Recent Advances and Applications of Markov Logic Networks

Deepak Venugopal*, Vibhav Gogate** and Vincent Ng**

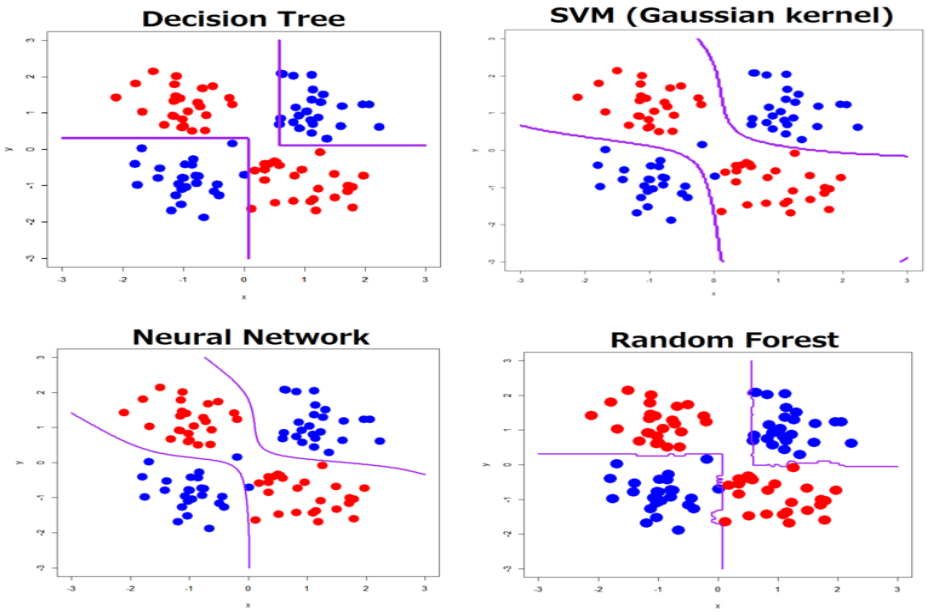
* The University of Memphis
** The University of Texas at Dallas

Outline

- Introduction
  - Motivation
  - MLN Basics
- Scalable Inference
  - Domain-Lifted Inference
  - Approximate Domain-Lifting
- Scalable Weight Learning
  - Lifted Generative Weight Learning
  - Discriminative/pseudo-likelihood learning with Approximate Counting Oracles
- Applications
  - Overview of MLN Software
  - NLP Applications using MLNs
- Conclusion

Traditional Machine Learning

Features
Instance_1
Instance_2
Instance_3
….

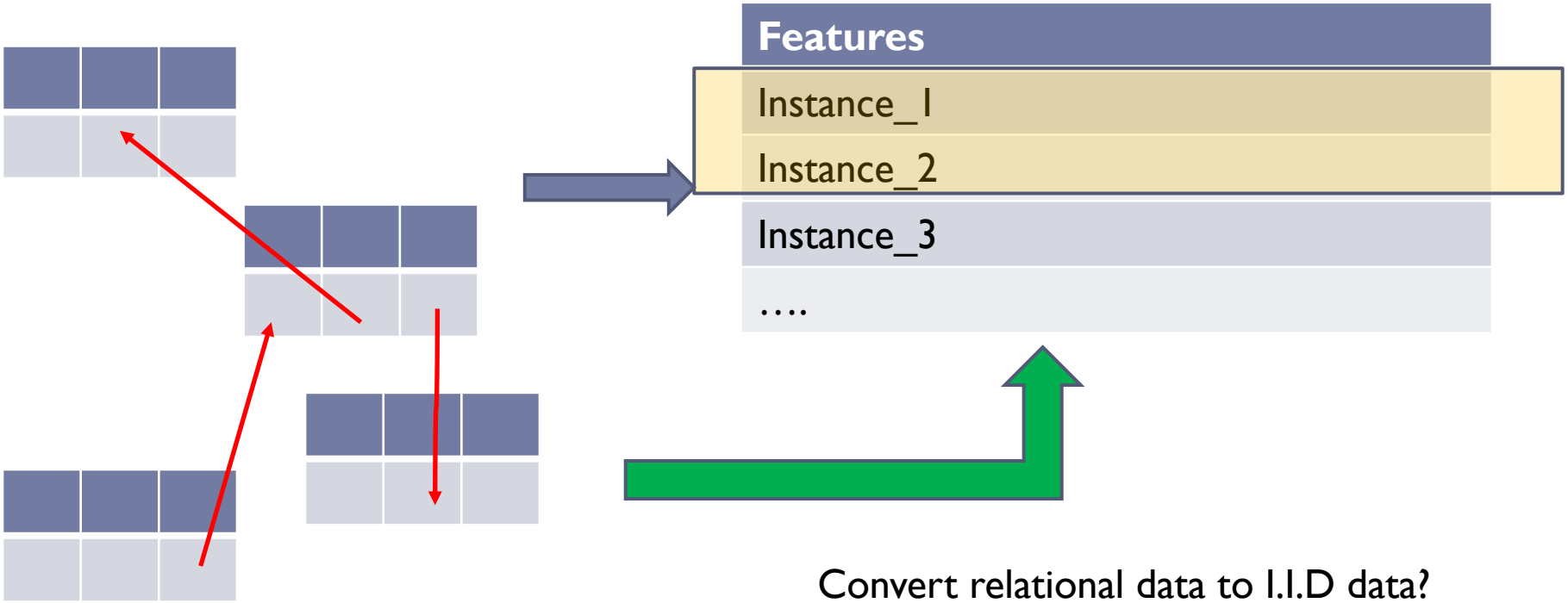


Features
Instance_1
Instance_2
Instance_3
….

Convert relational data to I.I.D data?

- Real-world data is often relational
  - Healthcare Records
  - Student Records
  - Social Networks
  - ….

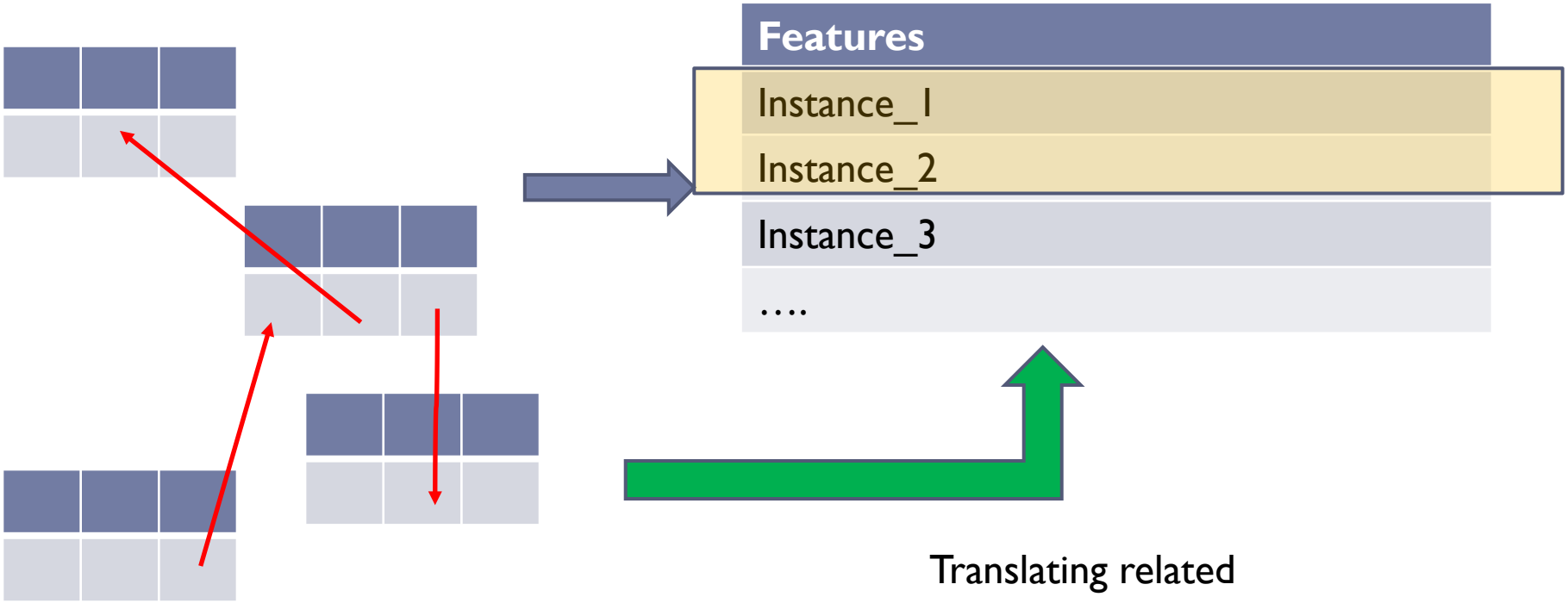
Traditional Machine Learning



Convert relational data to I.I.D data?

- Real-world data is often relational
  - Health Records
  - Student Records
  - Social Network
  - ….

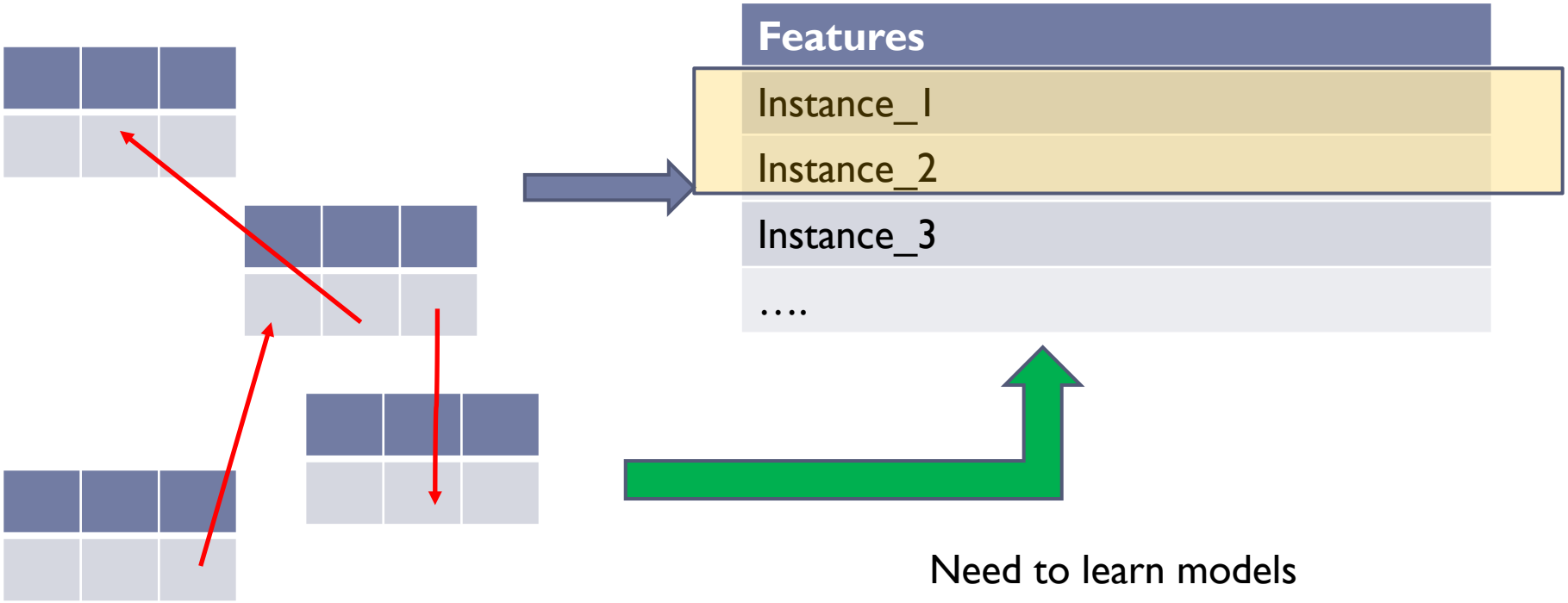
Traditional Machine Learning



- Real-world data is often relational
  - Health Records
  - Student Records
  - Social Network
  - ….

Translating related instances to i.i.d instances loses relational information

Traditional Machine Learning

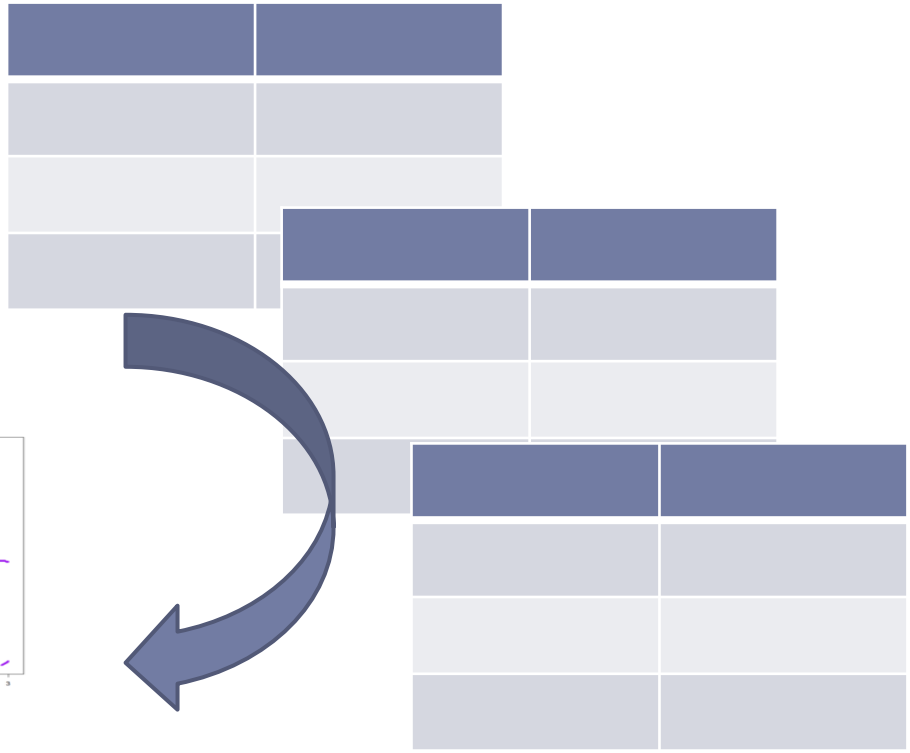


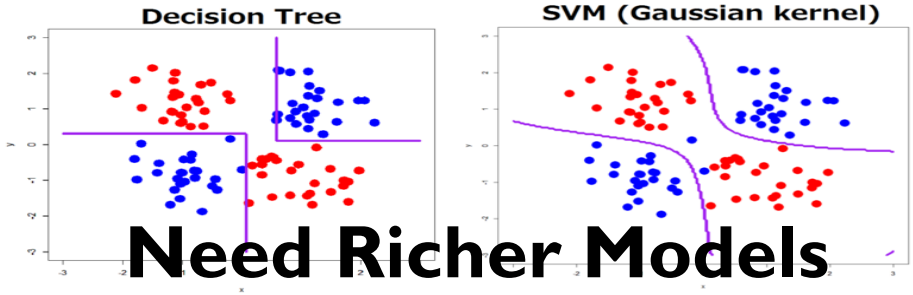
Need to learn models that can handle uncertainty

- Real-world data has inherent ambiguity
  - Smoking causes cancer
  - Similar skilled students perform poorly on the same problems
  - Friends have similar habits
  - ….

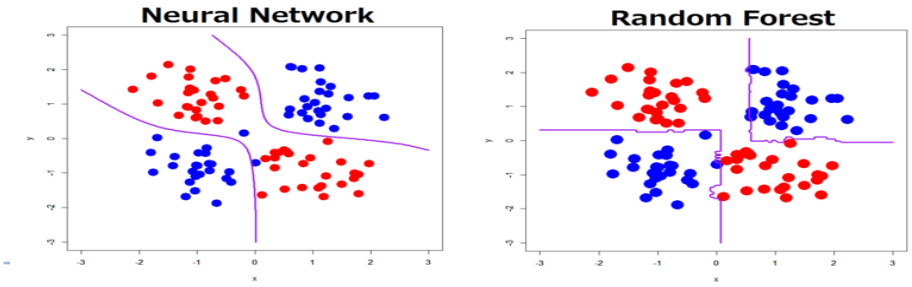
“New” Machine Learning

- Data is Relational
- Data has Uncertainty
- Data is Large





Need Richer Models



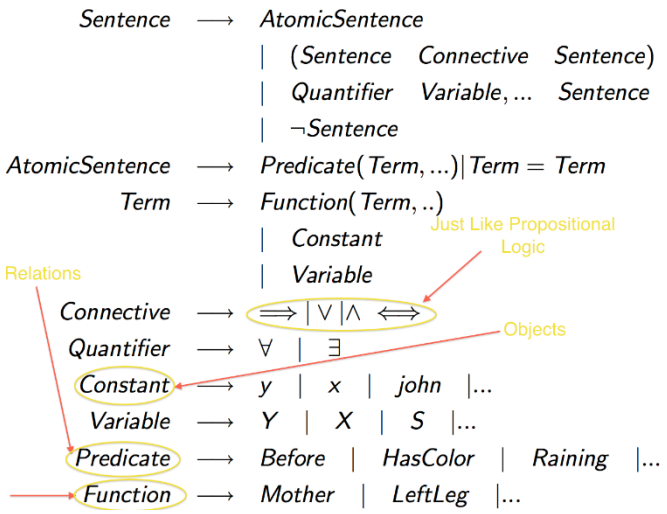
Statistical Relational Models

- Several popular models
  - Markov Logic (Pedro Domingos' group at UW)
  - BLOG (Stuart Russell's group at Berkeley)
  - PSL (Lise Getoor's group at UMD/UCSC)
  - ProbLog (Luc De Raedt's group at KU-Leuven)

Statistical Relational Models

- Several popular models
  - Markov Logic (Pedro Domingos' group at UW)
  - BLOG (Stuart Russell's group at Berkeley)
  - PSL (Lise Getoor's group at UMD/UCSC)
  - ProbLog (Luc De Raedt's group at KU-Leuven)
- MLNs are arguably one of the most popular Statistical Relational Models
  - Simple interpretation
  - Powerful representation
- This tutorial is mainly focused on MLNs though similar ideas are potentially useful in other Statistical Relational Models as well

What representation can we learn from noisy, relational big data?

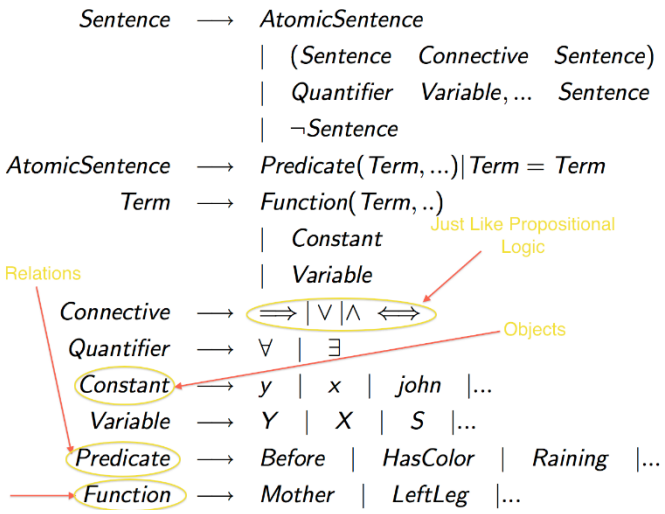


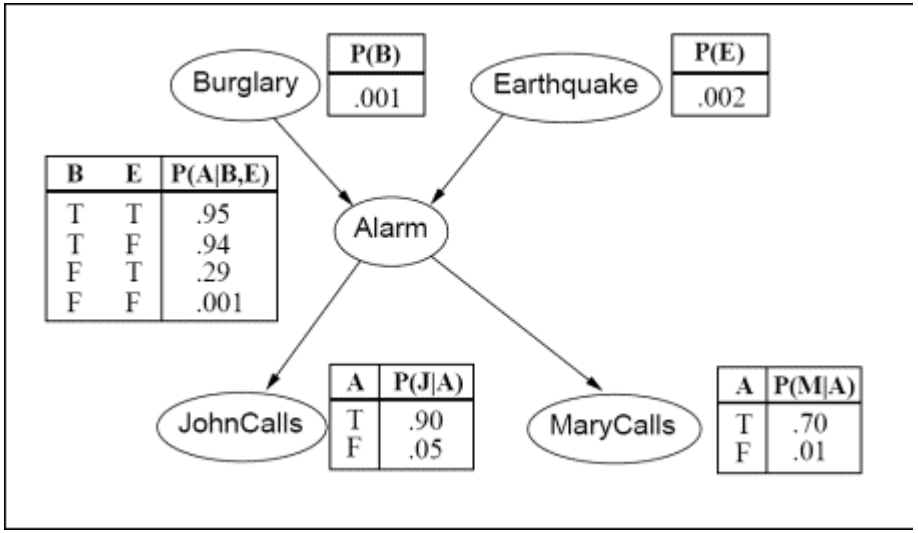
First-Order Logic represents complex relational knowledge extremely compactly

E.g. Rules of chess can be written in one page in FOL while it takes 10^{38} pages of finite automata to represent it*!

[*Russell CACM '15]

What representation can we learn from noisy, relational big data?





First-Order Logic represents complex relational knowledge extremely compactly

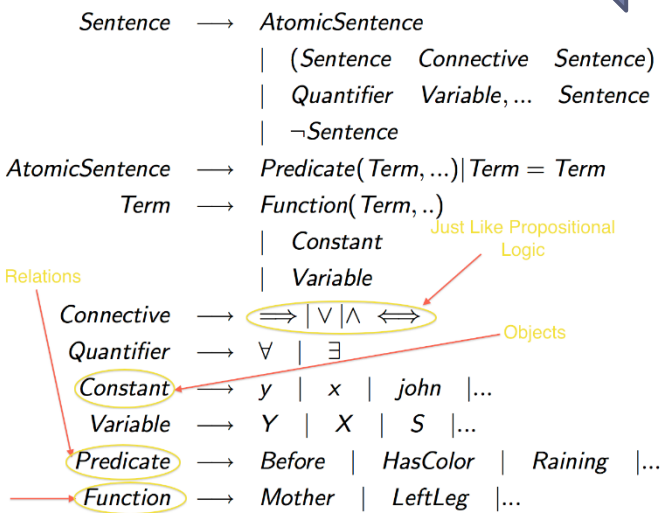
E.g. Rules of chess can be written in one page in FOL while it takes 10^{38} pages of finite automata to represent it*!

