Integer Linear Programming
in
NLP
Constrained Conditional Models

Ming-Wei Chang, Nick Rizzolo, Dan Roth

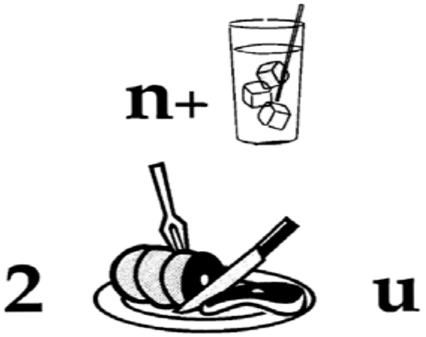
Department of Computer Science

University of Illinois at Urbana-Champaign

June 2010

NAACL

Nice to Meet You



ILP & Constraints Conditional Models (CCMs)

- Making global decisions in which several local interdependent decisions play a role.
- Informally:
  - Everything that has to do with constraints (and learning models)
- Forr
  -
  - s:
  - ons

Issues to attend to:

- **While we formulate the problem as an ILP problem, Inference can be done multiple ways**
  - □ Search; sampling; dynamic programming; SAT; ILP
- **The focus is on joint global inference**
- **Learning may or may not be joint.**
  - □ Decomposing models is often beneficial

- CCMs make predictions in the presence of /guided by constraints

Constraints Driven Learning and Decision Making

- Why Constraints?
  - □ **The Goal: Building a good NLP systems <u>easily</u>**
  - □ **We have prior knowledge at our hand**
    - How can we use it?
    - We suggest that knowledge can often be injected directly
      - □ **Can use it to guide learning**
      - □ **Can use it to improve decision making**
      - □ **Can use it to simplify the models we need to learn**
- How useful are constraints?
  - □ **Useful for supervised learning**
  - □ **Useful for semi-supervised & other label-lean learning paradigms**
  - □ **Sometimes more efficient than labeling data directly**

Inference

Comprehension

A process that maintains and updates a collection of propositions about the state of affairs.

(ENGLAND, June, 1989) - Christopher Robin is alive and well. He lives in England. He is the same person that you read about in the book, Winnie the Pooh. As a boy, Chris lived in a pretty home called Cotchfield Farm. When Chris was three years old, his father wrote a poem about him. The poem was printed in a magazine for others to read. Mr. Robin then wrote a book. He made up a fairy tale land where Chris lived. His friends were animals. There was a bear called Winnie the Pooh. There was also an owl and a young pig, called a piglet. All the animals were stuffed toys that Chris owned. Mr. Robin made them come to life with his words. The places in the story were all near Cotchfield Farm. Winnie the Pooh was written in 1925. Children still love to read about Christopher Robin and his animal friends. Most people don't know he is a real person who is grown now. He has written two books of his own. They tell what it is like to be famous.

1. Christopher Robin was born in England. 2. Winnie the Pooh is a title of a book.
3. Christopher Robin's dad was a magician. 4. Christopher Robin must be at least 65 now.

This is an Inference Problem

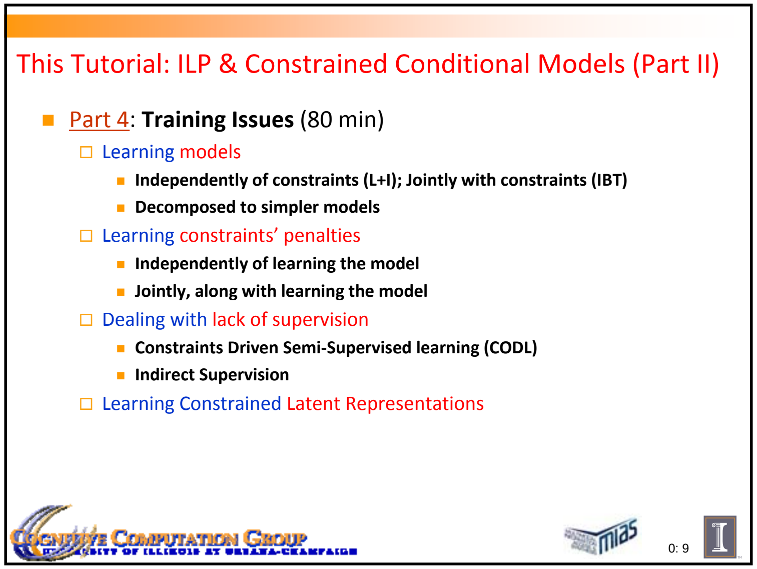
This Tutorial: ILP & Constrained Conditional Models

- Part 1: **Introduction to Constrained Conditional Models** (30min)
  - □ **Examples:**
    - **NE + Relations**
    - **Information extraction – correcting models with CCMS**
  - □ **First summary: Why are CCM important**
  - □ **Problem Setting**
    - **Features and Constraints; Some hints about training issues**

This Tutorial: ILP & Constrained Conditional Models

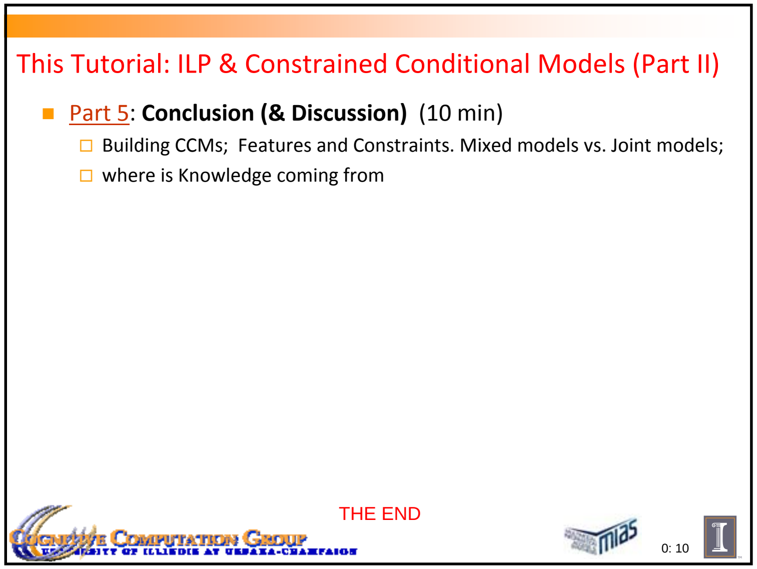
- Part 2: **How to pose the inference problem** (45 minutes)
  - □ Introduction to ILP
  - □ Posing NLP Problems as ILP problems
    - **1. Sequence tagging** **(HMM/CRF + global constraints)**
    - **2. SRL** **(Independent classifiers + Global Constraints)**
    - **3. Sentence Compression** **(Language Model + Global Constraints)**
  - □ Less detailed examples
    - **1. Co-reference**
    - **2. A bunch more ...**
- Part 3: **Inference Algorithms (ILP & Search)** (15 minutes)
  - □ Compiling knowledge to linear inequalities
  - □ Other algorithms like search

BREAK



This Tutorial: ILP & Constrained Conditional Models (Part II)

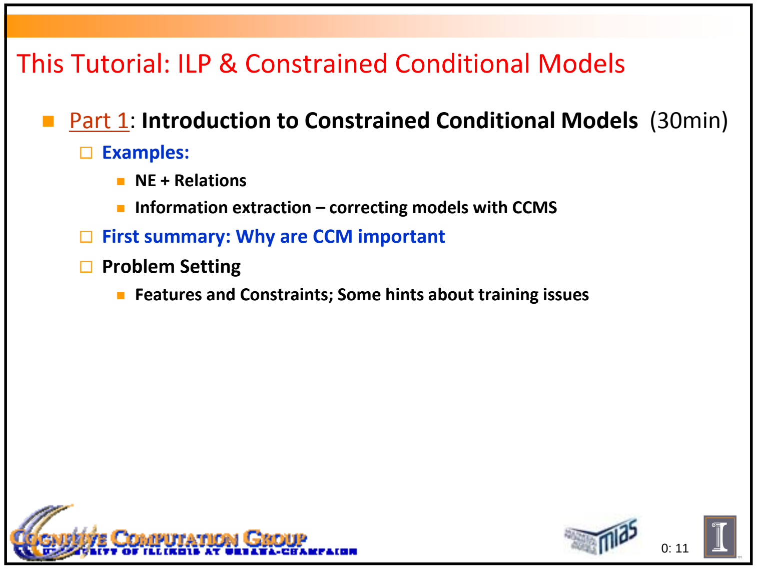
- Part 4: **Training Issues** (80 min)
  - □ Learning models
    - **Independently of constraints (L+I); Jointly with constraints (IBT)**
    - **Decomposed to simpler models**
  - □ Learning constraints' penalties
    - **Independently of learning the model**
    - **Jointly, along with learning the model**
  - □ Dealing with lack of supervision
    - **Constraints Driven Semi-Supervised learning (CODL)**
    - **Indirect Supervision**
  - □ Learning Constrained Latent Representations



This Tutorial: ILP & Constrained Conditional Models (Part II)

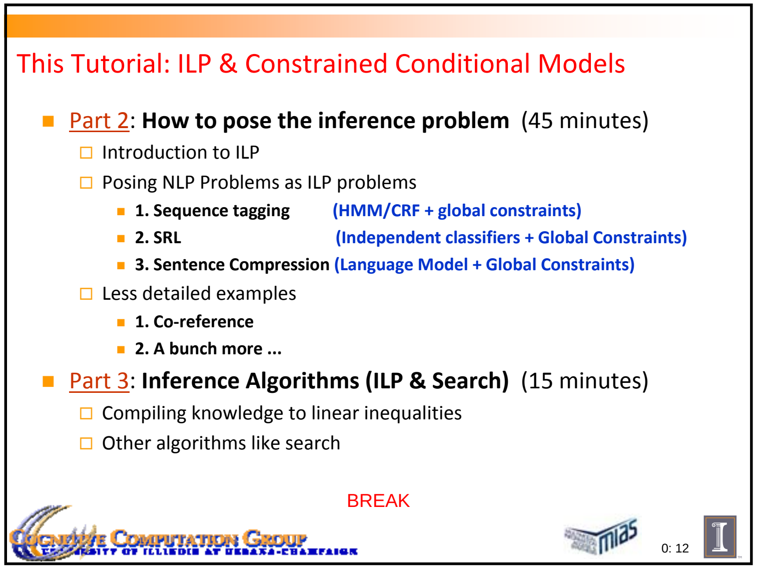
- Part 5: **Conclusion (& Discussion)** (10 min)
  - □ Building CCMs; Features and Constraints. Mixed models vs. Joint models;
  - □ where is Knowledge coming from

THE END



This Tutorial: ILP & Constrained Conditional Models

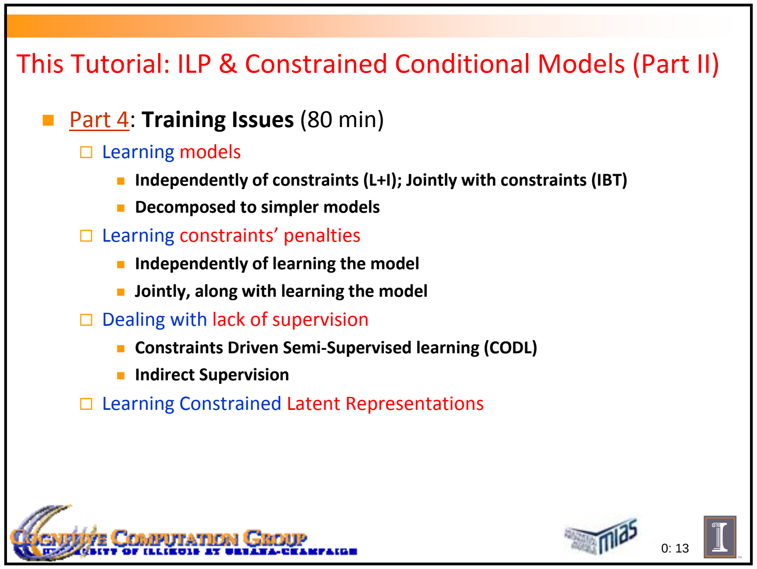
- Part 1: **Introduction to Constrained Conditional Models** (30min)
  - □ **Examples:**
    - **NE + Relations**
    - **Information extraction – correcting models with CCMS**
  - □ **First summary: Why are CCM important**
  - □ **Problem Setting**
    - **Features and Constraints; Some hints about training issues**



This Tutorial: ILP & Constrained Conditional Models

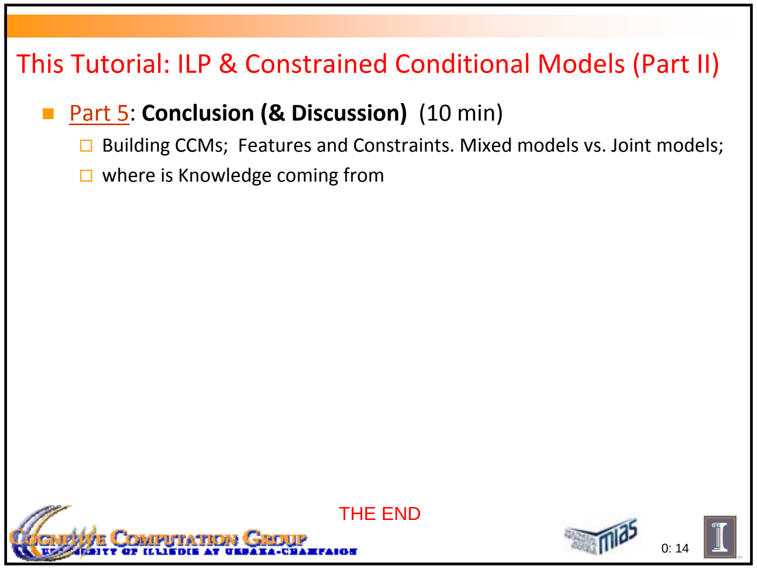
- Part 2: **How to pose the inference problem** (45 minutes)
  - □ Introduction to ILP
  - □ Posing NLP Problems as ILP problems
    - **1. Sequence tagging** **(HMM/CRF + global constraints)**
    - **2. SRL** **(Independent classifiers + Global Constraints)**
    - **3. Sentence Compression** **(Language Model + Global Constraints)**
  - □ Less detailed examples
    - **1. Co-reference**
    - **2. A bunch more ...**
- Part 3: **Inference Algorithms (ILP & Search)** (15 minutes)
  - □ Compiling knowledge to linear inequalities
  - □ Other algorithms like search

BREAK



This Tutorial: ILP & Constrained Conditional Models (Part II)

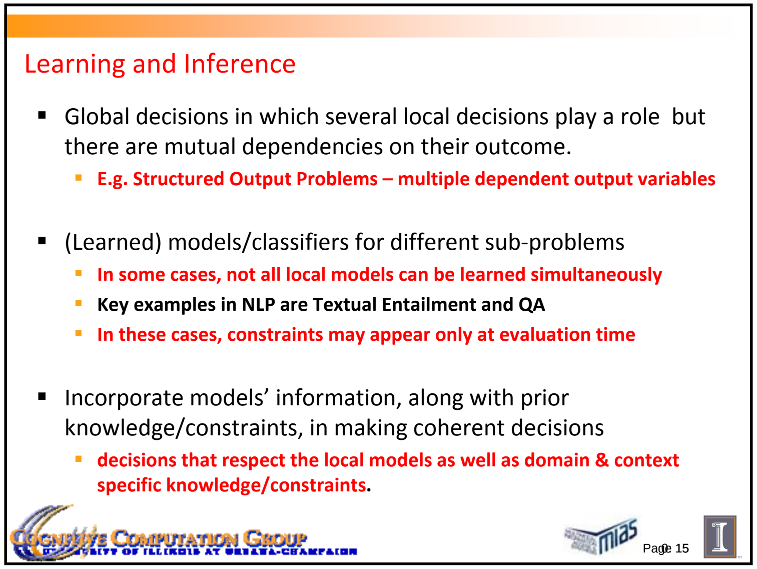
- Part 4: **Training Issues** (80 min)
  - □ Learning models
    - **Independently of constraints (L+I); Jointly with constraints (IBT)**
    - **Decomposed to simpler models**
  - □ Learning constraints' penalties
    - **Independently of learning the model**
    - **Jointly, along with learning the model**
  - □ Dealing with lack of supervision
    - **Constraints Driven Semi-Supervised learning (CODL)**
    - **Indirect Supervision**
  - □ Learning Constrained Latent Representations



This Tutorial: ILP & Constrained Conditional Models (Part II)

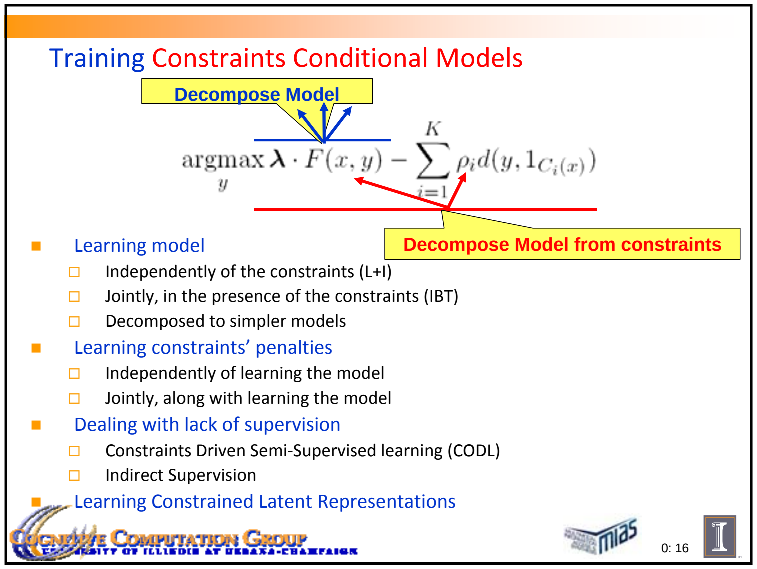
- Part 5: **Conclusion (& Discussion)** (10 min)
  - □ Building CCMs; Features and Constraints. Mixed models vs. Joint models;
  - □ where is Knowledge coming from

THE END



Learning and Inference

- 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 local models can 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 local models as well as domain & context specific knowledge/constraints.**



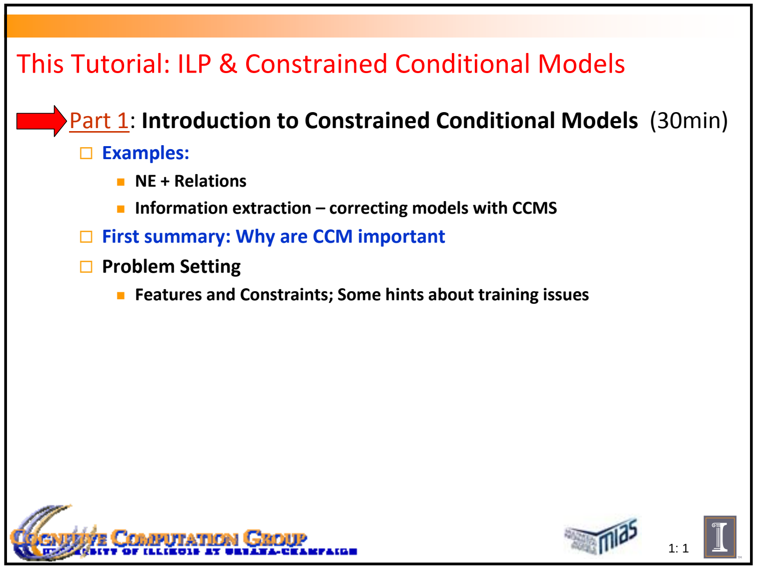
Training Constraints Conditional Models

Decompose Model

$$\operatorname*{argmax}_y \boldsymbol{\lambda} \cdot F(x,y) - \sum_{i=1}^{K} \rho_i d(y, 1_{C_i(x)})$$

Decompose Model from constraints

- Learning model
  - □ Independently of the constraints (L+I)
  - □ Jointly, in the presence of the constraints (IBT)
  - □ Decomposed to simpler models
- Learning constraints' penalties
  - □ Independently of learning the model
  - □ Jointly, along with learning the model
- Dealing with lack of supervision
  - □ Constraints Driven Semi-Supervised learning (CODL)
  - □ Indirect Supervision
- Learning Constrained Latent Representations

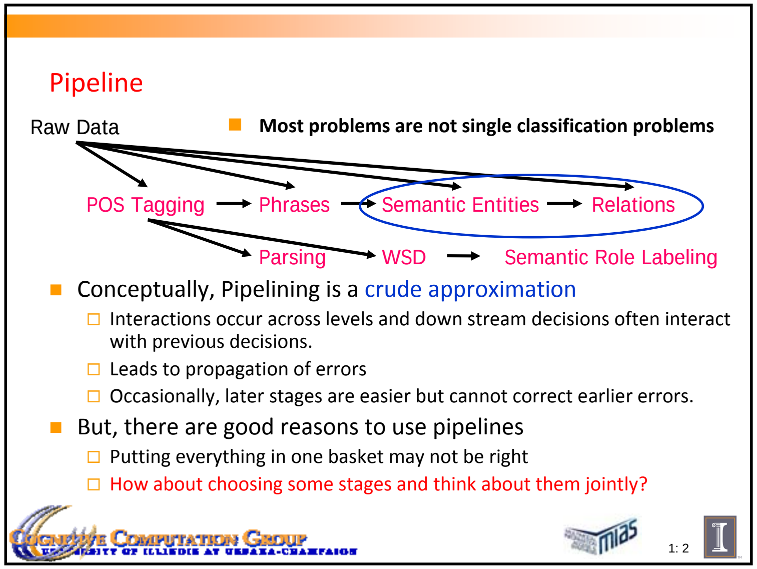


This Tutorial: ILP & Constrained Conditional Models

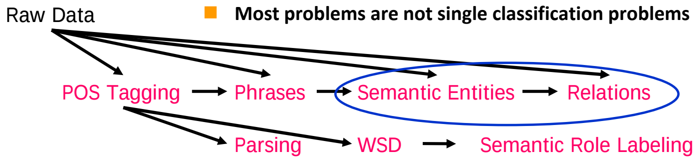
Part 1: **Introduction to Constrained Conditional Models** (30min)

- **Examples:**
  - **NE + Relations**
  - **Information extraction – correcting models with CCMS**
- **First summary: Why are CCM important**
- **Problem Setting**
  - **Features and Constraints; Some hints about training issues**

1: 1

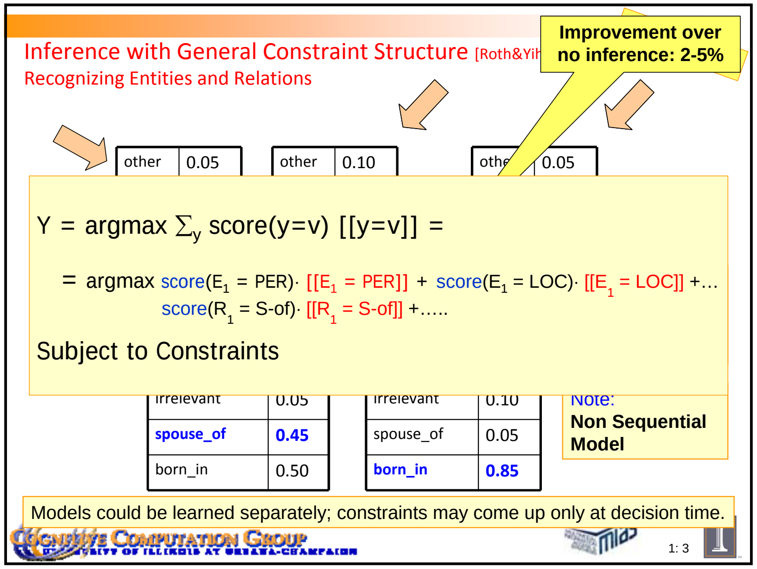


Pipeline



- **Most problems are not single classification problems**
- Conceptually, Pipelining is a crude approximation
  - Interactions occur across levels and down stream decisions often interact with previous decisions.
  - Leads to propagation of errors
  - Occasionally, later stages are easier but cannot correct earlier errors.
- But, there are good reasons to use pipelines
  - Putting everything in one basket may not be right
  - How about choosing some stages and think about them jointly?

1: 2



Inference with General Constraint Structure [Roth&Yih

Recognizing Entities and Relations

Improvement over no inference: 2-5%

other	0.05

other	0.10

othe	0.05

Y = argmax $\sum_y$ score(y=v) [[y=v]] =

= argmax score(E_1 = PER)· [[E_1 = PER]] + score(E_1 = LOC)· [[E_1 = LOC]] +…
score(R_1 = S-of)· [[R_1 = S-of]] +…..

