

Riemann-Theta Boltzmann machine

based on arXiv:1712.07581 and arXiv:1804.07768

Stefano Carrazza

ACAT 2019, 11-15 March 2019, Saas Fee

Universit degli Studi di Milano (UNIMI and INFN Milan)

Acknowledgement: This project has received funding from HICCUP ERC Consolidator grant (614577) and by the European Unions Horizon 2020 research and innovation programme under grant agreement no. 740006.



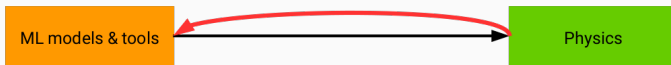
Introduction

Introduction

We started this project aiming to build a model with:

- well suited for pdf estimation and pdf sampling
- built-in pdf normalization (close form expression)
- very flexible with a small number of parameters

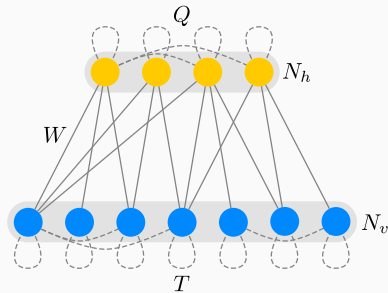
We decided to look at energy models, specifically Boltzmann Machines.



Theory

Boltzmann machine

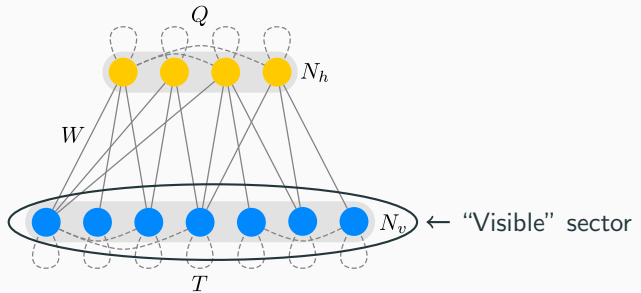
Graphical representation:



Boltzmann machine

Graphical representation:

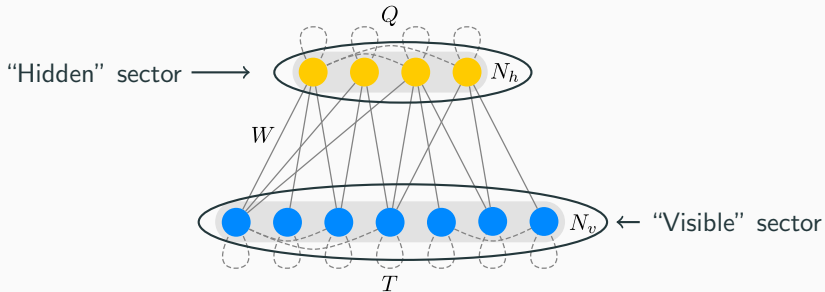
[Hinton, Sejnowski '86]



Boltzmann machine

Graphical representation:

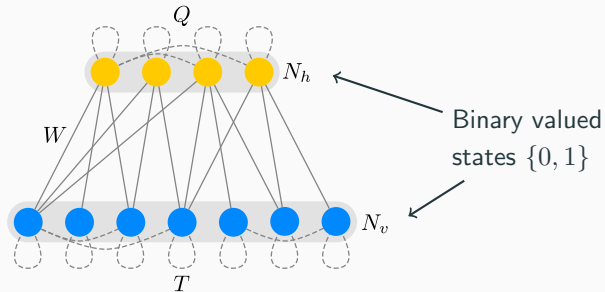
[Hinton, Sejnowski '86]



Boltzmann machine

Graphical representation:

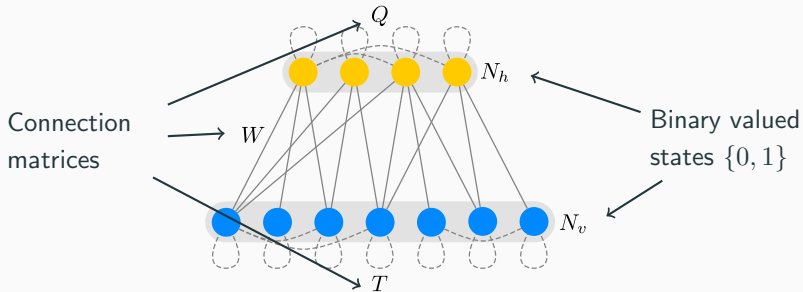
[Hinton, Sejnowski '86]