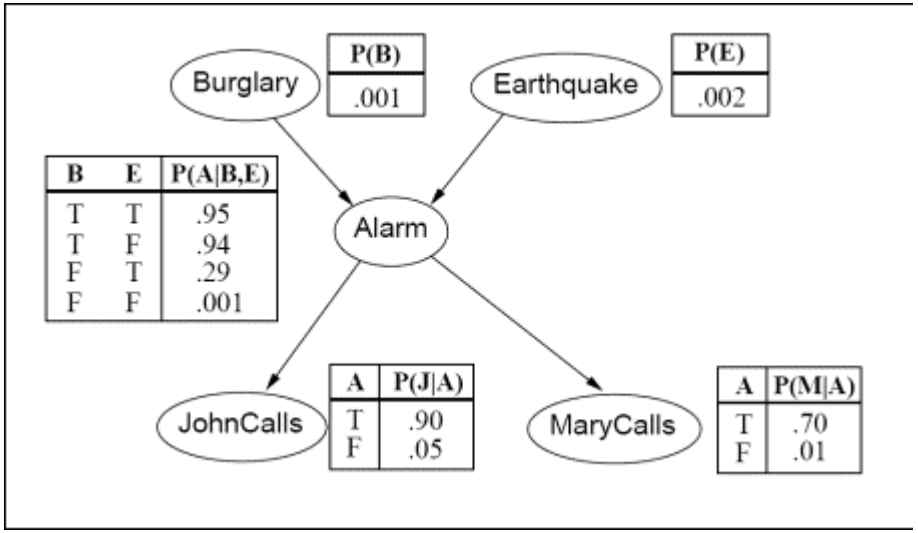
Probabilistic Graphical Models (Markov Networks, Bayesian Networks) can represent large joint distributions compactly

[*Russell CACM '15]

What representation can we learn from noisy, relational big data?

Markov Logic Networks (MLNs)





First-Order Logic represents complex relational knowledge extremely compactly

E.g. Rules of chess can be written in one page in FOL while it takes 10^{38} pages of finite automata to represent it*!

Probabilistic Graphical Models (Markov Networks, Bayesian Networks) can represent large joint distributions compactly

[*Russell CACM '15]

MLN Representation

- Weighted first-order formulas
  - $\forall x\, \forall y\, \boldsymbol{Smokes}(x) \wedge \boldsymbol{Friends}(x, y) \Rightarrow \neg \boldsymbol{Asthma}(y)$; **1.75**

MLN Representation

- Weighted first-order formulas
  - $\forall x\, \forall y\, \boldsymbol{Smokes}(x) \wedge \boldsymbol{Friends}(x, y) \Rightarrow \neg \boldsymbol{Asthma}(y)$; **1.75**
- Larger the weight, more likely is the formula to be true
- For weight ∞, the formula specifies a hard constraint just like first-order logic

MLN Representation

- Weighted first-order formulas
  - $\forall x\, \forall y\, \boldsymbol{Smokes}(x) \wedge \boldsymbol{Friends}(x, y) \Rightarrow \neg \boldsymbol{Asthma}(y)$; **1.75**
- Larger the weight, more likely is the formula to be true
- For weight ∞, the formula specifies a hard constraint just like first-order logic
- MLNs are a template for generating large Markov networks (undirected Probabilistic Graphical Model)

Markov Logic Networks (MLNs)

$$\forall x \, \forall y \; \boldsymbol{Smokes}(x) \wedge \boldsymbol{Friends}(x, y) \Rightarrow \neg \boldsymbol{Asthma}(y) \qquad \mathbf{0.75}$$

Markov Logic Networks (MLNs)

$$\forall x\, \forall y\, \boldsymbol{Smokes}(x) \wedge \boldsymbol{Friends}(x, y) \Rightarrow \neg \boldsymbol{Asthma}(y) \quad \mathbf{0.75}$$

$$\boldsymbol{Smokes}(\boldsymbol{Ana}) \wedge \boldsymbol{Friends}(\boldsymbol{Ana}, \boldsymbol{Bob}) \Rightarrow \neg \boldsymbol{Asthma}(\boldsymbol{Bob}) \; \mathbf{0.75}$$

$$\boldsymbol{Smokes}(\boldsymbol{Bob}) \wedge \boldsymbol{Friends}(\boldsymbol{Bob}, \boldsymbol{Ana}) \Rightarrow \neg \boldsymbol{Asthma}(\boldsymbol{Ana}) \; \mathbf{0.75}$$

$$\boldsymbol{Smokes}(\boldsymbol{Bob}) \wedge \boldsymbol{Friends}(\boldsymbol{Bob}, \boldsymbol{Bob}) \Rightarrow \neg \boldsymbol{Asthma}(\boldsymbol{Bob}) \; \mathbf{0.75}$$

$$\boldsymbol{Smokes}(\boldsymbol{Ana}) \wedge \boldsymbol{Friends}(\boldsymbol{Ana}, \boldsymbol{Ana}) \Rightarrow \neg \boldsymbol{Asthma}(\boldsymbol{Ana}) \; \mathbf{0.75}$$

Markov Logic Networks (MLNs)

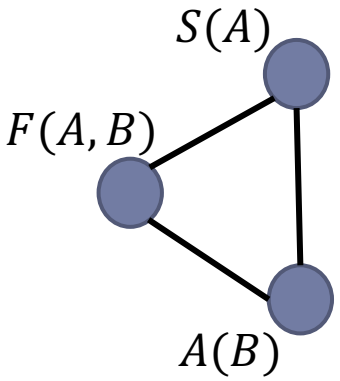
$$\forall x \, \forall y \; \boldsymbol{Smokes}(x) \wedge \boldsymbol{Friends}(x, y) \Rightarrow \neg \boldsymbol{Asthma}(y) \quad \mathbf{0.75}$$

$$\boldsymbol{Smokes}(\boldsymbol{Ana}) \wedge \boldsymbol{Friends}(\boldsymbol{Ana}, \boldsymbol{Bob}) \Rightarrow \neg \boldsymbol{Asthma}(\boldsymbol{Bob}) \; \mathbf{0.75}$$

$$\boldsymbol{Smokes}(\boldsymbol{Bob}) \wedge \boldsymbol{Friends}(\boldsymbol{Bob}, \boldsymbol{Ana}) \Rightarrow \neg \boldsymbol{Asthma}(\boldsymbol{Ana}) \; \mathbf{0.75}$$

$$\boldsymbol{Smokes}(\boldsymbol{Bob}) \wedge \boldsymbol{Friends}(\boldsymbol{Bob}, \boldsymbol{Bob}) \Rightarrow \neg \boldsymbol{Asthma}(\boldsymbol{Bob}) \; \mathbf{0.75}$$

$$\boldsymbol{Smokes}(\boldsymbol{Ana}) \wedge \boldsymbol{Friends}(\boldsymbol{Ana}, \boldsymbol{Ana}) \Rightarrow \neg \boldsymbol{Asthma}(\boldsymbol{Ana}) \; \mathbf{0.75}$$

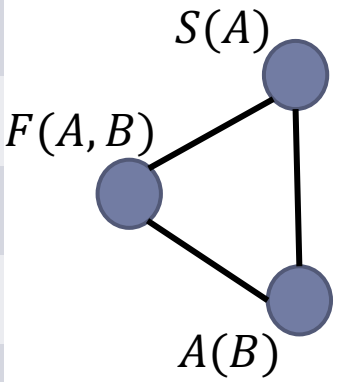


Markov Logic Networks (MLNs)

$$\forall x\, \forall y\, \boldsymbol{Smokes}(x) \wedge \boldsymbol{Friends}(x, y) \Rightarrow \neg \boldsymbol{Asthma}(y) \quad \mathbf{0.75}$$

$$\boldsymbol{Smokes}(\boldsymbol{Ana}) \wedge \boldsymbol{Friends}(\boldsymbol{Ana}, \boldsymbol{Bob}) \Rightarrow \neg \boldsymbol{Asthma}(\boldsymbol{Bob}) \; \mathbf{0.75}$$
$$\boldsymbol{Smokes}(\boldsymbol{Bob}) \wedge \boldsymbol{Friends}(\boldsymbol{Bob}, \boldsymbol{Ana}) \Rightarrow \neg \boldsymbol{Asthma}(\boldsymbol{Ana}) \; \mathbf{0.75}$$
$$\boldsymbol{Smokes}(\boldsymbol{Bob}) \wedge \boldsymbol{Friends}(\boldsymbol{Bob}, \boldsymbol{Bob}) \Rightarrow \neg \boldsymbol{Asthma}(\boldsymbol{Bob}) \; \mathbf{0.75}$$
$$\boldsymbol{Smokes}(\boldsymbol{Ana}) \wedge \boldsymbol{Friends}(\boldsymbol{Ana}, \boldsymbol{Ana}) \Rightarrow \neg \boldsymbol{Asthma}(\boldsymbol{Ana}) \; \mathbf{0.75}$$

Smokes (Ana)	Friends (Ana,Bob)	Asthma (Bob)	ϕ
0	0	0	$e^{0.75}$
0	0	I	$e^{0.75}$
0	I	0	$e^{0.75}$
…	…	…	…
I	I	0	e^{0}
I	I	I	$e^{0.75}$

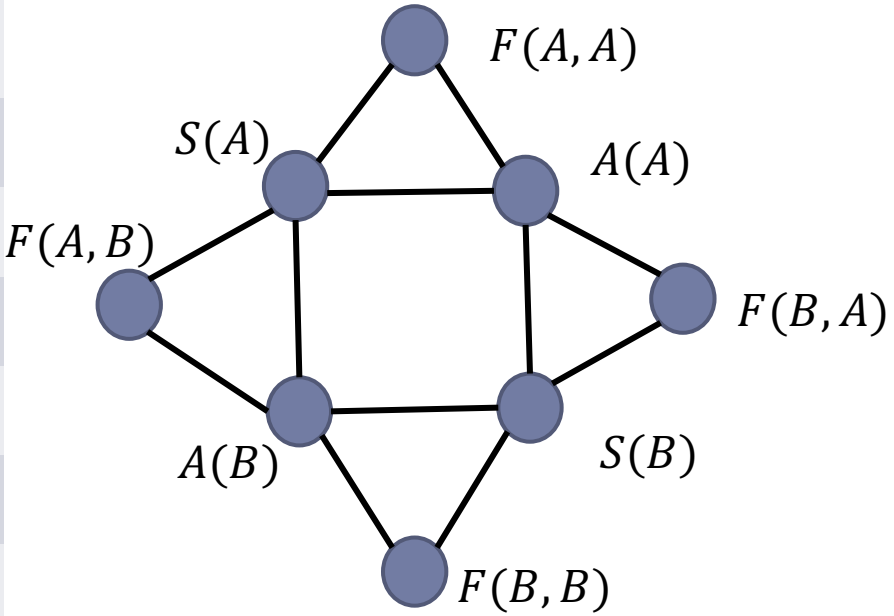


Markov Logic Networks (MLNs)

$$\forall x \, \forall y \; \boldsymbol{Smokes}(x) \wedge \boldsymbol{Friends}(x, y) \Rightarrow \neg \boldsymbol{Asthma}(y) \quad \mathbf{0.75}$$

$$\boldsymbol{Smokes}(\boldsymbol{Ana}) \wedge \boldsymbol{Friends}(\boldsymbol{Ana}, \boldsymbol{Bob}) \Rightarrow \neg \boldsymbol{Asthma}(\boldsymbol{Bob}) \; \mathbf{0.75}$$
$$\boldsymbol{Smokes}(\boldsymbol{Bob}) \wedge \boldsymbol{Friends}(\boldsymbol{Bob}, \boldsymbol{Ana}) \Rightarrow \neg \boldsymbol{Asthma}(\boldsymbol{Ana}) \; \mathbf{0.75}$$
$$\boldsymbol{Smokes}(\boldsymbol{Bob}) \wedge \boldsymbol{Friends}(\boldsymbol{Bob}, \boldsymbol{Bob}) \Rightarrow \neg \boldsymbol{Asthma}(\boldsymbol{Bob}) \; \mathbf{0.75}$$
$$\boldsymbol{Smokes}(\boldsymbol{Ana}) \wedge \boldsymbol{Friends}(\boldsymbol{Ana}, \boldsymbol{Ana}) \Rightarrow \neg \boldsymbol{Asthma}(\boldsymbol{Ana}) \; \mathbf{0.75}$$

Smokes (Ana)	Friends (Ana,Bob)	Asthma (Bob)	ϕ
0	0	0	$e^{0.75}$
0	0	1	$e^{0.75}$
0	1	0	$e^{0.75}$
…	…	…	…
1	1	0	e^{0}
1	1	1	$e^{0.75}$



Background

	S(A)	S(B)	M(A)	M(B)	F(A,B)	F(B,A)	F(A,A)	F(B,A)
ω_1	0	0	0	0	0	0	0	0
ω_2	0	0	0	0	0	0	0	I
ω_3	0	0	0	0	0	0	I	0
…	…	…	…	…	…	…	…	

Joint distribution over possible worlds

Background

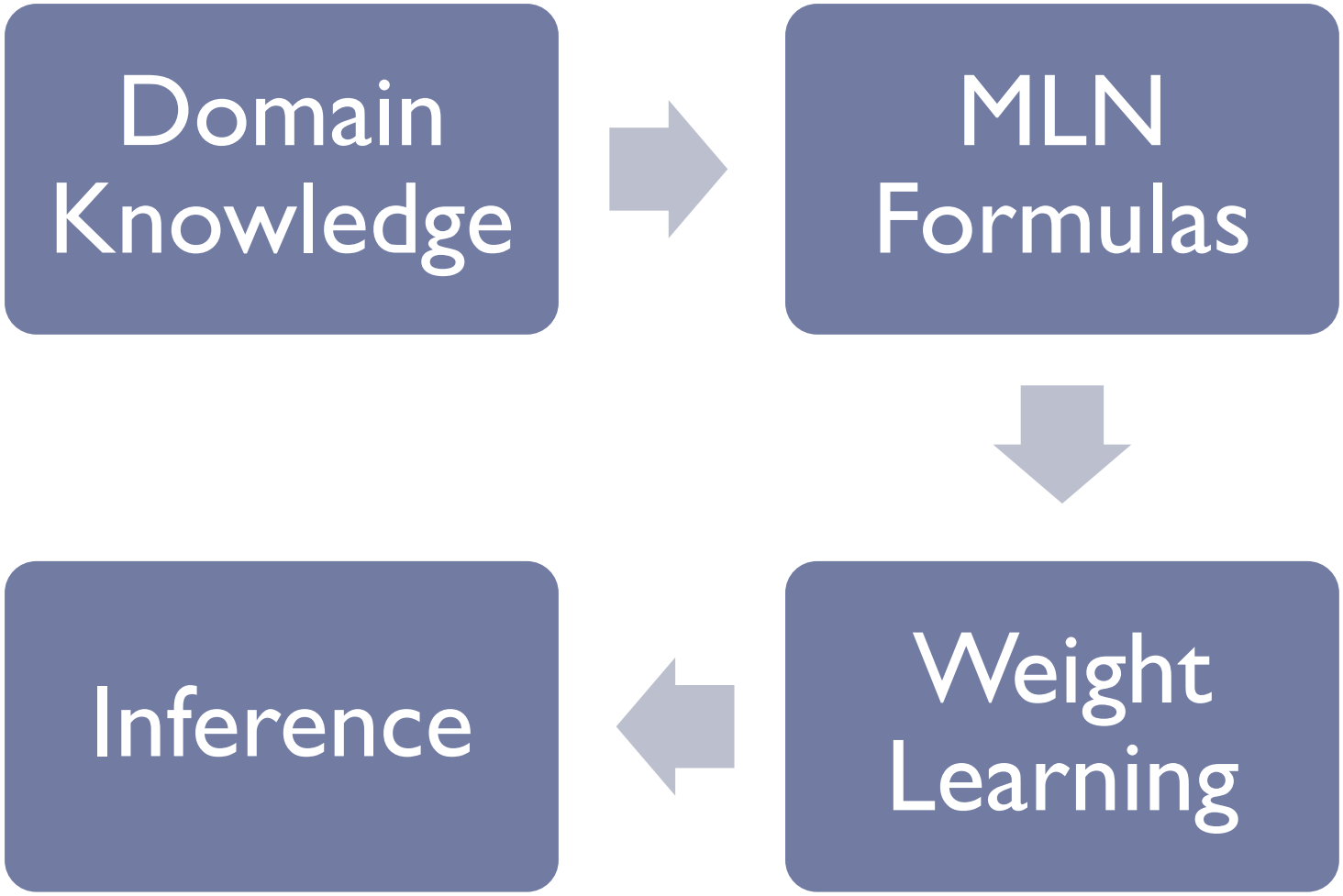
	S(A)	S(B)	M(A)	M(B)	F(A,B)	F(B,A)	F(A,A)	F(B,A)
ω_1	0	0	0	0	0	0	0	0
ω_2	0	0	0	0	0	0	0	I
ω_3	0	0	0	0	0	0	I	0
…	…	…	…	…	…	…	…	…

Joint distribution over possible worlds

$$P(\omega) = \frac{1}{Z} exp\left(\sum_f w_f \#SATGround(f, \omega)\right)$$

$$Z = \sum_\omega exp\left(\sum_f w_f \#SATGround(f, \omega)\right)$$

Typical MLN Application Design Process



MLNs

- Typical Inference Tasks: Marginal and MAP Inference
  - $P(Smokes(Ana)|Friends(Ana, Bob), Smokes(B) \ldots) =?$
  - $\max_x P(x|y)$, where y is the evidence and x is the set of remaining variables in the MLN.

MLNs

- Typical Inference Tasks: Marginal and MAP Inference
  - $P(Smokes(Ana)|Friends(Ana, Bob), Smokes(B) \ldots) = ?$
  - $\max_x P(x|y)$, where y is the evidence and x is the set of remaining variables in the MLN.
- Weight learning
  - Unlike traditional learning algorithms, in typical MLN learning there is just one instance to learn from (the relational DB)
  - Weight learning uses inference during each step

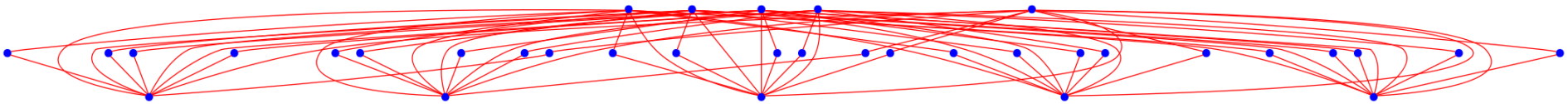
What are the challenges with MLNs?

- Scalability is the primary challenge in utilizing MLNs for large application domains such as NLP
  - Inference and Learning are hard problems in MLNs
  - Need tools for MLNs that can work as a "black-box" for application designers

What are the challenges with MLNs?

- Scalability is the primary challenge in utilizing MLNs for large application domains such as NLP
  - Inference and Learning are hard problems in MLNs
  - Need tools for MLNs that can work as a "black-box" for application designers
- Can't we just convert MLNs into a Markov Network and apply techniques there?
  - Unfortunately, MLNs typically encode Markov Networks that are orders of magnitude larger than what (propositional) PGM inference algorithms can handle

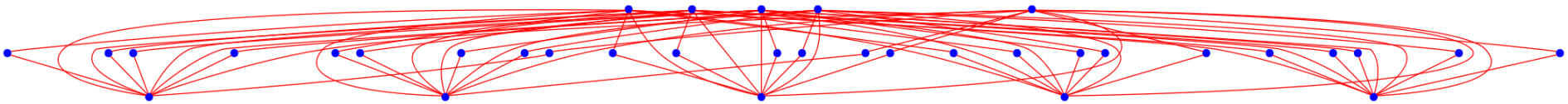
$$\boldsymbol{Smokes}(x) \wedge \boldsymbol{Friends}(x, y) \Rightarrow \neg \boldsymbol{Asthma}(y)$$



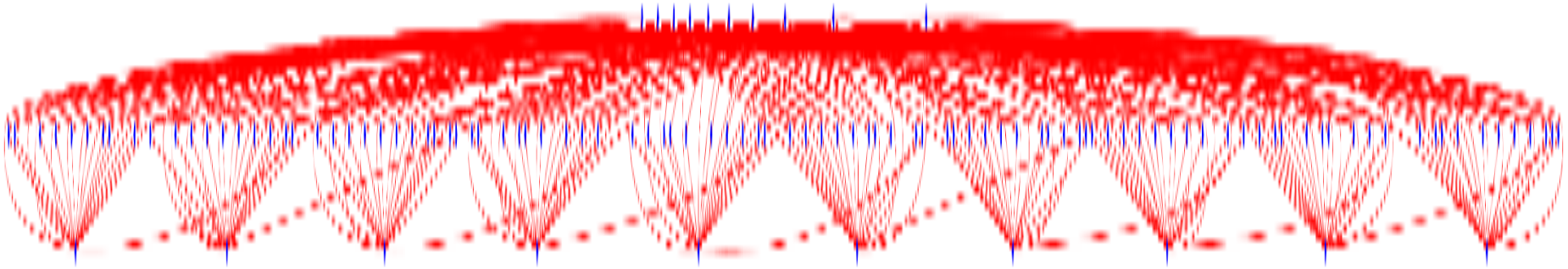
$\#Objects = 5$

$\#Objects = 50$

$$\boldsymbol{Smokes}(x) \wedge \boldsymbol{Friends}(x, y) \Rightarrow \neg \boldsymbol{Asthma}(y)$$

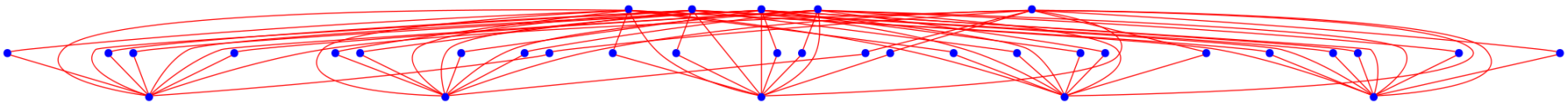


$\#Objects = 5$

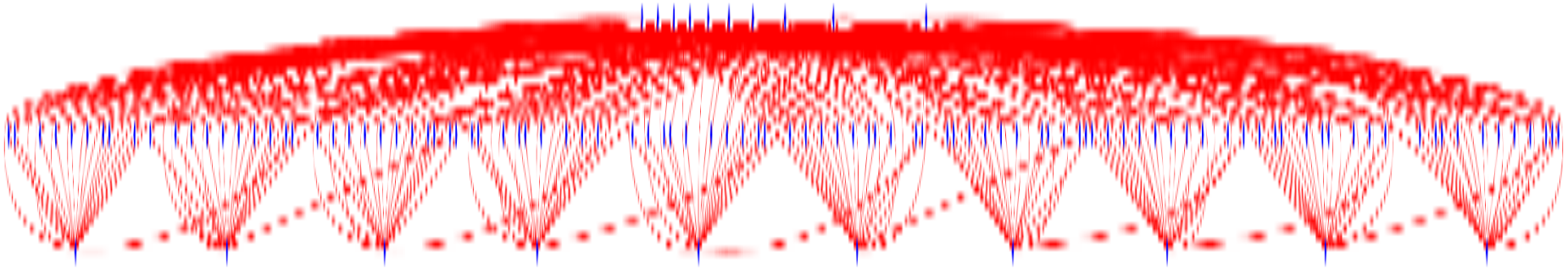


$\#Objects = 10$

$$\boldsymbol{Smokes(x) \wedge Friends(x,y) \Rightarrow \neg Asthma(y)}$$



$\#Objects = 5$



$\#Objects = 10$

$\#Objects = 50$

Outline

- Introduction
  - Motivation
  - MLN Basics
- Scalable Inference
  - Domain-Lifted Inference
  - Approximate Domain-Lifting
- Scalable Weight Learning
  - Lifted Generative Weight Learning
  - Discriminative/pseudo-likelihood learning with Approximate Counting Oracles
- Applications
  - Overview of MLN Software
  - NLP Applications using MLNs
- Conclusion