Subject to Constraints

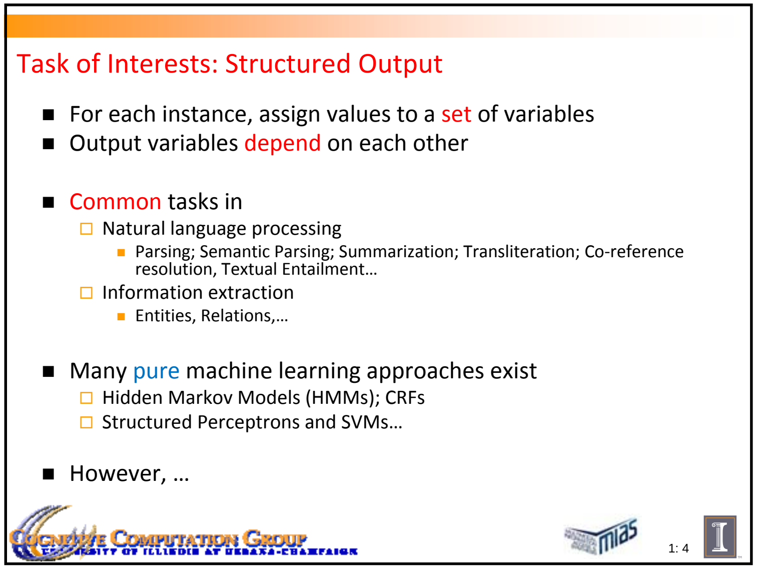
irrelevant	0.05
spouse_of	**0.45**
born_in	0.50

irrelevant	0.10
spouse_of	0.05
born_in	**0.85**

Note:
Non Sequential Model

Models could be learned separately; constraints may come up only at decision time.

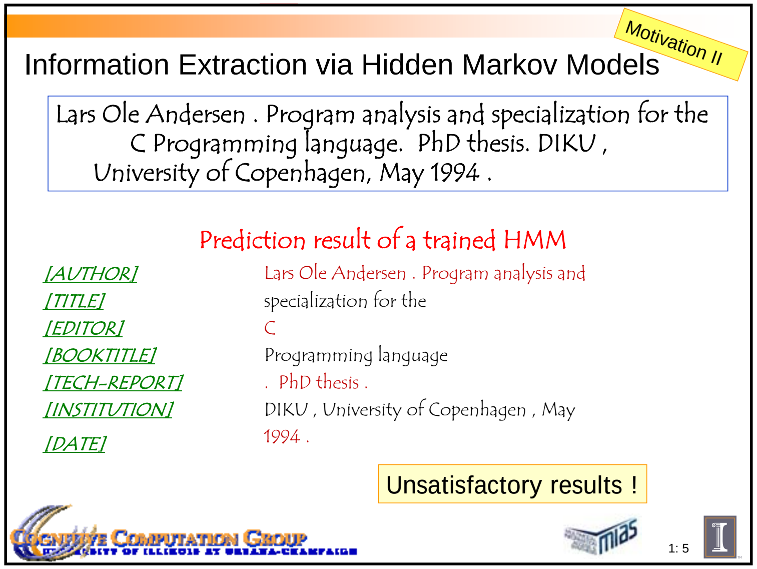
1: 3



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; Transliteration; Co-reference resolution, Textual Entailment…
  - Information extraction
    - Entities, Relations,…
- Many pure machine learning approaches exist
  - Hidden Markov Models (HMMs); CRFs
  - Structured Perceptrons and SVMs…
- However, …

1: 4



Information Extraction via Hidden Markov Models

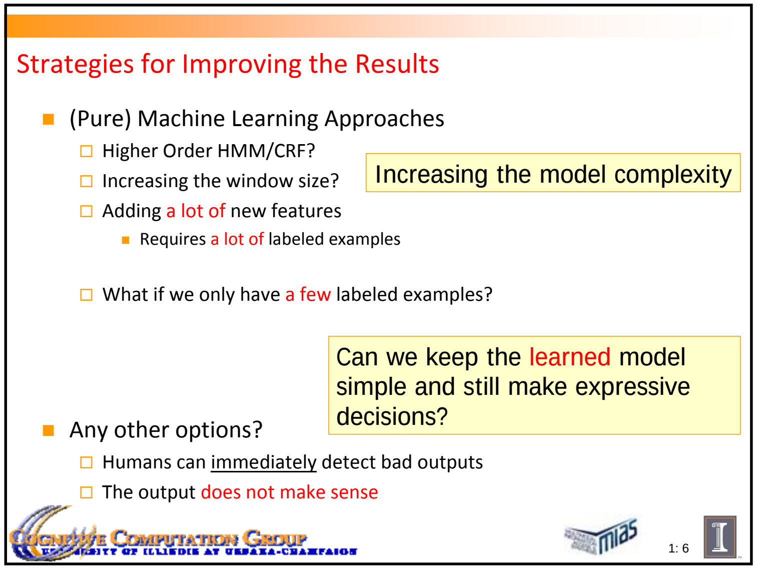
Motivation II

Lars Ole Andersen . Program analysis and specialization for the
C Programming language. PhD thesis. DIKU ,
University of Copenhagen, May 1994 .

Prediction result of a trained HMM

[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 .

Unsatisfactory results !

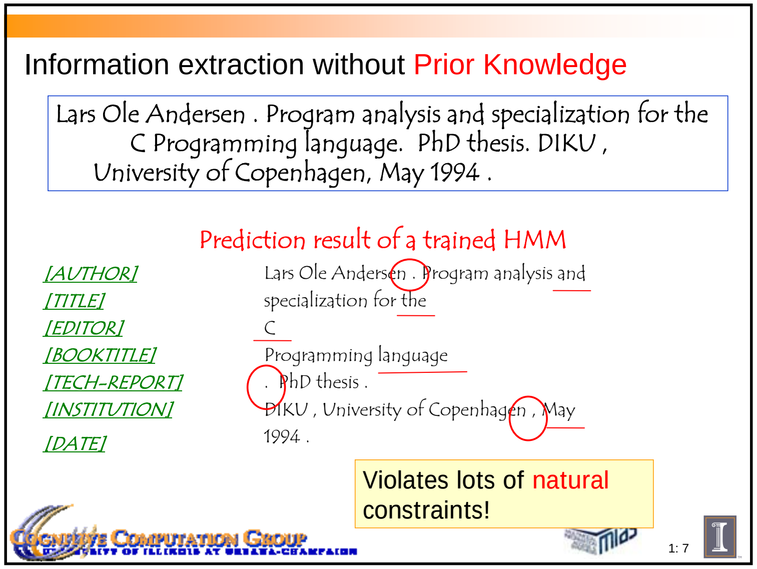


Strategies for Improving the Results

- (Pure) Machine Learning Approaches
  - □ Higher Order HMM/CRF?
  - □ 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?
- Any other options?
  - □ Humans can immediately detect bad outputs
  - □ The output does not make sense

Increasing the model complexity

Can we keep the learned model simple and still make expressive decisions?



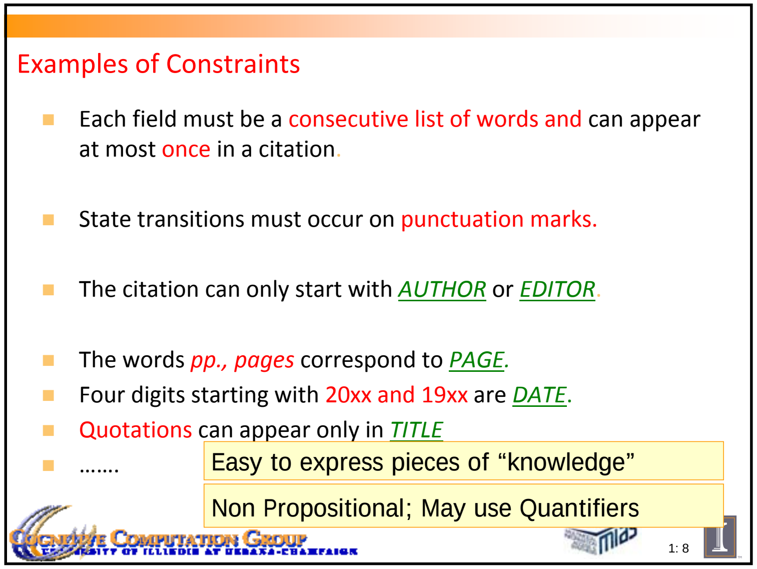
Information extraction without Prior Knowledge

Lars Ole Andersen . Program analysis and specialization for the
C Programming language. PhD thesis. DIKU ,
University of Copenhagen, May 1994 .

Prediction result of a trained HMM

[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 .

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

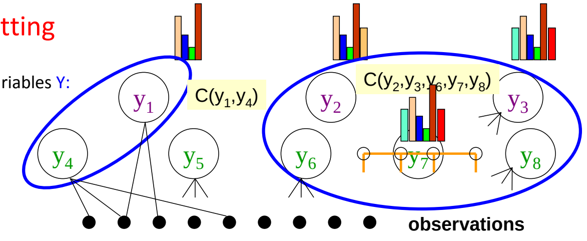
Information Extraction with Constraints

- 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 .

Constrained Conditional Models Allow:
- **Learning a simple model**
- **Make decisions with a more complex model**
- **Accomplished by directly incorporating constraints to bias/re-ranks decisions made by the simpler model**

Problem Setting



- Random Variables Y:
- Conditional Distributions P (learned by models/classifiers)
- Constraints C– any Boolean function
  defined over partial assignments (possibly: + weights W)
- Goal: Find the “best” assignment
  - □ **The assignment that achieves the highest global performance.**
- This is an Integer Programming Problem

$Y^*=\text{argmax}_Y\ P\bullet Y\ (+\ W\bullet C)$ subject to constraints C

Constrained Conditional Models (aka ILP Inference)

$$\underset{y}{\text{argmax}}\, \boldsymbol{\lambda}\cdot F(x,y) - \sum_{i=1}^{K} \rho_i d(y, 1_{C_i(x)})$$

Penalty for violating the constraint.

(Soft) constraints component

Weight Vector for “local” models

Features, classifiers; log-linear models (HMM, CRF) or a combination

How far y is from a “legal” assignment

CCMs can be viewed as a general interface to easily combine domain knowledge with data driven statistical models

How to solve?

This is an Integer Linear Program
Solving using ILP packages gives an exact solution.
Search techniques are also possible

How to train?

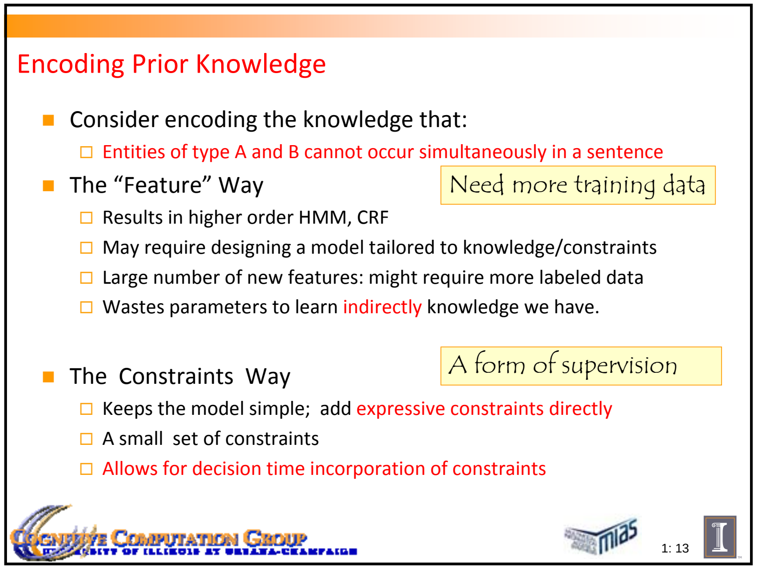
Training is learning the objective Function.
How to exploit the structure to minimize supervision?

$f_{\Phi,C}(\mathbf{x},\mathbf{y}) = \sum w_i\phi_i(\mathbf{x},\mathbf{y}) - \sum \rho_i d_{C_i}(\mathbf{x},\mathbf{y})$

Features Versus Constraints

- ϕ_i : X × Y → R; C_i : X × Y → {0,1}; d: X × Y → R;
  - □ In principle, constraints and features can encode the same propeties
  - □ In practice, they are very different
- Features
  - □ Local , short distance properties – to allow tractable inference
  - □ Propositional (grounded):
  - □ E.g. True if: “the” followed by a Noun occurs in the sentence”
- Constraints
  - □ Global properties
  - □ Quantified, first order logic expressions
  - □ E.g.True if: “all y_is in the sequence y are assigned different values.”

Indeed, used differently



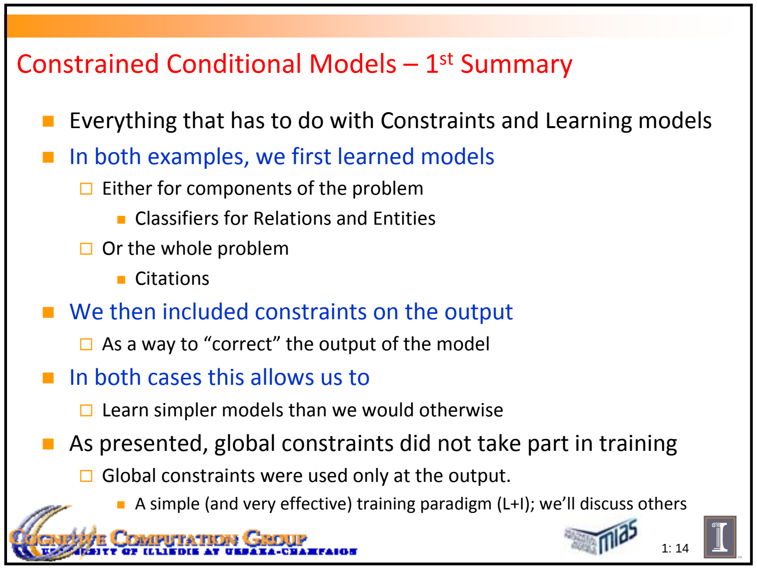
Encoding Prior Knowledge

- 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
  - Wastes parameters to learn indirectly knowledge we have.

Need more training data

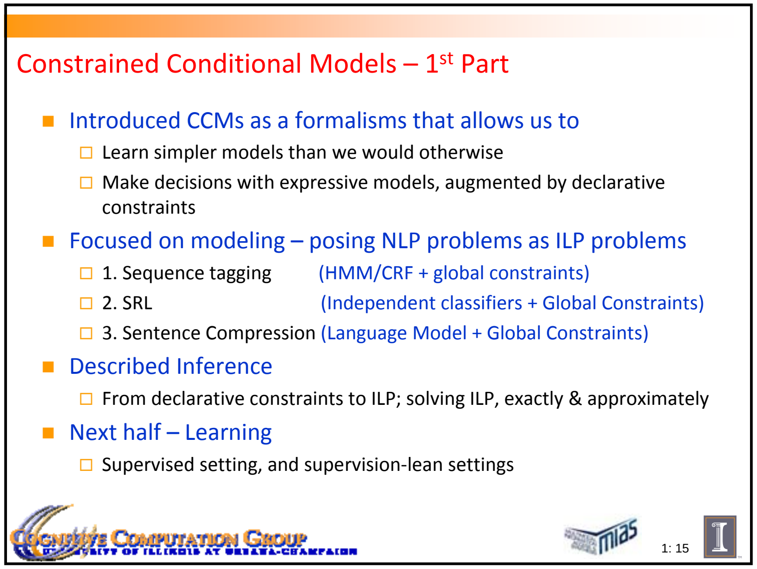
- The Constraints Way
  - Keeps the model simple; add expressive constraints directly
  - A small set of constraints
  - Allows for decision time incorporation of constraints

A form of supervision



Constrained Conditional Models – 1^{st} Summary

- Everything that has to do with Constraints and Learning models
- In both examples, we first learned models
  - Either for components of the problem
    - Classifiers for Relations and Entities
  - Or the whole problem
    - Citations
- We then included constraints on the output
  - As a way to “correct” the output of the model
- In both cases this allows us to
  - Learn simpler models than we would otherwise
- As presented, global constraints did not take part in training
  - Global constraints were used only at the output.
    - A simple (and very effective) training paradigm (L+I); we’ll discuss others



Constrained Conditional Models – 1^{st} Part

- Introduced CCMs as a formalisms that allows us to
  - Learn simpler models than we would otherwise
  - Make decisions with expressive models, augmented by declarative constraints
- Focused on modeling – posing NLP problems as ILP problems
  - 1. Sequence tagging (HMM/CRF + global constraints)
  - 2. SRL (Independent classifiers + Global Constraints)
  - 3. Sentence Compression (Language Model + Global Constraints)
- Described Inference
  - From declarative constraints to ILP; solving ILP, exactly & approximately
- Next half – Learning
  - Supervised setting, and supervision-lean settings

This Tutorial: ILP & Constrained Conditional Models

- Part 2: **How to pose the inference problem** (45 minutes)
  - Introduction to ILP
  - Posing NLP Problems as ILP problems
    - 1. Sequence tagging (HMM/CRF + global constraints)
    - 2. SRL (Independent classifiers + Global Constraints)
    - 3. Sentence Compression (Language Model + Global Constraints)
  - Less detailed examples
    - 1. Co-reference
    - 2. A bunch more ...
- Part 3: **Inference Algorithms (ILP & Search)** (15 minutes)
  - Compiling knowledge to linear inequalities
  - Other algorithms like search

BREAK

CCMs are Optimization Problems

- We pose inference as an optimization problem
  - Integer Linear Programming (ILP)
- Advantages:
  - *Keep model small; easy to learn*
  - *Still allowing expressive, long-range constraints*
  - Mathematical optimization is well studied
  - Exact solution to the inference problem is possible
  - Powerful off-the-shelf solvers exist
- Disadvantage:
  - The inference problem could be NP-hard

Linear Programming: Example

- Telfa Co. produces tables and chairs
  - Each table makes $8 profit, each chair makes $5 profit.
- We want to maximize the profit.

Decision Variables

$$x_1 = \text{number of tables manufactured}$$
$$x_2 = \text{number of chairs manufactured}$$

Objective function

$$\text{Profit} = 8x_1 + 5x_2$$

Linear Programming: Example

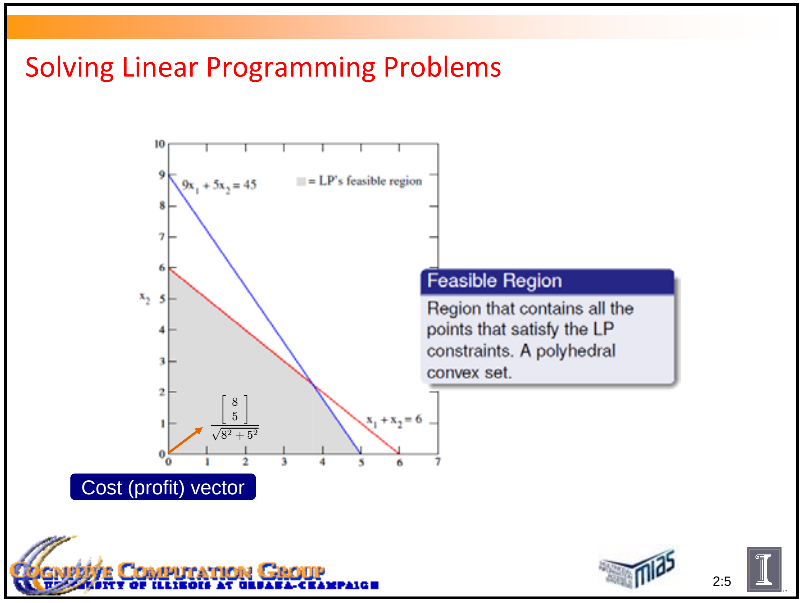
- Telfa Co. produces tables and chairs
  - Each table makes $8 profit, each chair makes $5 profit.
  - A table requires 1 hour of labor and 9 sq. feet of wood.
  - A chair requires 1 hour of labor and 5 sq. feet of wood.
  - We have only 6 hours of work and 45sq. feet of wood.
- We want to maximize the profit.

Objective function

$$\text{Profit} = 8x_1 + 5x_2 \qquad z = \vec{c} \cdot \vec{x}$$

Constraints

$$\begin{array}{lrcrcl}
\text{Labour constraint} & x_1 & + & x_2 & \leq & 6 \\
\text{Wood constraint} & 9x_1 & + & 5x_2 & \leq & 45 \\
\text{Variable constraints} & & & x_1 & \geq & 0 \\
 & & & x_2 & \geq & 0
\end{array} \qquad \mathbf{A}\vec{x} \leq \vec{b}$$

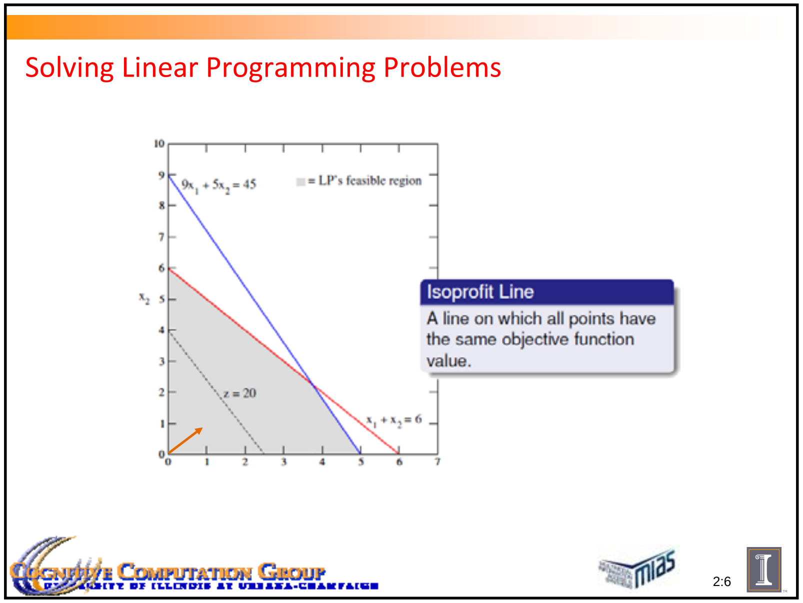


Solving Linear Programming Problems

Feasible Region

Region that contains all the points that satisfy the LP constraints. A polyhedral convex set.

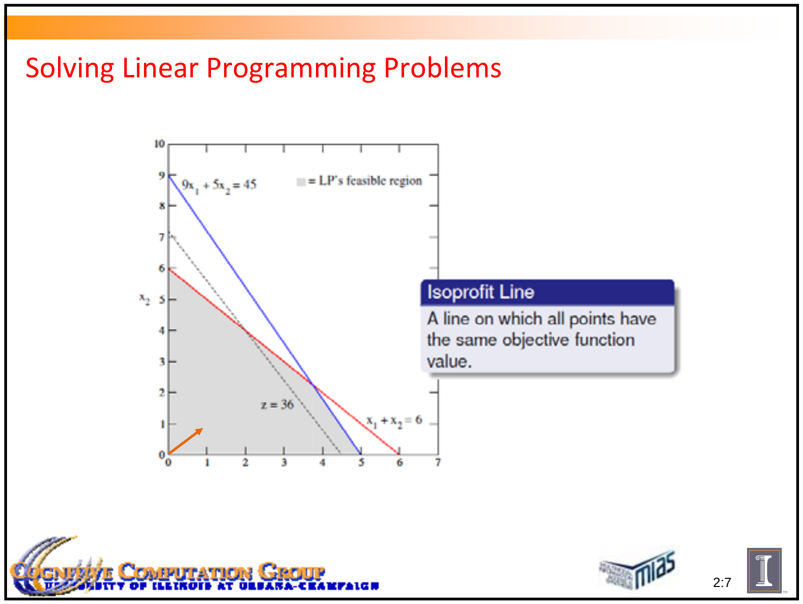
Cost (profit) vector



Solving Linear Programming Problems

Isoprofit Line

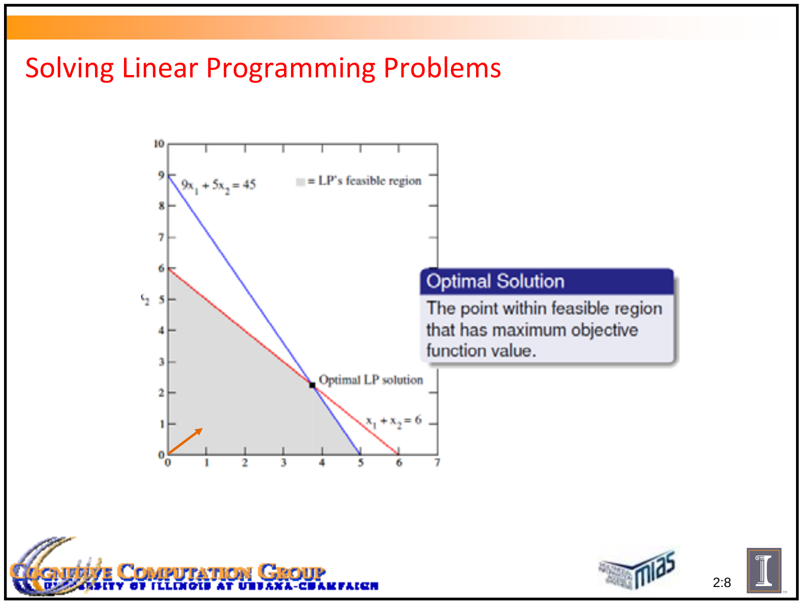
A line on which all points have the same objective function value.



