technische universität
dortmund

SFB 876 - Providing Information by
Resource-Constrained Data Analysis

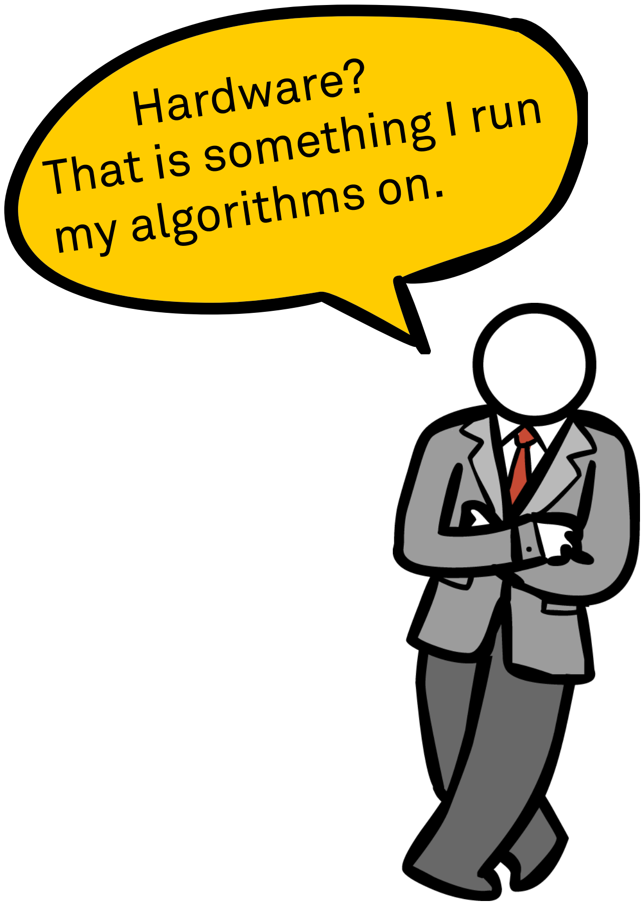
Machine Learning under Resource Constraints

Katharina Morik,
Computer Science 8, TU Dortmund University

Overview

- Machine learning and hardware
- Probabilistic graphical models
  - Spatio-temporal random fields
  - Integer Markov random fields
  - Stochastic Discrete Clenshaw Curtis Quadrature

Machine learning and hardware -- 1



- MatLab takes care:
  - distributes operations over cores,
  - executes for-loops in parallel,
  - executes on Hadoop,
  - exploits arrays for GPU,
  - generates code for FPGA
- Machine learning algorithms are independent of their execution platform.
- Compilers might offer specialized functions, e.g., matrix operations.

Good old days

- Machine learning algorithms were implemented on *some* computer.
- Data structures and algorithms were evaluated concerning runtime and memory consumption.

- Tests were run on a PC with a 1.8 GHz Intel P4 processor and 1 Gbytes of RAM. The operating system was Debian Linux (kernel version: 2.4.24). (Bodon 2004)
- We performed the experiments on a PC AMD Athlon™ XP 2000+ 1.6 GHz, 1 GB RAM, 2 GB Swap with 40GB Hard Disk running Fedora Core 1.... using g++ compiler. (Sucahyo}
- The experiments were conducted on a Windows XP PC equipped with a 2.8GHz Pentium IV and 512MB of RAM memory. (Lucchese et al. 2004)

Good old days

- Machine learning algorithms were implemented on *some* computer.
- Data structures and algorithms were evaluated concerning runtime and memory consumption.
- Tests were run on a PC with a 1.8 GHz Intel P4 processor and 1 Gbytes of RAM. The operating system was Debian Linux (kernel version: 2.4.24). (Bodon 2004)
- We performed the experiments on a PC AMD Athlon™ XP 2000+ 1.6 GHz, 1 GB RAM, 2 GB Swap with

Frequent Itemset Mining Implementations Repository

Home | Implementations | Datasets | Experiments | FIMI'03 | FIMI'04

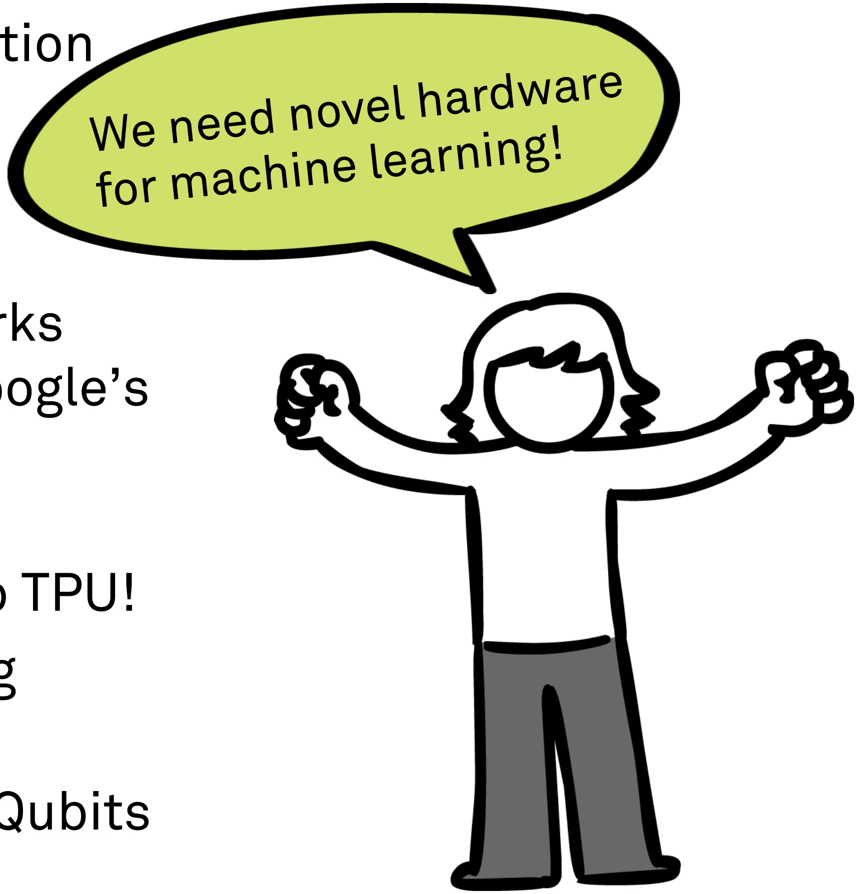
This repository is the result of the workshops on Frequent Itemset Mining Implementations, FIMI'03 and FIMI'04 which took place at IEEE ICDM'03, and IEEE ICDM'04 respectively.

This website serves as the FIMI repository containing the source codes of all implementations that were accepted at the FIMI workshops together with several publicly available datasets.

Bart Goethals

Machine learning and hardware -- 2

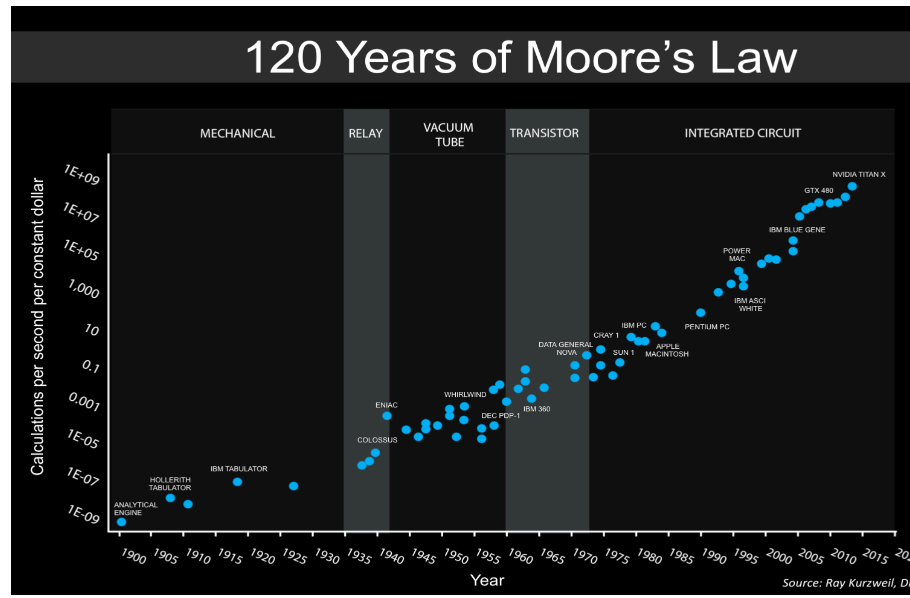
- Von Neumann bottleneck: instruction fetch and data operation sharing a bus.
  - New coprocessor: shared memory GPU!
- In 2013, with deep neural networks the computation demands on Google's data centers doubled.
  - Even newer coprocessor: inference by customized chip TPU!
- Intel: *Lake Crest* chip for learning
- Quantum computing (D-Wave): processor with more than 1 000 Qubits for fast optimization.



P. Dubey (2017)"The quest for the ultimate learning machine"

Moore’s law and other exponential trends

- The complexity for minimum component costs has increased at a rate of roughly a factor of two per year.
  Gordon E. Moore (1965)
- Software is getting slower more rapidly than hardware becomes faster.
  Niklaus Wirth (1995)
- An updated version of Moore's Law over 120 Years: calculations per second per constant Dollar. The 7 most recent data points are all NVIDIA GPUs.