Domain-Lifted Inference

- Exploit symmetries, reason about groups of objects, reason at first-order level, etc (Poole '03)

Domain-Lifted Inference

- Exploit symmetries, reason about groups of objects, reason at first-order level, etc (Poole '03)
- More formally,
  - Inference runs in time polynomial in the number of domain objects (Broeck '11, Jaeger '12)

Domain-Lifted Inference

- Informally,
  - Exploit symmetries, reason about groups of objects, reason at first-order level, etc (Poole '03)
- More formally,
  - Inference runs in time polynomial in the number of domain objects (Broeck '11, Jaeger '12)
- Exact lifted inference
  - Lifted Factor Graphs (Poole '03)
  - First-Order Variable Elimination (Braz et al. '07)
  - Weighted First-Order Model Counting (Broeck et al. '10)
  - Probabilistic Theorem Proving (Gogate and Domingos '11)

Approximate Inference

- Similar to inference in Markov Networks, exact inference in MLNs is infeasible in practical cases. Most practical cases use approximate inference

Approximate Inference

- Similar to inference in Markov Networks, exact inference in MLNs is infeasible in practical cases. Most practical cases use approximate inference
- Domain-lifted versions of several popular approximate inference algorithms have been developed over past few years
- Marginal Inference
  - Lifted Belief Propagation (Singla and Domingos '07, Kersting et al. '08)
  - Lifted MCMC (Niepert '12, Venugopal and Gogate '12)
  - Lifted Importance Sampling (Gogate et al. '12)
- MAP Inference
  - Lifted MAP (Sarkhel et al. '14, '15, Kersting et al. '14, '15)

Lifted Belief Propagation

- Exploits the fact that symmetries in MLNs cause similar messages to be sent and received in BP

Lifted Belief Propagation

- Exploits the fact that symmetries in MLNs cause similar messages to be sent and received in BP
- Creates a lifted network for BP that can sometimes be exponentially smaller than the ground network for BP
  - Groups atoms that send and receive same messages into supernodes
  - Groups ground formulas that send and receive the same messages into superfeatures

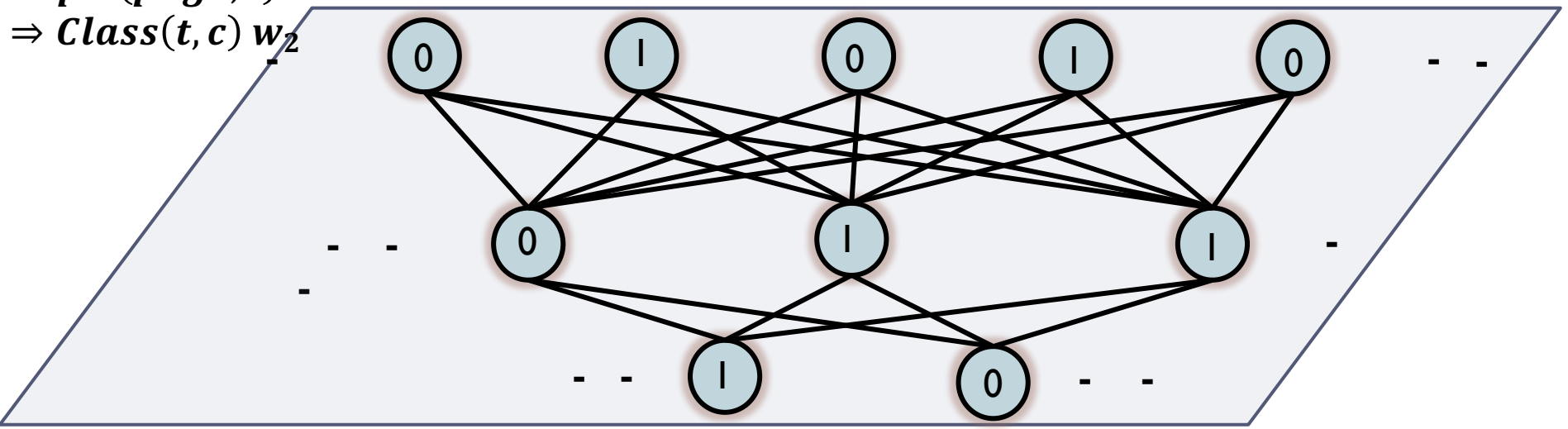
Lifted Blocked Gibbs Sampling

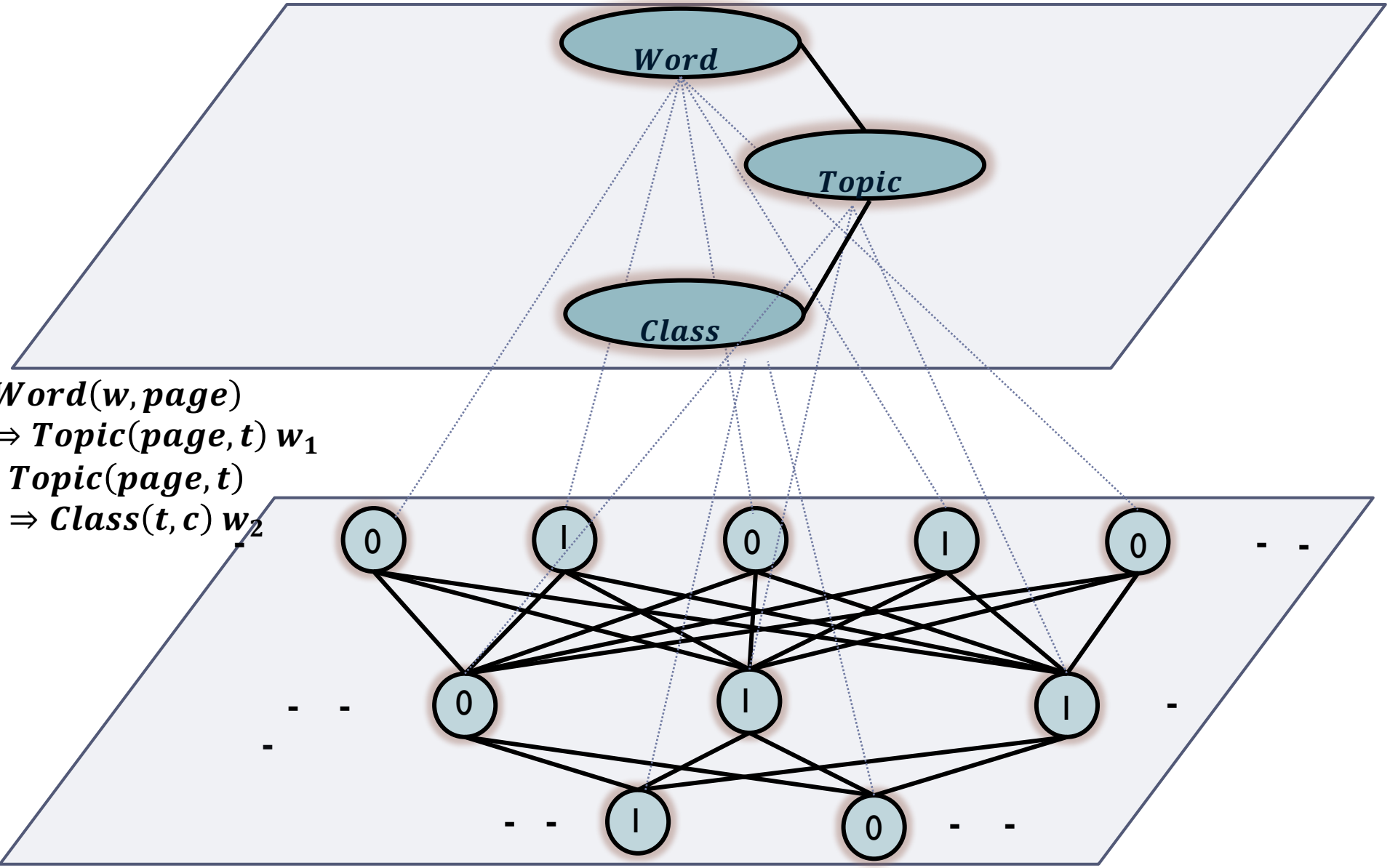
- Exploits the fact that using symmetries in MLNs, we can group atoms into blocks such that we only need to maintain sufficient statistics over the blocks

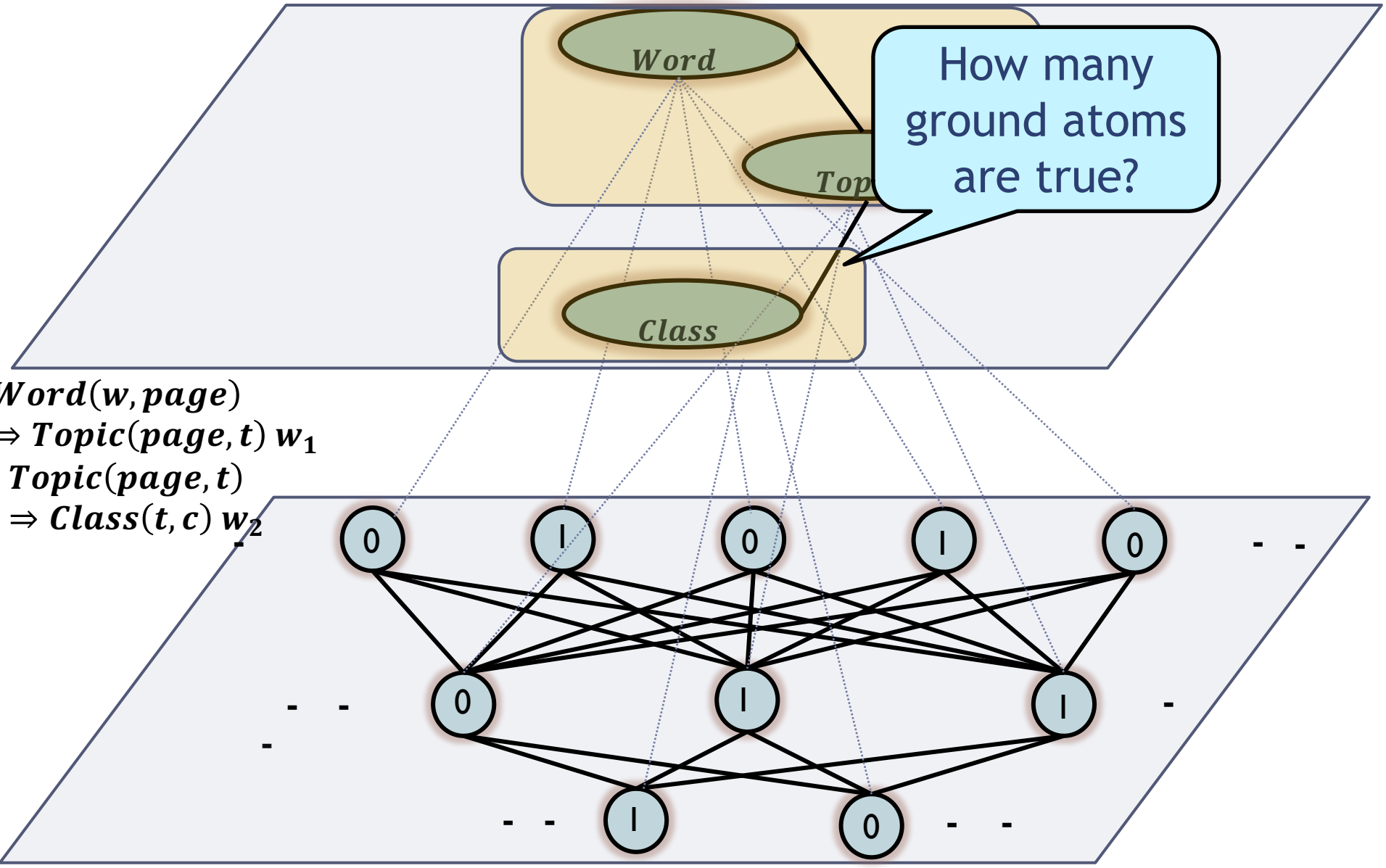
Lifted Blocked Gibbs Sampling

- Exploits the fact that using symmetries in MLNs, we can group atoms into blocks such that we only need to maintain sufficient statistics over the blocks
  - How many groundings are true vs which of the groundings are true?
  - The lifted sampling space is much smaller than the propositional sampling space

$Word(w, page)$
$\Rightarrow Topic(page, t)\ w_1$
$Topic(page, t)$
$\Rightarrow Class(t, c)\ w_2$




Word
Topic
Class
Word(w, page)
⇒ Topic(page, t) w1
Topic(page, t)
⇒ Class(t, c) w2


Word
Class
How many ground atoms are true?
$Word(w, page)$
$\Rightarrow Topic(page, t)$ w_1
$Topic(page, t)$
$\Rightarrow Class(t, c)$ w_2
0
1
0
1
0
0
1
1
1
0

Outline

- Introduction
  - Motivation
  - MLN Basics
- Scalable Inference
  - Domain-Lifted Inference
  - Approximate Domain-Lifting
- Scalable Weight Learning
  - Lifted Generative Weight Learning
  - Discriminative/pseudo-likelihood learning with Approximate Counting Oracles
- Applications
  - Overview of MLN Software
  - NLP Applications using MLNs
- Conclusion

Challenges to applying Lifted Inference in Practice

- Applicable to a very small subset of MLN structures

Challenges to applying Lifted Inference in Practice

- Applicable to a very small subset of MLN structures
- MLNs that contain less than 2 variables in a formula are domain-liftable
- $Token(t, i, c) \wedge Infield(i, f, c) \wedge Token(t, i', c') \wedge Infield(i', f, c') \Rightarrow SameField(f, c, c')$ (not liftable)

Broeck '11 Jaeger '12

Evidence Problem

$$\forall x, \forall y\ Strong(x) \Rightarrow Wins(x, y);\ 1.75$$

Atom	Pre-Evidence Marginals
$Wins(A, A)$	0.56
$Wins(A, B)$	0.56
$Wins(A, C)$	0.56
$Wins(B, A)$	0.56
$Wins(B, B)$	0.56
$Wins(B, C)$	0.56
$Wins(C, A)$	0.56
$Wins(C, B)$	0.56
$Wins(C, C)$	0.56

Evidence Problem

$$\forall x, \forall y\ Strong(x) \Rightarrow Wins(x, y);\ 1.75$$

Evidence(True) Atoms
$Strong(C)$
$Wins(A, C)$
$Wins(B, B)$
$Wins(B, C)$
$Wins(C, A)$

Atom	Pre-Evidence Marginals
$Wins(A, A)$	0.56
$Wins(A, B)$	0.56
$Wins(A, C)$	0.56
$Wins(B, A)$	0.56
$Wins(B, B)$	0.56
$Wins(B, C)$	0.56
$Wins(C, A)$	0.56
$Wins(C, B)$	0.56
$Wins(C, C)$	0.56

Atom	Evidence conditioned Marginals
$Wins(A, A)$	0.6
$Wins(A, B)$	0.6
$Wins(B, A)$	0.63
$Wins(C, B)$	0.85
$Wins(C, C)$	0.85

Evidence Problem

$$\forall x, \forall y\ Strong(x) \Rightarrow Wins(x, y);\ 1.75$$

Evidence(True) Atoms
$Strong(C)$
$Wins(A, C)$
$Wins(B, B)$
$Wins(B, C)$
$Wins(C, A)$

Atom	Pre-Evidence Marginals
$Wins(A, A)$	0.56
$Wins(A, B)$	0.56
$Wins(A, C)$	0.56
$Wins(B, A)$	0.56
$Wins(B, B)$	0.56
$Wins(B, C)$	0.56
$Wins(C, A)$	0.56
$Wins(C, B)$	0.56
$Wins(C, C)$	0.56

Atom	Evidence conditioned Marginals
$Wins(A, A)$	0.6
$Wins(A, B)$	0.6
$Wins(B, A)$	0.63
$Wins(C, B)$	0.85
$Wins(C, C)$	0.85

Inference with evidence only on unary predicates is efficient while inference with evidence on binary predicates is #P-hard (Broeck and Darwiche '13)

Approximate Lifting

- Exact symmetries are rare in practice

Approximate Lifting

- Exact symmetries are rare in practice

- There may be approximate symmetries, and exploiting them could scale up inference in non-liftable MLNs
  - Trade-off guarantees with computational complexity

Approximate Lifting

- Exact symmetries are rare in practice
- There may be approximate symmetries, and exploiting them could scale up inference in non-liftable MLNs
  - Trade-off guarantees with computational complexity
- Challenge: how do we identify approximate symmetries?

Boolean Factorization

- Conditioning on unary atoms is easy while conditioning on binary atoms is hard

Broeck and Darwiche '13

Boolean Factorization

- Conditioning on unary atoms is easy while conditioning on binary atoms is hard
- Represent binary evidence as a combination of unary evidences

Broeck and Darwiche '13

Boolean Factorization

- Conditioning on unary atoms is easy while conditioning on binary atoms is hard
- Represent binary evidence as a combination of unary evidences

Given binary evidence $P(A, A) = P(A, B) = P(B, A) = P(B, B) = P(C, A) = P(D, A) = P(D, D) = 1$, and the rest of the atoms are 0, we can represent evidence as a matrix

$$\begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \end{bmatrix}$$

Broeck and Darwiche '13

Boolean Factorization of Evidence

- Decompose the evidence matrix into binary products of vectors

Boolean Factorization of Evidence

- Decompose the evidence matrix into binary products of vectors

In general, we can factorize the binary evidence as

$$P = QR^T$$

where $QR^T = q_1 r_1^T \vee q_2 r_2^T \vee \cdots q_n r_n^T$. n is the Boolean rank of P

$$\begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 1 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 0 \\ 1 \end{bmatrix}^T \vee \begin{bmatrix} 1 \\ 1 \\ 0 \\ 1 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \end{bmatrix}^T \vee \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}^T$$

Boolean rank = 3

Boolean Factorization of Evidence

- Inference complexity can be characterized using Boolean rank of the evidence matrix
  - Inference is exponential in Boolean rank

Boolean Factorization of Evidence

- Inference complexity can be characterized using Boolean rank of the evidence matrix
  - Inference is exponential in Boolean rank
- How can we use this to introduce artificial symmetries?
  - Low rank Boolean matrix factorization
  - Bounding the rank of the factorization will automatically smooth the evidence introducing more symmetries

Boolean Factorization of Evidence

- Low rank approximation

$$\begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 1 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 0 \\ 1 \end{bmatrix}^T \vee \begin{bmatrix} 1 \\ 1 \\ 0 \\ 1 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \end{bmatrix}^T$$

Boolean Factorization of Evidence

- Low rank approximation

$$\begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 1 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 0 \\ 1 \end{bmatrix}^T \vee \begin{bmatrix} 1 \\ 1 \\ 0 \\ 1 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \end{bmatrix}^T$$

Boolean rank 2 approximation $\begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 \\ 0 & 0 & \mathbf{0} & 0 \\ 1 & 0 & 0 & 1 \end{bmatrix}$

The new evidence has more symmetries than the original evidence

Boolean Factorization of Evidence

Link(aaai,google)

Link(google,aaai)

Link(google,gmail)

Link(IBM,aaai)

Boolean Factorization of Evidence

Link(aaai,google)
Link(google,aaai)
Link(google,gmail)
Link(IBM,aaai)

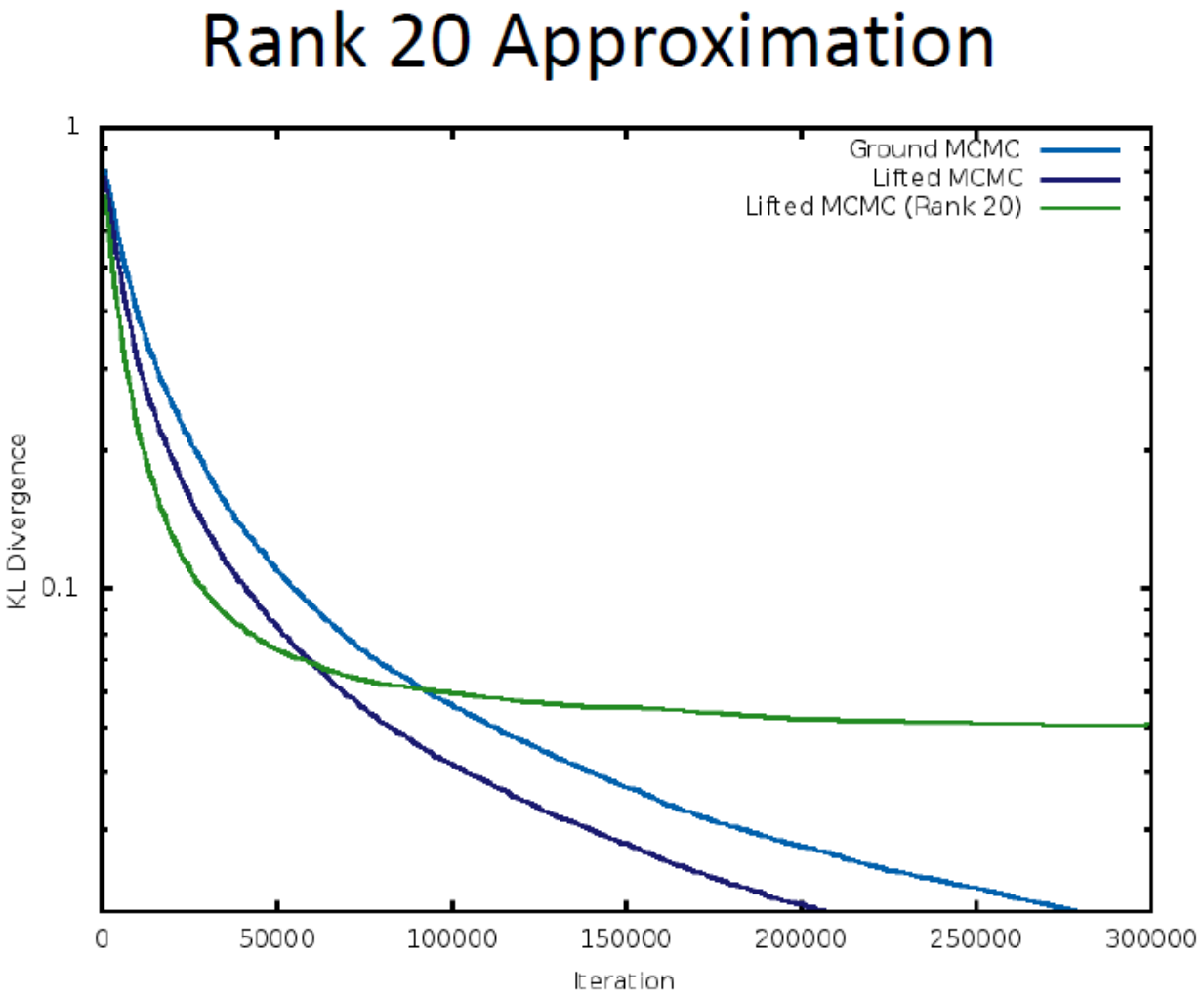
Link(aaai,google)
Link(google,aaai)
~~Link(google,gmail)~~
Link(IBM,aaai)
Link(aaai,IBM)