Solving Linear Programming Problems

Isoprofit Line

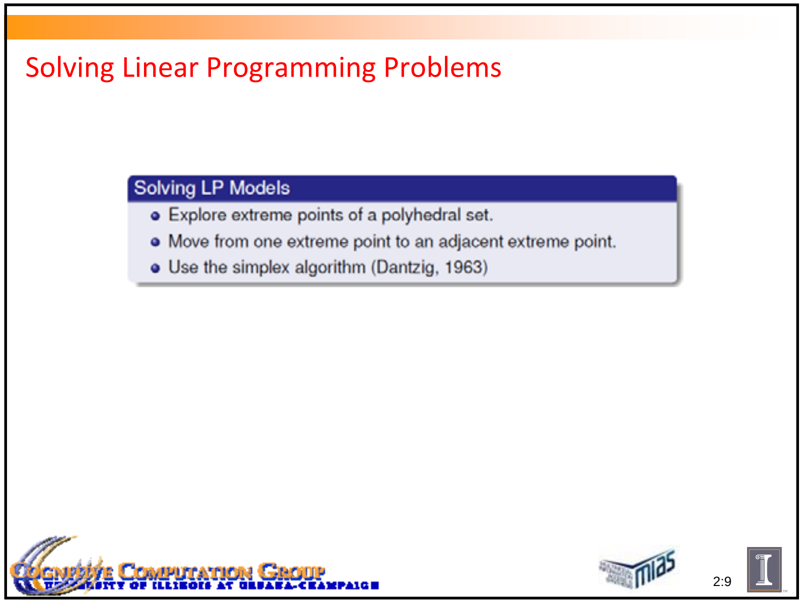
A line on which all points have the same objective function value.



Solving Linear Programming Problems

Optimal Solution

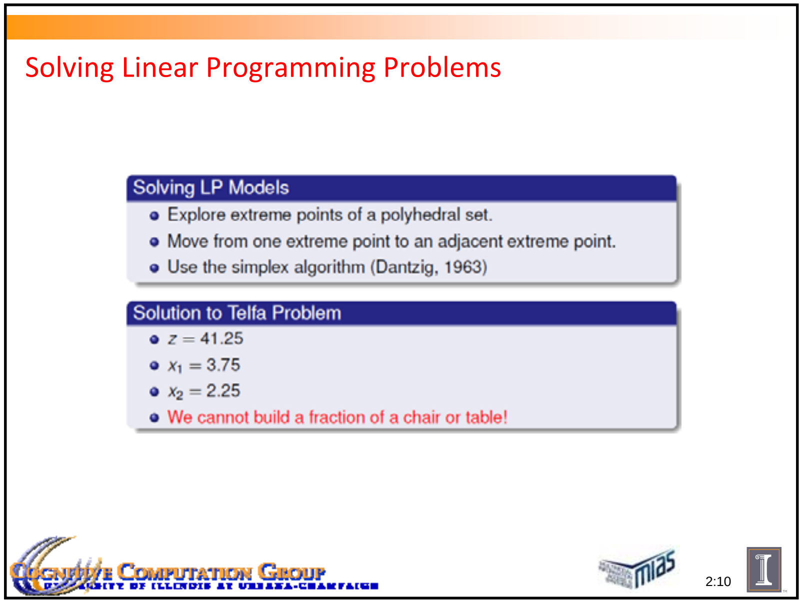
The point within feasible region that has maximum objective function value.



Solving Linear Programming Problems

Solving LP Models

- Explore extreme points of a polyhedral set.
- Move from one extreme point to an adjacent extreme point.
- Use the simplex algorithm (Dantzig, 1963)



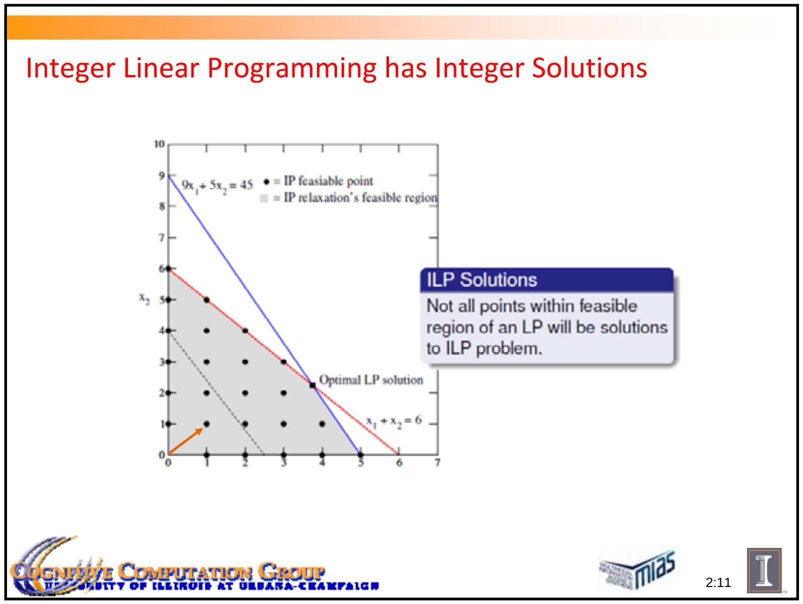
Solving Linear Programming Problems

Solving LP Models

- Explore extreme points of a polyhedral set.
- Move from one extreme point to an adjacent extreme point.
- Use the simplex algorithm (Dantzig, 1963)

Solution to Telfa Problem

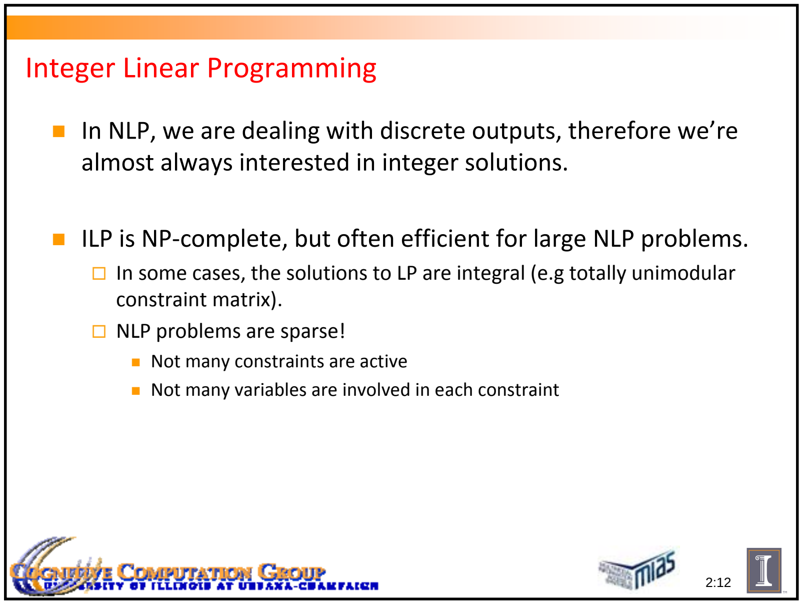
- $z = 41.25$
- $x_1 = 3.75$
- $x_2 = 2.25$
- We cannot build a fraction of a chair or table!



Integer Linear Programming has Integer Solutions

ILP Solutions

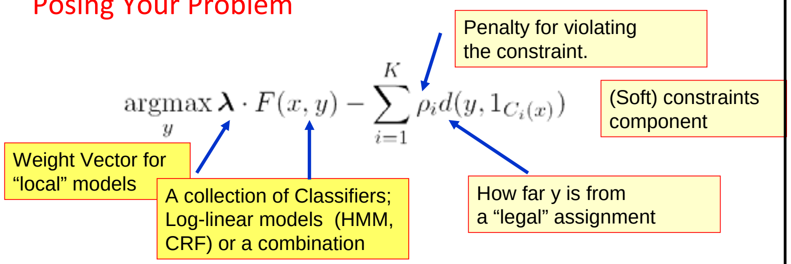
Not all points within feasible region of an LP will be solutions to ILP problem.



Integer Linear Programming

- In NLP, we are dealing with discrete outputs, therefore we're almost always interested in integer solutions.
- ILP is NP-complete, but often efficient for large NLP problems.
  - In some cases, the solutions to LP are integral (e.g totally unimodular constraint matrix).
  - NLP problems are sparse!
    - Not many constraints are active
    - Not many variables are involved in each constraint

Posing Your Problem



$$\underset{y}{\operatorname{argmax}}\ \boldsymbol{\lambda} \cdot F(x, y) - \sum_{i=1}^{K} \rho_i d(y, 1_{C_i(x)})$$

Penalty for violating the constraint.

(Soft) constraints component

Weight Vector for "local" models

A collection of Classifiers; Log-linear models (HMM, CRF) or a combination

How far y is from a "legal" assignment

- How do we write our models in this form?
  - □ What goes in an objective function?
  - □ How to design constraints?

CCM Examples

- Many works in NLP make use of constrained conditional models, implicitly or explicitly.
- Next we describe three examples in detail.

- ➡ Example 1: Sequence Tagging
  - □ Adding long range constraints to a simple model
- Example 2: Semantic Role Labeling
  - □ The use of inference with constraints to improve semantic parsing
- Example 3: Sentence Compression
  - □ Simple language model with constraints outperforms complex models

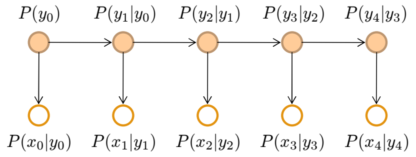
Example 1: Sequence Tagging

HMM / CRF:

$$\mathbf{y}^* = \underset{\mathbf{y} \in \mathcal{Y}}{\operatorname{argmax}}\ P(y_0)P(x_0|y_0) \prod_{i=1}^{n-1} P(y_i|y_{i-1})P(x_i|y_i)$$

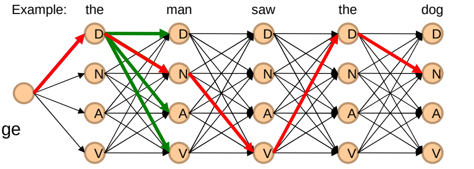
Here, *y*'s are variables; *x*'s are fixed.

Our objective function must include all entries of the CPTs.



Every edge is a Boolean variable that selects a transition CPT entry.

They are related: if we choose y_0 = D then we must choose an edge y_0 = D ∧ y_1 = ? .

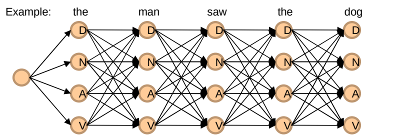


Every assignment to the *y*'s is a path.

Example 1: Sequence Tagging

HMM / CRF:

$$\mathbf{y}^* = \underset{\mathbf{y} \in \mathcal{Y}}{\operatorname{argmax}}\ P(y_0)P(x_0|y_0) \prod_{i=1}^{n-1} P(y_i|y_{i-1})P(x_i|y_i)$$



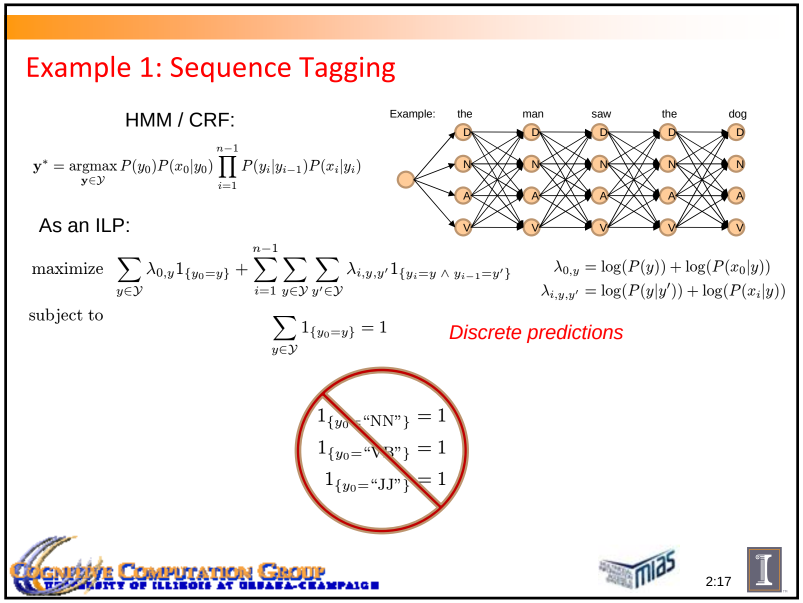
As an ILP:

Inference Variables

$$\text{maximize} \quad \sum_{y \in \mathcal{Y}} \lambda_{0,y} 1_{\{y_0 = y\}} + \sum_{i=1}^{n-1} \sum_{y \in \mathcal{Y}} \sum_{y' \in \mathcal{Y}} \lambda_{i,y,y'} 1_{\{y_i = y \ \wedge \ y_{i-1} = y'\}}$$

$$\begin{aligned} \lambda_{0,y} &= \log(P(y)) + \log(P(x_0|y)) \\ \lambda_{i,y,y'} &= \log(P(y|y')) + \log(P(x_i|y)) \end{aligned}$$

subject to



Example 1: Sequence Tagging

HMM / CRF:

$$\mathbf{y}^* = \underset{\mathbf{y}\in\mathcal{Y}}{\operatorname{argmax}}\, P(y_0)P(x_0|y_0)\prod_{i=1}^{n-1} P(y_i|y_{i-1})P(x_i|y_i)$$

As an ILP:

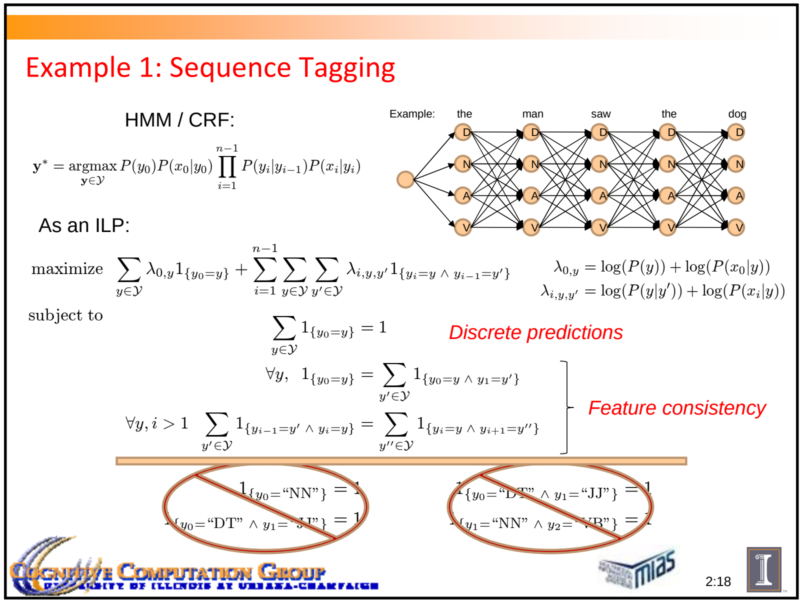
$$\text{maximize} \sum_{y\in\mathcal{Y}} \lambda_{0,y} 1_{\{y_0=y\}} + \sum_{i=1}^{n-1}\sum_{y\in\mathcal{Y}}\sum_{y'\in\mathcal{Y}} \lambda_{i,y,y'} 1_{\{y_i=y\ \wedge\ y_{i-1}=y'\}}$$

$$\lambda_{0,y} = \log(P(y)) + \log(P(x_0|y))$$
$$\lambda_{i,y,y'} = \log(P(y|y')) + \log(P(x_i|y))$$

subject to

$$\sum_{y\in\mathcal{Y}} 1_{\{y_0=y\}} = 1$$ *Discrete predictions*

$$\cancel{\begin{array}{l} 1_{\{y_0=\text{``NN''}\}} = 1 \\ 1_{\{y_0=\text{``VB''}\}} = 1 \\ 1_{\{y_0=\text{``JJ''}\}} = 1 \end{array}}$$



Example 1: Sequence Tagging

HMM / CRF:

$$\mathbf{y}^* = \underset{\mathbf{y}\in\mathcal{Y}}{\operatorname{argmax}}\, P(y_0)P(x_0|y_0)\prod_{i=1}^{n-1} P(y_i|y_{i-1})P(x_i|y_i)$$

As an ILP:

$$\text{maximize} \sum_{y\in\mathcal{Y}} \lambda_{0,y} 1_{\{y_0=y\}} + \sum_{i=1}^{n-1}\sum_{y\in\mathcal{Y}}\sum_{y'\in\mathcal{Y}} \lambda_{i,y,y'} 1_{\{y_i=y\ \wedge\ y_{i-1}=y'\}}$$

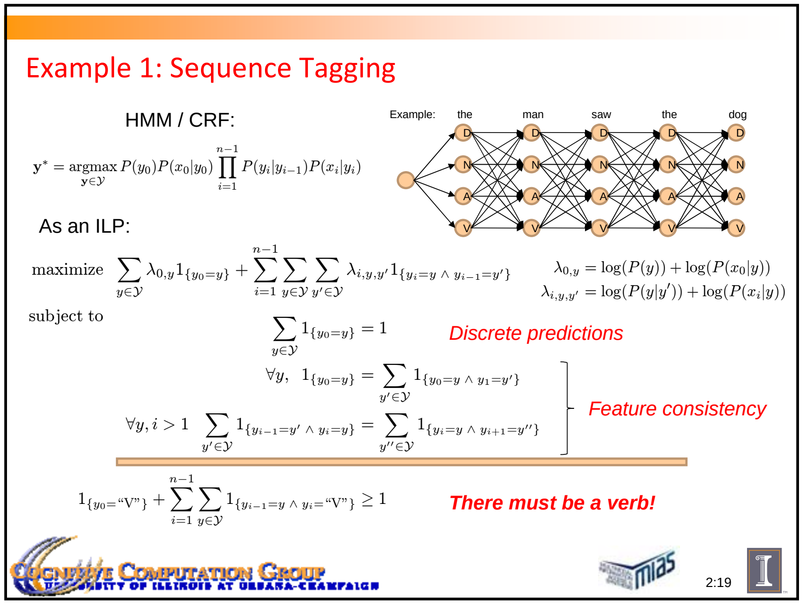
$$\lambda_{0,y} = \log(P(y)) + \log(P(x_0|y))$$
$$\lambda_{i,y,y'} = \log(P(y|y')) + \log(P(x_i|y))$$

subject to

$$\sum_{y\in\mathcal{Y}} 1_{\{y_0=y\}} = 1$$ *Discrete predictions*

$$\left.\begin{array}{r} \forall y,\ \ 1_{\{y_0=y\}} = \sum_{y'\in\mathcal{Y}} 1_{\{y_0=y\ \wedge\ y_1=y'\}} \\ \forall y, i>1\ \ \sum_{y'\in\mathcal{Y}} 1_{\{y_{i-1}=y'\ \wedge\ y_i=y\}} = \sum_{y''\in\mathcal{Y}} 1_{\{y_i=y\ \wedge\ y_{i+1}=y''\}} \end{array}\right\}$$ *Feature consistency*

$$\cancel{\begin{array}{l} 1_{\{y_0=\text{``NN''}\}} = 1 \\ 1_{\{y_0=\text{``DT''}\ \wedge\ y_1=\text{``JJ''}\}} = 1 \end{array}} \qquad \cancel{\begin{array}{l} 1_{\{y_0=\text{``DT''}\ \wedge\ y_1=\text{``JJ''}\}} = 1 \\ 1_{\{y_1=\text{``NN''}\ \wedge\ y_2=\text{``VB''}\}} = 1 \end{array}}$$



Example 1: Sequence Tagging

HMM / CRF:

$$\mathbf{y}^* = \underset{\mathbf{y}\in\mathcal{Y}}{\operatorname{argmax}}\, P(y_0)P(x_0|y_0)\prod_{i=1}^{n-1} P(y_i|y_{i-1})P(x_i|y_i)$$

As an ILP:

$$\text{maximize} \sum_{y\in\mathcal{Y}} \lambda_{0,y} 1_{\{y_0=y\}} + \sum_{i=1}^{n-1}\sum_{y\in\mathcal{Y}}\sum_{y'\in\mathcal{Y}} \lambda_{i,y,y'} 1_{\{y_i=y\ \wedge\ y_{i-1}=y'\}}$$

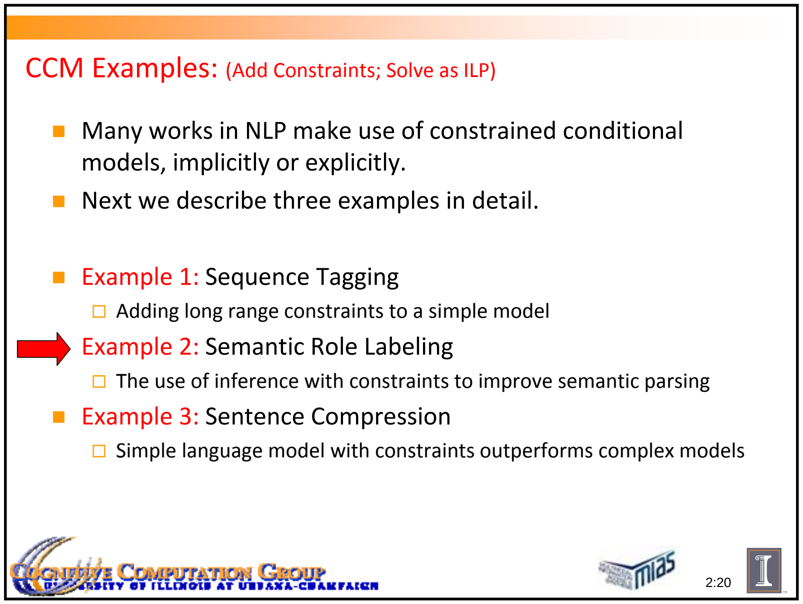
$$\lambda_{0,y} = \log(P(y)) + \log(P(x_0|y))$$
$$\lambda_{i,y,y'} = \log(P(y|y')) + \log(P(x_i|y))$$

subject to

$$\sum_{y\in\mathcal{Y}} 1_{\{y_0=y\}} = 1$$ *Discrete predictions*

$$\left.\begin{array}{r} \forall y,\ \ 1_{\{y_0=y\}} = \sum_{y'\in\mathcal{Y}} 1_{\{y_0=y\ \wedge\ y_1=y'\}} \\ \forall y, i>1\ \ \sum_{y'\in\mathcal{Y}} 1_{\{y_{i-1}=y'\ \wedge\ y_i=y\}} = \sum_{y''\in\mathcal{Y}} 1_{\{y_i=y\ \wedge\ y_{i+1}=y''\}} \end{array}\right\}$$ *Feature consistency*

$$1_{\{y_0=\text{``V''}\}} + \sum_{i=1}^{n-1}\sum_{y\in\mathcal{Y}} 1_{\{y_{i-1}=y\ \wedge\ y_i=\text{``V''}\}} \geq 1$$ ***There must be a verb!***



CCM Examples: (Add Constraints; Solve as ILP)

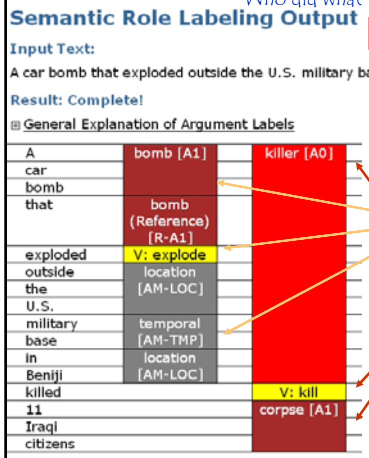
- Many works in NLP make use of constrained conditional models, implicitly or explicitly.
- Next we describe three examples in detail.

- Example 1: Sequence Tagging
  - □ Adding long range constraints to a simple model
- ➡ Example 2: Semantic Role Labeling
  - □ The use of inference with constraints to improve semantic parsing
- Example 3: Sentence Compression
  - □ Simple language model with constraints outperforms complex models

Example 2: Semantic Role Labeling

Who did what to whom, when, where, why,…

Demo:http://L2R.cs.uiuc.edu/~cogcomp



Semantic Role Labeling Output

Input Text:

A car bomb that exploded outside the U.S. military base in Beniji killed 11 Iraqi citizens.

Result: Complete!

⊞ General Explanation of Argument Labels

A	bomb [A1]	killer [A0]
car		
bomb		
that	bomb (Reference) [R-A1]	
exploded	V: explode	
outside	location [AM-LOC]	
the		
U.S.		
military	temporal [AM-TMP]	
base		
in	location [AM-LOC]	
Beniji		
killed		V: kill
11		corpse [A1]
Iraqi		
citizens		

Approach :
1) Reveals several relations.

2) Produces a very good semantic parser. F1~90%

3) Easy and fast: ~7 Sent/Sec (using Xpress-MP)

Top ranked system in CoNLL'05 shared task

Key difference is the Inference

Simple sentence:

I *left* my pearls to my daughter in my will .

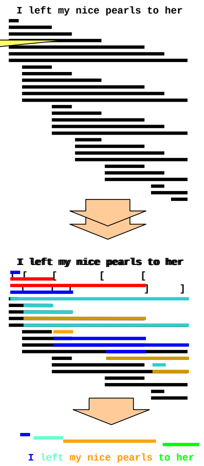
[I]$_{A0}$ *left* [my pearls]$_{A1}$ [to my daughter]$_{A2}$ [in my will]$_{AM-LOC}$.

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

I *left* my pearls to my daughter in my will .