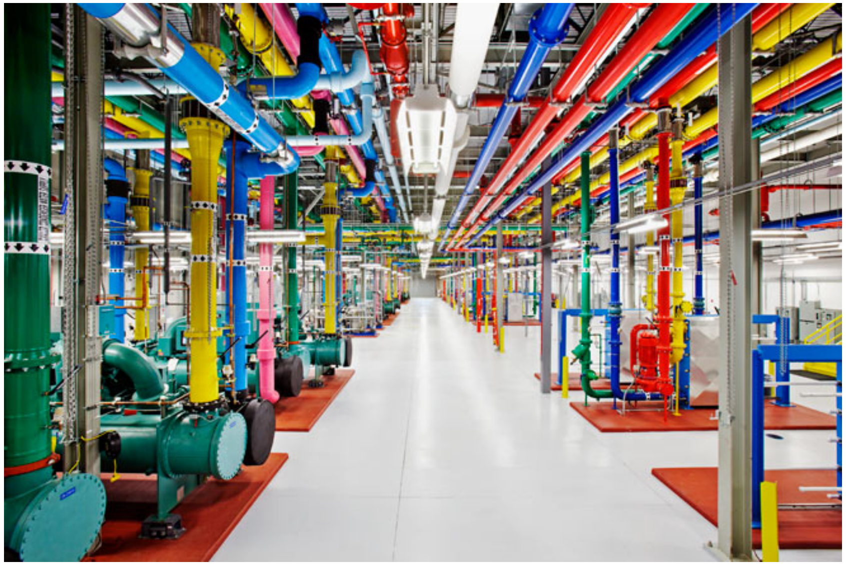
By Steve Jurvetson - https://www.flickr.com/photos/jurvetson/31409423572/

Resource restrictions energy and cooling

- Google's total yearly energy consumption is 2 terawatt hours (2024 watt hours).
  - 1 search request consumes 0.3 watt hours.
  - Asking and reading the result at a PC consumes about the same.

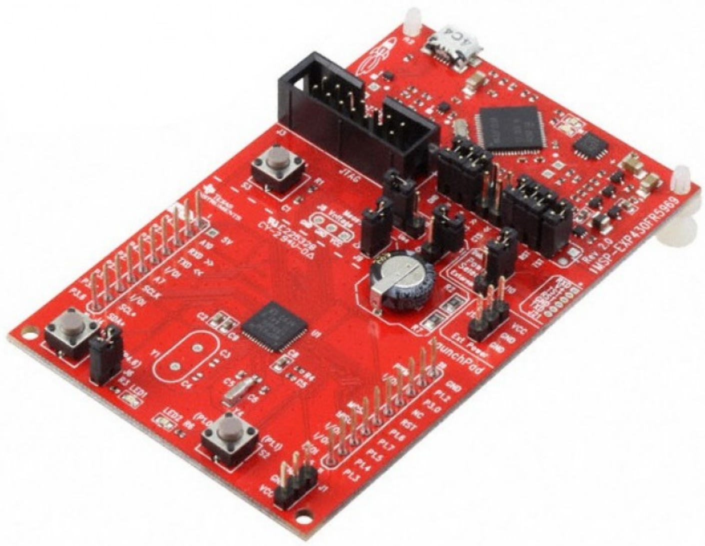
European Network of Excellence in Internet Science, report in Ubiquity June, 2015

Google Cooling, Georgia



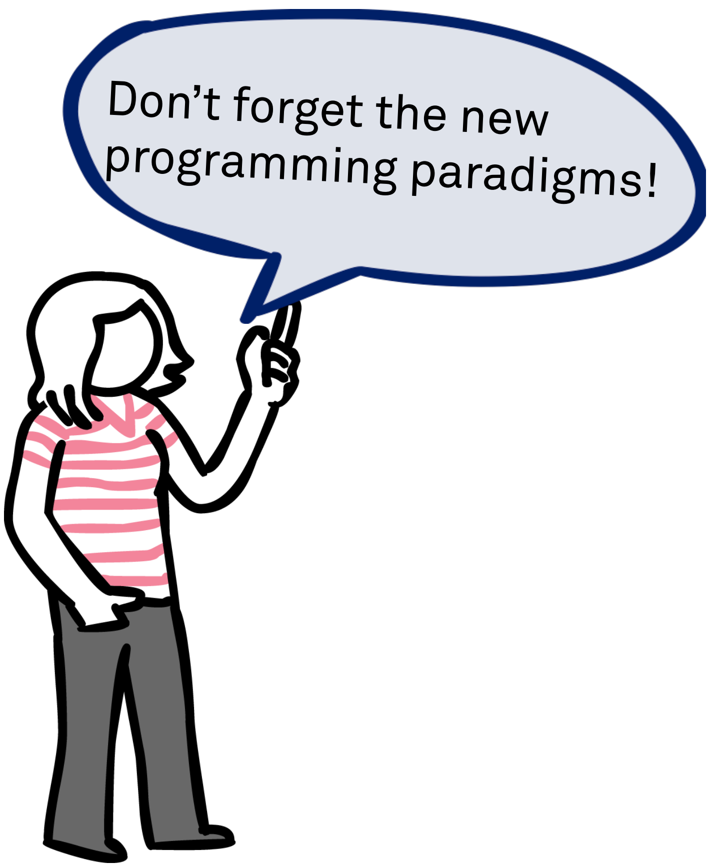
Ultra-low power microcontrollers

- Slow 16 MHz
- Small wordsize 16 Bit
- Small memory 64 Kb
- Restricted capabilities, no floating point unit
- Connectable to multiple sensors
- Energy around 0,0048 W



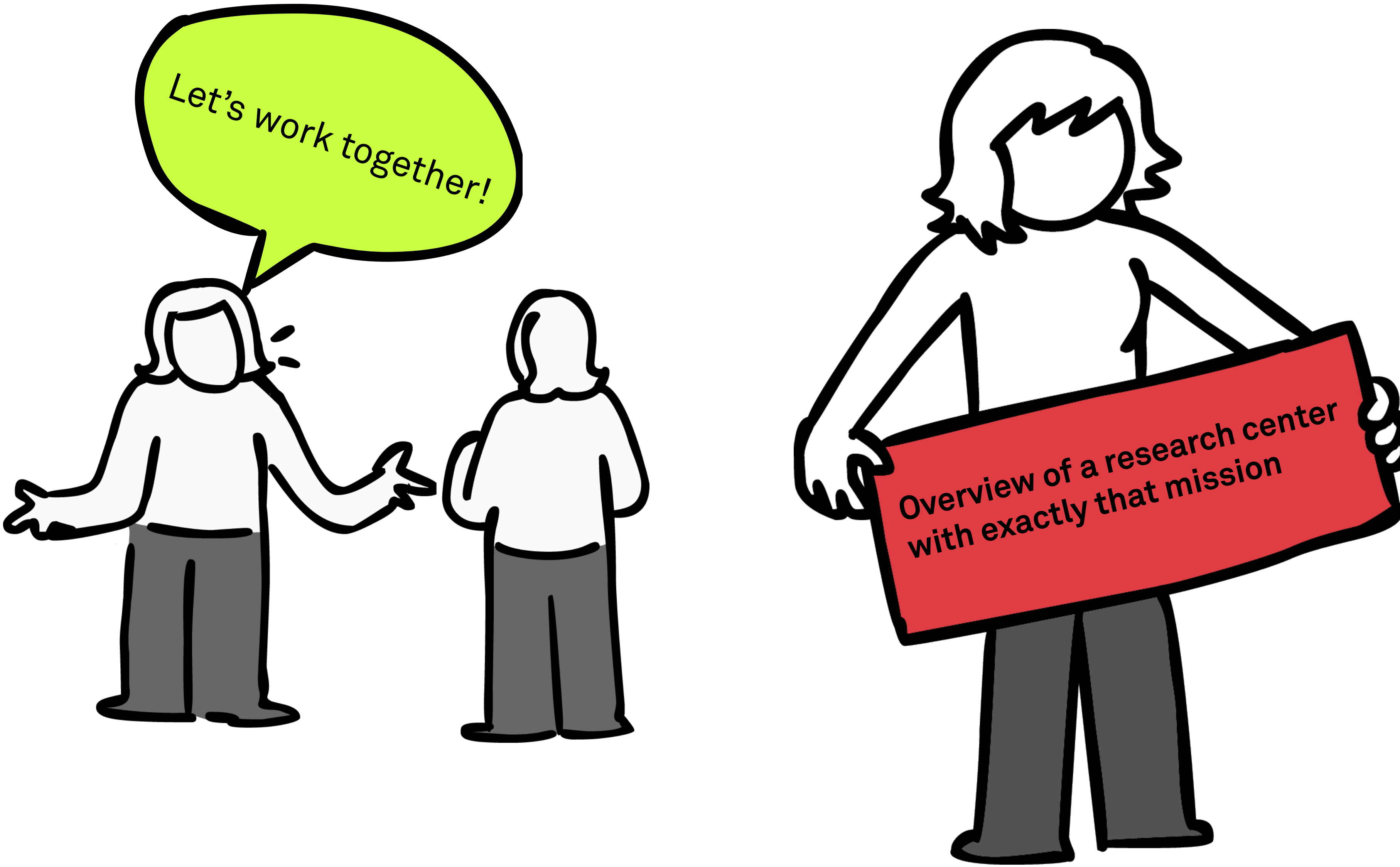
Texas Instruments MSP 430FR5969

Machine Learning and hardware -- 3



- Cloud computing
  - Hadoop
  - BigTable (Google Chrome)
  - HBase (Apache Cassandra)
- Lambda/Kappa paradigm
  - Map reduce
  - Stream processing
- GPU
  - Parallel computing
  - Multiple instruction, multiple data

Machine learning and hardware



Collaborative Research Center 876: Providing Information by Resource-Constrained Data Analysis

