Introducing Relational^{AI}

Probabilistic Graphical Models –and- Statistical Relational Learning

Molham Aref -- Relational^{AI}

OVERVIEW

Age: 9 months old AI/ML startup

Team:

- 16 people, 10 PhD's, 4 former university professors
- Faculty Network: 9 @ ~20% + summers, additional ~12+ in extended network

Location: distributed team (SV, Atlanta, Seattle, Toronto, NYC, Utrecht, London)
Offices in Berkeley and Utrecht

Financials: Funded through 2019 + revenue from consulting

Industries served: TBD (looking at Financial Services, Business Intelligence)

- If all else fails we're going to pivot to Deep Quantum Crypto Blockchain of Things and hope for the best

HOW WE SEE OURSELVES

Expertise in systems (8 people)

- We built sophisticated compilers & interpreters (OOPSLA, ECOOP, ICFP)
- We built databases that advanced the state of the art (SIGMOD, VLDB)

Expertise in theory (7 people)

- We prove things (PODS, ICDT, TODS, JACM, POPL)
- We designed first-of-a-kind asymptotically efficient algorithms
  - Worst case optimal join algorithms
  - Asymptotically superior query plans

Expertise in ML & AI (7 people)

- We implemented scalable ML & probabilistic systems (NIPS, ICML, AAAI)
- We developed sophisticated statistical and mathematical models

INDUSTRIAL EXPERIENCE

HNC

- Financial services: Fraud detection -- credit card, insurance (23 of top 25 credit card issuing banks)
- Neural networks on proprietary HW accelerators (also computational intelligence, Database mining)
- IPO 1995, acquired by FICO in 2002

Retek

- Retail: Demand forecasting, supply chain optimization, pricing (majority of Retail Global 250)
- Time series with approximate optimization – first hierarchical forecasting solution to scale to Retail volumes
- IPO 1999, acquired by Oracle in 2005

Optimi

- Telecom: Wireless network optimization (AT&T, Cingular, Next, America Movil, Telefonica)
- Monte Carlo simulation, heuristic search (simulated annealing)
- Acquired by Ericsson in 2010

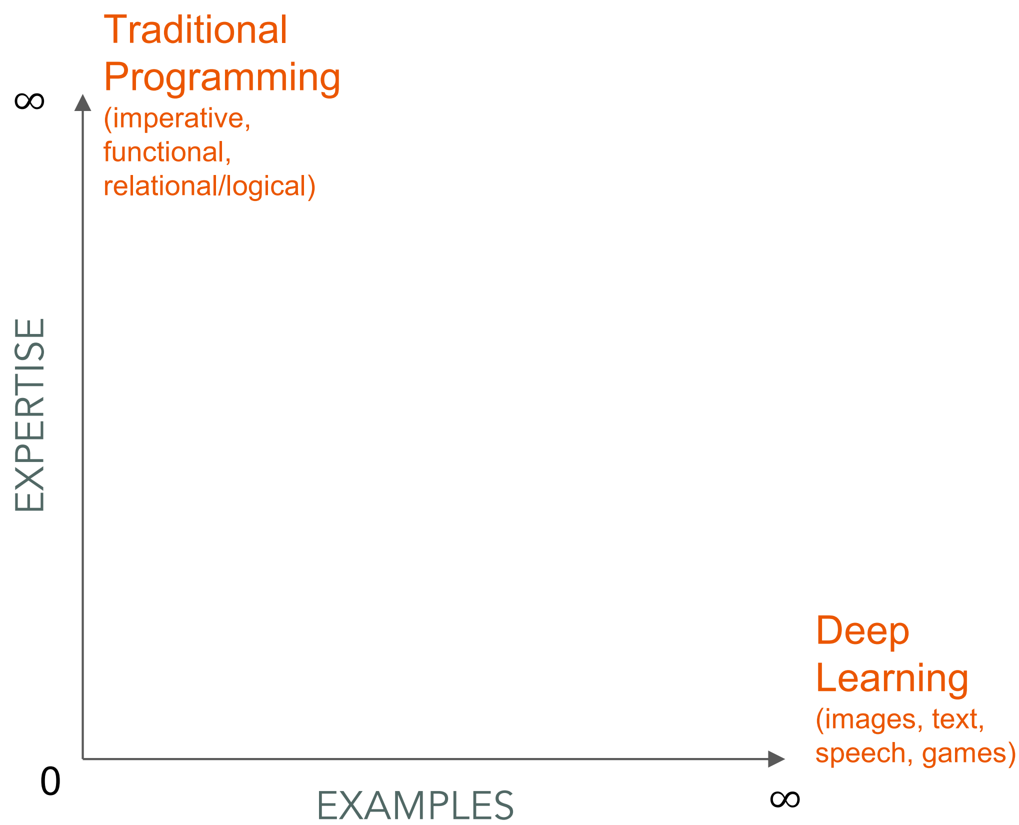
Brickstream

- Retail: in-store video analytics (>30% market share globally)
- Old-school computer vision (pre-deep learning) – first industrial use of stereo cameras
- Acquired by Point-Grey which was acquired by FLIR in 2015

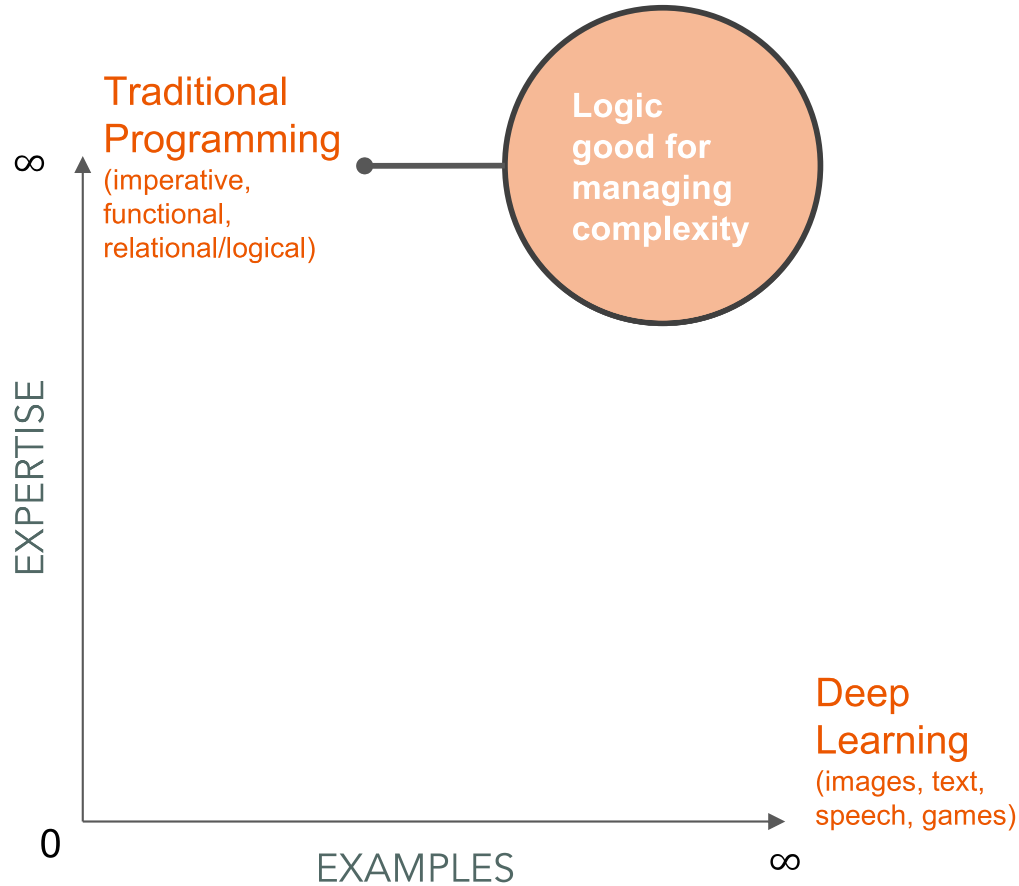
LogicBlox/Predictix

- Retail: Demand forecasting, supply chain optimization, pricing (3 of top 6 US retailers + dozen large global retailers)
- Factorization machines, linear programming, integer programming – first on cloud
- Acquired by Infor in 2016

