Probabilistic Soft Logic

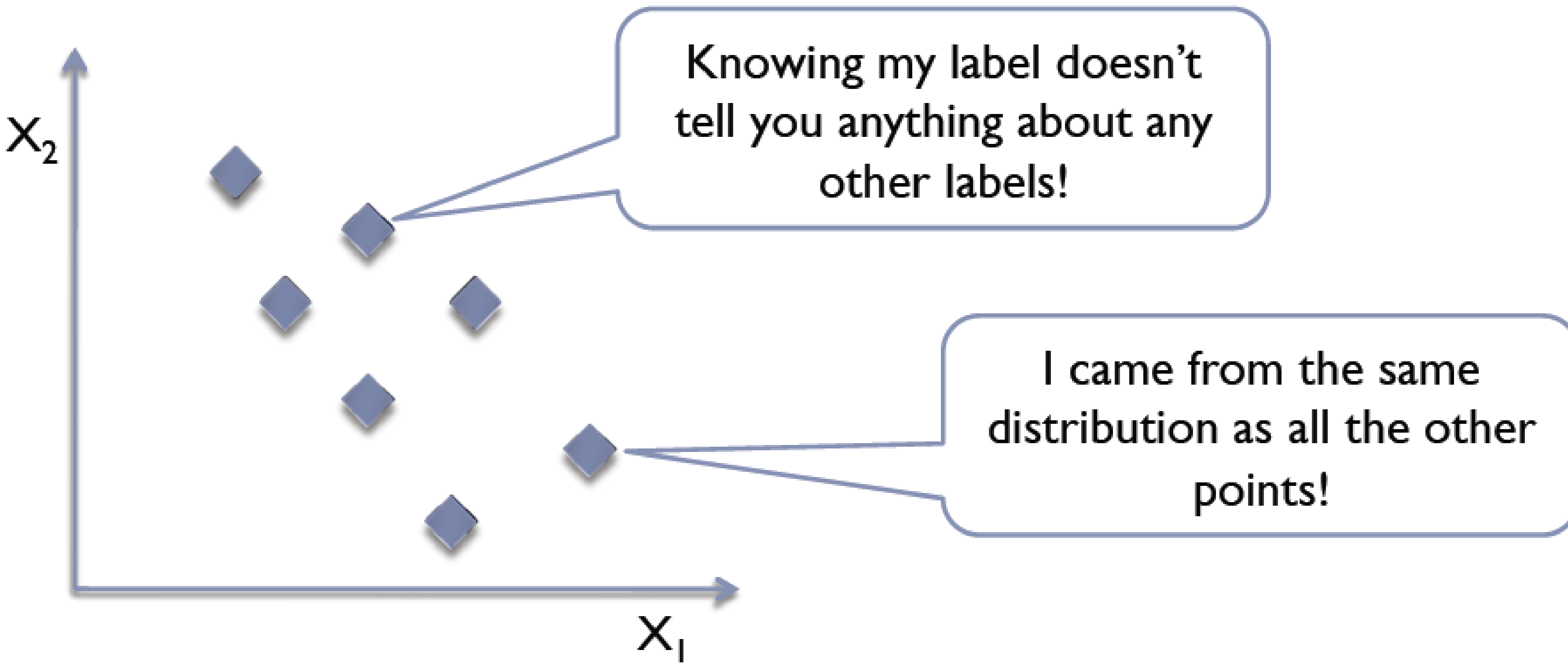
Dr. Fayyaz ul Amir Afsar Minhas

PIEAS Biomedical Informatics Research Lab
Department of Computer and Information Sciences
Pakistan Institute of Engineering & Applied Sciences
PO Nilore, Islamabad, Pakistan
http://faculty.pieas.edu.pk/fayyaz/

Adapted and copied from: Probabilistic Soft Logic: A New Framework for Statistical Relational Learning
https://pdfs.semanticscholar.org/presentation/2cf1/22badd9491bbd9c1272e20c69cf9bca0af15.pdf
Probabilistic Soft Logic: A Scalable Probabilistic Programming Language for Modeling Richly Structured Domains Golnoosh Farnadi, Lise Getoor LINQS Group, UCSC
Hinge-Loss Markov Random Fields and Probabilistic Soft Logic (JMLR 2017)
Probabilistic soft lofic: A short Introduction

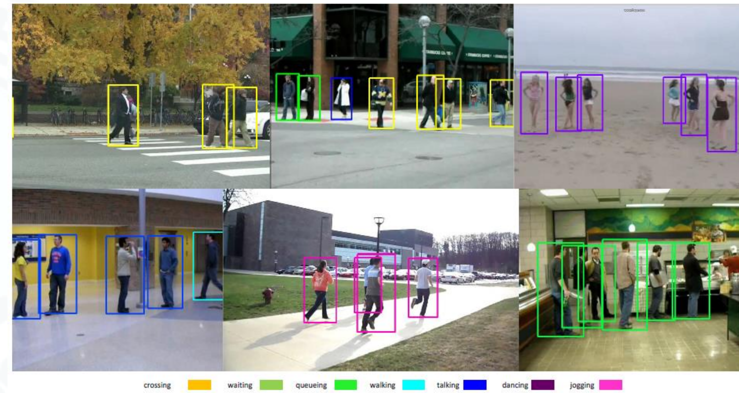
IID

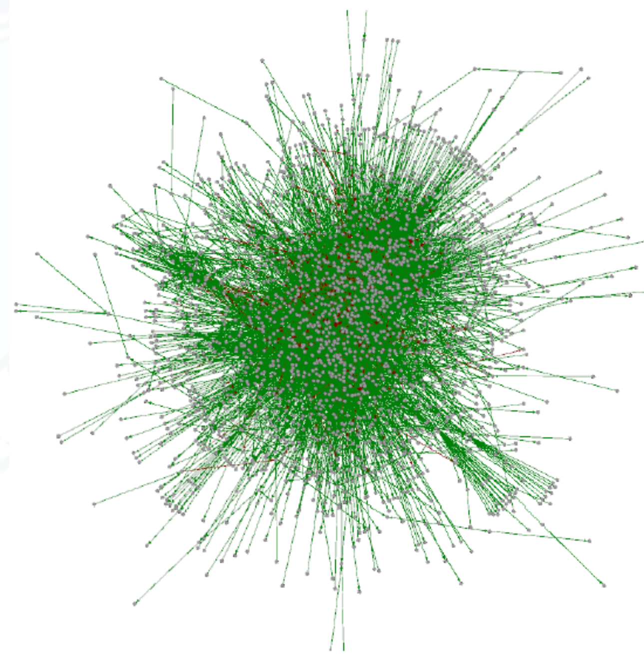
- Typical Machine learning assumes that examples are IID



