Probabilistic Programming is great

TECHNISCHE UNIVERSITÄT DARMSTADT

Centre for Cognitive Science

Fachbereich Informatik

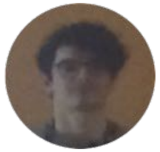
Kristian Kersting

Alejandro Molina
TU Darmstadt

Sriraam Natarajan
UT Dallas

Amir Globerson
U. Haifa

Robert Peharz
U. Cambridge



Antonio Vergari
MPI-IS

Christopher Re
Stanford

Zoubin Ghahramani
UBER AI Lab,
U. Cambridge

Isabell Valera
MPI-IS

Karl Stelzner
TU Darmstadt

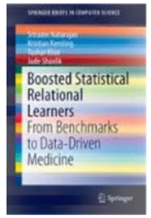
Martin Mladenov
Google Research

Martin Grohe
RWTH Aachen

… and many more

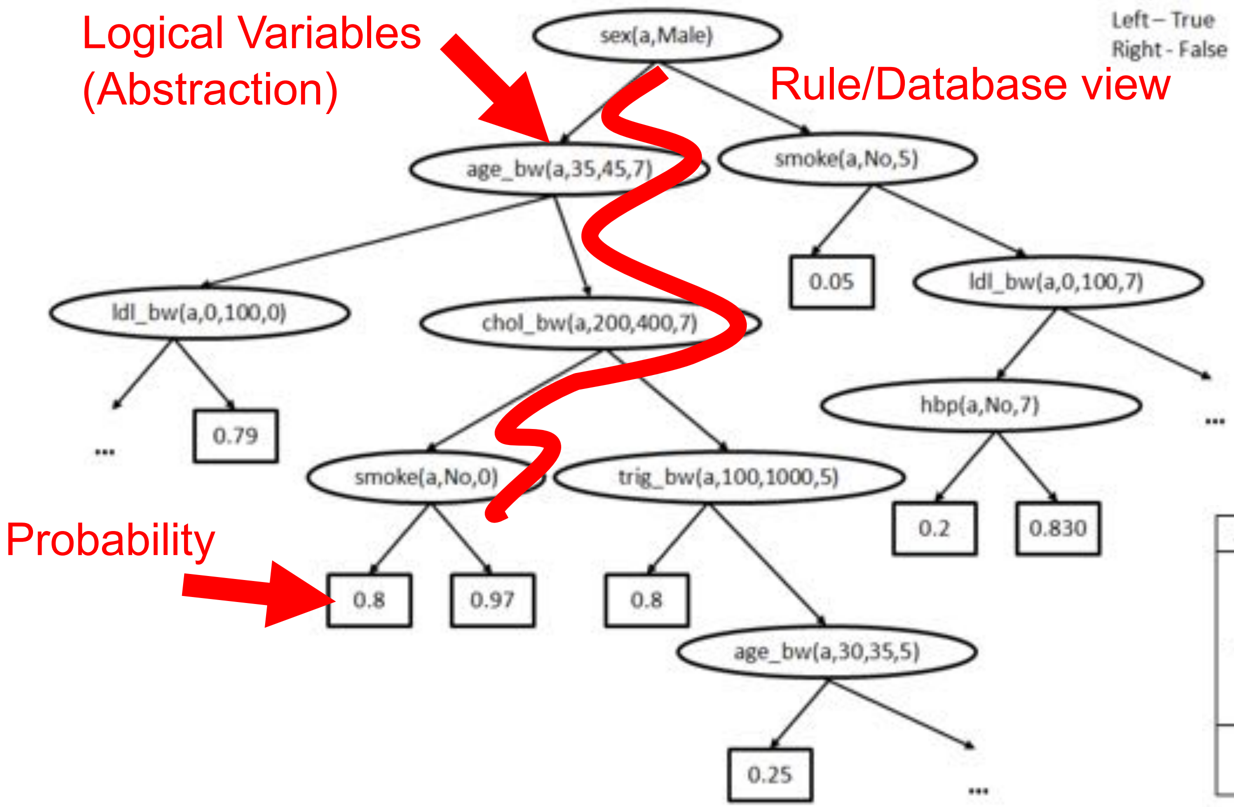
Probabilistic Programming is great

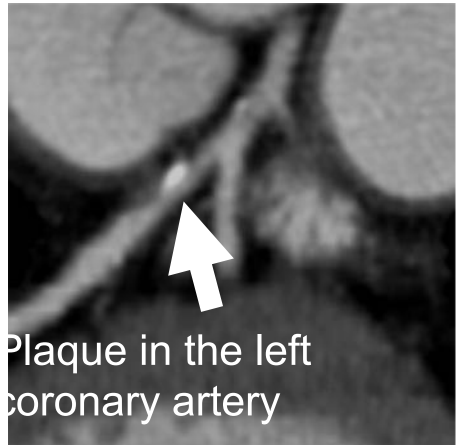
But what if you have to condition on a complete electronic health record (EHR)



ProbProg over relational DBs

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

[Kersting, Driessens ICML´08; Karwath, Kersting, Landwehr ICDM´08; Natarajan, Joshi, Tadepelli, Kersting, Shavlik. IJCAI´11; Natarajan, Kersting, Ip, Jacobs, Carr IAAI `13; Yang, Kersting, Terry, Carr, Natarajan AIME ´15; Khot, Natarajan, Kersting, Shavlik ICDM´13, MLJ´12, MLJ´15]

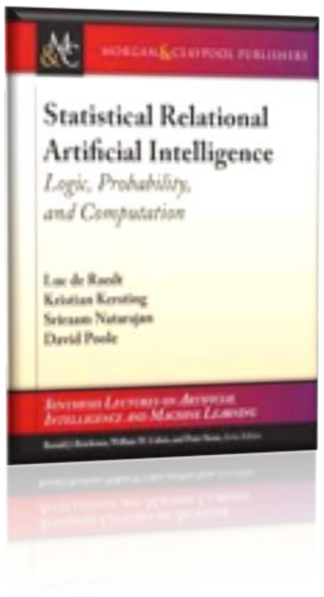
Neural Information Processing Systems
4. Dezember 2017 ·

NIPS 2017 Tutorial

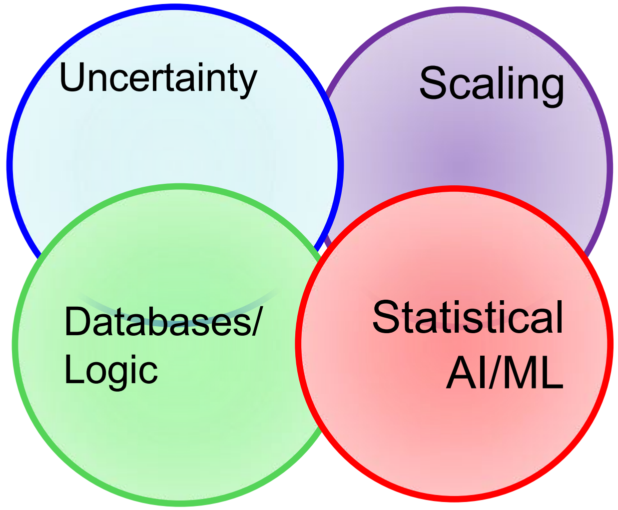
Now playing from NIPS 2017, Hall C, Statistical Relational Artificial Intelligence: Logic, Probability and Computation Tutorial.