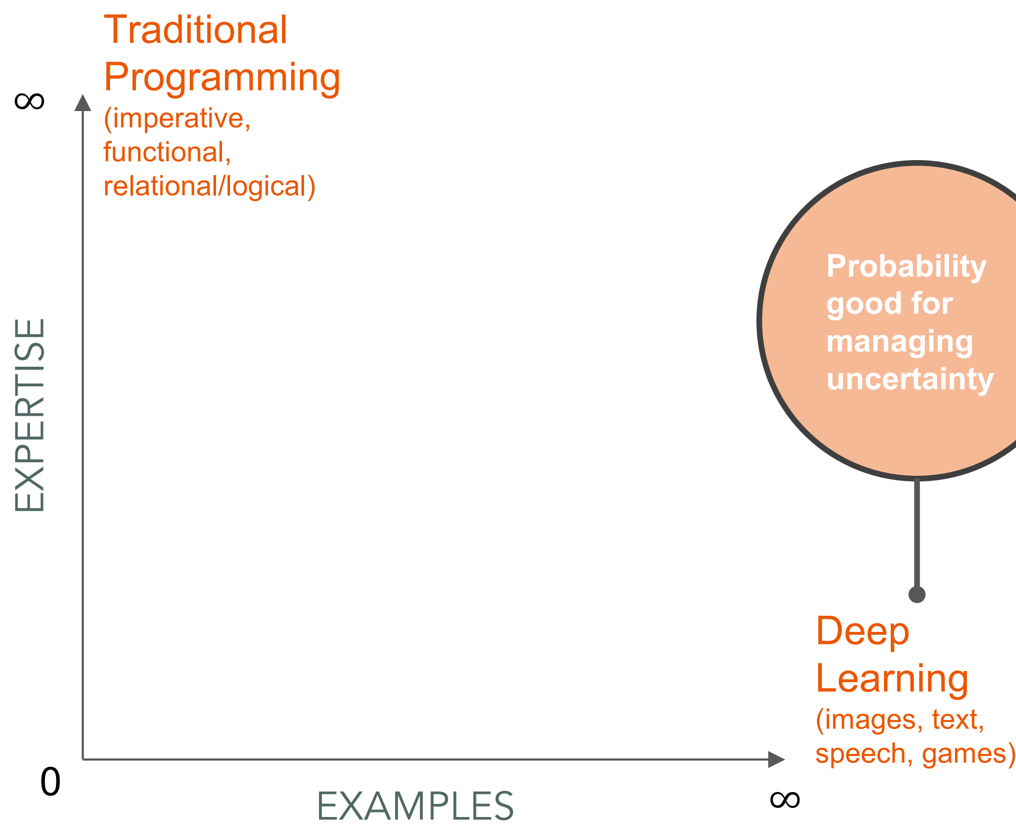
STATE OF THE PRACTICE



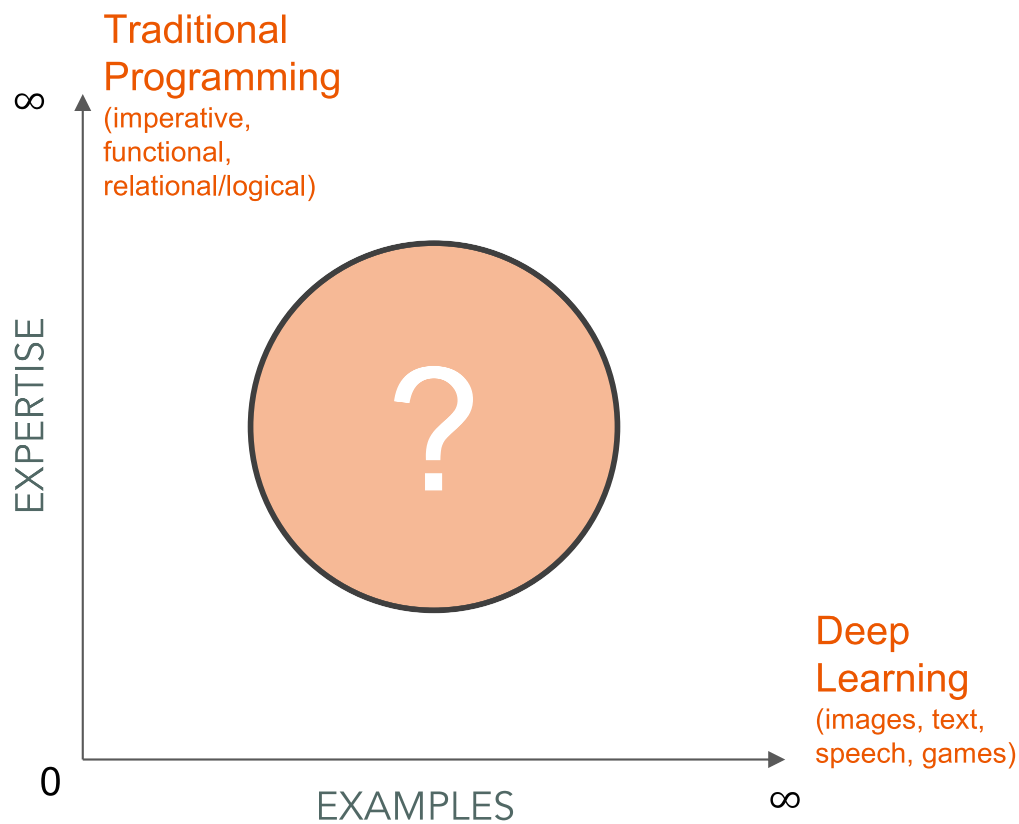
STATE OF THE PRACTICE



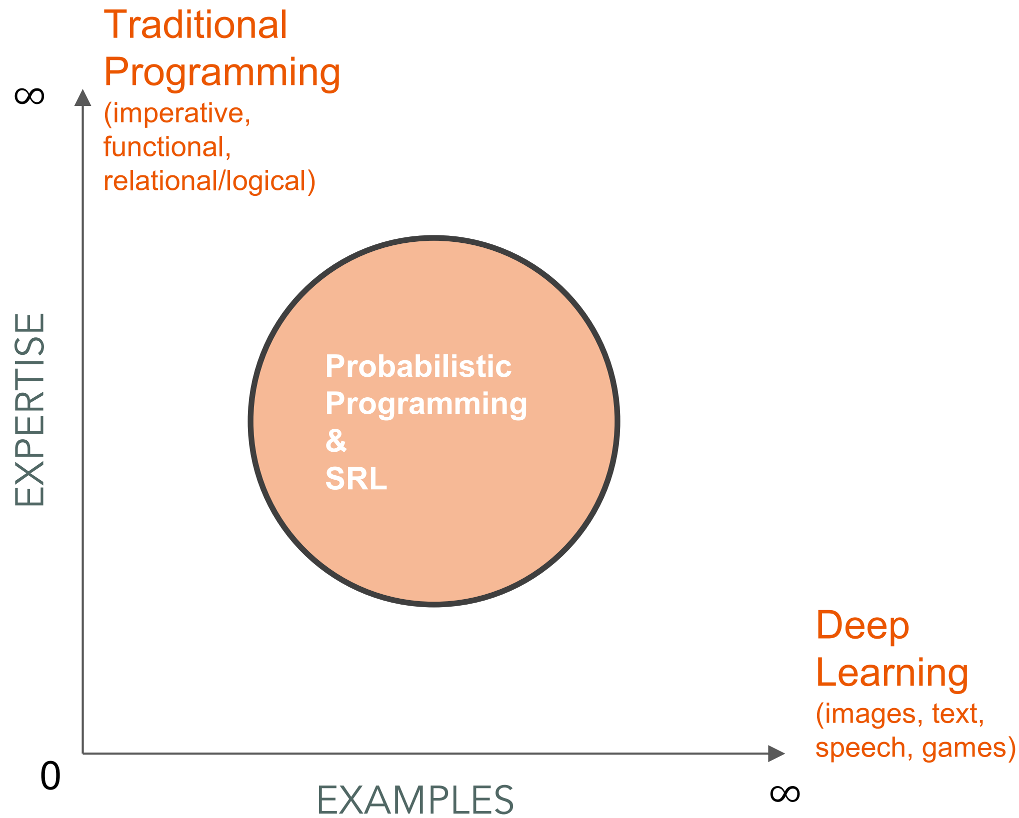
STATE OF THE PRACTICE



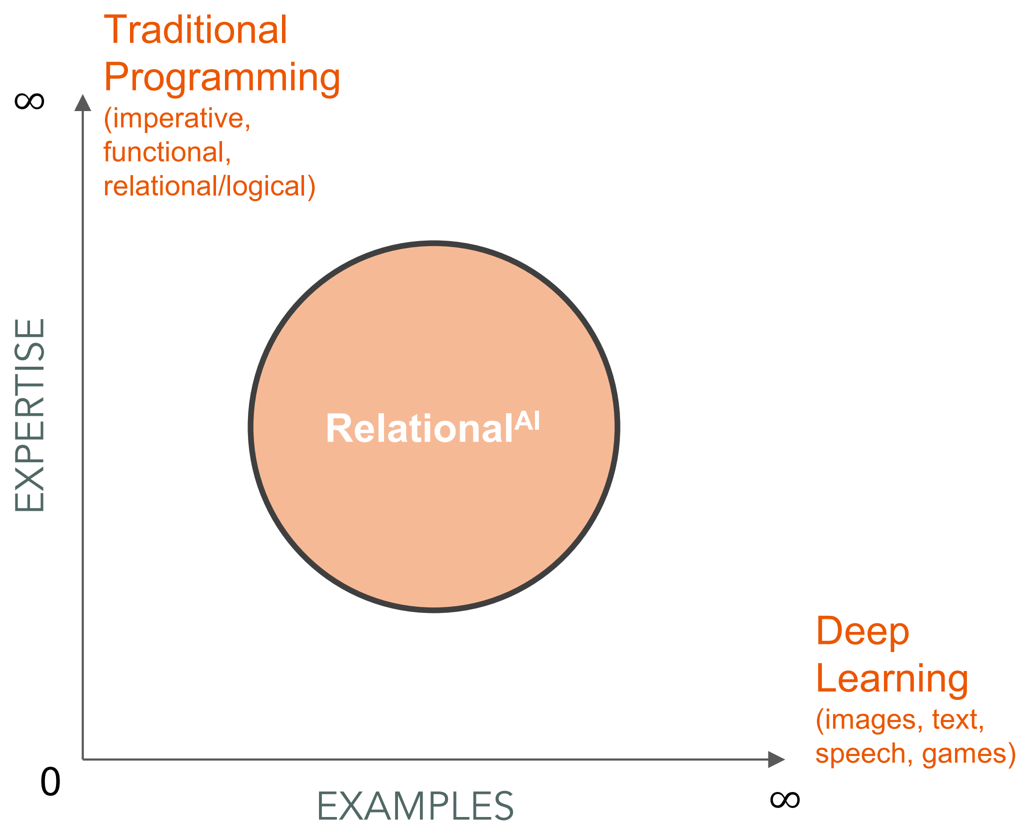
STATE OF THE PRACTICE



STATE OF THE PRACTICE



STATE OF THE PRACTICE



OUR SECRET SAUCE

- We know how to exploit problem structure to make optimization asymptotically faster
  - We know how to perform stochastic gradient descent and batch gradient descent directly on normalized relational data
  - Most ML Methods can be solved well with SGD, some with BGD