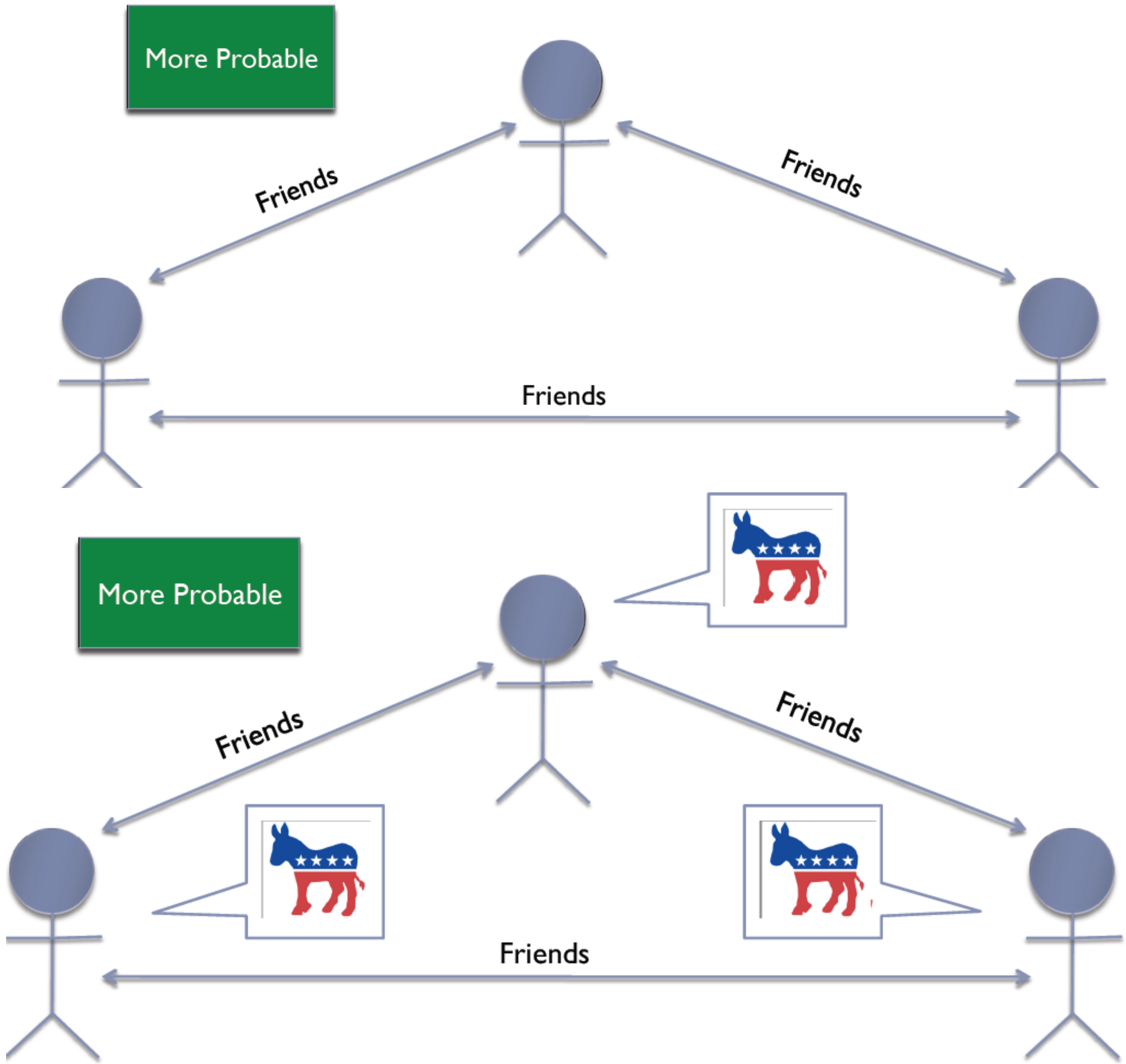
IID in real-world

- A lot of real-world data are not IID
- Data points often have relationships amongst them or model relationships between entities
  - Structured Data
  - Network Data
  - Heterogeneous networks
- Examples
  - Social networks (FB)
  - Document Nets (Wikipedia)
  - Biological networks
  - NLP Data
  - Video Data
- Most machine learning is "model independent"
  - Modeling of causal (cause-effect) relationships
    - Judea Pearl "Theoretical Impediments to Machine Learning with Seven Sparks from the Causal Revolution"
    - Judea Pearl "The Book of Why"
  - How can we model these relationships?





Relational Dependencies

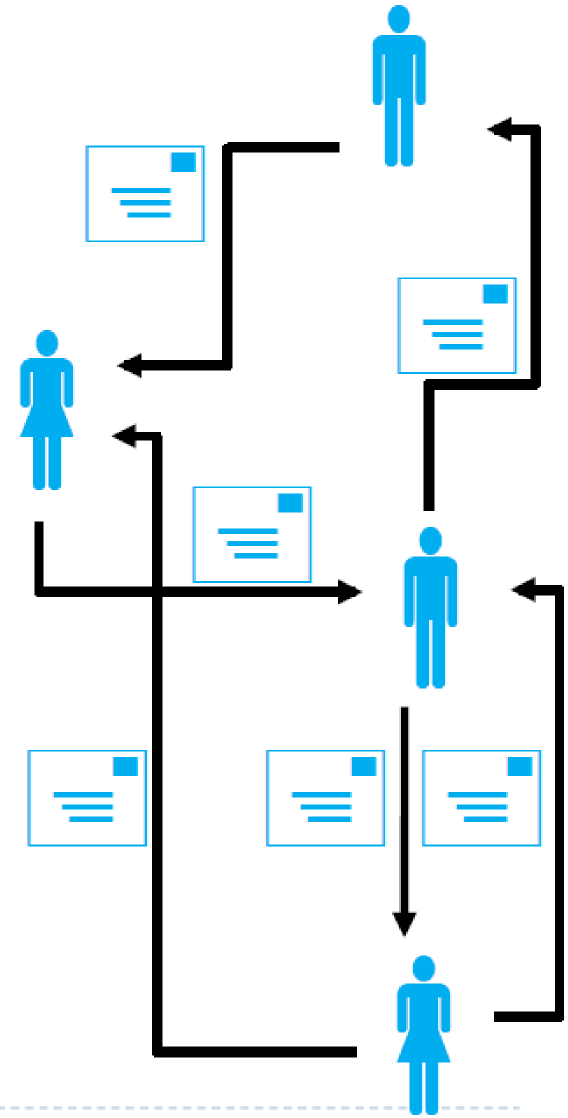


Probabilistic Soft Logic

- A way of
  - Modeling relationships between entities
  - Learning "soft" logical rules from data
- Declarative language based on logic to express statistical relational learning problems
  - Predicate: relationship or property
    e.g., Friends(A, B)
  - Atom: (continuous) random variable
    e.g., Friends(Steve, Jay) = ?
  - Rule: capture dependency or constraint
    e.g., 3.0 : Friends(A, B) & Friends(B, C) ➔ Friends(A, C)
  - Set: define aggregates
    e.g., Average[Friends(Steve, X)]

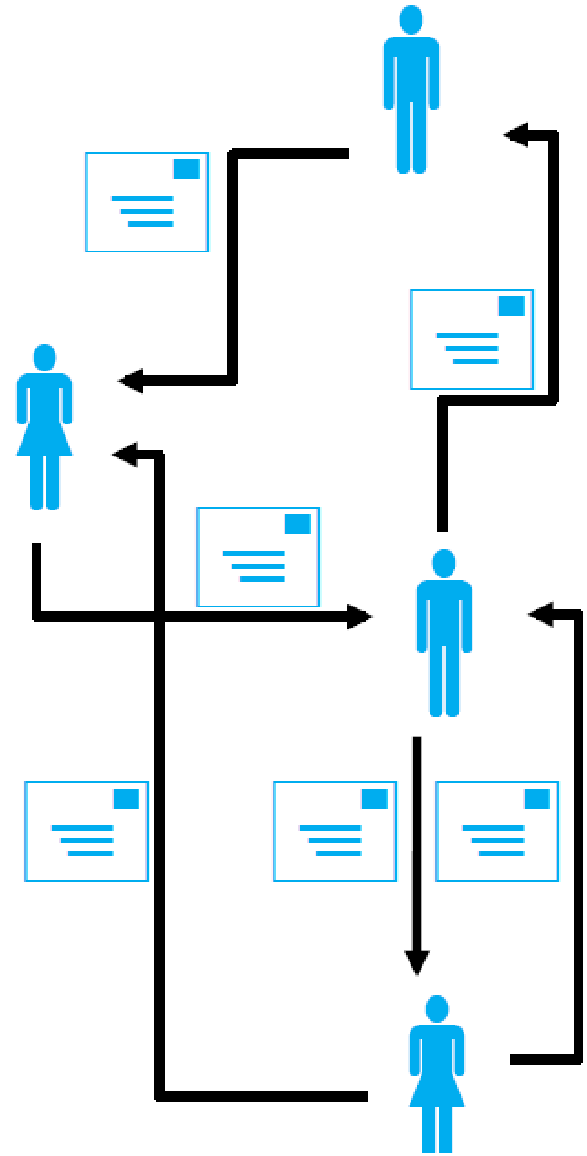
Example: Link Prediction

- Entities
  - People, Emails
- Attributes
  - Words in emails
- Relationships
  - communication, work relationship
- Goal: Identify work relationships
  - Supervisor, subordinate, colleague



