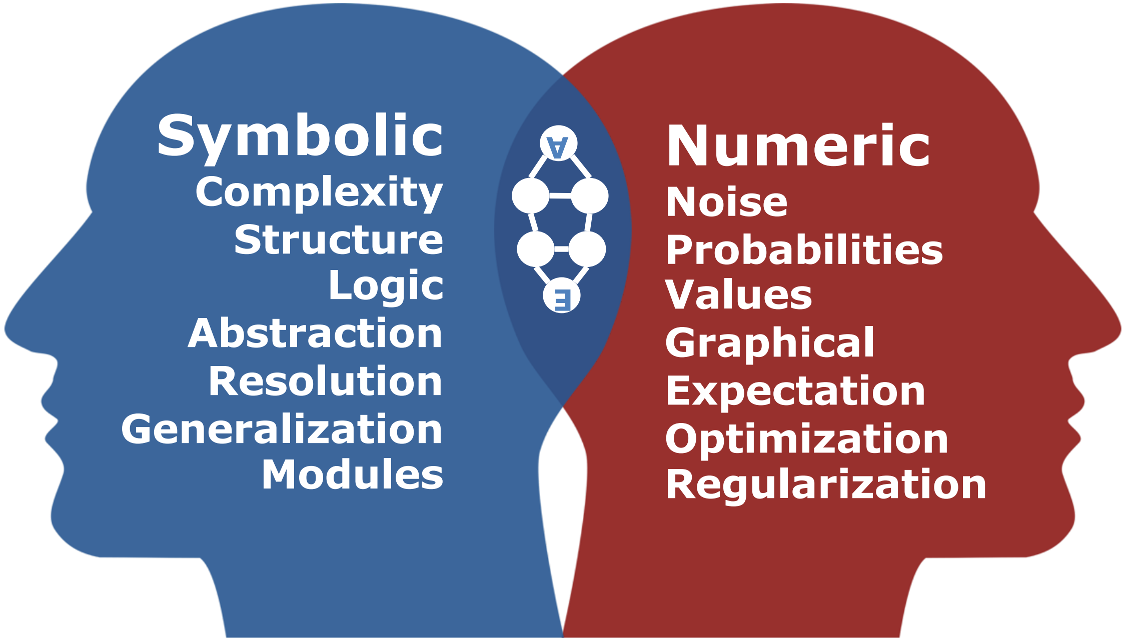


Statistical Relational AI

Big Picture and Motivation

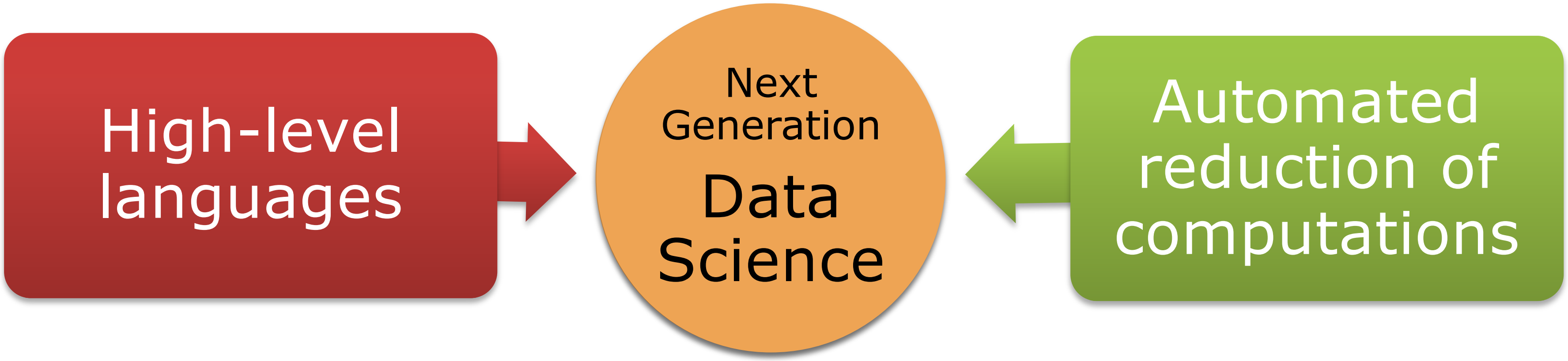
Kristian Kersting

Thanks to Vincent Conitzer, Rina Dechter, Luc De Raedt, Pedro Domingos, Peter Flach, Dieter Fensel, Florian Ficher, Vibhav Gogate, Carlos Guestrin, Daphen Koller, Nir Friedman, Ray Mooney, Sriraam Natarajan, David Poole, Fabrizio Riguzzi, Dan Suciu, Guy van den Broeck, and many others for making their slides publically available

TECHNISCHE UNIVERSITÄT DARMSTADT

Big Vision

Data science needs a crossover with statistical databases, software engineering, and AI



- High-level languages for data science increase the number of people who can successfully build data science machines and make experts more effective
- To deal with the computational complexity, we need ways to automatically reduce the solver costs

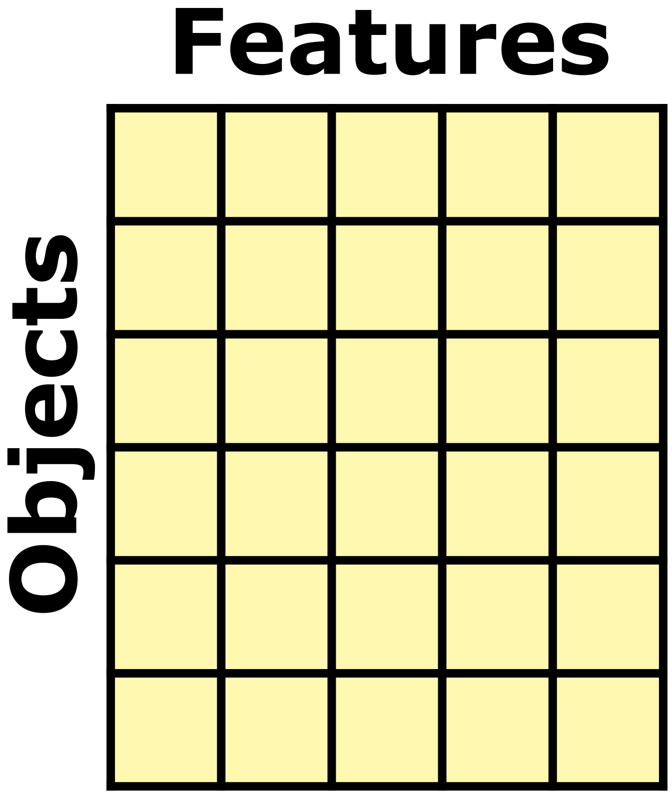
The Thesis

The next breakthrough in data science/AI may not only be a new data analytics algorithm…

…but also the ability to rapidly combine, deploy, and maintain existing algorithms

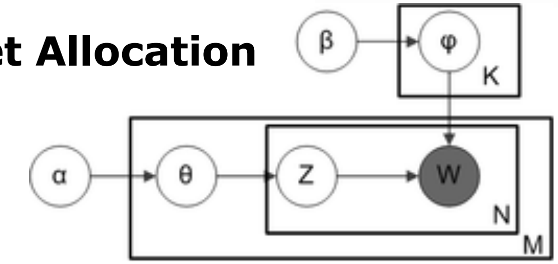
Arms race to deeply understand data

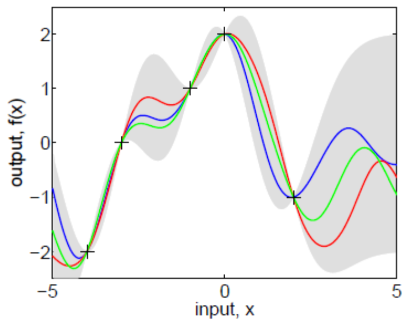
Typical sales pitch: Take your data spreadsheet …



... and apply data analytics

Latent Dirichlet Allocation

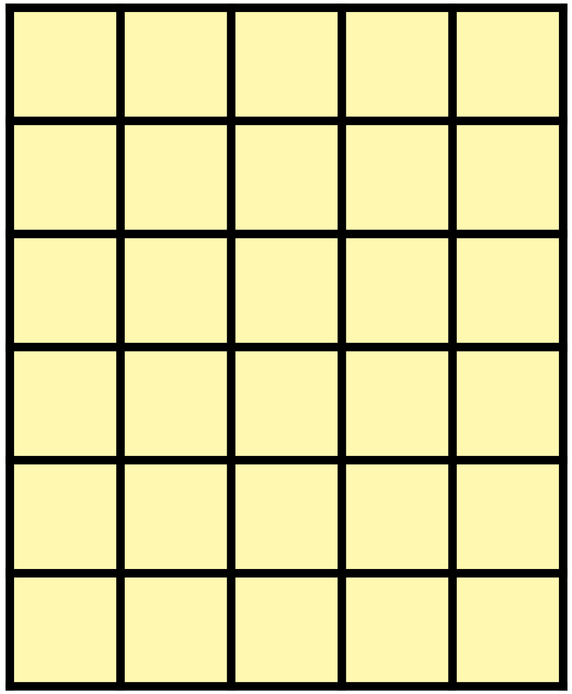




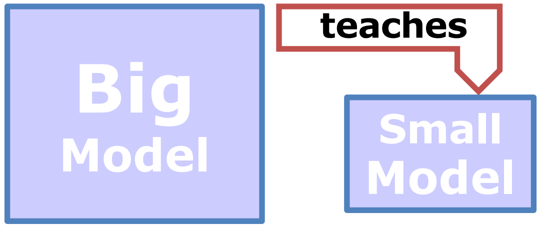
Gaussian Processes

Features

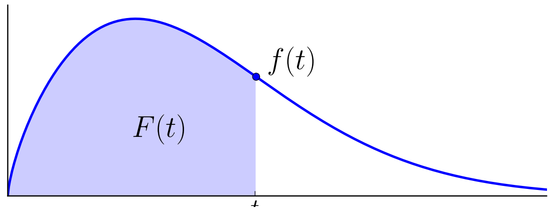
Objects



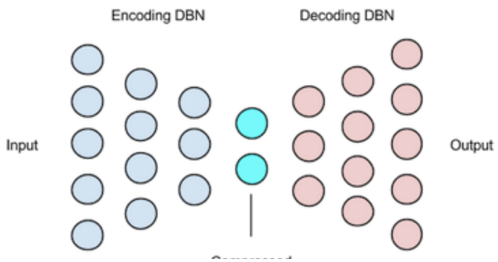
Boosting



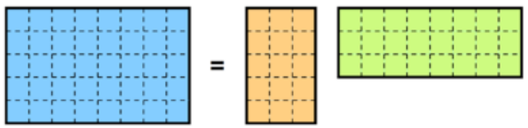
Distillation/LUPI



Diffusion Models

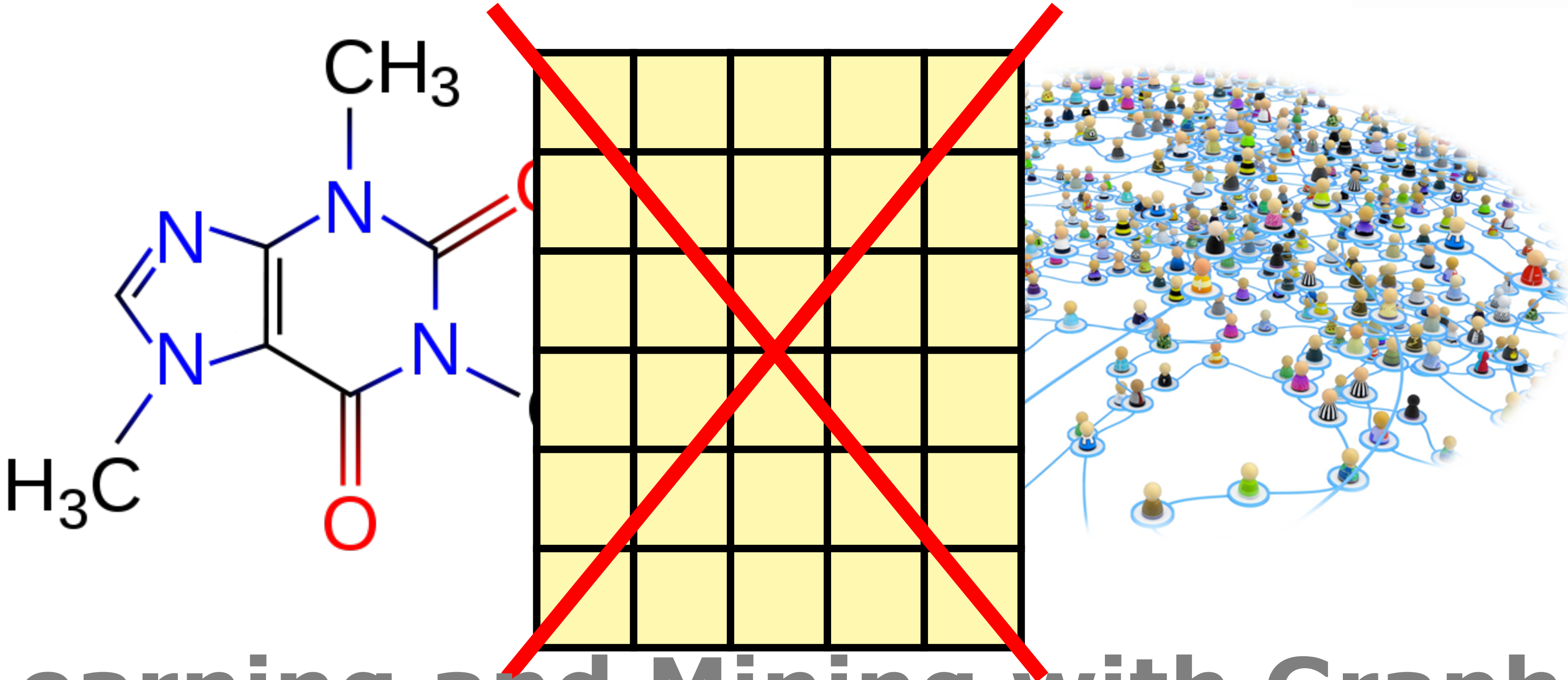


Autoencoder, Deep Learning



Big Data Matrix Factorization

and many more ...



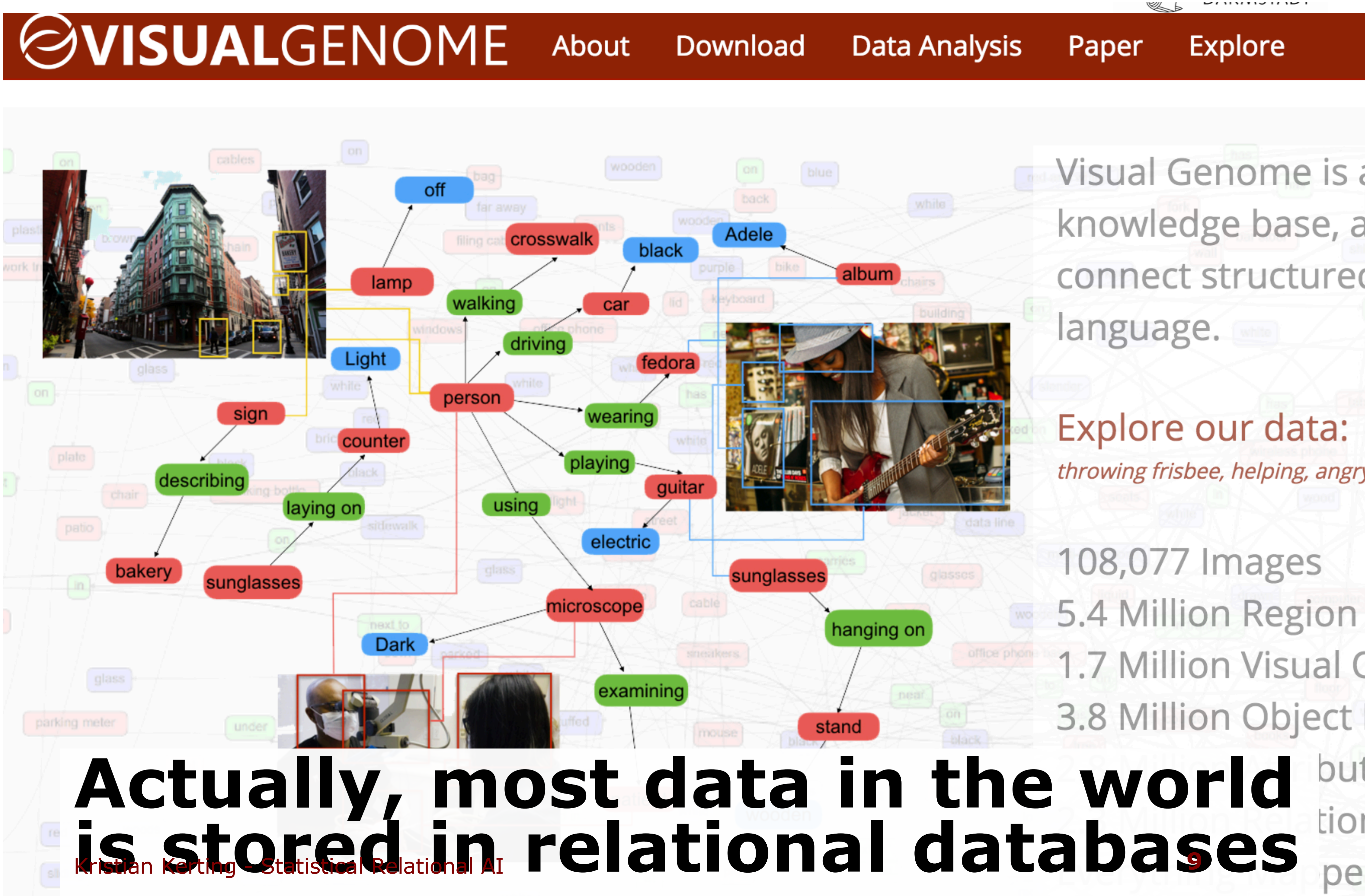
Learning and Mining with Graphs

[Haussler ’99, Gärtner, Flach, Wrobel COLT’03, Vishwanathan, Schraudolph, Kondor, Borgwardt JMLR’10, Shervashidze, Schweitzer, van Leeuwen, Mehlhorn, Borgwardt JMLR’11, Bauckhage, Kersting FaTWebScience´16, …]

IS IT AT ALL THAT SIMPLE?