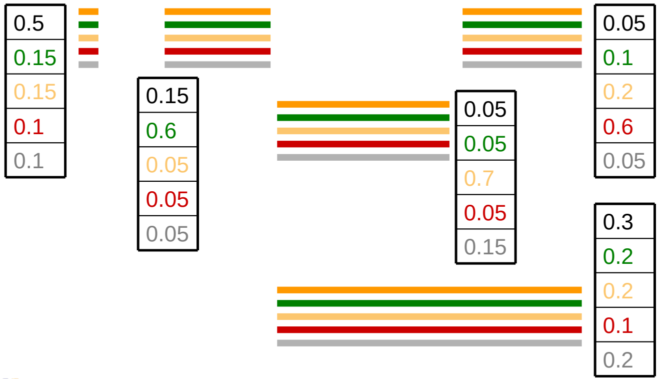
Algorithmic Approach

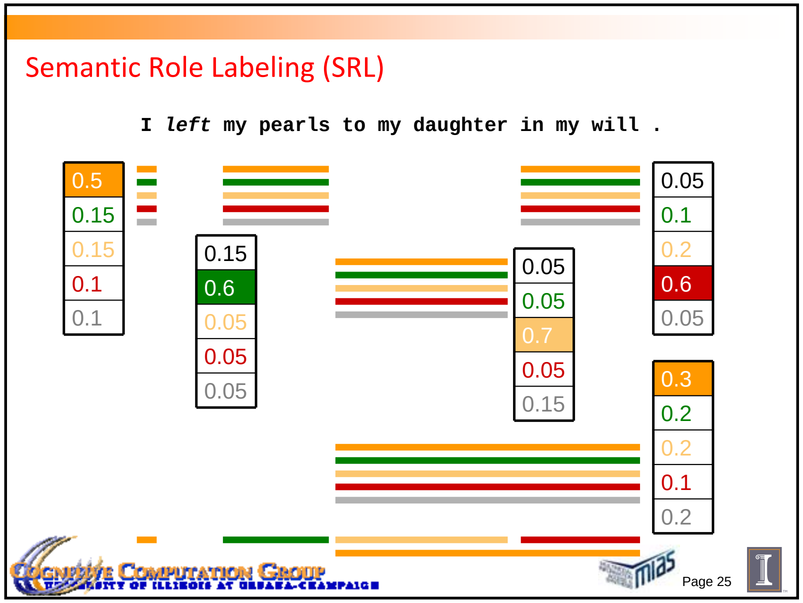
- **Identify** argument candidates
  - □ Pruning [Xue&Palmer, EMNLP'04]
  - □ Argument Identifier
    - Binary classification
- **Classify** argument candidates
  - □ Argument Classifier
    - Multi-class classification
- **Inference**
  - □ Use the estimated probability distribution given by the argument classifier
  - □ Use structural and linguistic constraints
  - □ Infer the optimal global output


candidate arguments
I left my nice pearls to her
I left my nice pearls to her
I left my nice pearls to her

Semantic Role Labeling (SRL)

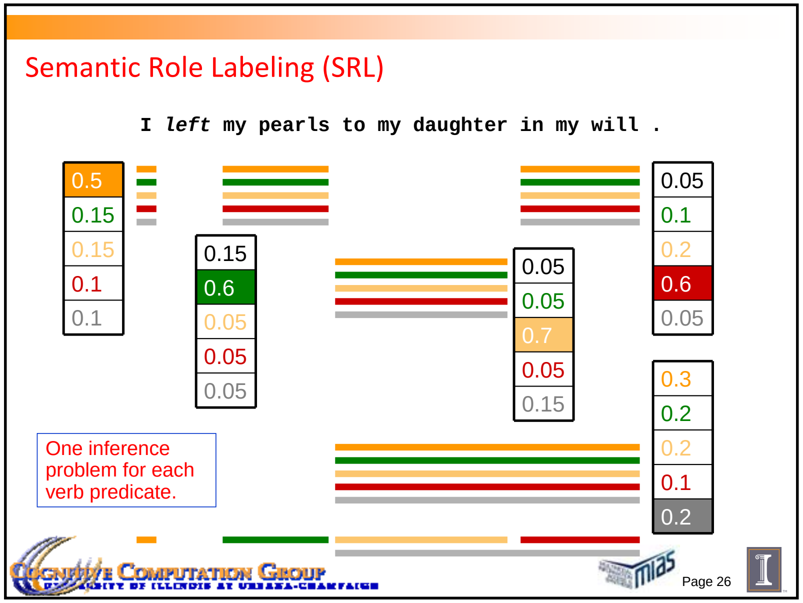
I *left* my pearls to my daughter in my will .





Semantic Role Labeling (SRL)

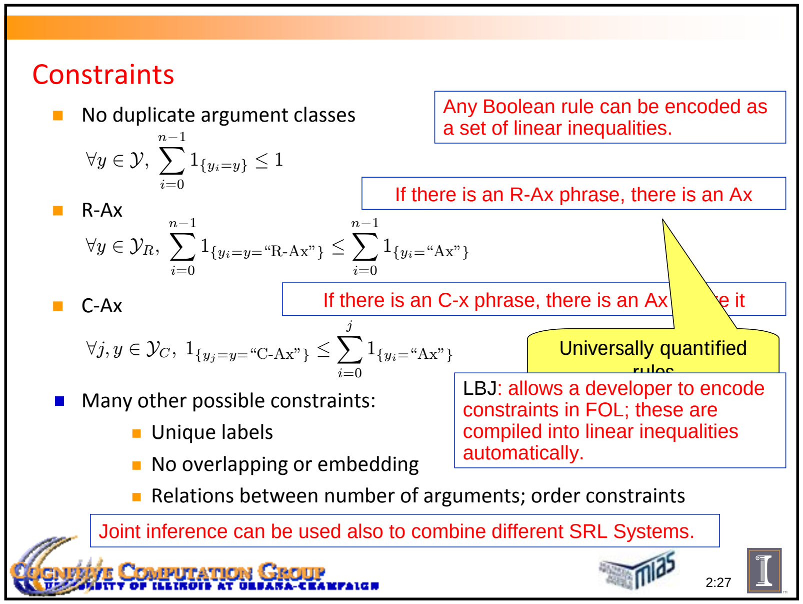
I ***left*** **my pearls to my daughter in my will .**



Semantic Role Labeling (SRL)

I ***left*** **my pearls to my daughter in my will .**

One inference problem for each verb predicate.



Constraints

Any Boolean rule can be encoded as a set of linear inequalities.

- No duplicate argument classes

$$\forall y \in \mathcal{Y},\ \sum_{i=0}^{n-1} 1_{\{y_i = y\}} \leq 1$$

If there is an R-Ax phrase, there is an Ax

- R-Ax

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

If there is an C-x phrase, there is an Ax e it

- C-Ax

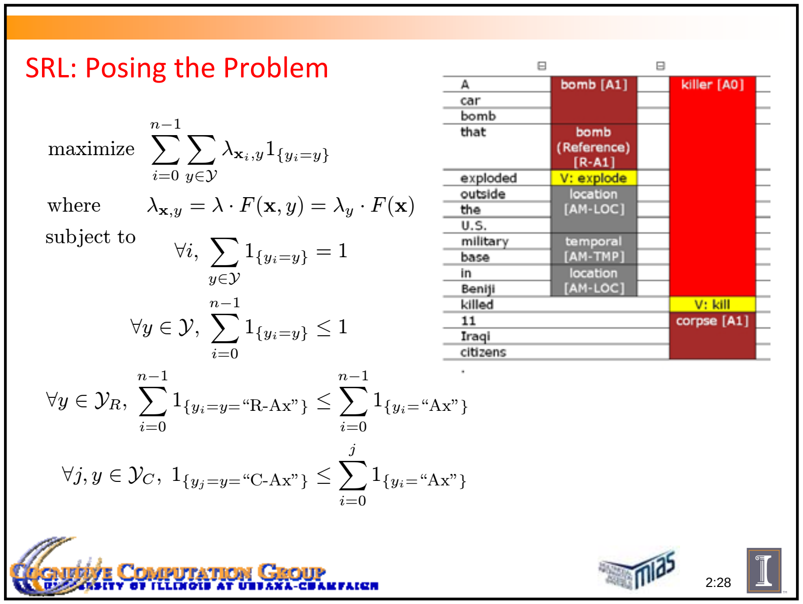
$$\forall j, y \in \mathcal{Y}_C,\ 1_{\{y_j = y = \text{“C-Ax”}\}} \leq \sum_{i=0}^{j} 1_{\{y_i = \text{“Ax”}\}}$$

Universally quantified rules

- Many other possible constraints:
  - Unique labels
  - No overlapping or embedding
  - Relations between number of arguments; order constraints

LBJ: allows a developer to encode constraints in FOL; these are compiled into linear inequalities automatically.

Joint inference can be used also to combine different SRL Systems.



SRL: Posing the Problem

$$\text{maximize} \quad \sum_{i=0}^{n-1} \sum_{y \in \mathcal{Y}} \lambda_{\mathbf{x}_i, y} 1_{\{y_i = y\}}$$

$$\text{where} \quad \lambda_{\mathbf{x},y} = \lambda \cdot F(\mathbf{x}, y) = \lambda_y \cdot F(\mathbf{x})$$

$$\text{subject to} \quad \forall i,\ \sum_{y \in \mathcal{Y}} 1_{\{y_i = y\}} = 1$$

$$\forall y \in \mathcal{Y},\ \sum_{i=0}^{n-1} 1_{\{y_i = y\}} \leq 1$$

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

$$\forall j, y \in \mathcal{Y}_C,\ 1_{\{y_j = y = \text{“C-Ax”}\}} \leq \sum_{i=0}^{j} 1_{\{y_i = \text{“Ax”}\}}$$

	⊟	⊟
A	bomb [A1]	killer [A0]
car		
bomb		
that	bomb (Reference) [R-A1]	
exploded	V: explode	
outside	location [AM-LOC]	
the		
U.S.		
military	temporal [AM-TMP]	
base		
in	location [AM-LOC]	
Beniji		
killed		V: kill
11		corpse [A1]
Iraqi		
citizens		
.		

CCM Examples: (Add Constraints; Solve as ILP)

- Many works in NLP make use of constrained conditional models, implicitly or explicitly.
- Next we describe three examples in detail.

- Example 1: Sequence Tagging
  - □ Adding long range constraints to a simple model
- Example 2: Semantic Role Labeling
  - □ The use of inference with constraints to improve semantic parsing
- Example 3: Sentence Compression
  - □ Simple language model with constraints outperforms complex models

Example 3: Sentence Compression (Clarke & Lapata)

He became a power player in Greek Politics in 1974, when he founded the socialist Pasok Party.

He became a player in politics.

We took these troubled youth who don't have fathers, and brought them into the room to Dads who don't have their children.

We took these youth and brought them into the room to Dads.

Example

Trigram Objective Function

$$\max \sum_{i=0}^{n-2} \sum_{j=i+1}^{n-1} \sum_{k=j+1}^{n} \gamma_{ijk} \cdot P(x_k | x_i, x_j)$$

Example:

0	1	2	3	4	5	6	7	8
Big	fish	eat	small	fish	in	a	small	pond
Big	fish				in	a		pond

$$\delta_0 = \delta_1 = \delta_5 = \delta_6 = \delta_8 = 1$$

$$\gamma_{015} = \gamma_{156} = \gamma_{568} = 1$$

Language model-based compression

Trigram Objective Function

$$\max \sum_{i=0}^{n-2} \sum_{j=i+1}^{n-1} \sum_{k=j+1}^{n} \gamma_{ijk} \cdot P(x_k | x_i, x_j)$$

Decision Variables

$$\delta_i = \begin{cases} 1 & \text{if } x_i \text{ is in the compression} \\ 0 & \text{otherwise} \end{cases} \quad (1 \le i \le n)$$

Auxiliary Variables

$$\gamma_{ijk} = \begin{cases} 1 & \text{if word sequence } x_i, x_j, x_k \text{ is in the compression} \\ 0 & \text{otherwise} \end{cases}$$

Example: Summarization

Trigram Objective Function

$$\max \sum_{i=0}^{n-2} \sum_{j=i+1}^{n-1} \sum_{k=j+1}^{n} \gamma_{ijk} \cdot P(x_k|x_i, x_j)$$

This formulation requires some additional constraints

Big fish eat small fish in a small pond

No selection of decision variables can make these trigrams appear consecutively in output.

We skip these constraints here.

Trigram model in action

He became a power player in Greek Politics in 1974, when he founded the socialist Pasok Party.

He became a player in the Pasok.

We took these troubled youth who don't have fathers, and brought them into the room to Dads who don't have their children.

We don't have, and don't have children.

Modifier Constraints

Modifier Constraints

- Ensure the relationships between head words and their modifiers remain grammatical.
- If a modifier is in the compression, its head word must be included:

$$\delta_{head} - \delta_{modifer} \geq 0$$

- Do not drop *not* if the head word is in the compression (same for words like *his*, *our* and genitives).

Example

He became a power player in Greek Politics in 1974, when he founded the socialist Pasok Party.

He became a player in the Pasok.

We took these troubled youth who don't have fathers, and brought them into the room to Dads who don't have their children.

We don't have, and don't have children.

Example

He became a power player in Greek Politics in 1974, when he founded the socialist Pasok Party.

He became a player in the Pasok Party.

We took these troubled youth who don't have fathers, and brought them into the room to Dads who don't have their children.

We don't have them don't have their children.

Sentential Constraints

Sentential Constraints

- Take the overall sentence structure into account.
- If a verb is in the compression then so are its arguments, and vice-versa:

$$\delta_{subject/object} - \delta_{verb} = 0$$

- The compression must contain at least one verb.

Example

He became a power player in Greek Politics in 1974, when he founded the socialist Pasok Party.

He became a player in the Pasok Party.

We took these troubled youth who don't have fathers, and brought them into the room to Dads who don't have their children.

We don't have them don't have their children.

Example

He became a power player in Greek Politics in 1974, when he founded the socialist Pasok Party.

He became a player in politics.

We took these troubled youth who don't have fathers, and brought them into the room to Dads who don't have their children.

We took these youth and brought them into the room to Dads.

More constraints

Discourse Constraints

- Preserve the discourse flow of the original document.
- Focus on local discourse.
- Retain personal pronouns.
  $\delta_{pronoun} = 1$
- Centering constraint over adjacent sentences.
  $\delta_{center} = 1$
- Lexical chains constraint on nouns in prevalent chains.
  $\delta_{topical} = 1$

Sentence Compression: Posing the Problem

Learned Parameters

Inference Variables

$$\text{maximize} \sum_{i=0}^{n-2} \sum_{j=i+1}^{n-1} \sum_{k=j+1}^{n} \lambda_{k,i,j}\, \gamma_{i,j,k}$$

If the three corresponding auxiliary variables are on, the inference variable must be on.

subject to

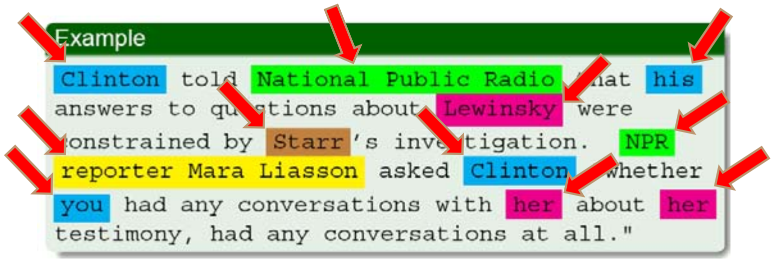
$$\forall i, j, k,\ 0 \le i < j < k \le n, \quad \boxed{3\gamma_{i,j,k} \le \delta_i + \delta_j + \delta_k}$$

$$\boxed{2 + \gamma_{i,j,k} \ge \delta_i + \delta_j + \delta_k}$$

$$\boxed{(k - i - 2)\gamma_{i,j,k} + \sum_{s=i+1}^{j-1} \delta_s + \sum_{s=j+1}^{k-1} \delta_s \le k - i - 2}$$

If the inference variable is on, no intermediate auxiliary variables may be on.

Other CCM Examples: Coref (Denis & Baldridge)

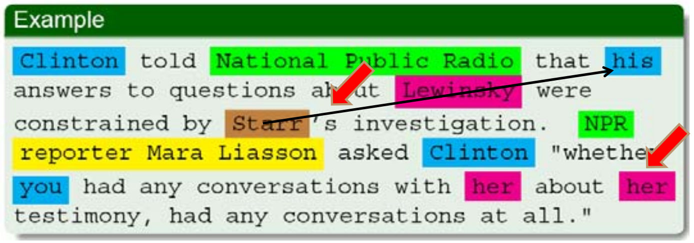


Example

Clinton told National Public Radio that his
answers to questions about Lewinsky were
constrained by Starr's investigation. NPR
reporter Mara Liasson asked Clinton "whether
you had any conversations with her about her
testimony, had any conversations at all."

Two types of entities:
➡ "Base entities"
➡ "Anaphors" (pointers)

Other CCM Examples: Coref (Denis & Baldridge)

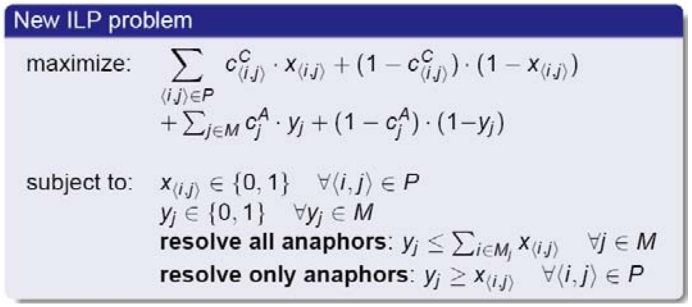


Example

Clinton told National Public Radio that his
answers to questions about Lewinsky were
constrained by Starr's investigation. NPR
reporter Mara Liasson asked Clinton "whether
you had any conversations with her about her
testimony, had any conversations at all."

Error analysis:
1)"Base entities" that "point" to anaphors.
2)Anaphors that don't "point" to anything.

Other CCM Examples: Coref (Denis & Baldridge)



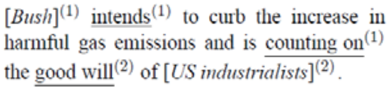
New ILP problem

maximize: $\sum_{\langle i,j\rangle\in P} c^C_{\langle i,j\rangle}\cdot x_{\langle i,j\rangle} + (1-c^C_{\langle i,j\rangle})\cdot(1-x_{\langle i,j\rangle})$
$+\sum_{j\in M} c^A_j\cdot y_j + (1-c^A_j)\cdot(1-y_j)$

subject to: $x_{\langle i,j\rangle}\in\{0,1\} \quad \forall\langle i,j\rangle\in P$
$y_j\in\{0,1\} \quad \forall y_j\in M$
resolve all anaphors: $y_j \le \sum_{i\in M_j} x_{\langle i,j\rangle} \quad \forall j\in M$
resolve only anaphors: $y_j \ge x_{\langle i,j\rangle} \quad \forall\langle i,j\rangle\in P$

Other CCM Examples: Opinion Recognition

- Y. Choi, E. Breck, and C. Cardie. Joint Extraction of Entities and Relations for Opinion Recognition EMNLP-2006

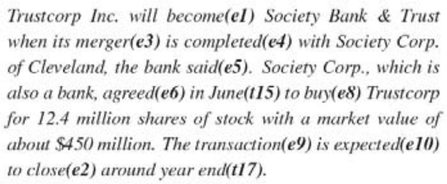


[Bush]$^{(1)}$ intends$^{(1)}$ to curb the increase in harmful gas emissions and is counting on$^{(1)}$ the good will$^{(2)}$ of *[US industrialists]*$^{(2)}$.

- Semantic parsing variation:
  - Agent=entity
  - Relation=opinion
- Constraints:
  - An agent can have at most two opinions.
  - An opinion should be linked to only one agent.
  - The usual non-overlap constraints.

Other CCM Examples: Temporal Ordering

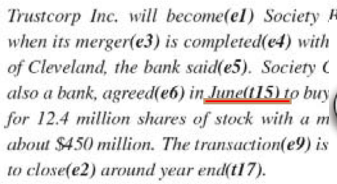
- N. Chambers and D. Jurafsky. Jointly Combining Implicit Constraints Improves Temporal Ordering. EMNLP-2008.



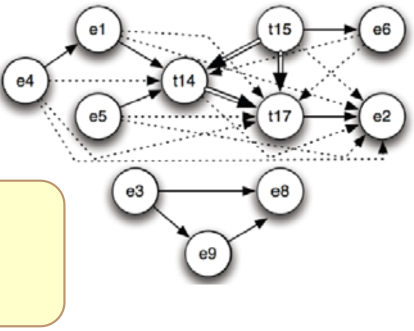
*Trustcorp Inc. will become(**e1**) Society Bank & Trust when its merger(**e3**) is completed(**e4**) with Society Corp. of Cleveland, the bank said(**e5**). Society Corp., which is also a bank, agreed(**e6**) in June(**t15**) to buy(**e8**) Trustcorp for 12.4 million shares of stock with a market value of about $450 million. The transaction(**e9**) is expected(**e10**) to close(**e2**) around year end(**t17**).*

Other CCM Examples: Temporal Ordering

- N. Chambers and D. Jurafsky. Jointly Combining Implicit Constraints Improves Temporal Ordering. EMNLP-2008.



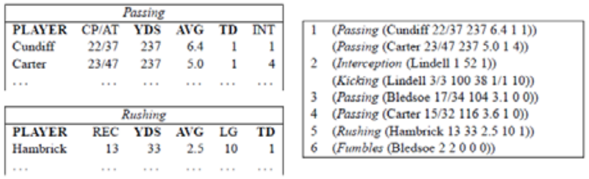
*Trustcorp Inc. will become(**e1**) Society Bank & Trust*
*when its merger(**e3**) is completed(**e4**) with*
*of Cleveland, the bank said(**e5**). Society C*
*also a bank, agreed(**e6**) in June(**t15**) to buy*
for 12.4 million shares of stock with a m
*about $450 million. The transaction(**e9**) is*
*to close(**e2**) around year end(**t17**).*



Three types of edges:
1)Annotation relations before/after
2)Transitive closure constraints
3)Time normalization constraints

Related Work: Language generation.

- Regina Barzilay and Mirella Lapata. Aggregation via Set Partitioning for Natural Language Generation.HLT-NAACL-2006.



Passing

PLAYER	CP/AT	YDS	AVG	TD	INT
Cundiff	22/37	237	6.4	1	1
Carter	23/47	237	5.0	1	4
…	…	…	…	…	…

Rushing

PLAYER	REC	YDS	AVG	LG	TD
Hambrick	13	33	2.5	10	1
…	…	…	…	…	…

1 (*Passing* (Cundiff 22/37 237 6.4 1 1))
(*Passing* (Carter 23/47 237 5.0 1 4))
2 (*Interception* (Lindell 1 52 1))
(*Kicking* (Lindell 3/3 100 38 1/1 10))
3 (*Passing* (Bledsoe 17/34 104 3.1 0 0))
4 (*Passing* (Carter 15/32 116 3.6 1 0))
5 (*Rushing* (Hambrick 13 33 2.5 10 1))
6 (*Fumbles* (Bledsoe 2 2 0 0 0))

- Constraints:
  - □ Transitivity: if (e_i,e_j)were aggregated, and (e_i,e_{jk}) were too, then (e_i,e_k) get aggregated.
  - □ Max number of facts aggregated, max sentence length.

MT & Alignment

- Ulrich Germann, Mike Jahr, Kevin Knight, Daniel Marcu, and Kenji Yamada. Fast decoding and optimal decoding for machine translation. ACL 2001.
- John DeNero and Dan Klein. The Complexity of Phrase Alignment Problems. ACL-HLT-2008.

Summary of Examples

- We have shown several different NLP solution that make use of CCMs.
- Examples vary in the way models are learned.
- In all cases, constraints can be expressed in a high level language, and then transformed into linear inequalities.
- Learning based Java (LBJ) [Rizzolo&Roth ’07, ’10] describe an automatic way to compile high level description of constraint into linear inequalities.

Solvers

- All applications presented so far used ILP for inference.
- People used different solvers
  - □ Xpress-MP
  - □ GLPK
  - □ lpsolve
  - □ R
  - □ Mosek
  - □ CPLEX

This Tutorial: ILP & Constrained Conditional Models

- Part 2: **How to pose the inference problem** (45 minutes)
  - □ Introduction to ILP
  - □ Posing NLP Problems as ILP problems
    - 1. Sequence tagging (HMM/CRF + global constraints)
    - 2. SRL (Independent classifiers + Global Constraints)
    - 3. Sentence Compression (Language Model + Global Constraints)
  - □ Less detailed examples
    - 1. Co-reference
    - 2. A bunch more ...
- Part 3: **Inference Algorithms (ILP & Search)** (15 minutes)
  - □ Compiling knowledge to linear inequalities
  - □ Other algorithms like search

BREAK

Learning Based Java: Translating to ILP

```
constraint References(SRLSentence sentence)
{
  for (int i = 0; i < sentence.verbCount(); ++i)
  {
    ParseTreeWord verb = sentence.getVerb(i);
    LinkedList forVerb = sentence.getCandidates(verb);

    (exists (Argument a in forVerb) ArgumentTypeLearner(a) :: "R-A0")
      => (exists (Argument a in forVerb) ArgumentTypeLearner(a) :: "A0");
    (exists (Argument a in forVerb) ArgumentTypeLearner(a) :: "R-A1")
      => (exists (Argument a in forVerb) ArgumentTypeLearner(a) :: "A1");
```

- Constraint syntax based on First Order Logic
  - □ Declarative; interspersed within pure Java
  - □ Grounded in the program's Java objects
- Automatic run-time translation to linear inequalities
  - □ Creates auxiliary variables
  - □ Resulting ILP size is linear in size of propositionalization

ILP: Speed Can Be an Issue

- Inference problems in NLP
  - □ Sometimes large problems are actually easy for ILP
    - E.g. Entities-Relations
    - Many of them are not "difficult"
- When ILP isn't fast enough, and one needs to resort to approximate solutions.
- The Problem: General Solvers vs. Specific Solvers
  - □ ILP is a very general solver
  - □ But, sometimes the structure of the problem allows for simpler inference algorithms.
- Next we give examples for both cases.