3.075 Aufrufe

https://www.facebook.com/nipsfoundation/videos/now-playing-from-nips-2017-hall-c-statistical-relational-artificial-intelligence/1552222671535633/



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



https://starling.utdallas.edu/software/boostsrl/wiki/

StARLinG LAB

People Publications Projects Software Datasets Blog

BOOSTSRL BASICS

- Getting Started
- File Structure
- Basic Parameters
- Advanced Parameters
- Basic Modes
- Advanced Modes

ADVANCED BOOSTSRL

- Default (RDN-Boost)
- MLN-Boost
- Regression
- One-Class Classification
- Cost-Sensitive SRL
- Learning with Advice
- Approximate Counting
- Discretization of Continuous-Valued Attributes
- Lifted Relational Random Walks
- Grounded Relational Random Walks

APPLICATIONS

- Natural Language Processing

BoostSRL Wiki

BoostSRL (Boosting for Statistical Relational Learning) is a gradient-boosting based approach to learning different types of SRL models. As with the standard gradient-boosting approach, our approach turns the model learning problem to learning a sequence of regression models. The key difference to the standard approaches is that we learn relational regression models i.e., regression models that operate on relational data. We assume the data in a predicate logic format and the output are essentially first-order regression trees where the inner nodes contain conjunctions of logical predicates. For more details on the models and the algorithm, we refer to our book on this topic.

Sriraam Natarajan, Tushar Khot, Kristian Kersting and Jude Shavlik, Boosted Statistical Relational Learners: From Benchmarks to Data-Driven Medicine . SpringerBriefs in Computer Science, ISBN: 978-3-319-13643-1, 2015

Human-in-the-loop learning

Probabilistic Programming is great

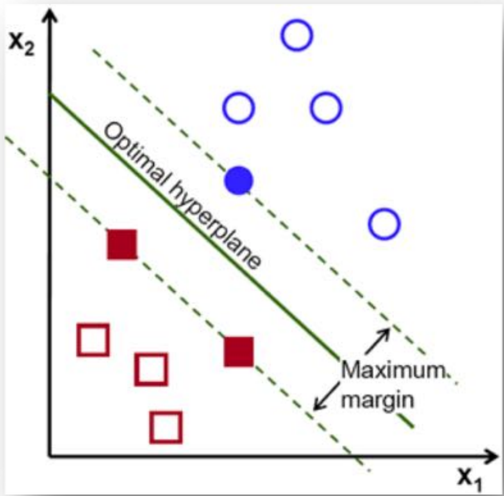
But what if you want to feed a relational DB into a Support Vector Machine?

Standard SVM

$$\min_{\mathbf{w},b,\boldsymbol{\xi}} \; \mathcal{P}(\mathbf{w},b,\boldsymbol{\xi}) = \frac{1}{2}\mathbf{w}^2 + C\sum_{i=1}^{n}\xi_i$$

$$\text{subject to} \quad \begin{cases} \forall i \quad y_i(\mathbf{w}^\top\Phi(\mathbf{x}_i)+b) \geq 1-\xi_i \\ \forall i \quad \xi_i \geq 0 \end{cases}$$

Support Vector Machines
Cortes, Vapnik MLJ 20(3):273-297, 1995



[Mladenov, Kleinhans, Kersting AAAI 2017]

High-level programming of QPs, using a few lines of code!

high-level AI / ML programming languages

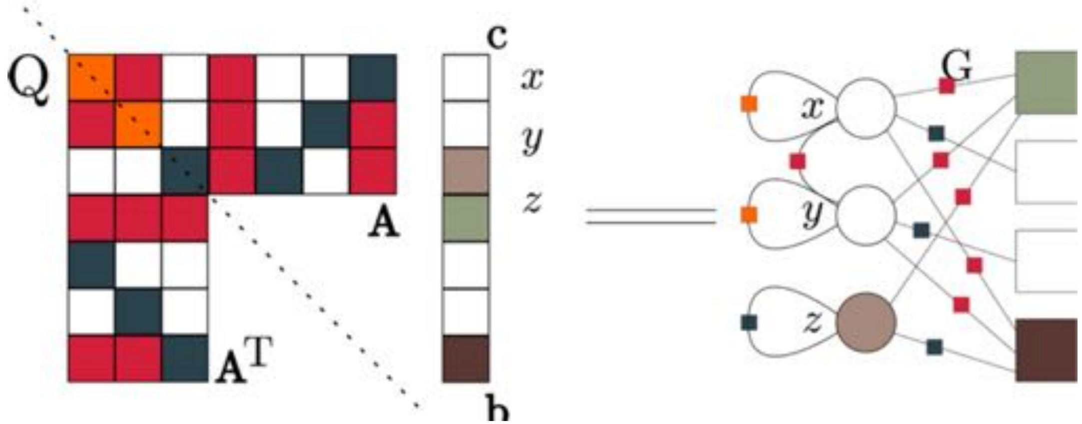
```
#QUADRATIC OBJECTIVE
minimize: sum{J in feature(I,J)} weight(J)**2 + c1 * slack + c2 * coslack;

#labeled examples should be on the correct side
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;

#TRANSDUCTIVE PART
#cited instances should have the same labels.
subject to forall {I1, I2 in linked(I1, I2)}: labeled(I1) * predict(I2) >= 1 - slack(I1, I2);
subject to forall {I1, I2 in linked(I1, I2)}: coslack(I1, I2) >= 0; #coslacks are positive
```

Citing papers should be on the same side of the hyperplane

Reduce the QP by running Weisfeiler-Lehman in quasi-linear time

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

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



Compilers for AI / ML languages

Grohe, Kersting, Mladenov, Selman ESA´14, Kersting, Mladenov, Tokmatov AIJ´17, Mladenov, Kleinhans, Kersting AAAI´17