Boltzmann machine

Graphical representation:

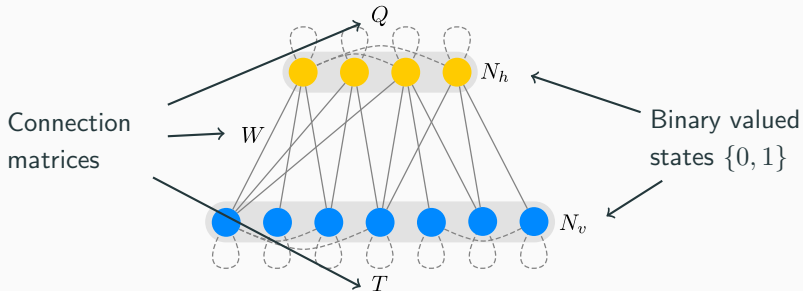
[Hinton, Sejnowski '86]



Boltzmann machine

Graphical representation:

[Hinton, Sejnowski '86]

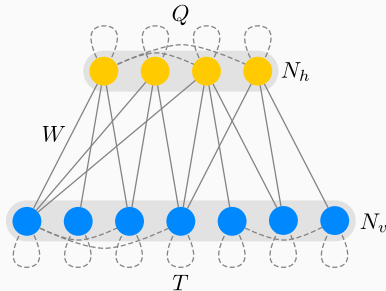


- Boltzmann machine (BM): T and $Q \neq 0$.
- Restricted Boltzmann machine (RBM): $T = Q = 0$.

Boltzmann machine

Energy based model:

[Hinton, Sejnowski '86]

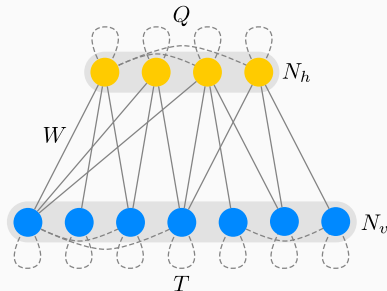


View as statistical mechanical system.

Boltzmann machine

Energy based model:

[Hinton, Sejnowski '86]



View as statistical mechanical system.

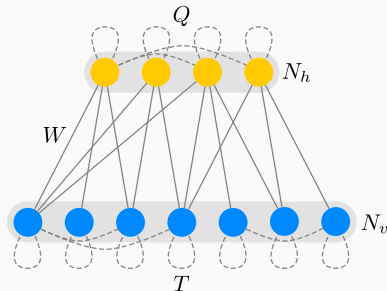
The system energy for given state vectors (v, h) :

$$E(v, h) = \frac{1}{2}v^t T v + \frac{1}{2}h^t Q h + v^t W h + B_h h + B_v v$$

Boltzmann machine

Energy based model:

[Hinton, Sejnowski '86]



View as statistical mechanical system.

The system energy for given state vectors (v, h) :

$$E(v, h) = \frac{1}{2} v^t T v + \frac{1}{2} h^t Q h + v^t W h + B_h h + B_v v$$

Diagram illustrating the components of the energy function $E(v, h)$:

- $\frac{1}{2} v^t T v$: State vectors (pointing to v)
- $\frac{1}{2} h^t Q h$: Connection matrices (pointing to Q)
- $v^t W h$: Connection matrices (pointing to W)
- $B_h h$: Biases (pointing to B_h)
- $B_v v$: Biases (pointing to B_v)

Boltzmann machine

Energy based model:

[Hinton, Sejnowski '86]

Starting from the system energy for given state vectors (v, h) :

$$E(v, h) = \frac{1}{2}v^tTv + \frac{1}{2}h^tQh + v^tWh + B_hh + B_vv$$

The canonical partition function is defined as:

$$Z = \sum_{h,v} e^{-E(v,h)}$$

Probability the system is in specific state given by Boltzmann distribution:

$$P(v, h) = \frac{e^{-E(v,h)}}{Z}$$

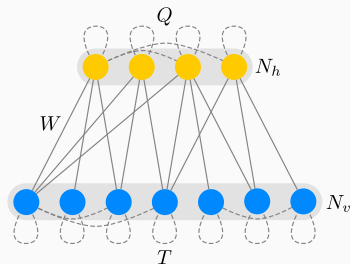
with marginalization:

$$P(v) = \frac{e^{-F(v)}}{Z} \quad \leftarrow \text{Free energy}$$

Boltzmann machine

Learning:

[Hinton, Sejnowski '86]



Theoretically, general compute medium.