Link Prediction

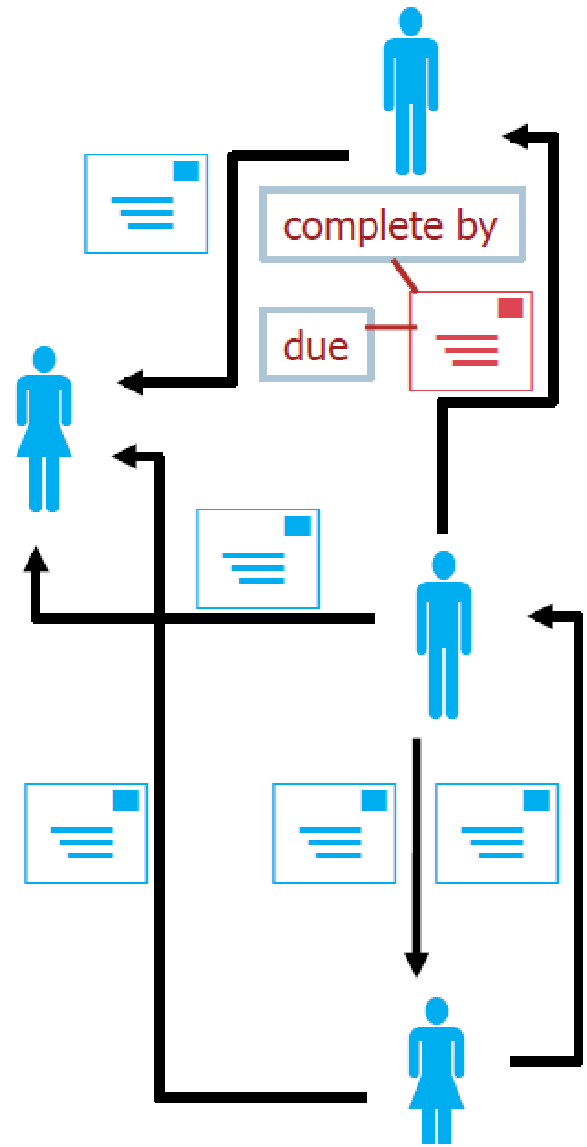
- Use rules to express evidence
  - "If email content suggests type X, it is of type X."
  - "If A sends deadline emails to B, then A is the supervisor of B."
  - "If A is the supervisor of B, and A is the supervisor of C, then B and C are colleagues."



Link Prediction

- Use rules to express evidence
  - "If email content suggests type X, it is of type X."
  - "If A sends deadline emails to B, then A is the supervisor of B."
  - "If A is the supervisor of B, and A is the supervisor of C, then B and C are colleagues."

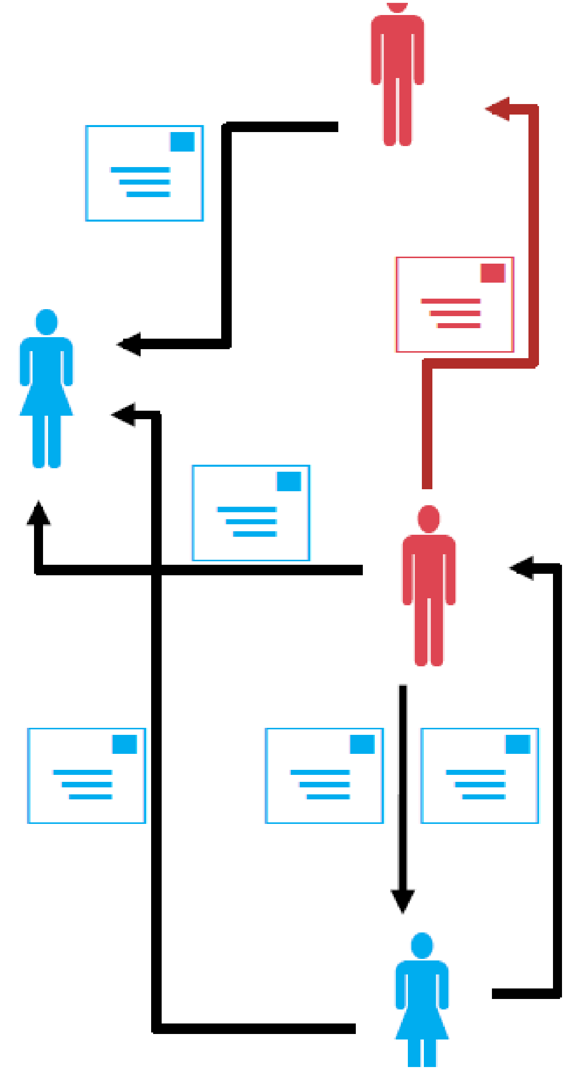
1.0 : Contains(E, "due")
=> HasType(E, "deadline")



Link Prediction

- Use rules to express evidence
  - "If email content suggests type X, it is of type X."
  - "If A sends deadline emails to B, then A is the supervisor of B."
  - "If A is the supervisor of B, and A is the supervisor of C, then B and C are colleagues."

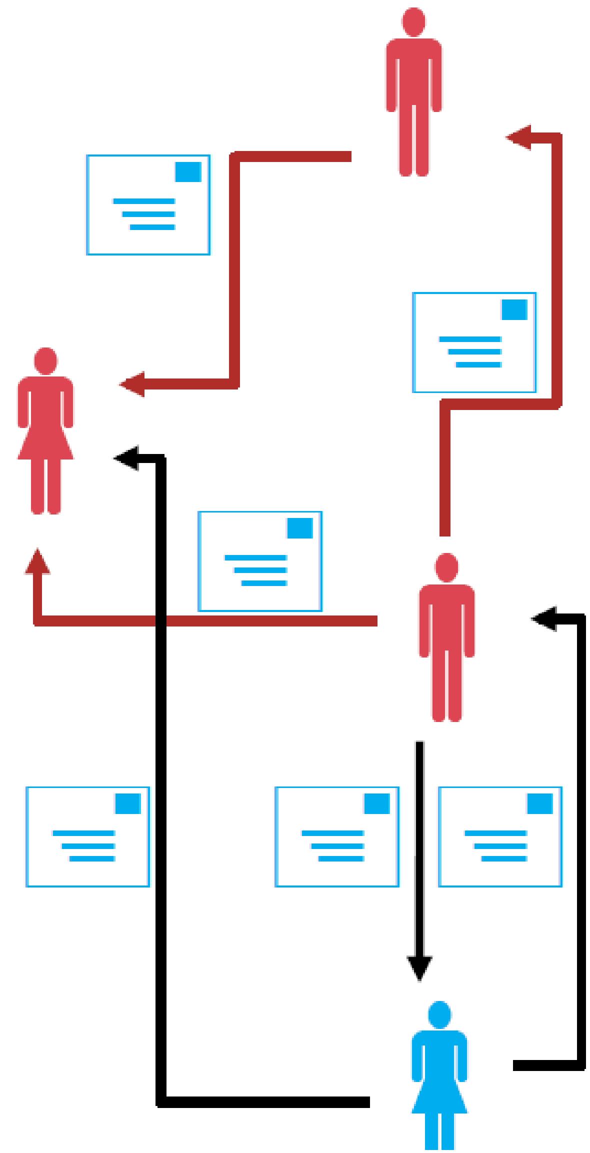
2.0 : Sent(A, B, E) & HasType(E, "deadline") => Supervisor(A, B)