Statistical ML within relational DBs

Collective Classification

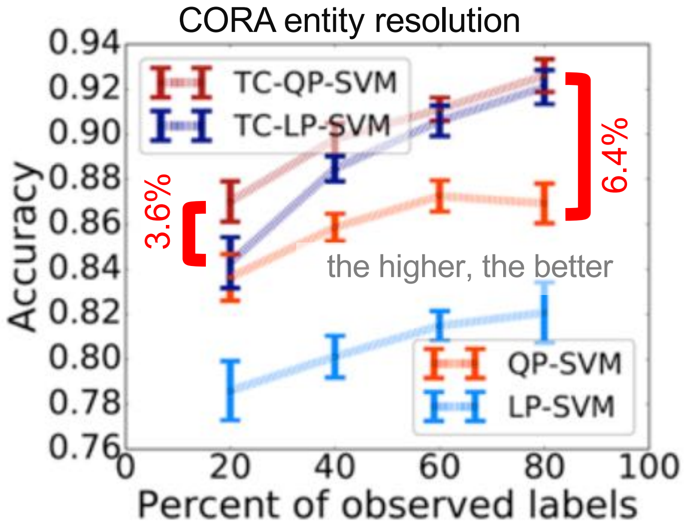
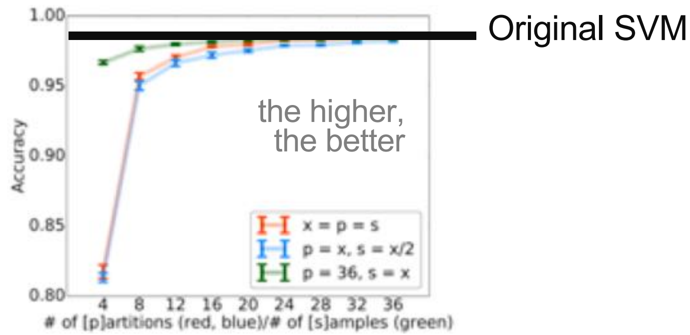
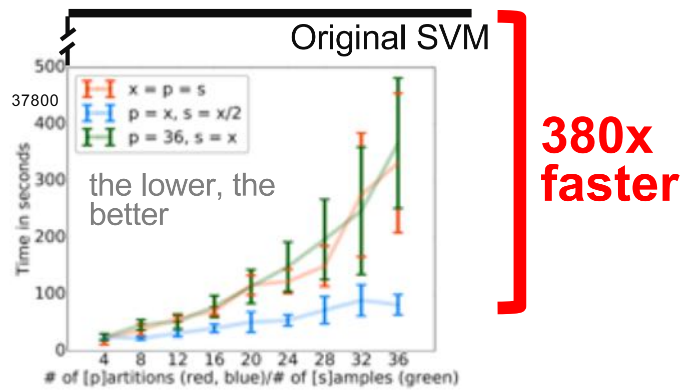


Image Classification

MNIST image classification with label-preserving data programming





(d) Learning time: Translation

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

Faster than state-of-the-art

reloop

- Overview
- Source
- Commits
- Branches
- Pull requests
- Pipelines
- Issues
- Downloads

Reloop

1. Prequisites as in requirements.txt

- Reloop requires Python 2.7+
- Scipy v0.15+
- Numpy v1.9.1+
- Cython v0.21.1+
- Cvxopt v1.1.7+
- Picos v1.0.1+
- infix v1.0.0+
- Ordered-Set v1.3.1+
- pyDatalog v0.14.6
- sympy v0.7.6+
- psycopg2 v.2.6.1+
- problog v.2.1.0.5+

Embedded within Python

RELOOP: A Toolkit for Relational Convex Optimization

https://bitbucket.org/reloopdev/reloop

If pip is available all prequisites can be installed at once by running

```
$ pip install -r requirements.txt --upgrade
```

1.1 Optional Dependencies

These optional dependencies enable additional knowledge bases for usage. While Problog and SWI-Prolog both interface Prolog, psycopg2 interface a postgres database.

- Problog v2.1+
- Psycopg2 v2.6.1+
- SWI-Prolog

2. Installation

Once all the prequisites have been installed simply run

```
python setup.py build_ext --inplace
```

followed by either

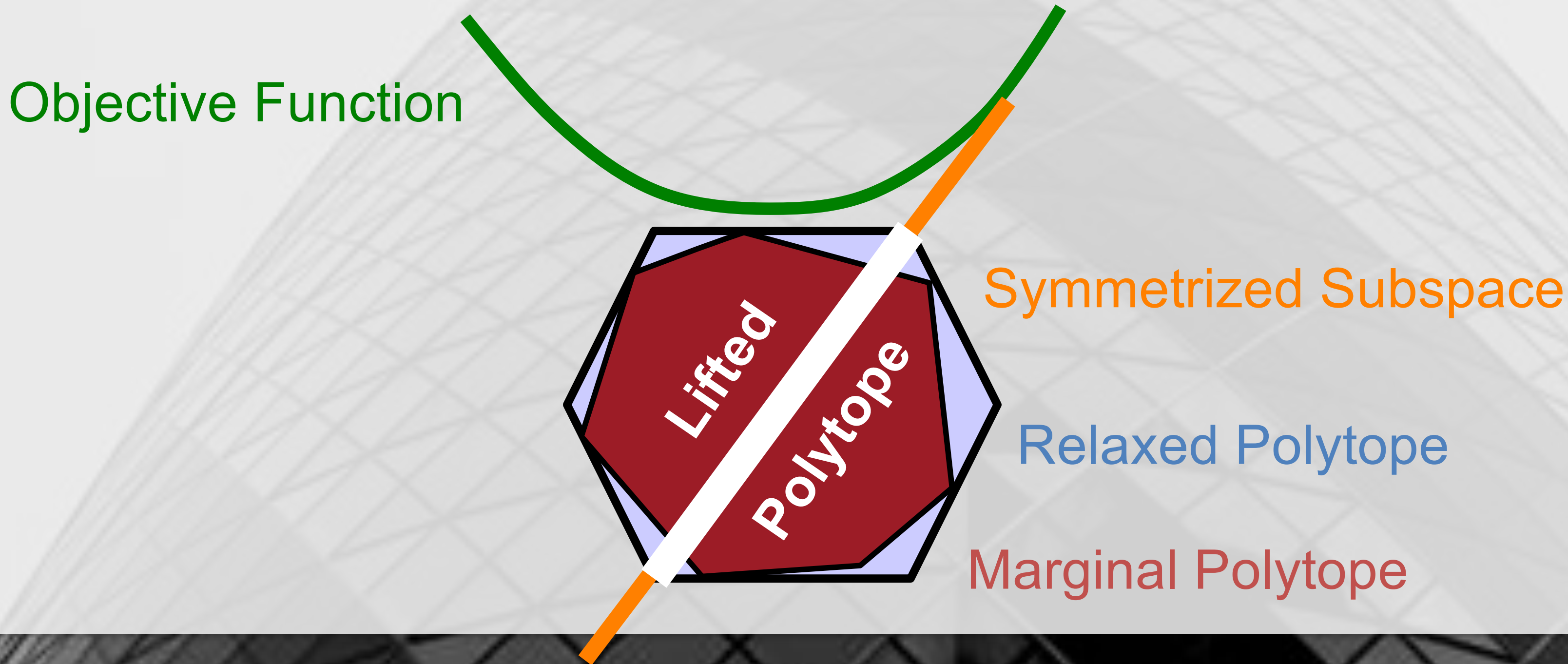
```
python setup.py install
```

or

[Mladenov, Globerson, Kersting UAI ´14, AISTATS ´14, Mladenov, Kersting UAI ´15]

