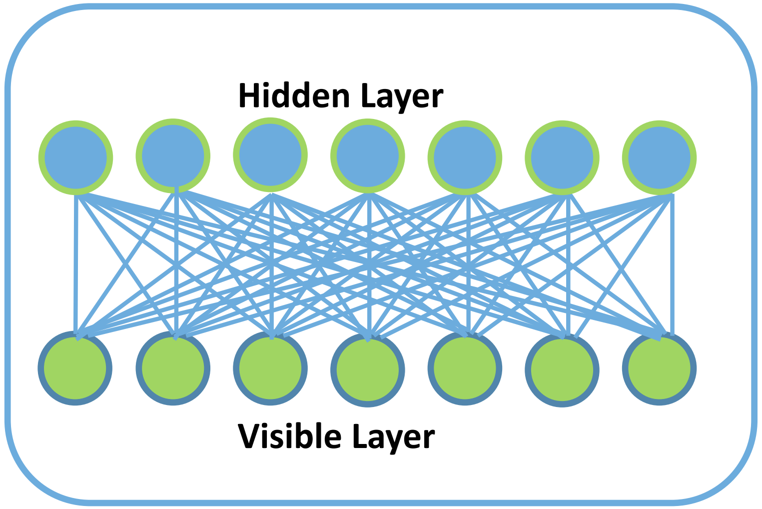
Relational Restricted Boltzmann Machines: A Probabilistic Logic Learning Approach

Navdeep Kaur, Gautam Kunapuli, Tushar Khot,
Kristian Kersting, William Cohen and Sriraam Natarajan

Currently at University of Texas Dallas

Motivation

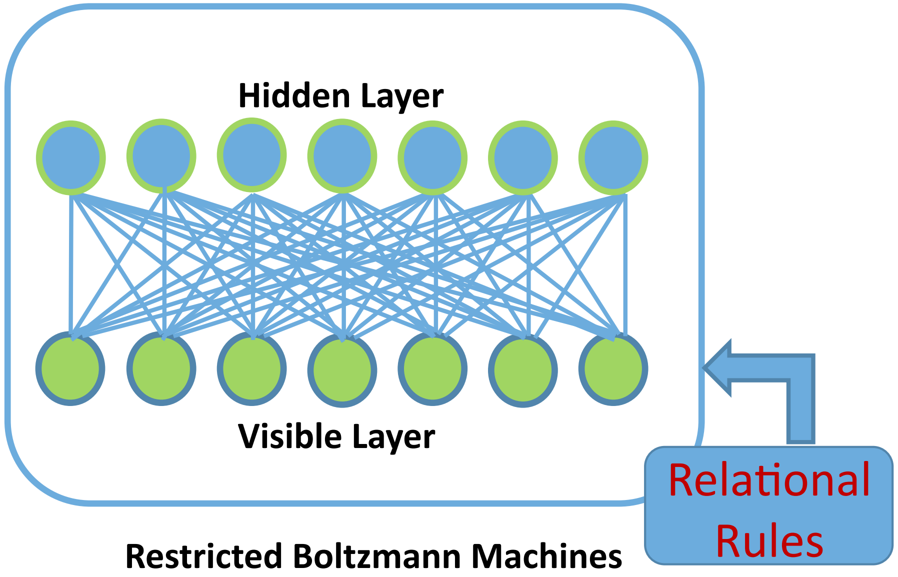


Restricted Boltzmann Machines

Restricted Boltzmann Machines (RBM) are:

- Powerful models with diverse applications
- Limited to *flat feature vectors*

Motivation



Restricted Boltzmann Machines (RBM) are

- Powerful models with diverse applications
- Limited to *flat feature vectors*

Goal:

