Guiding Semi-Supervision with Constraint-Driven Learning

Ming-Wei Chang\textsuperscript{1} Lev Ratinov\textsuperscript{2} Dan Roth\textsuperscript{3}

\textsuperscript{1}Department of Computer Science
University of Illinois at Urbana-Champaign

Paper presentation by: Drew Stone
Outline

1 Background
   - Semi-supervised learning
   - Constraint-driven learning

2 Constraint-driven learning with semi-supervision
   - Introduction
   - Model
   - Scoring
   - Constraint pipeline
   - Constraint Driven Learning (CODL) algorithm

3 Experiments
1 Background
   - Semi-supervised learning
   - Constraint-driven learning

2 Constraint-driven learning with semi-supervision
   - Introduction
   - Model
   - Scoring
   - Constraint pipeline
   - Constraint Driven Learning (CODL) algorithm

3 Experiments
Semi-supervised learning

Introduction

- **Question**: When labelled data is scarce, how can we take advantage of unlabelled data for training?
- **Intuition**: If there exists structure in the underlying distribution of samples, points close/clustered to one another may share labels.
Semi-supervised learning
Framework

- Given a model $\mathcal{M}$ trained on labelled samples from a distribution $\mathcal{D}$ and an unlabelled set of examples $U \subseteq \mathcal{D}^m$
- Learn labels for each example in $U \rightarrow \text{Learn}(U, \mathcal{M})$ and re-use examples (now labelled) to tune $\mathcal{M} \rightarrow \mathcal{M}^*$

Benefit: Access to more training data
Drawback: Learned model might drift from correct classifier if the assumptions on the distribution do not hold.
Semi-supervised learning

Framework

- Given a model $\mathcal{M}$ trained on labelled samples from a distribution $\mathcal{D}$ and an unlabelled set of examples $U \subseteq \mathcal{D}^m$
- Learn labels for each example in $U \rightarrow \text{Learn}(U, \mathcal{M})$ and re-use examples (now labelled) to tune $\mathcal{M} \rightarrow \mathcal{M}^*$
- **Benefit**: Access to more training data
Semi-supervised learning

Framework

- Given a model $\mathcal{M}$ trained on labelled samples from a distribution $\mathcal{D}$ and an unlabelled set of examples $U \subseteq \mathcal{D}^m$
- Learn labels for each example in $U$ $\rightarrow$ $\textit{Learn}(U, \mathcal{M})$ and re-use examples (now labelled) to tune $\mathcal{M} \rightarrow \mathcal{M}^*$
- **Benefit**: Access to more training data
- **Drawback**: Learned model might drift from correct classifier if the assumptions on the distribution do not hold.
Outline

1 Background
   - Semi-supervised learning
   - Constraint-driven learning

2 Constraint-driven learning with semi-supervision
   - Introduction
   - Model
   - Scoring
   - Constraint pipeline
   - Constraint Driven Learning (CODL) algorithm

3 Experiments
**Motivation**: Keep the learned model simple by using constraints to balance over-simplicity.
**Motivation:** Keep the learned model simple by using constraints to balance over-simplicity

**Benefits:** Simple models (less features) are more computationally efficient
Motivation: Keep the learned model simple by using constraints to balance over-simplicity

Benefits: Simple models (less features) are more computationally efficient

Intuition: Fix a set of task-specific constraints to enable the use of a simple machine learning model but encode task-specific constraints to make both learning easier and more correct.
Given an objective function

$$\arg\max_y \lambda \cdot F(x, y)$$

Define the set of linear (non-linear) constraints \( \{ C_i \}_{i \leq k} \)

$$C_i : \mathcal{X} \times \mathcal{Y} \rightarrow \{0, 1\}$$

Solve the optimization problem given the constraints
Given an objective function

\[
\text{argmax}_y \lambda \cdot F(x, y)
\]

Define the set of linear (non-linear) constraints \( \{ C_i \}_{i \leq k} \)

\[
C_i : \mathcal{X} \times \mathcal{Y} \rightarrow \{0, 1\}
\]