Algebraic approach for exploiting symmetries within probabilistic inference

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



WL induces <u>Fractional Automorphisms</u> of Linear and Quadratic Programs

[Mladenov, Globerson, Kersting UAI ´14, AISTATS ´14, Mladenov, Kersting UAI ´15]

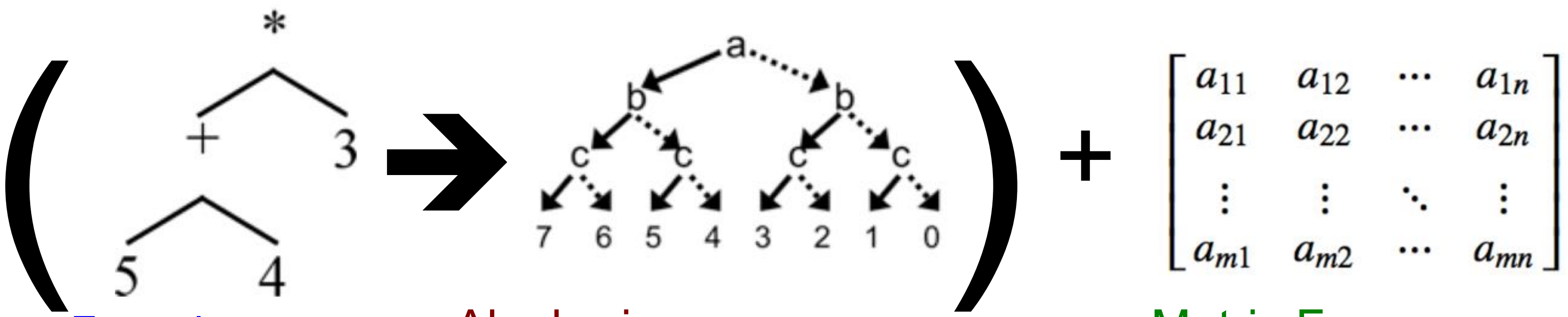
Algebraic approach for exploiting symmetries within probabilistic inference

Shows that all probabilistic inference approaches that relay on LPs/QPs such as MPLP, concave energies, … can exploit symmetries

WL induces <u>Fractional Automorphisms</u> of Linear and Quadratic Programs

Other „-O“ flags lying ahead
New field: Symbolic-numerical AI

TECHNISCHE UNIVERSITÄT DARMSTADT



Formulae parse trees

Algebraic Decision Diagrams

Matrix Free Optimization

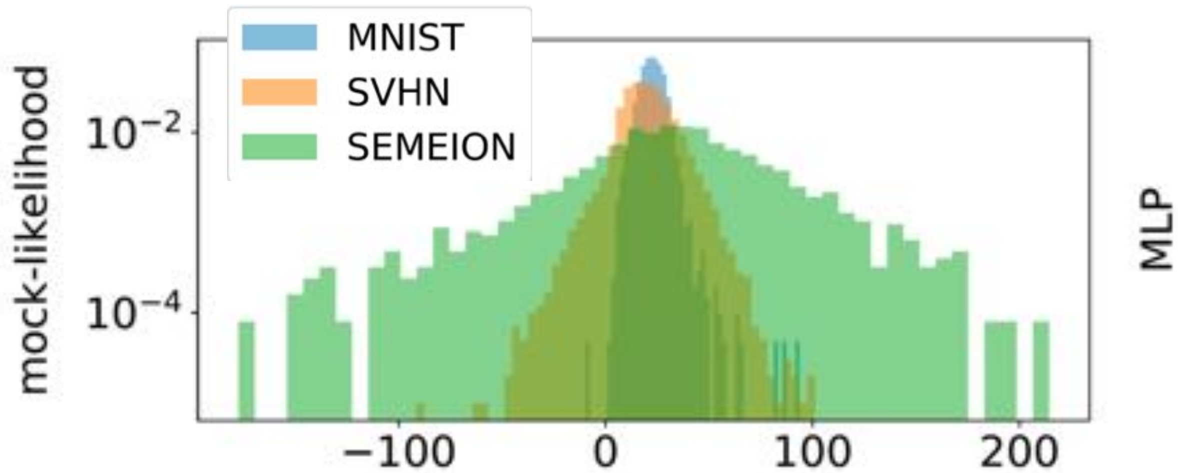
Problem Statistics				Symbolic IPM		Ground IPM
name	#vars	#constr	$nnz(A)$	|ADD|	time[s]	time[s]
factory	131.072	688.128	4.000.000	1819	6899	**516**
factory0	524.288	2.752.510	15.510.000	1895	**6544**	7920
factory1	2.097.150	11.000.000	59.549.700	2406	**34749**	159730
factory2	4.194.300	22.020.100	119.099.000	2504	**36248**	$\geq$ 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 methods are represented, each having an effect on the quality of the job, and each requiring tools. Rewards (required quality) range from 0 to 10 and a discounting factor of 0. 9 was used used

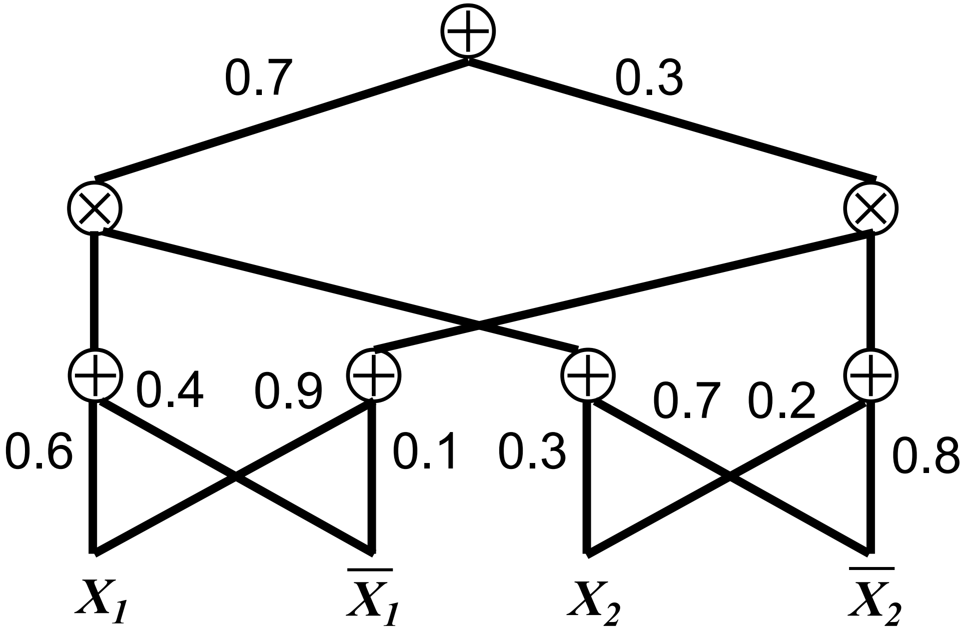
Probabilistic Programming is great

But what if you are not a statistician?