Link Prediction

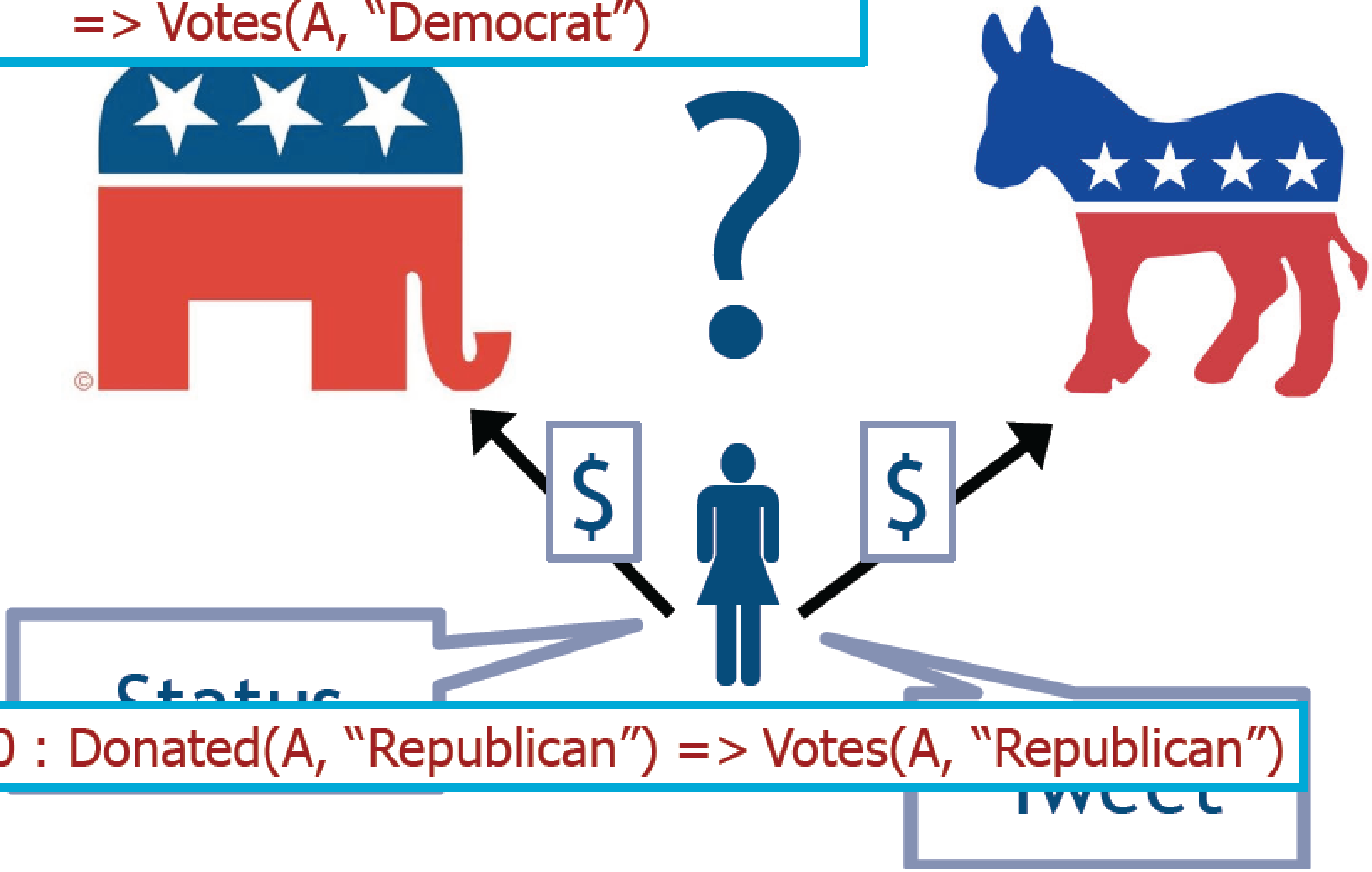
- Use rules to express evidence
  - "If email content suggests type X, it is of type X."
  - "If A sends deadline emails to B, then A is the supervisor of B."
  - "If A is the supervisor of B, and A is the supervisor of C, then B and C are colleagues."

1.5 : Supervisor(A, B) & Supervisor(A, C)
=> Colleagues(B, C)



Example: Collective Classification

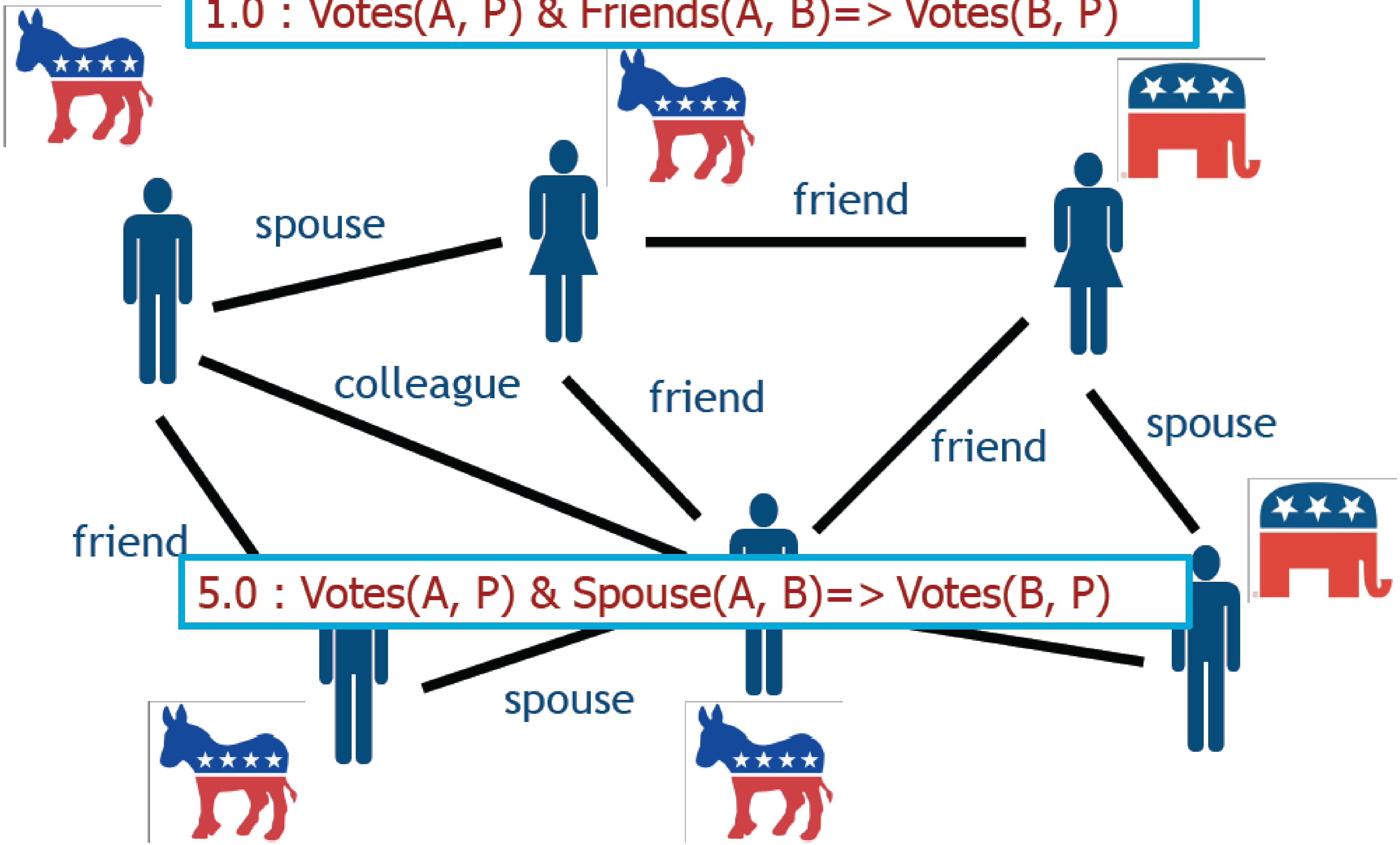
1.0 : Mentioned(A, “Barack Obama”)
=> Votes(A, “Democrat”)



5.0 : Donated(A, “Republican”) => Votes(A, “Republican”)

Example: Collective Classification

1.0 : Votes(A, P) & Friends(A, B)=> Votes(B, P)



5.0 : Votes(A, P) & Spouse(A, B)=> Votes(B, P)

PSL Modeling

- A PSL program consists of a set of first order logic rules with conjuctive bodies (ands) and single literal heads
- Rules are labeled with non-negative weights
- Example:
  - The following example program encodes a simple model to predict voter behavior based on a social network with two types of links denoting friend and spouse relationships

Body **Head**

$$0.3 : friend(B, A) \wedge votesFor(A, P) \rightarrow votesFor(B, P)$$

$$0.8 : spouse(B, A) \wedge votesFor(A, P) \rightarrow votesFor(B, P)$$

A Short Introduction to Probabilistic Soft Logic
Angelika Kimmig et al.