Via adjusting W, T, Q, B_h, B_v able to learn the underlying probability distribution of a given dataset.

However: practically not feasible

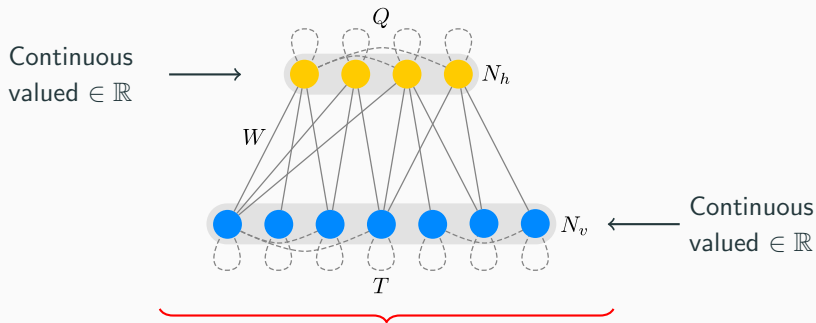
For applications only RBMs have been considered.

Riemann-Theta Boltzmann machine

How to change the status quo?

[Kreft, S.C., Haghighat, Kahlen '17]

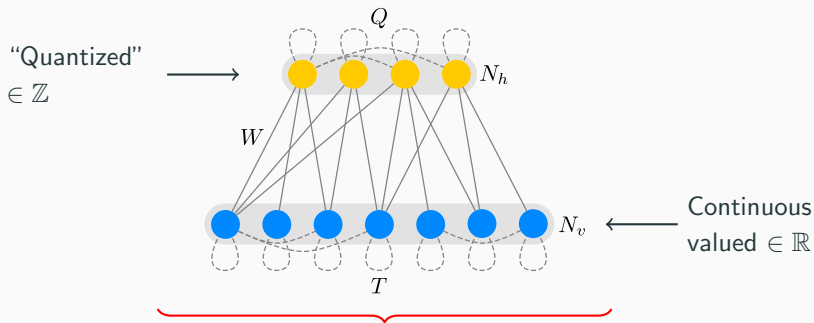
Keep the inner sector couplings non-trivial, but the machine solvable?



$P(v) \equiv$ multi-variate gaussian (*too trivial*)

Riemann-Theta Boltzmann machine

How to change the status quo? [Krefl, S.C., Haghighat, Kahlen '17]
Keep the inner sector couplings non-trivial, but the machine solvable?



Something interesting happens

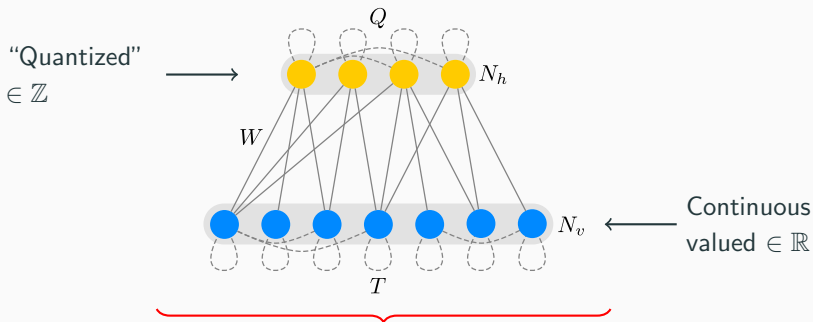
Under mild constraints on connection matrices (positive definiteness,...)

Riemann-Theta Boltzmann machine

How to change the status quo?