Deep neural networks may not be faithful probabilistic models

Deep Probabilistic Modelling using Sum-Product Networks



Computational graph (kind of TensorFlow graphs) that encodes how to compute probabilities

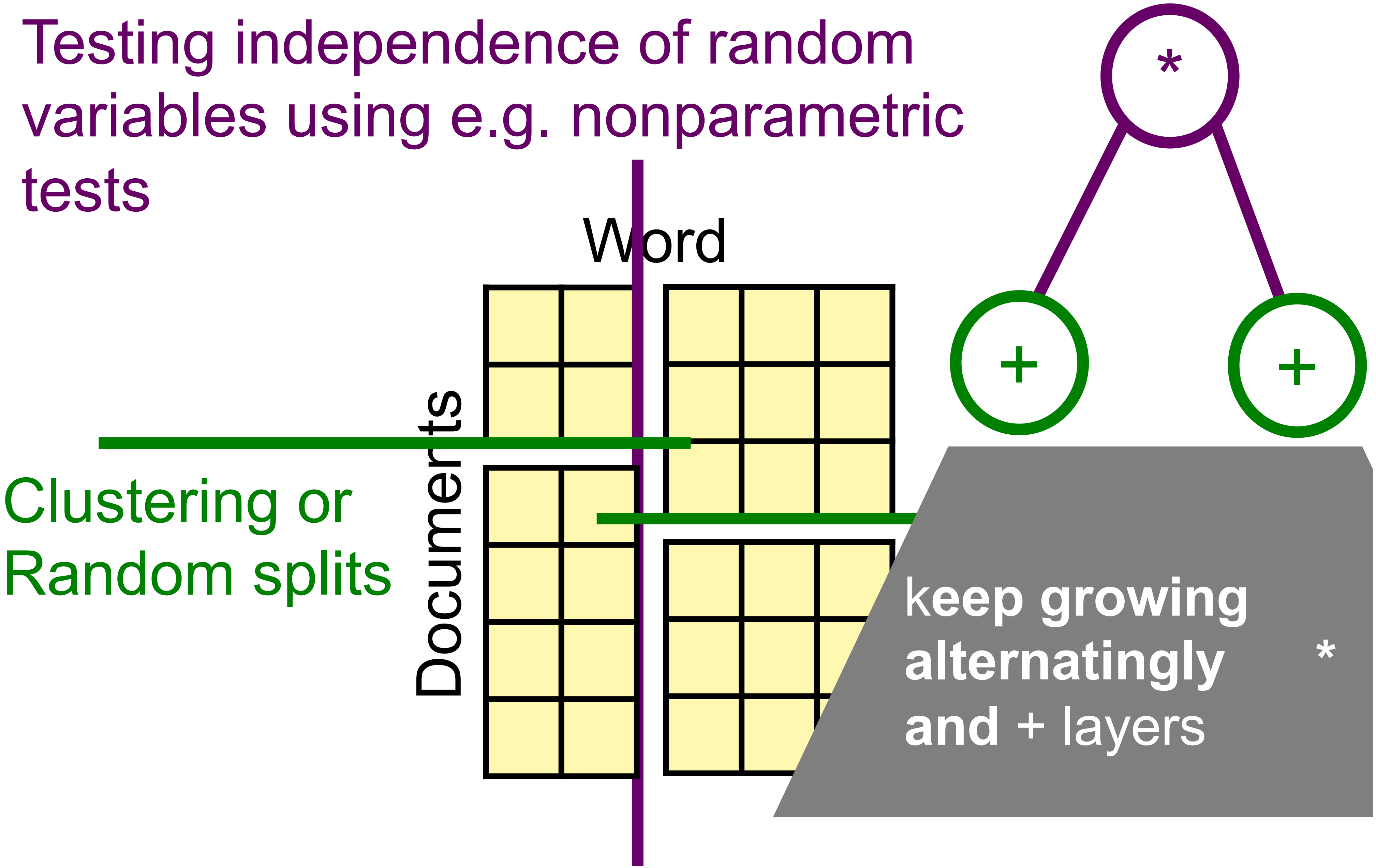
Inference is Linear in Size of Network

*SPNs are an instance of Arithmetic Circuits (ACs). ACs have been introduced into the AI literature more than15 years ago as a tractable representation of probability distributions
[Darwiche CACM 48(4):608-647 2001]

[Poon, Domingos UAI'11; Molina, Natarajan, Kersting AAAI`17, Molina et al AAAI ,18]