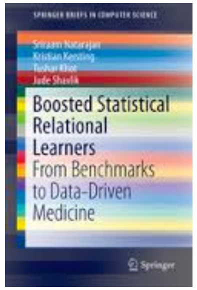
And heterogenous data graphs abound

[Lu, Krishna, Bernstein, Fei-Fei „Visual Relationship Detection“ CVPR 2016]



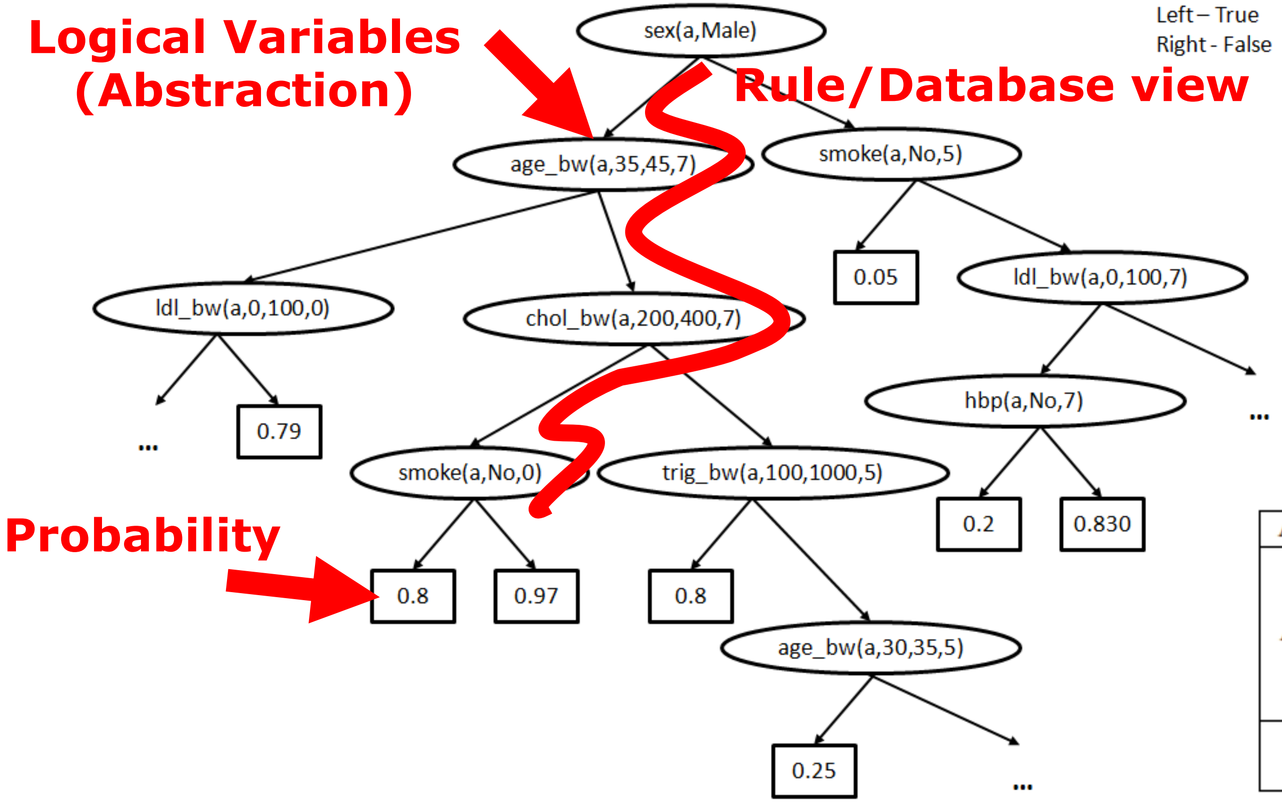
Actually, most data in the world is stored in relational databases

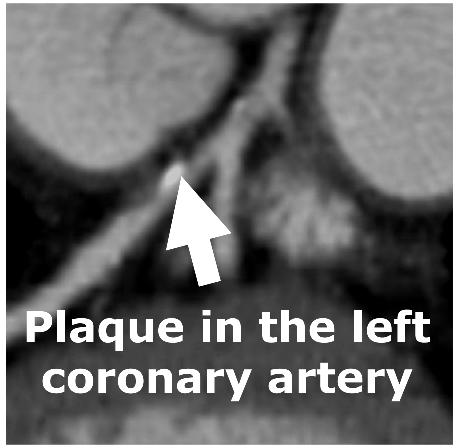
Natarajan, Khot, Kersting, Shavlik. Boosted Statistical Relational Learners. Springer Brief 2015



Mining Electronic Health Records

Atherosclerosis is the cause of the majority of Acute Myocardial Infarctions (heart attacks)





[Circulation; 92(8), 2157-62, 1995; JACC; 43, 842-7, 2004]

Algorithm	Accuracy	AUC-ROC
J48	0.667	0.607
SVM	0.667	0.5
AdaBoost	0.667	0.608
Bagging	0.677	0.613
NB	0.75	0.653
RPT	0.669*	0.778
RFGB	0.667*	0.819

The higher, the better

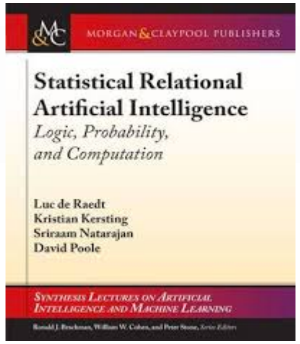
25%

Algorithm for Mining Markov Logic Networks	**Likelihood** The higher, the better	**AUC-ROC** The higher, the better	**AUC-PR** The higher, the better	**Time** The lower, the better
Boosting	0.81	0.96	0.93	**9s**
LSM	0.73	0.54	0.62	**93 hrs**

11% 78% 50% 37200x faster

State-of-the-art

De Raedt, Kersting, Natarajan, Poole, Statistical Relational Artificial Intelligence: Logic, Probability, and Computation. Morgan and Claypool Publishers, ISBN: 9781627058414, 2016.

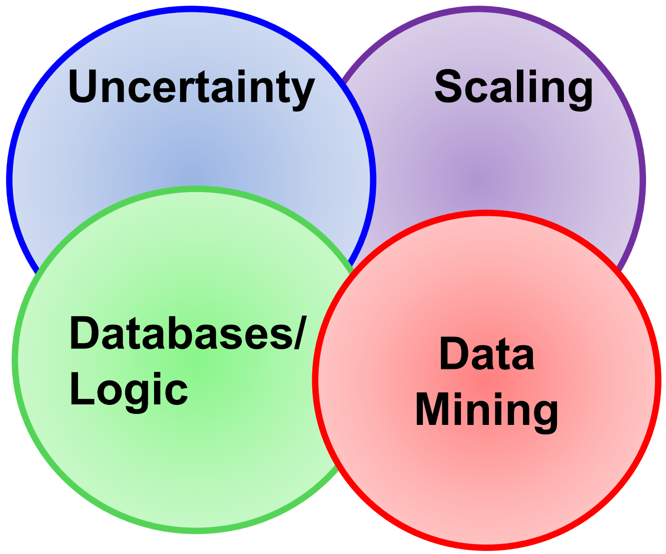


Punchline

Two trends driving modern data science and AI

1. Arms race to deeply understand data
2. Data in a large number of formats

Statistical AI and Data Science need **data& program abstractions**



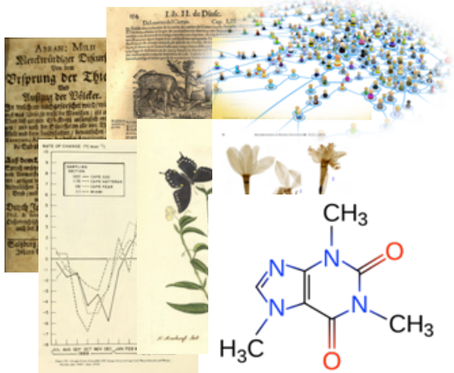
[Getoor, Taskar MIT Press ’07; De Raedt, Frasconi, Kersting, Muggleton, LNCS’08; Domingos, Lowd Morgan&Claypool ’09; Natarajan, Kersting, Khot, Shavlik Springer Brief’15; Russell CACM 58(7): 88-97 ’15]

[Ré, Sadeghian, Shan, Shin, Wang, Wu, Zhang IEEE Data Eng. Bull.'14; Natarajan, Picado, Khot, Kersting, Ré, Shavlik ILP'14; Natarajan, Soni, Wazalwar, Viswanathan, Kersting Solving Large Scale Learning Tasks'16, Mladenov, Heinrich, Kleinhans, Gonsior, Kersting DeLBP'16, …]

Declarative Data Science / AI

Feature Extraction → **Declarative Data Science/ AI Programming** → **Symbolic-Numerical Reasoning**

(Un-)Structured Data Sources



External Databases

Wikipedia The Free Encyclopedia — Freebase — GeoNames

Features and Rules

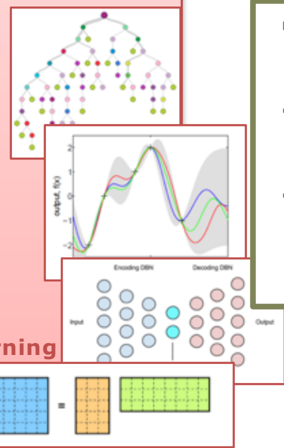
Weighted Data and Program Abstractions

Model Rules and DomainKnowledge

Representation Learning

DM, ML, and AI Algorithms

- Graph Kernels
- Diffusion Processes
- Random Walks
- Decision Trees
- Frequent Itemsets
- SVMs
- Graphical Models
- Topic Models
- Gaussian Processes
- Autoencoder
- Matrix and Tensor Factorization
- Reinforcement Learning
- …



Features and Data Rules

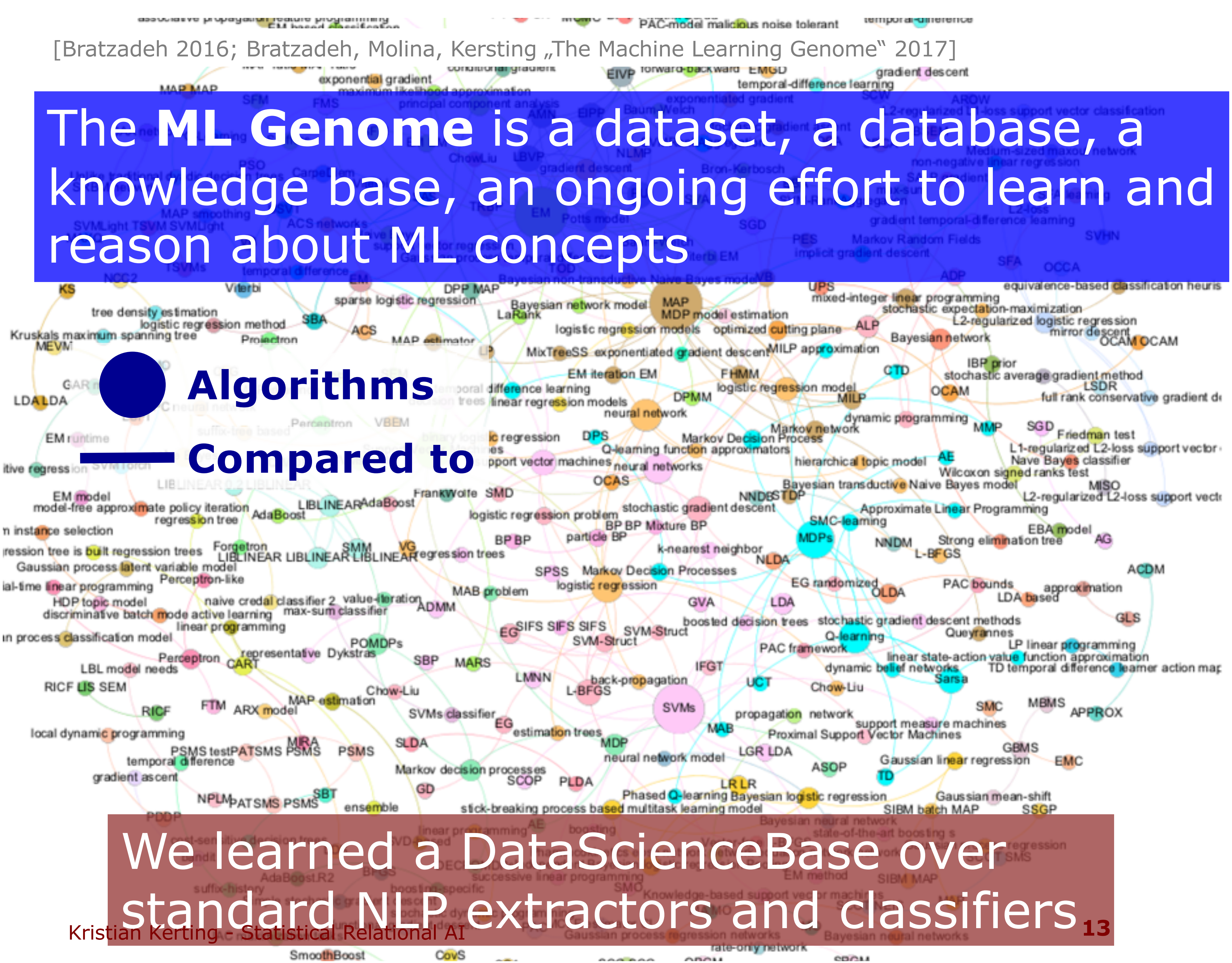
Inference Results

	p
	0.9
	0.6

Feedback/AutoDM

[Bratzadeh 2016; Bratzadeh, Molina, Kersting „The Machine Learning Genome" 2017]

The **ML Genome** is a dataset, a database, a knowledge base, an ongoing effort to learn and reason about ML concepts



Algorithms

Compared to

We learned a DataScienceBase over standard NLP extractors and classifiers

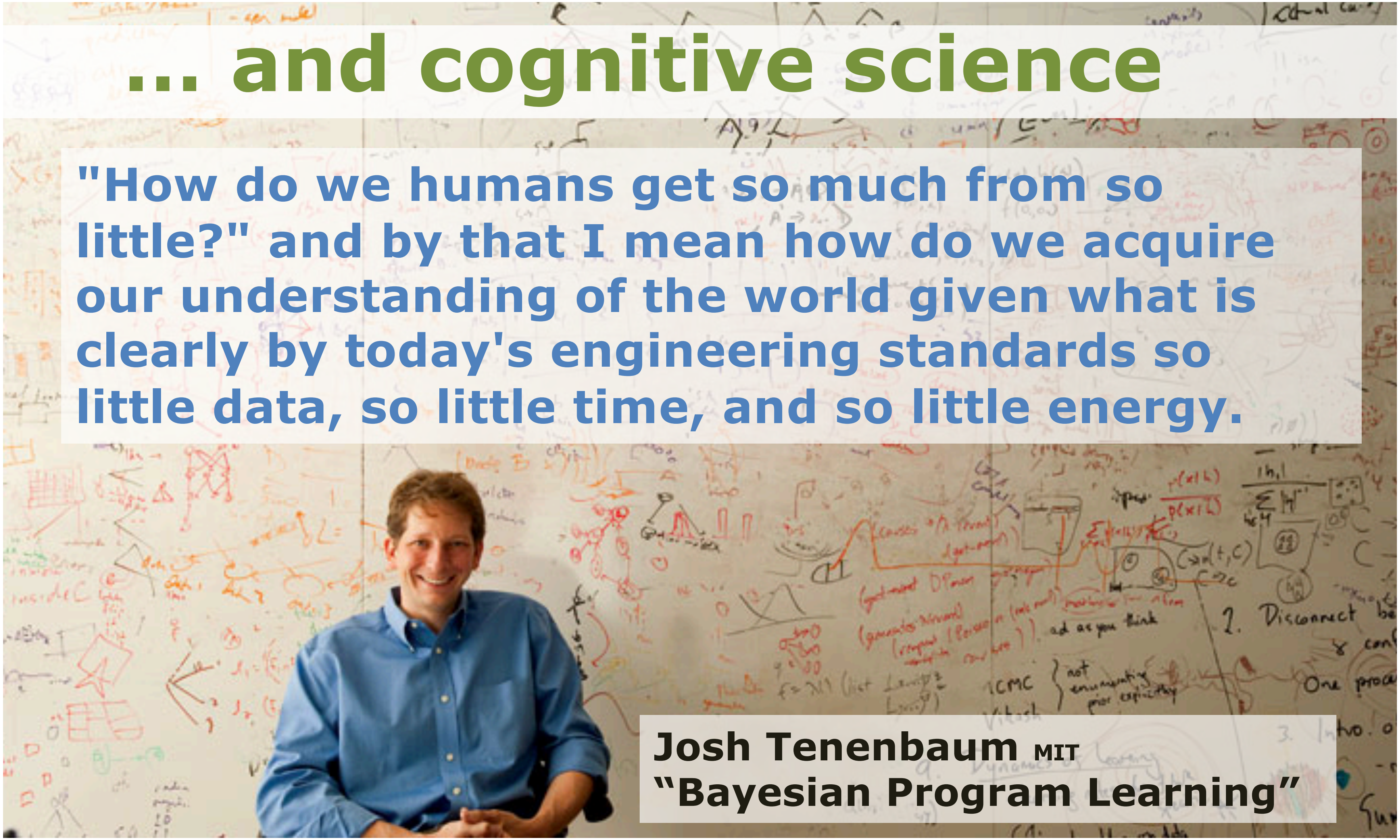
This connects well to databases

Jim Gray Turing Award 1998
“Automated Programming”

Mike Stonebraker Turing Award 2014
“One size does not fit all”

… and cognitive science

"How do we humans get so much from so little?" and by that I mean how do we acquire our understanding of the world given what is clearly by today's engineering standards so little data, so little time, and so little energy.