- Asymptotically faster optimization means anything that depends on it can go faster
  - Learning/Parameter optimization
  - Hyperparameter optimization
  - Feature engineering
  - Structure learning
  - Inference
  - …

STRUCTURE FOR TRADITIONAL/DISCRIMINATIVE ML

5 COMPONENTS OF MACHINE LEARNING

- Method (or model class)
  - e.g. FM, decision tree, neural network, …
- Loss (error) function
  - e.g. Absolute error (L1 norm), Square error (L2 norm), …
- Generalization mechanism
  - e.g. Regularization (norm * penalty), cross validation
- Evaluation function
  - Takes the model parameters and the input and produces a prediction
- Optimizer
  - E.g. Gradient descent, EM algorithm

5 TYPES OF METHODS or MODEL CLASSES

- Regression: predict a number
  - Linear regression with VIF
  - LASSO regression
  - Multi-time series prediction
  - Nonparametric regression
  - Mixture of experts
  - Factorization machines
  - Polynomial regression
- Classification: predict a category
  - Naïve Bayes classifier
  - Non-parametric Bayes classifier
  - K-nearest neighbor classifier
  - Support vector machine
  - Decision tree
  - Hidden Markov model
- Density Estimation: find likelihood of objects
  - Histograms
  - Kernel density estimation
- Clustering: find natural groups
  - K-means
  - Spectral clustering
  - Mean shift clustering
- Dimension Reduction: combine features
  - Singular value decomposition
  - Maximum variance unfolding
  - Non-negative matrix factorization
  - Kernel principal component analysis
  - (Ensemble) Singular value decomposition
  - GROUSE
  - Random projections
  - Tensor factorization PARFAC/CANDECOMP

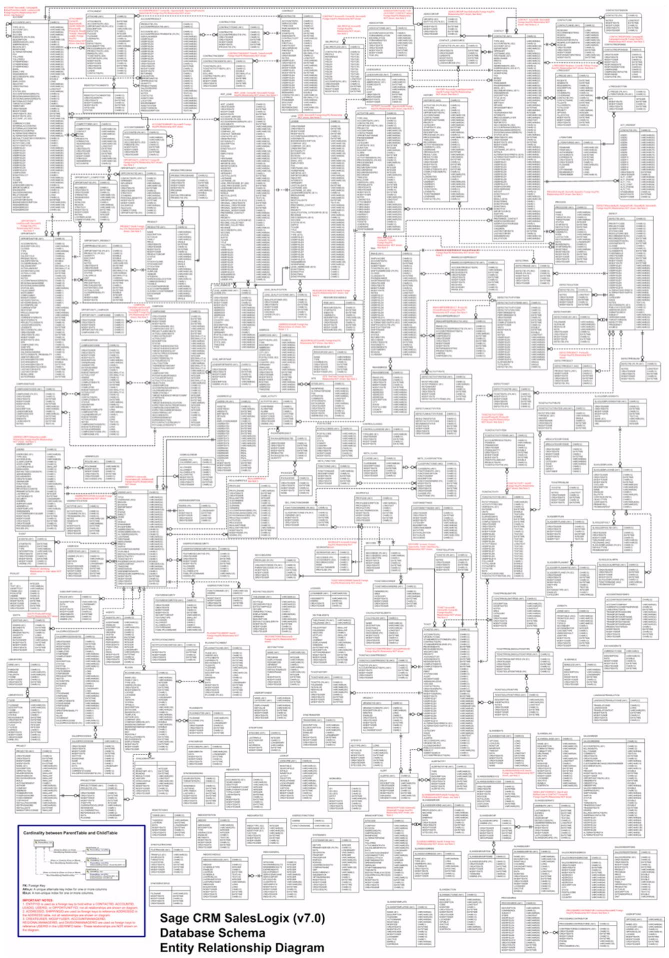
DESIGN MATRIX

Features

Entities

ID	x1	x2	x3	…	y
i_1					
i_2					
…					
i_{1B}					

DESIGN MATRIX IS A VIEW ON STRUCTURED (RELATIONAL) DATA



Features

Entities

ID	x1	x2	x3	…	y

WE CAN EXPLOIT THE RELATIONAL STRUCTURE

We use algebraic structure (e.g. semi-rings) to “push the aggregations through the joins” to implement lifted stochastic and batch gradient descent for efficient learning of a variety of model classes

- Linear regression
- Polynomial regression
- Factorization machines
- Decision trees
- Neural nets
- …

(many more on the way)

BENCHMARK (1/3)

		v_1	v_2	v_3	v_4
Join Representation	Listing	774M	3.614G	3.614G	3.614G
(#values)	Factorized	37M	169M	169M	169M
Compression	Fact/List	20.9×	21.4×	21.4×	21.4×
Join Computation (PSQL) for R, TensorFlow, libFM		50.63	216.56	216.56	216.56
Factorized Computation of 43 Counts over Join		8.02	34.15	34.15	34.15
Linear regression					
Features	without FDs	33 + 55	33+55	33+1340	33+2702
(continuous+categorical)	with FDs	same as above, there are no FDs			33+2653
Aggregates	without FDs	595+2,418	595+2,421	595+111,549	595+157,735
(scalar+group-by)	with FDs	same as above, there are no FDs			595+144,589
MADLib (ols)	Learn	1,898.35	8,855.11	> 79,200.00	–
R (QR)	Export/Import	308.83	–	–	–
	Learn	490.13	–	–	–
TensorFlow (FTLR)	Export/Import	74.72	372.70	372.70	372.70
(1 epoch, batch size 1K)	Learn	2762.50	12710.53	12724.94	12708.11
F	Aggregate	93.31	424.81	OOM	OOM
	Converge (runs)	0.01 (359)	0.01 (359)		
AC/DC	Aggregate	25.51	116.64	117.94	895.22
	Converge (runs)	0.02 (343)	0.02 (367)	0.42 (337)	0.66 (365)
AC/DC+FD	Aggregate	same as **AC**			380.31
	Converge (runs)	there are no FDs			8.82 (366)
Speedup of **AC/DC+FD** over	MADlib	74.36×	75.91×	> 669.14×	∞
	R	33.28×	∞	∞	∞
	TensorFlow	113.12×	114.01×	112.49×	34.17×
	F	3.65×	3.64×	∞	∞
	AC/DC	same as **AC/DC**, there are no FDs			2.30×

BENCHMARK (2/3)

		v_1	v_2	v_3	v_4
Join Representation	Listing	774M	3.614G	3.614G	3.614G
(#values)	Factorized	37M	169M	169M	169M
Compression	Fact/List	20.9×	21.4×	21.4×	21.4×
Join Computation (PSQL) for R, TensorFlow, libFM		50.63	216.56	216.56	216.56
Factorized Computation of 43 Counts over Join		8.02	34.15	34.15	34.15
Polynomial regression degree 2					
Features	without FDs	562+2,363	562+2,366	562+110,209	562+154,033
(continuous+categorical)	with FDs	same as above, there are no FDs			562+140,936
Aggregates	without FDs	158k+742k	158k+746k	158k+65,875k	158k+46,113k
(scalar+group-by)	with FDs	same as above, there are no FDs			158k+36,712k
MADlib (ols)	Learn	$> 79,200.00$	$> 79,200.00$	$> 79,200.00$	–
AC/DC	Aggregate	132.43	517.40	820.57	7,012.84
	Converge (runs)	3.27 (321)	3.62 (365)	349.15 (400)	115.65 (200)
AC/DC+FD	Aggregate	same as **AC/DC**			1,819.80
	Converge (runs)	there are no FDs			219.51 (180)
Speedup of **AC/DC+FD** over	MADlib	> 583.64×	> 152.01×	> 67.71×	∞
	AC/DC	same as **AC/DC**, there are no FDs			3.50×

BENCHMARK (3/3)

		v_1	v_2	v_3	v_4
Join Representation	Listing	774M	3.614G	3.614G	3.614G
(#values)	Factorized	37M	169M	169M	169M
Compression	Fact/List	20.9×	21.4×	21.4×	21.4×
Join Computation (PSQL) for R, TensorFlow, libFM		50.63	216.56	216.56	216.56
Factorized Computation of 43 Counts over Join		8.02	34.15	34.15	34.15
Factorization machine degree 2 **rank** 8					
Features	without FDs	530+2,363	530+2,366	530+110,209	530+154,033
(continuous+categorical)	with FDs	same as above, there are no FDs			562+140,936
Aggregates	without FDs	140k+740k	140k+744k	140k+65,832k	140k+45,995k
(scalar+group-by)	with FDs	same as above, there are no FDs			140k+36,595k
libFM (MCMC)	Export/Import	412.84	1462.54	3,096.90	3,368.06
	Learn (runs)	19,692.90 (300)	103,225.50 (300)	79,839.13 (300)	87,873.75 (300)
AC/DC	Aggregate	128.97	498.79	772.42	6,869.47
	Converge (runs)	3.03 (300)	3.05 (300)	262.54 (300)	166.60 (300)
AC/DC+FD	Aggregate	same as **AC/DC**			1,672.83
	Converge (runs)	there are no FDs			144.07 (300)
Speedup of **AC/DC+FD** over	libFM	152.70×	209.03×	80.34×	50.33×
	AC/DC	same as **AC/DC**, there are no FDs			3.87×