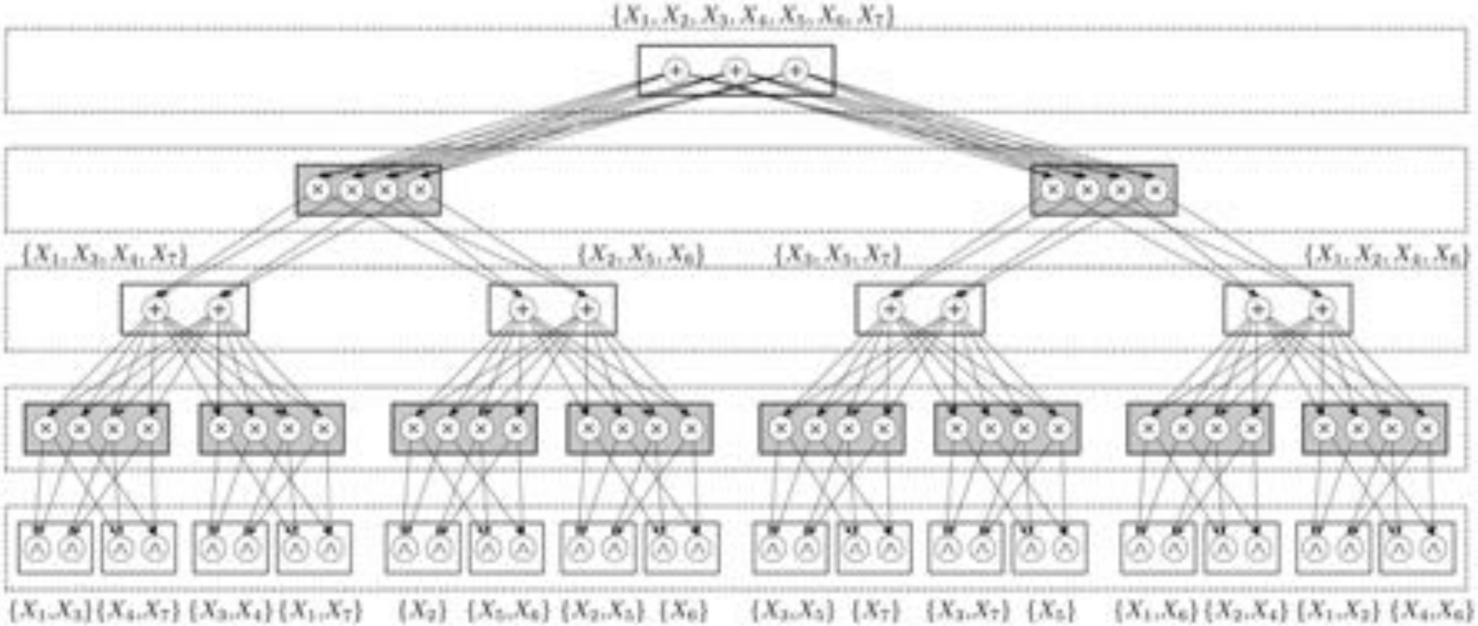
Greedy structure learning

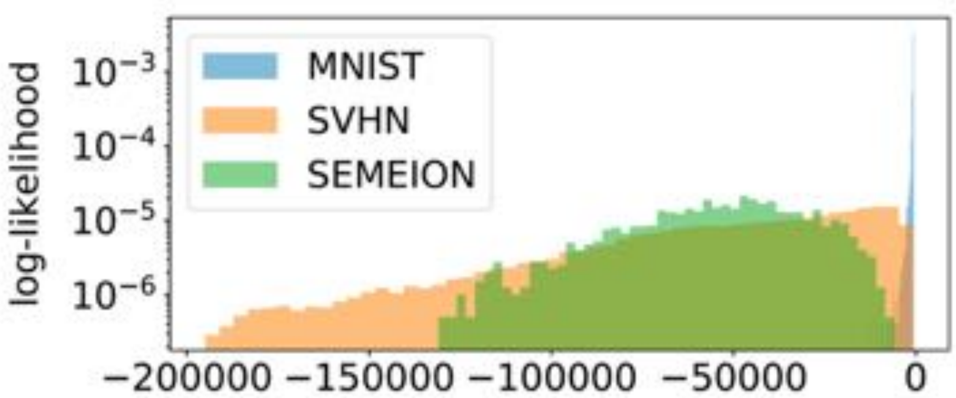
Testing independence of random variables using e.g. nonparametric tests



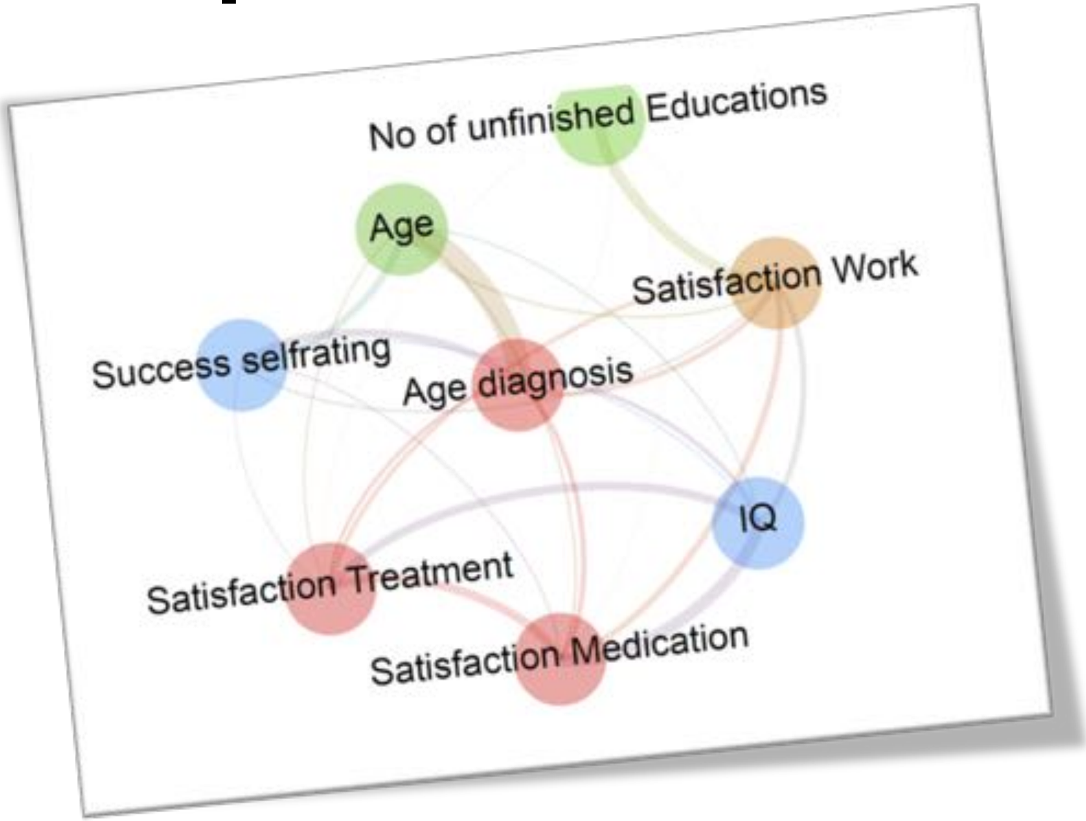
Random sum-product networks

[Peharz, Vergari, Molina, Stelzner, Trapp, Kersting, Ghahramani UDL@UAI 2018]

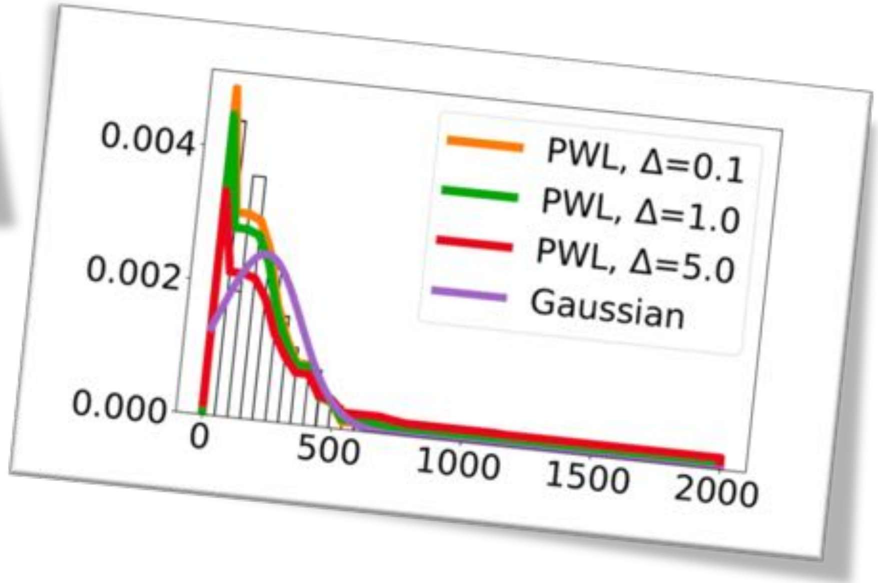




Distribution-agnostic Deep Probabilistic Learning



Use nonparametric independency tests and piece-wise linear approximations



[Molina, Natarajan, Vergari, Di Mauro, Esposito, Kersting AAAI 2018]