PSL Modeling

- In PSL, the logic is "soft" – it uses soft truth values from the interval [0,1] instead of the extremes 0 (false) and 1 (true) only
  - Given a set of atoms $l = \{l_1, l_2, \dots, l_n\}$ (atomic statements), e.g., Friend(P,Q), we call the mapping $I: l \rightarrow [0,1]^n$ from atoms to soft-truth values an interpretation
  - PSL defines a probability distribution over interpretations that makes those satisfying more ground rule instances more probable by assigning weights to rules based on data

PSL Modeling: Combination Functions

- $\vee$, $\wedge$: $[0,1]^n \rightarrow [0,1]$
- Rules will behave like Boolean logic
  - If body is high, rule only "happy" if head is high

PSL Modeling: Combination Functions

- $\vee, \wedge : [0,1]^n \to [0,1]$

- Rules will behave like Boolean logic
  - If body is low, rule is always "happy"

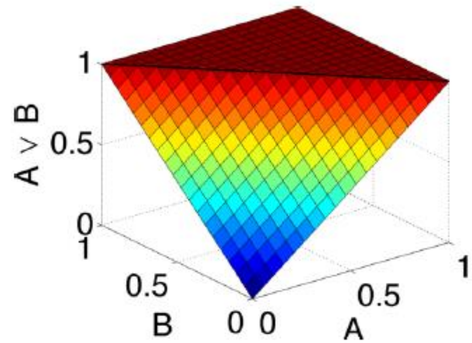
$$\text{Friends(Jay, Steve)} \wedge \text{Friends(Steve, Ben)} \rightarrow \text{Friends(Jay, Ben)}$$

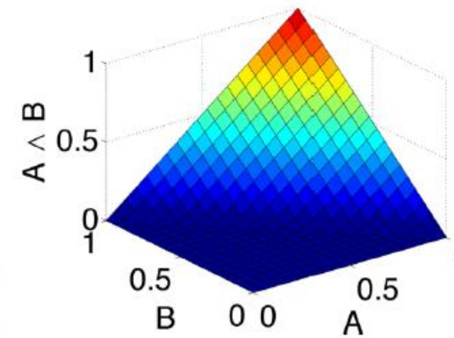
$$\text{Friends(Jay, Steve)} \wedge \text{Friends(Steve, Ben)} \rightarrow \text{Friends(Jay, Ben)}$$

Softening Logic Operations

- PSL Defines soft logic operations as follows

$$\ell_1 \tilde{\wedge} \ell_2 = \max\{0, I(\ell_1) + I(\ell_2) - 1\},$$
$$\ell_1 \tilde{\vee} \ell_2 = \min\{I(\ell_1) + I(\ell_2), 1\},$$
$$\tilde{\neg} l_1 = 1 - I(\ell_1),$$





- A rule $r_{body} \rightarrow r_{head} \equiv \tilde{\neg} r_{body} \tilde{\vee} r_{head}$ is satisfied if
  - Body is low
  - Body is high and head is high
- We use this to define a "distance from satisfaction" of a rule for a given interpretation

$$d_r(I) = \max\{0, I(r_{body}) - I(r_{head})\}.$$

Soft Logic Operations Example

- Consider the following

$$0.3 : friend(B, A) \wedge votesFor(A, P) \rightarrow votesFor(B, P) \quad (1)$$

$$0.8 : spouse(B, A) \wedge votesFor(A, P) \rightarrow votesFor(B, P). \quad (2)$$

For instance, consider the interpretation $I = \{spouse(b, a) \mapsto 1, votesFor(a, p) \mapsto 0.9, votesFor(b, p) \mapsto 0.3\}$, and let r be the corresponding ground instance of Rule (2) above. We get $I(r_{body}) = \max\{0, 1 + 0.9 - 1\} = 0.9$ and thus $d_r(I) = \max\{0, 0.9 - 0.3\} = 0.6$, whereas the distance would be 0 if the head had truth value 0.9 or greater.

Probabilistic Learning

- Given a set of ground atoms l of interest, a PSL program induces a distribution over possible interpretations I
- Let R be the set of all ground rules that are instances of a rule in the program and only
- mention atoms in l. The probability density function f over I is:

$$f(I) = \frac{1}{Z}\exp[-\sum_{r \in R} \lambda_r (d_r(I))^p] \quad ; \quad Z = \int_I \exp[-\sum_{r \in R} \lambda_r (d_r(I))^p],$$

Probabilistic Model

Rule's weight

Rule's distance to satisfaction

$$d_r(I) = \max(0, I_{r,\text{body}} - I_{r,\text{head}})$$

Probability density over interpretation I

$$f(I) = \frac{1}{Z}\exp[-\sum_{r\in R}\lambda_r(d_r(I))^p]$$

Distance exponent in {1, 2}

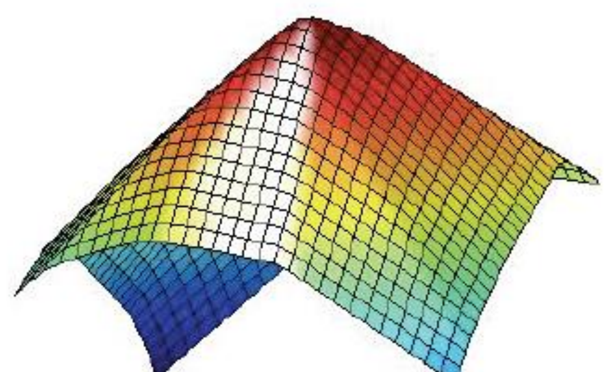
Normalization constant

Set of ground rules

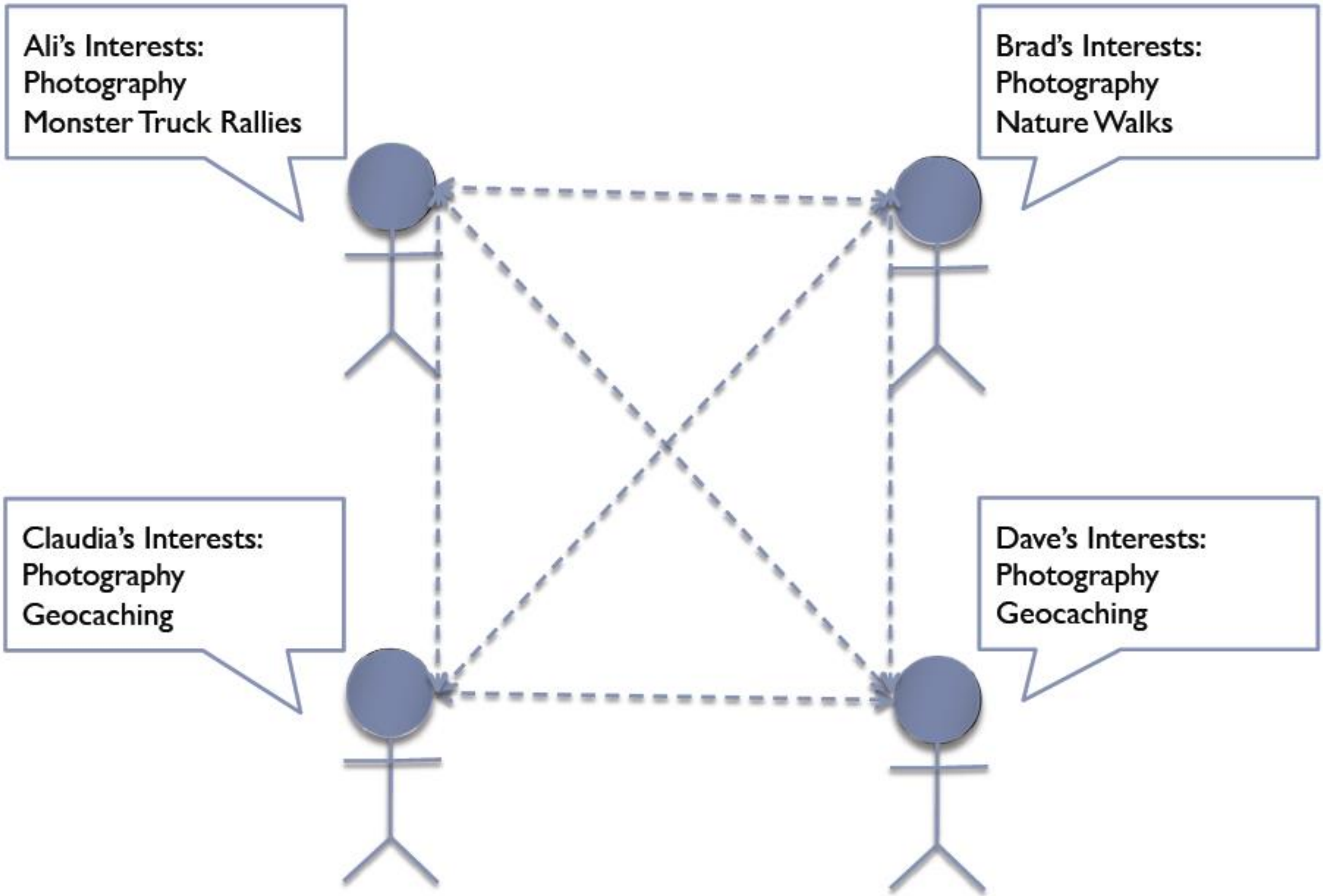
Hingle Loss Markov Random Fields

$$p(\mathbf{Y}|\mathbf{X}) = \frac{1}{Z}\exp\left[-\sum_{j=1}^{m} w_j \max\{\ell_j(\mathbf{Y},\mathbf{X}),0\}^{p_j}\right]$$

- PSL models ground out to HL-MRFs
- Continuous variables in [0,1]
- Potentials are hinge-loss functions
- Subject to arbitrary linear constraints
- Log-concave!