STATISTICAL RELATIONAL LEARNING

SRL and StarAI

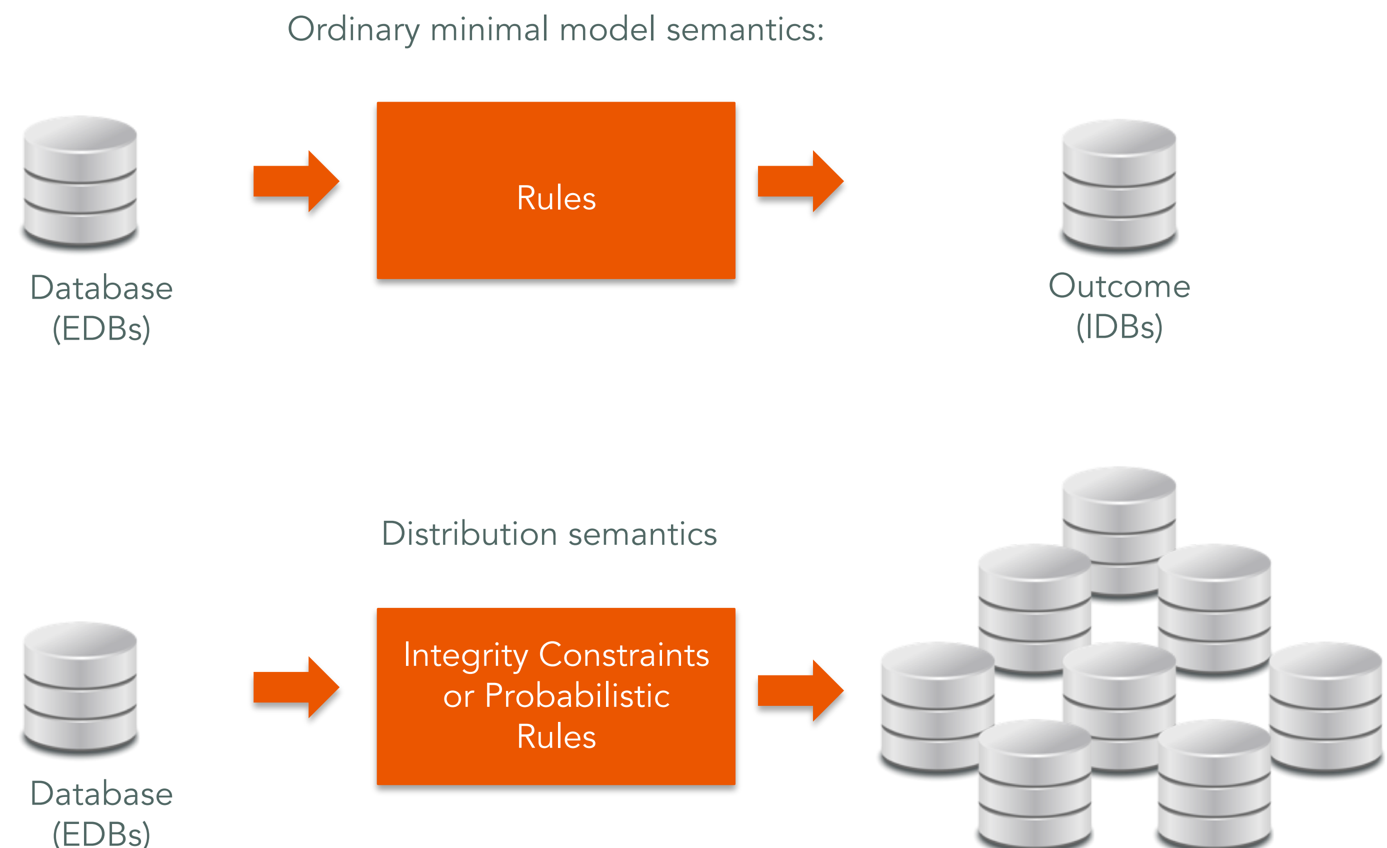
MOTIVATION

- Graphical models are considered by some to be "one of the most exciting advances in machine learning (AI, signal processing, coding, control, …) in the last decades"
- Graphical models allow us to gain global insight based on local observations
- There are different types of graphical models
    - Directed: eg. Bayesian Networks (aka belief networks)
    - Undirected: e.g. Mark Networks (aka Markov Random Fields), Factor Graphs
    - Mixed: e.g. Chain Graphs - both directed acyclic graphs and undirected graphs are special cases of chain graphs, which can therefore provide a way of unifying and generalizing Bayesian and Markov networks
- Statistical Relational models generalize PGM's in the same way that first order logic generalizes propositional logic – they allow us to quantify over individuals/entities

STATISTICAL RELATIONAL LEARNING (SRL)

- Knowledge is represented as a distribution over possible worlds
  - Finite and infinite sets of possible worlds are supported
  - Undirected models: via integrity constraints
    - We specify the constraints that determine the legal set of possible worlds & a function to score each of them.
    - We don't have to provide a program to generate each possible world.
  - Directed models: via probabilistic programs
    - We have to provide a program to generate each possible world. Score by normalizing - frequency of a given world relative to all others.
  - Normalize the score of each world by the sum of scores of all the worlds
- Inference
  - Unlike "traditional" methods where prediction is the input applied to the parameters of the model class, inference in SRL requires expensive optimization or (approximate) integration over possible worlds
- Learning
  - Unlike traditional learning algorithms, <u>just one instance to learn from (the relational DB)</u>
  - Structure learning uses inference during each step

SEMANTICS

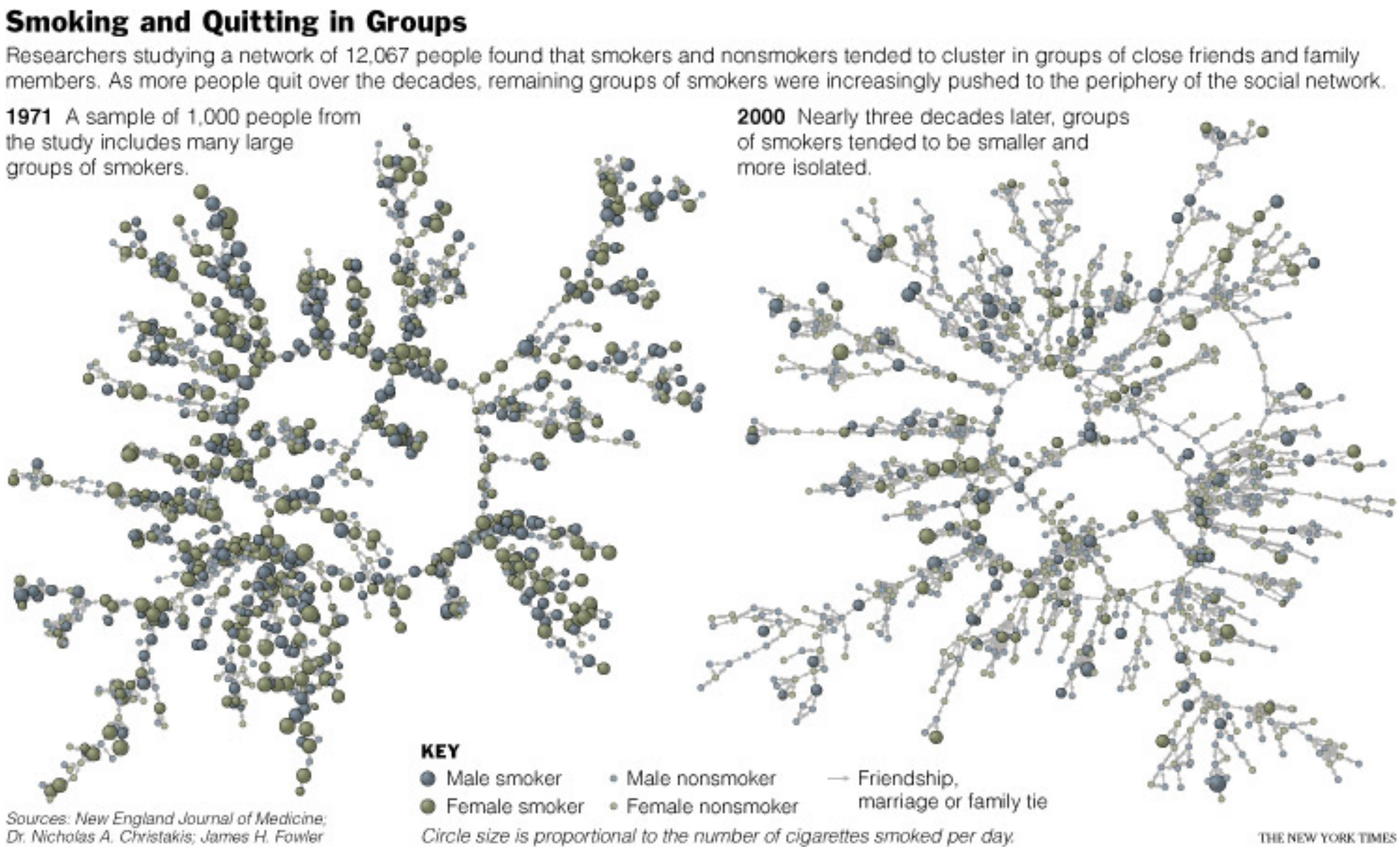


Slide thanks to Benny Kimelfeld

UNDIRECTED MODELS

Use Integrity Constraints to specify a set of possible worlds & define a scoring function for each