The objects google and IBM become more symmetric

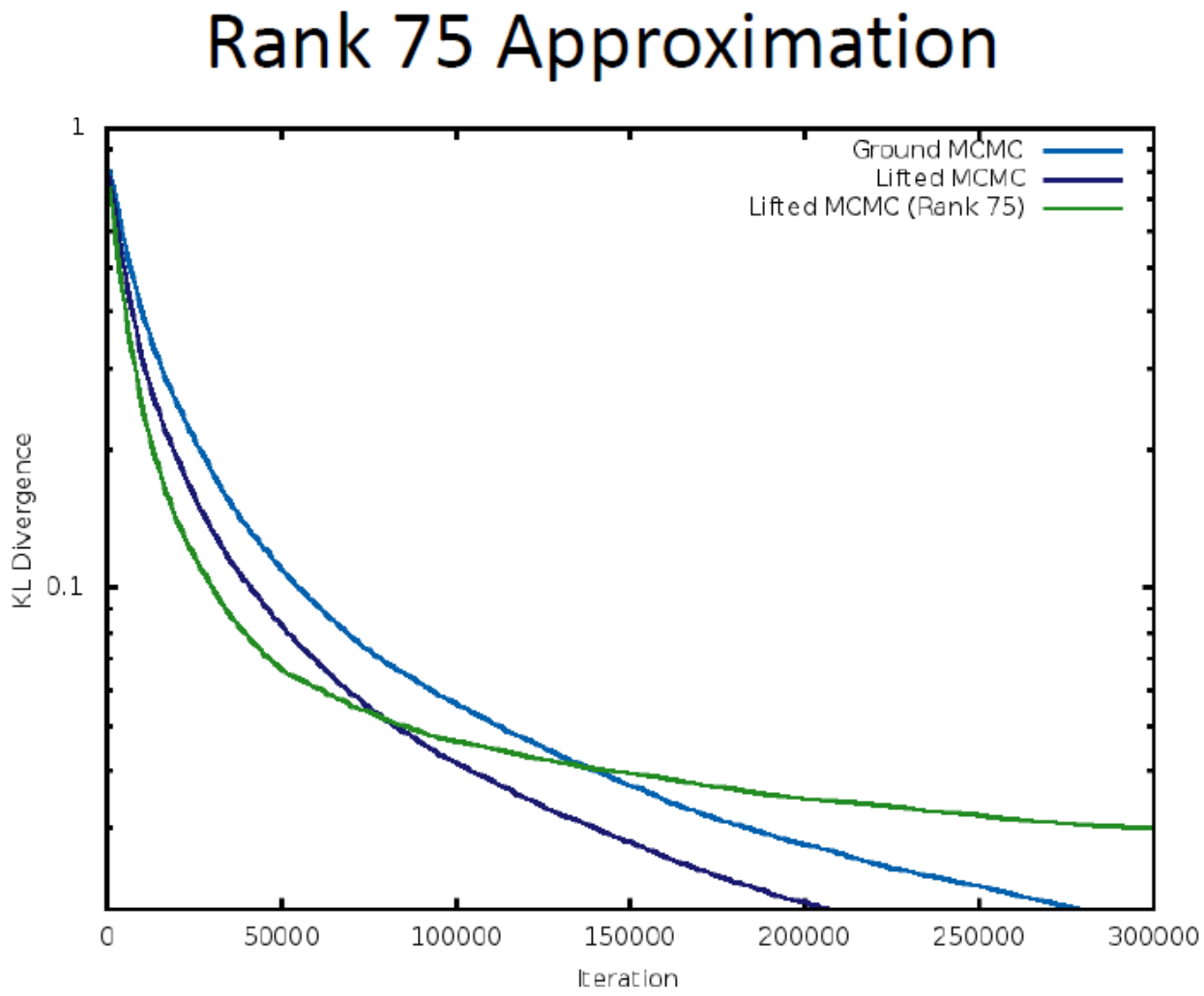
Boolean Factorization of Evidence

- Lift the MLN by converting binary evidences into their low rank approximations
- Problem: No guarantees on results since we are changing the underlying MLN distribution

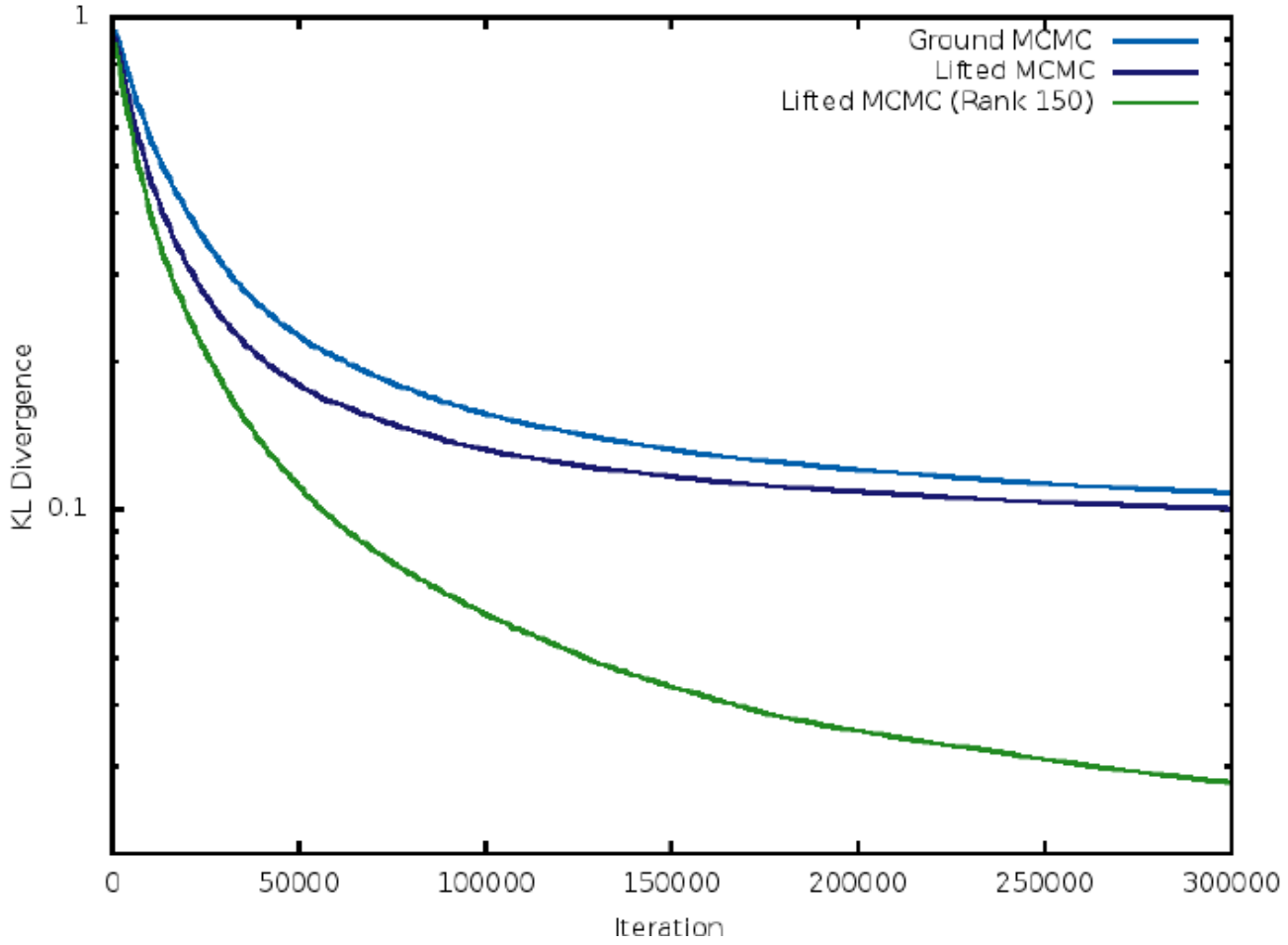
Performance on WebKB



Performance on WebKB



Performance on WebKB



Approximate Lifting as clustering

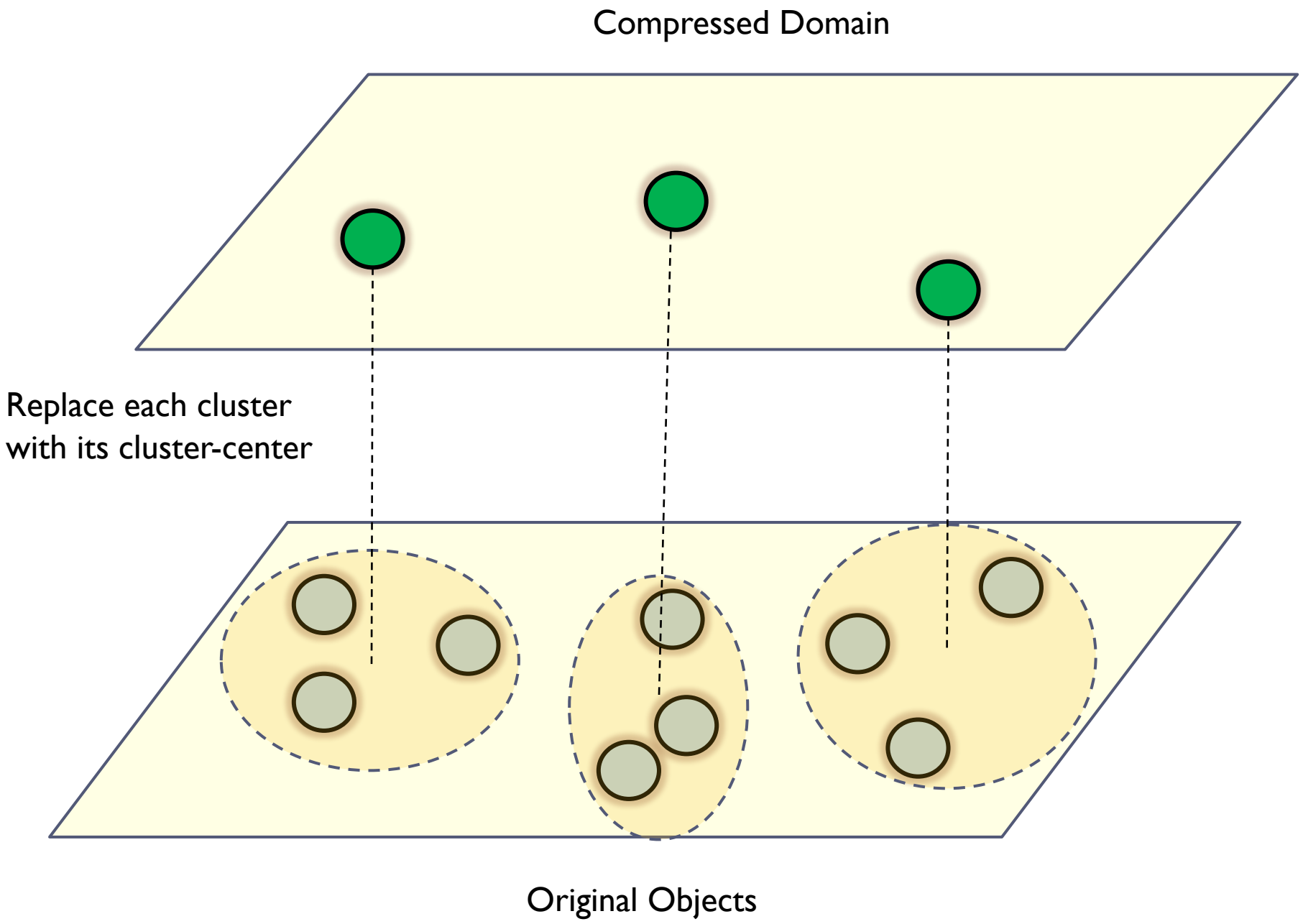
- Learn approximate symmetries using clustering methods such as K-Means

Venugopal and Gogate '14

Approximate Lifting as clustering

- Learn approximate symmetries using clustering methods such as K-Means
- Example: $word(w,p) \wedge topic(p,t) \wedge linked(p,q) \Longrightarrow topic(q,t)$
  - Is $P\big(topic(*, baseball)\big) \approx P\big(topic(*, basketball)\big)$?
  - Cluster similar topics together to compress the MLN