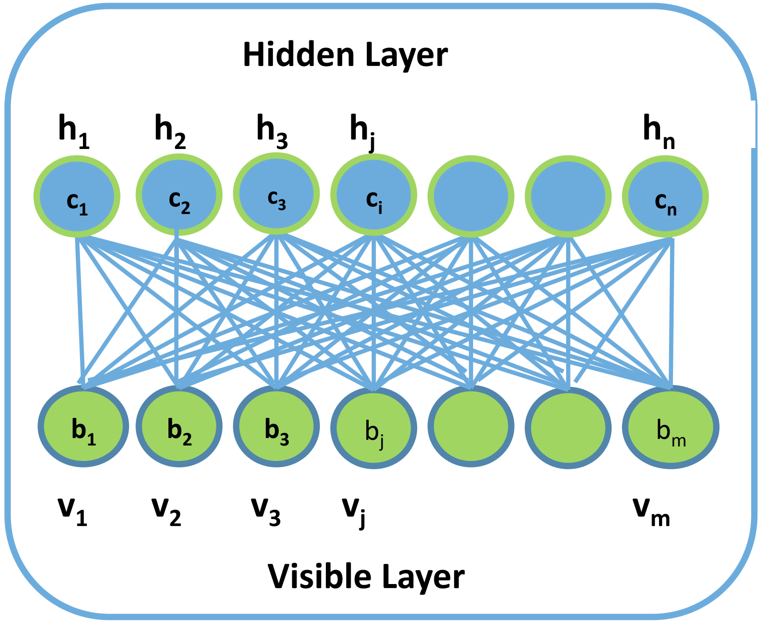
- Lift RBMs to relational data
- Relational random walks to construct relational rules

Outline

- Background
- R^2BM
- Experimental Results
- Conclusion

- Background
- R^2BM
- Experimental Results
- Conclusion

Restricted Boltzmann Machines



$$p(\boldsymbol{v},\boldsymbol{h}) = \frac{e^{-E(\boldsymbol{v},\boldsymbol{h})}}{\boldsymbol{Z}} \text{ where } Z = \sum_{\boldsymbol{v},\boldsymbol{h}} e^{-E(\boldsymbol{v},\boldsymbol{h})}$$

$$\boldsymbol{Energy} : \boldsymbol{E}(\boldsymbol{v},\boldsymbol{h}) = -\sum_{i=1}^{n}\sum_{j=1}^{m} w_{ij} h_i v_j - \sum_{j=1}^{m} b_j v_j - \sum_{i=1}^{n} c_i h_i$$

- Probabilistic Graphical Models (or Stochastic Neural Networks)

- 2 types of Units:
  - visible (Component of observation)
  - hidden (non-linear feature detectors) to model dependencies

- *Popular* and *powerful* probabilistic models

Lifted Relational Random Walks

- Lifted Relational Graph
  - Nodes: Object Types
  - Edges: Relations
- Path Constrained Random Walks (PCRW):
  - Random walks to discover structure
  - Constrained by start and end of the walks
- Features of PCRW:
  - Inverse of Relation Possible
  - Constraints: NoFF, NoBF, NoFB, NoBB

Random Walk:

$S \xrightarrow{\text{takes}} C \xrightarrow{\text{taughtBy}} P$

Clausal Form:

takes(S,C) ∧ taughtBy(C,P)

Random Walk:

$P \xrightarrow{\text{author}} A \xrightarrow{\text{\_author}} P$

Clausal Form:

author(P,A) ∧ _author(A,P)

Lao et al, ECML 2010

- Background
- R^2BM
- Experimental Results
- Conclusion

Relational Restricted Boltzmann Machines

- Lift RBMs to Relational data (R^2BM)

- <u>**Key intuition**</u>: Make the features of RBM relational (and expressive)

- Construct the distributions similar to an SRL model − Different aggregators are possible

R^2BM Learning

- Step 1: Data Transformation
  - Convert predicate logic data to PRW data form

  - Convert N-ary predicates to binary form by introducing Compound Value Type[1]

    Example: N-ary predicate: taught(Prof, Course, Semester)
    Corresponding Binary Predicates:
    taught1(t_id, Prof)
    taught2(t_id,Course)
    taught3(t_id,Semester)