13 projects
20 professors
50 Ph D students

Integrated graduate school

2011 - 2018
4 more years are possible

SFB 876: Resource constraints

Small devices

- Small memory
- Low power
- Restricted arithmetic
- Novel architectures

Small devices **collect data**

- Internet of Things
- Cyber-physical systems
- Sensor measurements

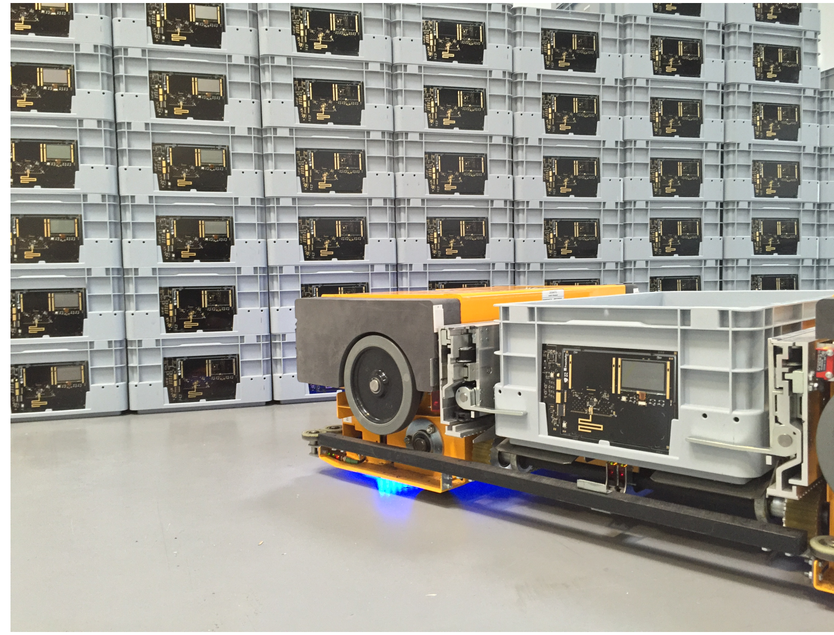
Small devices **apply models**

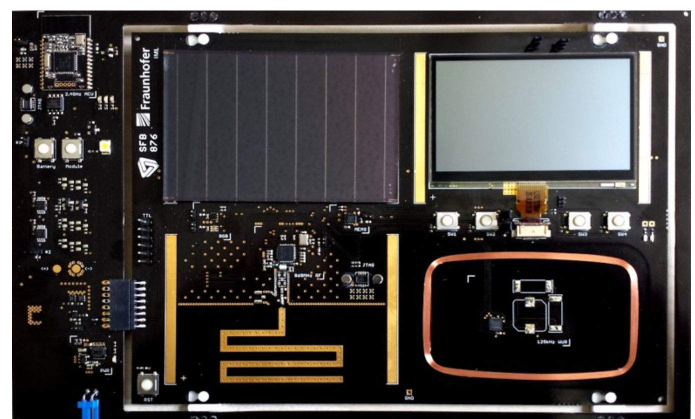
- Analysis and prediction available:
  - Anytime,
  - Anywhere!

Small devices

- Raspberry Pie
- FPGA
- Phy Node
- Logistics chip by SFB 876
  - with antenna,
  - photovoltaic for energy harvesting
  - smart way-bill for better routing

  Michael ten Hompel et al.
  project A4

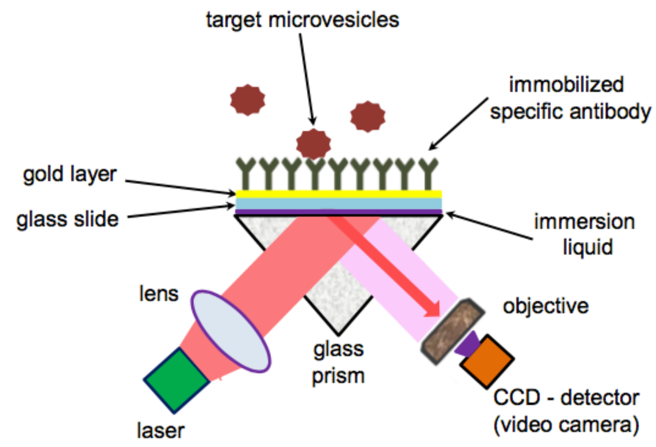


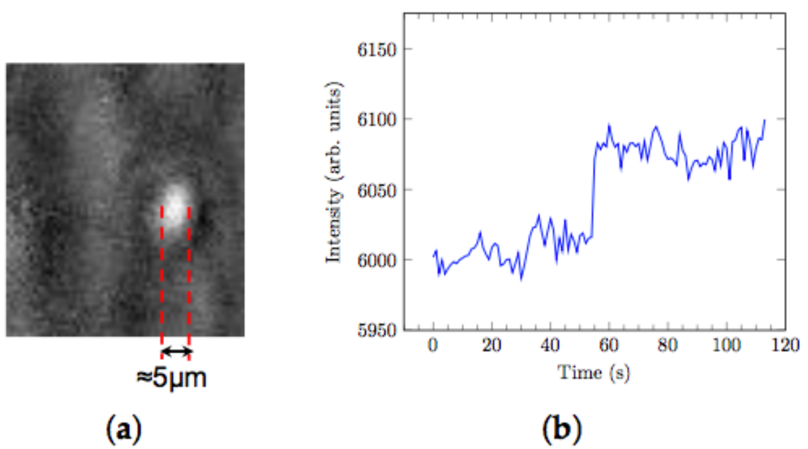


PAMONO

- Microscopy of nano-objects
  - immediate virus detection
  - DNA-DNA interactions
  - Intercellular communication through cell-derived vesicles
- Local changes of reflectivity image the binding events of nano-particles.
- Group of bright pixels indicates a binding event.
- Change of light intensity shows moment of binding and then stabilizing.

Victoria Shpacovitch, Heinrich Müller et al. project B2





SFB 876: Resource constraints

Big data

- Large volume, velocity, variety data
- High dimensions
- Complex models

Applications

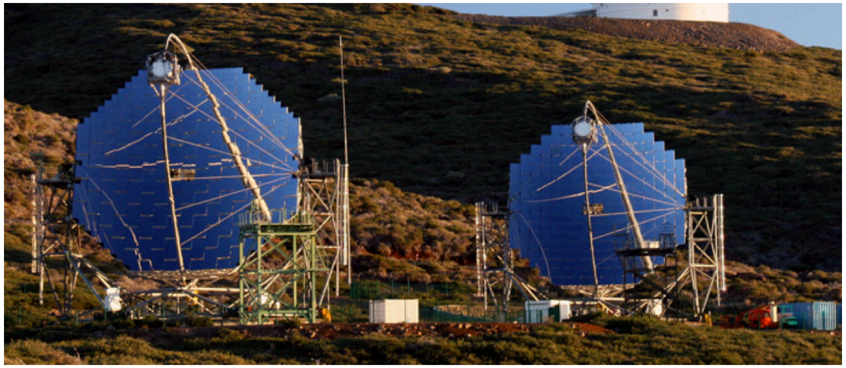
- Data-driven science
  - astro- and particle physics,
  - biomedicine and genetics

➢ **Goals**:
  - ➢ Scalable algorithms
  - ➢ Real-time inference
  - ➢ Compressed models

A terabyte a day

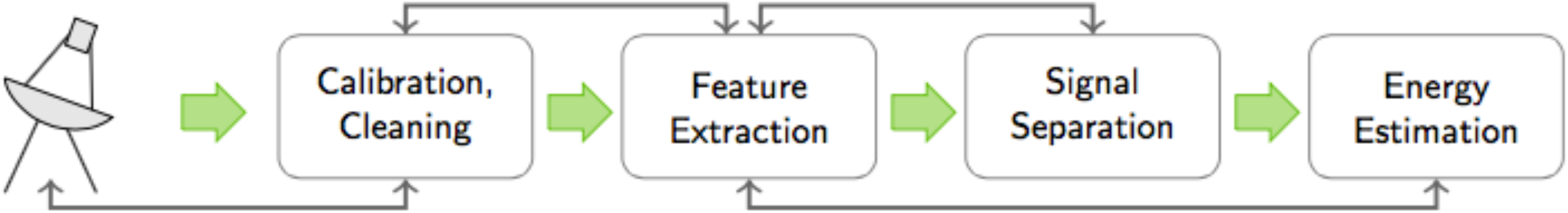
- Calibration, cleaning
- Feature extraction
- Signal separation
- Energy estimation
- A simulator provides labeled observations.
- Gamma rays of high energy are **rare events** as opposed to hadrons, ratio 1 to 1000

Project C3 in SFB 876 with Wolfgang Rhode, Tim Ruhe



MAGIC I, MAGIC II, FACT
La Palma, Roque de los Muchachos

C. Bockermann, K. Brügge, J.Buss, A.Egorov, K.Morik, W.Rhode, T.Ruhe
“Online Analysis of High-Volume Data Streams in Astroparticle Physics”
Best Paper Award ECML PKDD 2015