Example 1: Search based Inference for SRL

- The objective function

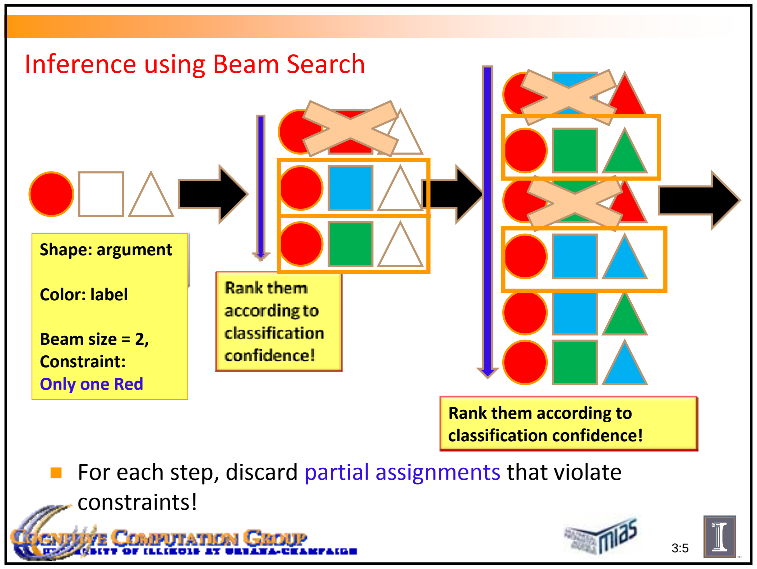
$$\max \sum_{i,j} c_{ij} \cdot x_{ij}$$

Maximize summation of the scores subject to linguistic constraints

Classification confidence

Indicator variable
assigns the j-th class for the i-th token

- **Constraints**
  - □ Unique labels
  - □ No overlapping or embedding
  - □ If verb is of type A, no argument of type B
  - □ …
- **Intuition**: check constraints' violations on **partial assignments**



Inference using Beam Search

Shape: argument

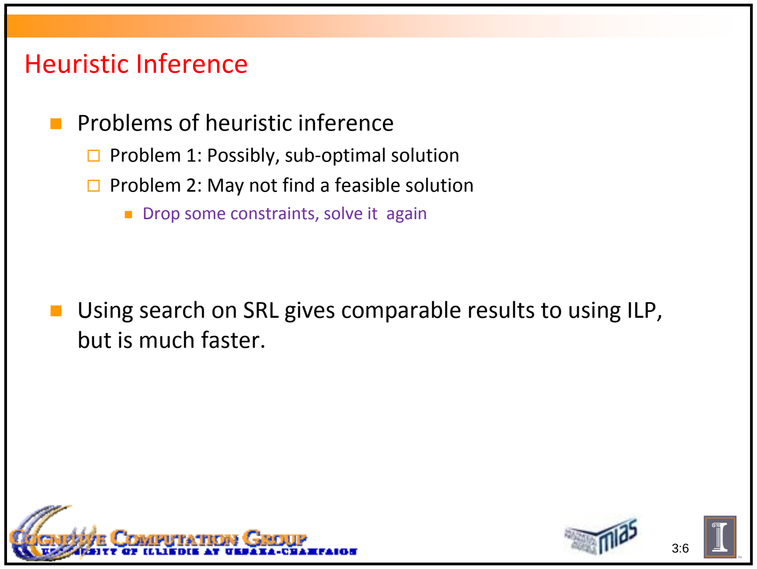
Color: label

Beam size = 2,
Constraint:
Only one Red

Rank them according to classification confidence!

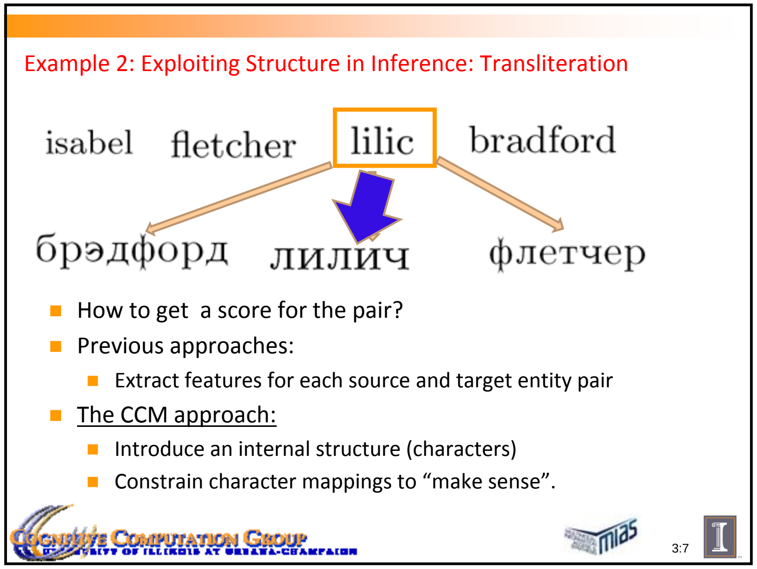
Rank them according to classification confidence!

- For each step, discard partial assignments that violate constraints!



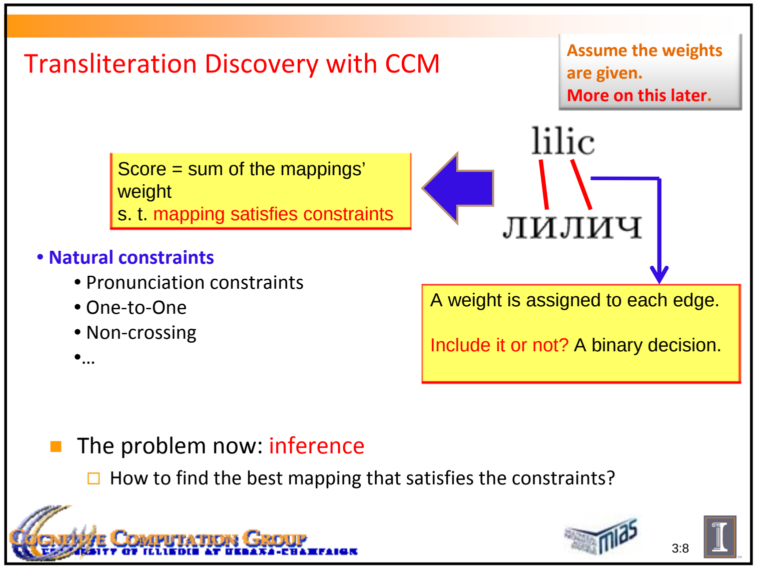
Heuristic Inference

- Problems of heuristic inference
  - □ Problem 1: Possibly, sub-optimal solution
  - □ Problem 2: May not find a feasible solution
    - Drop some constraints, solve it again
- Using search on SRL gives comparable results to using ILP, but is much faster.



Example 2: Exploiting Structure in Inference: Transliteration

- How to get a score for the pair?
- Previous approaches:
  - Extract features for each source and target entity pair
- The CCM approach:
  - Introduce an internal structure (characters)
  - Constrain character mappings to “make sense”.



Transliteration Discovery with CCM

Assume the weights are given. More on this later.

Score = sum of the mappings’ weight
s. t. mapping satisfies constraints

- **Natural constraints**
  - Pronunciation constraints
  - One-to-One
  - Non-crossing
  - …

A weight is assigned to each edge.

Include it or not? A binary decision.

- The problem now: inference
  - □ How to find the best mapping that satisfies the constraints?

Finding The Best Character Mappings

- **An Integer Linear Programming Problem**

$$\max \sum_{i\in S,\, j\in T} c_{ij} x_{ij}$$

Maximize the mapping score

$$0 \le x_{ij} \le 1,\ x_{ij} \in Z$$

Pronunciation constraint

$$\forall (i, j) \in B,\ x_{ij} = 0,$$

One-to-one constraint

$$\forall i, \sum_j x_{ij} = 1,$$

Non-crossing constraint

$$\forall i, j, k, m, i > k, m > j,$$
$$x_{ij} + x_{km} \le 1$$
…

- **Is this the best inference algorithm?**

Finding The Best Character Mappings

- **A Dynamic Programming Algorithm**

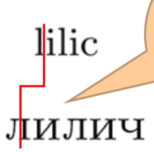
Maximize the mapping score

Restricted mapping constraints

One-to-one constraint

Non-crossing constraint

- **Exact** and fast!



We can decompose the inference problem into two parts

Take Home Message:
Although ILP can solve most problems, the fastest inference algorithm depends on the constraints and can be simpler

Other Inference Options

- **Constraint Relaxation Strategies**
  - □ Try Linear Programming
    - **[Roth and Yih, ICML 2005]**
  - □ Cutting plane algorithms 🡐 do not use all constraints at first
    - Dependency Parsing: Exponential number of constraints
    - **[Riedel and Clarke, EMNLP 2006]**
- **Other search algorithms**
  - □ ***A-star***, Hill Climbing…
  - □ Gibbs Sampling Inference **[Finkel et. al, ACL 2005]**
    - Named Entity Recognition: enforce long distance constraints
    - Can be considered as : Learning + Inference
    - One type of constraints only

Inference Methods – Summary

- Why ILP? A powerful way to **formalize** the problems
  - □ **However, not necessarily the best algorithmic solution**
- Heuristic inference algorithms are useful sometimes!
  - □ Beam search
  - □ Other approaches: annealing …
- Sometimes, a specific inference algorithm can be designed
  - □ According to your constraints

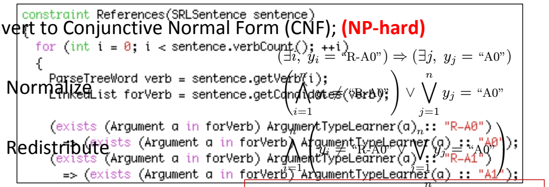
Constrained Conditional Models – 1st Part

- Introduced CCMs as a formalisms that allows us to
  - □ Learn simpler models than we would otherwise
  - □ Make decisions with expressive models, augmented by declarative constraints
- Focused on modeling – posing NLP problems as ILP problems
  - □ 1. Sequence tagging (HMM/CRF + global constraints)
  - □ 2. SRL (Independent classifiers + Global Constraints)
  - □ 3. Sentence Compression (Language Model + Global Constraints)
- Described Inference
  - □ From declarative constraints to ILP; solving ILP, exactly & approximately
- Next half – Learning
  - □ Supervised setting, and supervision-lean settings

Extra Slides

Learning Based Java: Translating to ILP (1/2)

- Modeling language for use with Java
- Classifiers use other classifiers as feature extractors
- Constraints written in FOL over Java objects
  - □ Automatically translated to linear inequalities at run-time



```
constraint References(SRLSentence sentence)
{
  for (int i = 0; i < sentence.verbCount(); ++i)
  {
    ParseTreeWord verb = sentence.getVerb(i);
    LinkedList forVerb = sentence.getCandidates(verb);

    (exists (Argument a in forVerb) ArgumentTypeLearner(a) :: "R-A0")
      => (exists (Argument a in forVerb) ArgumentTypeLearner(a) :: "A0");
    (exists (Argument a in forVerb) ArgumentTypeLearner(a) :: "R-A1")
      => (exists (Argument a in forVerb) ArgumentTypeLearner(a) :: "A1");
```

- Convert to Conjunctive Normal Form (CNF); **(NP-hard)**

$$(\exists i,\ y_i = \text{“R-A0”}) \Rightarrow (\exists j,\ y_j = \text{“A0”})$$

- Normalize

$$\left(\bigwedge_{i=1}^{n} y_i \neq \text{“R-A0”}\right) \vee \bigvee_{j=1}^{n} y_j = \text{“A0”}$$

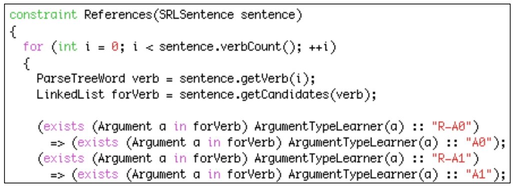
- Redistribute

$$\bigwedge_{i=1}^{n}\left(y_i \neq \text{“R-A0”} \vee \bigvee_{j=1}^{n} y_j = \text{“A0”}\right)$$

- Create indicator variables

$$\forall i,\ \left(1 - 1_{\{y_i = \text{“R-A0”}\}}\right) + \sum_{j=1}^{n} 1_{\{y_j = \text{“A0”}\}} \geq 1$$

Learning Based Java: Translating to ILP (2/2)



```
constraint References(SRLSentence sentence)
{
  for (int i = 0; i < sentence.verbCount(); ++i)
  {
    ParseTreeWord verb = sentence.getVerb(i);
    LinkedList forVerb = sentence.getCandidates(verb);

    (exists (Argument a in forVerb) ArgumentTypeLearner(a) :: "R-A0")
      => (exists (Argument a in forVerb) ArgumentTypeLearner(a) :: "A0");
    (exists (Argument a in forVerb) ArgumentTypeLearner(a) :: "R-A1")
      => (exists (Argument a in forVerb) ArgumentTypeLearner(a) :: "A1");
```

- Create temporary variables

$$(\exists i,\ y_i = \text{“R-A0”}) \Rightarrow (\exists j,\ y_j = \text{“A0”})$$

$$\left(\bigwedge_{i=1}^{n} y_i \neq \text{“R-A0”}\right) \vee \bigvee_{j=1}^{n} y_j = \text{“A0”}$$

$$t_1 \vee \bigvee_{j=1}^{n} y_j = \text{“A0”},\ \text{where } t_1 \equiv \bigwedge_{i=1}^{n} y_i \neq \text{“R-A0”}$$

$$\forall i,\ \ 1_{t_1} + \sum_{j=1}^{n} 1_{\{y_j = \text{“A0”}\}} \geq 1$$

The original constraint

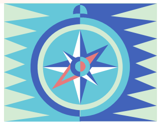
$$n - \sum_{i=1}^{n} 1_{\{y_i = \text{“R-A0”}\}} \geq n 1_{t_1}$$

$$1 - \sum_{i=1}^{n} 1_{\{y_i = \text{“R-A0”}\}} \leq 1_{t_1}$$

Definition of t_1

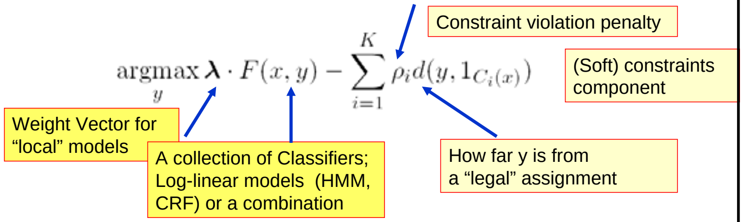
- Every temporary variable is defined by exactly 2 inequalities

Where Are We ?



- We hope we have already convinced you that
  - □ Using constraints is a good idea for addressing NLP problems
  - □ Constrained conditional models provide a good platform
- We were talking about using expressive constraints
  - □ To improve existing models
  - □ Learning + Inference
  - □ The problem: inference
- A powerful inference tool: Integer Linear Programming
  - □ SRL, co-ref, summarization, entity-and-relation…
  - □ Easy to inject domain knowledge

Constrained Conditional Model : Inference



$$\operatorname{argmax}_y \boldsymbol{\lambda} \cdot F(x,y) - \sum_{i=1}^{K} \rho_i d(y, 1_{C_i(x)})$$

Constraint violation penalty

(Soft) constraints component

Weight Vector for "local" models

A collection of Classifiers; Log-linear models (HMM, CRF) or a combination

How far y is from a "legal" assignment

How to solve?

This is an Integer Linear Program

Solving using ILP packages gives an exact solution.

Search techniques are also possible

How to train?

How to decompose the global objective function?

Should we incorporate constraints in the learning process?

Advantages of ILP Solvers: Review

- ILP is Expressive: We can solve many inference problems
  - □ Converting inference problems into ILP is easy
- ILP is Easy to Use: Many available packages
  - □ (Open Source Packages): LPSolve, GLPK, …
  - □ (Commercial Packages): XPressMP, Cplex
  - □ No need to write optimization code!
- Why should we consider other inference options?

This Tutorial: ILP & Constrained Conditional Models (Part II)

- Part 4: Training Issues (80 min)
  - □ Learning models
    - Independently of constraints (L+I); Jointly with constraints (IBT)
    - Decomposed to simpler models
  - □ Learning constraints' penalties
    - Independently of learning the model
    - Jointly, along with learning the model
  - □ Dealing with lack of supervision
    - Constraints Driven Semi-Supervised learning (CODL)
    - Indirect Supervision
  - □ Learning Constrained Latent Representations

Training Constrained Conditional Models

Decompose Model

$$\underset{y}{\operatorname{argmax}}\, \boldsymbol{\lambda}\cdot F(x,y) - \sum_{i=1}^{K} \rho_i d(y, 1_{C_i(x)})$$

Decompose Model from constraints

- Learning model
  - □ Independently of the constraints (L+I)
  - □ Jointly, in the presence of the constraints (IBT)
  - □ Decomposed to simpler models
- Learning constraints' penalties
  - □ Independently of learning the model
  - □ Jointly, along with learning the model
- Dealing with lack of supervision
  - □ Constraints Driven Semi-Supervised learning (CODL)
  - □ Indirect Supervision
- Learning Constrained Latent Representations

Where are we?

Modeling & Algorithms for Incorporating Constraints

- □ Showed that CCMs allow for formalizing many problems
- □ Showed several ways to incorporate global constraints in the decision.

Training: Coupling vs. Decoupling Training and Inference.

- □ Incorporating global constraints is important but
- □ Should it be done only at evaluation time or also at training time?
- □ How to decompose the objective function and train in parts?
- □ Issues related to:
  - **Modularity, efficiency and performance, availability of training data**
  - **Problem specific considerations**

Training Constrained Conditional Models

$$\underset{y}{\operatorname{argmax}}\, \boldsymbol{\lambda}\cdot F(x,y) - \sum_{i=1}^{K} \rho_i d(y, 1_{C_i(x)})$$

Decompose Model from constraints

- Learning model
  - □ Independently of the constraints (L+I)
  - □ Jointly, in the presence of the constraints (IBT)
- First Term: Learning from data (could be further decomposed)
- Second Term: Guiding the model by constraints
  - □ Can choose if constraints' weights trained, when and how, or taken into account only in evaluation.
  - □ At this point – the case of hard constraints

Comparing Training Methods

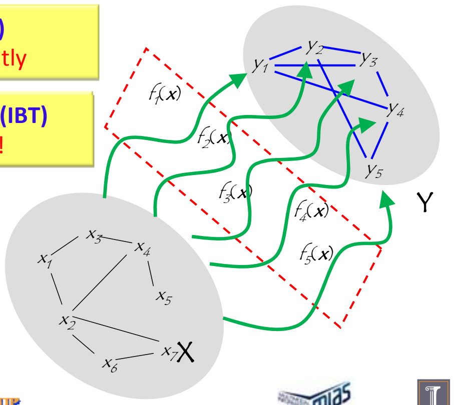
- Option 1: Learning + Inference (with Constraints)
  - Ignore constraints during training
- Option 2: Inference (with Constraints) Based Training
  - Consider constraints during training
- In both cases: Global Decision Making with Constraints
- Question: Isn't Option 2 always better?
- Not so simple…
  - Next, the "Local model story"

Training Methods

Learning + Inference (L+I)
Learn models independently

Inference Based Training (IBT)
Learn all models together!

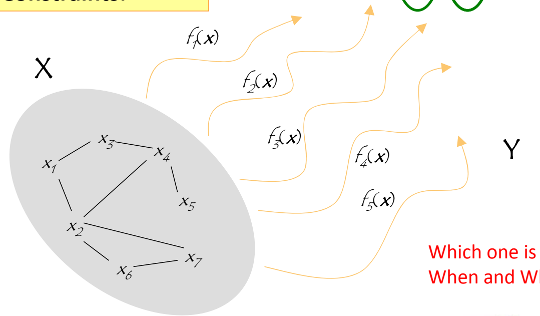
Intuition
Learning with constraints may make learning more difficult



Training with Constraints

Example: Perceptron-based Global Learning

True Global Labeling	Y	-1	1	-1	-1	1
Apply Constraints:	Y'	-1	1	11	1	1



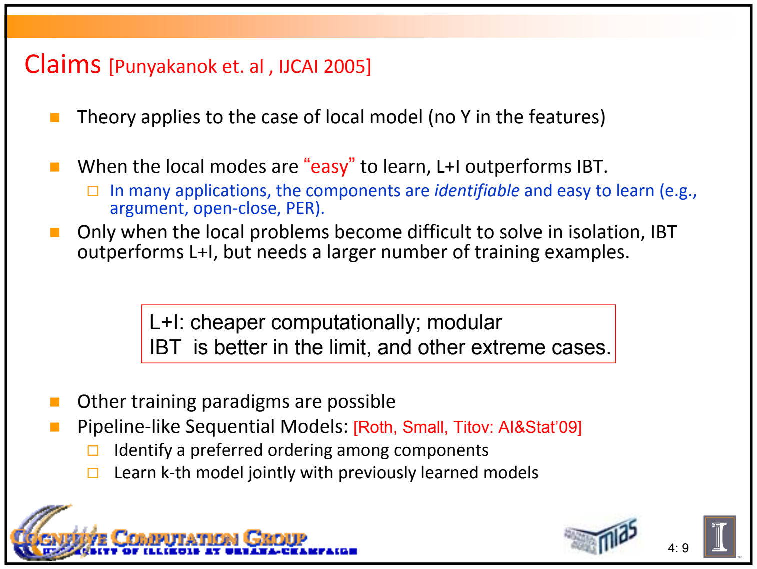
Which one is better?
When and Why?

L+I & IBT: General View – Structured Perceptron

- Graphics for the case: F(x,y) = F(x)

```
For each iteration
    For each (X, Y_GOLD ) in the training data
        $Y_{PRED} = \operatorname{argmax}_y \boldsymbol{\lambda} \cdot F(x,y) - \boxed{\sum_{i=1}^{K} \rho_i d(y, 1_{C_i(x)})}$
        If Y_PRED != Y_GOLD
            $\lambda = \lambda + F(X, \mathbf{Y}_{GOLD}) - F(X, \mathbf{Y}_{PRED})$
        endif
    endfor
```

The difference between L+I and IBT

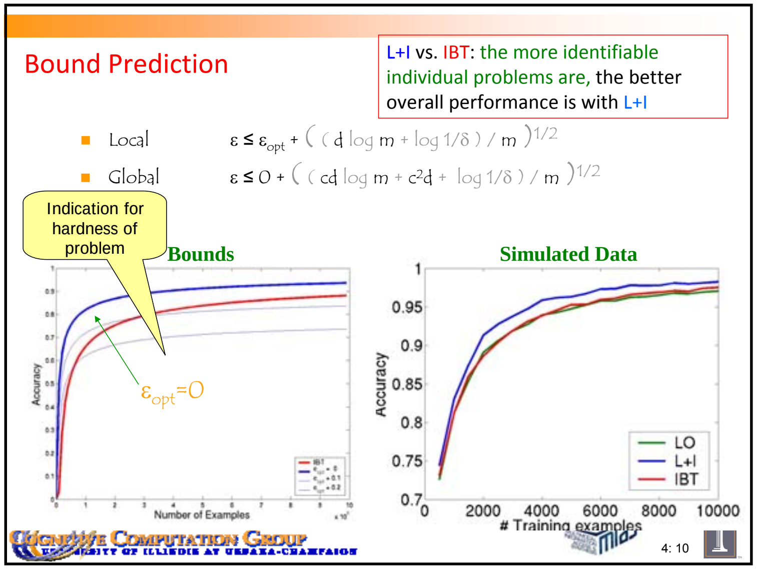


Claims [Punyakanok et. al , IJCAI 2005]

- Theory applies to the case of local model (no Y in the features)
- When the local modes are "easy" to learn, L+I outperforms IBT.
  - □ In many applications, the components are *identifiable* and easy to learn (e.g., argument, open-close, PER).
- Only when the local problems become difficult to solve in isolation, IBT outperforms L+I, but needs a larger number of training examples.

L+I: cheaper computationally; modular
IBT is better in the limit, and other extreme cases.