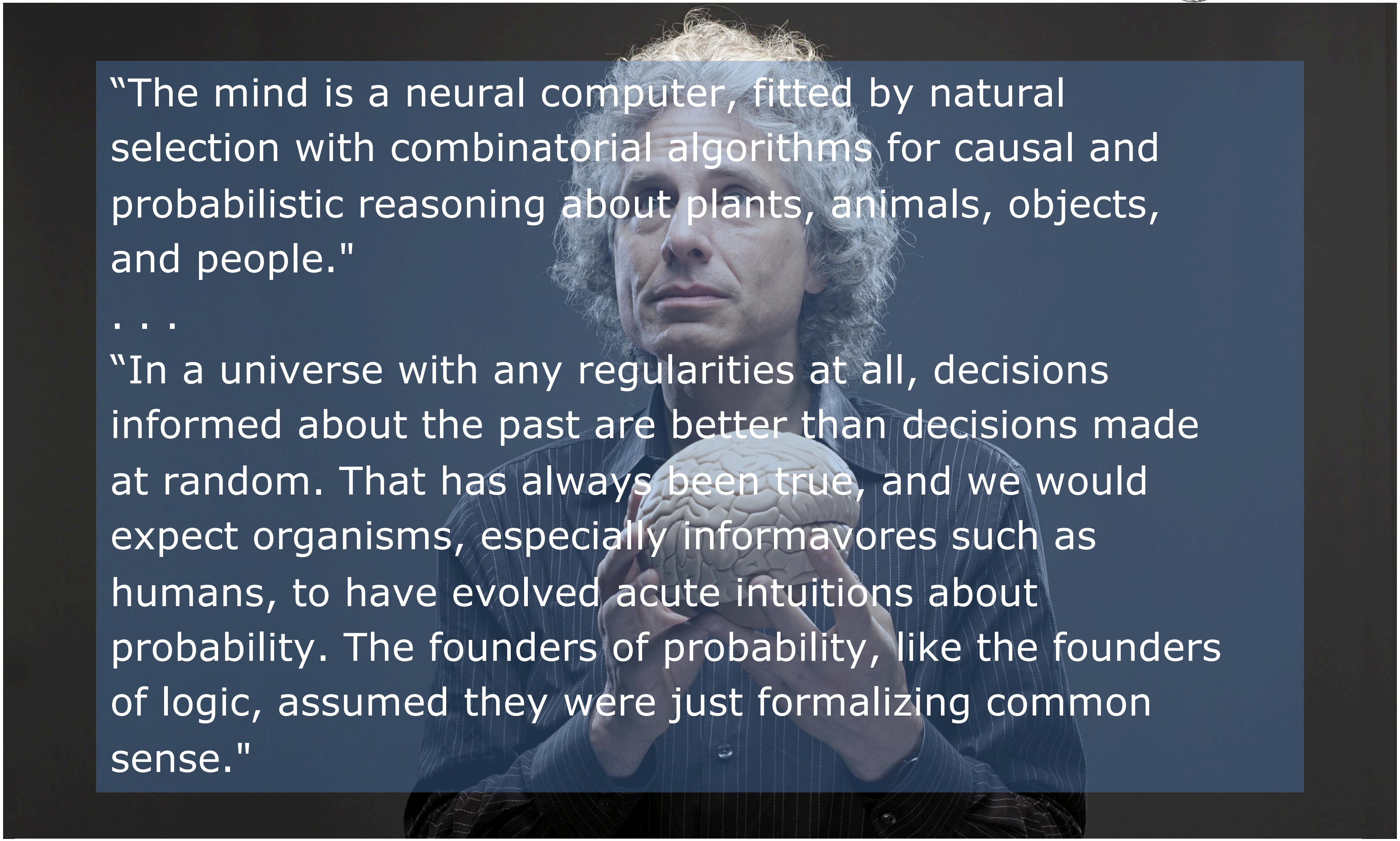


Josh Tenenbaum MIT
"Bayesian Program Learning"

Lake, Salakhutdinov, Tenenbaum, Science 350 (6266), 1332-1338, 2015
Tenenbaum, Kemp, Griffiths, Goodman, Science 331 (6022), 1279-1285, 2011

And this is also deep!

TECHNISCHE UNIVERSITÄT DARMSTADT



"The mind is a neural computer, fitted by natural selection with combinatorial algorithms for causal and probabilistic reasoning about plants, animals, objects, and people."

. . .

"In a universe with any regularities at all, decisions informed about the past are better than decisions made at random. That has always been true, and we would expect organisms, especially informavores such as humans, to have evolved acute intuitions about probability. The founders of probability, like the founders of logic, assumed they were just formalizing common sense."

-Steven Pinker, How the Mind Works, 1997, pp. 524, 343.

LET'S CONSIDER SOME MORE CONCRETE MOTIVATING EXAMPLES

Bayes' Rule

$$P(h|e) = \frac{\overbrace{P(e|h)}^{\text{Likelihood}}\;\overbrace{P(h)}^{\text{Prior}}}{\underbrace{P(e)}_{\text{Normalizing constant}}}$$

- What if h is the effect of a drug on a particular patient, and e is the patient's electronic health record?
- What if e is the electronic health records for all of the people in the world?
- What if e is a collection of student records in a university?
- What if e is a description of everything known about the geology of Earth?

Example: Predicting Relations

- Students s3 and s4 have the same averages, on courses with the same averages.
- Which student would you expect to do better?

Student	Course	Grade
s_1	c_1	A
s_2	c_1	C
s_1	c_2	B
s_2	c_3	B
s_3	c_2	B
s_4	c_3	B
s_3	c_4	?
s_4	c_4	?

Example: BN for Predicting Grades

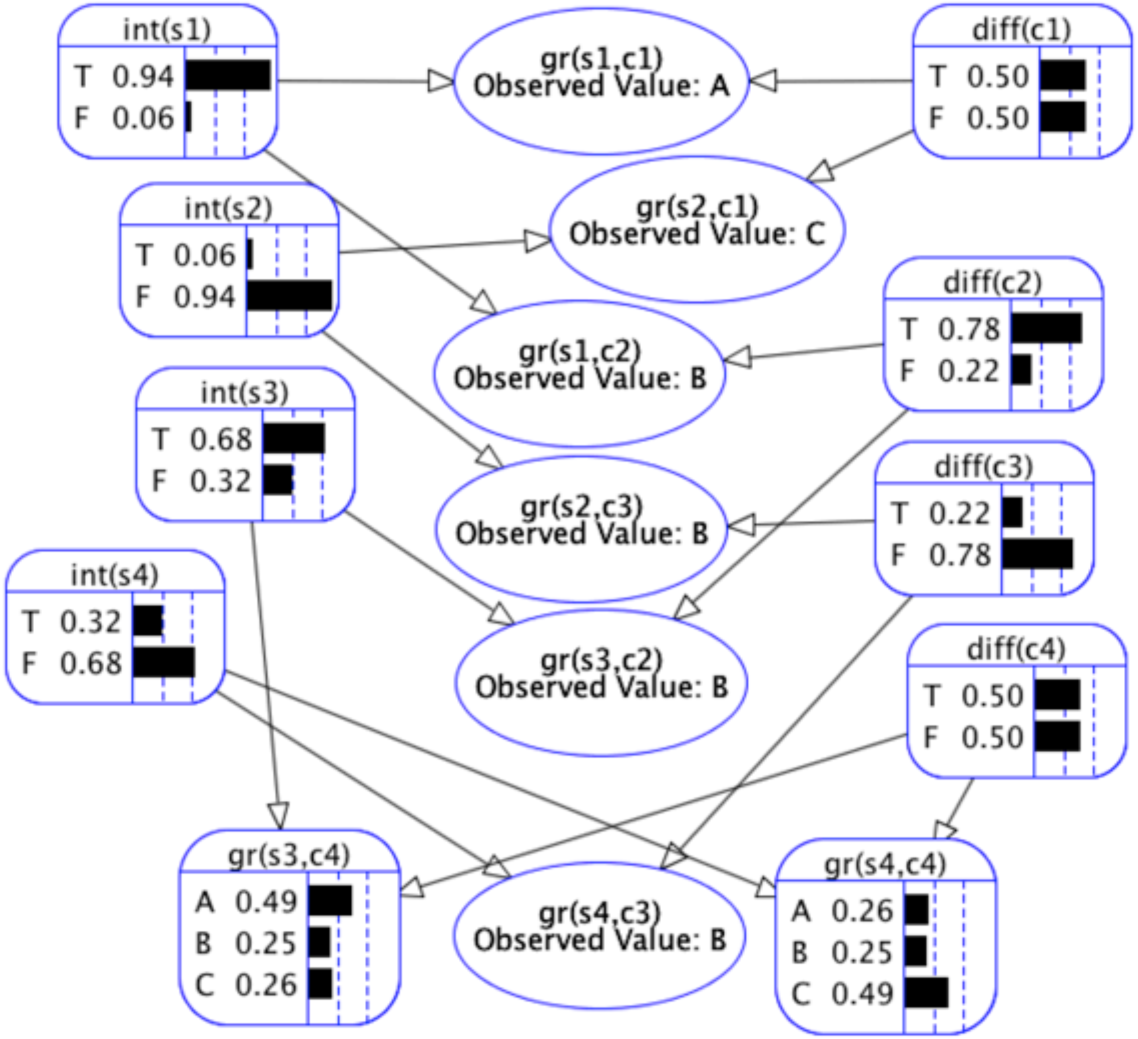
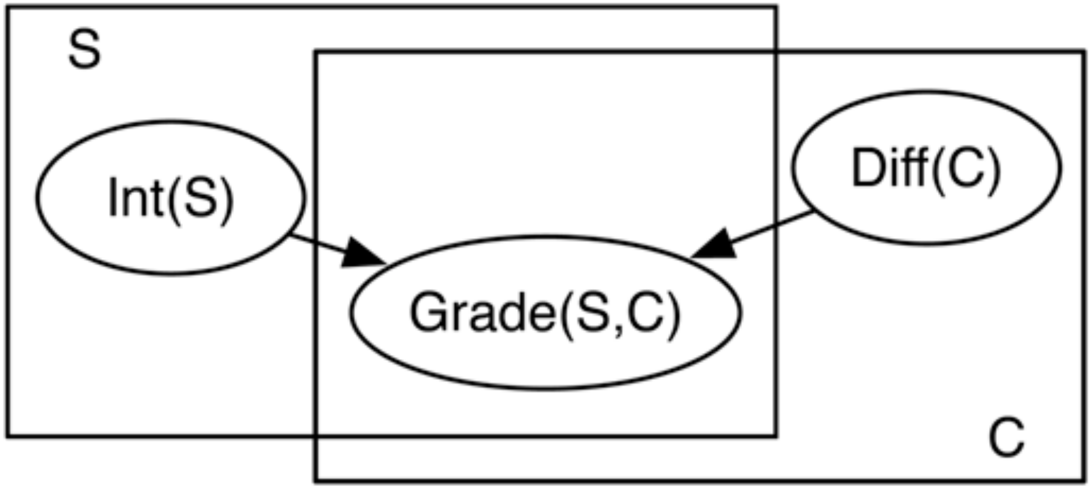


Plate Notation



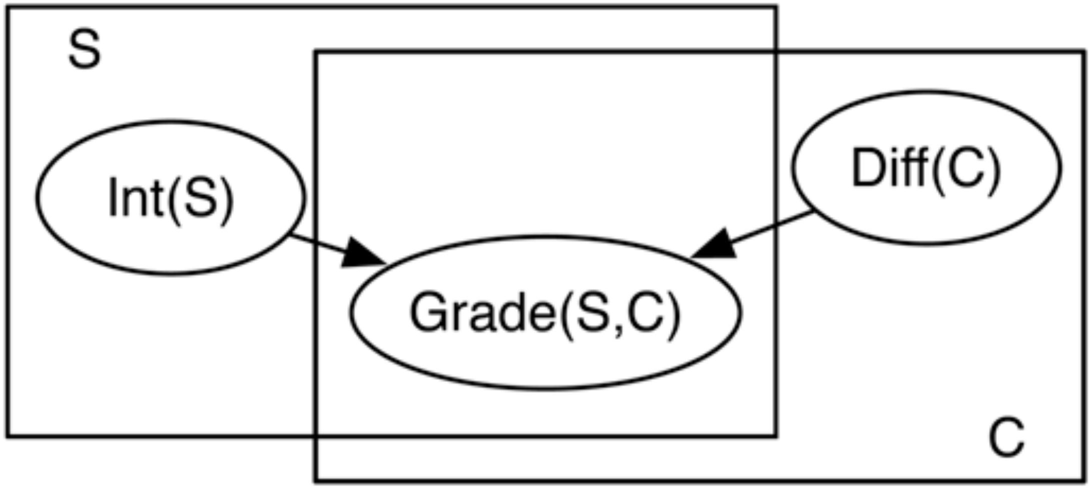
"Program" Abstraction:

- S, C logical variable representing students, courses
- the set of individuals of a type is called a population
- Int(S), Grade(S, C), D(C) are parametrized random variables

Grounding:

- for every student s, there is a random variable Int(s)
- for every course c, there is a random variable Di(c)
- for every s, c pair there is a random variable Grade(s,c)
- all instances share the same structure and parameters

Plate Notation



- If there were 1000 students and 100 courses:
- Grounding contains
  - 1000 I(s) variables
  - 100 D(c) variables
  - 100000 Gr(s,c) variables
- **total: 101100 variables**

- **Numbers to be specified to define the probabilities: 1 for I (S), 1 for D(C), 8 for Gr(S,C) = 10 parameters.**

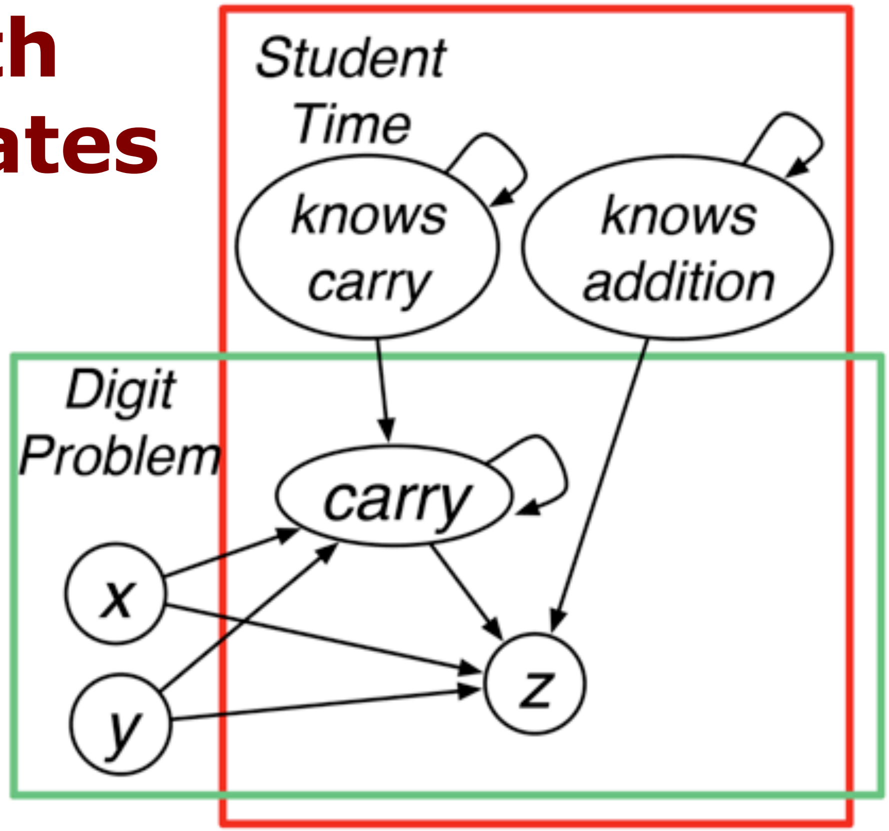
Bayesian Belief Propagation

$$
\begin{array}{rccc}
 & & x_2 & x_1 \\
+ & & y_2 & y_1 \\
\hline
 & z_3 & z_2 & z_1
\end{array}
$$

- What if there were multiple digits, problems, students, times?
- How can we build a model before we know the individuals?