Solve the optimization problem given the constraints

Any Boolean rule can be encoded as a set of linear inequalities.
Constraint-driven learning

Example task

- Sequence tagging with HMM/CRF + global constraints

\[ y^* = \arg\max_{y \in \mathcal{Y}} P(y_0) P(x_0|y_0) \prod_{i=1}^{n-1} P(y_i|y_{i-1}) P(x_i|y_i) \]

Example: the man saw the dog

Every assignment to the y’s is a path.
Constraint-driven learning

Sequence tagging constraints

- Unique label for each word
- Edges must for a path
- A verb must exist
Constraint-driven learning

Example task

- Semantic role labelling with independent classifiers + global constraints

I left my pearls to my daughter in my will.
Each verb predicate carries with it a new inference problem

I left my pearls to my daughter in my will.
Constraint-driven learning

Example task

I left my pearls to my daughter in my will.
Constraint-driven learning

SRL constraints

- No duplicate argument classes
- Unique labels
- Order constraints
1 Background
   - Semi-supervised learning
   - Constraint-driven learning

2 Constraint-driven learning with semi-supervision
   - Introduction
   - Model
   - Scoring
   - Constraint pipeline
   - Constraint Driven Learning (CODL) algorithm

3 Experiments
Intuition: Use constraints to provide better training samples from our unlabelled set of samples
**Intuition:** Use constraints to provide better training samples from our unlabelled set of samples

**Benefit:** Deviations from simple model only do so towards a more expressive answer, since constraints guide the model
Consider the constraint over \(\{0, 1\}^n\) of \(1^*0^*\). For every possible sequence \(\{1^*010^*\}\), there are 2 good fixes \(\{1^*110^*, 1^*000^*\}\). What is the best approach for learning?

Consider the constraint, state transitions must occur on punctuation marks. There could be many good sequences abiding by this constraint.
Outline

1. Background
   - Semi-supervised learning
   - Constraint-driven learning

2. Constraint-driven learning with semi-supervision
   - Introduction
   - Model
     - Scoring
     - Constraint pipeline
     - Constraint Driven Learning (CODL) algorithm

3. Experiments
Consider a data and output domain \( \mathcal{X}, \mathcal{Y} \) and a distribution \( \mathcal{D} \) defined over \( \mathcal{X} \times \mathcal{Y} \).

Input/output pairs \((x, y) \in \mathcal{X} \times \mathcal{Y}\) are given as sequence pairs:

\[
x = (x_1, \cdots, x_N), \quad y = (y_1, \cdots, y_M)
\]

We wish to find a structured output classifier \( h : \mathcal{X} \rightarrow \mathcal{Y} \) that uses a structured scoring function \( f : \mathcal{X} \times \mathcal{Y} \rightarrow \mathbb{R} \) to choose the most likely output sequence, where the correct output sequence is given the highest score.

We are given (or define) a set of constraints \( \{ C_i \}_{i \leq K} \)

\[
C_i : \mathcal{X} \times \mathcal{Y} \rightarrow \{0, 1\}
\]
1 Background
   • Semi-supervised learning
   • Constraint-driven learning

2 Constraint-driven learning with semi-supervision
   • Introduction
   • Model
   • **Scoring**
     • Constraint pipeline
     • Constraint Driven Learning (CODL) algorithm

3 Experiments
Scoring

- Scoring rule:

\[ f(x, y) = \sum_{i=1}^{M} \lambda_i f_i(x, y) = \lambda \cdot F(x, y) \equiv w^T \cdot \phi(x, y) \]

- Compatible for any linear discriminative and generative model such as HMMs and CRFs
- Local feature functions \( \{f_i\}_{i \leq M} \) allow for tractable inference by capturing contextual structure
  - Space of structured pairs could be huge
  - Locally there are smaller distances to account for