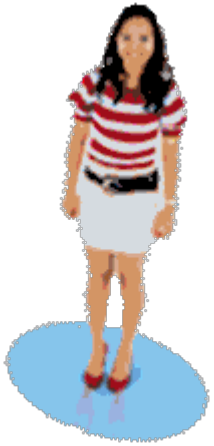
Venugopal and Gogate '14


Compressed Domain
Replace each cluster
with its cluster-center
Original Objects

Features for Clustering

Alice

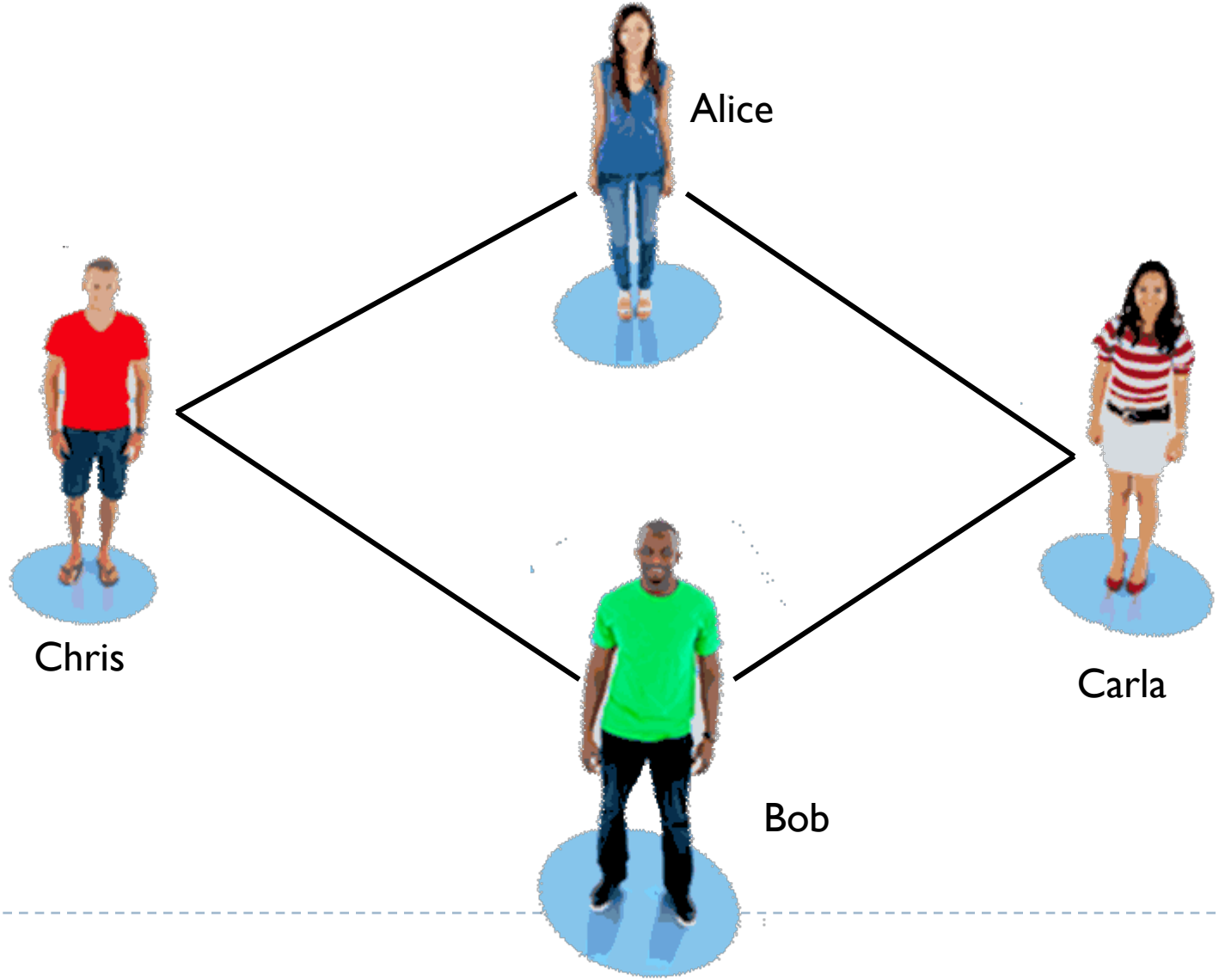
Chris



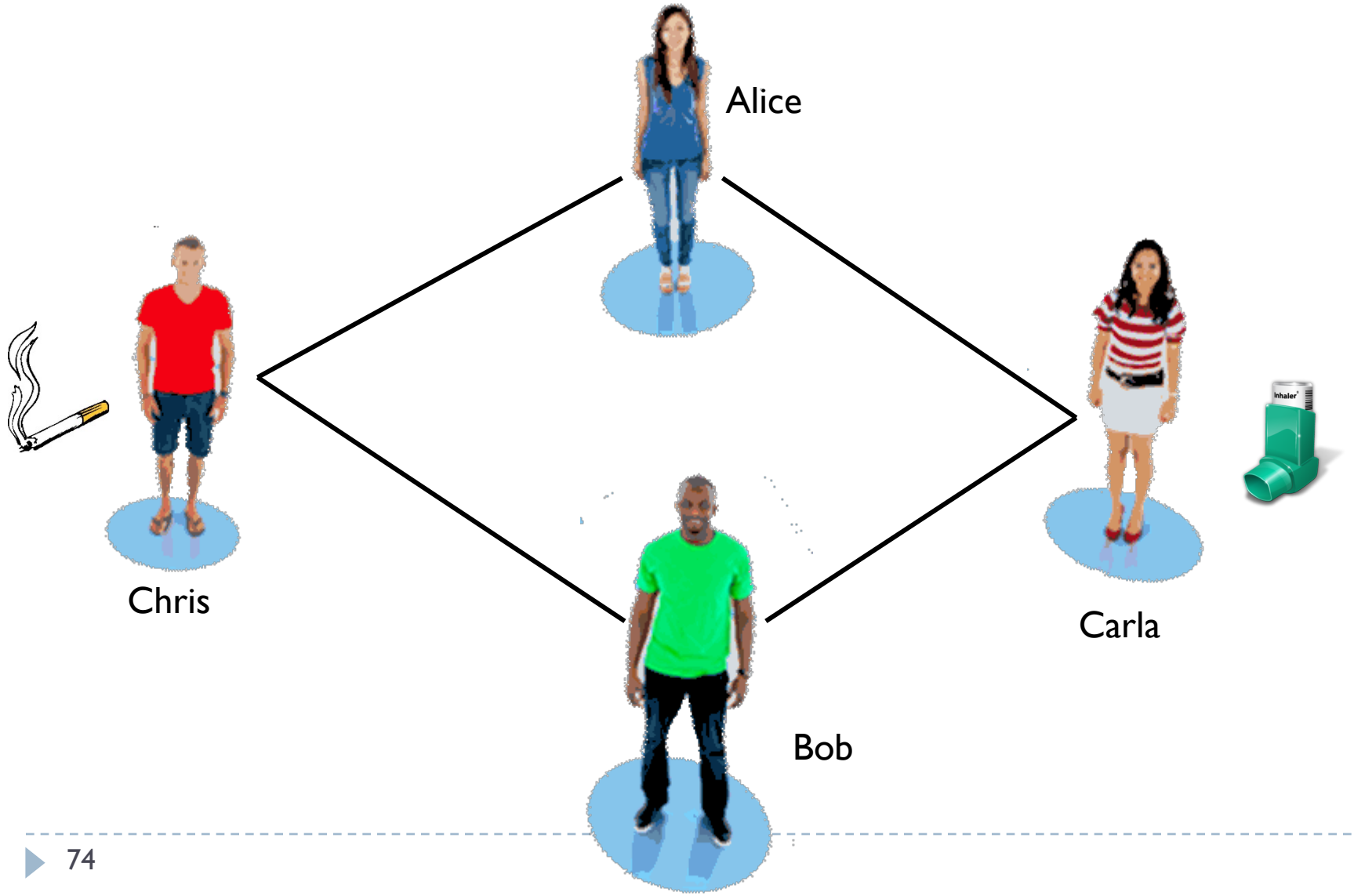
Carla

Bob

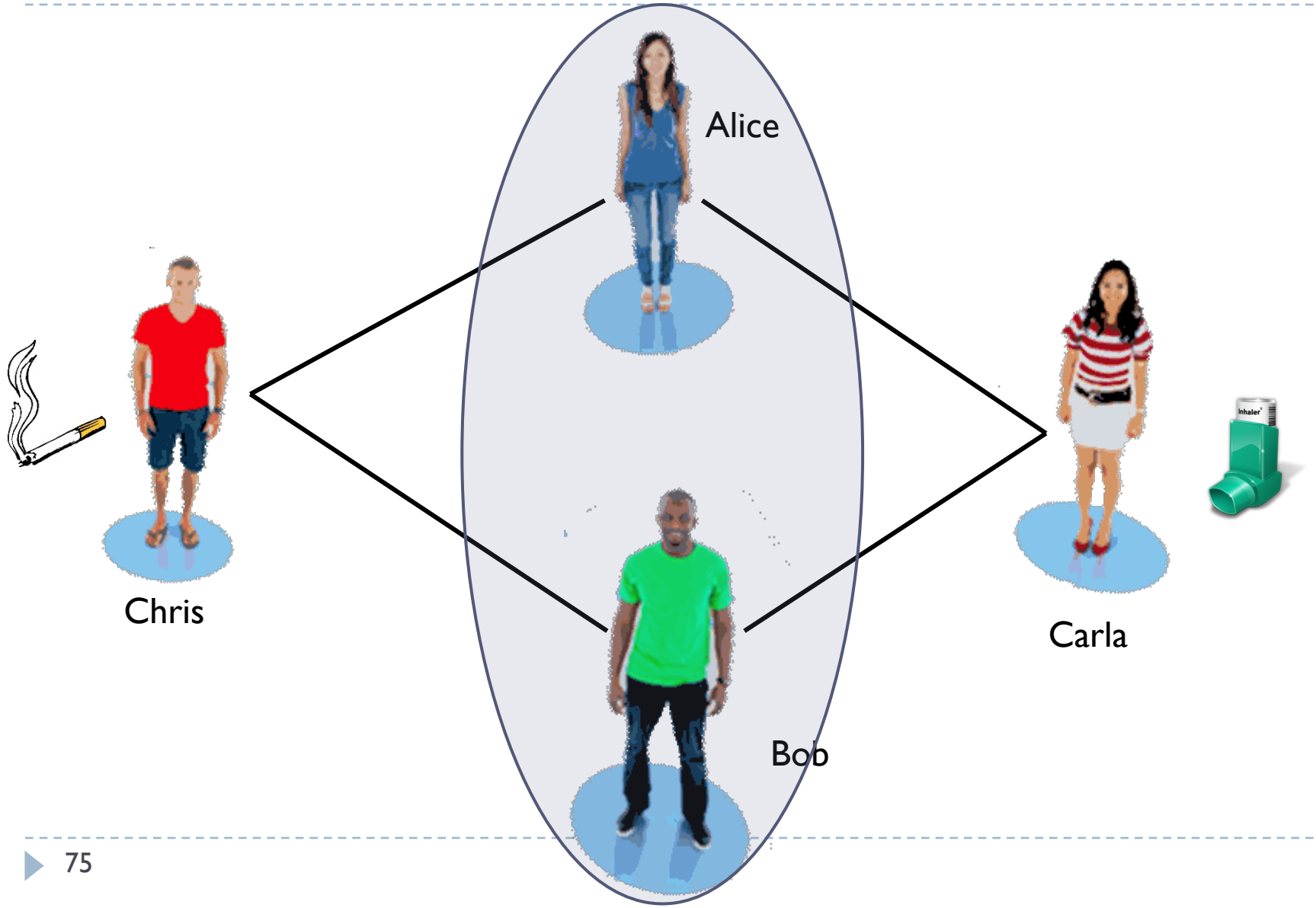
Features for Clustering



Features for Clustering

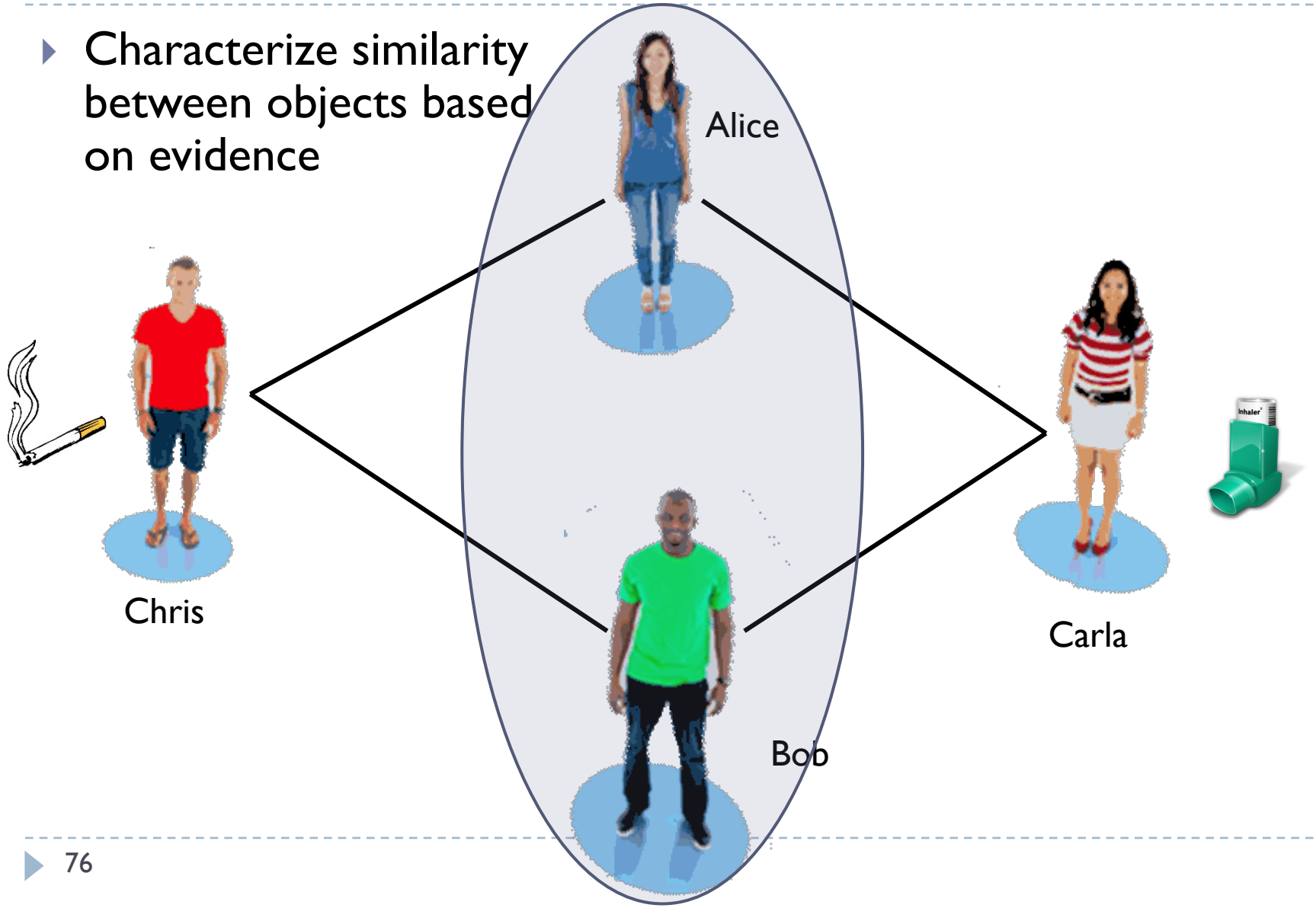


Features for Clustering



Features for Clustering

- Characterize similarity between objects based on evidence



Features for Clustering

- Extract features based on evidence presented to the MLN

Features for Clustering

- Extract features based on evidence presented to the MLN
- Idea: Objects that have similar evidence in their Markov blanket tend to be similar

$$\forall x\, \forall y\, \boldsymbol{Smokes}(x) \wedge \boldsymbol{Friends}(x, y) \Rightarrow \neg \boldsymbol{Asthma}(y) \quad 0.75$$

$$\boldsymbol{Smokes}(\boldsymbol{Ana}) \wedge \boldsymbol{Friends}(\boldsymbol{Ana}, \boldsymbol{Bob}) \Rightarrow \neg \boldsymbol{Asthma}(\boldsymbol{Bob}) \; \boldsymbol{0.75}$$

$$\boldsymbol{Smokes}(\boldsymbol{Bob}) \wedge \boldsymbol{Friends}(\boldsymbol{Bob}, \boldsymbol{Ana}) \Rightarrow \neg \boldsymbol{Asthma}(\boldsymbol{Ana}) \; \boldsymbol{0.75}$$

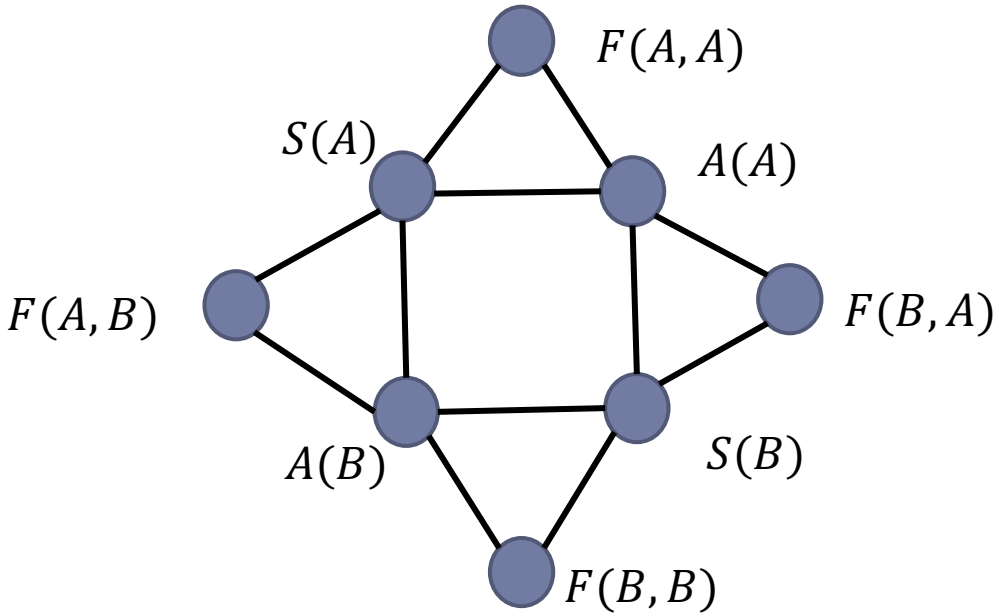
$$\boldsymbol{Smokes}(\boldsymbol{Bob}) \wedge \boldsymbol{Friends}(\boldsymbol{Bob}, \boldsymbol{Bob}) \Rightarrow \neg \boldsymbol{Asthma}(\boldsymbol{Bob}) \; \boldsymbol{0.75}$$

$$\forall x\, \forall y\, \boldsymbol{Smokes}(x) \wedge \boldsymbol{Friends}(x, y) \Rightarrow \neg \boldsymbol{Asthma}(y) \quad \mathbf{0.75}$$

$$\boldsymbol{Smokes}(\boldsymbol{Ana}) \wedge \boldsymbol{Friends}(\boldsymbol{Ana}, \boldsymbol{Bob}) \Rightarrow \neg \boldsymbol{Asthma}(\boldsymbol{Bob}) \; \mathbf{0.75}$$

$$\boldsymbol{Smokes}(\boldsymbol{Bob}) \wedge \boldsymbol{Friends}(\boldsymbol{Bob}, \boldsymbol{Ana}) \Rightarrow \neg \boldsymbol{Asthma}(\boldsymbol{Ana}) \; \mathbf{0.75}$$

$$\boldsymbol{Smokes}(\boldsymbol{Bob}) \wedge \boldsymbol{Friends}(\boldsymbol{Bob}, \boldsymbol{Bob}) \Rightarrow \neg \boldsymbol{Asthma}(\boldsymbol{Bob}) \; \mathbf{0.75}$$

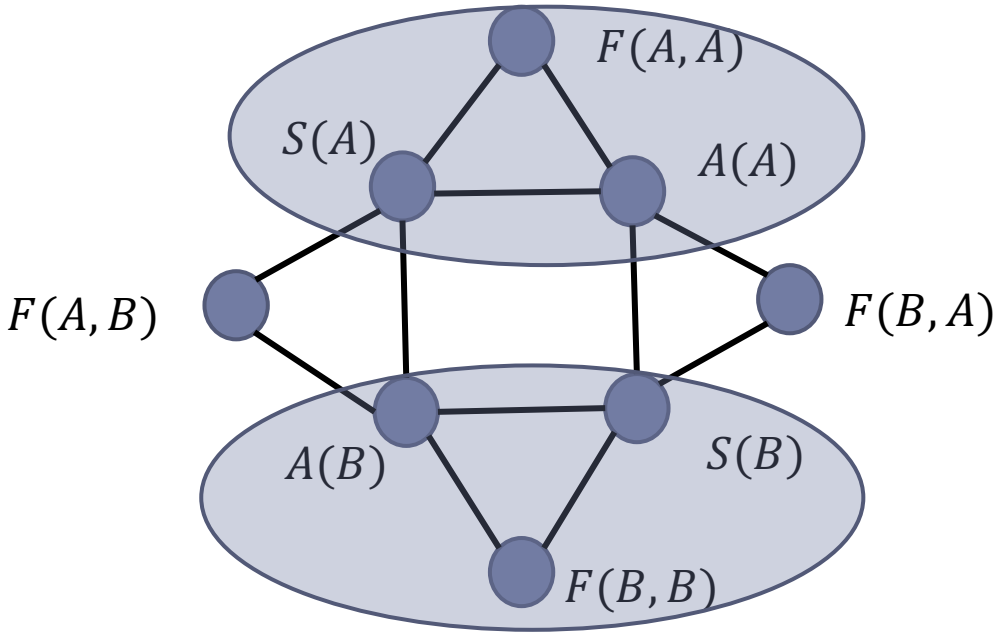


$$\forall x\, \forall y\, \boldsymbol{Smokes}(x) \wedge \boldsymbol{Friends}(x, y) \Rightarrow \neg \boldsymbol{Asthma}(y) \quad \mathbf{0.75}$$

$$\boldsymbol{Smokes}(\boldsymbol{Ana}) \wedge \boldsymbol{Friends}(\boldsymbol{Ana}, \boldsymbol{Bob}) \Rightarrow \neg \boldsymbol{Asthma}(\boldsymbol{Bob}) \; \mathbf{0.75}$$

$$\boldsymbol{Smokes}(\boldsymbol{Bob}) \wedge \boldsymbol{Friends}(\boldsymbol{Bob}, \boldsymbol{Ana}) \Rightarrow \neg \boldsymbol{Asthma}(\boldsymbol{Ana}) \; \mathbf{0.75}$$

$$\boldsymbol{Smokes}(\boldsymbol{Bob}) \wedge \boldsymbol{Friends}(\boldsymbol{Bob}, \boldsymbol{Bob}) \Rightarrow \neg \boldsymbol{Asthma}(\boldsymbol{Bob}) \; \mathbf{0.75}$$



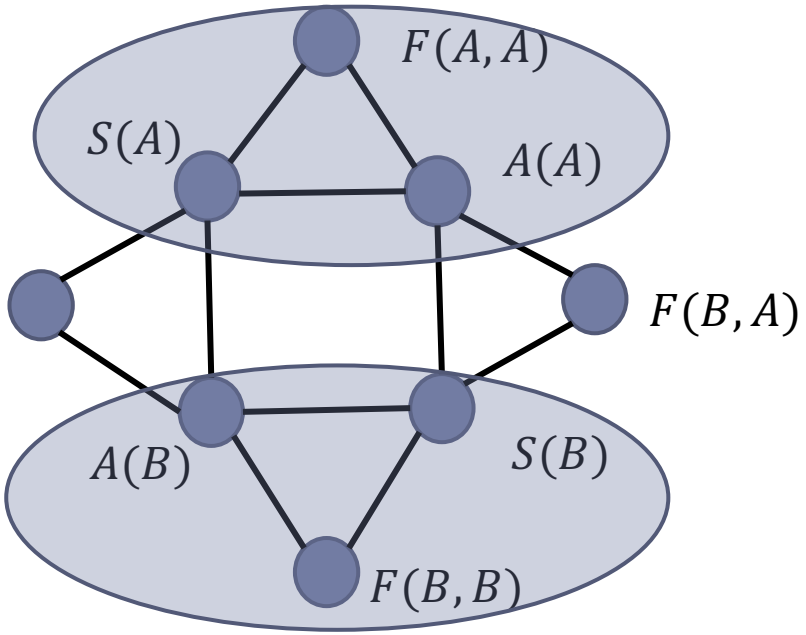
$$\forall x\, \forall y\, \boldsymbol{Smokes}(x) \wedge \boldsymbol{Friends}(x, y) \Rightarrow \neg \boldsymbol{Asthma}(y) \quad \mathbf{0.75}$$

$$\boldsymbol{Smokes}(\boldsymbol{Ana}) \wedge \boldsymbol{Friends}(\boldsymbol{Ana}, \boldsymbol{Bob}) \Rightarrow \neg \boldsymbol{Asthma}(\boldsymbol{Bob}) \; \mathbf{0.75}$$

$$\boldsymbol{Smokes}(\boldsymbol{Bob}) \wedge \boldsymbol{Friends}(\boldsymbol{Bob}, \boldsymbol{Ana}) \Rightarrow \neg \boldsymbol{Asthma}(\boldsymbol{Ana}) \; \mathbf{0.75}$$

$$\boldsymbol{Smokes}(\boldsymbol{Bob}) \wedge \boldsymbol{Friends}(\boldsymbol{Bob}, \boldsymbol{Bob}) \Rightarrow \neg \boldsymbol{Asthma}(\boldsymbol{Bob}) \; \mathbf{0.75}$$

Number of satisfied groundings containing Ana = Number of satisfied groundings containing Bob
=> Ana is symmetric to Bob



Features for Clustering

- For every object count the number of satisfied groundings it appears in for each formula

Features for Clustering

- For every object count the number of satisfied groundings it appears in for each formula
- Symmetries change as we change the evidence database

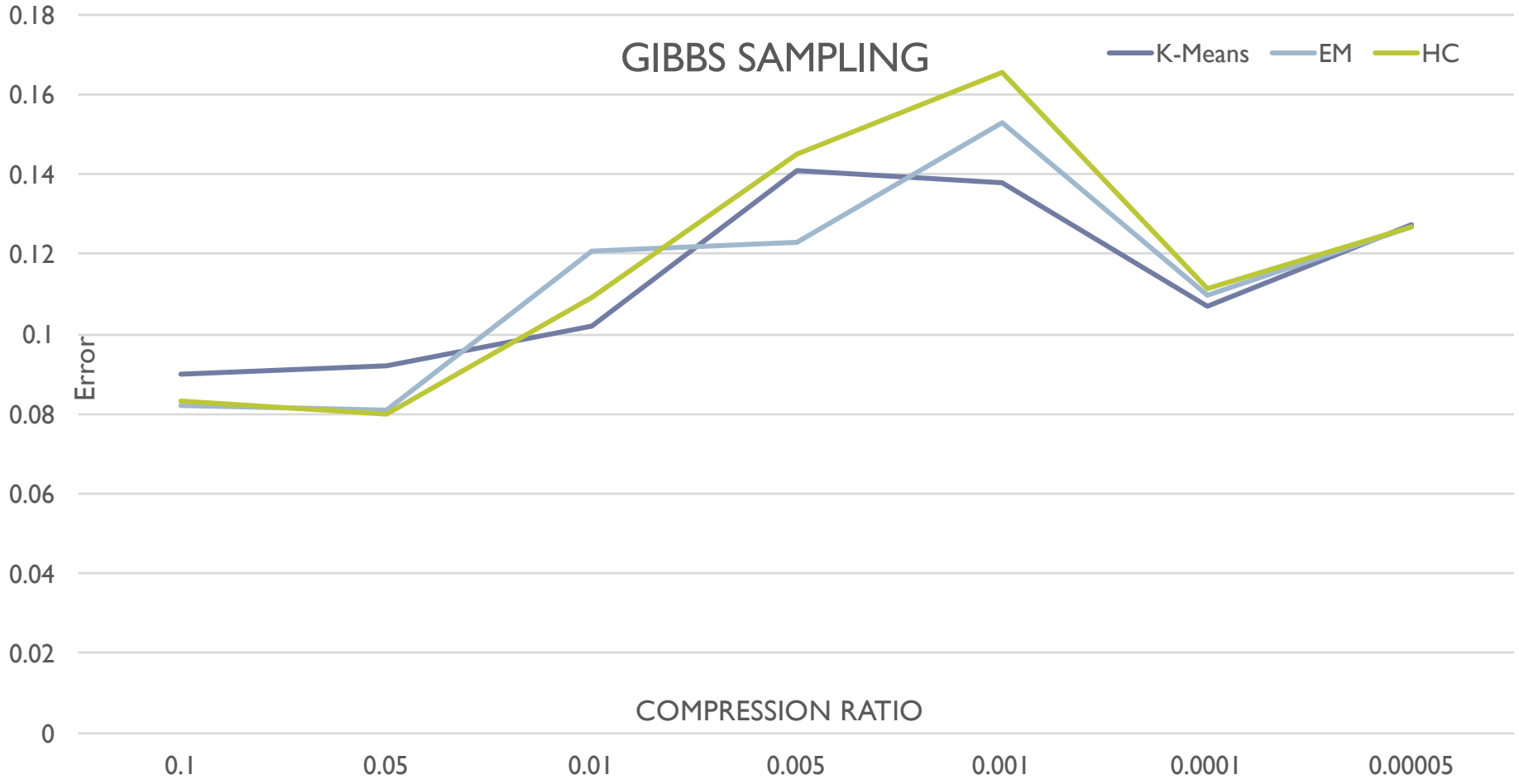
Features for Clustering

- For every object count the number of satisfied groundings it appears in for each formula
- Symmetries change as we change the evidence database
- Assume that results obtained on cluster center can be projected to all objects in the cluster

Features for Clustering

- For every object count the number of satisfied groundings it appears in for each formula
- Symmetries change as we change the evidence database
- Assume that results obtained on cluster center can be projected to all objects in the cluster
- Changes the MLN distribution (we are reducing a cluster of objects with one object)
  - No guarantees on solution

Performance: Citation Segmentation



Non-Parametric Approximate Lifting

- How many clusters should we choose to reasonably approximate the original distribution?

Venugopal et al. '16

Non-Parametric Approximate Lifting

- How many clusters should we choose to reasonably approximate the original distribution?
- Integrate Bayesian non-parametrics with K-Means clustering

Venugopal et al. '16

Non-Parametric Approximate Lifting

- How many clusters should we choose to reasonably approximate the original distribution?
- Integrate Bayesian non-parametrics with K-Means clustering
- Use DP-Means (Kullis and Jordan '11)
  - K-Means like objective with an additional penalty term for the number of clusters

$$\min_{\{\ell_{cj}\}_{c=1;j=1}^{|D'_j|;M}} \sum_{j=1}^{M} \sum_{c=1}^{|D'_j|} \sum_{\mathbf{x} \in \ell_{cj}} ||\mathbf{x} - \mu_{cj}||^2 + \lambda |D'_j|$$

Venugopal et al. '16

Non-Parametric Approximate lifting

- Compute the clustering by tuning the penalty hyper-parameter

Non-Parametric Approximate lifting

- Compute the clustering by tuning the penalty hyper-parameter
- Decrease penalty, find the optimal clustering, and verify if constraints are satisfied

Non-Parametric Approximate lifting

- Compute the clustering by tuning the penalty hyper-parameter
- Decrease penalty, find the optimal clustering, and verify if constraints are satisfied
  - Expected error in samples drawn from approximate distribution is bounded
  - The total number of atoms in the MLN is bounded

Approximate lifting of BP

- Start with a coarse network and refine it by keeping track identical BP messages

Singla et al. '14

Approximate lifting of BP

- Start with a coarse network and refine it by keeping track identical BP messages
- Stop refinement when the number of supernodes or superfeatures reaches a pre-specified threshold

Singla et al. '14

Approximate lifting of BP

- Start with a coarse network and refine it by keeping track identical BP messages
- Stop refinement when the number of supernodes or superfeatures reaches a pre-specified threshold
- Lifted network is approximate but is much smaller than the propositional network

Singla et al. '14

Approximate Lifting with Guarantees

- Can we provide guarantees even if we use the approximate MLN distribution?

Approximate Lifting with Guarantees

- Can we provide guarantees even if we use the approximate MLN distribution?
- Instead of using the approximate MLN directly for inference, use it as a proposal distribution within samplers

Approximate Lifting with Guarantees

- Can we provide guarantees even if we use the approximate MLN distribution?