Multi-digit addition with parametrized BNs / plates

$$
\begin{array}{rcccc}
 & x_{j_x} & \cdots & x_2 & x_1 \\
+ & y_{j_z} & \cdots & y_2 & y_1 \\
\hline
 & z_{j_z} & \cdots & z_2 & z_1
\end{array}
$$



Random Variables: x(D,P), y(D,P), knowsCarry(S,T), knowsAddition(S,T), carry (D,P,S,T), z(D,P,S,T) for each:

digit D, problem P, student S, time T

Relational Probabilistic Models

Random variables for combinations of individuals in populations

- build a probabilistic model before knowing (all of) the individuals
- learn the model for one set of individuals
- apply the model to existing and new individuals
- allow complex relationships between individuals

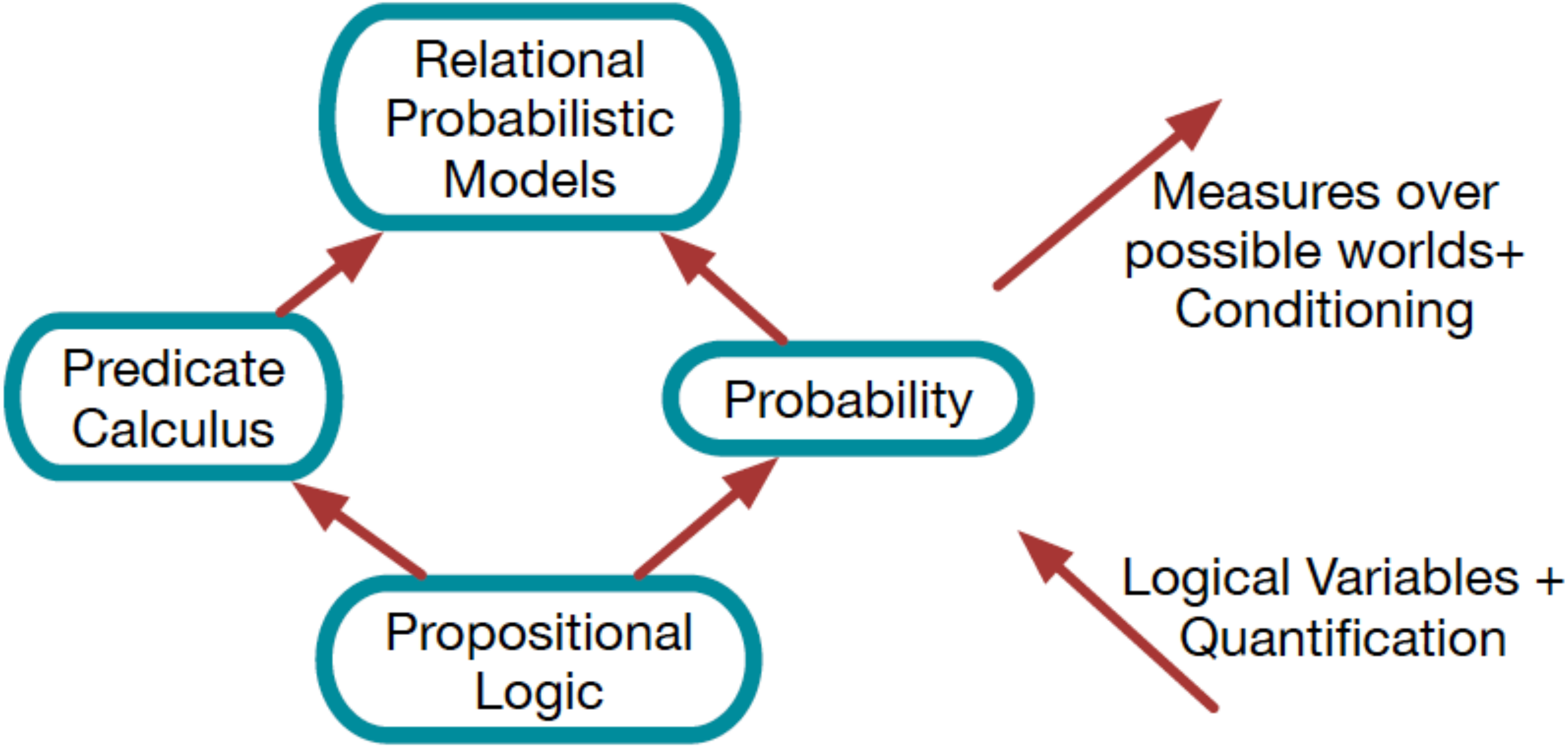
Exchangeability:

- Before we know anything about individuals, they are indistinguishable, and so should be treated identically.

Uncertainty about:

- Properties of individuals
- Relationships among individuals
- Identity (equality) of individuals
- Existence (and number) if individuals

Statistical Relational Artificial Intelligence



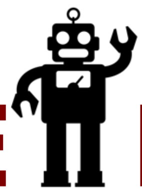
Declarative DS and AI ask: How can machines, like we humans, get so much from so little?

Let’s make the data science/AI experts more effective

Let’s increases the number of people who can successfully build data science/AI machines

Let’s make the data science/AI machines think

However, declarative DS/AI may also enlarge the underlying model, making solving it potentially very slow

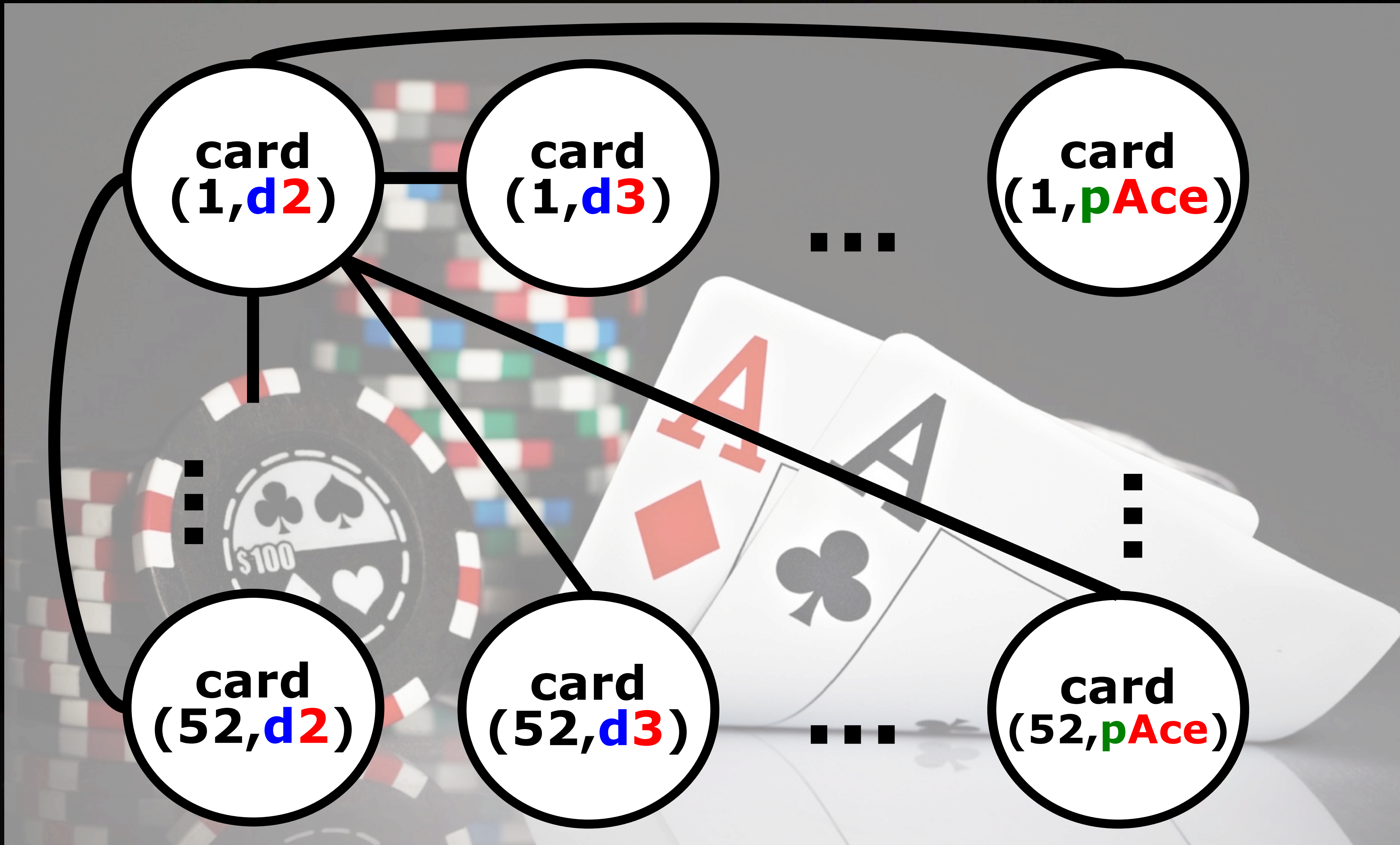


CAN THE MACHINE HELP TO REDUCE THE MINING COSTS?

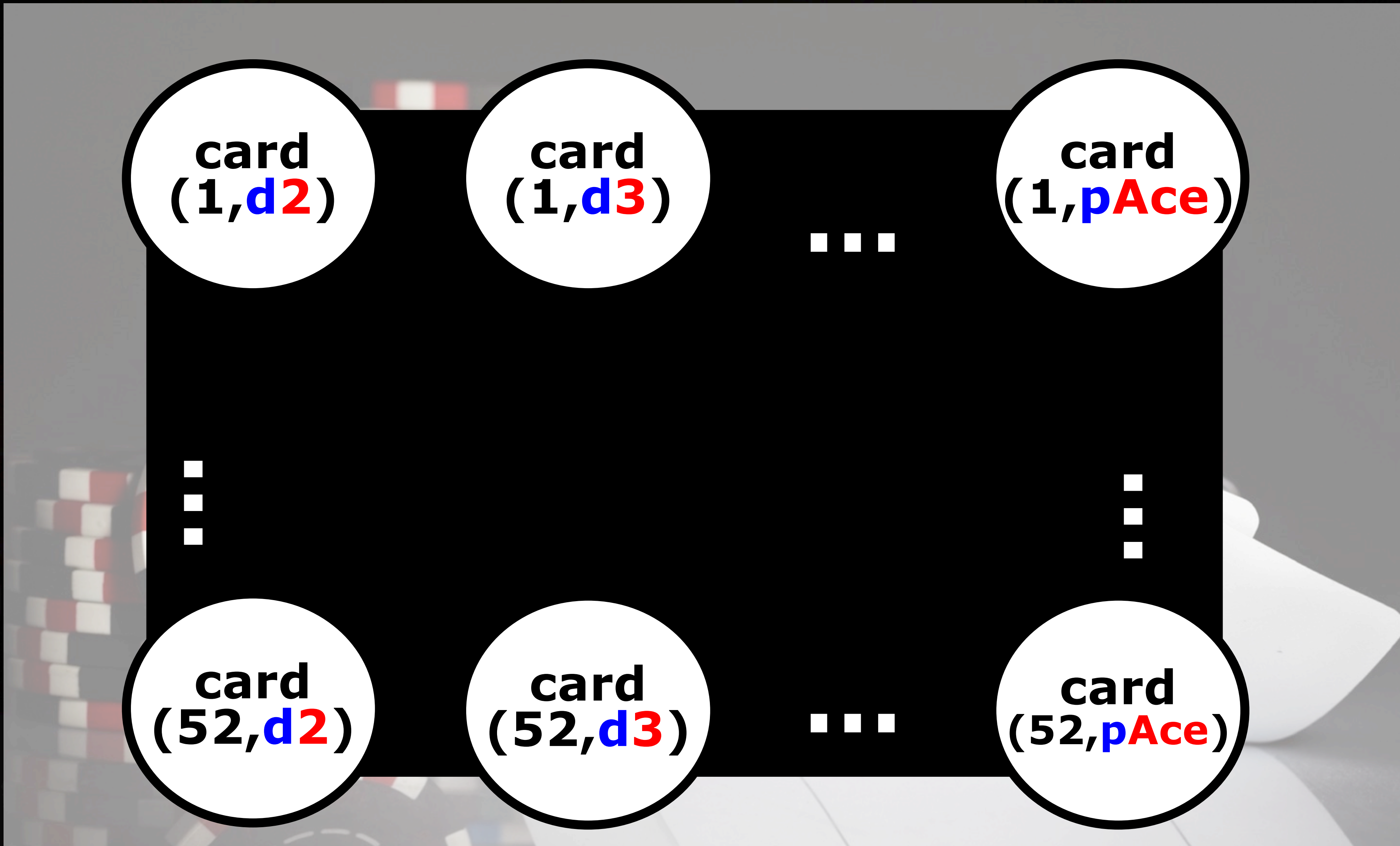
A simple graphical model example

What is the probability that the first card of a randomly shuffled deck with 52 cards is an Ace?

A simple graphical model example



A simple graphical model example



A simple graphical model example

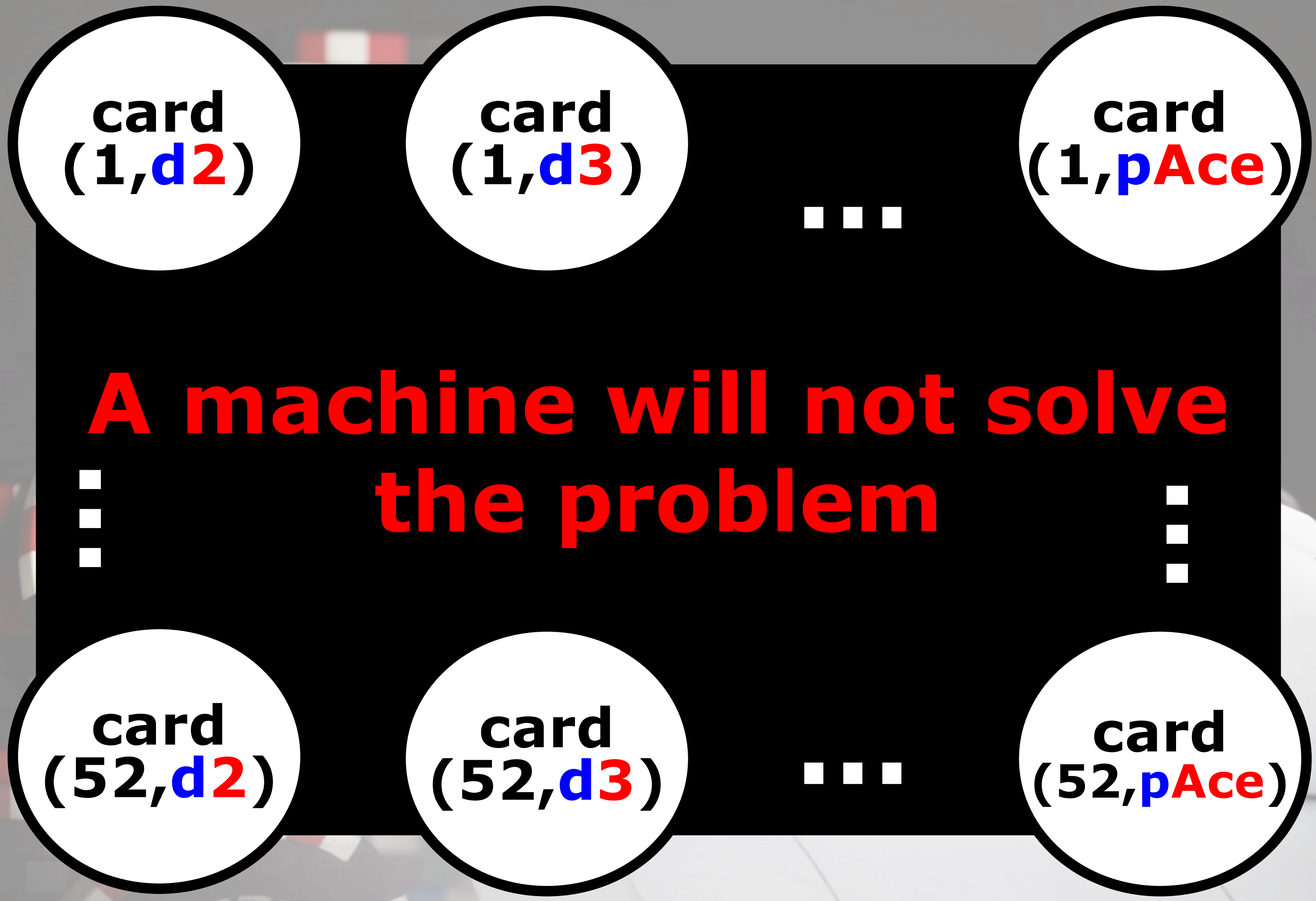
card (1,d2) | card (1,d3) | … | card (1,pAce)