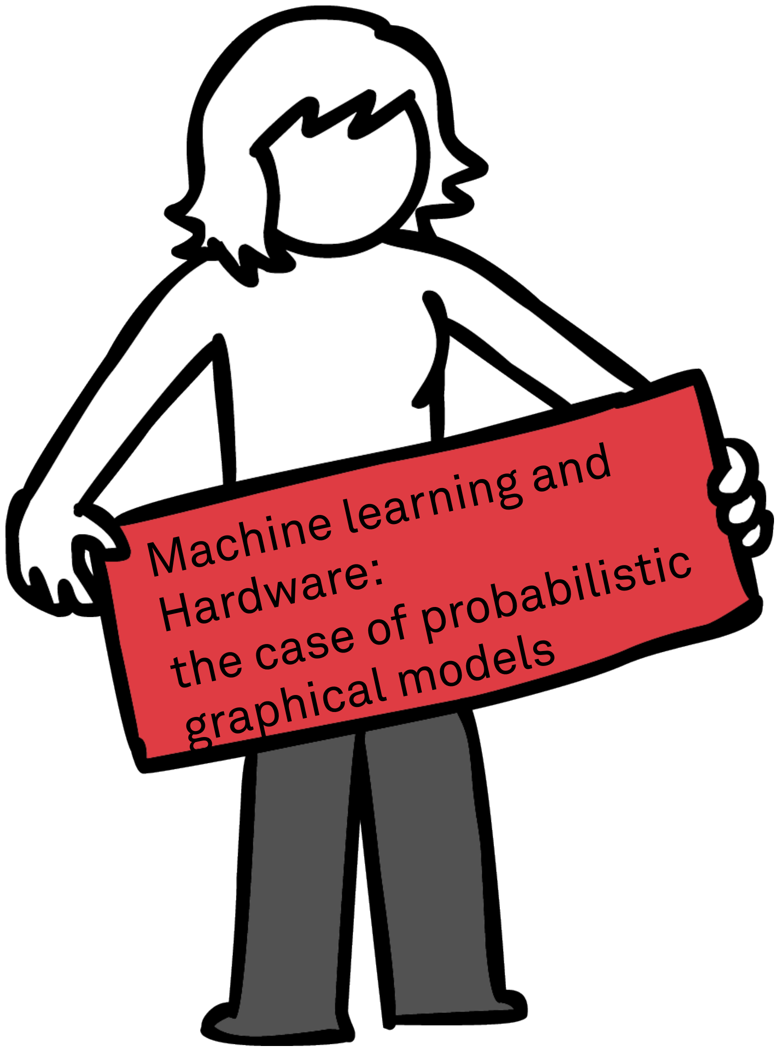
SFB 876: Resource-aware machine learning

- Cyberphysical systems
  - produce big data.
- Big data analytics
  - delivers data summaries, models for prediction.
- Push some analytics to CPS!
  - Less communication, energy
- Foundations
  - ➢ Beyond runtime and sample complexity!
  - ➢ Memory- and energy-efficient analytics!
  - ➢ Models that take resources into account!

➢ **Machine learning and computing machinery – a new challenge!**

Overview

- Machine learning and hardware
- Probabilistic graphical models
  - Spatio-temporal random fields
  - Integer Markov random fields
  - Stochastic Discrete Clenshaw Curtis Quadrature



Probabilistic models

- Data $D = \left\{ \vec{x}^{1}, \vec{x}^{2}, \ldots, \vec{x}^{N} \right\}$
- Observation x is realization of random variable X
  with state space $\mathcal{X}$
- State space $\mathcal{X} = \mathcal{X}_1$ x $\mathcal{X}_2$ x … x $\mathcal{X}_N$
- Probability P(x) of an event X=x

- Predict probability from data
- Estimate probability density
  - Topic models,
  - embeddings,
  - …
- Supervised: predict state with maximum likelihood given observations
  - Regression,
  - Naive Bayes,
  - Conditional Random Fields

Why exponential families?

- Sufficient statistic aggregates data:

$$\phi(D) = \frac{1}{D} \sum_{\vec{x} \in D} \phi(\vec{x})$$

- The dimension of is finite and independent of |D|
iff P(x) is in an exponential family. (Pitman 1936)
- ... and exp(.) > 0

- Markov Random Field (MRF)
= probability distribution that can be factorized into positive functions defined on cliques that cover all the nodes and edges of G.
(Hemmersley Clifford 1990)

Graphical Models

- Graph G=(V,E)
- Sufficient statistic: implicitly mapping joint vertex assignment into vector space $\phi(\vec{x})$: $\mathcal{X}$ --> $\mathcal{R}^d$
- Parameter vector to be learned: θ in $\mathcal{R}^d$
- Log partition function: $$A(\vec{\theta}) = \ln \sum_i \exp\left(\left\langle \vec{\theta}, \phi(\vec{x}_i) \right\rangle\right)$$

$CRF:$

$$p(\vec{y}|\vec{x}) = \frac{1}{Z(\vec{\theta},\vec{x})} \exp\left(\sum_i \theta_i \phi_i(\vec{y},\vec{x})\right) \quad \left| \frac{1}{a} = \exp(-\ln a) \right.$$

$$= \exp\left[\left(\sum_i \theta_i \phi_i(\vec{y},\vec{x})\right) - \ln Z(\vec{\theta},\vec{x})\right]$$

$$= \exp\left[\left\langle \vec{\theta}, \phi(\vec{y},\vec{x}) \right\rangle - A(\vec{\theta})\right]$$

$MRF:$

$$p(\vec{x}) = \frac{1}{Z(\vec{\theta})} \exp\left(\sum_i \theta_i \phi_i(\vec{x})\right)$$

$$= \exp\left[\left\langle \vec{\theta}, \phi(\vec{x}) \right\rangle - A(\vec{\theta})\right]$$

Discrete Random Field

$$\phi(\vec{x}) \in \{0,1\}^d$$

$$\phi_V(\vec{x}) = \begin{pmatrix} 1_{\{x_i=1\}} \\ 1_{\{x_i=2\}} \\ \ldots \\ 1_{\{x_i=k_i\}} \end{pmatrix} \qquad \phi_E(\vec{x}) = \begin{pmatrix} 1_{\{x_i=1\}}1_{\{x_j=1\}} \\ 1_{\{x_i=1\}}1_{\{x_j=2\}} \\ \ldots \\ 1_{\{x_i=k_i\}}1_{\{x_j=k_j\}} \end{pmatrix}$$

$$i \in V \qquad (i,j) \in E$$

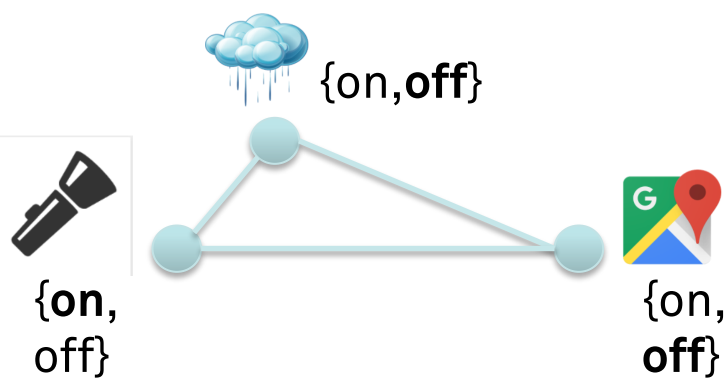
$$\phi(\vec{x}) = \left(\phi_V(\vec{x})^T, \phi_E(\vec{x})^T\right)^T$$

$$P(\vec{x}) = \exp\left(\left\langle \theta, \phi(\vec{x}) \right\rangle\right) - A(\theta)$$

Discrete Random Fields – Example: app usage

Observation

- $\mathbf{x}_1$:(on,off,off)
- $\phi(\mathbf{x}_1) = (1,0,0,1,0,1,1,0,0,1,0,0,0,0,1)$
- d = 15



Graph G=(V,E)

$\phi(\mathbf{x})$: (/* Verticeses*/

1. **on**, /*dom(torch)*/
2. off,
3. on, /*dom(rain)*/
4. **off**,
5. map, /*dom(map)*/
6. **off**,

/*Edges*/

1. **on, off**, /* edge torch-rain*/
2. on,on,
3. off,off,
4. **on,off**, /*edge torch-map*/
5. on, on,
6. off, off,
7. on, off, /*edge rain-map*/
8. on, on,
9. **off, off**)

)

Machine learning and hardware -- 4

- Ultra-low power devices offer resources.
- It is the (to be) learned model which demands resources.
  - Redundancies
  - Real values
  - Exponential complexity
    $P_E(\vec{x}), A(\theta)$
  - Parameter storing, sufficient statistics (graph)

- Investigate model demands:
  - Application independent
  - Dependency preserving
  - Theoretically well-based not heuristic
  - Derived from first principles
  - Implemented.