- Instead of using the approximate MLN directly for inference, use it as a proposal distribution within samplers
  - Importance Sampling (Venugopal and Gogate '14)
  - Metropolis-Hastings (Niepert and Broeck '14)

Approximate Lifting with Guarantees

- Can we provide guarantees even if we use the approximate MLN distribution?

- Instead of using the approximate MLN directly for inference, use it as a proposal distribution within samplers
  - Importance Sampling (Venugopal and Gogate '14)
  - Metropolis-Hastings (Niepert and Broeck '14)

- Advantage: Guarantees that the estimates generated converge
  - Asymptotic Unbiasedness (Venugopal and Gogate '14)
  - Mix samples from approximate distribution with samples from true distribution (Niepert and Broeck '14)

Approximate Lifting with Guarantees

- Instead of using the approximate MLN directly for inference, use it as a proposal distribution within samplers
  - Importance Sampling (Venugopal and Gogate '14)
  - Metropolis-Hastings (Niepert and Broeck '14)

- Advantage: Guarantees that the estimates generated converge
  - Asymptotic Unbiasedness (Venugopal and Gogate '14)
  - Mix samples from approximate distribution with samples from true distribution (Niepert and Broeck '14)

- However, there are still some scalability issues. E.g. computing the importance weight or the transition probabilities turns out to be expensive for large MLNs

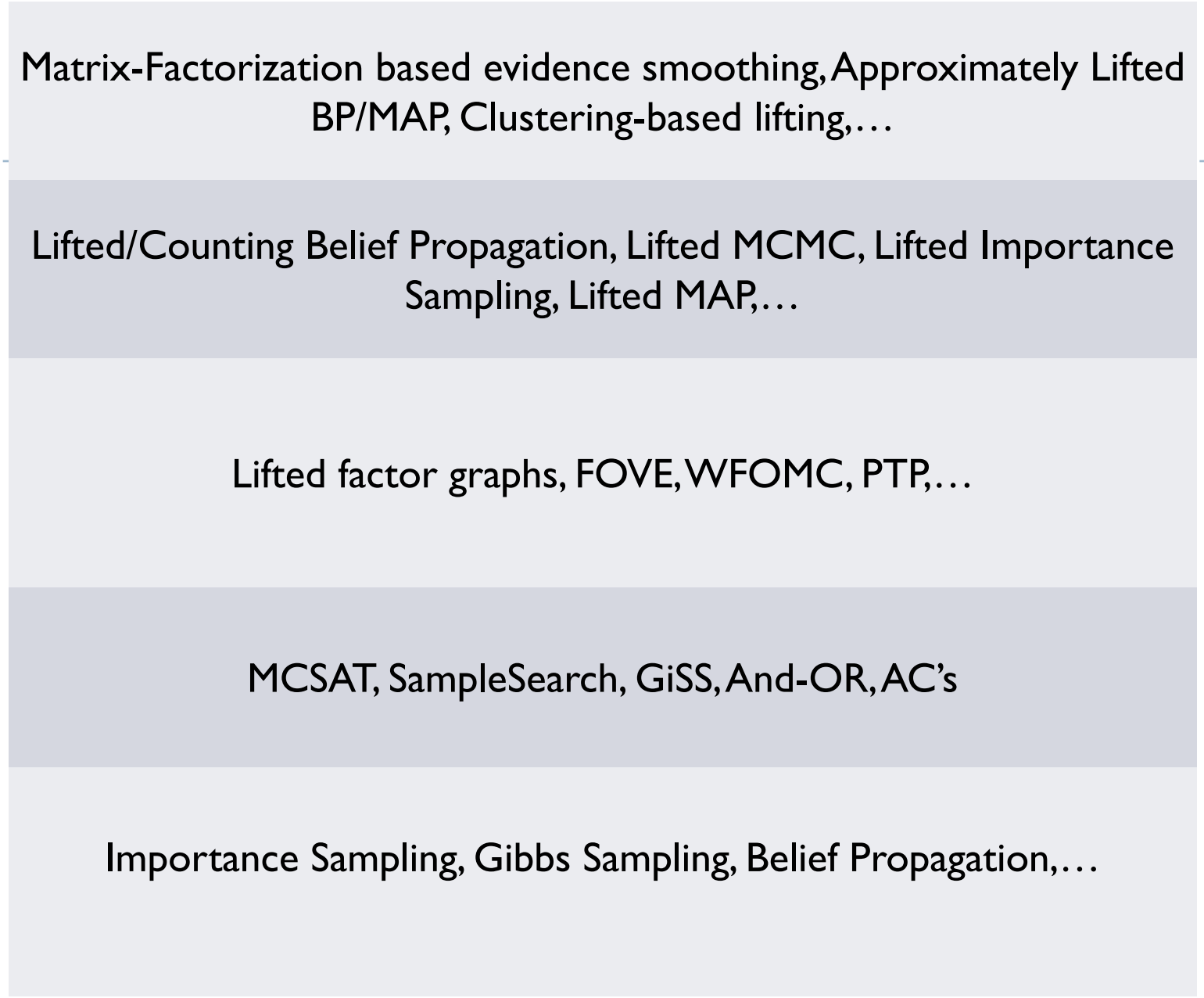
Matrix-Factorization based evidence smoothing, Approximately Lifted BP/MAP, Clustering-based lifting,…

Lifted/Counting Belief Propagation, Lifted MCMC, Lifted Importance Sampling, Lifted MAP,…

Lifted factor graphs, FOVE, WFOMC, PTP,…

MCSAT, SampleSearch, GiSS, And-OR, AC's

Importance Sampling, Gibbs Sampling, Belief Propagation,…


Matrix-Factorization based evidence smoothing, Approximately Lifted BP/MAP, Clustering-based lifting,…
Lifted/Counting Belief Propagation, Lifted MCMC, Lifted Importance Sampling, Lifted MAP,…
Lifted factor graphs, FOVE, WFOMC, PTP,…
MCSAT, SampleSearch, GiSS, And-OR, AC's
Importance Sampling, Gibbs Sampling, Belief Propagation,…
Exploit logical structure and symmetries

Outline

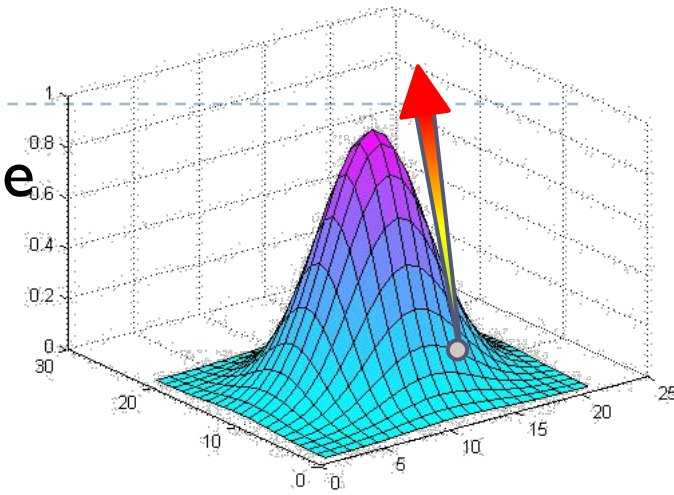
- Introduction
  - Motivation
  - MLN Basics
- Scalable Inference
  - Domain-Lifted Inference
  - Approximate Domain-Lifting
- Scalable Weight Learning
  - Lifted Generative Weight Learning
  - Discriminative/pseudo-likelihood learning with Approximate Counting Oracles
- Applications
  - Overview of MLN Software
  - NLP Applications using MLNs
- Conclusion

Weight Learning

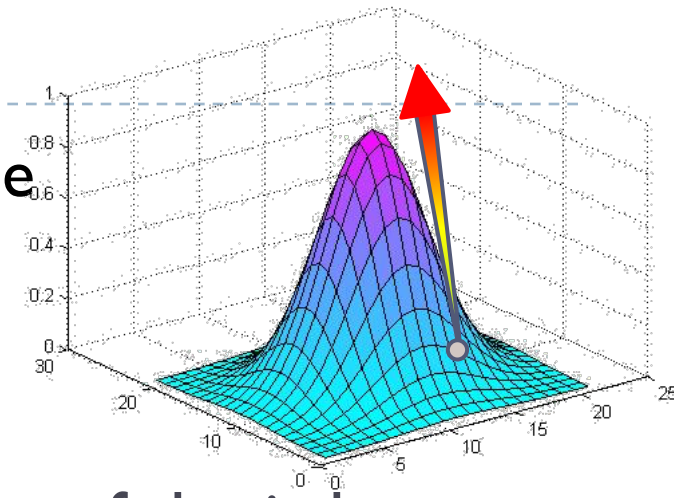
- Learning from single relational database
  - Computing the gradient is hard

Weight Learning

- Learning from single relational database
  - Computing the gradient is hard
- $\frac{\partial}{\partial \theta_i} \ell(\boldsymbol{\theta}: \omega) = N_i(\omega) - \mathbb{E}_{\boldsymbol{\theta}}[N_i(\omega)]$



Weight Learning



- Learning from single relational database
  - Computing the gradient is hard
- $\frac{\partial}{\partial \theta_i} \ell(\boldsymbol{\theta}: \omega) = N_i(\omega) - \mathbb{E}_{\boldsymbol{\theta}}[N_i(\omega)]$
  - $N_i(\omega)$ is the number of groundings of the i-th formula that is true in the training database
  - $\mathbb{E}_{\theta}[N_i]$ is the expected number of groundings of the i-th formula that is true in the data using the weights θ

Lifted Generative Learning

- Lifted generative learning uses a lifted inference oracle to compute $\mathbb{E}_\theta[N_i(\omega)]$
  - Note that there is no conditioning on evidence when computing the expectation

Haaren et al. '15

Lifted Generative Learning

- Lifted generative learning uses a lifted inference oracle to compute $\mathbb{E}_\theta[N_i(\omega)]$
  - Note that there is no conditioning on evidenced when computing the expectation
- Consider an MLN with a single formula F with N groundings $f_1, f_2, \ldots f_N$
- $\mathbb{E}_\theta[N_i(\omega)] = \mathrm{P}(f_1) + \mathrm{P}(f_2) + \cdots \mathrm{P}(f_N)$
- We require to compute marginal probabilities for each grounding of the formula

Haaren et al. '15

Lifted Generative Learning

- Two key benefits
  - Can compute gradient from a small number of marginal probabilities
  - Each marginal can be computed in a domain-lifted manner

Lifted Generative Learning

- Two key benefits
  - Can compute gradient from a small number of marginal probabilities
  - Each marginal can be computed in a domain-lifted manner
- Group the groundings into equiprobable sets, and compute the marginal over groups, multiply by group-size
  - $\mathbb{E}_\theta[N_i(\omega)] = |E_1|\mathrm{P}(f_{E_1}) + |E_2|\mathrm{P}(f_{E_2}) + \cdots |E_N|\mathrm{P}(f_{E_N})$

Lifted Generative Learning

- Two key benefits
  - Can compute gradient from a small number of marginal probabilities
  - Each marginal can be computed in a domain-lifted manner
- Group the groundings into equiprobable sets by shattering the MLN, and compute the marginal over groups, multiply by group-size
  - $\mathbb{E}_\theta[N_i(\omega)] = |E_1|\mathrm{P}(f_{E_1}) + |E_2|\mathrm{P}(f_{E_2}) + \cdots |E_N|\mathrm{P}(f_{E_N})$
- Compute the probability of each equiprobable set using a lifted inference oracle (WFOMC)
  - However, grouping may end up with too many groups (shattering)

Discriminative weight learning

- Query predicates are pre-specified, and we maximize conditional log-likelihood

Discriminative weight learning

- Query predicates are pre-specified, and we maximize conditional log-likelihood

- Techniques used in generative learning does not work well in discriminative learning
  - Conditioning on query variables destroys symmetries

Discriminative weight learning

- Query predicates are pre-specified, and we maximize conditional log-likelihood
- Techniques used in generative learning does not work well in discriminative learning
  - Conditioning on query variables destroys symmetries
- We can exploit MLN "shared-structure" for scalable discriminative learning

Discriminative learning using Approximate Counting Oracles

- Computing $N_i(\omega) - \mathbb{E}_\theta[N_i(\omega)]$ is a hard problem
  - Need to visit groundings of a formula

Discriminative learning using Approximate Counting Oracles

- Computing $N_i(\omega) - \mathbb{E}_\theta[N_i(\omega)]$ is a hard problem
  - Need to visit groundings of a formula

- How can we approximately count the satisfied groundings of a formula in a world without grounding the formula?

Discriminative learning using Approximate Counting Oracles

- Computing $N_i(\omega) - \mathbb{E}_\theta[N_i(\omega)]$ is a hard problem
  - Need to visit groundings of a formula

- How can we approximately count the satisfied groundings of a formula in a world without grounding the formula?

- Encode a formula as a graphical model such that computing the partition function of the encoded graphical model is equivalent to computing $N_i(\omega)$ (Venugopal et al. '15)

Discriminative learning using Approximate Counting Oracles

- Computing $N_i(\omega) - \mathbb{E}_\theta[N_i(\omega)]$ is a hard problem
  - Need to visit groundings of a formula
- How can we approximately count the satisfied groundings of a formula in a world without grounding the formula?
- Encode a formula as a graphical model such that computing the partition function of the encoded graphical model is equivalent to computing $N_i(\omega)$ (Venugopal et al. '15)
- Use approximate counting solvers to estimate the partition function efficiently

Scalable Weight Learning using Approximate Counting Oracles

$$f = \forall x\, \forall y\, \forall z\, \neg R(x, y) \vee S(y, z)$$
$$\Delta_x = \Delta_y = \Delta_z = \{A, B\}$$

Scalable Weight Learning using Approximate Counting Oracles

$$f = \forall x\, \forall y\, \forall z\, \neg R(x,y) \vee S(y,z)$$

$$\Delta_x = \Delta_y = \Delta_z = \{A, B\}$$

$R(A,A)$	1
$R(A,B)$	0
$R(B,A)$	0
$R(B,B)$	1

$S(A,A)$	0
$S(A,B)$	1
$S(B,A)$	0
$S(B,B)$	1

Scalable Weight Learning using Approximate Counting Oracles

$$f = \forall x\, \forall y\, \forall z\, \neg R(x,y) \vee S(y,z)$$

$$\Delta_x = \Delta_y = \Delta_z = \{A, B\}$$

$$f' = R(x,y) \wedge \neg S(y,z)$$

$R(A,A)$	1
$R(A,B)$	0
$R(B,A)$	0
$R(B,B)$	1

$S(A,A)$	0
$S(A,B)$	1
$S(B,A)$	0
$S(B,B)$	1

Scalable Weight Learning using Approximate Counting Oracles

$$f = \forall x\, \forall y\, \forall z\, \neg R(x,y) \vee S(y,z)$$

$$\Delta_x = \Delta_y = \Delta_z = \{A, B\}$$

$$f' = R(x,y) \wedge \neg S(y,z)$$

$R(A,A)$	1
$R(A,B)$	0
$R(B,A)$	0
$R(B,B)$	1

Identity Function

x	y	$\emptyset_1$
A	A	1
A	B	0
B	A	0
B	B	1

$S(A,A)$	0
$S(A,B)$	1
$S(B,A)$	0
$S(B,B)$	1

Inverse Function

y	z	$\emptyset_2$
A	A	1
A	B	0
B	A	1
B	B	0

Scalable Weight Learning using Approximate Counting Oracles

- Any exact or approximate counting oracle for the graphical model can be used to compute $N_i(\omega)$

Scalable Weight Learning using Approximate Counting Oracles

- Any exact or approximate counting oracle for the graphical model can be used to compute $N_i(\omega)$

d^n possible groundings

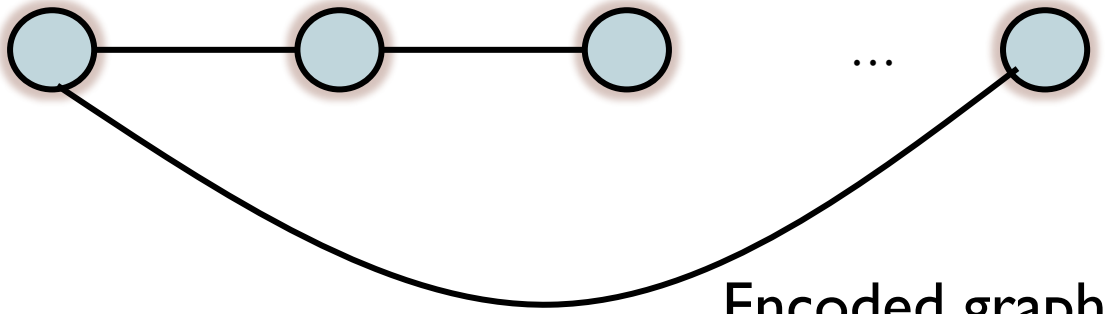
$$R_1(x_1, x_2) \vee R_2(x_2, x_3) \ldots R_n(x_n, x_1)$$

Scalable Weight Learning using Approximate Counting Oracles

- Any exact or approximate counting oracle for the graphical model can be used to compute $N_i(\omega)$

d^n possible groundings

$$R_1(x_1, x_2) \vee R_2(x_2, x_3) \ldots R_n(x_n, x_1)$$



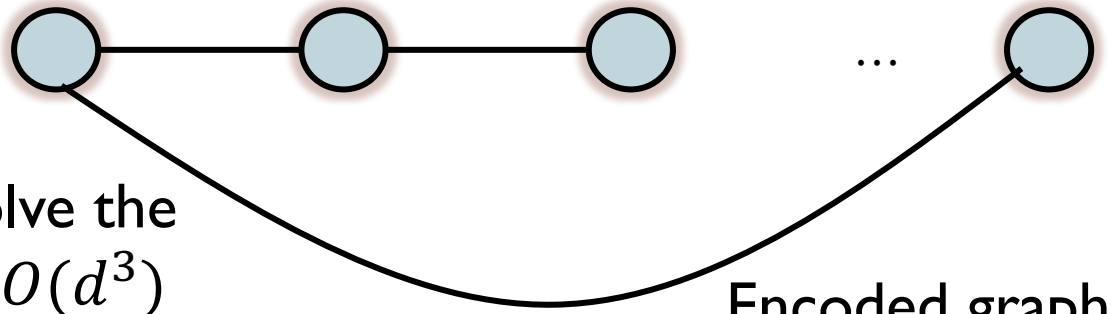
Encoded graphical model has tree-width 2

Scalable Weight Learning using Approximate Counting Oracles

- Any exact or approximate counting oracle for the graphical model can be used to compute $N_i(\omega)$

d^n possible groundings

$$R_1(x_1, x_2) \vee R_2(x_2, x_3) \ldots R_n(x_n, x_1)$$



Junction Trees can solve the counting problem in $O(d^3)$ time/space

Encoded graphical model has tree-width 2

Scalable Weight Learning with Approximate Counting Oracles

- The direction of the gradient is more important than obtaining the correct expectation
  - Use approximate counts to compute the gradient in each step of weight learning

Sarkhel et al. '16

Scalable Weight Learning with Approximate Counting Oracles

- The direction of the gradient is more important than obtaining the correct expectation
  - Use approximate counts to compute the gradient in each step of weight learning

- As the counts become more accurate, the parameters learned approach the true max-likelihood parameters
  - Existing counters such as iterative join graph propagation often yield excellent estimates of the counts, and their complexity can be controlled easily

Sarkhel et al. '16

Contrastive Divergence using Approximate Counters

- Approximate the expectation term in the gradient using a few Gibbs samples (Hinton '02)
  - The approximate gradient $\frac{\partial}{\partial \theta_i} = M_i(\omega) - \overline{\mathbb{E}_\theta}\left[M_i(\omega)\right]$

Contrastive Divergence using Approximate Counters

- Approximate the expectation term in the gradient using a few Gibbs samples (Hinton '02)
  - The approximate gradient $\frac{\partial}{\partial \theta_i} = M_i(\omega) - \overline{\mathbb{E}_\theta}\left[M_i(\omega)\right]$
  - $M_i(\omega)$ is obtained from the counting oracle

Contrastive Divergence using Approximate Counters

- Approximate the expectation term in the gradient using a few Gibbs samples (Hinton '02)
  - The approximate gradient $\frac{\partial}{\partial \theta_i} = M_i(\omega) - \overline{\mathbb{E}_\theta}\left[M_i(\omega)\right]$
  - $M_i(\omega)$ is obtained from the counting oracle
  - $\overline{\mathbb{E}_\theta}\left[M_i(\omega)\right]$ is estimated from k Gibbs samples
  - A counting oracle is used to compute the conditional probabilities in Gibbs sampling $P\left(A = x \middle| {\omega^{(t)}}_{-A}\right) \propto exp\left(\sum_f w_f N_f\left({\omega^{(t)}}_{-A}, A = x\right)\right)$

Performance

System	WebKB	Protein	ER	Smoker
Alchemy	X	X	X	1.21 (0.03)
Tuffy	-0.89 (0.05)	X	X	-0.69 (0.04)
CD	-0.64 (0.004)	-0.779 (0.0002)	-0.694 (0.001)	-0.7 (0.004)
VP	-0.91 (0.001)	-0.78 (0.001)	-0.693 (0.001)	-1.16 (0.1)
PLL	-0.72 (0.001)	-0.74 (0.0004)	-0.72 (0.001)	-1.61 (0.08)

Conditional Log Likelihood Scores along with standard deviation for 5-fold cross validation

Outline

- Introduction
  - Motivation
  - MLN Basics
- Scalable Inference
  - Domain-Lifted Inference
  - Approximate Domain-Lifting
- Scalable Weight Learning
  - Lifted Generative Weight Learning
  - Discriminative/pseudo-likelihood learning with Approximate Counting Oracles
- Applications
  - Overview of MLN Software
  - NLP Applications using MLNs
- Conclusion

An Overview of MLN Software

- Alchemy 1.0
  - Earliest implementation of MLN inference and learning algorithms
  - MCSAT, Gibbs Sampling, Belief Propagation, MaxWalkSAT,…
  - Contrastive Divergence, Voted Perceptron, …
- Pros
  - Several inference and learning algorithms along with various tunable options
- Cons
  - Lack of lifted inference algorithms (except Lifted BP)
  - Uses a pre-grounding approach and therefore the grounding process takes extremely long and/or runs out memory often
- Website
  - http://alchemy.cs.washington.edu/alchemy1.html

An Overview of MLN Software

- Alchemy 2.0
  - A suite of lifted inference algorithms built on top of Alchemy
  - PTP, Lifted Gibbs, Lifted Importance Sampling
- Pros
  - When the right symmetries are present, lifted inference algorithms are highly scalable
- Cons
  - Only uses exact symmetries. Therefore, not scalable for general MLNs
  - Often fails when arbitrary evidence is presented to the MLN since it breaks symmetries
- Website
  - https://code.google.com/archive/p/alchemy-2/