Ming-Wei Chang Lev Ratniov Dan Roth (UniGuiding Semi-Supervision with Constraint-Dri
Outline

1 Background
   - Semi-supervised learning
   - Constraint-driven learning

2 Constraint-driven learning with semi-supervision
   - Introduction
   - Model
   - Scoring
   - Constraint pipeline
   - Constraint Driven Learning (CODL) algorithm

3 Experiments
### (a)-Citations

<table>
<thead>
<tr>
<th>Constraint</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Each field must be a consecutive list of words, and can appear at most once in a citation.</td>
</tr>
<tr>
<td>2</td>
<td>State transitions must occur on punctuation marks.</td>
</tr>
<tr>
<td>3</td>
<td>The citation can only start with author or editor.</td>
</tr>
<tr>
<td>4</td>
<td>The words <em>pp.</em>, <em>pages</em> correspond to <em>PAGE</em>.</td>
</tr>
<tr>
<td>5</td>
<td>Four digits starting with 20xx and 19xx are <em>DATE</em>.</td>
</tr>
<tr>
<td>6</td>
<td>Quotations can appear only in titles.</td>
</tr>
<tr>
<td>7</td>
<td>The words <em>note</em>, <em>submitted</em>, <em>appear</em> are <em>NOTE</em>.</td>
</tr>
<tr>
<td>8</td>
<td>The words <em>CA</em>, <em>Australia</em>, <em>NY</em> are <em>LOCATION</em>.</td>
</tr>
<tr>
<td>9</td>
<td>The words <em>tech</em>, <em>technical</em> are <em>TECH.REPORT</em>.</td>
</tr>
<tr>
<td>10</td>
<td>The words <em>proc</em>, <em>journal</em>, <em>proceedings</em>, <em>ACM</em> are <em>JOURNAL</em> or <em>BOOKTITLE</em>.</td>
</tr>
<tr>
<td>11</td>
<td>The words <em>ed</em>, <em>editors</em> correspond to <em>EDITOR</em>.</td>
</tr>
</tbody>
</table>
Constraint-driven learning w/ semi-supervision
Constraint pipeline (hard)

- Define $\mathbf{1}_{C(x)} \subseteq \mathcal{Y}$ as the set of output sequences that assign the value 1 for a given $(x, y)$
- Our objective function then becomes

$$\arg\max_{y \in \mathbf{1}_{C(x)}} \lambda \cdot F(x, y)$$

Intuition: Find best output sequence $y$ that maximizes the score, fitting the hard constraints.
Define $1_{C(x)} \subseteq \mathcal{Y}$ as the set of output sequences that assign the value 1 for a given $(x, y)$.

Our objective function then becomes

$$\arg\max_{y \in 1_{C(x)}} \lambda \cdot F(x, y)$$

**Intuition**: Find best output sequence $y$ that maximizes the score, fitting the hard constraints.
Constraint-driven learning w/ semi-supervision

Constraint pipeline (soft)

- Given a suitable distance metric between an output sequence and the space of outputs respecting a single constraint
- Given a set of soft constraints \( \{ C_i \}_{i \leq K} \) with penalties \( \{ \rho_i \}_{i \leq K} \), we get a new objective function:

\[
\text{argmax}_{y} \quad \lambda \cdot F(x, y) - \sum_{i=1}^{K} \rho_i \cdot d(y, 1_{C_i(x)})
\]
Given a suitable distance metric between an output sequence and the space of outputs respecting a single constraint

Given a set of soft constraints \( \{ C_i \}_{i \leq K} \) with penalties \( \{ \rho_i \}_{i \leq K} \), we get a new objective function:

\[
\arg\max_y \lambda \cdot F(x, y) - \sum_{i=1}^{K} \rho_i \cdot d(y, 1_{C_i(x)})
\]

**Goal**: Find best output sequence under our model that violates the least amount of constraints
Constraint-driven learning w/ semi-supervision

Constraint pipeline (soft)