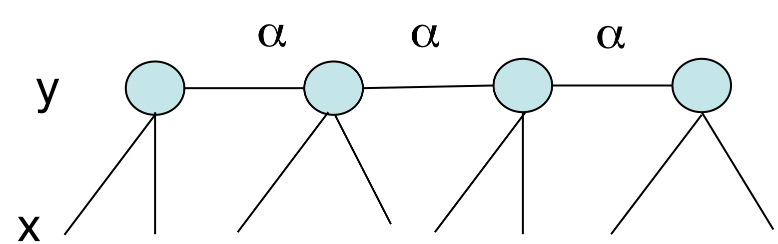
Goal 1: Application independence

- Application dependent
  - Physics: Ising
    graph of adjacent atomic spins
    with states {+1, -1}

    $$P_\beta(\vec{x}) = \frac{1}{Z_\beta} \exp\left(-\beta H(\vec{x})\right)$$

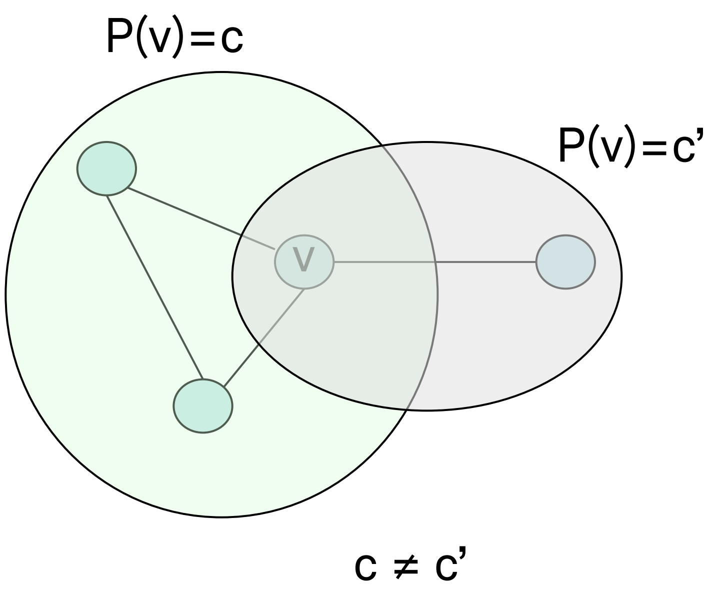
    Exploit structure given by the
    application! Ferromagnetic,
    later hierarchical classification.
  - Linguistics: CRF
    The transition is always the
    same, x, y are distinct.
- Application independent:
  - Restrict the resource demands
    of the model.

β	0
0	β



Goal 2: Dependency preserving

- Variational inference destroys some dependencies, because not all cliques are considered.
- Inconsistencies possible.



Models and hardware demands

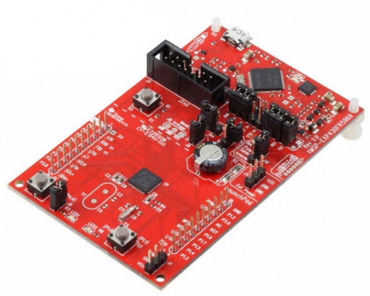
- Resource demands of models
- Where can we save resources?

$$P(\vec{x}) = \exp\left(\left\langle \vec{\theta}, \phi(\vec{x}) \right\rangle - A\left(\vec{\theta}\right)\right)$$

- Parameters and redundancies

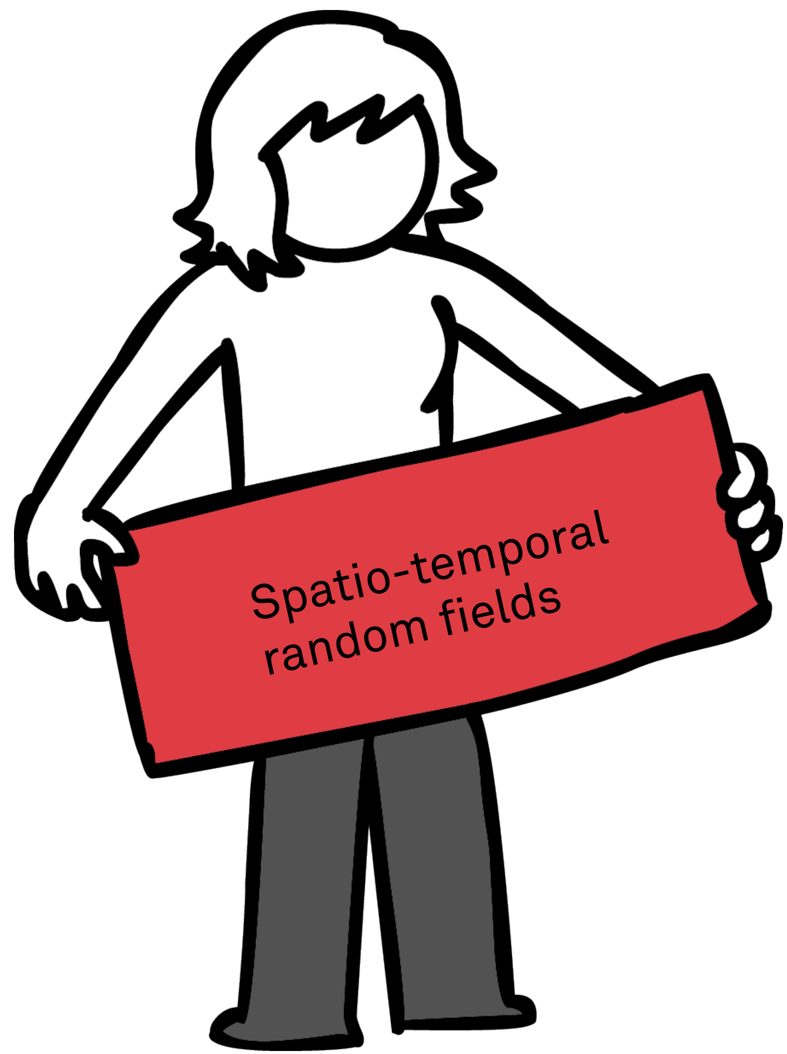
Ultra-low power device

- Slow 16 MHz
- Small wordsize 16 Bit
- **Small memory 64 Kb**
- **Restricted capabilities, no floating point unit**
- Connectable to multiple sensors
- Energy around 0,0048 W



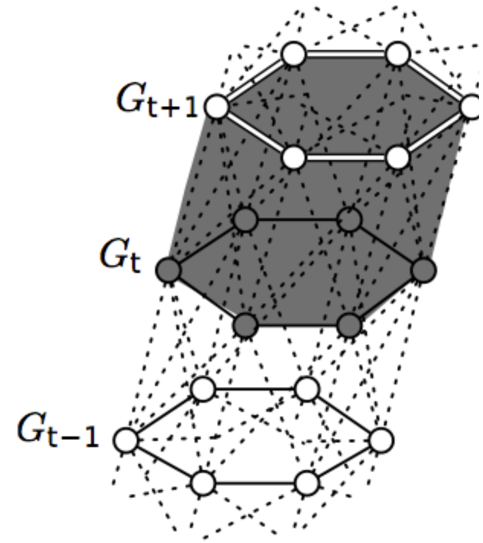
Overview

- Machine learning and hardware
- Probabilistic graphical models
  - Spatio-temporal random fields
  - Integer Markov random fields
  - Stochastic Discrete Clenshaw Curtis Quadrature



Spatio-temporal random fields

- The spatio-temporal graph is trained to predict each node's maximum a posteriori probability with the marginal probabilities.
  - Generative model predicting all nodes.
- Dimension
  $T \times |V_0| \times |\mathcal{X}| +$
  $[(T-1)(|V_0|+3|E_0|)+ |E_0|] \times |\mathcal{X}|^2$
- Remember: vectors are sparse we have to exploit that!



User queries:
Given traffic densities at all nodes at t_1, t_2, t_3, what is the probability of traffic density at node A at time t_5?
Given state "jam" at place A t_s, which other places have a higher probability for "jam" in $t_s < t < t_e$?

Spatio-temporal random fields

- Parameter sharing
- Reparametrization
- Regularization
- Distributed optimization

If edges in some subset represent similar relations and have a common state space, then instead of

	c_1	c_2	c_3
d_1	$\theta_{cd=c_1 d_1}$	$\theta_{cd=c_2 d_1}$	$\theta_{cd=c_3 d_1}$
d_2	$\theta_{cd=c_1 d_2}$	$\theta_{cd=c_2 d_2}$	$\theta_{cd=c_3 d_2}$
d_3	$\theta_{cd=c_1 d_3}$	$\theta_{cd=c_2 d_3}$	$\theta_{cd=c_3 d_3}$

we may share parameters

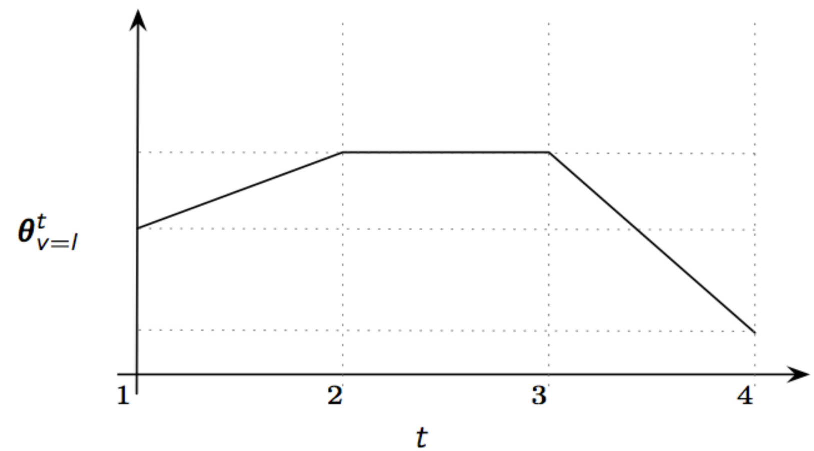
	z_1	z_2	z_3
z_1	$\theta_{vw=z_1 z_1}$	$\theta_{vw=z_2 z_1}$	$\theta_{vw=z_3 z_1}$
z_2	$\theta_{vw=z_1 z_2}$	$\theta_{vw=z_2 z_2}$	$\theta_{vw=z_3 z_2}$
z_3	$\theta_{vw=z_1 z_3}$	$\theta_{vw=z_2 z_3}$	$\theta_{vw=z_3 z_3}$

Reparametrization compresses the model

- Reparametrize model
  $\Delta_t \approx \theta_{t+1} - \theta_t$
  Δ regularized by L1, L2 norm
- There are not many changes over time. Model is highly compressed.
- Bound on distance between true θ and $\nu(\Delta)$;
  Sparsity in estimate implies redundancy in the true parameter. Proof Piatkowski
- Learning is faster.
- Quality is not at all less than MRF, 4NN.