SMOKERS AND FRIENDS



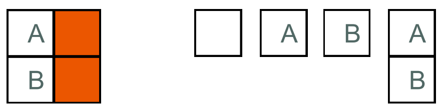
CERTAIN KNOWLEDGE WITH INTEGRITY CONSTRAINTS

A logical Knowledge Base is a set of Integrity Constraints that define a set of possible worlds:

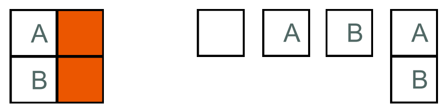
```
person(x)
smokes(x) -> person(x)
cancer(x) -> person(x)
friends(x, y) -> person(x), person(y)
```

Assuming persons Alice (A) and Bob (B), then there are 4 possible relations for each of:

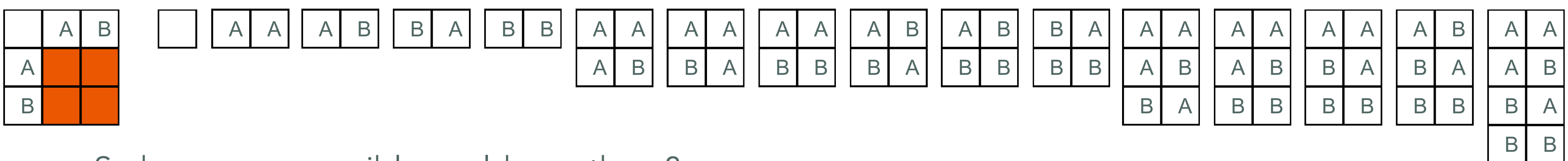
smokes,



cancer:



There are 16 possible relations for friends



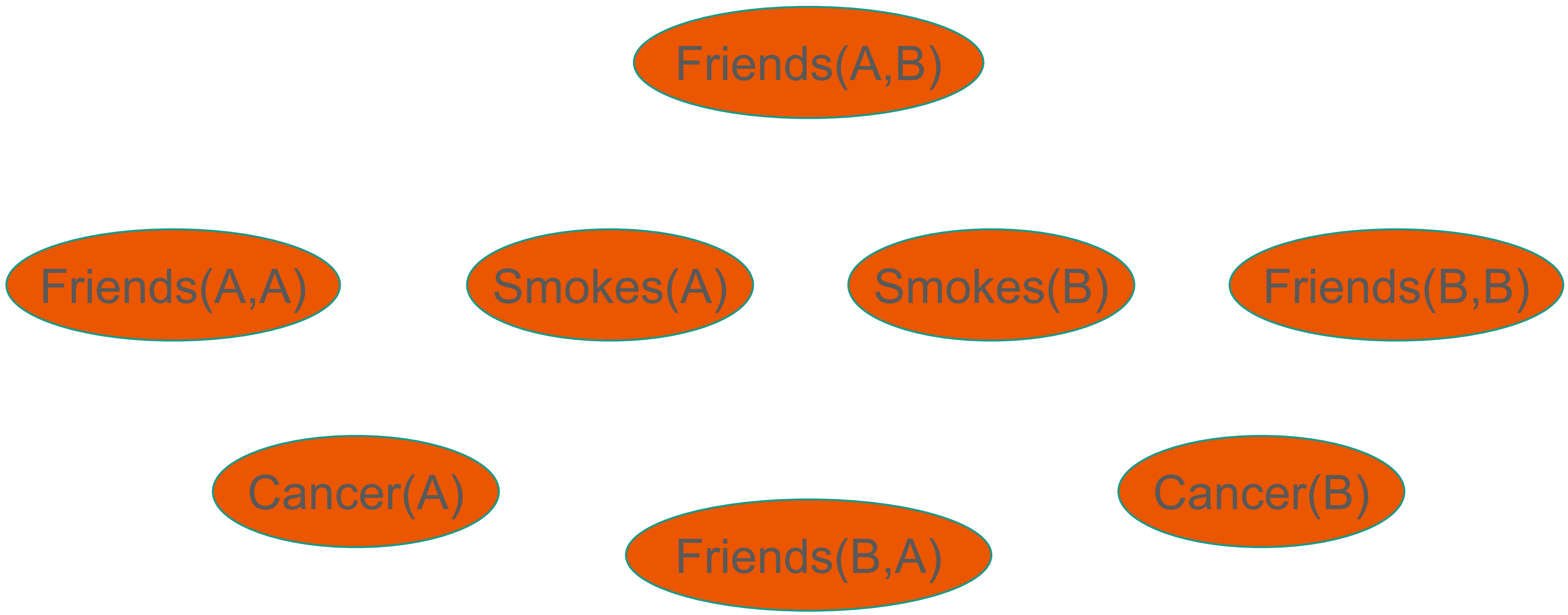
So how many possible worlds are there?

- 2 bits for for each of smokes cancer and 4 bits for friends: **2^8 or 256 possible worlds**

And how many if we add a 3rd person Carla (C)?

- 3 bits * 2 unary relations + 9 bits for binary relation: **2^{15} or 32K possible worlds**

EACH POSSIBLE TUPLE IS A NODE IN A GRAPHICAL MODEL



Slide thanks to Pedro Domingos

PROBABILISTIC KNOWLEDGE WITH WEIGHTED INTEGRITY CONSTRAINTS

Smoking causes cancer
Friends have similar smoking habits

Represent probabilistic knowledge with soft (weighted) Integrity Constraints

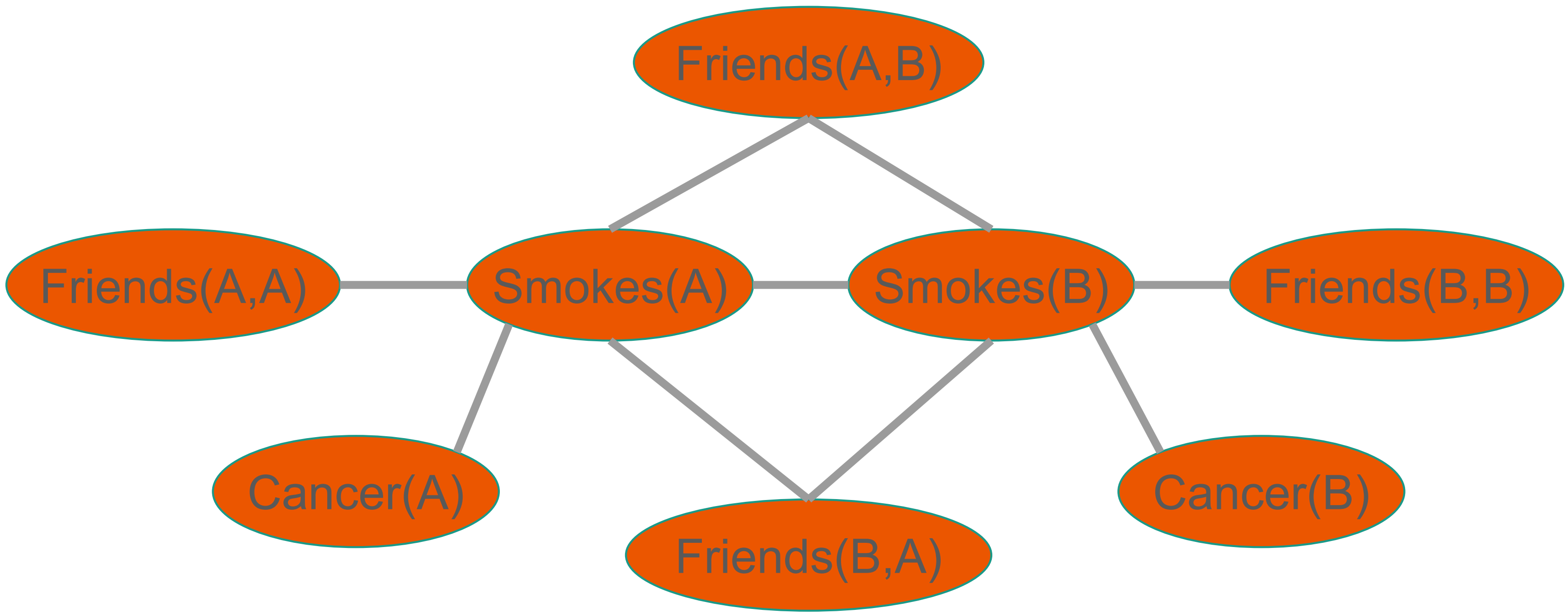
```
person(x)
smokes(x) -> person(x)
cancer(x) -> person(x)
friends(x, y) -> person(x), person(y)
w1  smokes(x) -> cancer(x)
w2  smokes(x), friends(x, y) -> smokes(y)
```