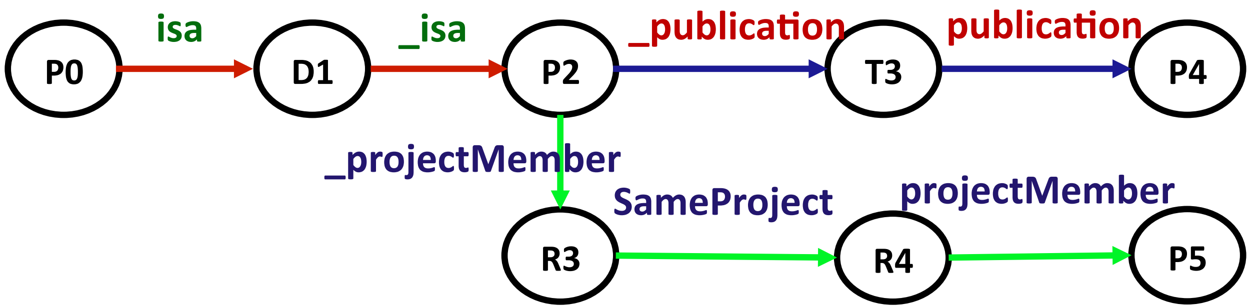
  - Convert Unary Predicate to binary form
    Example = Unary predicate: student(Person)
    isa(Person,'student')

[1]wiki.freebase.com/wiki/Compound_Value_Type

R^2BM Learning

Step 2: Construct Relational Walks

- 2(a): Learn *m* Random Walks on Lifted Relational graph connecting argument type of target example



P : person
D: designation
T: title
R: project

RW1:	isa(P0,D1) ∧ _isa(D1,P2) =>advisedBy(P0,P2)
RW2:	isa(P0,D1) ∧ _isa(D1,P2) ∧ _publication(P2, T3) ∧ publication(T3, P4) =>advisedBy(P0,P4)
RW3:	_publication(P2, T3) ∧ publication(T3, P4) =>advisedBy(P2,P4)
RW4:	_projectMember(P2,R3) ∧ SameProject(R3, R4) ∧ projectMember(R4, P5) =>advisedBy(P2,P5)
RW5:	isa(P0,D1) ∧ _isa(D1,P2) ∧ _projectMember(P2,R3) ∧ SameProject(R3, R4) ∧ projectMember(R4, P5) =>advisedBy(P0,P5)

R^2BM Learning

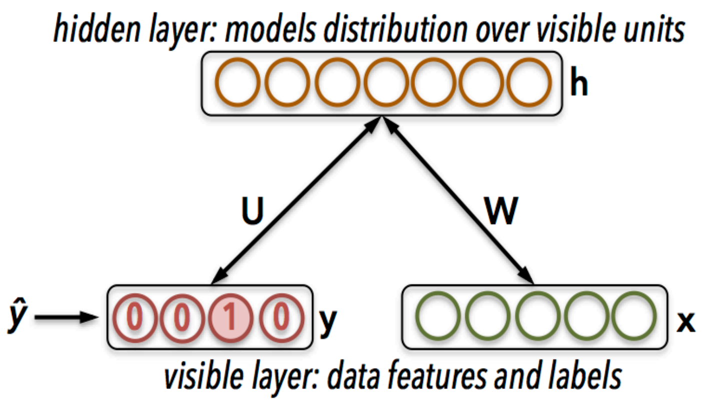
- Step 2: Construct Relational Features
  - 2(b): Two types of aggregators
    - *Existentials* (RRBM-E) – Learning RDNs (Natarajan et al 2012)
    - *Counts* (RRBM-C) – Learning MLNs (Lowd and Domingos 2004)

RRBM-E							RRBM-C					
RW1	**RW2**	**RW3**	**RW4**	**….**	**RWm**		**RW1**	**RW2**	**RW3**	**RW4**	**….**	**RWm**
1	1	0	1	…	1	**advisedBy('A','B')**	6	7	0	10	…	2
0	1	1	1	…	1	**advisedBy('E','F')**	0	17	4	25	…	13
1	1	1	0	…	1	**advisedBy('X','Y')**	10	9	6	0	…	11

Paper discusses an empirical relation to MLN weight learning

R^2BM Learning

- Step 3: Discriminative Learning
  - Learn Discriminative RBM by utilizing the features learnt at Transformation layer



$$p(\hat{y}|\boldsymbol{x}) = \frac{e^{d_{\hat{y}} + \sum_{j=1}^{n} softplus(c_j + U_{j\hat{y}} + \sum_{f=1}^{m} W_{jy} x_f)}}{\sum_{k=1}^{C} e^{d_k + \sum_{j=1}^{n} softplus(c_j + U_{jk} + \sum_{f=1}^{m} W_{jf} x_f)}}$$

softplus(z) = log(1+exp(z))