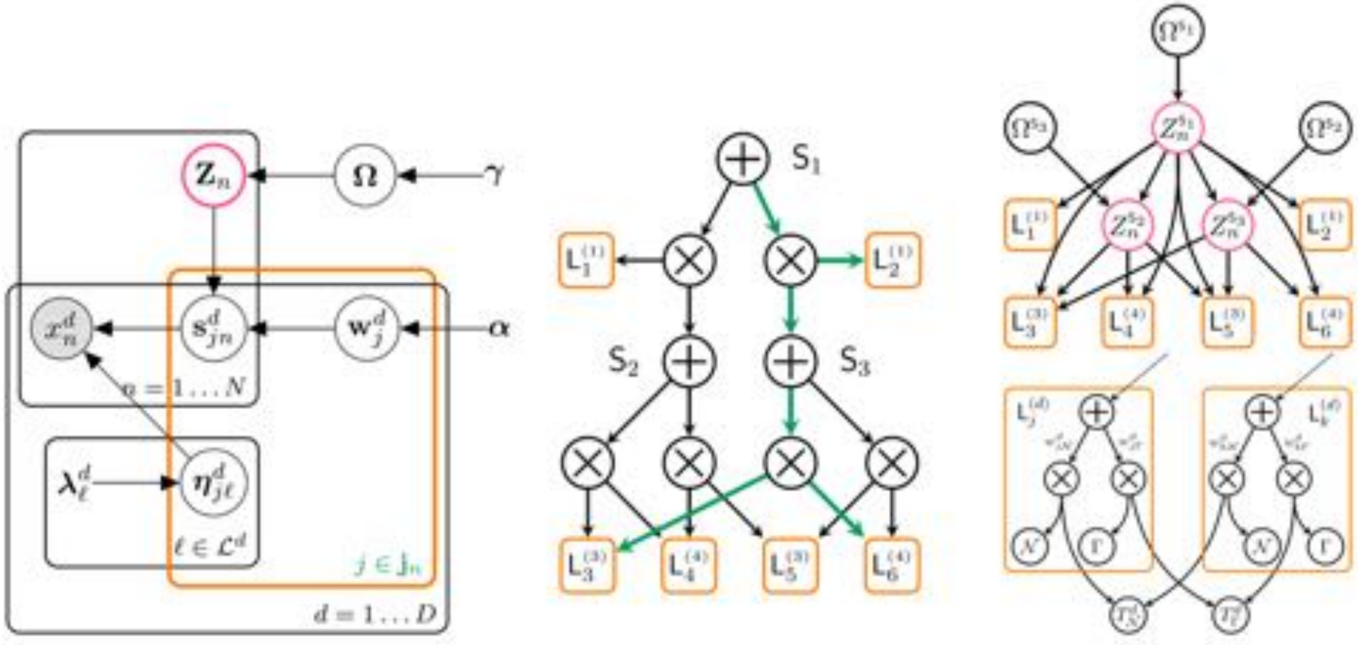
Distribution-agnostic Deep Probabilistic Learning

Use nonparametric independency tests and piece-wise linear approximations

However, we have to provide the statistical types and do not gain insights into the parametric forms of the variables. **Are they Gaussians? Gammas? …**

[Vergari, Molina, Peharz, Ghahramani, Kersting, Valera TPM@ICML 2018]

Automatic Bayesian Density Analysis



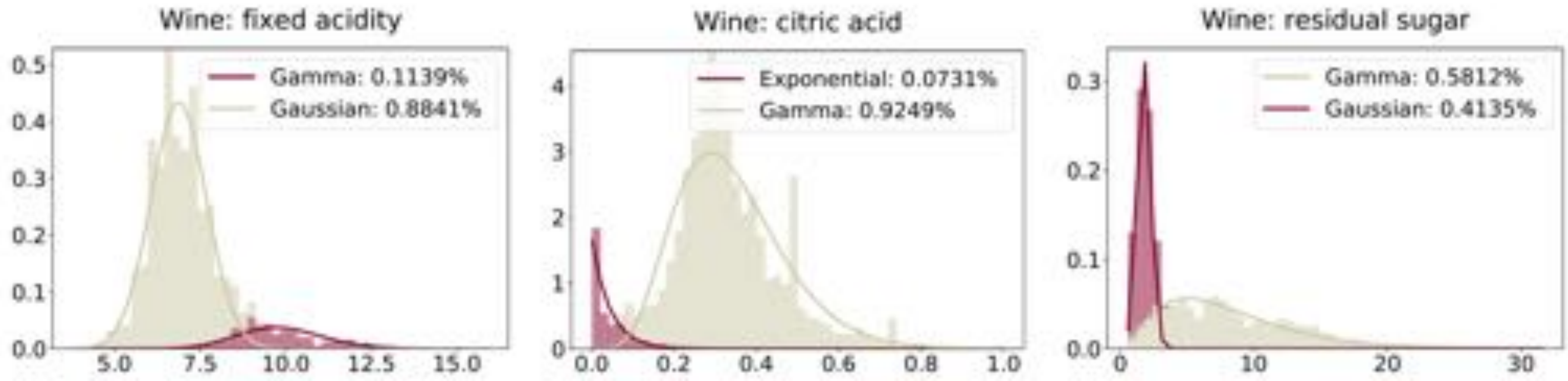
(a) Graphical model (b) SPN (c) Type-augmented SPN

