train?
  - How to decompose global objective function?
  - Should we incorporate constraints in the learning process?

Penalty for violating the constraint
Features Versus Constraints

- In principle, constraints and features can encode the same properties.
- In practice, they are very different.

**Features**

- Local, short distance properties – to support tractable inference.
- Propositional (grounded):
  - E.g. True if “the followed by a Noun occurs in the sentences”.

**Constraints**

- Global properties
- Quantified, first order logic expressions
- E.g. True iff “all y_i's in the sequence y are assigned different values.”
Encoding Prior Knowledge

- Consider encoding the knowledge that:
  - Entities of type A and B cannot occur simultaneously in a sentence

- The “Feature” Way
  - Results in higher order HMM, CRF
  - May require designing a model tailored to knowledge/constraints
  - Large number of new features: might require more labeled data
  - Waste parameters to learn indirectly knowledge we have.

- The Constraints Way
  - Keep the model simple; add expressive constraints directly
  - A small set of constraints
  - Allows for decision time incorporation of constraints
Constraints and Inference

- Degree of constraint violation is modeled as:

\[ f_{\Phi,C}(x, y) = \sum w_i \phi_i(x, y) - \sum \rho_i d_{C_i}(x, y). \]

- Compute an estimated distance from partial assignments.
- Bias the search to the right solution space as early as possible.

Solvers

- This work: Beam search
- A* with admissible heuristics
- Earlier works: Integer Linear Programming
Outline

- Constrained Conditional Model
  - Feature v.s Constraints
  - Inference
  - Training
    - Semi-supervised Learning

- Results

- Discussion
Training Strategies

- Hard Constraints or Weighted Constraints
  - Hard constraints: set penalties to \textit{infinity}
    - No more degrees of violation
  - Weighted Constraints
    - Need to figure out penalties values

- Factored / Jointed Approaches
  - Factored Models (L+I)
    - Learn model weights and constraints’ penalties separately
  - Joint Models (IBT)
    - Learn the model weights and constraints’ penalties \textit{jointly}
  - L+I vs IBT: [Punyakanok et. al. 05]

Training Algorithms:
- L+ CI, L+ wCI
- CIBT, wCIBT
Factored (L+I) Approaches

- Learning model weights
  - HMM

- Constraints Penalties
  - Hard Constraints: infinity
  - Weighted Constraints:
    - $\hat{\lambda} = -\log P\{\text{Constraint } C_i \text{ is violated in training data}\}$
Joint Approaches

**Algorithm 1** IBT training: CIBT & wCIBT

**Require:** $D$ is the training dataset, $K$ is the number of constraints, $M$ is the number of iterations

```plaintext
1: for $i = 1 \ldots K$ do
2:   if (hardConstraints) then $\rho_i = \infty$ else $\rho_i = 0$
3:   end for
4: for $i = 1 \ldots M$ do
5:   for $(x, y^*) \in D$ do
6:     $\hat{y} = \arg\max_y [\sum w_i \phi_i(x, y) - \sum \rho_i d_{C_i}(x, y)]$
7:   end for
8:   $w = w + \Phi(x, y^*) - \Phi(x, \hat{y})$
9:   if weightedConstraints then
10:      $\rho = \rho + d_C(x, y^*) - d_C(x, \hat{y})$
11:   end if
12: end for
```

Structured Perceptron
Outline

- Constrained Conditional Model
  - Feature v.s Constraints
  - Inference
  - Training
  - Semi-supervised Learning

- Results

- Discussion
Semi-supervised Learning with Constraints

\[ \lambda = \text{learn}(T) \]

For N iterations do

\[ T = \emptyset \]

For each \( x \) in unlabeled dataset

\[ \{y_1, \ldots, y_K\} \leftarrow \text{InferenceWithConstraints}(x, C, \lambda) \]

\[ T = T \cup \{(x, y_i)\}_{i=1}^k \]

\[ \lambda = \gamma \lambda + (1-\gamma) \text{learn}(T) \]

Supervised learning algorithm parameterized by \( \lambda \)

Inference based augmentation of the training set (feedback) (inference with constraints).