- Given a suitable distance metric between an output sequence and the space of outputs respecting a single constraint
- Given a set of soft constraints \( \{ C_i \}_{i \leq K} \) with penalties \( \{ \rho_i \}_{i \leq K} \), we get a new objective function:

\[
\arg\max_y \lambda \cdot F(x, y) - \sum_{i=1}^{K} \rho_i \cdot d(y, 1_{C_i(x)})
\]

- **Goal**: Find best output sequence under our model that violates the least amount of constraints
- **Intuition**: Given a learned model, \( \lambda \cdot F(x, y) \), we can bias its decisions using the amount by which the output sequence violates each soft constraint
- **Option**: Use minimal Hamming distance to a sequence

\[
d(y, 1_{C_i(x)}) = \min_{y' \in C_i(x)} H(y, y')
\]
Distance example

<table>
<thead>
<tr>
<th>Lars</th>
<th>Ole</th>
<th>Andersen</th>
<th>.</th>
</tr>
</thead>
<tbody>
<tr>
<td>AUTH</td>
<td>AUTH</td>
<td>EDITOR</td>
<td>EDITOR</td>
</tr>
<tr>
<td>$\Phi_c(y_1) = 0$</td>
<td>$\Phi_c(y_2) = 0$</td>
<td>$\Phi_c(y_3) = 1$</td>
<td>$\Phi_c(y_4) = 0$</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Lars</th>
<th>Ole</th>
<th>Andersen</th>
<th>.</th>
</tr>
</thead>
<tbody>
<tr>
<td>AUTH</td>
<td>BOOK</td>
<td>EDITOR</td>
<td>EDITOR</td>
</tr>
<tr>
<td>$\Phi_c(y_1) = 0$</td>
<td>$\Phi_c(y_2) = 1$</td>
<td>$\Phi_c(y_3) = 1$</td>
<td>$\Phi_c(y_4) = 0$</td>
</tr>
</tbody>
</table>

- Taken from *CCM-NAACL-12-Tutorial*

$$d(y, \mathbf{1}_{C_i(x)}) = \sum_{j=1}^{M} \phi_{C_i}(y_j) \rightarrow \text{count violations}$$
Constraint-driven learning
Recall: sequence tagging

\[ y^* = \arg\max_{y \in \mathcal{Y}} P(y_0)P(x_0|y_0) \prod_{i=1}^{n-1} P(y_i|y_{i-1})P(x_i|y_i) \]

Example: the man saw the dog

Every assignment to the y’s is a path.
Constraint-driven learning
Recall: sequence tagging

\[
\text{maximize } \sum_{y \in \mathcal{Y}} \lambda_{o,y} 1\{y_0 = y\} + \sum_{i=1}^{n-1} \sum_{y \in \mathcal{Y}} \sum_{y' \in \mathcal{Y}} \lambda_{i,y,y'} 1\{y_i = y \land y_{i-1} = y'\}
\]

subject to

\[
\sum_{y \in \mathcal{Y}} 1\{y_0 = y\} = 1
\]

\[
\forall y, \ 1\{y_0 = y\} = \sum_{y' \in \mathcal{Y}} 1\{y_0 = y \land y_1 = y'\}
\]

\[
\forall y, i > 1 \sum_{y' \in \mathcal{Y}} 1\{y_{i-1} = y' \land y_i = y\} = \sum_{y' \in \mathcal{Y}} 1\{y_i = y \land y_{i+1} = y'\}
\]

\[
1\{y_0 = "NN"\} = 1
\]

\[
1\{y_0 = "DT" \land y_1 = "JJ"\} = 1
\]

\[
1\{y_1 = "NN" \land y_2 = "VB"\} = 1
\]

Unique label for each word
Edges that are chosen must form a path
Constraint-driven learning
Recall: sequence tagging

(a) [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. ]

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

- (a) correct citation parsing
- (b) HMM predicted citation parsing
- Adding punctuation state transition constraint returns correct parsing under the same HMM
I left my pearls to my daughter in my will.
Constraint-driven learning