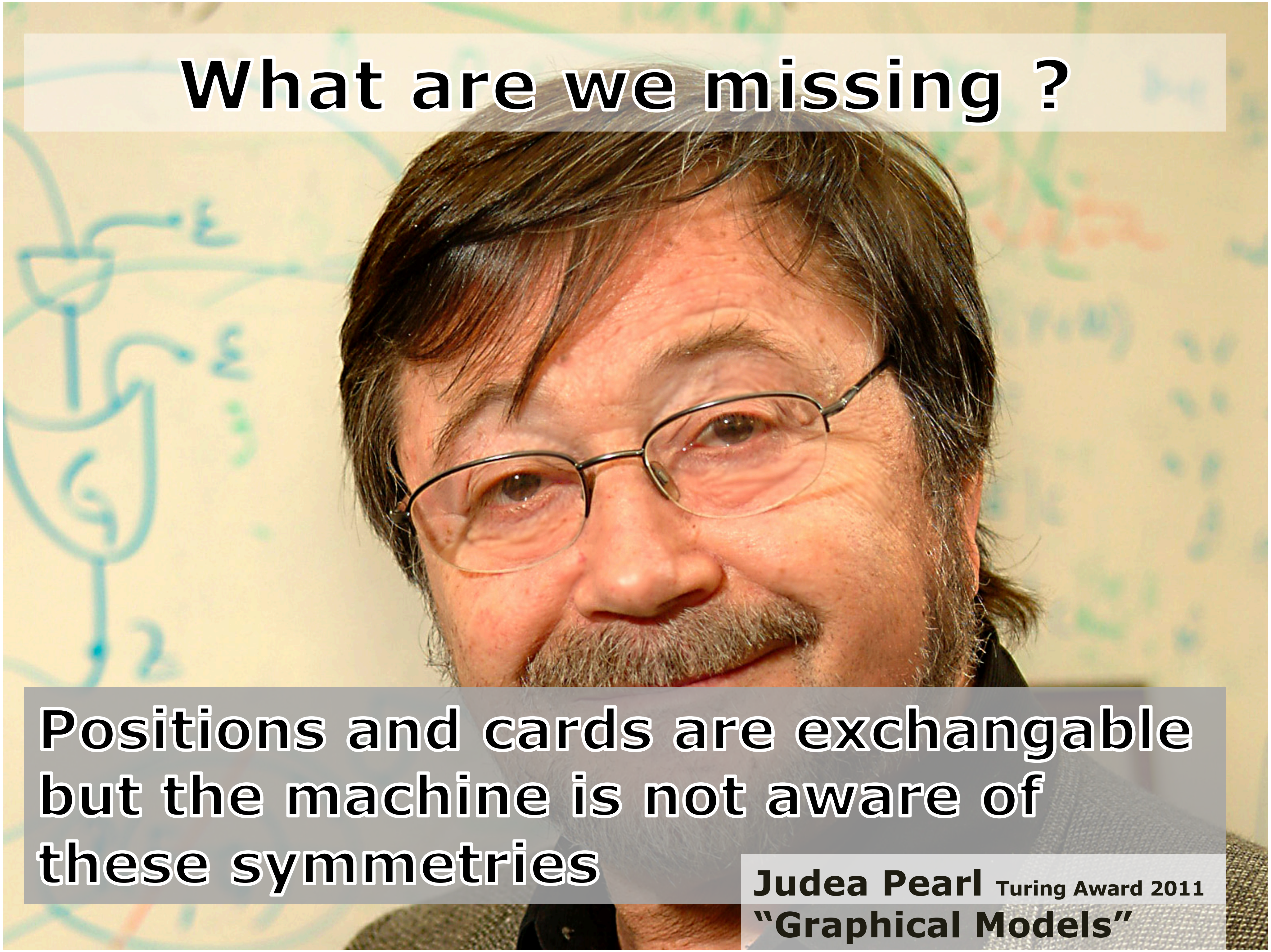
⋮ | | ⋮

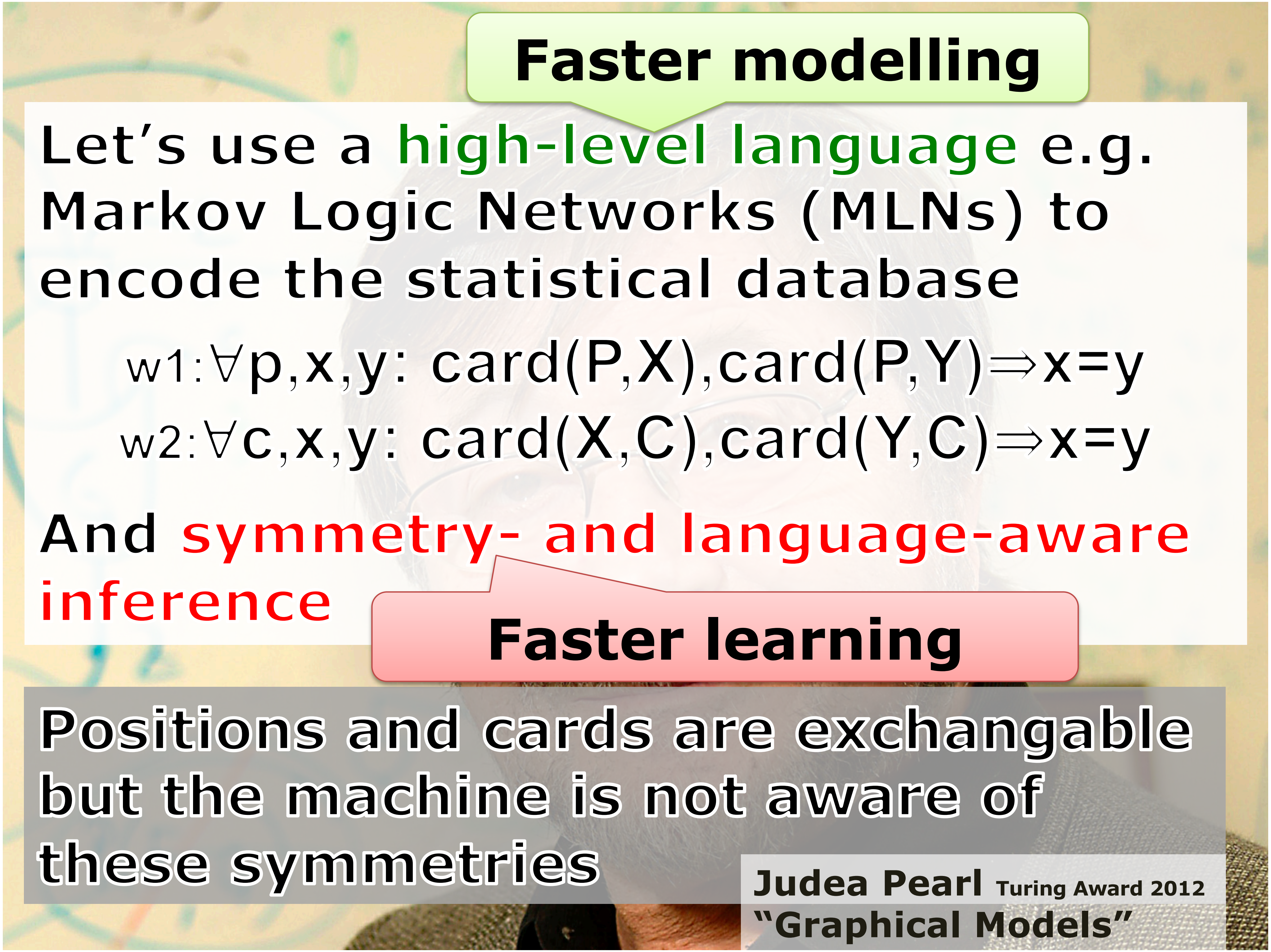
card (52,d2) | card (52,d3) | … | card (52,pAce)

No independencies.
Fully connected.
2^{2704} states

A simple graphical model example




What are we missing ?
Positions and cards are exchangable but the machine is not aware of these symmetries
Judea Pearl Turing Award 2011
"Graphical Models"



Faster modelling
Let's use a high-level language e.g. Markov Logic Networks (MLNs) to encode the statistical database
w1: ∀p,x,y: card(P,X),card(P,Y)⇒x=y
w2: ∀c,x,y: card(X,C),card(Y,C)⇒x=y
And symmetry- and language-aware inference
Faster learning
Positions and cards are exchangable but the machine is not aware of these symmetries
Judea Pearl Turing Award 2012
"Graphical Models"

[Singla, Domingos AAAI'08; Kersting, Ahmadi, Natarajan UAI'09; Ahmadi, Kersting, Mladenov, Natarajan MLJ'13]

Lifted Loopy Belief Propagation

Exploiting computational symmetries

TECHNISCHE UNIVERSITÄT DARMSTADT

Big Model

automatically compressed

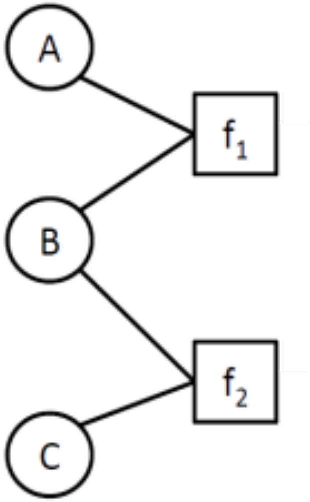
Small Model

Run Loopy Belief Propagation

Run a modified Loopy Belief Propagation

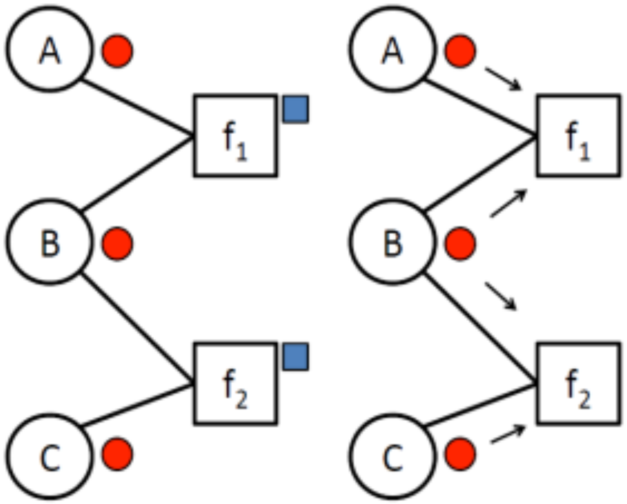
What are symmetries in approximate probabilistic inference, one of the working horses of data science?

Compression: Coloring the graph



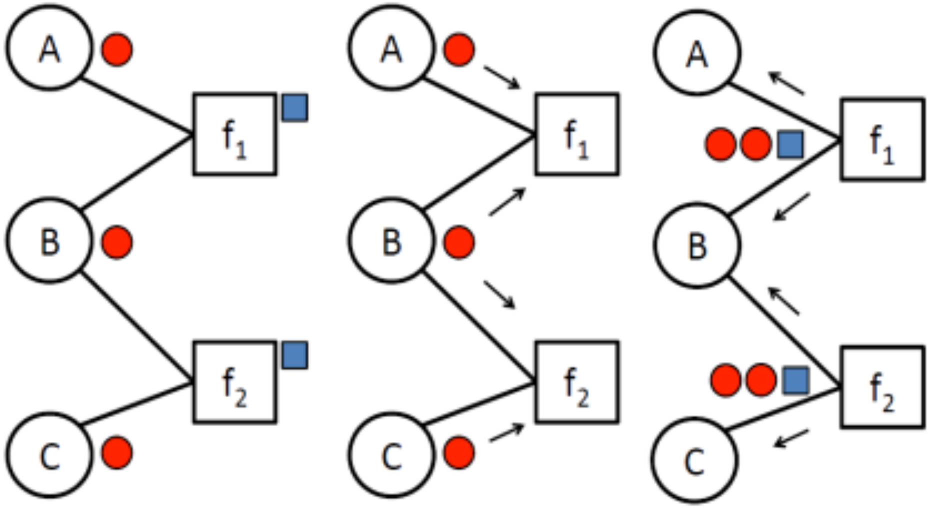
- **Color nodes according to the evidence you have**
  - No evidence, say **red**
  - State „one", say **brown**
  - State „two", say **orange**
  - ...
- **Color factors distinctively according to their equivalences** For instance, assuming f_1 and f_2 to be identical and B appears at the second position within both, say **blue**

Compression: Pass the colors around



1. Each factor collects the colors of its neighboring nodes

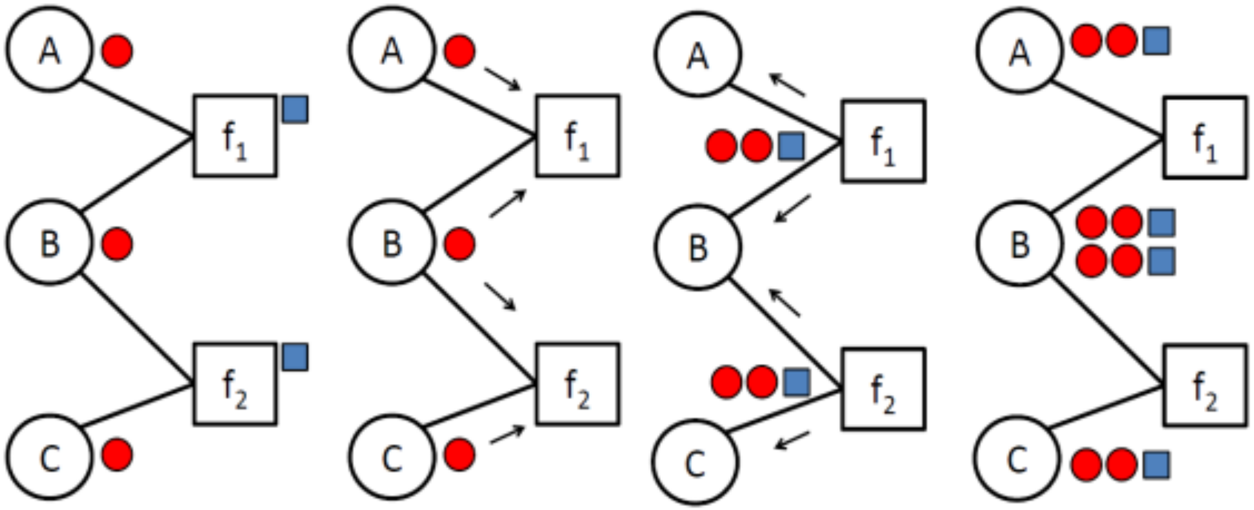
Compression: Pass the colors around



1. Each factor collects the colors of its neighboring nodes
2. Each factor „signs" its color signature with its own color

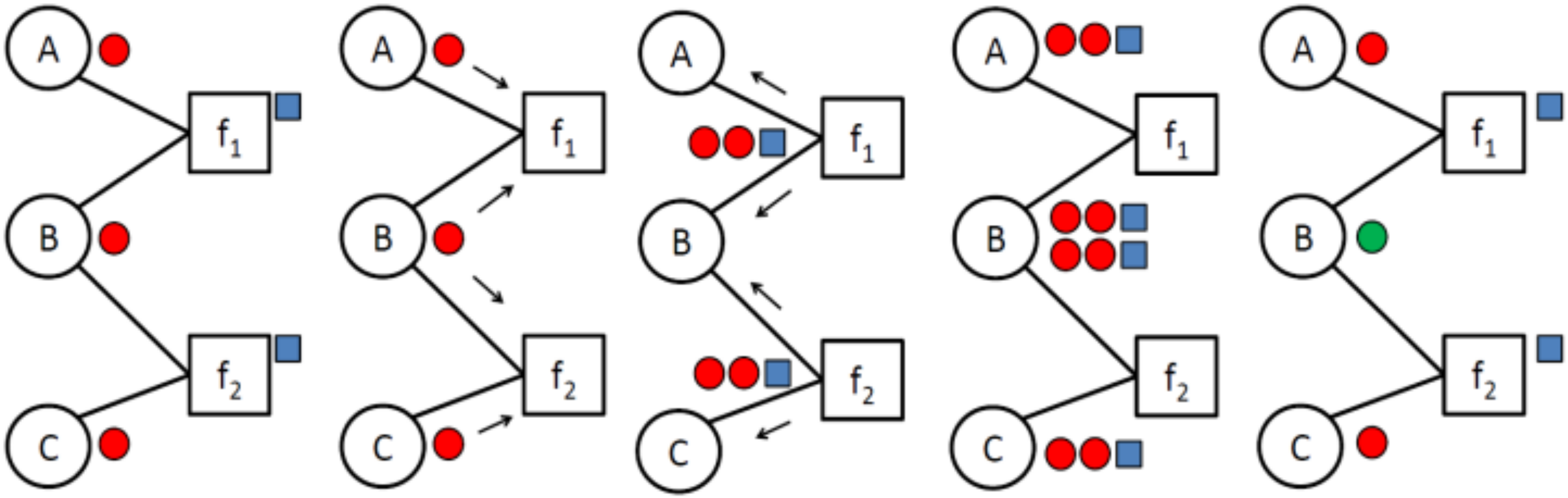
Compression: Pass the colors around

TECHNISCHE UNIVERSITÄT DARMSTADT



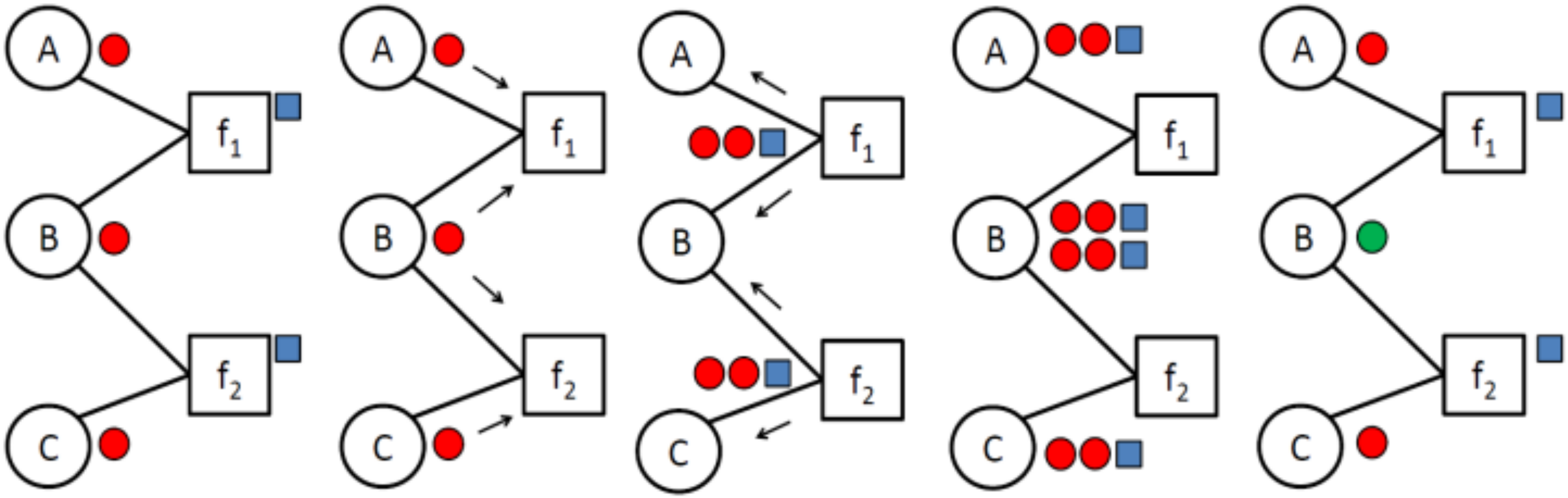
1. Each factor collects the colors of its neighboring nodes
2. Each factor „signs" its color signature with its own color
3. Each node collects the signatures of its neighboring factors