- Discriminative RBM:
  - Multinomial input layer
  - Sigmoidal hidden layer
  - Bernoulli output layer
- x_f : Random Walks Learnt in previous step
- Model learnt by Stochastic Gradient Descent

To do: Learn with different distributions corresponding to the aggregators

Larochelle et al, ICML 2008

Relational Restricted Boltzmann Machines

- Step 1: Relational Data Transformation
  - Bring relational data to lifted graphical form
  - Bring N-ary predicates to binary form by introducing Compound Value Type
- Step 2: Relational Transformation Layer
  - Learn *m* Random Walks on Lifted Relational graph connecting argument type of target example
  - Two ways of transformation
    - *Existential Semantics* (RRBME): if there exist at least one instance of random walk satisfied for target example
    - *Counts (RRBMC):* # groundings of random walk satisfied for target example
- Step 3: Learning Relational RBM
  - Learn Discriminative RBM by utilizing the features learnt at Transformation layer

- Background
- R^2BM
- Experimental Results
- Conclusion

Experimental Setup

Domains:

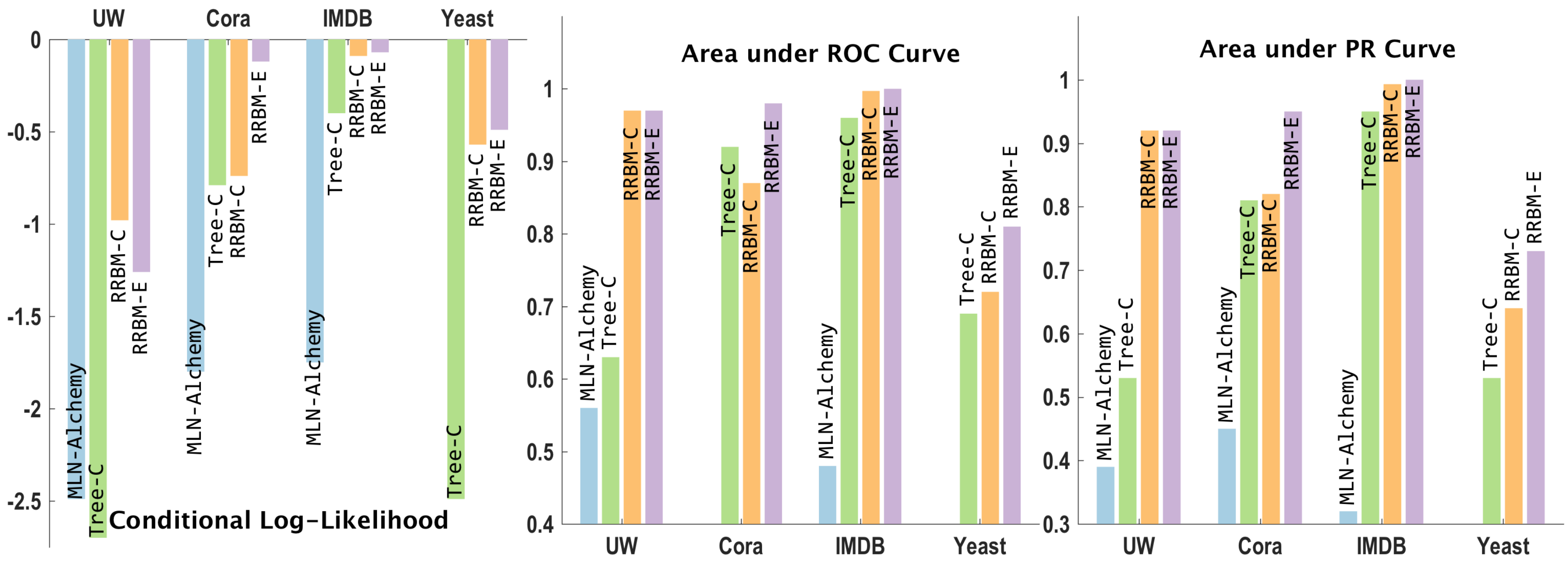
Domain	Target Predicate
UW-CSE	advisedBy(Person,Person)
Cora Entity Resolution	SameVenue(Venue,Venue)
IMDB	WorkedUnder(Person,Person)
Yeast (Lao et al, 2010)	Cites(Paper,Paper)