An Overview of MLN Software

- Tuffy/Hazy
  - Built on top of Postgres database
  - MCSAT, MaxWalkSAT implementations
- Pros
  - Efficient, specialized grounding technique using databases
- Cons
  - Not too many algorithm options
  - No lifted Inference, therefore scalability is limited
- Website
  - http://i.stanford.edu/hazy/tuffy/

Alchemy Lite

- Implements a subset of the MLN language in which inference is tractable called Tractable Markov Logic
- Pros
  - Tractable Inference
  - Interactive Mode
- Cons
  - May not be able to represent very complex knowledge
- Website
  - http://alchemy.cs.washington.edu/lite/

An Overview of MLN Software

- Markov the Beast
  - MAP inference
- Pros
  - Integrates cutting plane algorithms with MLN MAP inference
- Cons
  - No Lifted Inference
- Website
  - https://code.google.com/archive/p/thebeast/

An Overview of MLN Software

- Rockit
  - MAP Inference
- Pros
  - Integrates ILP solvers into MLN inference
  - Parallel implementation
- Cons
  - No Lifted Inference
- Website
  - https://code.google.com/archive/p/rockit/

An Overview of MLN Software

- Magician (beta version)
  - Approximate Inference and learning
  - Gibbs sampling and Contrastive Divergence
- Pros
  - Does not pre-ground the MLN
  - Uses an approximate counting oracle (Iterative Join Graph Propagation) within Gibbs sampling
  - Pre-processing to approximately lift the MLN using clustering
- Cons
  - Not many algorithm options
  - Still Under development!!
- Website
  - github.com/dvngp/CD-Learn

NLP Applications

- Text classification
- Fine-grained sentiment analysis
- Entity coreference resolution
- Event extraction

NLP Applications

- Text classification
- Fine-grained sentiment analysis
- Entity coreference resolution
- Event extraction

Text Classification

Task: Label the topic of a web-page

For almost three quarters it looked like this one was going to be different. Newton got a head start on a second straight NFL MVP award, a new touchdown…	Sports
The Oscars were a big success. Several celebrities gathered…	Entertainment

Typical assumption: each page has exactly one label

Typical Approach

- Train a (SVM) classifier to predict the topic of each document independently of other documents

Typical Approach

- Train a (SVM) classifier to predict the topic of each document independently of other documents

- Can we do better by exploiting the **relationship** between documents?
  - Rather than classify the documents independently, perform joint inference using MLNs

Designing an MLN: Predicates and Formulas

- Predicate Definitions
  - $Topic(page, topic!)$:The ! mark signifies mutual exclusion
  - $HasWord(word, page)$
  - $Linked(page, page)$:

Query predicate

Evidence predicates

Designing an MLN: Predicates and Formulas

- Predicate Definitions

  Query predicate

  - $Topic(page, topic!)$:The ! mark signifies mutual exclusion
  - $HasWord(word, page)$
  - $Linked(page, page)$:

  Evidence predicates

- Formulas

  Encode knowledge as hard/soft constraints

  - $HasWord(w, p) \Rightarrow Topic(p, t)$
  - However, the above formula is problematic because it learns a single weight for all possible words and topics.
  - We can specify that we need a different weight for each possible word-topic pair by augmenting variables with "+"
  - $HasWord(+w, p) \Rightarrow Topic(p, +t)$

Designing an MLN: Predicates and Formulas

- Predicate Definitions

  Query predicate

  - $Topic(page, topic!)$:The ! mark signifies mutual exclusion
  - $HasWord(word, page)$
  - $Linked(page, page)$:

  Evidence predicates

- Formulas

  Encode knowledge as hard/soft constraints

  - $HasWord(w, p) \Rightarrow Topic(p, t)$
  - However, the above formula is problematic because it learns a single weight for all possible words and topics.
  - We can specify that we need a different weight for each possible word-topic pair by augmenting variables with "+"
  - $HasWord(+w, p) \Rightarrow Topic(p, +t)$

Topic(p,+t) ^ Linked(p,p') => Topic(p',+t)

Possible Approximate Symmetries

- Some words are similar
  - Happy; Elated; Glad
- Related topics may also exhibit approximate symmetries
  - Technology; Science
- Clustering can help us exploit some of these approximate symmetries

NLP Applications

- Text classification
- Fine-grained sentiment analysis
- Entity coreference resolution
- Event extraction

Fine-Grained Sentiment Analysis (Zirn et al., 2011)

- Lots of work on document-level sentiment analysis
  - A binary/ternary polarity value is assigned to a document
- But coarse-grained sentiment analysis may be insufficient
  - "Despite the pretty design I would never recommend it, because the sound quality is unacceptable"

Fine-Grained Sentiment Analysis (Zirn et al., 2011)

- Lots of work on document-level sentiment analysis
  - A binary/ternary polarity value is assigned to a document
- But coarse-grained sentiment analysis may be insufficient
  - "Despite the pretty design I would never recommend it, because the sound quality is unacceptable"
  - Has both **positive** and **negative** polarity statements
  - Need fine grained sentiment analysis
    - Polarity determination at the level of discourse segments

Fine-Grained Sentiment Analysis (Zirn et al., 2011)

- Lots of work on document-level sentiment analysis
  - A binary/ternary polarity value is assigned to a document
- But coarse-grained sentiment analysis may be insufficient
  - "Despite the pretty design I would never recommend it, because the sound quality is unacceptable"
  - Has both **positive** and **negative** polarity statements
  - Need fine grained sentiment analysis
    - Polarity determination at the level of discourse segments

s1 = Despite the pretty design,
s2 = I would never recommend it
s3 = because the sound quality is unacceptable,

The set of constants correspond to discourse segments

A Simple Approach

- Train a (SVM) classifier to classify each discourse segment as positive or negative independently

A Simple Approach

- Train a (SVM) classifier to classify each discourse segment as positive or negative independently

- Can we do better by modeling the dependencies between discourse segments?

Query Predicates

- negative(x)
  - whether discourse segment x has negative polarity
- positive(x)
  - whether discourse segment x has positive polarity

Basic Formulas

$$\forall x : \neg positive(x) \Rightarrow negative(x)$$
$$\forall x : negative(x) \Rightarrow \neg positive(x)$$

Formulas encoding 3 types of knowledge

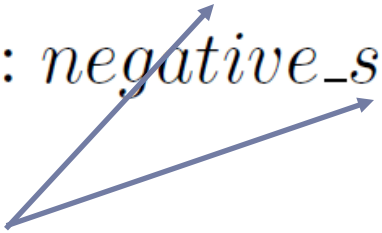
- Prior polarity
- Neighborhood polarity
- Discourse relations

Formulas encoding 3 types of knowledge

- Prior polarity
- Neighborhood polarity
- Discourse relations

Modeling Prior Polarity

$$\forall x : positive\_source_\ell(x) \Leftrightarrow positive(x)$$
$$\forall x : negative\_source_\ell(x) \Leftrightarrow negative(x)$$



Prior Features

- Three sources of prior polarity
  - SentiWordNet: positivity scores and negativity scores
  - Turney's adjective list
  - Unigram lexicon

Formulas encoding 3 types of knowledge

- Prior polarity
- Neighborhood polarity
- Discourse relations

Modeling Neighborhood

Precedence relationships with previous segment

$$\forall x, y : pre(x, y) \wedge positive(x) \Rightarrow positive(y)$$

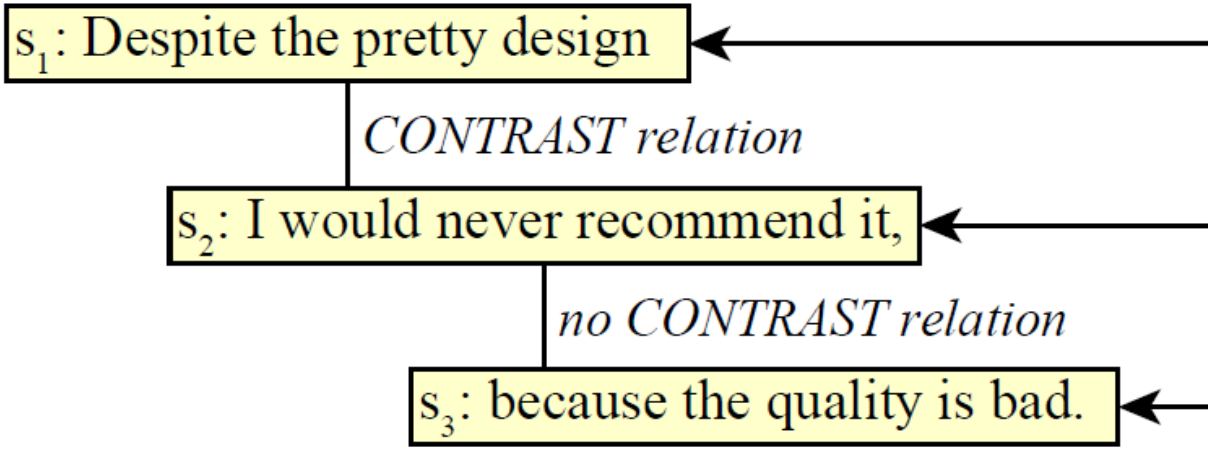
$$\forall x, y : pre(x, y) \wedge negative(x) \Rightarrow negative(y)$$

Formulas encoding 3 types of knowledge

- Prior polarity
- Neighborhood polarity
- Discourse relations

Modeling Discourse Relations

Models two types of discourse relations across segments: CONTRAST and NO-CONTRAST



Formulas Exploiting Discourse Relations

$$\forall x, y : contrast(x, y) \ \wedge \ positive(x) \quad \Rightarrow negative(y)$$
$$\forall x, y : contrast(x, y) \ \wedge \ negative(x) \quad \Rightarrow \ positive(y)$$
$$\forall x, y : ncontrast(x, y) \wedge positive(x) \quad \Rightarrow \ positive(y)$$
$$\forall x, y : ncontrast(x, y) \wedge negative(x) \quad \Rightarrow negative(y)$$

Evaluation

- Dataset: Multi-domain sentient dataset (Blitzer et al., 2007)

Topic	p	n	total
Cell Phones & Service	1392	1785	3177
Gourmet Food	990	616	1606
Kitchen & Housewares	1188	1405	2593
Sum	3570	3806	7376

- Weight learning
  - Ran voted perceptron for 20 epochs

Results

	positive			negative			
	P	R	F	P	R	F	A
majority baseline	0.00	0.00	0.00	51.60	100.00	68.07	51.60
SVM	57.05	43.06	49.08	56.44	69.47	62.28	56.66
MLN_polarity	53.21	69.58	60.31	59.90	42.62	49.80	55.67
MLN_neighborhood	66.38	72.94	**69.50**	72.02	65.34	**68.52**	**69.02**
MLN_contrast	61.39	73.47	66.89	69.48	56.65	62.41	64.79

NLP Applications

- Text classification
- Fine-grained sentiment analysis
- Entity coreference resolution
- Event extraction

Entity Coreference Resolution (Song et al., 2012)

Identify all noun phrases that refer to the same entity

> Queen Elizabeth set about transforming her husband, King George VI, into a viable monarch. Logue, a renowned speech therapist, was summoned to help the King overcome his speech impediment...

Entity Coreference Resolution (Song et al., 2012)

Identify all noun phrases that refer to the same entity

Queen Elizabeth set about transforming her husband, King George VI, into a viable monarch. Logue, a renowned speech therapist, was summoned to help the King overcome his speech impediment...

Entity Coreference Resolution (Song et al., 2012)

Identify all noun phrases that refer to the same entity

Queen Elizabeth set about transforming her husband, King George VI, into a viable monarch. Logue, a renowned speech therapist, was summoned to help the King overcome his speech impediment...

Entity Coreference Resolution (Song et al., 2012)

Identify all noun phrases that refer to the same entity

> Queen Elizabeth set about transforming her husband, King George VI, into a viable monarch. Logue, a renowned speech therapist, was summoned to help the King overcome his speech impediment...

Entity Coreference Resolution (Song et al., 2012)

Identify all noun phrases that refer to the same entity

> Queen Elizabeth set about transforming her husband, King George VI, into a viable monarch. Logue, a renowned speech therapist, was summoned to help the King overcome his speech impediment...

Entity Coreference Resolution (Song et al., 2012)

Identify all noun phrases that refer to the same entity

Queen Elizabeth set about transforming her husband, King George VI, into a viable monarch. Logue, a renowned speech therapist, was summoned to help the King overcome his speech impediment...

Three main tasks in the pipeline

- **Mention Detection**
  - Extracts the mentions (pronouns, names, nominals)
- **Pairwise classification**
  - classify each pair of mentions as coreferent or not coreferent
- **Mention clustering**
  - resolves conflicting classification decisions and generates the coreference clusters

- Last two tasks are typically addressed in a **pipeline** fashion
  - Error propagation
  - Can we perform joint inference over these two tasks?

Query predicate

- Coref(x,y)
  - whether mention x and mention y are coreferent

MLN observed predicates

describing the attributes of m_i	
mentionType(i,t)	m_i has mention type NAM(named entities), NOM(nominal) or PRO(pronouns).
entityType(i,e)	m_i has entity type PERSON, ORG, GPE or UN...
genderType(i,g)	m_i has gender type MALE, FEMALE, NEUTRAL or UN.
numberType(i,n)	m_i has number type SINGULAR, PLURAL or UN.
hasHead(i,h)	m_i has head word h, here h can represent all possible head words.
firstMention(i)	m_i is the first mention in its sentence.
reflexive(i)	m_i is reflexive.
possessive(i)	m_i is possessive.
definite(i)	m_i is definite noun phrase.
indefinite(i)	m_i is indefinite noun phrase.
demonstrative(i)	m_i is demonstrative.

MLN observed predicates

describing the attributes of relations between m_j and m_i	
mentionDistance(j,i,m)	Distance between m_j and m_i in mentions.
sentenceDistance(j,i,s)	Distance between m_j and m_i in sentences.
bothMatch(j,i,b)	Gender and number of both m_j and m_i match: AGREE_YES, AGREE_NO and AGREE_UN).
closestMatch(j,i,c)	m_j is the first agreement in number and gender when looking backward from m_i: CAGREE_YES, CAGREE_NO and CAGREE_UN.
exactStrMatch(j,i)	Exact strings match between m_j and m_i.
pronounStrMatch(j,i)	Both are pronouns and their strings match.
nopronounStrMatch(j,i)	Both are not pronouns and their strings match.
properStrMatch(j,i)	Both are proper names and their strings match.
headMatch(j,i)	Head word strings match between m_j and m_i.
subStrMatch(j,i)	Sub-word strings match between m_j and m_i.
animacyMatch(j,i)	Animacy types match between m_j and m_i.
nested(j,i)	$m_{j/i}$ is included in $m_{i/j}$.
c_command(j,i)	$m_{j/i}$ C-Commands $m_{i/j}$.
sameSpeaker(j,i)	m_j and m_i have the same speaker.
entityTypeMatch(j,i)	Entity types match between m_j and m_i.
alias(j,i)	$m_{j/i}$ is an alias of $m_{i/j}$.
srlMatch(j,i)	m_j and m_i have the same semantic role.
verbMatch(j,i)	m_j and m_i have semantic role for the same verb.

Local Formulas

Lexical Features

$\text{mentionType}(j,t_1+) \wedge \text{mentionType}(i,t_2+) \wedge \text{exactStrMatch}(j,i) \wedge j \neq i \Rightarrow \text{coref}(j,i)$
$\text{mentionType}(j,t_1+) \wedge \text{mentionType}(i,t_2+) \wedge \text{pronounStrMatch}(j,i) \wedge j \neq i \Rightarrow \text{coref}(j,i)$
$\text{mentionType}(j,t_1+) \wedge \text{mentionType}(i,t_2+) \wedge \text{properStrMatch}(j,i) \wedge j \neq i \Rightarrow \text{coref}(j,i)$
$\text{mentionType}(j,t_1+) \wedge \text{mentionType}(i,t_2+) \wedge \text{nopronounStrMatch}(j,i) \wedge j \neq i \Rightarrow \text{coref}(j,i)$
$\text{mentionType}(j,t_1+) \wedge \text{mentionType}(i,t_2+) \wedge \text{headMatch}(j,i) \wedge j \neq i \Rightarrow \text{coref}(j,i)$
$\text{mentionType}(j,t_1+) \wedge \text{mentionType}(i,t_2+) \wedge \text{subStrMatch}(j,i) \wedge j \neq i \Rightarrow \text{coref}(j,i)$
$\text{hasHead}(j,h_1+) \wedge \text{hasHead}(i,h_2+) \wedge j \neq i \Rightarrow \text{coref}(j,i)$

Semantic Features

$\text{mentionType}(j,t_1+) \wedge \text{mentionType}(i,t_2+) \wedge \text{alias}(j,i) \wedge j \neq i \Rightarrow \text{coref}(j,i)$
$\text{mentionType}(j,t_1+) \wedge \text{mentionType}(i,t_2+) \wedge \text{sameSpeaker}(j,i) \wedge j \neq i \Rightarrow \text{coref}(j,i)$
$\text{mentionType}(j,t_1+) \wedge \text{mentionType}(i,t_2+) \wedge \text{entityTypeMatch}(j,i) \wedge j \neq i \Rightarrow \text{coref}(j,i)$
$\text{mentionType}(j,t_1+) \wedge \text{mentionType}(i,t_2+) \wedge \text{srlMatch}(j,i) \wedge j \neq i \Rightarrow \text{coref}(j,i)$
$\text{mentionType}(j,t_1+) \wedge \text{mentionType}(i,t_2+) \wedge \text{verbMatch}(j,i) \wedge j \neq i \Rightarrow \text{coref}(j,i)$
$(\text{entityType}(j,e_1+) \vee \text{entityType}(i,e_2+)) \wedge j \neq i \Rightarrow \text{coref}(j,i)$

Local Formulas

Lexical Features

$\text{mentionType}(j,t_1+) \wedge \text{mentionType}(i,t_2+) \wedge \text{exactStrMatch}(j,i) \wedge j \neq i \Rightarrow \text{coref}(j,i)$

$\text{mentionType}(j,t_1+) \wedge \text{mentionType}(i,t_2+) \wedge \text{pronounStrMatch}(j,i) \wedge j \neq i \Rightarrow \text{coref}(j,i)$

$\text{mentionType}(j,t_1+) \wedge \text{mentionType}(i,t_2+) \wedge \text{properStrMatch}(j,i) \wedge j \neq i \Rightarrow \text{coref}(j,i)$

$\text{mentionType}(j,t_1+) \wedge \text{mentionType}(i,t_2+) \wedge \text{nopronounStrMatch}(j,i) \wedge j \neq i \Rightarrow \text{coref}(j,i)$

$\text{mentionType}(j,t_1+) \wedge \text{mentionType}(i,t_2+) \wedge \text{headMatch}(j,i) \wedge j \neq i \Rightarrow \text{coref}(j,i)$

$\text{mentionType}(j,t_1+) \wedge \text{mentionType}(i,t_2+) \wedge \text{subStrMatch}(j,i) \wedge j \neq i \Rightarrow \text{coref}(j,i)$

$\text{hasHead}(j,h_1+) \wedge \text{hasHead}(i,h_2+) \wedge j \neq i \Rightarrow \text{coref}(j,i)$

Semantic Features

$\text{mentionType}(j,t_1+) \wedge \text{mentionType}(i,t_2+) \wedge \text{alias}(j,i) \wedge j \neq i \Rightarrow \text{coref}(j,i)$

$\text{mentionType}(j,t_1+) \wedge \text{mentionType}(i,t_2+) \wedge \text{sameSpeaker}(j,i) \wedge j \neq i \Rightarrow \text{coref}(j,i)$

$\text{mentionType}(j,t_1+) \wedge \text{mentionType}(i,t_2+) \wedge \text{entityTypeMatch}(j,i) \wedge j \neq i \Rightarrow \text{coref}(j,i)$

$\text{mentionType}(j,t_1+) \wedge \text{mentionType}(i,t_2+) \wedge \text{srlMatch}(j,i) \wedge j \neq i \Rightarrow \text{coref}(j,i)$

$\text{mentionType}(j,t_1+) \wedge \text{mentionType}(i,t_2+) \wedge \text{verbMatch}(j,i) \wedge j \neq i \Rightarrow \text{coref}(j,i)$

$(\text{entityType}(j,e_1+) \vee \text{entityType}(i,e_2+)) \wedge j \neq i \Rightarrow \text{coref}(j,i)$

Local Formulas

Lexical Features

$\text{mentionType}(j,t_1+) \wedge \text{mentionType}(i,t_2+) \wedge \text{exactStrMatch}(j,i) \wedge j \neq i \Rightarrow \text{coref}(j,i)$

$\text{mentionType}(j,t_1+) \wedge \text{mentionType}(i,t_2+) \wedge \text{pronounStrMatch}(j,i) \wedge j \neq i \Rightarrow \text{coref}(j,i)$

$\text{mentionType}(j,t_1+) \wedge \text{mentionType}(i,t_2+) \wedge \text{properStrMatch}(j,i) \wedge j \neq i \Rightarrow \text{coref}(j,i)$

$\text{mentionType}(j,t_1+) \wedge \text{mentionType}(i,t_2+) \wedge \text{nopronounStrMatch}(j,i) \wedge j \neq i \Rightarrow \text{coref}(j,i)$

$\text{mentionType}(j,t_1+) \wedge \text{mentionType}(i,t_2+) \wedge \text{headMatch}(j,i) \wedge j \neq i \Rightarrow \text{coref}(j,i)$

$\text{mentionType}(j,t_1+) \wedge \text{mentionType}(i,t_2+) \wedge \text{subStrMatch}(j,i) \wedge j \neq i \Rightarrow \text{coref}(j,i)$

$\text{hasHead}(j,h_1+) \wedge \text{hasHead}(i,h_2+) \wedge j \neq i \Rightarrow \text{coref}(j,i)$

Semantic Features

$\text{mentionType}(j,t_1+) \wedge \text{mentionType}(i,t_2+) \wedge \text{alias}(j,i) \wedge j \neq i \Rightarrow \text{coref}(j,i)$

$\text{mentionType}(j,t_1+) \wedge \text{mentionType}(i,t_2+) \wedge \text{sameSpeaker}(j,i) \wedge j \neq i \Rightarrow \text{coref}(j,i)$

$\text{mentionType}(j,t_1+) \wedge \text{mentionType}(i,t_2+) \wedge \text{entityTypeMatch}(j,i) \wedge j \neq i \Rightarrow \text{coref}(j,i)$

$\text{mentionType}(j,t_1+) \wedge \text{mentionType}(i,t_2+) \wedge \text{srlMatch}(j,i) \wedge j \neq i \Rightarrow \text{coref}(j,i)$

$\text{mentionType}(j,t_1+) \wedge \text{mentionType}(i,t_2+) \wedge \text{verbMatch}(j,i) \wedge j \neq i \Rightarrow \text{coref}(j,i)$