Bayesian discovery of statistical types and parametric forms of variables

+

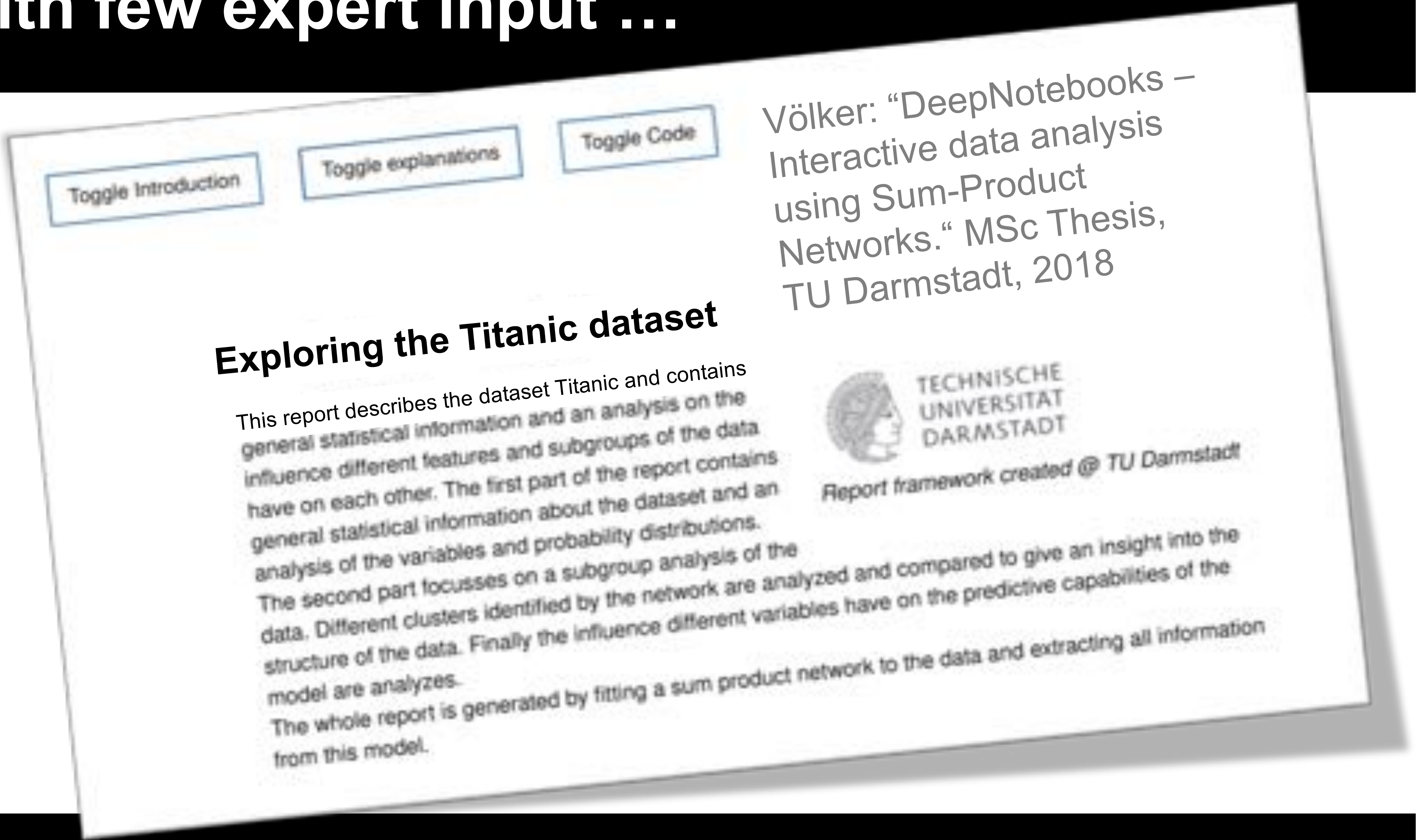
Type-agnostic deep probabilistic learning

Automatic Bayesian Density Analysis



… can automatically discovers the statistical types and parametric forms of the variables

The machine understands the data with few expert input …



Völker: "DeepNotebooks – Interactive data analysis using Sum-Product Networks." MSc Thesis, TU Darmstadt, 2018

…and can compile data reports automatically

SPFlow: An Easy and Extensible Library for Sum-Product Networks

[Molina, Vergari, Stelzner, Peharz, Kersting 2018]

https://github.com/alejandromolinaml/SPFlow

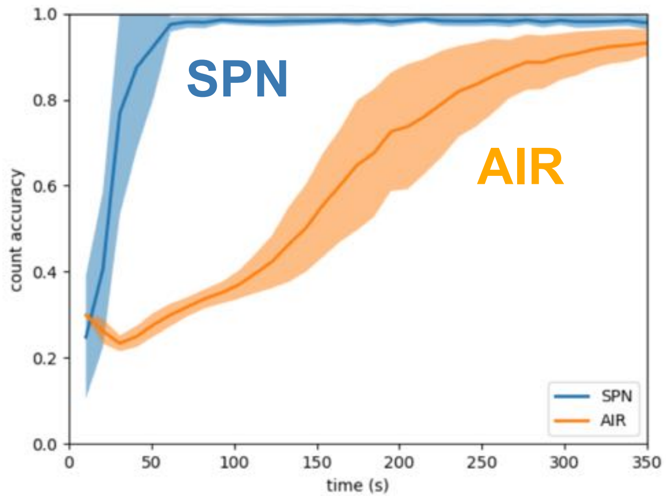
Compiles SPNs into flat, library-free code even suitable for running on devices: C/C++,GPU, FPGA [Sommer et al ICDD 2018]

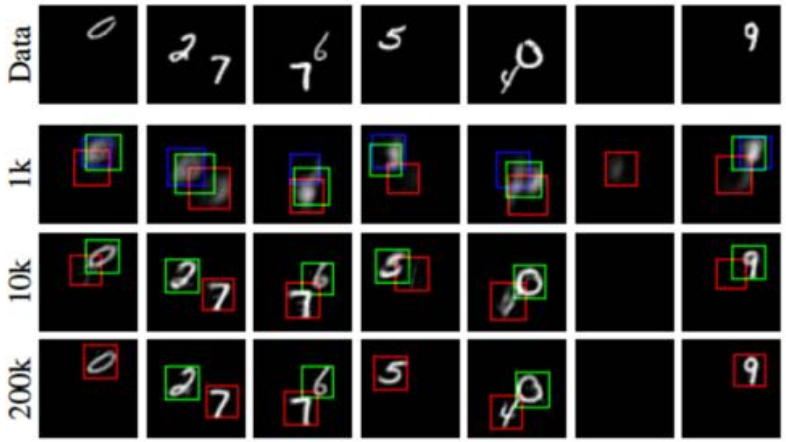
PixelSPNs

[Shao, Molina, Kersting 2018]

SPN AIR

[Stelzner, Peharz, Kersting 2018]





Next Steps

TECHNISCHE UNIVERSITÄT DARMSTADT

Symmetry-aware Deep Probabilistic Learning

Open-universe Mathematical Programming

Sum-Product Probabilistic Programming

Thanks

TECHNISCHE UNIVERSITÄT DARMSTADT

RelationalAI and Apple, among others, have invested hundreds of millions of US dollars

IBM CPLEX

GUROBI OPTIMIZATION

And it appears in industrial strength solvers such as CPLEX and GUROBI

frontiers in Big Data | Machine Learning and Artificial Intelligence