- Other training paradigms are possible
- Pipeline-like Sequential Models: [Roth, Small, Titov: AI&Stat'09]
  - □ Identify a preferred ordering among components
  - □ Learn k-th model jointly with previously learned models



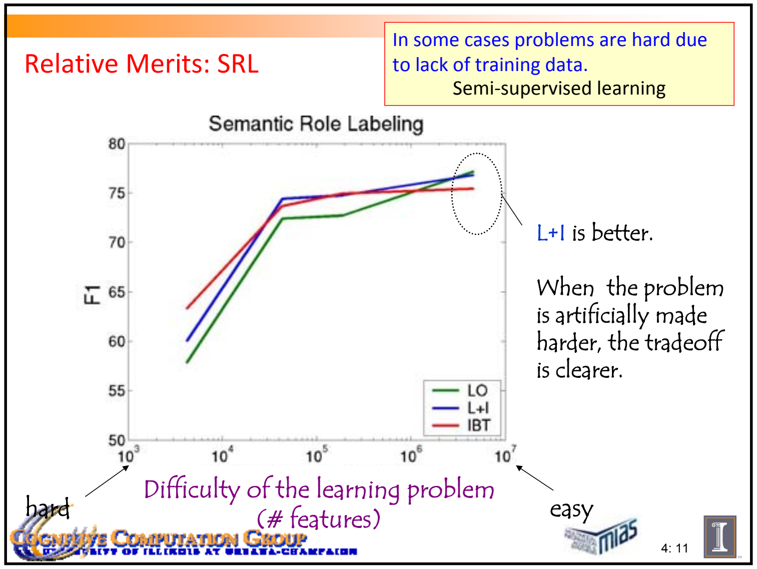
Bound Prediction

L+I vs. IBT: the more identifiable individual problems are, the better overall performance is with L+I

- Local $\varepsilon \le \varepsilon_{opt} + ((d \log m + \log 1/\delta) / m)^{1/2}$
- Global $\varepsilon \le 0 + ((cd \log m + c^2 d + \log 1/\delta) / m)^{1/2}$

Indication for hardness of problem

$\varepsilon_{opt}=0$



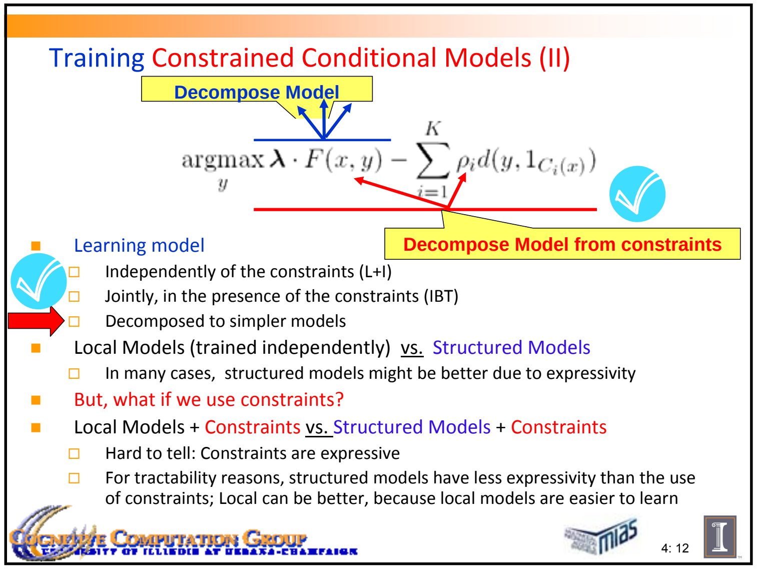
Relative Merits: SRL

In some cases problems are hard due to lack of training data.
Semi-supervised learning

L+I is better.

When the problem is artificially made harder, the tradeoff is clearer.

Difficulty of the learning problem (# features)



Training Constrained Conditional Models (II)

Decompose Model

$$\underset{y}{\operatorname{argmax}} \boldsymbol{\lambda} \cdot F(x,y) - \sum_{i=1}^{K} \rho_i d(y, 1_{C_i(x)})$$

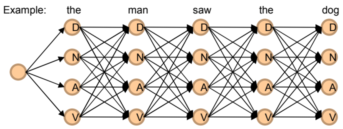
Decompose Model from constraints

- Learning model
  - □ Independently of the constraints (L+I)
  - □ Jointly, in the presence of the constraints (IBT)
  - □ Decomposed to simpler models
- Local Models (trained independently) vs. Structured Models
  - □ In many cases, structured models might be better due to expressivity
- But, what if we use constraints?
- Local Models + Constraints vs. Structured Models + Constraints
  - □ Hard to tell: Constraints are expressive
  - □ For tractability reasons, structured models have less expressivity than the use of constraints; Local can be better, because local models are easier to learn

Recall: Example 1: Sequence Tagging (HMM/CRF)

HMM / CRF:

$$\mathbf{y}^* = \operatorname*{argmax}_{\mathbf{y}\in\mathcal{Y}} P(y_0)P(x_0|y_0)\prod_{i=1}^{n-1} P(y_i|y_{i-1})P(x_i|y_i)$$



As an ILP:

$$\text{maximize} \sum_{y\in\mathcal{Y}} \lambda_{0,y} 1_{\{y_0=y\}} + \sum_{i=1}^{n-1}\sum_{y\in\mathcal{Y}}\sum_{y'\in\mathcal{Y}} \lambda_{i,y,y'} 1_{\{y_i=y\ \wedge\ y_{i-1}=y'\}}$$

$$\lambda_{0,y} = \log(P(y)) + \log(P(x_0|y))$$
$$\lambda_{i,y,y'} = \log(P(y|y')) + \log(P(x_i|y))$$

subject to

$$\sum_{y\in\mathcal{Y}} 1_{\{y_0=y\}} = 1$$

Discrete predictions

$$\forall y,\ \ 1_{\{y_0=y\}} = \sum_{y'\in\mathcal{Y}} 1_{\{y_0=y\ \wedge\ y_1=y'\}}$$

$$\forall y, i>1\ \ \sum_{y'\in\mathcal{Y}} 1_{\{y_{i-1}=y'\ \wedge\ y_i=y\}} = \sum_{y''\in\mathcal{Y}} 1_{\{y_i=y\ \wedge\ y_{i+1}=y''\}}$$

Feature consistency

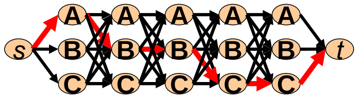
$$1_{\{y_0=\text{“V”}\}} + \sum_{i=1}^{n-1}\sum_{y\in\mathcal{Y}} 1_{\{y_{i-1}=y\ \wedge\ y_i=\text{“V”}\}} \geq 1$$

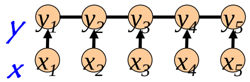
There must be a verb!

Example: CRFs are CCMs

But, you can do better

- Consider a common model for sequential inference: HMM/CRF
  - Inference in this model is done via the Viterbi Algorithm.

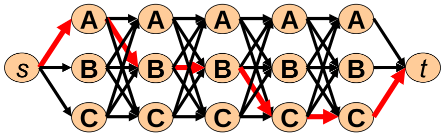




- Viterbi is a special case of the Linear Programming based Inference.
  - It is a shortest path problem, which is a LP, with a canonical matrix that is totally unimodular. Therefore, you get integrality constraints for free.
  - One can now incorporate non-sequential/expressive/declarative constraints by modifying this canonical matrix
    - No value can appear twice; a specific value must appear at least once; A→B
  - And, run the inference as an ILP inference.

Learn a rather simple model; make decisions with a more expressive model

Example: Semantic Role Labeling Revisited



- Sequential Models
  - Conditional Random Field
  - Global perceptron
- Training: Sentence based
- Testing: Find best global assignment (shortest path)
  - **+ with constraints**

- Local Models
  - Logistic Regression
  - Avg. Perceptron
- Training: Token based.
- Testing: Find best assignment locally
  - **+ with constraints (Global)**

Which Model is Better? Semantic Role Labeling

- Experiments on SRL: [Roth and Yih, ICML 2005]
  - Story: Inject expressive Constraints into conditional random field

	Sequential Models			**Local**
	L+I		IBT	**L+I**
Model	**CRF**	**CRF-D**	**CRF-IBT**	**Avg. P**

Local Models are now better than Sequential Models! (With constraints)

Summary: Training Methods – Supervised Case

- Many choices for training a CCM
  - □ Learning + Inference (Training w/o constraints; add constraints later)
  - □ Inference based Learning (Training with constraints)
- Based on this, what kind of models should you use?
  - Decomposing models can be better that structured models
- Advantages of L+I
  - □ Require fewer training examples
  - □ More efficient; most of the time, better performance
  - □ Modularity; easier to incorporate already learned models.
- Next: Soft Constraints; Supervision-lean models

Training Constrained Conditional Models

$$\underset{y}{\operatorname{argmax}}\, \boldsymbol{\lambda} \cdot F(x,y) - \sum_{i=1}^{K} \boxed{\rho_i} d(y, 1_{C_i(x)})$$

- Learning model
  - □ Independently of the constraints (L+I)
  - □ Jointly, in the presence of the constraints (IBT)
  - □ Decomposed to simpler models
- Learning constraints' penalties
  - □ Independently of learning the model
  - □ Jointly, along with learning the model
- Dealing with lack of supervision
  - □ Constraints Driven Semi-Supervised learning (CODL)
  - □ Indirect Supervision
- Learning Constrained Latent Representations

Soft Constraints

$$-\sum_{i=1}^{K} \rho_k d(y, 1_{C_i(x)})$$

- Hard Versus Soft Constraints
  - □ Hard constraints: Fixed Penalty $\rho_i = \infty$
  - □ Soft constraints: Need to set the penalty
- Why soft constraints?
  - □ Constraints might be violated by gold data
  - □ Some constraint violations are more serious
  - □ An example can violate a constraint multiple times!
  - □ Degree of violation is only meaningful when constraints are soft!

Example: Information extraction

Lars Ole Andersen . Program analysis and specialization for the C Programming language. PhD thesis. DIKU , University of Copenhagen, May 1994 .

Prediction result of a trained HMM

[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 .

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*
- ....

Degree of Violations

One way: Count how many times the assignment y violated the constraint

$$d(y, 1_{C(x)}) = \sum_{j=1}^{T} \phi_C(y_j)$$

$$\phi_C(y_j) = \begin{cases} \text{1 - if assigning } y_i \text{ to } x_i \text{ violates the constraint } C \text{ with respect to assignment } (x_1,..,x_{i-1};y_1,\dots,y_{i-1}) \\ \text{0 - otherwise} \end{cases}$$

State transition must occur on punctuations ⇨ $\forall i, y_{i-1} \neq y_i \Rightarrow x_{i-1}$ is a punctuation

Lars	Ole	Andersen	.
AUTH	BOOK	EDITOR	EDITOR
$\Phi_c(y_1)=0$	$\Phi_c(y_2)=1$	$\Phi_c(y_3)=1$	$\Phi_c(y_4)=0$

$\sum \Phi_c(y_j) = 2$

Reason for using degree of violation

- An assignment might violate a constraint multiple times
- Allow us to chose a solution with fewer constraint violations

Lars	Ole	Andersen	.
AUTH	AUTH	EDITOR	EDITOR
$\Phi_c(y_1)=0$	$\Phi_c(y_2)=0$	$\Phi_c(y_3)=1$	$\Phi_c(y_4)=0$

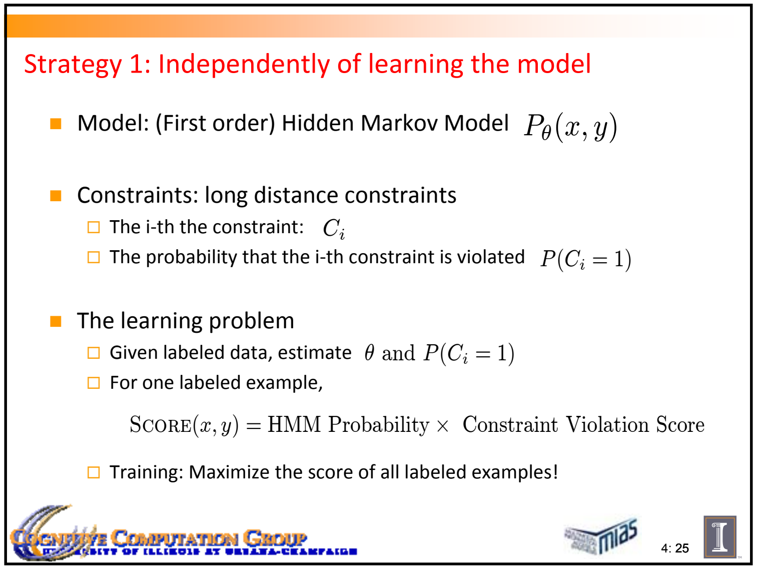
The first one is better because of $d(y,1_{c(x)})$!

Lars	Ole	Andersen	.
AUTH	BOOK	EDITOR	EDITOR
$\Phi_c(y_1)=0$	$\Phi_c(y_2)=1$	$\Phi_c(y_3)=1$	$\Phi_c(y_4)=0$

Learning the penalty weights

$$\boxed{\boxed{\lambda \cdot F(x,y)} - \boxed{\sum_{i=1}^{K} \rho_k d(y, 1_{C_i(x)})}}$$

- **Strategy 1: Independently of learning the model**
  - □ Handle the learning parameters λ and the penalty ρ separately
  - □ Learn a feature model and a constraint model
  - □ Similar to L+I, but also learn the penalty weights
  - □ Keep the model simple
- **Strategy 2: Jointly, along with learning the model**
  - □ Handle the learning parameters λ and the penalty ρ together
  - □ Treat soft constraints as high order features
  - □ Similar to IBT, but also learn the penalty weights

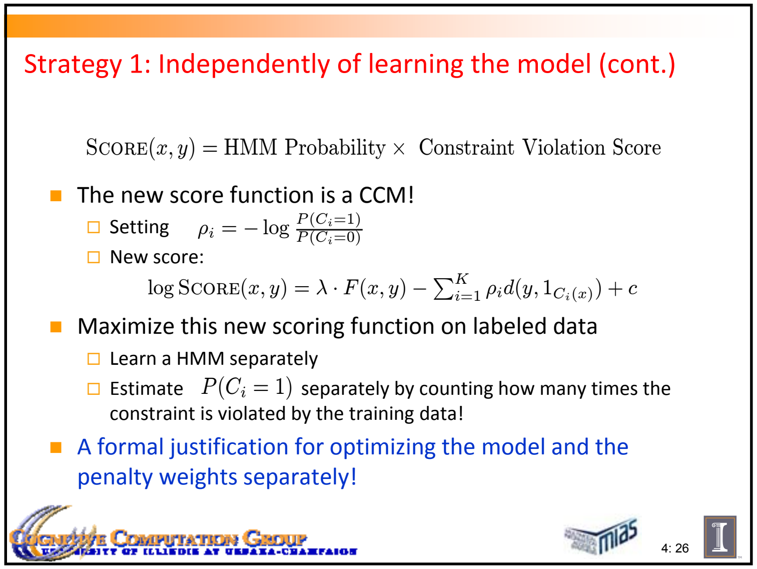


Strategy 1: Independently of learning the model

- Model: (First order) Hidden Markov Model $P_\theta(x,y)$
- Constraints: long distance constraints
  - □ The i-th the constraint: C_i
  - □ The probability that the i-th constraint is violated $P(C_i = 1)$
- The learning problem
  - □ Given labeled data, estimate θ and $P(C_i = 1)$
  - □ For one labeled example,

$$\textsc{Score}(x,y) = \text{HMM Probability} \times \text{ Constraint Violation Score}$$

  - □ Training: Maximize the score of all labeled examples!



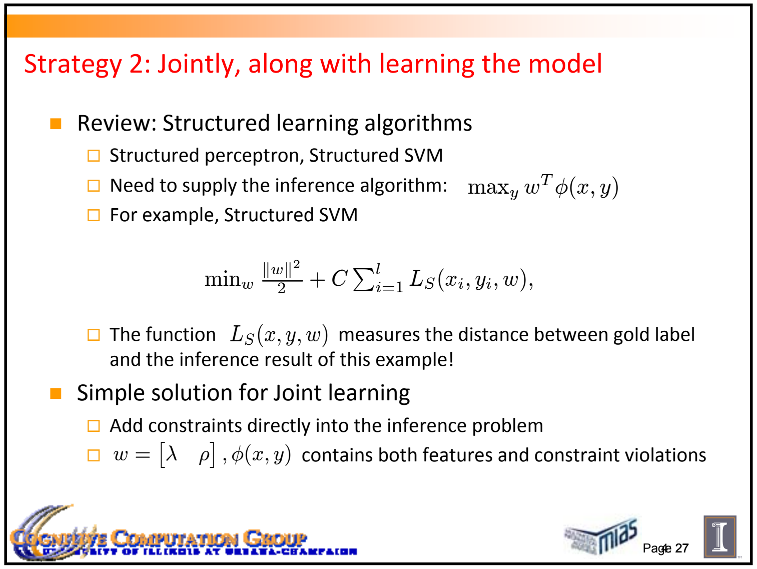
Strategy 1: Independently of learning the model (cont.)

$$\textsc{Score}(x,y) = \text{HMM Probability} \times \text{ Constraint Violation Score}$$

- The new score function is a CCM!
  - □ Setting $\rho_i = -\log \frac{P(C_i=1)}{P(C_i=0)}$
  - □ New score:

$$\log \textsc{Score}(x,y) = \lambda \cdot F(x,y) - \sum_{i=1}^{K} \rho_i d(y, 1_{C_i(x)}) + c$$

- Maximize this new scoring function on labeled data
  - □ Learn a HMM separately
  - □ Estimate $P(C_i = 1)$ separately by counting how many times the constraint is violated by the training data!
- A formal justification for optimizing the model and the penalty weights separately!

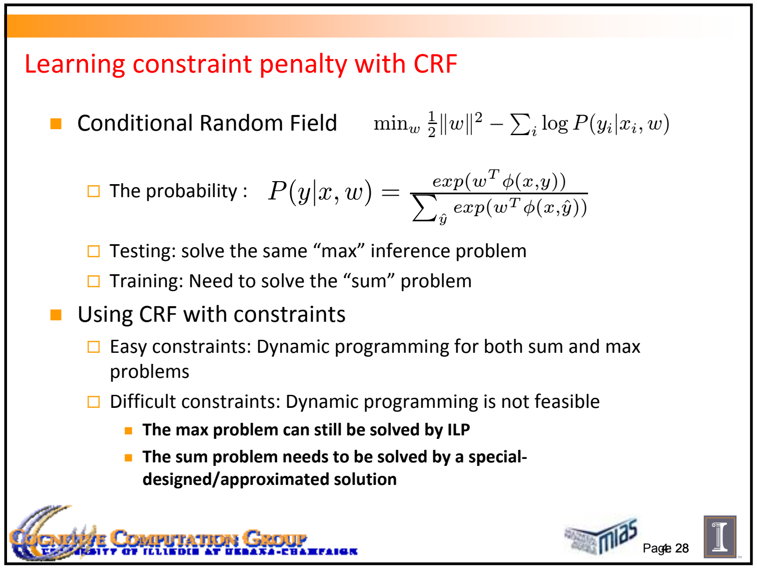


Strategy 2: Jointly, along with learning the model

- Review: Structured learning algorithms
  - □ Structured perceptron, Structured SVM
  - □ Need to supply the inference algorithm: $\max_y w^T \phi(x,y)$
  - □ For example, Structured SVM

$$\min_w \frac{\|w\|^2}{2} + C \sum_{i=1}^{l} L_S(x_i, y_i, w),$$

  - □ The function $L_S(x,y,w)$ measures the distance between gold label and the inference result of this example!
- Simple solution for Joint learning
  - □ Add constraints directly into the inference problem
  - □ $w = \left[\lambda \quad \rho\right], \phi(x,y)$ contains both features and constraint violations



Learning constraint penalty with CRF

- Conditional Random Field $\min_w \frac{1}{2}\|w\|^2 - \sum_i \log P(y_i|x_i,w)$
  - □ The probability : $P(y|x,w) = \frac{exp(w^T\phi(x,y))}{\sum_{\hat{y}} exp(w^T\phi(x,\hat{y}))}$
  - □ Testing: solve the same “max” inference problem
  - □ Training: Need to solve the “sum” problem
- Using CRF with constraints
  - □ Easy constraints: Dynamic programming for both sum and max problems
  - □ Difficult constraints: Dynamic programming is not feasible
    - **The max problem can still be solved by ILP**
    - **The sum problem needs to be solved by a special-designed/approximated solution**

Summary: learning constraints' penalty weights

- Learning the penalty for soft constraints is important
  - Constraints can be violated by gold data
  - Degree of violation
  - Some constraints are more important
- Learning constraints' penalty weights
  - Learning penalty weights is a learning problem
  - Independent approach: fix the model
    - **Generative models + constraints**
  - Joint approach
    - **Treat constraints as long distance features**
    - **Max is generally easier than the sum problem**

Training Constrained Conditional Models

$$\operatorname*{argmax}_y \boldsymbol{\lambda} \cdot F(x,y) - \sum_{i=1}^{K} \rho_i d(y, 1_{C_i(x)})$$

- Learning model
  - Independently of the constraints (L+I)
  - Jointly, in the presence of the constraints (IBT)
  - Decomposed to simpler models
- Learning constraints' penalties
  - Independently of learning the model
  - Jointly, along with learning the model
- Dealing with lack of supervision
  - Constraints Driven Semi-Supervised learning (CODL)
  - Indirect Supervision
- Learning Constrained Latent Representations

Dealing with lack of supervision

- Goal of this tutorial: learning structured models
- Learning structured models requires annotating structures.
  - Very expensive process
- IDEA1: Can we use constraints as a supervision resource?
  - Setting: semi-supervised learning
- IDEA2: Can we use binary labeled data to learn a structured model?
  - Setting: indirect supervision (will explain latter)

Constraints As a Way To Encode Prior Knowledge

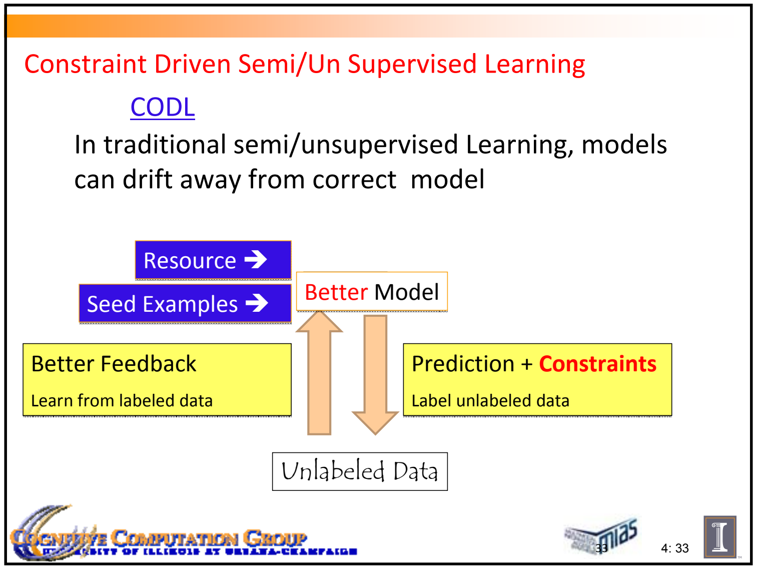
- **Consider encoding the knowledge that:**
  - Entities of type A and B cannot occur simultaneously in a sentence
- **The "Feature" Way**
  - Requires larger models

Need more training data

A effective way to inject knowledge

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

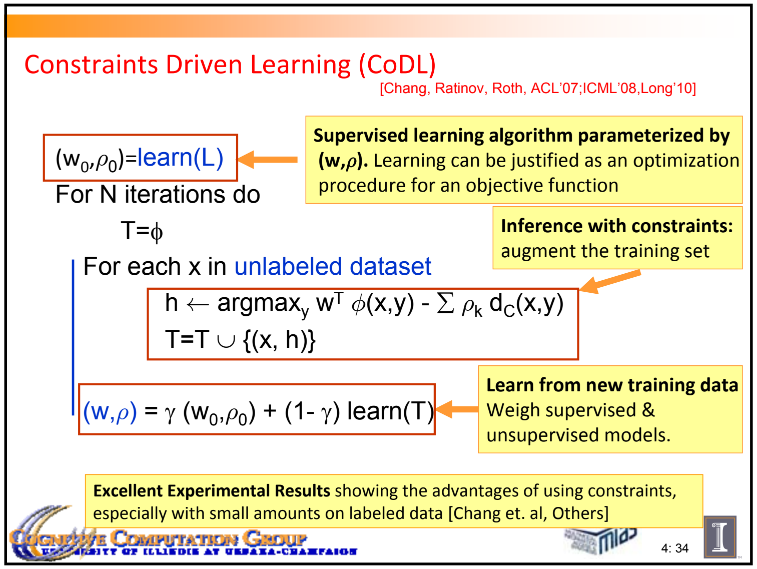
We can use constraints as a way to replace training data



Constraint Driven Semi/Un Supervised Learning

CODL

In traditional semi/unsupervised Learning, models can drift away from correct model



Constraints Driven Learning (CoDL)

[Chang, Ratinov, Roth, ACL'07;ICML'08,Long'10]

```
(w_0,ρ_0)=learn(L)
For N iterations do
    T=ϕ
    For each x in unlabeled dataset
        h ← argmax_y w^T ϕ(x,y) - Σ ρ_k d_C(x,y)
        T=T ∪ {(x, h)}
    (w,ρ) = γ (w_0,ρ_0) + (1- γ) learn(T)
```

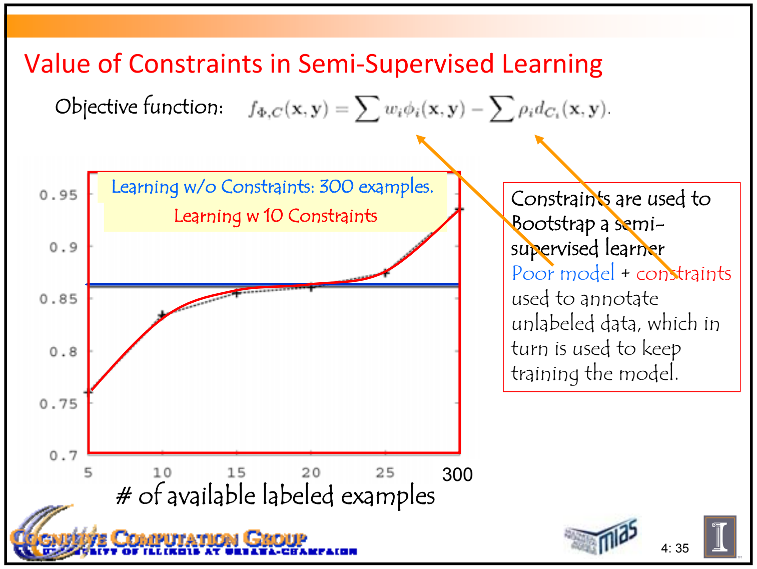
(w_0,ρ_0)=learn(L); $h \leftarrow \arg\max_y w^T \phi(x,y) - \sum \rho_k d_C(x,y)$; $T=T \cup \{(x, h)\}$; $(w,\rho) = \gamma (w_0,\rho_0) + (1-\gamma)$ learn(T)

Supervised learning algorithm parameterized by (w,ρ). Learning can be justified as an optimization procedure for an objective function

Inference with constraints: augment the training set

Learn from new training data Weigh supervised & unsupervised models.