When a world violates a formula, it becomes less probable, not impossible

Weights give indication of certainty in domain knowledge or expertise

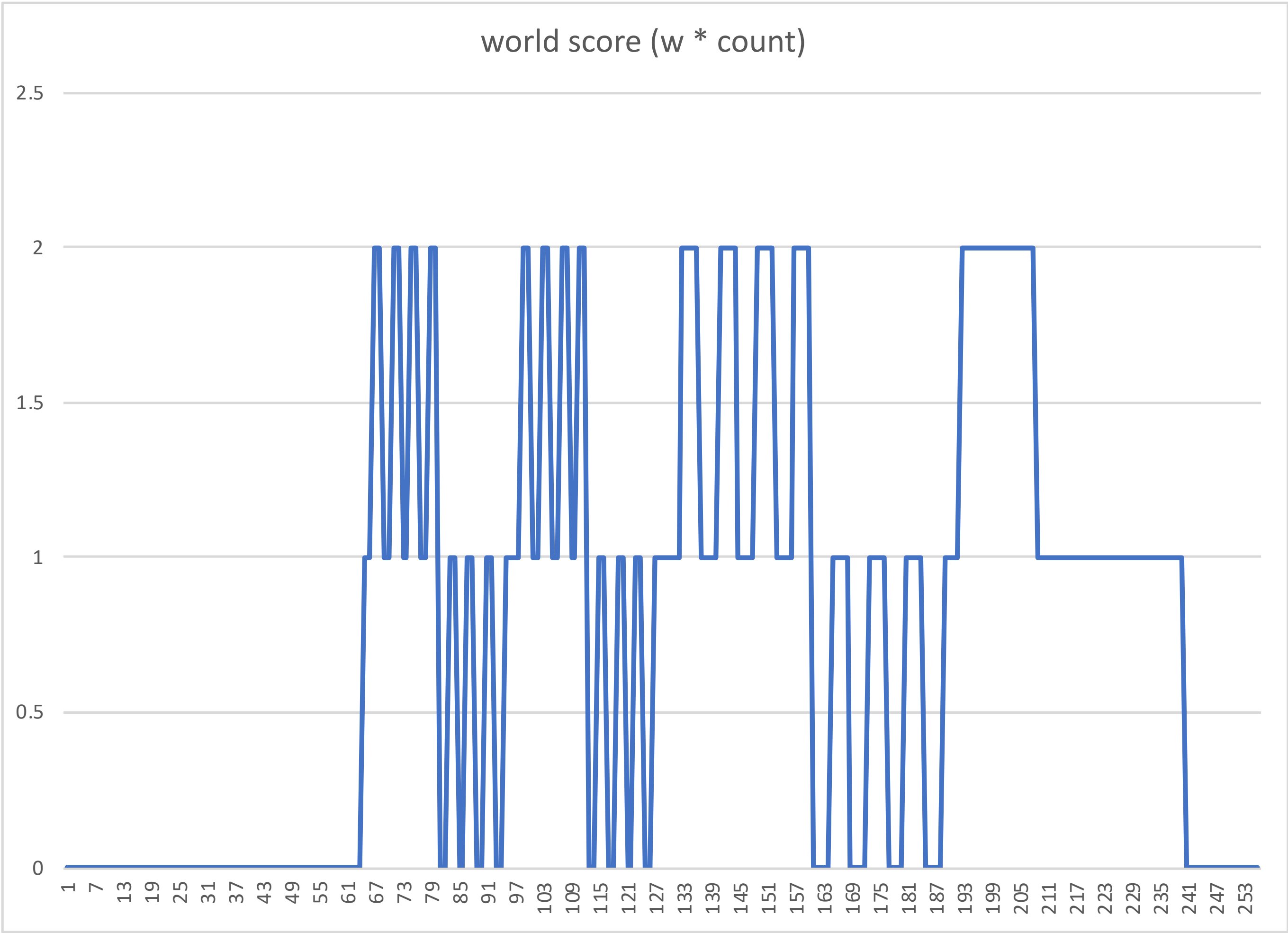
- 0 -> no confidence -- might as well not have the IC
- Infinity -> absolute certainty – hard constraint
- -Infinity -> absolute certainty in the converse – hard constraint on negated IC