Recall: SRL

\[
\text{maximize} \quad \sum_{i=0}^{n-1} \sum_{y \in \mathcal{Y}} \lambda_{x_i, y} 1\{y_i = y\}
\]

where \( \lambda_{x, y} = \lambda \cdot F(x, y) = \lambda_y \cdot F(x) \)

subject to

\[
\forall i, \quad \sum_{y \in \mathcal{Y}} 1\{y_i = y\} = 1
\]

\[
\forall y \in \mathcal{Y}, \quad \sum_{i=0}^{n-1} 1\{y_i = y\} \leq 1
\]

\[
\forall y \in \mathcal{Y}_R, \quad \sum_{i=0}^{n-1} 1\{y_i = y = \text{“R-Ax”} \} \leq \sum_{i=0}^{n-1} 1\{y_i = \text{“Ax”} \}
\]

\[
\forall j, y \in \mathcal{Y}_C, \quad 1\{y_j = y = \text{“C-Ax”} \} \leq \sum_{i=0}^{j} 1\{y_i = \text{“Ax”} \}
\]
I left my pearls to my daughter in my will.

$[I]_{A0}$ left $[my~pearls]_{A1}$ to $[my~daughter]_{A2}$ in $[my~will]_{AM-LOC}$.

- $A0$: Leaver
- $A1$: Things left
- $A2$: Benefactor
- **AM-LOC**: Location

I left my pearls to my daughter in my will.
1 Background
   • Semi-supervised learning
   • Constraint-driven learning

2 Constraint-driven learning with semi-supervision
   • Introduction
   • Model
   • Scoring
   • Constraint pipeline
   • Constraint Driven Learning (CODL) algorithm

3 Experiments
Constraint-driven learning w/ semi-supervision
CODL algorithm

Constraints are used in inference not learning

Maximum likelihood estimation of $\lambda$ (EM algorithm)

Semi-supervision occurs on line 7,9 (i.e. for each unlabelled sample, we generate $K$ training examples)

All unlabelled samples are re-labelled each training cycle, unlike "self-training" perspective
EM procedures typically assign a distribution over all input/output pairs for a given unlabelled $x \in \mathcal{X}$.

Instead, we choose top $K$, $y \in \mathcal{Y}$ maximizing our soft objective function and assign uniform probability to each output.

Constraints mutate distribution each step.
Constraint-driven learning w/ semi-supervision

Motivation for $K > 1$

- There may be multiple ways to correct constraint-violating samples
- We have access to more training data

![Diagram showing a constraint-directed graph with nodes and constraints](image-url)

**Paper presentation by:** Drew Stone
## Experiments

### Citations

<table>
<thead>
<tr>
<th>$N$</th>
<th>Inf.</th>
<th>sup.</th>
<th>H</th>
<th>H&amp;W</th>
<th>H&amp;W&amp;C (Top-1)</th>
<th>H&amp;W&amp;C (Top-K)</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>no I</td>
<td>55.1</td>
<td>60.9</td>
<td>63.6</td>
<td>70.6</td>
<td>71.0</td>
</tr>
<tr>
<td>I</td>
<td></td>
<td>66.6</td>
<td>69.0</td>
<td>72.5</td>
<td>76.0</td>
<td>77.8</td>
</tr>
<tr>
<td>300</td>
<td>no I</td>
<td>86.1</td>
<td>80.7</td>
<td>87.1</td>
<td>88.2</td>
<td>88.2</td>
</tr>
<tr>
<td>I</td>
<td></td>
<td>92.5</td>
<td>89.6</td>
<td>93.4</td>
<td>93.6</td>
<td>93.5</td>
</tr>
</tbody>
</table>

![Graphs showing performance trends](image)
Experiments

HMM

![Graph](image)

- HMM
- HMM train with constraints
- HMM train/test with constraints

Accuracy vs. Number of labeled example
Pictures taken from:

- **CCM-NAACL-12-Tutorial**