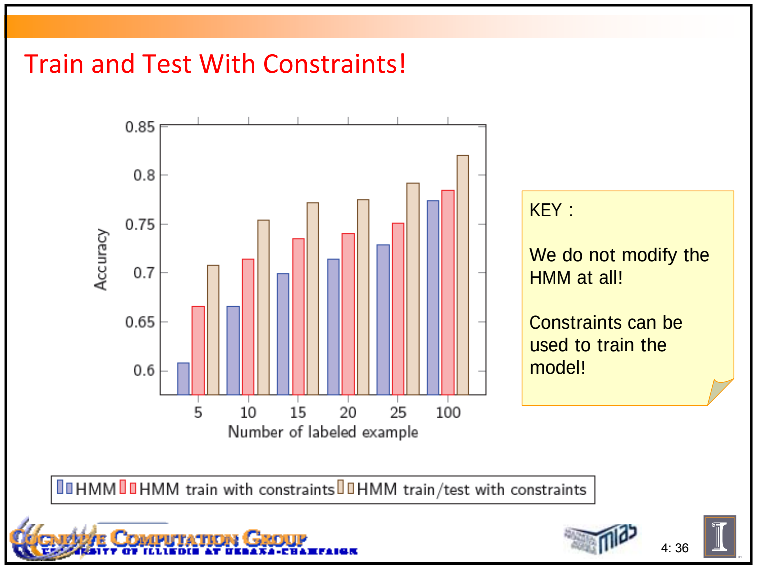
Excellent Experimental Results showing the advantages of using constraints, especially with small amounts on labeled data [Chang et. al, Others]



Value of Constraints in Semi-Supervised Learning

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

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.



Train and Test With Constraints!

KEY :

We do not modify the HMM at all!

Constraints can be used to train the model!

Exciting Recent Research

- **Generalized Expectation Criteria**
  - □ The idea: instead of labeling examples, label constraint features!
  - □ G. Mann and A. McCallum. JMLR, 2009
- **Posterior Regularization**
  - □ Reshape the posterior distribution with constraints
  - □ Instead of doing the “hard-EM” way, do the soft-EM way!
  - □ K. Ganchev, J. Graça, J. Gillenwater and B. Taskar, JMLR, 2010
- Different learning algorithms, the same idea;
  - □ Use constraints and unlabeled data as a form of supervision!
    - **To train a generative/discriminative model**
  - □ Word alignment, Information Extraction, document classification…

Word Alignment via Constraints

- **Posterior Regularization**
  K. Ganchev, J. Graça, J. Gillenwater and B. Taskar, JMLR, 2010
- Goal: find the word alignment between an English sentence and a French sentence
- Learning without using constraints
- □ Train a E-> F model (via EM), Train a F-> E model (via EM)
- □ Enforce the constraints at the end! One-to-one mapping, consistency
- Learning with constraints
- □ Enforce the constraints during training
- □ Use constraints to guide the learning procedure
- □ Running (soft) EM with constraints!

Probability Interpretation of CCM

- With a probabilistic model

$$\max_y \log P(x,y) - \sum_{k=1}^{m} \rho_i d(y, 1_{C_k(x)})$$

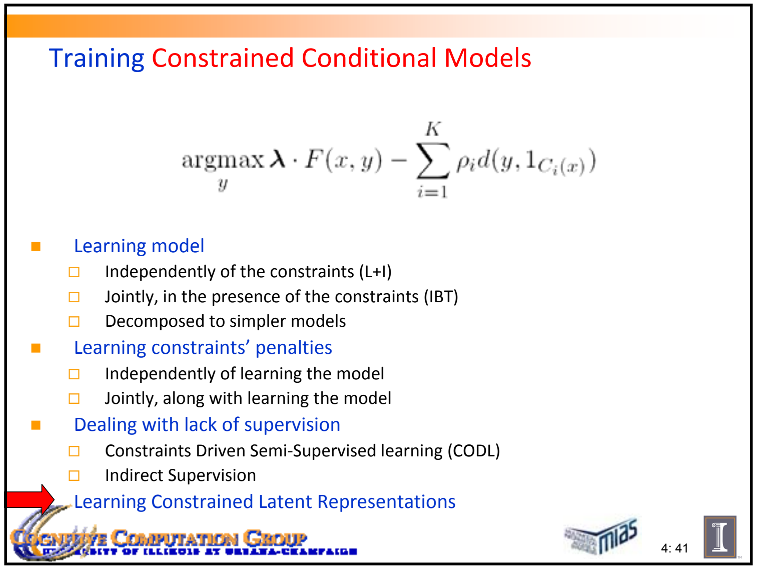
- Implication
  - □ $\text{New distribution} \propto P(x,y) \exp^{-\sum \rho_i d(y, 1_{C_k(x)})}$
- Constraint Driven Learning with full distribution
  - □ Step 1: find the best distribution that satisfy the “constraints”
  - □ Step 2: update the model according to the distribution

Theoretical Support

- In K. Ganchev, J. Graça, J. Gillenwater and B. Taskar, JMLR, 2010

Given any distribution P(x,y), the closest distribution that “satisfies the constraints” is in the form of CCM!

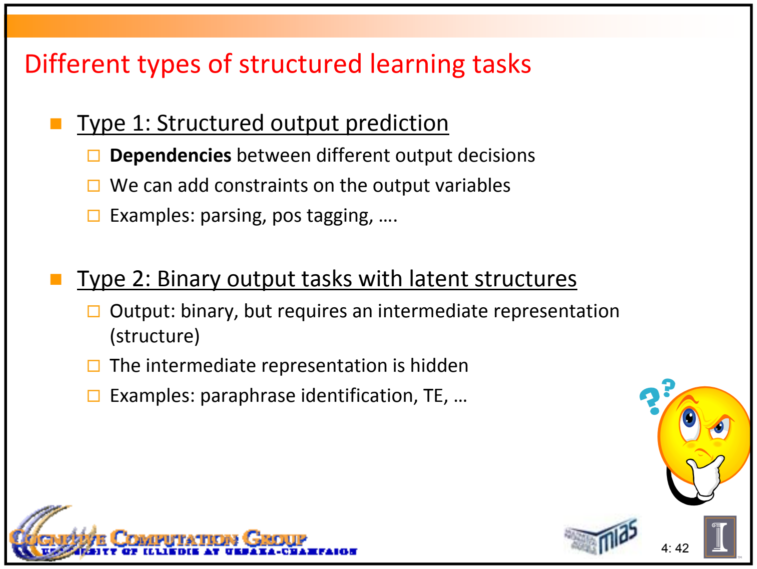
$$\text{New distribution} \propto P(x,y) \exp^{-\sum \rho_i d(y, 1_{C_k(x)})}$$



Training Constrained Conditional Models

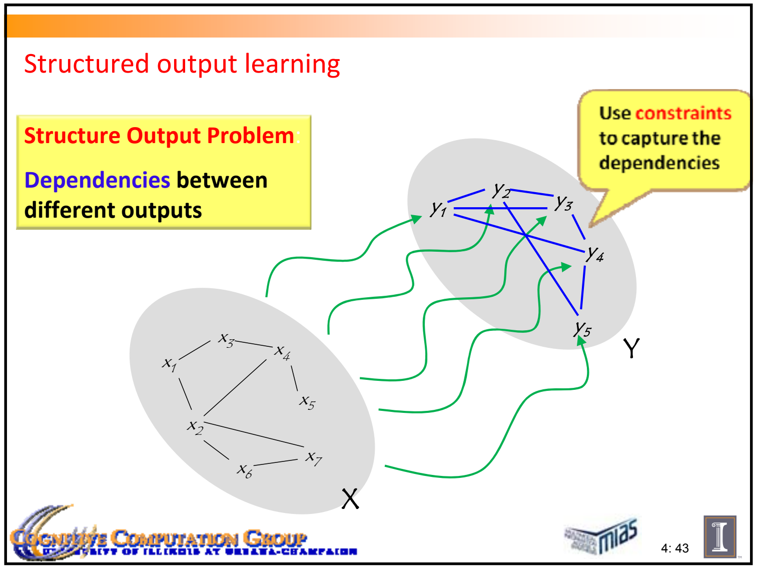
$$\underset{y}{\operatorname{argmax}}\, \boldsymbol{\lambda} \cdot F(x,y) - \sum_{i=1}^{K} \rho_i d(y, 1_{C_i(x)})$$

- Learning model
  - □ Independently of the constraints (L+I)
  - □ Jointly, in the presence of the constraints (IBT)
  - □ Decomposed to simpler models
- Learning constraints' penalties
  - □ Independently of learning the model
  - □ Jointly, along with learning the model
- Dealing with lack of supervision
  - □ Constraints Driven Semi-Supervised learning (CODL)
  - □ Indirect Supervision
- Learning Constrained Latent Representations



Different types of structured learning tasks

- Type 1: Structured output prediction
  - □ **Dependencies** between different output decisions
  - □ We can add constraints on the output variables
  - □ Examples: parsing, pos tagging, ….
- Type 2: Binary output tasks with latent structures
  - □ Output: binary, but requires an intermediate representation (structure)
  - □ The intermediate representation is hidden
  - □ Examples: paraphrase identification, TE, …

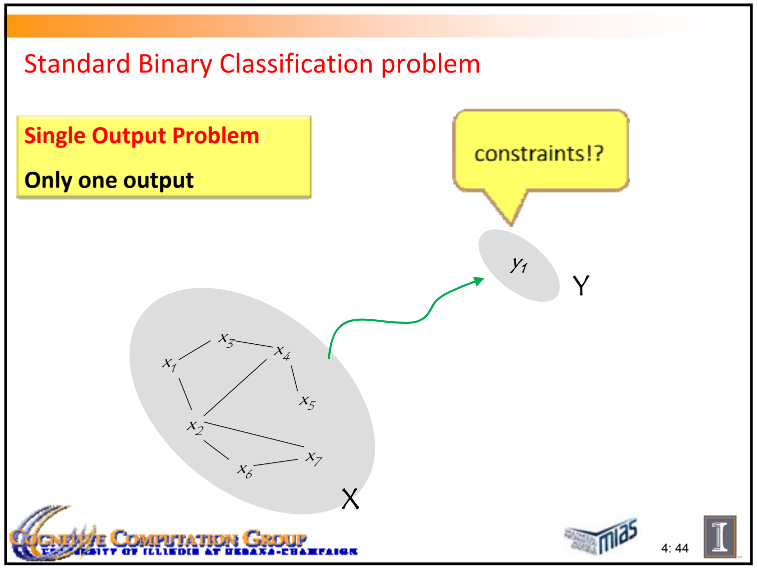


Structured output learning

Structure Output Problem

Dependencies between different outputs

Use constraints to capture the dependencies



Standard Binary Classification problem

Single Output Problem

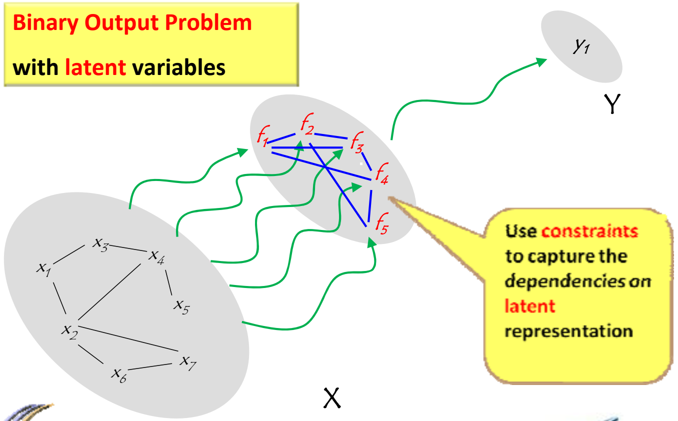
Only one output

constraints!?

Binary classification problem with latent representation

Binary Output Problem

with latent variables

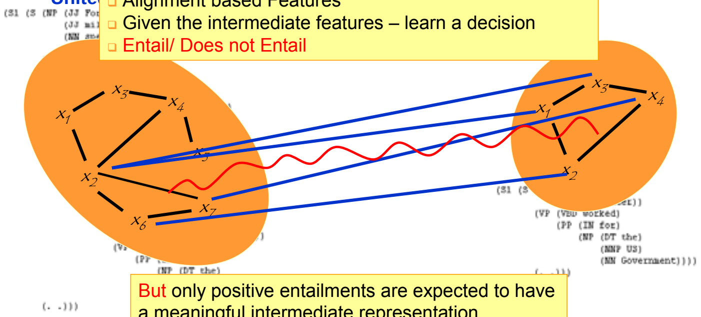


Use constraints to capture the dependencies on latent representation

Textual Entailment

Former military specialist Carpenter took the helm at FictitiousCom Inc. after ... in the Unite...

- Entailment Requires an Intermediate Representation
- Alignment based Features
- Given the intermediate features – learn a decision
- Entail/ Does not Entail



But only positive entailments are expected to have a meaningful intermediate representation

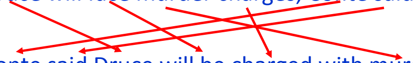
Jim Carpenter worked for the US Government.

Paraphrase Identification

Given an input x ∈ X
Learn a model f : X → {-1, 1}

- Consider the following sentences:

- S1: Druce will face murder charges, Conte said.



- S2: Conte said Druce will be charged with murder .

We need latent variables that explain: why this is a positive example.

- Are S1 and S2 a paraphrase of each other?
- There is a need for an intermediate representation to justify this decision

Given an input x ∈ X
Learn a model f : X → H → {-1, 1}

Algorithms: Two Conceptual Approaches

- Two stage approach (typically used for TE and paraphrase identification)
  - □ Learn hidden variables; fix it
    - **Need supervision for the hidden layer (or heuristics)**
  - □ For each example, extract features over x and (the fixed) h.
  - □ Learn a binary classier
- Proposed Approach: Joint Learning
  - □ Drive the learning of h from the binary labels
  - □ Find the best h(x)
  - □ **An intermediate structure representation is good to the extent is supports better final prediction.**
  - □ Algorithm?

Learning with Constrained Latent Representation (LCLR): Intuition

- If x is positive
  - □ There must exist a good explanation (intermediate representation)
  - □ $\exists$ h, $w^T \phi(x,h) \geq 0$
  - □ or, $\max_h w^T \phi(x,h) \geq 0$
- If x is negative
  - □ No explanation is good enough to support the answer
  - □ $\forall$ h, $w^T \phi(x,h) \leq 0$
  - □ or, $\max_h w^T \phi(x,h) \leq 0$
- Decision function: $\max_h w^T \phi(x,h)$:
  - □ See if the latent structure is good enough to support the labels!
  - □ An ILP formulation: CCM on the latent structure!

Learning with Constrained Latent Representation (LCLR): Framework

- LCLR provides a general inference formulation that allows that use of expressive constraints
  - □ Flexibly adapted for many tasks that require latent representations.

LCLR Model ⟵ Declarative model

- Paraphrasing: Model input as graphs, $V(G_{1,2})$, $E(G_{1,2})$
  - □ Four Hidden variables:
    - $h_{v1,v2}$ – possible vertex mappings; $h_{e1,e2}$ – possible edge mappings

$$\forall v_1 \in V(G_1), \sum_{v_2 \in V(G_2)} h_{v_1,v_2} + h_{v_1,*} = 1, \quad \forall v_2 \in V(G_2), \sum_{v_1 \in V(G_1)} h_{v_1,v_2} + h_{*,v_2} = 1$$

$$\forall e_1 \in E(G_1), \sum_{e_2 \in E(G_2)} h_{e_1,e_2} + h_{e_1,*} = 1, \quad \forall e_2 \in E(G_2), \sum_{e_1 \in E(G_1)} h_{e_1,e_2} + h_{*,e_2} = 1$$

$$h_{v_1,v_2} + h_{v_1',v_2'} - h_{e_1,e_2} \leq 1, \quad h_{v_1,v_2} \geq h_{e_1,e_2}, \quad h_{v_1',v_2'} \geq h_{e_1,e_2}$$

LCLR: The learning framework

- Altogether, this can be combined into an objective function:

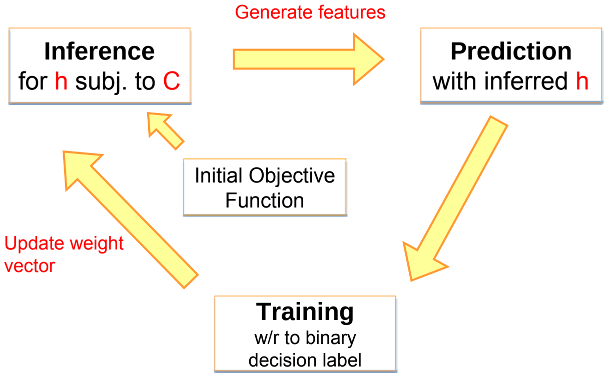
New feature vector for the final decision. Chosen h selects a representation.

$$\min_w \frac{1}{2}\|w\|^2 + \sum_{i=1}^{l} \ell(-y_i \max_{h \in C} w^T \sum_{s \in \Gamma(x)} h_s \Phi_s(x))$$

Inference: best h subject to constraints C

- Inference procedure inside the minimization procedure
- Why does inference help?
- Similar to what we mentioned with $S=\phi$
- **Focus: The binary classification task**

Iterative Objective Function Learning


Generate features
Inference for h subj. to C
Prediction with inferred h
Initial Objective Function
Update weight vector
Training w/r to binary decision label

- □ Formalized as Structured SVM + Constrained Hidden Structure
- □ **LCRL: Learning Constrained Latent Representation**

Optimization

- Non Convex, due to the maximization term inside the global minimization problem
- In each iteration:
  - □ Find the best feature representation h* for all positive examples (off-the shelf ILP solver)
  - □ Having fixed the representation for the positive examples, update w solving the convex optimization problem:

$$\min_{\mathbf{w}} \frac{1}{2}\|\mathbf{w}\|^2 + C \underline{\sum_{i:z_i=1} \ell(1 - \mathbf{w}^T \sum_s h^*_{i,s}\Phi_s(\mathbf{x}_i))} + C \underline{\sum_{i:z_i=-1} \ell(1 + \max_{\mathbf{h}\in\mathcal{H}} \mathbf{w}^T \sum_s h_s \Phi_s(\mathbf{x}_i))}$$

  - □ Not the standard SVM/LR: need inference
- Asymmetry: Only positive examples require a good intermediate representation that justifies the positive label.
  - □ Consequently, the objective function decreases monotonically

Experimental Results

Transliteration:

Transliteration System	Acc	MRR
(Goldwasser and Roth 2008)	N/A	89.4
Alignment + Learning	80.0	85.7
LCLR	**92.3**	**95.4**

Recognizing Textual Entailment:

Entailment System	Acc
Median of TAC 2009 systems	61.5
Alignment + Learning	65.0
LCLR	**66.8**

Paraphrase Identification:*

Alignment + Learning	72.00
LCLR	**72.75**

Summary

- Many important NLP problems require latent structures
- LCLR:
  - □ An algorithm that applies CCM on latent structures with ILP inference
  - □ Suitable for many different NLP tasks
  - □ Easy to inject linguistic constraints on latent structures
  - □ A general learning framework that is good for many loss functions
- Take home message:
  - □ It is possible to apply constraints on many important problems with latent variables!

Training Constrained Conditional Models

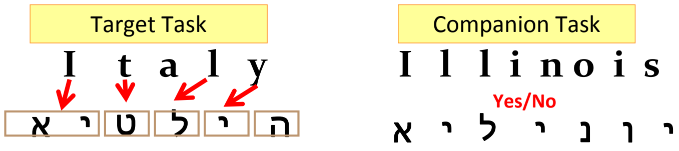
$$\operatorname*{argmax}_{y} \boldsymbol{\lambda} \cdot F(x,y) - \sum_{i=1}^{K} \rho_i d(y, 1_{C_i(x)})$$

- Learning model
  - □ Independently of the constraints (L+I)
  - □ Jointly, in the presence of the constraints (IBT)
  - □ Decomposed to simpler models
- Learning constraints' penalties
  - □ Independently of learning the model
  - □ Jointly, along with learning the model
- Dealing with lack of supervision
  - □ Constraints Driven Semi-Supervised learning (CODL)
  - ➡ □ Indirect Supervision
- Learning Constrained Latent Representations

Indirect Supervision for Structured Prediction

- Can we use other "weaker" supervision resources?
  - It is possible to use binary labeled data for structured output prediction tasks!
- Invent a companion binary decision problem!
  - **Parse Citations:** Lars Ole Andersen . Program analysis and specialization for the C Programming language. PhD thesis. DIKU , University of Copenhagen, May 1994 .
  - **Companion:** Given a citation; does it have a legitimate parse?
  - **POS Tagging**
  - **Companion:** Given a word sequence, does it have a legitimate POS tagging sequence?
- The binary supervision is easier to get. But is it helpful?

Predicting phonetic alignment (For Transliteration)



- Target Task
  - **Input:** an English Named Entity and its Hebrew Transliteration
  - **Output:** Phonetic Alignment (character sequence mapping)
  - A **structured** output prediction task (many constraints), hard to label
- Companion Task

Why it is a companion task?

  - **Input:** an English Named Entity and an Hebrew Named Entity
  - **Companion Output:** Do they form a transliteration pair?
  - A binary output problem, easy to label
  - Negative Examples are FREE, given positive examples

Companion Task Binary Label as Indirect Supervision

- The two tasks are related just like the binary and structured tasks discussed earlier

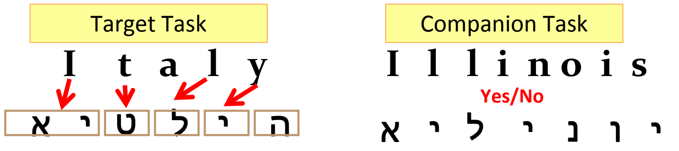
Positive transliteration pairs must have "good" phonetic alignments

Negative transliteration pairs cannot have "good" phonetic alignments

- All positive examples must have a good structure
- Negative examples cannot have a good structure
- We are in the same setting as before
  - Binary labeled examples are easier to obtain
  - We can take advantage of this to help learning a structured model
- Here: combine binary learning and structured learning

Joint Learning with Indirect Supervision (J-LIS)

- Joint learning : If available, make use of both supervision types



Loss function: L_B, as before; L_S, Structural learning
Key: the same parameter **w** for both components

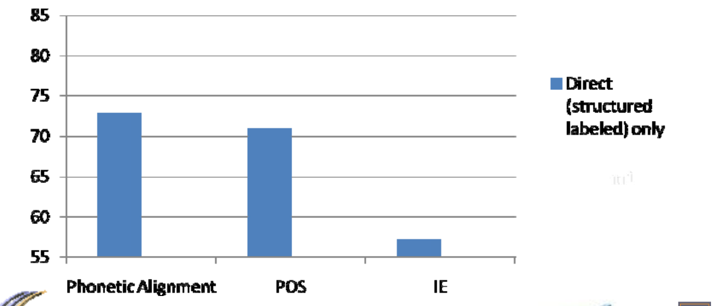
$$\min_w \frac{1}{2} w^T w + C_1 \sum_{i \in S} L_S(x_i, y_i; w)$$

Loss on Target Task

Loss on Companion Task

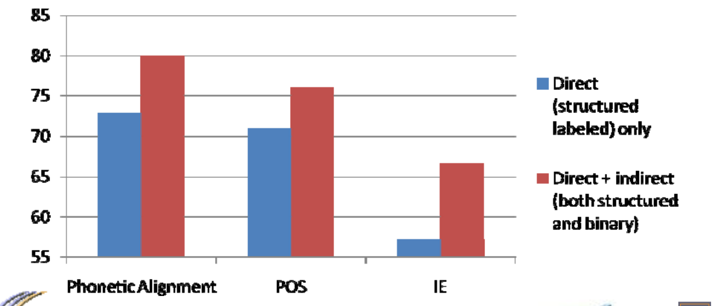
Experimental Result

- Very little direct (structured) supervision.
- (Almost free) Large amount binary indirect supervision



Experimental Result

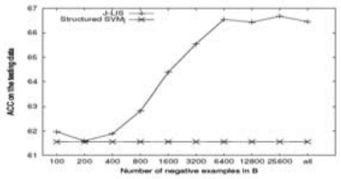
- Very little direct (structured) supervision.
- (Almost free) Large amount binary indirect supervision



Relations to Other Frameworks

- B=ϕ, l=(squared) hinge loss: Structural SVM
- S=ϕ, LCLR
  - Related to Structural Latent SVM (Yu & Johachims) and to Felzenszwalb.
- If S=ϕ, Conceptually related to Contrastive Estimation
  - No "grouping" of good examples and bad neighbors
  - Max vs. Sum: we do not marginalize over the hidden structure space
    - **Allows for complex domain specific constraints**
- Related to some

  Semi-Supervised approaches, but can use negative examples (Sheffer et. al)

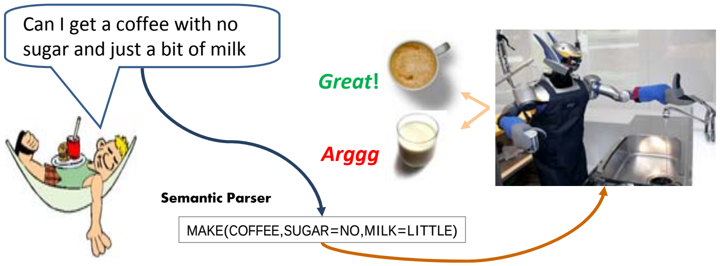


Dealing with lack of supervision

- Constraint Driven Learning
  - Use constraints to guide semi-supervised learning!
    [Chang, Ratinov, Roth, ACL'07;ICML'08,Long'10]
- Use Binary Labeled data to help structure output prediction
  - Training Structure Predictors by Inventing (easy to supervise) binary labels [ICML'10]

➡ □ Driving supervision signal from World's Response
  - **Efficient Semantic Parsing = ILP base inference + world's response**

Connecting Language to the World

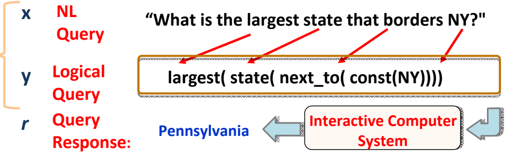

Can I get a coffee with no sugar and just a bit of milk
Great!
Arggg
Semantic Parser
MAKE(COFFEE,SUGAR=NO,MILK=LITTLE)

Can we rely on this interaction to provide supervision?