GRAPHICAL MODEL



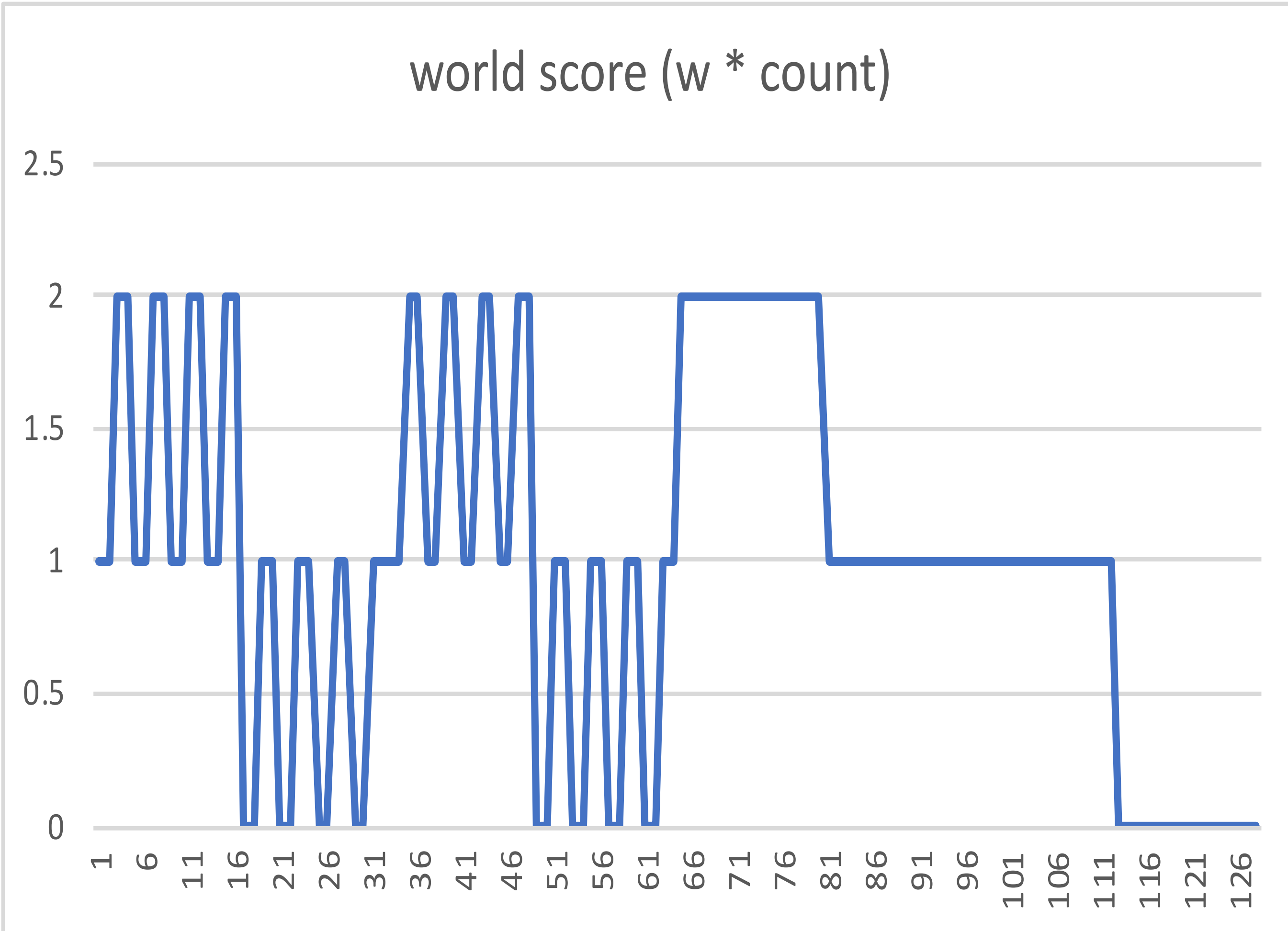
Slide thanks to Pedro Domingos

ALL WORLDS

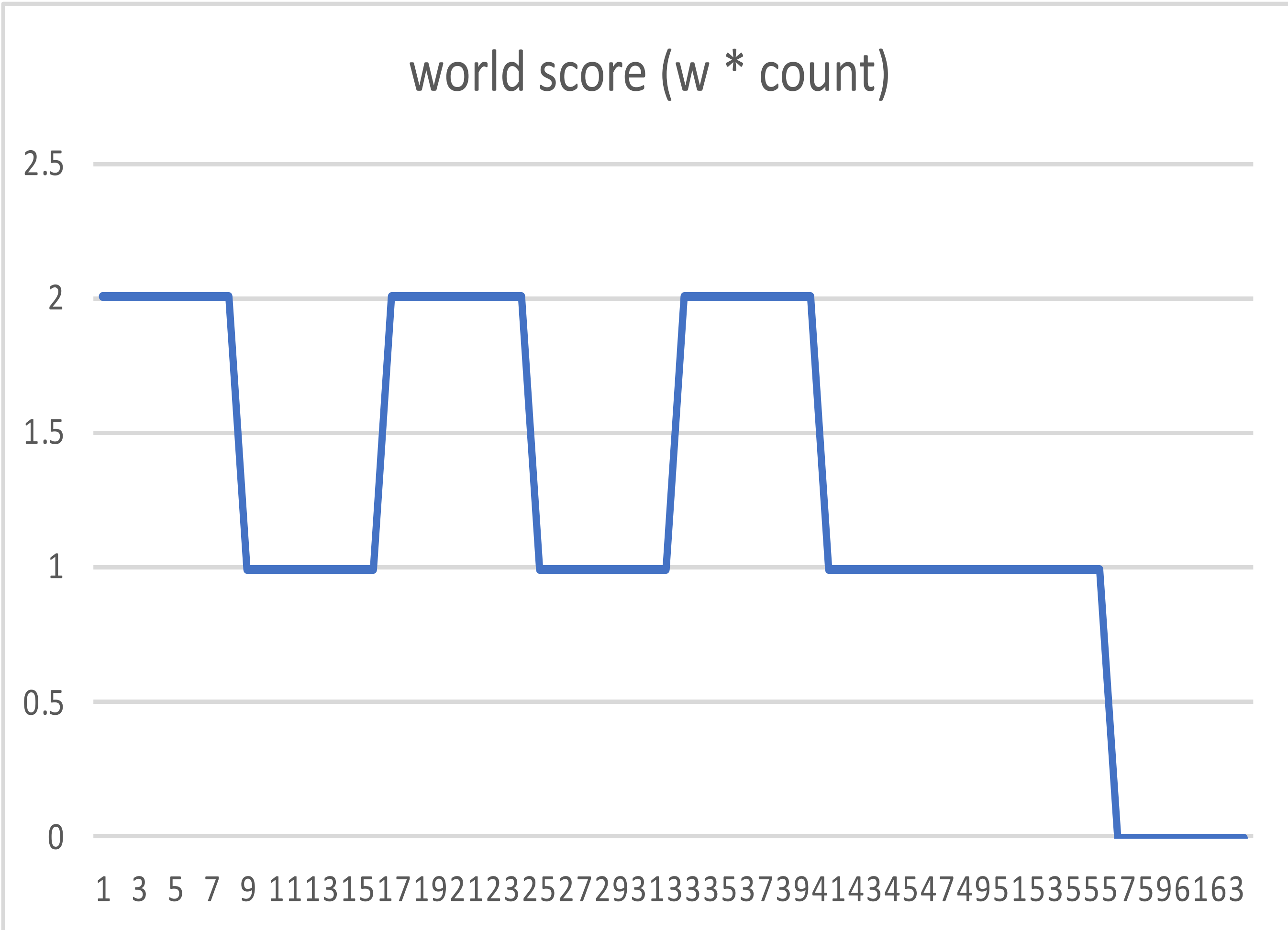


world score (w * count)
2.5
2
1.5
1
0.5
0
1
7
13
19
25
31
37
43
49
55
61
67
73
79
85
91
97
103
109
115
121
127
133
139
145
151
157
163
169
175
181
187
193
199
205
211
217
223
229
235
241
247
253

ALL WORLDS where B smokes



ALL WORLDS where B smokes and B is friends with A



QUANTIFYING OVER POSSIBLE WORLDS

Observations/measurements eliminate some of the possible worlds

Finding the mostly likely world can be computed with optimization

Computing the probability of any world requires us to aggregate/integrate over all possible worlds.

.

HOW DO WE MAKE SRL EFFICIENT?

There are 2 important dimensions to consider

- Brawn (i.e. the constant factors)
  - Latency hiding: memory hierarchy and network latencies (e.g. in memory)
  - Specialization: specialize for workload (e.g. JIT compilation), specialize for data
  - Parallelization: SIMD, multi-core, accelerators (e.g. GPU, TPU), in memory computing

- Brain (i.e. the asymptotics)
  - Lifting and Structure exploitation: algebraic (e.g. semi rings, groups), combinatorial, statistical, geometric
  - Approximation (with error bars): e.g. variational methods

both ---> approximate lifted inference

SUMMARY: ADVANTAGES OF (STATISTICAL) RELATIONAL AI

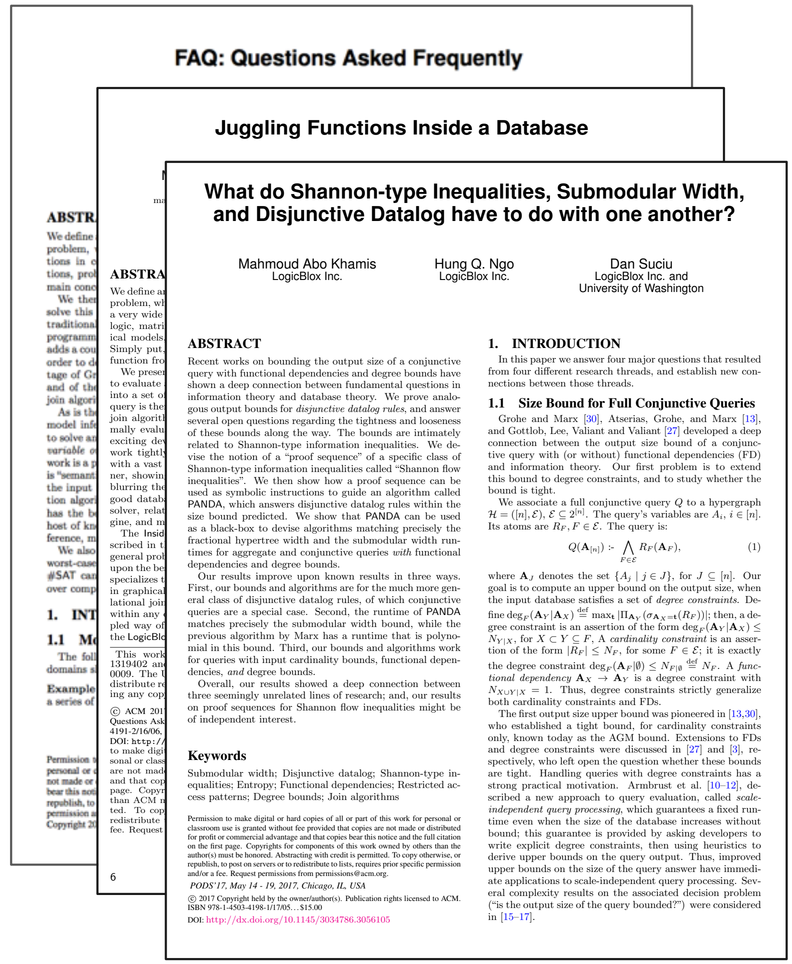
- Performance
  - Exploits the relational structure for asymptoticly better performance
- Understandability
  - Declarative relational language can be used to codify knowledge/expertise **(human to machine)** and to return insight **(machine to human)**l
- Quality
  - Fewer assumptions regarding independence, identical distributions, # of observations per example, etc. can produce more accurate models
- Versatility
  - Generalized inference: from observations to unknowns in any direction

WORST-CASE OPTIMAL MULTI-WAY JOIN

- Worst-Case Optimal Join Algorithms: Techniques, Results, and Open Problems. Ngo. (**Gems of PODS 2018**)
- Worst-Case Optimal Join Algorithms: Techniques, Results, and Open Problems. Ngo, Porat, Re, Rudra. (**Journal of the ACM 2018**)
- What do Shannon-type inequalities, submodular width, and disjunctive datalog have to do with one another? Abo Khamis, Ngo, Suciu, (PODS 2017 - **Invited to Journal of ACM**)
- Computing Join Queries with Functional Dependencies. Abo Khamis, Ngo, Suciu. (PODS 2017)
- Joins via Geometric Resolutions: Worst-case and Beyond. Abo Khamis, Ngo, Re, Rudra. (PODS 2015, **Invited to TODS 2015**)
- Beyond Worst-Case Analysis for Joins with Minesweeper. Abo Khamis, Ngo, Re, Rudra. (PODS 2014)
- Leapfrog Triejoin: A Simple Worst-Case Optimal Join Algorithm. Veldhuizen (ICDT 2014 - **Best Newcomer**)
- Skew Strikes Back: New Developments in the Theory of Join Algorithms. Ngo, Re, Rudra. (**Invited to SIGMOD Record** 2013)
- Worst Case Optimal Join Algorithms. Ngo, Porat, Re, Rudra. (PODS 2012 – **Best Paper**)

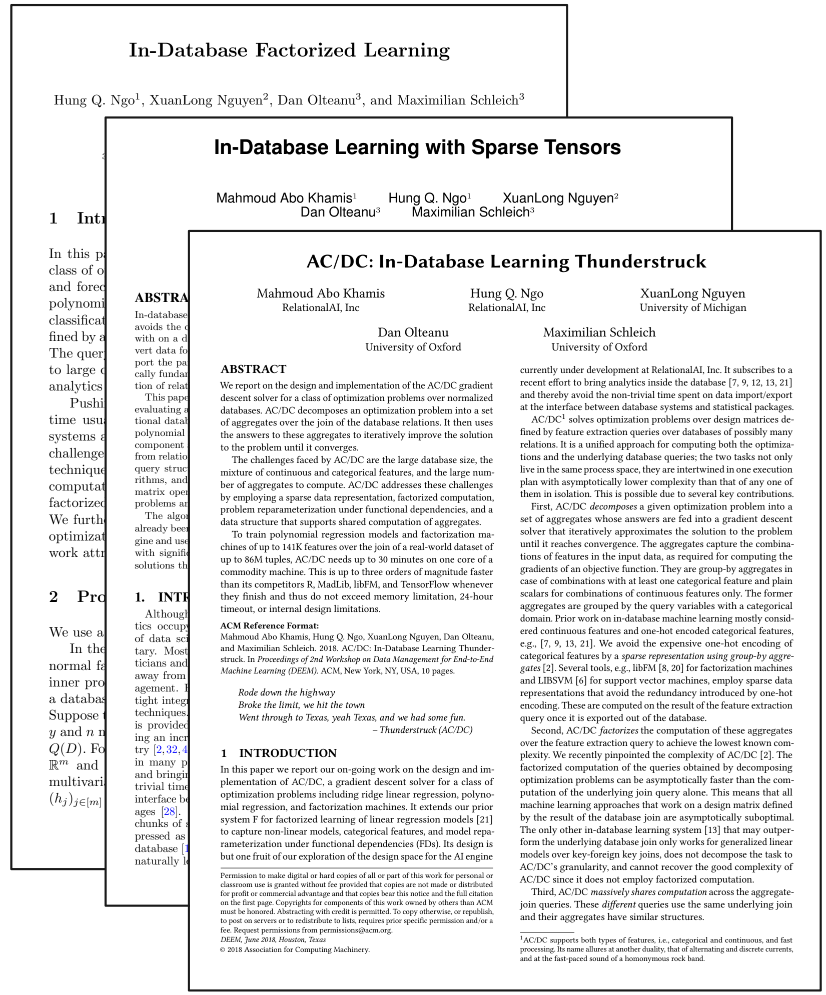
OPTIMAL QUERY PLANS FOR MULTI-WAY JOINS

- What do Shannon-type inequalities, submodular width, and disjunctive datalog have to do with one another? Abo Khamis, Ngo, Suciu, (PODS 2017 - **Invited to Journal of ACM**)
- FAQ: Questions Asked Frequently, Abu Khamis, Ngo, Rudra, (PODS 2016 – **Best Paper, Invited to Journal of ACM**).
- Juggling functions inside a database, Abo Khamis, Ngo, Suciu (**Invited to SIGMOD Record**)