Learn from new training data. Weigh supervised and unsupervised model.

[Chang, Ratinov, Roth, ACL'07]
Outline

- Constrained Conditional Model
  - Feature vs. Constraints
  - Inference
  - Training
  - Semi-supervised Learning

Results

- Discussion
Results on Factored Model -- Citations

In all cases: semi = 1000 unlabeled examples.

In all cases: Significantly better results than existing results [Chang et. al. ’07]
Results on Factored Model -- Advertisements

![Graph showing accuracy for different models with varying number of training examples.](image-url)
Hard Constraints vs. Weighted Constraints

Constraints are close to perfect

<table>
<thead>
<tr>
<th></th>
<th>Training samples</th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>5</td>
<td>10</td>
<td>20</td>
<td>300</td>
<td></td>
</tr>
<tr>
<td>Weighted constraints</td>
<td>77.09</td>
<td>81.25</td>
<td>85.00</td>
<td>94.51</td>
<td></td>
</tr>
<tr>
<td>Hard constraints</td>
<td>78.18</td>
<td>81.11</td>
<td>85.16</td>
<td>92.80</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>Training samples</th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>5</td>
<td>10</td>
<td>20</td>
<td>300</td>
<td></td>
</tr>
<tr>
<td>Weighted constraints</td>
<td>71.46</td>
<td>75.61</td>
<td>77.76</td>
<td>82.06</td>
<td></td>
</tr>
<tr>
<td>Hard constraints</td>
<td>69.91</td>
<td>73.46</td>
<td>75.25</td>
<td>79.59</td>
<td></td>
</tr>
</tbody>
</table>

Labeled data might not follow the constraints
Factored vs. Jointed Training

- Using the best models for both settings
  - Factored training: HMM + weighted constraints
  - Jointed training: Perceptron + weighted constraints
  - Same feature set

- Without constraints
  - Factored Model is better
  - Few labeled examples, HMM > perceptron
  - Many labeled examples, perceptron > HMM

Agrees with earlier results in the supervised setting ICML’05, IJCAI’05
Value of Constraints in Semi-Supervised Learning

Objective function: \[ f_{\Phi,C}(x, y) = \sum w_i \phi_i(x, y) - \sum \rho_i d_{C_i}(x, y). \]

Constraints are used to Bootstrap a semi-supervised learner. Poor model + constraints used to annotate unlabeled data, which in turn is used to keep training the model.

Learning w/o Constraints: 300 examples.

Learning w/ 10 Constraints

Factored model.

# of available labeled examples
Summary: Constrained Conditional Models

- \( y^* = \arg\max_y \sum w_i \cdot (x; y) \)

- Linear objective functions
- Typically \( \cdot (x,y) \) will be local functions, or \( \cdot (x,y) = \cdot (x) \)

- Clearly, there is a joint probability distribution that represents this mixed model.
- We would like to:
  - Learn a simple model or several simple models
  - Make decisions with respect to a complex model

- Expressive constraints over output variables
- Soft, weighted constraints
- Specified declaratively as FOL formulae
Discussion

- **Adding Expressive Constraints via CCMs**
  - Improves supervised and semi-supervised learning quite a bit
  - Curial when the number of labeled data is small

- **How to use Constraints?**
  - Weighted constraints
  - Factored Training Approaches
  - Other ways?

- **Constraints vs. additional Labeling**
  - What kind of supervision should we get?
    - Adding more annotation?
    - Adding more prior knowledge?
    - Both?
**Conclusion**

- **Constrained Conditional Models combining**
  - Learning models and using expressive constraints
  - Within a constrained optimization framework

- **Use constraints!**
  - The framework support a clean way of incorporating constraints and improving decisions of supervised learning models
    - Significant success on several NLP and IE tasks
  - Here we’ve shown that it can be used successfully as a way to model prior knowledge for semi-supervised learning

- Training protocol matters
Factored v.s. Jointed Training

- Semi-supervised
  - We do not manage to improve Joint approaches through semi-supervised learning