Compression: Pass the colors around



1. Each factor collects the colors of its neighboring nodes
2. Each factor „signs" its color signature with its own color
3. Each node collects the signatures of its neighboring factors
4. Nodes are recolored according to the collected signatures

Compression: Pass the colors around

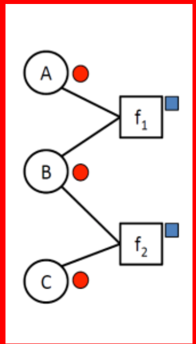


1. Each factor collects the colors of its neighboring nodes
2. Each factor „signs" its color signature with its own color
3. Each node collects the signatures of its neighboring factors
4. Nodes are recolored according to the collected signatures
5. If no new color is created stop, otherwise go back to 1

Lifted Loopy Belief Propagation

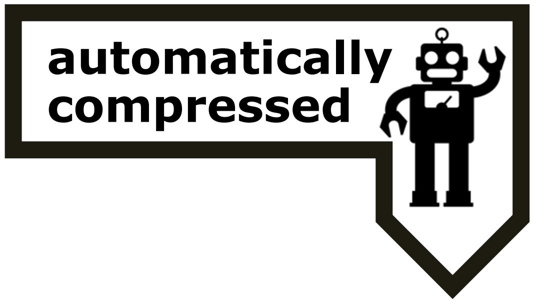
Exploiting computational symmetries

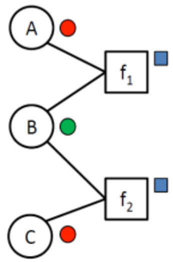
[Singla, Domingos AAAI’08; Kersting, Ahmadi, Natarajan UAI’09; Ahmadi, Kersting, Mladenov, Natarajan MLJ’13]



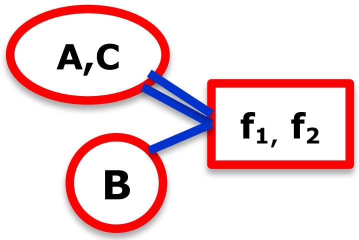
Big Model

quasi-linear time





Small Model



Run Loopy Belief Propagation

Run a modified Loopy Belief Propagation

Compression can considerably speed up inference and training

[Singla, Domingos AAAI'08; Kersting, Ahmadi, Natarajan UAI'09; Ahmadi, Kersting, Mladenov, Natarajan MLJ'13]

Probabilistic inference using lifted (loopy) belief propagation

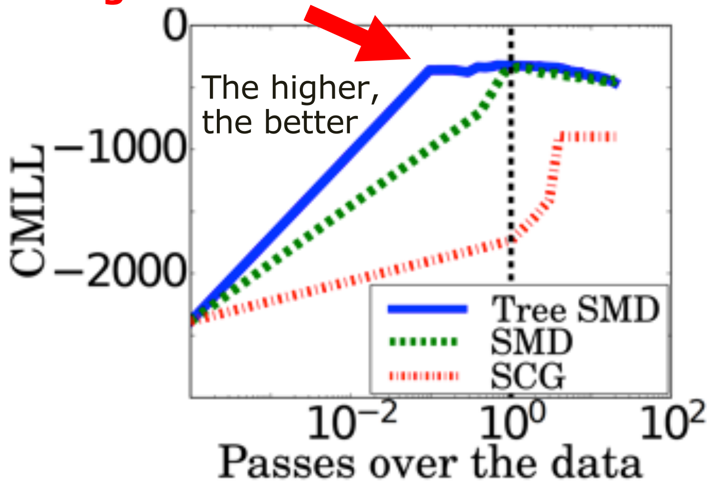
Domain	Time (in seconds) The lower, the better						No. of (Super) Features	
	Construction		BP		Total			
	Ground	Lifted	Ground	Lifted	Ground	Lifted	Ground	Lifted
Cora	263.1	1173.3	12368.4	3997.7	12631.6	5171.1	2078629	295468
UW-CSE	6.9	22.1	1015.8	602.5	1022.8	624.7	217665	86459
Friends & Smokers	38.8	89.7	10702.2	4.4	10741.0	94.2	1900905	58

114x faster

Parameter training using a lifted stochastic gradient

CORA entity resolution

converges before data has been seen once



State-of-the-art

What is going on algebraically?

Can we generalize this to other data science approaches?

Instead of looking at DM through the glasses of probabilities, let‘s approach it now using optimization

WL computes a fractional automorphism of some matrix A

$$\mathbf{X}_{Q}\mathbf{A} = \mathbf{A}\mathbf{X}_{P}$$

where X_Q and X_p are doubly-stochastic matrixes (relaxed form of automorphism)

It turns out that color passing is well-known in graph theory:

The Weisfeiler-Lehman Algorithm

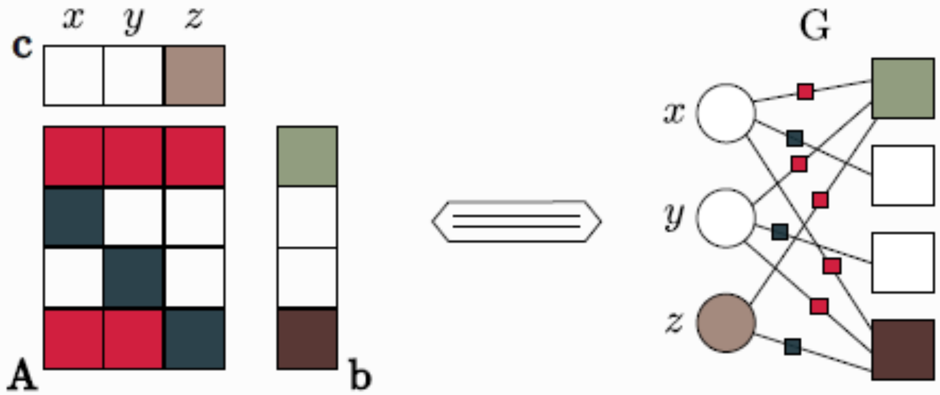
Graphen, Physic Nobel Prize 2010

TECHNISCHE UNIVERSITÄT DARMSTADT

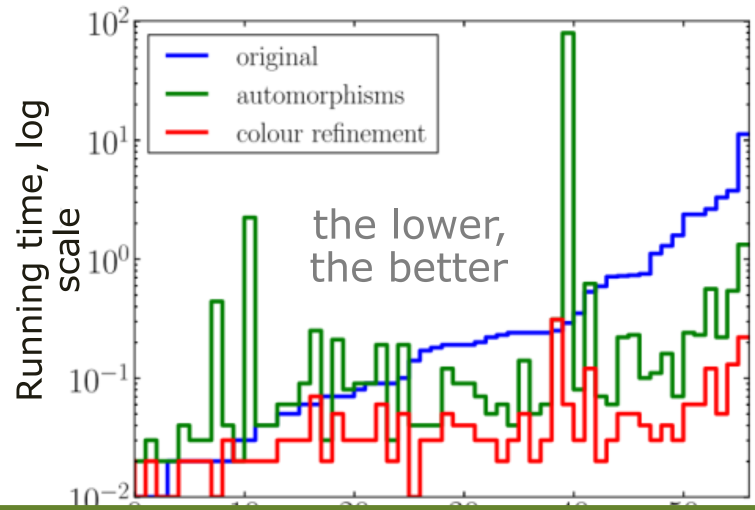
Lifted Linear Programming

$$\max_{[x,y,z]^T\in\mathbb{R}^3} \quad 0x + 0y + 1z$$

$$\text{s.t.} \quad \begin{bmatrix} 1 & 1 & 1 \\ -1 & 0 & 0 \\ 0 & -1 & 0 \\ 1 & 1 & -1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} \leq \begin{bmatrix} 1 \\ 0 \\ 0 \\ -1 \end{bmatrix}$$

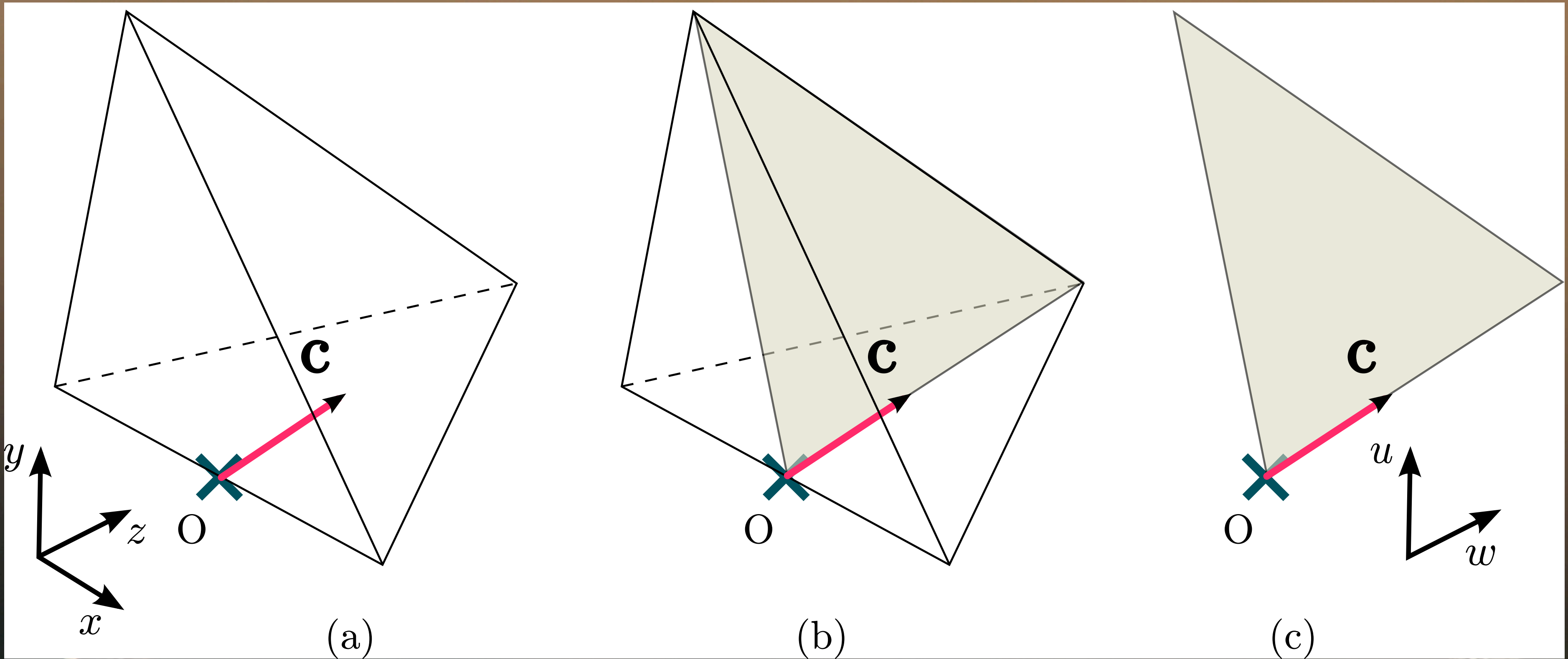


(1) Reduce the LP by running WL on the LP-Graph
(2) Run any solver on the (hopefully) smaller LP



quasi-linear overhead that may result in exponential speed up

State-of-the-art



Feasible region of LP and the objective vectors

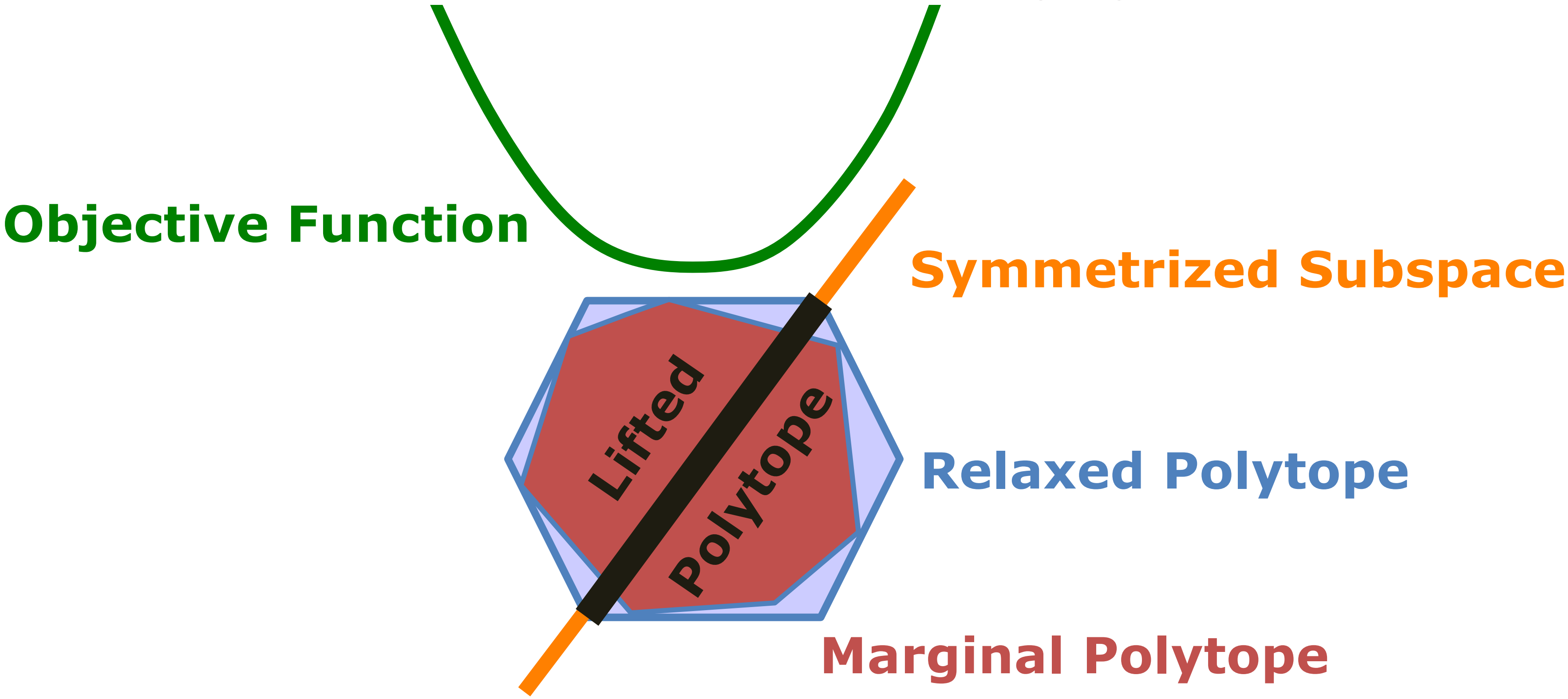
Span of the fractional auto-morphism of the LP

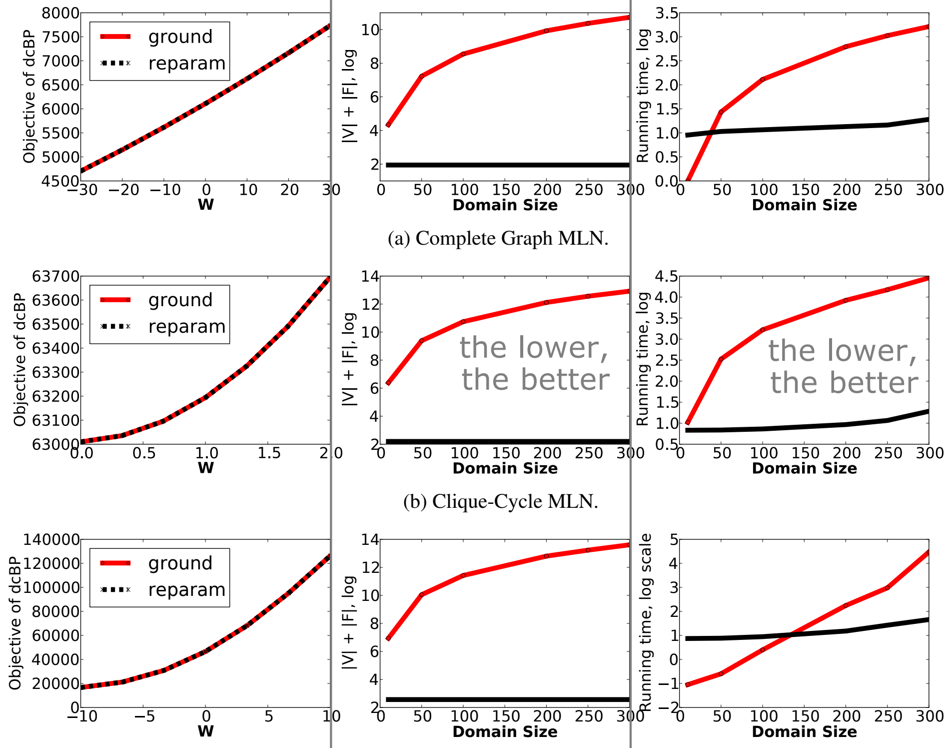
Projections of the feasible region onto the span of the fractional auto-morphism

Why does this work?