Example



Ali's Interests:
Photography
Monster Truck Rallies
Brad's Interests:
Photography
Nature Walks
Claudia's Interests:
Photography
Geocaching
Dave's Interests:
Photography
Geocaching

Example

RULES:

2.0 Interest(P1, I) & Interest(P2, I) ➔ Friends(P1, P2)

2.5 Friends(P1, P2) & Friends(P2, P3) ➔ Friends(P1, P3)

5.0 Interest(P1, Monster Truck Rallies) & Interest(P2, Nature Walks) ➔ !Friends(P1,P2)

1.0 !Friends(P1,P2)

Constructing PSL

- 1. Ground out all rules

  - Interest(P1, I) & Interest(P2, I) → Friends(P1, P2)
    - Interest(Ali, Photography) & Interest(Brad, Photography) → Friends(Ali, Brad)
    - Interest(Claudia, Photography) & Interest(Dave, Photography) → Friends(Claudia, Dave)
    - Interest(Claudia, Geocaching) & Interest(Dave, Geocaching) → Friends(Claudia, Dave)
    - etc.

  - Interest(P1, M.T.R.) & Interest(P2, N.W.) → !Friends(P1, P2)
    - Interest(Ali, M.T.R.) & Interest(Brad, N.W.) → !Friends(Ali, Brad)

  - Friends(P1, P2) & Friends(P2, P3) → Friends(P1, P3)
    - Friends(Ali, Brad) & Friends(Brad, Dave) → Friends(Ali, Dave)
    - etc.

Constructing PSL

- Convert ground rules to hinge-loss functions
  - Start with a ground rule
    - Friends(Ali, Brad) & Friends(Brad, Dave) ➔ Friends(Ali, Dave)
  - Map atoms to random variables
    - Friends(Ali, Brad) $= Y_1$
    - Friends(Brad, Dave) $= Y_2$
    - Friends(Ali, Dave) $= Y_3$
  - Interpret with t-norm
    - $\min\{2 - Y_1 - Y_2 + Y_3, 1\}$
  - Subtract from 1 to find distance to satisfaction

$$
\begin{aligned}
& 1 - \min\{2 - Y_1 - Y_2 + Y_3, 1\} \\
= \quad & \max\{Y_1 + Y_2 - Y_3 - 1, 0\}
\end{aligned}
$$

Constructing a HL-MRF

- 1) Ground out all rules
- 2) Convert ground rules to hinge-loss functions
- 3) Weight hinge-loss functions and embed in HL-MRF

$$p(\mathbf{Y}|\mathbf{X}) = \frac{1}{Z(w, \mathbf{X})} \exp\left[-\sum_{j=1}^{m} w_j \left[\max\left\{ \ell_j(\mathbf{Y}, \mathbf{X}), 0 \right\} \right]^{\{1,2\}} \right]$$

Making Predictions

Making Predictions

- Want to find a most probable explanation (MPE)

$$
\begin{aligned}
\arg\max_{\mathbf{Y}} P(\mathbf{Y}|\mathbf{X}) \quad &\equiv \quad \arg\min_{\mathbf{Y}\in[0,1]^n} f(\mathbf{Y},\mathbf{X}) \\
&\equiv \quad \arg\min_{\mathbf{Y}\in[0,1]^n} \sum_{j=1}^{m} w_j \phi_j(\mathbf{Y},\mathbf{X}) \\
&\quad\ \text{s.t.} \quad C_k(\mathbf{Y},\mathbf{X}) = 0, \quad \forall k \in \mathcal{E} \\
&\quad\ \text{and} \quad C_k(\mathbf{Y},\mathbf{X}) \geq 0, \quad \forall k \in \mathcal{I}
\end{aligned}
$$

Learning

- Learning from training data
- No need to hand-code rule-weights
- Large-Margin estimation (Bach, Huang, London, Getoor 2013)

Implementation

- Paper: Hinge-Loss Markov Random Fields and Probabilistic Soft Logic
  - Just download the examples from https://github.com/linqs/psl-examples
  - Should have Java and Internet (simply run.sh)

- https://github.com/linqs/psl/wiki
- https://github.com/linqs/psl/wiki/Using-the-CLI

Inferring "Knows"

- Prediction task: Determine if two people know each other or not
- Model

```
• 20:    Lived(P1,L) & Lived(P2,L) & P1!=P2   -> Knows(P1,P2) ^2
• 5:     Lived(P1,L1) & Lived(P2,L2) & P1!=P2 & L1!=L2  -> !Knows(P1,P2) ^2
• 10:    Likes(P1,L) & Likes(P2,L) & P1!=P2  -> Knows(P1,P2) ^2
• 5:     Knows(P1,P2) & Knows(P2,P3) & P1!=P3 -> Knows(P1,P3) ^2
• 10000: Knows(P1,P2) -> Knows(P2,P1) ^2
• 5:     !Knows(P1,P2) ^2
```

- The model is expressing the intuition that people who have lived in the same location or like the same thing may know each other. The integer values at the beginning of rules indicate the weight of the rule. Intuitively, this tells us the relative importance of satisfying this rule compared to the other rules. The ^2 at the end of the rules indicates that the hinge-loss functions based on groundings of these rules are squared, for a smoother tradeoff.
- Data

```
predicates:
  Knows/2: open //TO BE PREDICTED
  Likes/2: closed
  Lived/2: closed

observations:
  Knows : ../data/knows_obs.txt
  Lived : ../data/lived_obs.txt
  Likes : ../data/likes_obs.txt

targets:
  Knows : ../data/knows_targets.txt

truth:
  Knows : ../data/knows_truth.txt
```

Output

```
KNOWS('Sabina', 'Arti') = 0.7194742867561412
KNOWS('Dhanya', 'Elena') = 0.3682973941849134
KNOWS('Elena', 'Sabina') = 0.3287882658219531
```

https://github.com/linqs/psl-examples

Practical Application: Inferring drug-target interactions

Given known interactions, predict unseen interactions between drugs and targets

```
//Drug similarity propagation
20: Interacts(D1,T) & ChemicalSimilar(D1,D2)   ->
     Interacts(D2,T)
20: Interacts(D1,T) & SideEffectSimilar(D1,D2) ->
     Interacts(D2,T)
30: Interacts(D1,T) & AnnotationSimilar(D1,D2) ->
     Interacts(D2,T)

//Target similarity propagation
30: Interacts(D,T1) & SequenceSimilar(T1,T2) -> Interacts(D,T2)
20: Interacts(D,T1) & OntologySimilar(T1,T2) -> Interacts(D,T2)


//Prior
10: !Interacts(D,T)
```

https://github.com/shobeir/fakhraei_tcbb2014

Task: Find new interactions between drugs and targets in the DrugBank dataset.