$(\text{entityType}(j,e_1+) \vee \text{entityType}(i,e_2+)) \wedge j \neq i \Rightarrow \text{coref}(j,i)$

Local Formulas

Lexical Features

$\text{mentionType}(j,t_1+) \wedge \text{mentionType}(i,t_2+) \wedge \text{exactStrMatch}(j,i) \wedge j \neq i \Rightarrow \text{coref}(j,i)$
$\text{mentionType}(j,t_1+) \wedge \text{mentionType}(i,t_2+) \wedge \text{pronounStrMatch}(j,i) \wedge j \neq i \Rightarrow \text{coref}(j,i)$
$\text{mentionType}(j,t_1+) \wedge \text{mentionType}(i,t_2+) \wedge \text{properStrMatch}(j,i) \wedge j \neq i \Rightarrow \text{coref}(j,i)$
$\text{mentionType}(j,t_1+) \wedge \text{mentionType}(i,t_2+) \wedge \text{nopronounStrMatch}(j,i) \wedge j \neq i \Rightarrow \text{coref}(j,i)$
$\text{mentionType}(j,t_1+) \wedge \text{mentionType}(i,t_2+) \wedge \text{headMatch}(j,i) \wedge j \neq i \Rightarrow \text{coref}(j,i)$
$\text{mentionType}(j,t_1+) \wedge \text{mentionType}(i,t_2+) \wedge \text{subStrMatch}(j,i) \wedge j \neq i \Rightarrow \text{coref}(j,i)$
$\text{hasHead}(j,h_1+) \wedge \text{hasHead}(i,h_2+) \wedge j \neq i \Rightarrow \text{coref}(j,i)$

Semantic Features

$\text{mentionType}(j,t_1+) \wedge \text{mentionType}(i,t_2+) \wedge \text{alias}(j,i) \wedge j \neq i \Rightarrow \text{coref}(j,i)$
$\text{mentionType}(j,t_1+) \wedge \text{mentionType}(i,t_2+) \wedge \text{sameSpeaker}(j,i) \wedge j \neq i \Rightarrow \text{coref}(j,i)$
$\text{mentionType}(j,t_1+) \wedge \text{mentionType}(i,t_2+) \wedge \text{entityTypeMatch}(j,i) \wedge j \neq i \Rightarrow \text{coref}(j,i)$
$\text{mentionType}(j,t_1+) \wedge \text{mentionType}(i,t_2+) \wedge \text{srlMatch}(j,i) \wedge j \neq i \Rightarrow \text{coref}(j,i)$
$\text{mentionType}(j,t_1+) \wedge \text{mentionType}(i,t_2+) \wedge \text{verbMatch}(j,i) \wedge j \neq i \Rightarrow \text{coref}(j,i)$
$(\text{entityType}(j,e_1+) \vee \text{entityType}(i,e_2+)) \wedge j \neq i \Rightarrow \text{coref}(j,i)$

Local Formulas

Lexical Features

$\text{mentionType}(j,t_1+) \wedge \text{mentionType}(i,t_2+) \wedge \text{exactStrMatch}(j,i) \wedge j \neq i \Rightarrow \text{coref}(j,i)$
$\text{mentionType}(j,t_1+) \wedge \text{mentionType}(i,t_2+) \wedge \text{pronounStrMatch}(j,i) \wedge j \neq i \Rightarrow \text{coref}(j,i)$
$\text{mentionType}(j,t_1+) \wedge \text{mentionType}(i,t_2+) \wedge \text{properStrMatch}(j,i) \wedge j \neq i \Rightarrow \text{coref}(j,i)$
$\text{mentionType}(j,t_1+) \wedge \text{mentionType}(i,t_2+) \wedge \text{nopronounStrMatch}(j,i) \wedge j \neq i \Rightarrow \text{coref}(j,i)$
$\text{mentionType}(j,t_1+) \wedge \text{mentionType}(i,t_2+) \wedge \text{headMatch}(j,i) \wedge j \neq i \Rightarrow \text{coref}(j,i)$
$\text{mentionType}(j,t_1+) \wedge \text{mentionType}(i,t_2+) \wedge \text{subStrMatch}(j,i) \wedge j \neq i \Rightarrow \text{coref}(j,i)$
$\text{hasHead}(j,h_1+) \wedge \text{hasHead}(i,h_2+) \wedge j \neq i \Rightarrow \text{coref}(j,i)$

Semantic Features

$\text{mentionType}(j,t_1+) \wedge \text{mentionType}(i,t_2+) \wedge \text{alias}(j,i) \wedge j \neq i \Rightarrow \text{coref}(j,i)$
$\text{mentionType}(j,t_1+) \wedge \text{mentionType}(i,t_2+) \wedge \text{sameSpeaker}(j,i) \wedge j \neq i \Rightarrow \text{coref}(j,i)$
$\text{mentionType}(j,t_1+) \wedge \text{mentionType}(i,t_2+) \wedge \text{entityTypeMatch}(j,i) \wedge j \neq i \Rightarrow \text{coref}(j,i)$
$\text{mentionType}(j,t_1+) \wedge \text{mentionType}(i,t_2+) \wedge \text{srlMatch}(j,i) \wedge j \neq i \Rightarrow \text{coref}(j,i)$
$\text{mentionType}(j,t_1+) \wedge \text{mentionType}(i,t_2+) \wedge \text{verbMatch}(j,i) \wedge j \neq i \Rightarrow \text{coref}(j,i)$
$(\text{entityType}(j,e_1+) \vee \text{entityType}(i,e_2+)) \wedge j \neq i \Rightarrow \text{coref}(j,i)$

Local Formulas

Grammatical Features

$\text{mentionType}(j,t_1+) \wedge \text{mentionType}(i,t_2+) \wedge \text{genderType}(j,g_1+) \wedge \text{genderType}(i,g_2+) \wedge j \neq i \Rightarrow \text{coref}(j,i)$
$\text{mentionType}(j,t_1+) \wedge \text{mentionType}(i,t_2+) \wedge \text{numberType}(j,n_1+) \wedge \text{numberType}(i,n_2+) \wedge j \neq i \Rightarrow \text{coref}(j,i)$
$\text{mentionType}(j,t_1+) \wedge \text{mentionType}(i,t_2+) \wedge \text{bothMatch}(j,i,b+) \wedge j \neq i \Rightarrow \text{coref}(j,i)$
$\text{mentionType}(j,t_1+) \wedge \text{mentionType}(i,t_2+) \wedge \text{closestMatch}(j,i,c+) \wedge j \neq i \Rightarrow \text{coref}(j,i)$
$\text{mentionType}(j,t_1+) \wedge \text{mentionType}(i,t_2+) \wedge \text{animacyMatch}(j,i) \wedge j \neq i \Rightarrow \text{coref}(j,i)$
$\text{mentionType}(j,t_1+) \wedge \text{mentionType}(i,t_2+) \wedge \text{nested}(j,i) \wedge j \neq i \Rightarrow \text{coref}(j,i)$
$\text{mentionType}(j,t_1+) \wedge \text{mentionType}(i,t_2+) \wedge \text{c\_command}(j,i) \wedge j \neq i \Rightarrow \text{coref}(j,i)$
$(\text{mentionType}(j,t_1+) \vee \text{mentionType}(i,t_2+)) \wedge j \neq i \Rightarrow \text{coref}(j,i)$
$(\text{reflexive}(j) \vee \text{reflexive}(i)) \wedge j \neq i \Rightarrow \text{coref}(j,i)$
$(\text{possessive}(j) \vee \text{possessive}(i)) \wedge j \neq i \Rightarrow \text{coref}(j,i)$
$(\text{definite}(j) \vee \text{definite}(i)) \wedge j \neq i \Rightarrow \text{coref}(j,i)$
$(\text{indefinite}(j) \vee \text{indefinite}(i)) \wedge j \neq i \Rightarrow \text{coref}(j,i)$
$(\text{demonstrative}(j) \vee \text{demonstrative}(i)) \wedge j \neq i \Rightarrow \text{coref}(j,i)$

Distance and position Features

$\text{mentionType}(j,t_1+) \wedge \text{mentionType}(i,t_2+) \wedge \text{sentenceDistance}(j,i,s+) \wedge j \neq i \Rightarrow \text{coref}(j,i)$
$\text{mentionType}(j,t_1+) \wedge \text{mentionType}(i,t_2+) \wedge \text{mentionDistance}(j,i,m+) \wedge j \neq i \Rightarrow \text{coref}(j,i)$
$(\text{firstMention}(j) \vee \text{firstMention}(i)) \wedge j \neq i \Rightarrow \text{coref}(j,i)$

Local Formulas

Grammatical Features

mentionType(j,t_1+) ∧ mentionType(i,t_2+) ∧ genderType(j,g_1+) ∧ genderType(i,g_2+) ∧ j≠ i ⇒ coref(j,i)
mentionType(j,t_1+) ∧ mentionType(i,t_2+) ∧ numberType(j,n_1+) ∧ numberType(i,n_2+) ∧ j≠ i ⇒ coref(j,i)
mentionType(j,t_1+) ∧ mentionType(i,t_2+) ∧ bothMatch(j,i,b+) ∧ j≠ i ⇒ coref(j,i)
mentionType(j,t_1+) ∧ mentionType(i,t_2+) ∧ closestMatch(j,i,c+) ∧ j≠ i ⇒ coref(j,i)
mentionType(j,t_1+) ∧ mentionType(i,t_2+) ∧ animacyMatch(j,i) ∧ j≠ i ⇒ coref(j,i)
mentionType(j,t_1+) ∧ mentionType(i,t_2+) ∧ nested(j,i) ∧ j≠ i ⇒ coref(j,i)
mentionType(j,t_1+) ∧ mentionType(i,t_2+) ∧ c_command(j,i) ∧ j≠ i ⇒ coref(j,i)
(mentionType(j,t_1+) ∨ mentionType(i,t_2+)) ∧ j≠ i ⇒ coref(j,i)
(reflexive(j) ∨ reflexive(i)) ∧ j≠ i ⇒ coref(j,i)
(possessive(j) ∨ possessive(i)) ∧ j≠ i ⇒ coref(j,i)
(definite(j) ∨ definite(i)) ∧ j≠ i ⇒ coref(j,i)
(indefinite(j) ∨ indefinite(i)) ∧ j≠ i ⇒ coref(j,i)
(demonstrative(j) ∨ demonstrative(i)) ∧ j≠ i ⇒ coref(j,i)

Distance and position Features

mentionType(j,t_1+) ∧ mentionType(i,t_2+) ∧ sentenceDistance(j,i,s+) ∧ j≠ i ⇒ coref(j,i)
mentionType(j,t_1+) ∧ mentionType(i,t_2+) ∧ mentionDistance (j,i,m+) ∧ j≠ i ⇒ coref(j,i)
(firstMention(j) ∨ firstMention(i)) ∧ j≠ i ⇒ coref(j,i)

Local Formulas

Grammatical Features

$\text{mentionType}(j,t_1+) \wedge \text{mentionType}(i,t_2+) \wedge \text{genderType}(j,g_1+) \wedge \text{genderType}(i,g_2+) \wedge j \neq i \Rightarrow \text{coref}(j,i)$
$\text{mentionType}(j,t_1+) \wedge \text{mentionType}(i,t_2+) \wedge \text{numberType}(j,n_1+) \wedge \text{numberType}(i,n_2+) \wedge j \neq i \Rightarrow \text{coref}(j,i)$
$\text{mentionType}(j,t_1+) \wedge \text{mentionType}(i,t_2+) \wedge \text{bothMatch}(j,i,b+) \wedge j \neq i \Rightarrow \text{coref}(j,i)$
$\text{mentionType}(j,t_1+) \wedge \text{mentionType}(i,t_2+) \wedge \text{closestMatch}(j,i,c+) \wedge j \neq i \Rightarrow \text{coref}(j,i)$
$\text{mentionType}(j,t_1+) \wedge \text{mentionType}(i,t_2+) \wedge \text{animacyMatch}(j,i) \wedge j \neq i \Rightarrow \text{coref}(j,i)$
$\text{mentionType}(j,t_1+) \wedge \text{mentionType}(i,t_2+) \wedge \text{nested}(j,i) \wedge j \neq i \Rightarrow \text{coref}(j,i)$
$\text{mentionType}(j,t_1+) \wedge \text{mentionType}(i,t_2+) \wedge \text{c\_command}(j,i) \wedge j \neq i \Rightarrow \text{coref}(j,i)$
$(\text{mentionType}(j,t_1+) \vee \text{mentionType}(i,t_2+)) \wedge j \neq i \Rightarrow \text{coref}(j,i)$
$(\text{reflexive}(j) \vee \text{reflexive}(i)) \wedge j \neq i \Rightarrow \text{coref}(j,i)$
$(\text{possessive}(j) \vee \text{possessive}(i)) \wedge j \neq i \Rightarrow \text{coref}(j,i)$
$(\text{definite}(j) \vee \text{definite}(i)) \wedge j \neq i \Rightarrow \text{coref}(j,i)$
$(\text{indefinite}(j) \vee \text{indefinite}(i)) \wedge j \neq i \Rightarrow \text{coref}(j,i)$
$(\text{demonstrative}(j) \vee \text{demonstrative}(i)) \wedge j \neq i \Rightarrow \text{coref}(j,i)$

Distance and position Features

$\text{mentionType}(j,t_1+) \wedge \text{mentionType}(i,t_2+) \wedge \text{sentenceDistance}(j,i,s+) \wedge j \neq i \Rightarrow \text{coref}(j,i)$
$\text{mentionType}(j,t_1+) \wedge \text{mentionType}(i,t_2+) \wedge \text{mentionDistance}(j,i,m+) \wedge j \neq i \Rightarrow \text{coref}(j,i)$
$(\text{firstMention}(j) \vee \text{firstMention}(i)) \wedge j \neq i \Rightarrow \text{coref}(j,i)$

Global Formulas

Transitivity constraint:

$$coref(j,k) \wedge coref(k,i) \wedge j < k < i \Rightarrow coref(j,i) \quad (3)$$

$$coref(j,k) \wedge coref(j,i) \wedge j < k < i \Rightarrow coref(k,i)$$

$$coref(j,i) \wedge coref(k,i) \wedge j < k < i \Rightarrow coref(j,k)$$

Performance on CoNLL-11 shared task

- Online parameter learning using MIRA

System	Mention Detection			MUC			B-cube			CEAF			Avg
	R	P	F	R	P	F	R	P	F	R	P	F	F
MLN-Local	62.52	74.75	68.09	56.07	65.55	60.44	65.67	72.95	69.12	45.55	37.19	40.95	56.84
MLN-Local+Trans	**68.49**	70.32	69.40	**57.16**	60.98	59.01	**66.97**	72.90	69.81	46.96	**43.34**	**45.08**	57.97
MLN-Joint(Trans)	64.46	**75.37**	**69.49**	55.48	**67.15**	**60.76**	64.00	**78.11**	**70.36**	**50.63**	39.84	44.60	**58.57**

Table 3: Comparison between different MLN-based systems, using 10-fold cross validation on the training dataset.

Performance on CoNLL-11 shared task

System	Mention Detection			MUC			B-cube			CEAF			Avg
	R	P	F	R	P	F	R	P	F	R	P	F	F
MLN-Joint(Trans)	67.28	72.88	69.97	58.00	64.10	**60.90**	67.12	74.13	**70.45**	47.70	41.96	44.65	**58.67**
MaxEnt+Trans	61.36	76.11	67.94	51.46	68.40	58.73	59.79	**81.69**	69.04	**53.03**	37.84	44.17	57.31
Lee's System	-	-	-	57.50	59.10	58.30	**71.00**	69.20	70.10	48.10	**46.50**	**47.30**	58.60
Sapena's System	**92.45**	27.34	42.20	54.53	62.25	58.13	63.72	73.83	68.40	47.20	40.01	43.31	56.61
Chang's System	-	-	64.69	-	-	55.8	-	-	69.29	-	-	43.96	56.35

Table 4: Comparisons with state-of-the-art systems on the development dataset.

NLP Applications

- Text classification
- Fine-grained sentiment analysis
- Entity coreference resolution
- Event extraction

Event Extraction (Venugopal et al., 2014)

- Event extraction is the task of extracting and labeling all instances in a text document that correspond to pre-defined event types
  - BioNLP Genia extraction task concerns the extraction of instances of bio-molecular event types (Kim et al., 2009)

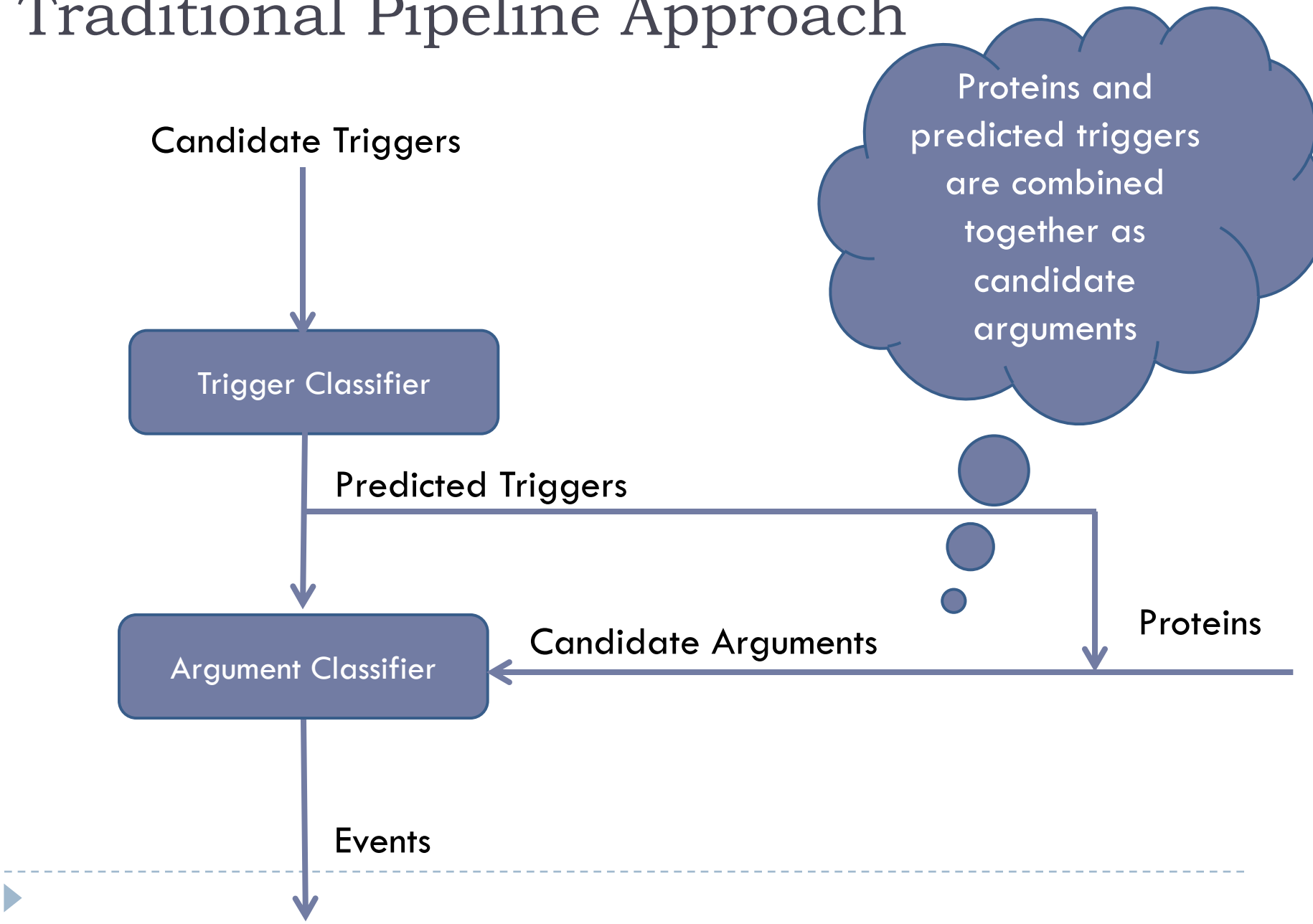
Task Definition (BioNLP'13 Genia)

.... demonstrated that HOIL-1L interacting protein (HOIP), a ubiquitin ligase that can catalyze the assembly of linear polyubiquitin chains, is recruited to DC40 in a TRAF2-dependent manner following engagement of CD40 ...

ID	Event Type	Trigger	Arguments
E11	Binding	recruited	Theme={HOIL-1L interacting protein,CD40}
E12	Regulation	dependent	Theme=E11, Cause=TRAF2
E13	Regulation	following	Theme=E12, Cause=E14
E14	Binding	engagement	Theme=CD40

Recursive events that have other events as arguments

Traditional Pipeline Approach



Joint Inference for Event Extraction

- Pipeline models are inadequate
  - Error propagation
  - Fail to consider relational dependencies
- Joint models are essential to exploit the inter-dependencies
  - Our system uses Markov Logic Networks (MLN) to specify the joint model
- Challenging to apply joint inference effectively using MLNs
  - Event extraction requires encoding linguistic features that are extremely high dimensional in nature
  - High dimensional features require infeasible amount of data for robust MLN weight learning

Joint Inference for Event Extraction

- Our approach = MLN + SVM
- Best of both worlds: SVMs handle high dimensional, sparse data better than MLNs while MLNs handle relational dependencies better than SVMs.
- Learn weights of high dimensional features using SVM and embed them as low dimensional formulas in MLN
  - Introduce SVM output as prior knowledge (soft evidence) in the joint MLN

Results

- Best performing system: BioNLP'11 and BioNLP'13
- On par with the best performing system on BioNLP'09
  - Significantly better than previous MLN-based joint inference systems developed for this task

System	F1
Our System	**53.61**
EVEX	50.97
TEES-2.1	50.74
BIOSEM	50.68
NCBI	48.93
DLUTNLP	47.56

Bio'13

System	F1
Our System	**58.07**
Miwa12	57.98
Riedel11	56
UTurku	53.3
MSR-NLP	51.50

Bio'11

System	F1
Miwa12	58.27
Our System	**58.16**
Riedel11	57.4
Miwa10	56.28
Bjorne	51.95
PoonMLN	50.0
RiedelMLN	44.4

Bio'09

Other Applications

- Temporal relation extraction (Yoshikawa et al., 2009)
- Situated natural language understanding (Kennington & Schlangen 2012)
- Texual entailment and semantic similarity (Beltagy et al., 2013)
- Question answering (Khot et al., 2015)
- Event coreference resolution (Lu et al., 2016)
- Discourse mode identification (Venugopal & Rus, 2016)

Conclusion

- MLNs show a lot of potential for several applications
  - Compact representation for complex domain knowledge and large relational data
  - Ability to handle uncertainty
- Bottleneck: Lack of scalable inference and learning algorithms
  - To scale up to large problems, we need approximations that trade-off scalability with guarantees
  - Approximate lifting and learning seem to be the most promising directions to achieve desired scalability
  - Easy-to-use software and tools will make MLNs more applicable to a wider research community