Approximate inference in graphical models is closely connected to linear programs

$$\hat{\mathbf{x}} \in \arg\max_{\mathbf{x}\in\mathcal{X}^N} \left\{ \sum_{s\in V} \theta_s(x_s) + \sum_{(s,t)\in E} \theta_{st}(x_s, x_t) \right\}$$





(a) Complete Graph MLN.

(b) Clique-Cycle MLN.

(c) Friends-smokers MLN.

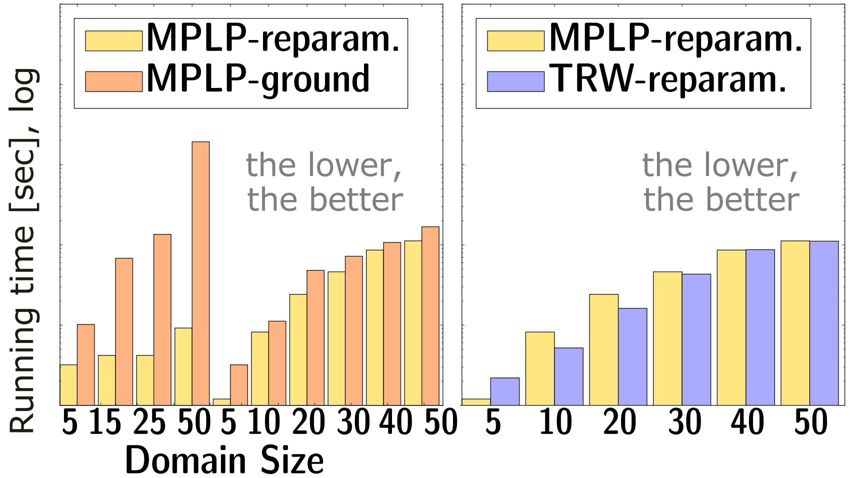
Marginals

Any concave free energy is liftable

First lifted distributed message-passing approach based on e.g. [Schwing et al. CVPR 2014]

MAP

Any MAP-LP message-passing approach is liftable

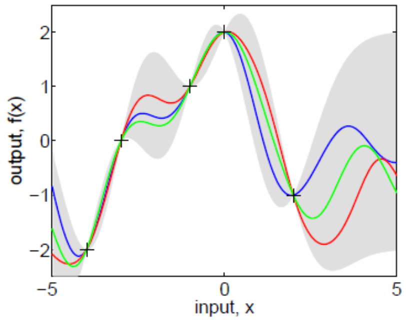


State of the art, which can also speed up training

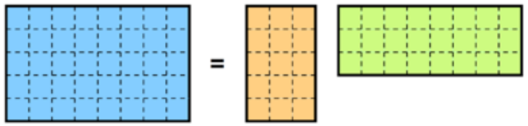
[Boyd, Diaconis, Parrilo, Xiao: Internet Mathematics 2(1):31-71´05]

As also noted by Stephen Boyd

DENSE VS. SPARSE IS NOT ENOUGH, SOLVERS NEED TO BE AWARE OF SYMMETRIES



Gaussian Processes

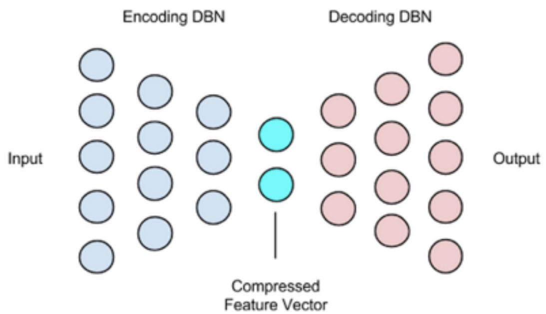


Matrix Factorization

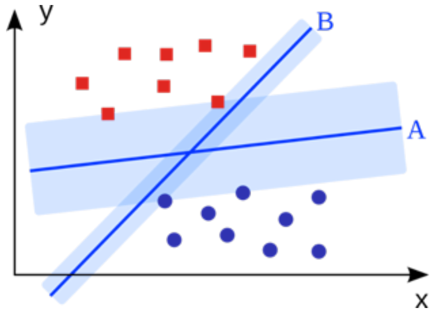
Transaction ID	Items Bought
2000	A,B,C
1000	A,C
4000	A,D
5000	B,E,F

Frequent itemsets

Frequent Itemset	Support
{A}	75%
{B}	50%
{C}	50%
{A,C}	50%



Autoencoder/Deep Learning



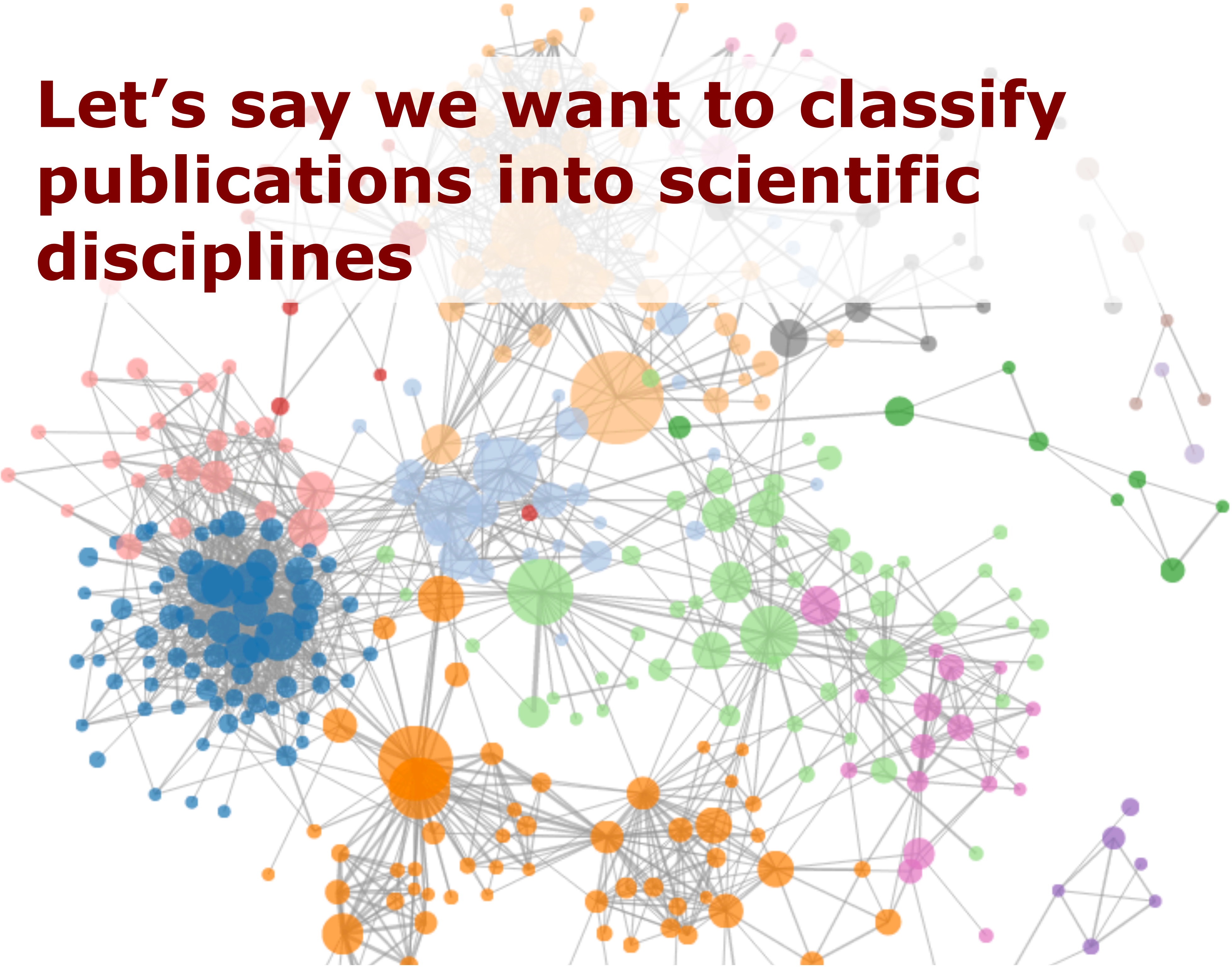
Support Vector Machines

Decision Trees/Boosting

and many more …

THIS PAVES THE WAY TO “**COMPILERS**” **FOR DATA SCIENCE MACHINES** IN GENERAL, NOT JUST GRAPHICAL MODELS!

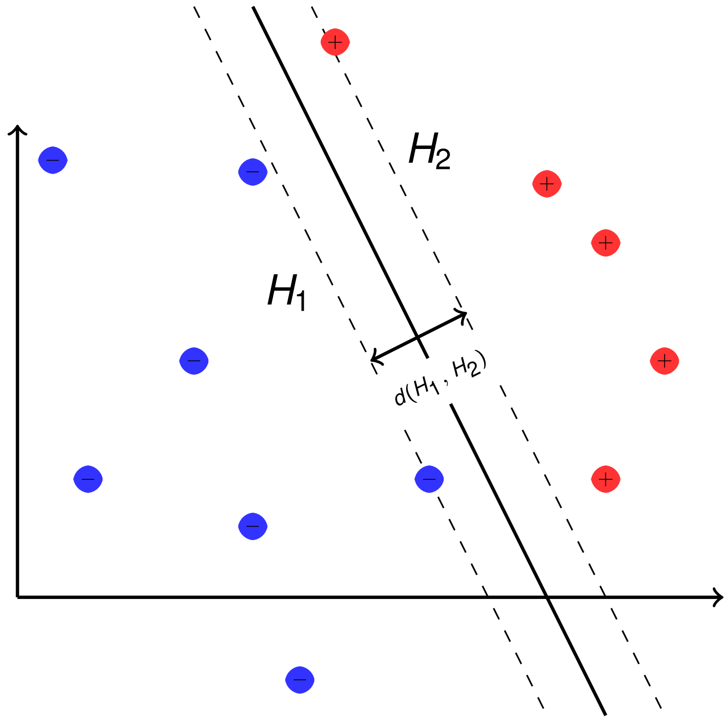
Let’s say we want to classify publications into scientific disciplines



[Vapnik ´79; Joachims ´98; Platt ´99; Bennett´99; Mangasarian´99; Collobert, Bengio ´01; Zhou, Zhang, Jiao´02; Schölkopf, Smola ´02; Chang, Lin ´11, Gu, Han ´13; Hsieh et al ´14; ...]

Standard data science approach: Support Vector Machines (SVMs)

$$H^* = \left\{ \vec{x} \mid \left\langle \vec{x}, \vec{\beta} \right\rangle + \beta_0 = 0 \right\}$$



$$d(H_1, H_2) = \frac{2}{||\vec{\beta}||}$$

This is a **quadratic program**. If you replace l_2- by l_1-, l_∞-norm you get a **linear program**

[Kersting, Mladenov, Tokmakov AIJ´15; Mladenov, Heinrich, Kleinhans, Gonsio, Kersting DeLBP´16; Mladenov, Kleinhans, Kersting AAAI´17]

Declarative Data Science Programming

Write down the problem in „paper form". The machine compiles it automatically into solver form.

reloop

http://www-ai.cs.uni-dortmund.de/weblab/static/RLP/html/

RELOOP: A Toolkit for Relational Convex Optimization

Embedded within Python s.t. loops and rules can be used

```
#inline definitions
slacks = sum{I in labeled(I)} slack(I);

#QUADRATIC OBJECTIVE
minimize: sum{J in feature(I,J)} weight(J)**2 + c1 * slack;

#labeled examples
subject to forall {I in labeled(I)}: labeled(I)*predict(I) >= 1 - slack(I)

#slacks are positive
subject to forall {I in labeled(I)}: slack(I) >= 0;
```

Logically parameterized variable (set of ground variables)

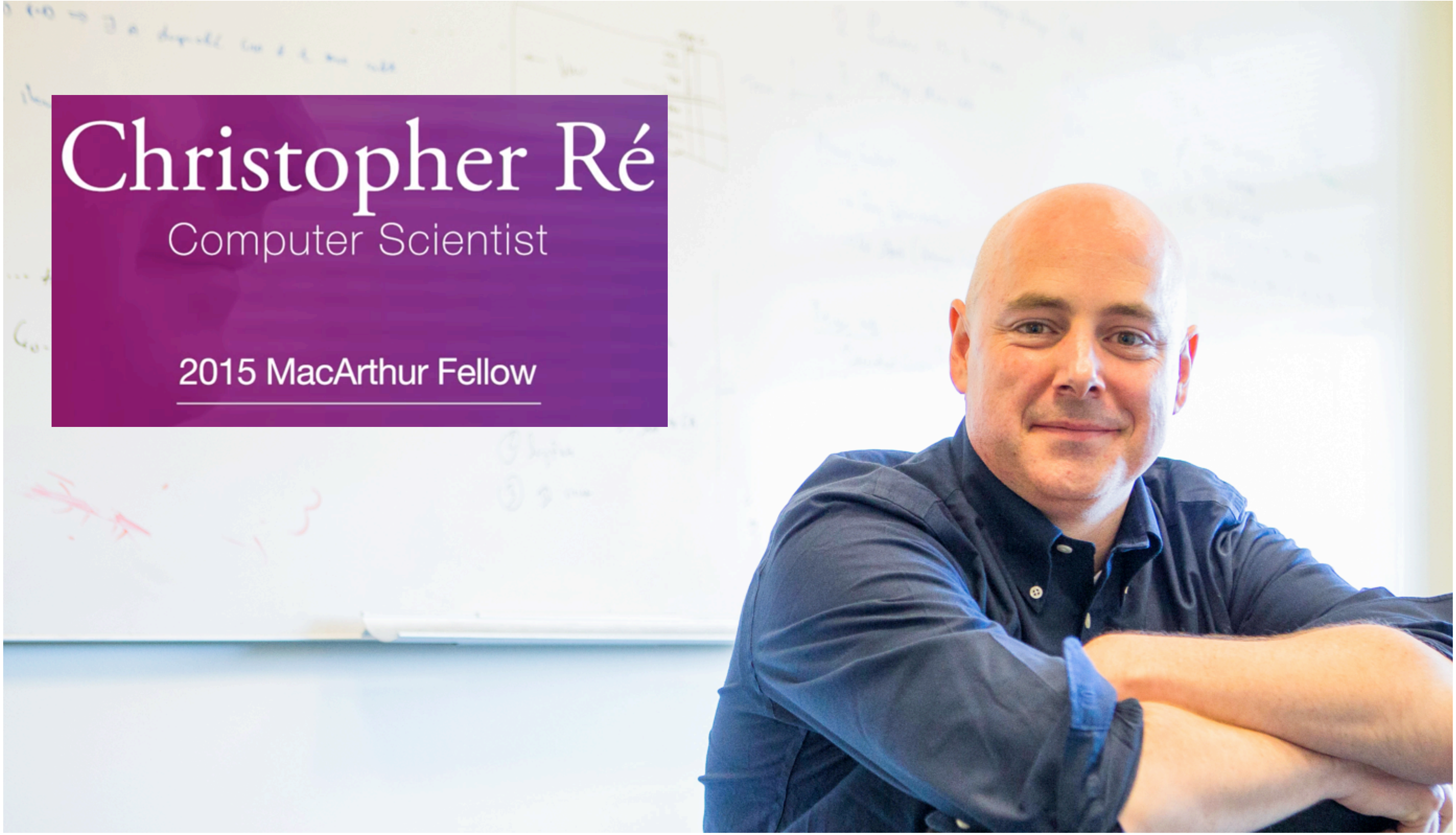
Logically parameterized objective

Data stored in an external DB

Logically parameterized constraint

LogicBlox, Apple and Uber are investing hundreds of millions of dollars

LogicBlox, Apple and Uber are investing hundreds of millions of dollars



But wait, publications are citing each other. OMG, I have to use graph kernels!