Universal reparametrization
Proof Piatkowski (forthcoming)

Assumption: Smoothness over time



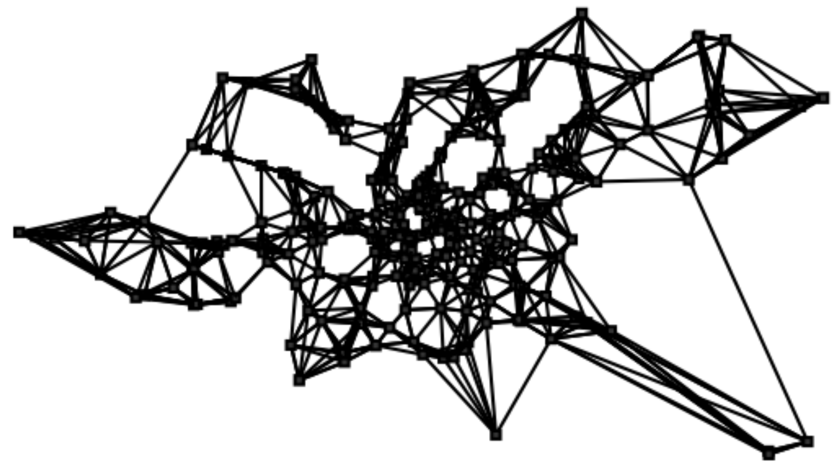
Idea:

- Remove the near zero slopes, while retaining the performance

Smart trip modeling for Dublin

- Open Street Map → graph topology
- Open Trip Planner: user query (v,w), route planning based on traffic costs.
- Traffic costs learned:
  - Spatio-temporal random field based on sensor data stream;
  - Gaussian process estimates values for non-sensor locations.
- Framework for real-time processing of data streams, XML configuration of data flow, connecting data, traffic model and planner.

Constructing the spatio-temporal graph of Dublin



OpenStreetMap streets segmented according to junctions.
966 sensors transmit traffic flow every 6 minutes (www.dublinked.ie).
Aggregate sensor readings for 30 minutes, aggregate sensor nodes by 7NN.
Traffic flow discretized into 6 intervals of density.
48 time layers for each day (48* 30=1440 minutes make a day).
Training for every weekday.
Predicting density of each node and edge – interpreted as costs.

Using STRF for smart trip modeling -- Evaluation

- Confusion matrix of predicting the number of vehicles (6 intervals) for all sensors and all half hours following 1 pm on Fridays, tested on March 1.,8.,15.,22.,29.
- given the traffic at 1 p.m. (bold is true).

Predicted → / True ↓	**0**	**1-5**	**6-20**	**21-30**	**31-60**	**61-**	**Prec**
0	**840**	32	10	6	3	0	**0.943**
1-5	2	**632**	498	3	0	1	0.556
6-20	91	156	**12169**	2006	83	25	**0.838**
21-30	32	0	1223	**5637**	717	14	0.739
31-60	43	0	60	893	**1945**	29	0.655
61-	0	0	16	3	12	**35**	0.530
Recall	**0.833**	0.771	**0.871**	0.659	0.705	0.34	

Smart traffic for smart cities

- Several questions can be answered using the same learned model.
- The answers come along with their probabilities. This might be helpful for decision makers.
- Integration of Spatio-temporal random fields into the Open Trip Planner and Gaussian information completion resulted in an excellent navigation system.

EU project INSIGHT

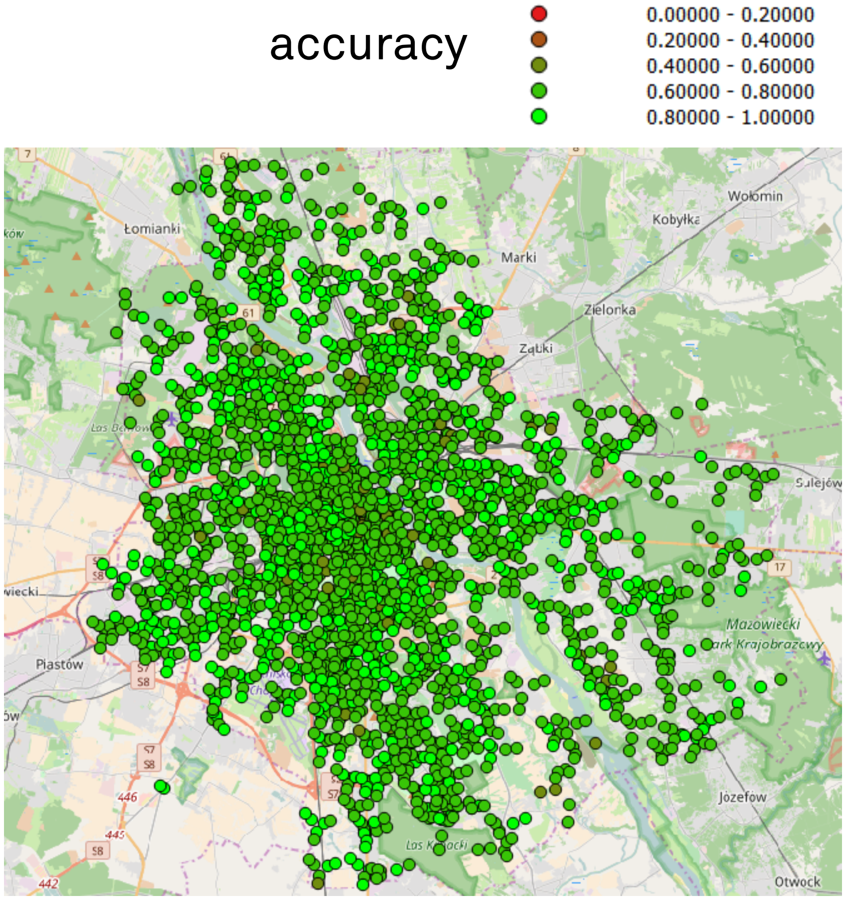
Liebig et al (2014)

Piatkowski, Lee, Morik (2013) Spatio-temporal random fields: compressible representation and distributed estimation, Machine Learning Journal 93:1, 115 – 140.
Liebig, Piatkowski, Bockermann, Morik (2014) Predictive Trip Planning – Smart Routing in Smart Cities, Mining Urban Data Workshop at 17th Intern. Conf. on Extending Database Technology.

Prediction of phone calls in cells in the next hour

- Data from Orange Warsaw
- 3923 cells with at least 4/5 data points registered
- 16.5.2016 – 26.6.2016
- 24 h * 3923 random variables
- 3 classes separated at 1/3, 2/3 quantile

	L	M	H	N
L	**489113**	49828	2640	26853
M	67597	**404590**	73971	7865
H	19731	181128	**463748**	1824
N	0	0	0	0



STRF

- STRF compress model to meet resource constraints of devices.
  - Small memory

- Investigate model demands:
  - ✓ Application independent
  - ✓ Dependency preserving
  - ✓ Theoretically well-based not heuristic
  - ✓ Derived from first principles
  - ✓ Implemented.

Overview

- Machine learning and hardware
- Probabilistic graphical models
  - Spatio-temporal random fields
  - Integer Markov random fields
  - Stochastic Discrete Clenshaw Curtis Quadrature

Graphical models on resource-restricted processors

- Floating point arithmetics (real) costs more clock cycles than integer arithmetics.
- "The most obvious technique to conserve power is to reduce the number of cycles it takes to complete a workload." (Intel 64, IA-32 architectures optimization reference manual, guidelines for extending battery life).
- Restrict the parameter space of MRF

$$\theta \in \{0,1,\ldots,K\} \subset \mathrm{N}$$

	Sandy Bridge		**ARM 11**	
	Real	Int	Real	Int
+	3	**1**	8	**1**
*	5	**3**	8	4-5
/	14	13-15	19	-
Bit shift	-	3	-	2

Clock cycles for arithmetics on different processors: Real vs. integer.

Parameter space transformation

- Graph model tree-structured
- Transform the parameter space:

$$\eta_i(\theta) = \theta_i \ln 2$$

$MRF:$

$$p(\vec{x}) = \frac{1}{Z(\theta)} \exp\left(\sum_i \theta_i \phi_i(\vec{x})\right)$$

$$= \exp\left[\langle \theta, \phi(\vec{x}) \rangle - A(\theta)\right]$$

$IntegerMRF:$

$$p(\vec{x}) = \exp\left[\langle \eta(\theta), \phi(\vec{x}) \rangle\right]$$

$$= 2^{\left[\langle \theta, \phi(x) \rangle - A(\eta(\theta))\right]}$$

$$= \frac{2^{\langle \theta, \phi(x) \rangle}}{\sum_{y \in \aleph} 2^{\langle \theta, \phi(y) \rangle}}$$

Integer belief propagation

- Simply replacing the exp(.) by $2^{(.)}$ is not sufficient
  - Overflows are normally avoided by normalization.
  - Normalization is impossible in integer division.
- Magnitude of messages corresponds to probability
  - Use the length of each message
  - Bit-length is similar to log