~300 drugs and 200 targets

Newly Discovered Interactions

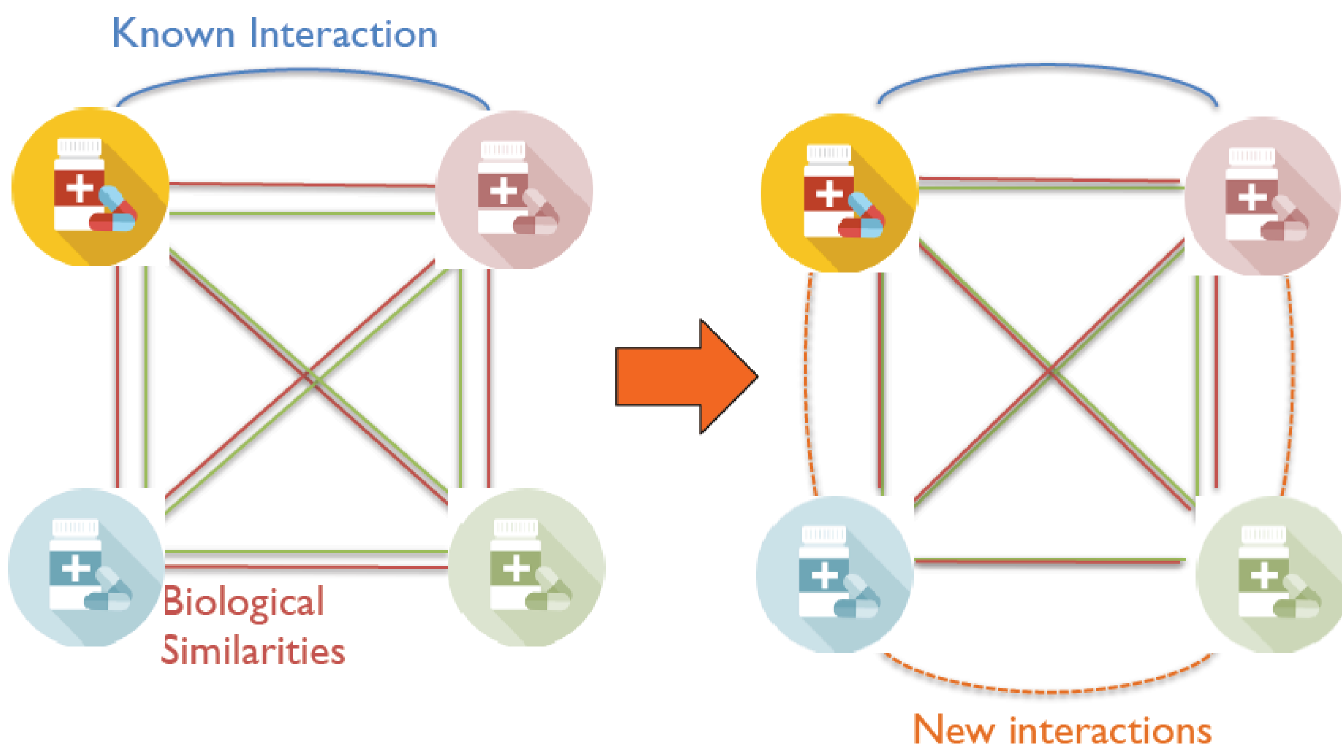
	AUC	AUPR	P@130
Perlman et al.	0.921	0.309	0.393
PSL-Model	**0.926**	**0.344**	**0.460**

Found 197 out of 78,750 possible interactions!

"Network-based Drug-Target Interaction Prediction with Probabilistic Soft Logic", S. Fakhraei, B. Huang, L. Raschid, and L. Getoor, IEEE Transactions on Computational Biology and Bioinformatics (IEEE-TCBB), 2014.

Practical Example: Inferring drug interactions

Predicting interactions between drugs identifies potentially harmful combinations and aids in patient care



Model

```
//Drug similarity triadic structure
20: Interacts(D1,D2) & ChemicalSimilar(D2,D3)  -> Interacts(D1,D3)
20: Interacts(D1,D2) & SideEffectSimilar(D2,D3) -> Interacts(D1,D3)
30: Interacts(D1,D2) & AnnotationSimilar(D2,D3) -> Interacts(D1,D3)
30: Interacts(D1,D2) & PPISimilar(D2,D3) -> Interacts(D1,D3)
30: Interacts(D1,D2) & GeneOntologySimilar(D2,D3) ->
     Interacts(D1,D3)
30: Interacts(D1,D2) & LigandSimilar(D2,D3) -> Interacts(D1,D3)
30: Interacts(D1,D2) & SequenceSimilar(D2,D3) -> Interacts(D1,D3)


//Symmetry Constraints
Interacts(D1,D2) = Interacts(D2,D1)

//Prior
10: !Interacts(D1,D2)
```

https://bitbucket.com/linqs/psl-drug-interaction-prediction

Predicting Drug Interactions in DrugBank

Task: Predict unseen held-out drug-drug interactions from known interactions in DrugBank

~300 drugs, both adverse and beneficial interactions

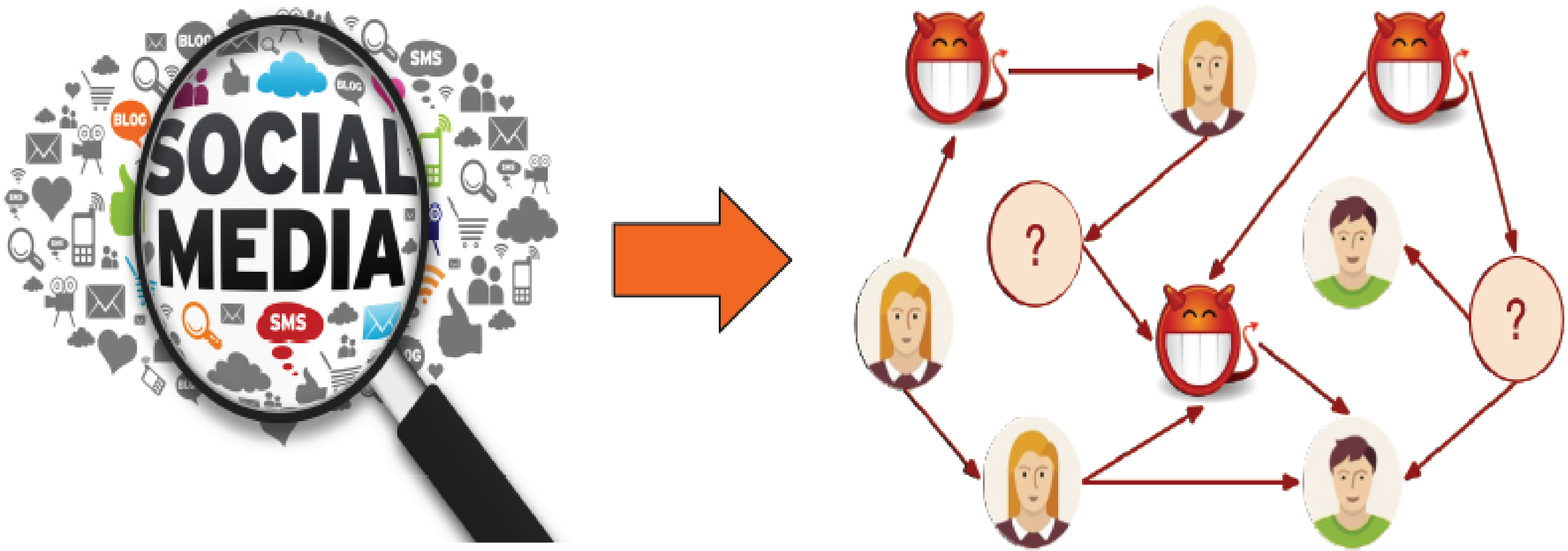
Predicted Interactions

DRUGBANK
Open Data Drug & Drug Target Database

	AUPR	AUC	F1
INDI System	0.47	0.91	0.51
PSL-Model	**0.69**	**0.96**	**0.67**

Sridhar, D., Fakhraei, S., & Getoor, L. (2016). "A probabilistic approach for collective similarity-based drug–drug interaction prediction." *Bioinformatics*, *32*(20), 3175-3182.