REALLY?

[Kersting, Mladenov, Tokmakov AIJ´15; Mladenov, Heinrich, Kleinhans, Gonsio, Kersting DeLBP´16; Mladenov, Kleinhans, Kersting AAAI´17]

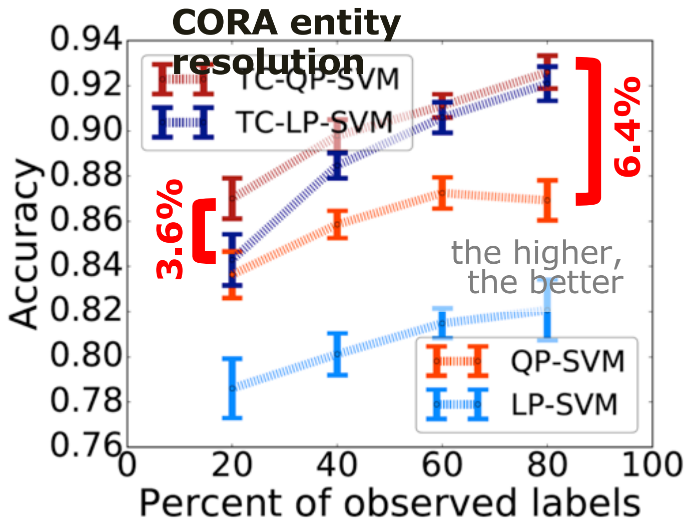
Kernels often scare non-experts. Our alternative: Simply program additional constraints

```
#inline definitions
slacks = sum{I in labeled(I)} slack(I);


#QUADRATIC OBJECTIVE
minimize: sum{J in feature(I,J)} weight(J)**2 + c1 * sla

#labeled examples should be on the correct side
subject to forall {I in labeled(I)}: labeled(I)*predict(

#slacks are positive
subject to forall {I in labeled(I)}: slack(I) >= 0;
```



Papers that cite each other should be on the same side of the hyperplane

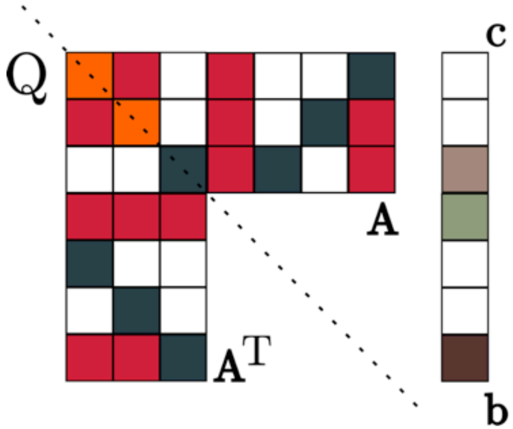
On par with state-of-the-art by just three lines of code

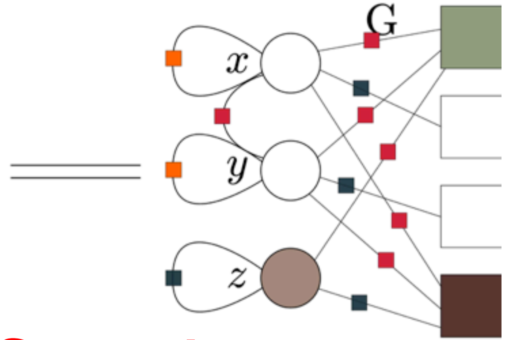
[Kersting, Mladenov, Tokmakov AIJ´15; Mladenov, Heinrich, Kleinhans, Gonsio, Kersting DeLBP´16; Mladenov, Kleinhans, Kersting AAAI´17]

Not only better predictive performance but also speed ups: the „-O1“ flag

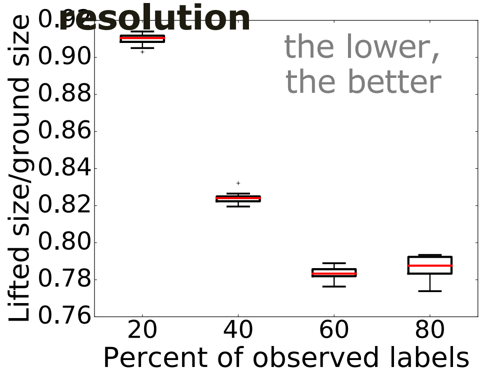
$$\max_{[x,y,z]^T \in \mathbb{R}^3} \quad 0x + 0y + 1z - 1z^2 - 2x^2 - 2y^2 + 1xy + 1yx$$

$$\text{s.t.} \quad \begin{bmatrix} 1 & 1 & 1 \\ -1 & 0 & 0 \\ 0 & -1 & 0 \\ 1 & 1 & -1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} \leq \begin{bmatrix} 1 \\ 0 \\ 0 \\ -1 \end{bmatrix}$$





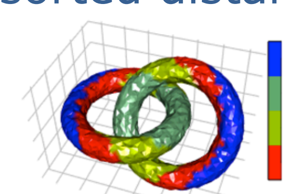
(1) Reduce the QP via WL on the QP graph
(2) Run any solver on the reduced QP

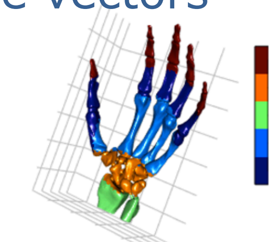


Again smaller models

Approximately Lifted SVM:

Cluster via K-means using sorted distance vectors

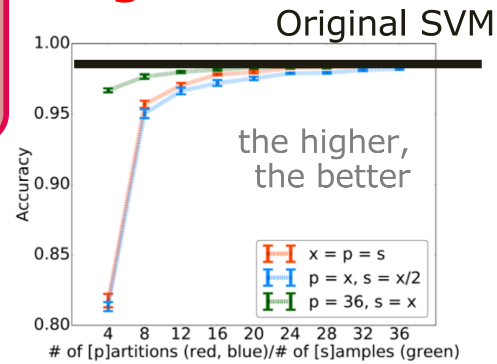


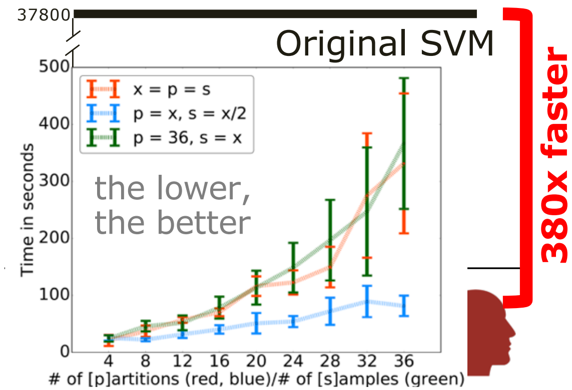


PAC-style generalization bound:
the approximately lifted SVM will very likely have a small expected error rate if it has a small empirical loss over the original dataset.

Symmetry Augmentation

MNIST image classification

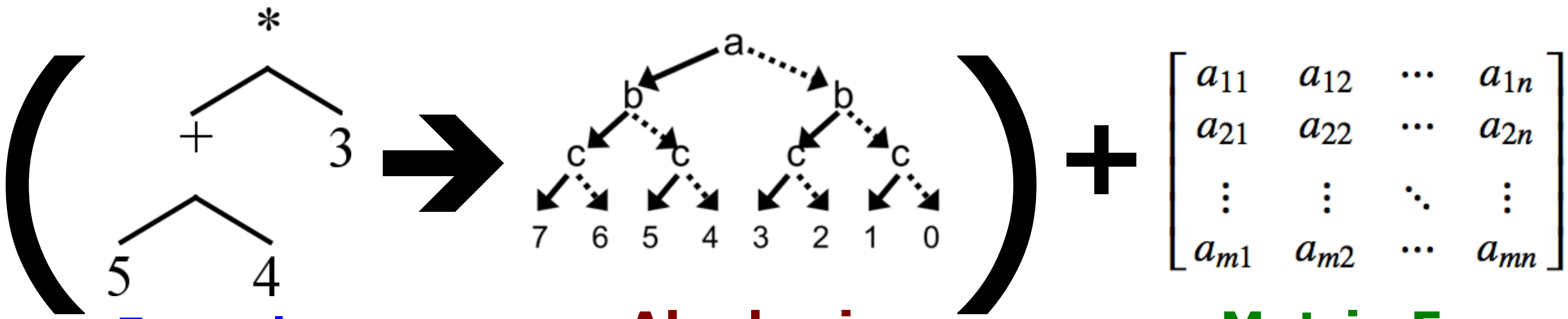




Industrial Strength Solvers such as CPLEX and GUROBI

And, there are other "-02", "-03", … flags lying ahead, e.g symbolic-numerical interior point solvers

[Mladenov, Belle, Kersting AAAI´17]



Formulae parse trees | **Algebraic Decision Diagrams** | **Matrix Free Optimization**

All this opens the general data science toolbox to DataScienceBases:

feature selection, least-squares regression, label propagation, ranking, collaborative filtering, community detection, subgroup discovery, …

factory2	4.194.500	22.020.100	119.099.000	2304	30248	> 48hrs.

>4.8x faster

Applies to QPs but here illustrated on MDPs for a factory agent which must paint two objects and connect them. The objects must be smoothed, shaped and polished and possibly drilled before painting, each of which actions require a number of tools which are possibly available. Various painting and connection

State-of-the-art

Conclusions so far

- Data Science is more than a single table. Graphs, different data types, relational DBs, … are central to data science but make the models more complex and larger
- We need probabilities, but graphs are not enough: we need logic!
- (Fractional) symmetry / group theory are a natural foundation for compiling into compact models, i.e., for fast data science
- Symmetries = “unimportant” variants of data, models, and computations
- Generally, probabilities are not enough, we need the whole AI spectrum!

Together with high-level data science languages

DECLARATIVE ML AND AI

- Shortens data science/AI code to make data science/AI techniques faster to write and easier to understand
- Reduces the level of expertise necessary to build data science/AI machines
- Facilitates the construction of more sophisticated data science/AI machines that incorporate rich domain knowledge and separate queries from underlying code
- Supports integrated data science/AI machines that think across a wide variety of domains and tasks as well as acknowledges data management
- Accelerates data science/AI machines by exploiting language properties, compression, and compilation

Mission and Schedule of the course: StarAI = Logical and Statistical AI

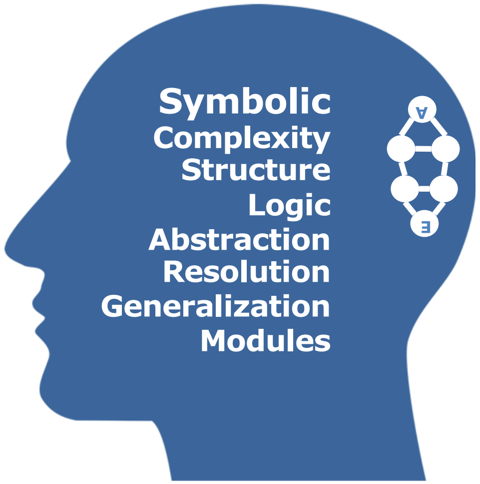
Field	Logical approach	Statistical approach
Knowledge representation	First-order logic	Graphical models
Automated reasoning	Satisfiability testing	Markov chain Monte Carlo
Machine learning	Inductive logic programming	Neural networks
Planning	Classical planning	Markov decision processes
Natural language processing	Definite clause grammars	Prob. context-free grammars

- **The real world is complex and uncertain**
- **Logic handles complexity**
- **Probability handles uncertainty**

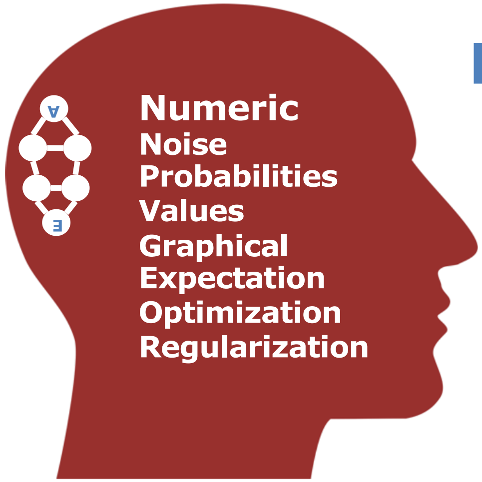
Mission and Schedule of the course: StarAI = Logical and Statistical AI

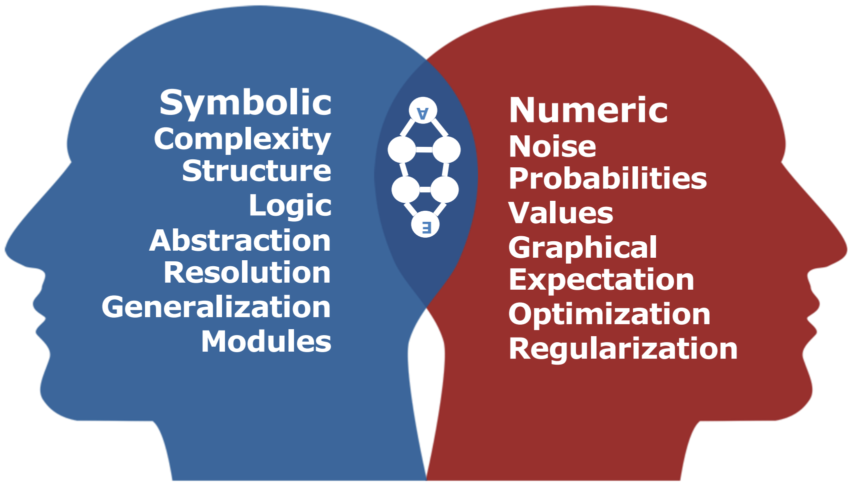
Essentially three blocks:

Logic



Probability





Logic + Probability

Mission and Schedule of the course: StarAI = Logical and Statistical AI

Providing an overview and a synthesis of StarAI

- (Inductive) Logic (Programming)

- Graphical Models: Inference and Learning

- Statistical Relational Representations
  - Problog
  - Markov Logic Networks
- Inference
  - (Exact) Lifted Inference
  - Approximate Inference
- Learning
- Beyond Probabilities

Prerequisites

- We provide background, but class can be fast paced
- Ability to deal with "abstract mathematical concepts"
- Do not hesitate to ask questions. Let's make this course as interactive as possible

devendra.dhami@tu-darmstadt.de

kersting@cs.tu-darmstadt.de

Exercise Sessions

- Very useful!
  - Review material, present background, answer questions, Discussion, just chatting about everything.
- „Hands on" Project ?
  - How does this work in practice?
- 1.5 hours per (every second?) week
- Date and place: **Wednesday 13:30-15:10**

Grading

- Final exam will count
- Entry to the final exam
  - Regularly attending exercise sessions
  - Regularly working on the homework

Homeworks (if there are paper exercises)

- Homeworks might be hard, start early ☺
- Due in the beginning of exercise sessions
- Collaboration
  - You may **discuss the questions**
  - Each student writes its own answers
  - Write on your homework anyone with whom you collaborated
- **IMPORTANT:** We may use some material from other classes or from papers for the homeworks. Unless otherwise specified, please only look at the readings when doing your homework ! **You are taking this class because you want to learn, so this rule is self-enforced**

But, you may have to run a project!

- Use our own library SPflow for sum-product networks:

https://github.com/SPFlow/SPFlow

This is probabilistic programming but not full-fledged

1. Integrate it with pyro (http://pyro.ai/)
   - As basic distribution
   - As deep layer within variational inference
2. Integrate it with ProbLog (https://dtai.cs.kuleuven.be/problog/)
   - https://arxiv.org/pdf/1805.10872.pdf

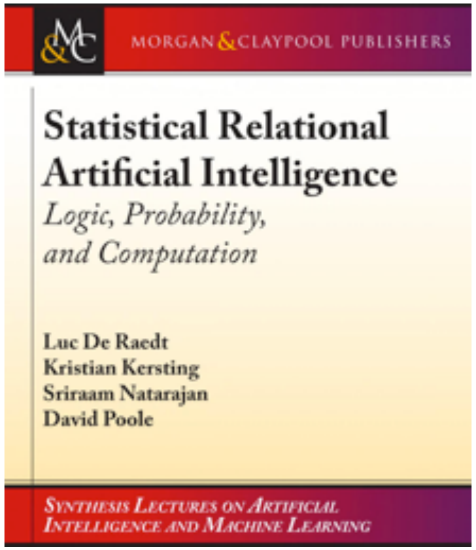
Disclaimer

TECHNISCHE UNIVERSITÄT DARMSTADT

- Thanks to the SRL/StarAI crowd for all their exciting contributions!
- The course is necessarily incomplete; apologize to anyone whose work is not cited

Based on

Luc De Raedt, Kristian Kersting, Sriraam Natarajan, David Poole: **Statistical Relational Artificial Intelligence: Logic, Probability, and Computation.** Morgan and Claypool Publishers, Synthesis Lectures on Artificial Intelligence and Machine Learning, ISBN: 9781627058414, 2016.



Let's start and have fun