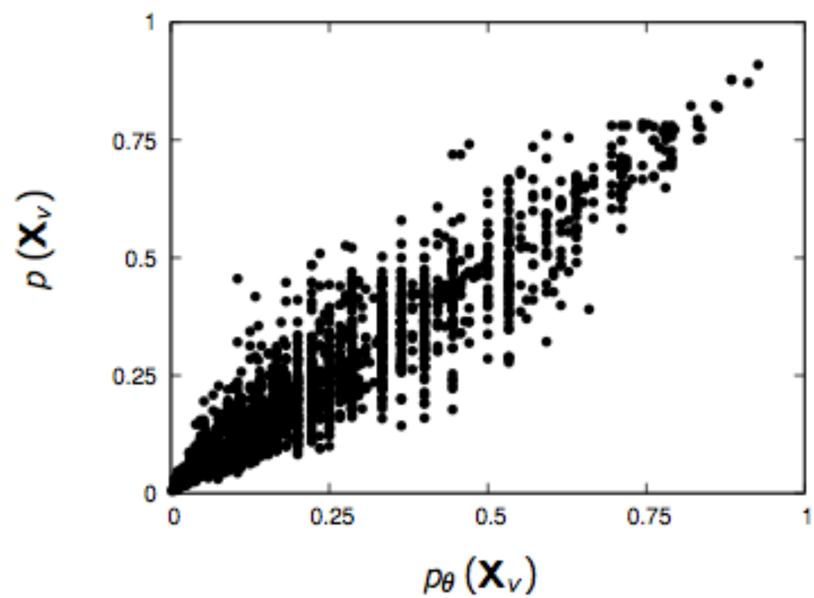
$$m_{v\to u}(y) = \sum_{x\in\aleph_v} \exp\left(\theta_{vu=xy} + \theta_{v=x}\right) \prod_{w\in N_v-\{u\}} m_{wu}(x)$$

$$\tilde{m}_{v\to u}(y) = \sum_{x\in\aleph_v} 2^{\left(\theta_{vu=xy} + \theta_{v=x}\right)} \prod_{w\in N_v-\{u\}} \tilde{m}_{wu}(x)$$

$$\beta_{v\to u}(y) = \max_{x\in\aleph_v} \theta_{vu=xy} + \theta_{v=x} + \sum_{w\in N_v-\{u\}} \beta_{wu}(x)$$

Discretized probability space

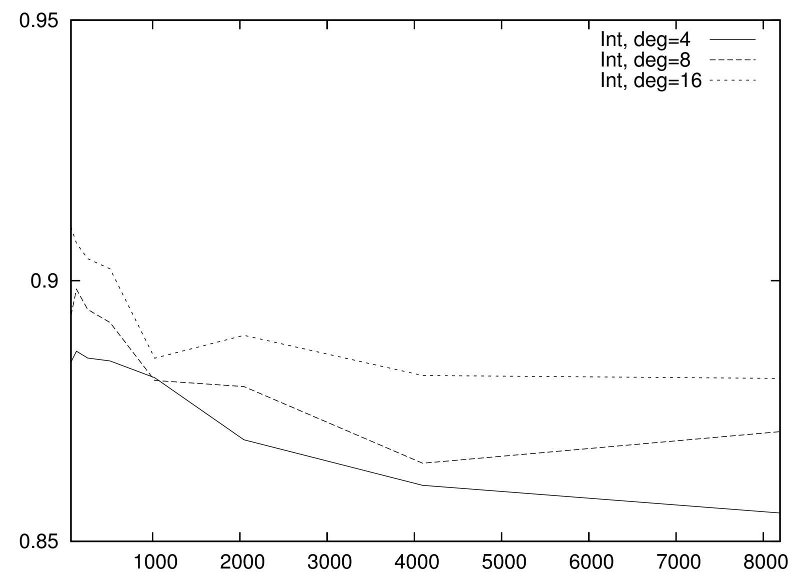
- Belief propagation is now bit-length propagation, i.e. the MAP and marginals are computed using the bit-length.
- The approximation error depends on the number of neighboring nodes and the space of states.
- Some true probabilities (y axis) cannot be expressed by the integer approximation (x axis).



Evaluation: accuracy

- Different vertex degrees
  - 8000 nodes in the graph
  - 100 runs of IntMRF

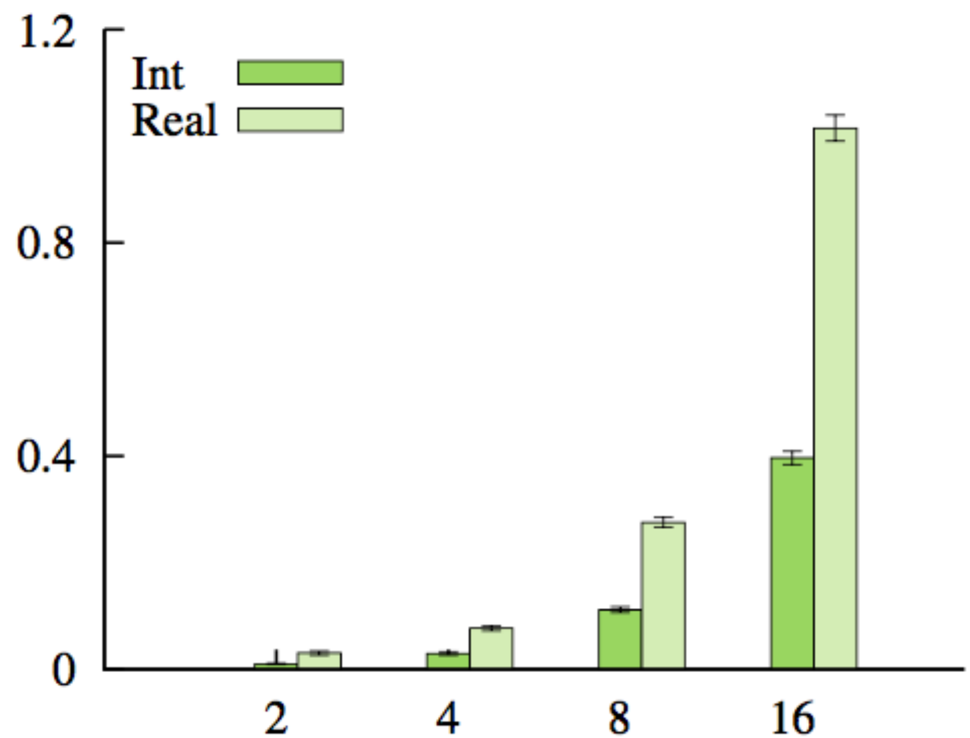
- Real MRF is 100% accuracy.



Integer model on ARM

- >> 10 times faster on ultra-low devices
- State of the art performance in NLP and other real-world tasks.

seconds



Vertex state space $|\mathcal{X}|$

Integer MRF

- IntMRF can be executed on devices even those without floating point unit.
-
- Investigate model demands:
  - Application independent
  - Dependency preserving
  - Theoretically well-based not heuristic
  - Derived from first principles
  - Implemented.

Overview

- Machine learning and hardware
- Probabilistic graphical models
  - Spatio-temporal random fields
  - Integer Markov random fields
  - Stochastic Discrete Clenshaw Curtis Quadrature

Models and hardware demands

- Resource demands of models
  - ✓ Parameters and redundancies

- **Still exponential complexity!**

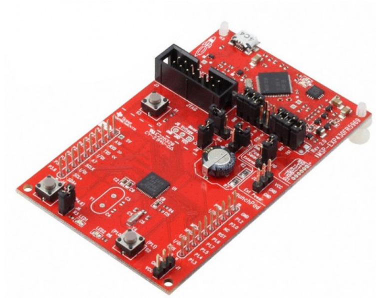
$$P(\vec{x}) = \exp\left(\left\langle \vec{\theta}, \phi(\vec{x}) \right\rangle - A\left(\vec{\theta}\right)\right)$$

potential

partition

Ultra-low power device

- Slow 16 MHz
- Small wordsize 16 Bit
- **Small memory 64 Kb**
- **Restricted capabilities, no floating point unit**
- Connectable to multiple sensors
- Energy around 0,0048 W



Calculation of the partition function

$$\exp\left(A(\theta)\right) = Z(\theta)$$

$$Z(\theta) = \int_{\aleph} \psi(\vec{x})\, d\nu(\vec{x})$$

$$\int_{l}^{u} h_k(\vec{x})\, d\vec{x} = \sum_{i=1}^{k} w_i f(x_i)$$

$$h_k(\vec{x}) = \sum_{i=1}^{k} c_i x^i$$

$$\tilde{A}_k(\theta) = \log \sum_{i=1}^{k} w_i\, E\left[\prod_{l=1}^{i} \theta_{Jl} \,\middle|\, i\right]$$

- In general, evaluating $Z(\theta)$ is #P-complete.
- Numerical approximate integration based on general quadrature:
  - replace f by h
  - x, w, x_i need to be determined
- Chebyshev polynomials as h
  $T_k(x) = 2x\, T_{k-1}(x) - T_{k-2}(x)$
- Chebyshev interpolation
- Expensive part w_i depends on G, $\mathcal{X}$, $||\theta||$
  can be pre-computed!