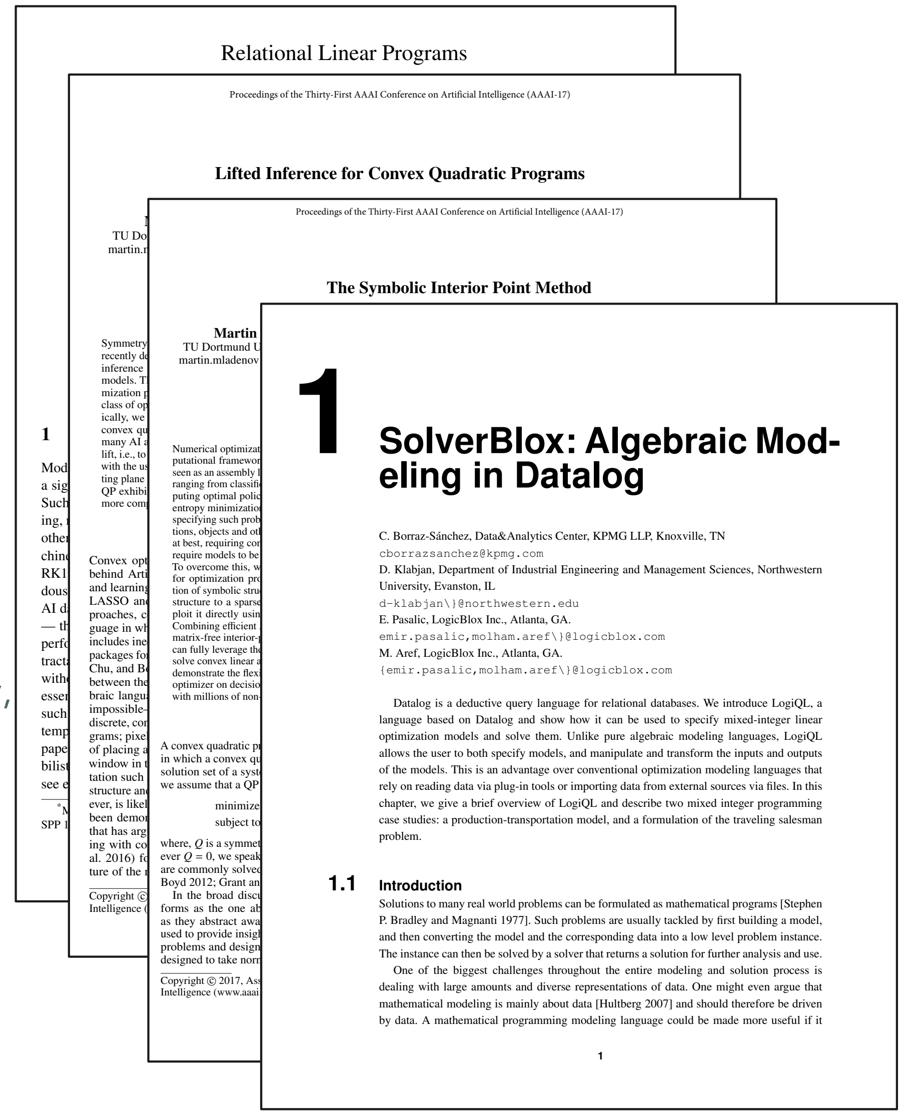
IN-DB LEARNING

- In-Database Learning with Sparse Tensors, Abo Khamis, Ngo, Nguyen, Olteanu, Schleich (PODS 2018)
- AC/DC: In-Database Learning Thunderstruck, Abo Khamis, Ngo, Nguyen, Olteanu, Schleich (DEEM 2018)
- Modelling Machine Learning Algorithms on Relational Data with Datalog. Makrynioti, Vasiloglou, Pasalic, Vassalos. (DEEM 2018)
- In-Database Factorized Learning, Ngo, Nguyen, Olteanu, Schleich (AMW 2017)
- Data Science with Linear Programming. Makrynioti, Vasiloglou, Pasalic, Vassalos. (DeLBP 2017)

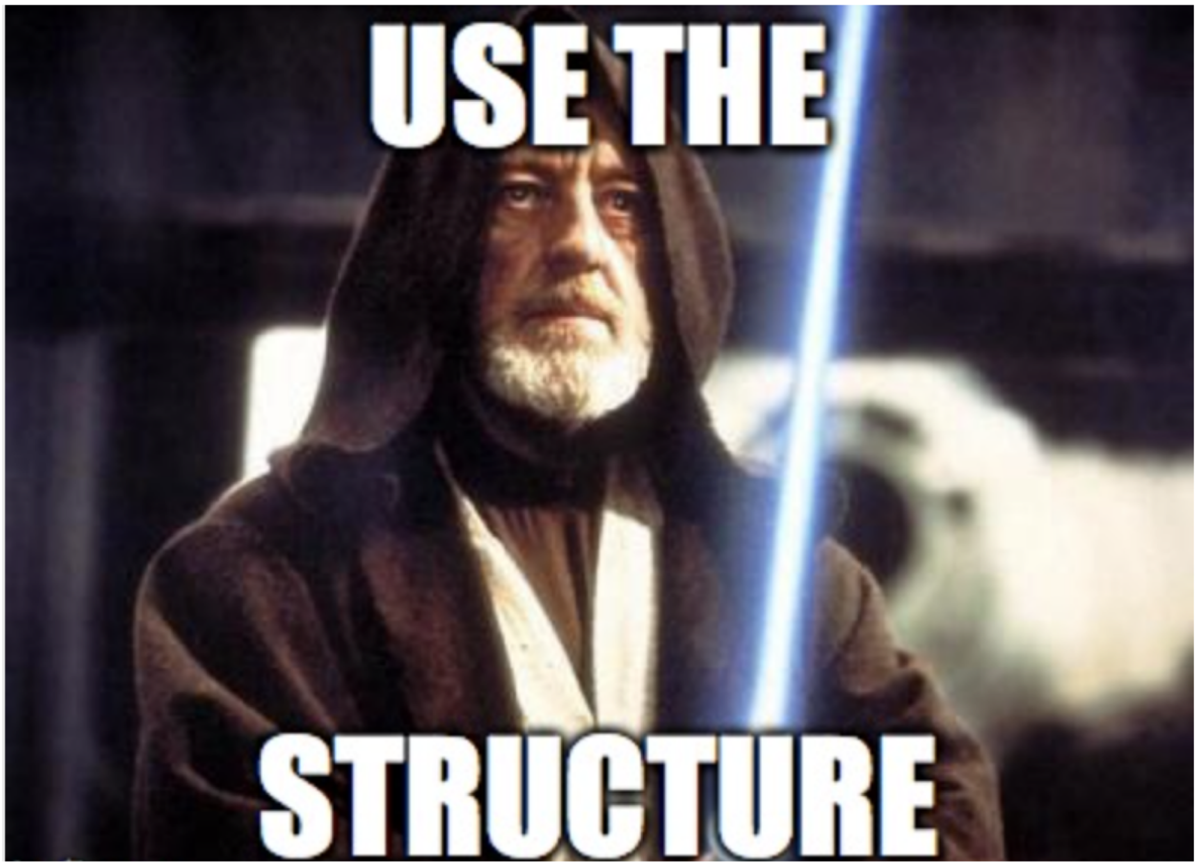


IN-DB & LIFTED LP/QP/IP SOLVERS

- SolverBlox: Algebraic Modeling in Datalog. Borraz-Sanchez, Klabjan, Pasalic, Aref. (Declarative Logic Progamming – Morgan & Claypool 2018)
- The Symbolic Interior Point Method. Mladenov, Belle, Kersting. (AAAI 2017)
- Lifted Inference for Convex Quadratic Programs. Mladenov, Kleinhans, Kersting. (AAAI 2017)
- RELOOP/ A Python-Embedded Declarative Language for Relational Optimization. Mladenov, Heinrich, Kleinhans, Gonsior, Kersting. (AAAI 2016)
- Relational Linear Programs. Kersting, Mladenov, Tokmanov. (2015)
- Lifted Linear Programming. Mladenov, Ahmadi, Kersting. (AISTATS 2012)



FOR A SMALL GROUP OF REBELS TO BEAT THE EMPIRE, WE HAVE TO…



FIN