Comparative Algorithms:

- Baselines: *Tree-Count,* MLN (Alchemy)
- State-of-the-art SRL Methods: RDN-Boost, MLN-Boost

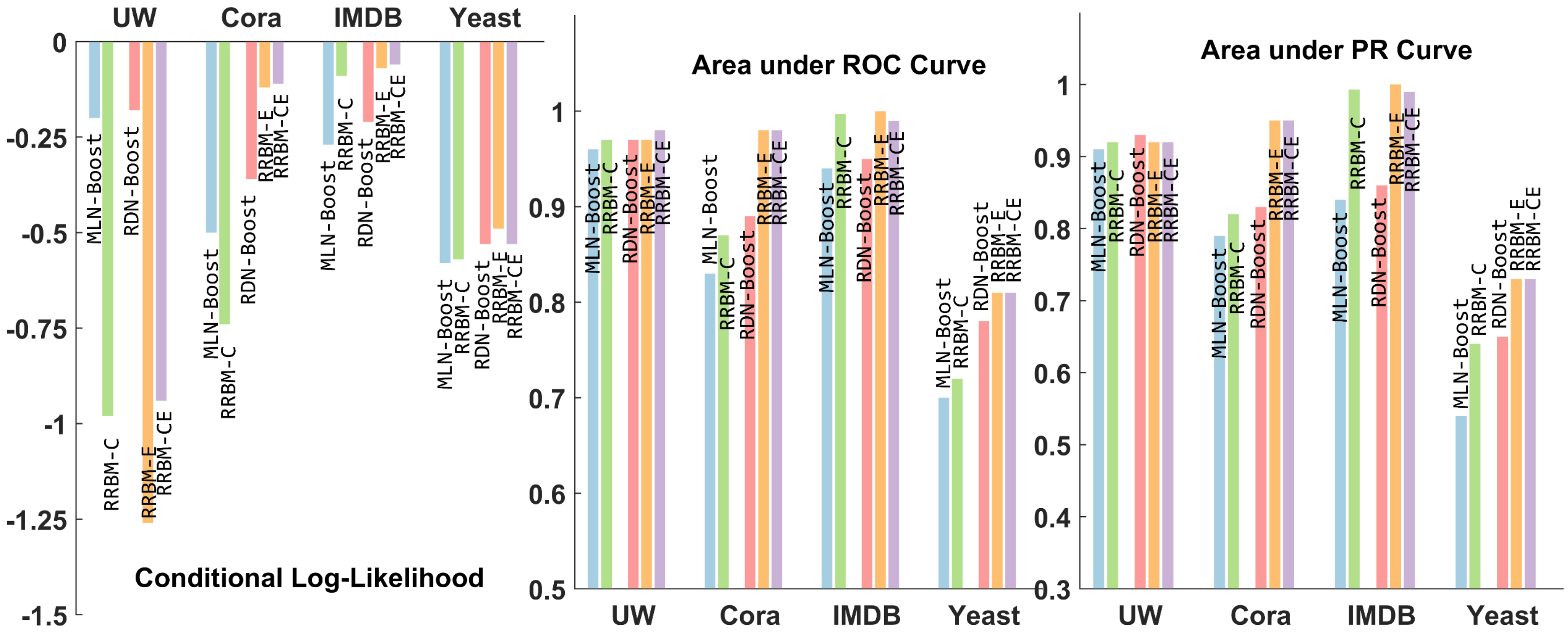
Natarajan et. al, MLJ 2012, Khot et. al, ICDM 2011

RRBM vs Baselines



RRBM outperforms baseline MLN and decision-tree (Tree-C) models

RRBM vs SRL Methods



Better or Comparable performance of RRBM-C to MLN-Boost, RRBM-E to RDN-Boost

- Background
- R^2BM
- Experimental Results
- Conclusion

Discussion

- Method to augment RBMs with relational features
- Connections to existing SRL approaches
- On par with state-of-the-art SRL results
- Future work
  - Multiple distributions
  - Predicate invention using RWs and RBMs
  - More interesting deep models
  - Exploring closing of loop − using deep features to improve log-linear model

Thanks!

Questions??