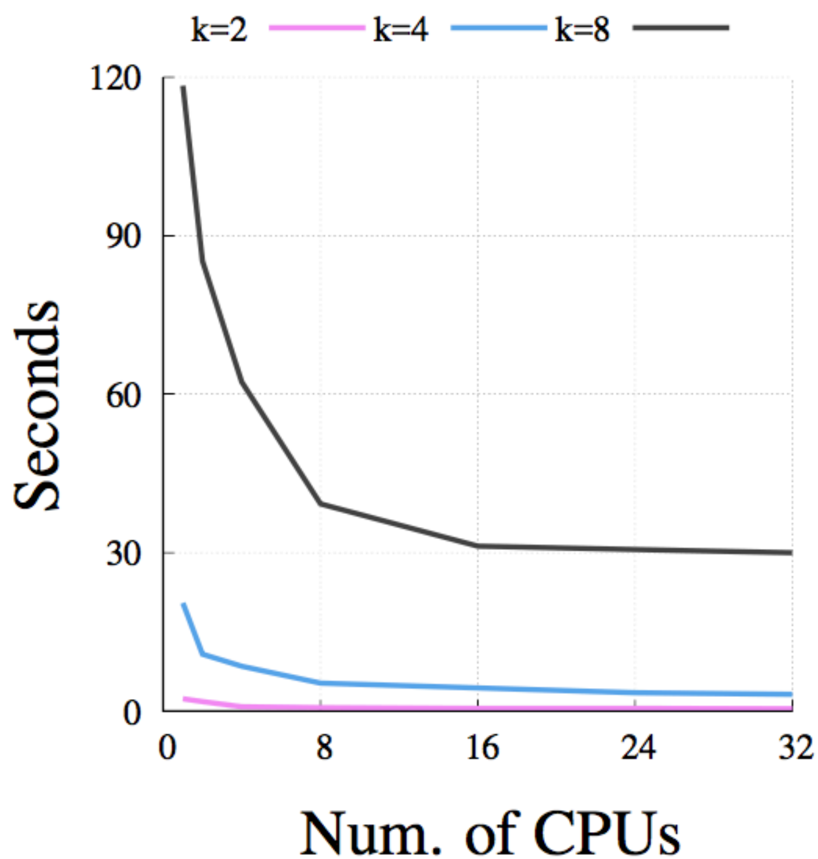
Quadrature-based inference

- Numerical approximation technique with bounded error independent of the graph structure.
- Discrete Clenshaw-Curtis Quadrature:
  - In: G, θ in $\mathcal{R}^d$, degree k
  - Out: $|Z(\theta) - Z_k(\theta)| \leq \varepsilon/2\, Z(\theta)$
- Randomized algorithm SDCCQ based on Chebyshev polynomials
  - Pre-compute w_i on server
  - Send to device and perform there inference with quality guarantees.

Algorithm	Complexity	Quality
JT (Lauritzen & Spiegelhalter, 1988)	$\mathcal{O}(L^w)$	Exact
MF (Weiss, 2001)	$\mathcal{O}(InL\Delta)$	Lower bound
LBP (Heskes, 2002; Yedidia et al., 2003)	$\mathcal{O}(ImL^2\Delta)$	Local minimum of Bethe free energy
TRW (Wainwright et al., 2005)	$\mathcal{O}(ImL^2\Delta + m\log n)$	Upper bound
WISH (Ermon et al., 2013)	$\mathcal{O}(n\ln(n/\zeta)) \times \text{Time(MAP)}$	$(16,\zeta)$-approx
DCCQ (Alg. 1)	$\mathcal{O}(k_\varepsilon^2 d^{2k_\varepsilon})$	ε-approx (Theorem 5)
SDCCQ (Alg. 2)	$\mathcal{O}(k_\varepsilon^2 d^{2k_\varepsilon}) + \mathcal{O}(k_\varepsilon^2 m_\zeta)$	(ε,ζ)-approx (Theorem 7)

Scalability

- Runtime in seconds as a function of the number of CPU cores for different polynomial degrees.
- 40 E5 2697 Xeon CPU cores.
- Algorithm is easily made parallel.

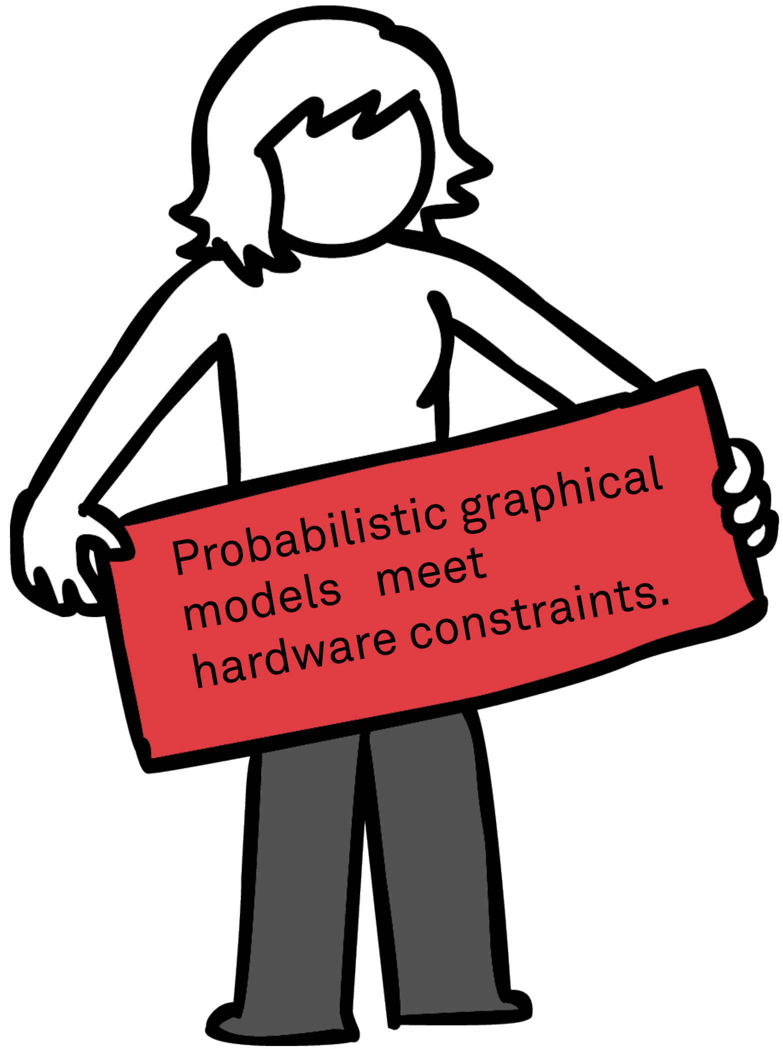


Discrete Clenshaw-Curtis Quadrature

- Decoupling most costly computation from the rest and pre-compute it.
- Use quadrature for partition function.
- Investigate model demands:
  - Application independent
  - Dependency preserving
  - Theoretically well-based not heuristic
  - Derived from first principles
  - Implemented.

Overview

- Machine learning and hardware
- Probabilistic graphical models
  - Spatio-temporal random fields
  - Integer Markov random fields
  - Stochastic Discrete Clenshaw Curtis Quadrature



Contributors

Nico Piatkowski
- STRF
- Integer MRF
- Stochastic Quadrature

Sangkyun Lee
- Optimization
- Regularization

Christian Bockermann
- Streams framework
- FACT Tools

Thomas Liebig
- Traffic prognosis
- Routing

Resource-aware Machine Learning - 4th International Summer School 2017

TU Dortmund, Germany, 25.09. - 28.09.2017

HOME | REGISTRATION | DATES | PROGRAM | LOCATION | LECTURES | GRANTS | CONTACT

Important Dates Summer School 2017

Registration Opens	1st of March 2017
Early Registration Deadline	30th of June 2017 Early registration fee is 350,-€.
Late Registration Deadline	31st of August 2017 Late registration fee is 400,-€. No on-site registration or payment is possible.
Application Deadline for Student Grants	15th of July 2017
Student Grant Decision	25th of July 2017
Summer School	25th - 28th of September 2017

We are hiring

Thank you for your attention

References

- C. Bockermann et al. (2015) “Online Analysis of High-Volume Data Streams in Astroparticle Physics” Best Industrial Paper ECML PKDD
- Christian Bockermann “Mining Big Data Streams for Multiple Concepts” 2015, Ph D thesis, TU Dortmund University, https://eldorado.tu-dortmund.de/handle/2003/34363
- N. Ding, J. Ding, K. Murphy, H. Neven (2015) “Probabilistic label proportion graphs with Ising models” ICCV
- P. Dubey (2017)“The quest for the ultimate learning machine” ACM Int. Symposium on Physical Design
- Liebig, Piatkowski, Bockermann, Morik (2014) „Predictive Trip Planning – Smart Routing in Smart Cities“ Mining Urban Data Workshop at 17th Intern. Conf. on Extending Databases
- S. Mittal, J.S. Vetter (2014) “A Survey of methods for analyzing and improving GPU energy efficiency”, *ACM computing surveys,* 47:2
- Nico Piatkowski, Sangkyun Lee, Katharina Morik (2013) “Spatio-temporal Random Fields: Compressible Representation and Distributed Estimation” *Machine Learning Journal*, 93:1, 115 – 140.

References

- Nico Piatkowski, Sangkyun Lee, Katharina Morik (2016) "Integer Undirected Graphical Models for Resource-Constrained Systems" *Neurocomputing*, 173:1, 9 – 23
- Nico Piatkowski, Katharina Morik (2016) "Stochastic Discrete Clenshaw-Curtis Quadrature" ICML, Proceedings JMLR
- Nico Piatkowski "Exponential Families and Resource Constrained Systems" (forthcoming) Ph D thesis
- James G. Pitman (1936) "Sufficient statistics and intrinsic accuracy" Math. Procs. Cambridge Philosophical Society, 32:4, 567 - 579
- Shpacovitch et al. (2017) "Application of the PAMONO Sensor for quantification of microvesicles and determination of nano-particle size distribution" *Sensors*, Vol. 17, 244
- C.Timm, A Gelenberg, F. Weichert, P. Marwedel (2010)"Reducing the energy consumption of embedded systems by integating GPUs" Tech. Report in Computer Science, TU Dortmund,
- R. Venkatapathy et al. (2015) "PhyNode: An intelligent, cyber-physical system with energy neutral operation for PhyNetLab" *Smart Sys Tech*