Real World Feedback

Traditional approach: learn from logical forms and gold alignments

EXPENSIVE!


x NL Query
"What is the largest state that borders NY?"
y Logical Query
largest(state(next_to(const(NY))))
r Query Response:
Pennsylvania
Interactive Computer System

Semantic parsing is a structured prediction problem: identify mappings from text to a meaning representation

The inference problem: a CCM formulation, with many constraints

Real World Feedback

Supervision = Expected Response

Our approach: use **only** the responses

x **NL Query** **"What is the largest state that borders NY?"**

r **Query Response:** **Pennsylvania**

Binary Supervision

Check if Predicted response == Expected response

Expected : Pennsylvania **Predicted** : Pennsylvania Positive Response	**Expected** : Pennsylvania **Predicted** : NYC Negative Response

Train a structured predictor with this binary supervision !

Empirical Evaluation

- Key Question: **Can we learn from this type of supervision?**

Algorithm	# training structures	Test set accuracy
No Learning: Initial Objective Fn Binary signal: **Protocol I**	0 0	22.2% 69.2 %
Binary signal: **Protocol II**	0	**73.2 %**
WM*2007 (fully supervised – uses gold structures)	**310**	**75 %**

*[WM] Y.-W. Wong and R. Mooney. 2007. Learning synchronous grammars for semantic parsing with lambda calculus. ACL.

Summary

- Constrained Conditional Models: Computational Framework for global inference and a vehicle for incorporating knowledge
- Direct supervision for structured NLP tasks is **expensive**
  - □ Indirect supervision is cheap and easy to obtain
- We suggested learning protocols for Indirect Supervision
  - □ Make use of simple, easy to get, binary supervision
  - □ Showed how to use it to learn structure
  - □ Done in the context of Constrained Conditional Models
    - **Inference is an essential part of propagating the simple supervision**
- Learning Structures from Real World Feedback
  - □ Obtain binary supervision from “real world” interaction
  - □ Indirect supervision replaces direct supervision

Summary: Training Constrained Conditional Models

$$\operatorname*{argmax}_y \boldsymbol{\lambda} \cdot F(x,y) - \sum_{i=1}^{K} \rho_i d(y, 1_{C_i(x)})$$

- Learning model
  - □ Independently of the constraints (L+I)
  - □ Jointly, in the presence of the constraints (IBT)
  - □ Decomposed to simpler models
- Learning constraints’ penalties
  - □ Independently of learning the model
  - □ Jointly, along with learning the model
- Dealing with lack of supervision
  - □ Constraints Driven Semi-Supervised learning (CODL)
  - □ Indirect Supervision
- Learning Constrained Latent Representations

This Tutorial: ILP & Constrained Conditional Models (Part II)

- Part 5: **Conclusion (& Discussion)** (10 min)
  - □ Building CCMs; Features and Constraints. Mixed models vs. Joint models;
  - □ where is Knowledge coming from

THE END

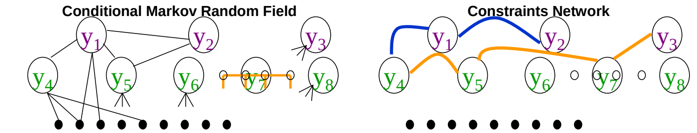
Conclusion

- Constrained Conditional Models combine
  - □ Learning conditional models with using declarative expressive constraints
  - □ Within a constrained optimization framework
- Our goal was to describe:
  - □ A clean way of incorporating constraints to bias and improve decisions of learned models
  - □ A clean way to use (declarative) prior knowledge to guide semi-supervised learning
  - □ Ways to make use of (declarative) prior knowledge when choosing intermediate (latent) representations.
- Provide examples for the diverse usage CCMs have already found in NLP
  - □ Significant success on several NLP and IE tasks (often, with ILP)

Technical Conclusions

- Presented and discussed modeling issues
  - □ How to improve existing models using declarative information
  - □ Incorporating expressive global constraints into simpler learned models
- Discussed Inference issues
  - □ Often, the formulation is via an Integer Linear Programming formulation, but algorithmic solutions can employ a variety of algorithms.
- Training issues – Training protocols matters
  - □ Training with/without constraints; soft/hard constraints;
  - □ Performance, modularity and ability to use previously learned models.
  - □ Supervision-lean models
- We did not attend to the question of "how to find constraints"
  - □ Emphasis on: background knowledge is important, exists, use it.
  - □ But, it's clearly possible to learn constraints.

Summary: Constrained Conditional Models



$y^* = \text{argmax}_y \sum w_i \phi(x; y)$

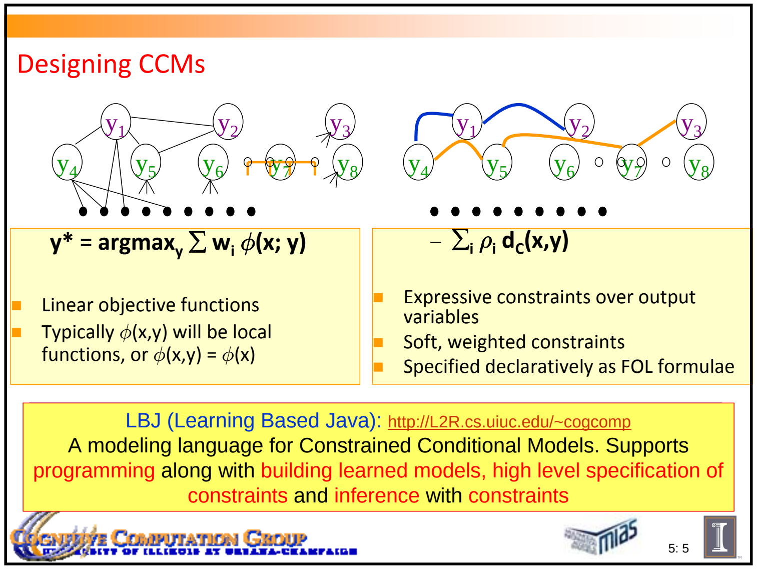
- Linear objective functions
- Typically $\phi(x,y)$ will be local functions, or $\phi(x,y) = \phi(x)$

$- \sum_i \rho_i d_C(x,y)$

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

- 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

Key difference from MLNs which provide a concise definition of a model, but the whole joint one.



Designing CCMs

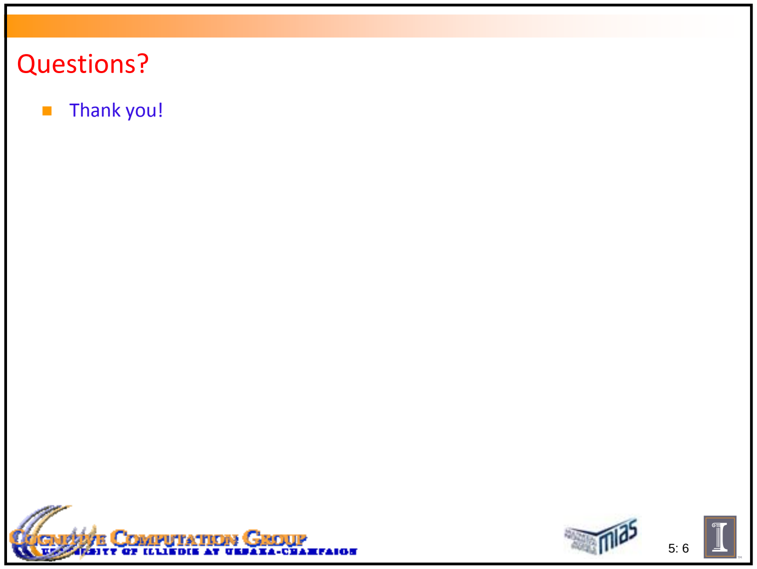
$$y^* = \mathrm{argmax}_y \sum w_i\, \phi(x; y)$$

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

$$- \sum_i \rho_i\, d_C(x,y)$$

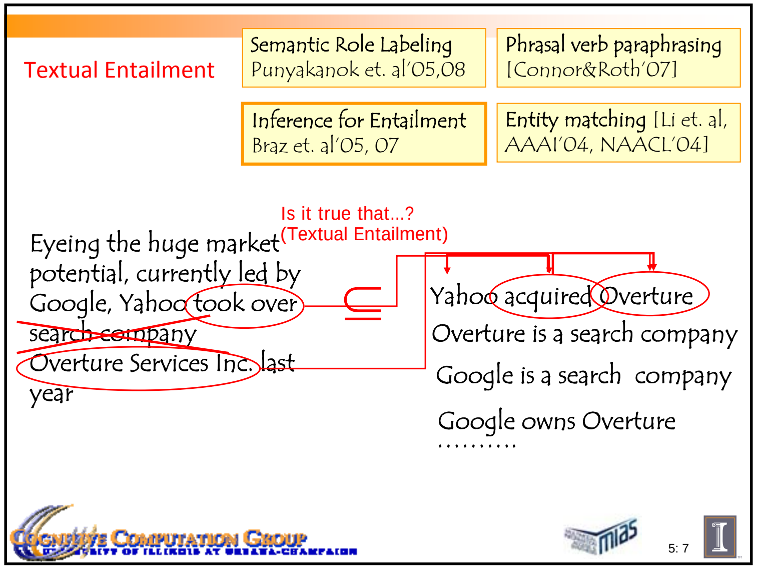
- Expressive constraints over output variables
- Soft, weighted constraints
- Specified declaratively as FOL formulae

LBJ (Learning Based Java): http://L2R.cs.uiuc.edu/~cogcomp
A modeling language for Constrained Conditional Models. Supports programming along with building learned models, high level specification of constraints and inference with constraints



Questions?

- Thank you!



Textual Entailment

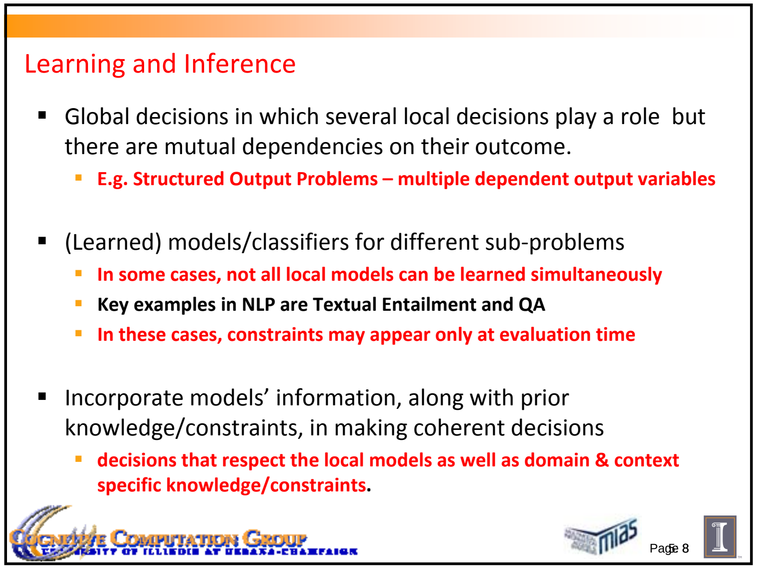
Semantic Role Labeling Punyakanok et. al'05,08	Phrasal verb paraphrasing [Connor&Roth'07]
Inference for Entailment Braz et. al'05, 07	Entity matching [Li et. al, AAAI'04, NAACL'04]

Is it true that...?
(Textual Entailment)

Eyeing the huge market potential, currently led by Google, Yahoo took over ~~search company~~ Overture Services Inc. last year

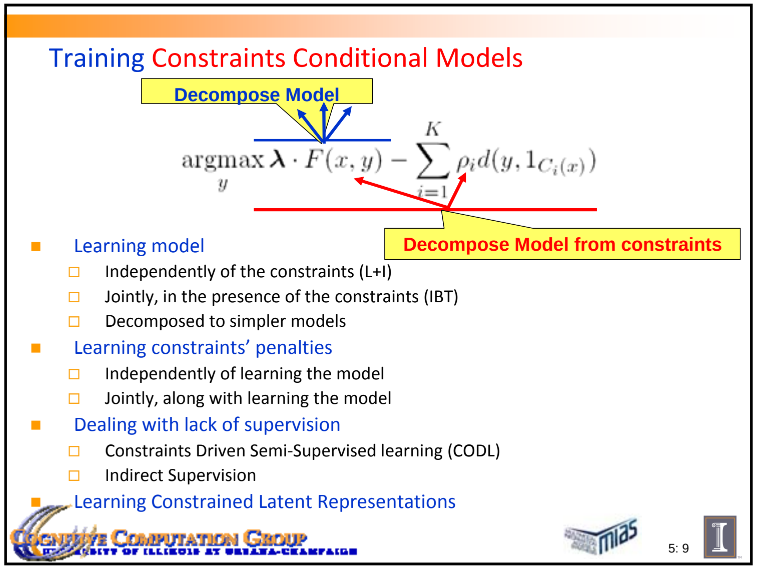
$\subseteq$

Yahoo acquired Overture
Overture is a search company
Google is a search company
Google owns Overture
……….



Learning and Inference

- 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 local models can 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 local models as well as domain & context specific knowledge/constraints.**



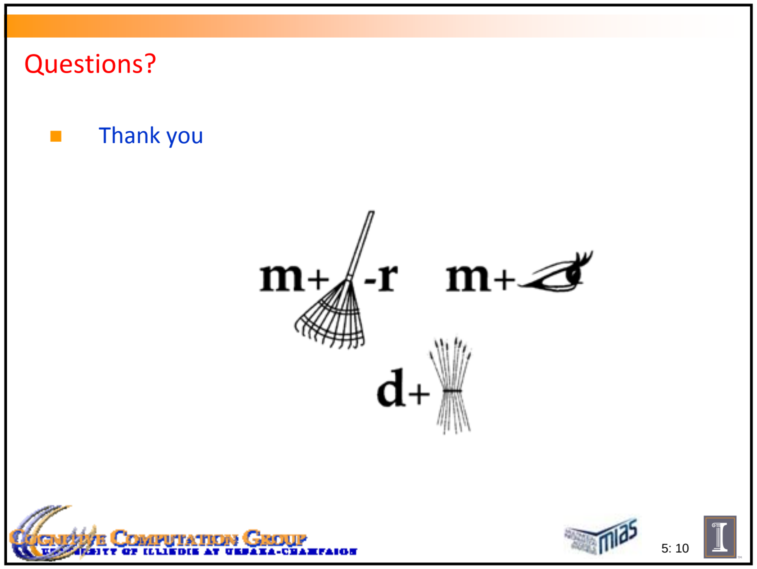
Training Constraints Conditional Models

Decompose Model

$$\underset{y}{\operatorname{argmax}}\, \boldsymbol{\lambda} \cdot F(x,y) - \sum_{i=1}^{K} \rho_i d(y, 1_{C_i(x)})$$

Decompose Model from constraints

- Learning model
  - Independently of the constraints (L+I)
  - Jointly, in the presence of the constraints (IBT)
  - Decomposed to simpler models
- Learning constraints' penalties
  - Independently of learning the model
  - Jointly, along with learning the model
- Dealing with lack of supervision
  - Constraints Driven Semi-Supervised learning (CODL)
  - Indirect Supervision
- Learning Constrained Latent Representations



Questions?

- Thank you

Bibliography on Constrained Conditional Models and Using Integer Linear Programming in NLP

June 1, 2010

References

Althaus, E., N. Karamanis, and A. Koller (2004, July). Computing locally coherent discourses. In *ACL*, Barcelona, Spain, pp. 399–406.

Barzilay, R. and M. Lapata (2006a). Aggregation via set partitioning for natural language generation. In *Proc. of the Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics (HLT-NAACL)*.

Barzilay, R. and M. Lapata (2006b, June). Aggregation via set partitioning for natural language generation. In *Proceedings of the Human Language Technology Conference of the NAACL, Main Conference*, New York City, USA, pp. 359–366. Association for Computational Linguistics.

Bellare, K., G. Druck, and A. McCallum (2009). Alternating projections for learning with expectation constraints. In *UAI*.

Bramsen, P., P. Deshpande, Y. K. Lee, and R. Barzilay (2006, July). Inducing temporal graphs. In *EMNLP*, Sydney, Australia, pp. 189–198. Association for Computational Linguistics.

Chambers, N. and D. Jurafsky (2008). Jointly combining implicit constraints improves temporal ordering. In *Proc. of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Chang, M., D. Goldwasser, D. Roth, and V. Srikumar (2010, Jun). Discriminative learning over constrained latent representations. In *NAACL*.

Chang, M., L. Ratinov, N. Rizzolo, and D. Roth (2008, July). Learning and inference with constraints. In *Proceedings of the National Conference on Artificial Intelligence (AAAI)*.

Chang, M., L. Ratinov, and D. Roth (2007, Jun). Guiding semi-supervision with constraint-driven learning. In *Proc. of the Annual Meeting of the ACL*, Prague, Czech Republic, pp. 280–287. Association for Computational Linguistics.

Chang, M., L. Ratinov, and D. Roth (2008, July). Constraints as prior knowledge. In *ICML Workshop on Prior Knowledge for Text and Language Processing*, pp. 32–39.

Chang, M., V. Srikumar, D. Goldwasser, and D. Roth (2010). Structured output learning with indirect supervision. In *ICML*.

Che, W., Z. Li, Y. Hu, Y. Li, B. Qin, T. Liu, and S. Li (2008, August). A cascaded syntactic and semantic dependency parsing system. In *CoNLL*, Manchester, England, pp. 238–242. Coling 2008 Organizing Committee.

Choi, Y., E. Breck, and C. Cardie (2006). Joint extraction of entities and relations for opinion recognition. In *Proc. of the 2006 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Clarke, J., D. Goldwasser, M. Chang, and D. Roth (2010, July). Driving semantic parsing from the world's response. In *Proceedings of the Fourteenth Conference on Computational Natural Language Learning (CoNLL-2010)*.

Clarke, J. and M. Lapata (2006). Constraint-based sentence compression: An integer programming approach. In *Proc. of the COLING/ACL 2006 Main Conference Poster Sessions (ACL)*.

Clarke, J. and M. Lapata (2007). Modelling compression with discourse constraints. In *Proc. of the Conference on Empirical Methods in Natural Language Processing and on Computational Natural Language Learning (EMNLP-CoNLL)*.

Clarke, J. and M. Lapata (2008). Global inference for sentence compression: An integer linear programming approach. *Journal of Artificial Intelligence Research (JAIR) 31*, 399–429.

Daumé III, H. (2008, October). Cross-task knowledge-constrained self training. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, Honolulu, Hawaii, pp. 680–688. Association for Computational Linguistics.

DeNero, J. and D. Klein (2008, June). The complexity of phrase alignment problems. In *Proceedings of ACL-08: HLT, Short Papers*, Columbus, Ohio, pp. 25–28. Association for Computational Linguistics.

Denis, P. and J. Baldridge (2007). Joint determination of anaphoricity and coreference resolution using integer programming. In *Proc. of the Annual Meeting of the North American Chapter of the Association for Computational Linguistics - Human Language Technology Conference (NAACL-HLT)*.

Deshpande, P., R. Barzilay, and D. Karger (2007, April). Randomized decoding for selection-and-ordering problems. In *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Proceedings of the Main Conference*, Rochester, New York, pp. 444–451. Association for Computational Linguistics.

Filippova, K. and M. Strube (2008a). Dependency tree based sentence compression. In *INLG*.

Filippova, K. and M. Strube (2008b, October). Sentence fusion via dependency graph compression. In *EMNLP*, Honolulu, Hawaii, pp. 177–185. Association for Computational Linguistics.

Finkel, J. R. and C. D. Manning (2008). The importance of syntactic parsing and inference in semantic rolelabeling. In *Proc. of the Annual Meeting of the Association for Computational Linguistics - Human Language Technology Conference, Short Papers (ACL-HLT)*.

Ganchev, K., J. Graça, J. Gillenwater, and B. Taskar (2010). Posterior regularization for structured latent variable models. *JMLR*.

Germann, U., M. Jahr, K. Knight, D. Marcu, and K. Yamada (2001, July). Fast decoding and optimal decoding for machine translation. In *Proceedings of 39th Annual Meeting of the Association for Computational Linguistics*, Toulouse, France, pp. 228–235. Association for Computational Linguistics.

Graca, J. V., K. Ganchev, and B. Taskar (2007). Expectation maximization and posterior constraints. In *NIPS*, Volume 20.

Klenner, M. (2006). Grammatical role labeling with integer linear programming. In *EACL*.

Klenner, M. (2007a). Enforcing consistency on coreference sets. In *RANLP*.

Klenner, M. (2007b, June). Shallow dependency labeling. In *ACL*, Prague, Czech Republic, pp. 201–204. Association for Computational Linguistics.

Koomen, P., V. Punyakanok, D. Roth, and W. Yih (2005). Generalized inference with multiple semantic role labeling systems (shared task paper). In I. Dagan and D. Gildea (Eds.), *Proc. of the Annual Conference on Computational Natural Language Learning (CoNLL)*, pp. 181–184.

Mann, G. and A. McCallum (2008). Generalized expectation criteria for semi-supervised learning of conditional random fields. In *ACL*, Number 870 - 878.

Martins, A., N. A. Smith, and E. Xing (2009a, August). Concise integer linear programming formulations for dependency parsing. In *ACL*.

Martins, A. F. T., N. A. Smith, and E. P. Xing (2009b). Polyhedral outer approximations with application to natural language parsing. In *ICML*, New York, NY, USA, pp. 713–720. ACM.

McDonald, R. (2007). A study of global inference algorithms in multi-document summarization. In *ECIR*.

Punyakanok, V., D. Roth, W. Yih, and D. Zimak (2004, August). Semantic role labeling via integer linear programming inference. In *Proc. the International Conference on Computational Linguistics (COLING)*, Geneva, Switzerland, pp. 1346–1352.

Punyakanok, V., D. Roth, W. Yih, and D. Zimak (2005). Learning and inference over constrained output. In *Proc. of the International Joint Conference on Artificial Intelligence (IJCAI)*.

Punyakanok, V., D. Roth, W. Yih, D. Zimak, and Y. Tu (2004). Semantic role labeling via generalized inference over classifiers (shared task paper). In H. T. Ng and E. Riloff (Eds.), *Proc. of the Annual Conference on Computational Natural Language Learning (CoNLL)*, pp. 130–133.

Riedel, S. and J. Clarke (2006, July). Incremental integer linear programming for non-projective dependency parsing. In *EMNLP*, Sydney, Australia, pp. 129–137. Association for Computational Linguistics.

Rizzolo, N. and D. Roth (2007, September). Modeling Discriminative Global Inference. In *Proc. of the First International Conference on Semantic Computing (ICSC)*, Irvine, California, pp. 597–604. IEEE.

Rizzolo, N. and D. Roth (2010, May). Learning Based Java for Rapid Development of NLP Systems. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC)*, Valletta, Malta.

Roth, D. (2005). Learning based programming.

Roth, D. and W. Yih (2005). Integer linear programming inference for conditional random fields. In *Proc. of the International Conference on Machine Learning (ICML)*, pp. 737–744.

Roth, D. and W. Yih (2007). Global inference for entity and relation identification via a linear programming formulation. In L. Getoor and B. Taskar (Eds.), *Introduction to Statistical Relational Learning*. MIT Press.

Sagae, K., Y. Miyao, and J. Tsujii (2007, June). Hpsg parsing with shallow dependency constraints. In *ACL*, Prague, Czech Republic, pp. 624–631. Association for Computational Linguistics.

Tsai, T., C. Wu, Y. Lin, and W. Hsu (2005, June). Exploiting full parsing information to label semantic roles using an ensemble of ME and SVM via integer linear programming. In *CoNLL*, Ann Arbor, Michigan, pp. 233–236. Association for Computational Linguistics.