Social Spammer Detection



Importance:

- 1 in 200 social messages contain spam
- Social spam grew by more than 350% between Jan-Jul 2013

Collective Spammer Detection in Evolving Multi-Relational Social Networks, S. Fakhraei, J. Foulds, M. Shashanka, L. Getoor. ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD) 2015

Model

```
//User generated reports
30: Credible(U1) & ReportedSpammer(U1,U2) -> Spammer(U2)

//Collective credibility
25: Spammer(U2)  & ReportedSpammer(U1,U2) -> Credible(U1)
25: !Spammer(U2) & ReportedSpammer(U1,U2) -> !Credible(U1)

//Prior credibility
20: PriorCredible(U)  -> Credible(U)
20: !PriorCredible(U) -> !Credible(U)

//Prior
10: !Spammer(U)
```

Finding Social Spammers in Tagged.com

Task: Detecting social spammers in tagged.com social network using user-generated spammer reports.

Attributes: Gender, Age, Account Age, Label

Links: 8 Actions such as Like, Poke, Report Abuse, etc.

Spammers Detected — TAGGED

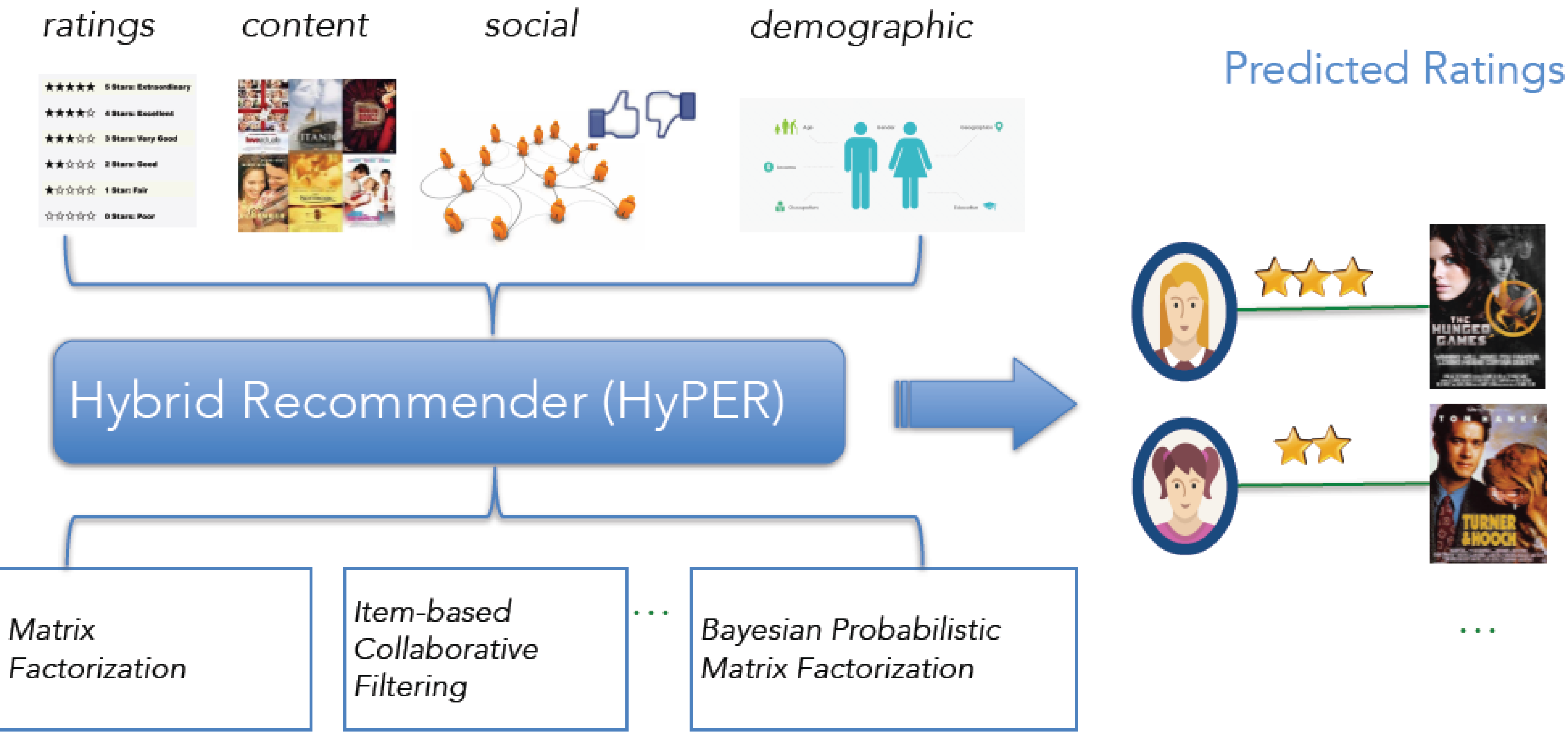
	AUC	AUPR
Using only reports	0.611	0.674
Using report and credibility	0.862	0.869
PSL (fully collective model)	**0.873**	**0.884**

Finding the 4% spammers out of 116,284 users

https://obj.umiacs.umd.edu/tagged_social_spam/index.html

Hybrid Recommender Systems

Improve recommendations by combining data sources & recommenders



HyPER: A Flexible and Extensible Probabilistic Framework for Hybrid Recommender Systems Kouki, Fakhraei, Foulds, Eirinaki, Getoor, RecSys15

```
//Similar Items
10: Rating(U,I1) & PearsonSimilarityItems(I1,I2) ->
Rating(U,I2)
10: Rating(U,I1) & ContentSimilarityItems(I1,I2) ->
Rating(U,I2)

//Similar Users
10: Rating(U1,I) & PearsonSimilarityUsers(U1,U2) ->
Rating(U2,I)
10: Rating(U1,I) & CosineSimilarityUsers (U1,U2) ->
Rating(U2,I)

//Social Information
10: Friends(U1,U2) & Rating(U1,I) -> Rating(U2,I)

//Other Recommenders
10: MFRating(U,I)   -> Rating(U,I)
10: BPMFRating(U,I) -> Rating(U,I)

//Average Priors
1:  AvgUserRating(U) -> Rating(U,I)
1:  AvgItemRating(I) -> Rating(U,I)
```

https://github.com/pkouki/recsys2015

Predicting Ratings with HyPER

Task: Predict missing ratings

- Yelp: 34K users, 3.6K items, 99K ratings, 81K friendships, 500 business categories
- Last.fm: 1.8K users, 17K items, 92K ratings, 12K friendships, 9.7K artist tags

yelp

Model	RMSE
Item-based	1.216
MF	1.251
BPMF	1.191
Naïve Hybrid	1.179
BPMF-SRIC	1.191
HyPER	1.173

last.fm

Model	RMSE
Item-based	1.408
MF	1.178
BPMF	1.008
Naïve Hybrid	1.067
BPMF-SRIC	1.015
HyPER	1.001

How a Pioneer of Machine Learning Became One of Its Sharpest Critics

- **Interviewer:** People are excited about the possibilities for AI. You're not?
  - **Pearl:** As much as I look into what's being done with deep learning, I see they're all stuck there on the level of associations. Curve fitting. That sounds like sacrilege, to say that all the impressive achievements of deep learning amount to just fitting a curve to data. From the point of view of the mathematical hierarchy, no matter how skillfully you manipulate the data and what you read into the data when you manipulate it, it's still a curve-fitting exercise, albeit complex and nontrivial.
- **Interviewer:** The way you talk about curve fitting, it sounds like you're not very impressed with machine learning.
  - **Pearl:** No, I'm very impressed, because we did not expect that so many problems could be solved by pure curve fitting. It turns out they can. But I'm asking about the future—what next? Can you have a robot scientist that would plan an experiment and find new answers to pending scientific questions? That's the next step. We also want to conduct some communication with a machine that is meaningful, and meaningful means matching our intuition. If you deprive the robot of your intuition about cause and effect, you're never going to communicate meaningfully. Robots could not say "I should have done better," as you and I do. And we thus lose an important channel of communication.

End of Lecture

We have to equip machines with a model of the environment. If a machine does not have a model of reality, you cannot expect the machine to behave intelligently in that reality. The first step, one that will take place in maybe 10 years, is that conceptual models of reality will be programmed by humans.

The next step will be that machines will postulate such models on their own and will verify and refine them based on empirical evidence. That is what happened to science; we started with a geocentric model, with circles and epicycles, and ended up with a heliocentric model with its ellipses.

Judea Peal

https://www.theatlantic.com/technology/archive/2018/05/machine-learning-is-stuck-on-asking-why/560675/