[Krefl, S.C., Haghighat, Kahlen '17]

Keep the inner sector couplings non-trivial, but the machine solvable?



$$P(v) \equiv \sqrt{\frac{\det T}{(2\pi)^{N_v}}} e^{-\frac{1}{2} v^t T v - B_v^t v - B_v^t T^{-1} B_v} \frac{\tilde{\theta}(B_h^t + v^t W | Q)}{\tilde{\theta}(B_h^t - B_v^t T^{-1} W | Q - W^t T^{-1} W)}$$

Closed form analytic solution still available!

Riemann-Theta Boltzmann machine

RTBM

Novel very generic probability density:

[Krefl, S.C., Haghighat, Kahlen '17]

$$P(v) \equiv \sqrt{\frac{\det T}{(2\pi)^{N_v}}} e^{-\frac{1}{2} v^t T v - B_v^t v - B_v^t T^{-1} B_v} \frac{\tilde{\theta}(B_h^t + v^t W | Q)}{\tilde{\theta}(B_h^t - B_v^t T^{-1} W | Q - W^t T^{-1} W)}$$

↑
Damping factor

← Riemann-Theta function

The Riemann-Theta definition:

$$\theta(z, \Omega) := \sum_{n \in \mathbb{Z}^{N_h}} e^{2\pi i \left(\frac{1}{2} n^t \Omega n + n^t z \right)}$$

Key properties: Periodicity, modular invariance, solution to heat equation, etc.

Note: Gradients can be calculated analytically as well so gradient descent can be used for optimization.

RTBM properties

We observe that $P(v)$ stays in the same distribution under affine transformations, *i.e.* rotation and translation

$$\mathbf{w} = A\mathbf{v} + b, \quad \mathbf{w} \sim P_{A,b}(v),$$

if the linear transformation A has full column rank.

$P_{A,b}(v)$ is the distribution $P(v)$ with parameters rotated as

$$\begin{aligned} T^{-1} &\rightarrow AT^{-1}A^t, & B_v &\rightarrow (A^+)^t B_v - Tb, \\ W &\rightarrow (A^+)^t W, & B_h &\rightarrow B_h - W^t b. \end{aligned}$$

where A^+ is the left pseudo-inverse defined as

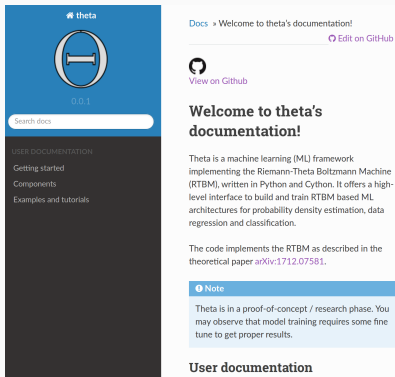
$$A^+ = (A^t A)^{-1} A^t.$$

Applications

Implementation

In order to perform tests we prepared a public RTBM framework:

Theta: Python machine learning framework for RTBMs and TNNs (with heavy lifting done by numpy, cython and C)



The screenshot shows the Theta framework documentation website. The header is blue with the Theta logo (a stylized 'O' with a horizontal bar) and the text 'theta' and '0.0.1'. Below the header is a search bar labeled 'Search docs'. The main content area is dark grey and contains a sidebar with 'USER DOCUMENTATION' links: 'Getting started', 'Components', and 'Examples and tutorials'. The main text area has a heading 'Welcome to theta's documentation!' and a subheading 'Docs » Welcome to theta's documentation!'. It includes a link to 'Edit on GitHub' and a 'View on GitHub' link. The text describes Theta as a machine learning (ML) framework implementing the Riemann-Theta Boltzmann Machine (RTBM), written in Python and Cython. It offers a high-level interface to build and train RTBM based ML architectures for probability density estimation, data regression and classification. A note mentions that the code implements the RTBM as described in the theoretical paper arXiv:1712.07581. A blue box with a note icon contains the text: 'Note: Theta is in a proof-of-concept / research phase. You may observe that model training requires some fine tune to get proper results.'

User documentation

- Easy interface: Keras like definition of model.
- SGD and genetic optimizer out of the box.
Easy integration of custom optimizers.
- Easy to extend functionality (object oriented)
- CPU based (for current version)

[<http://riemann.ai/theta>]

In the next we show examples of RTBMs for

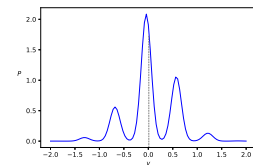
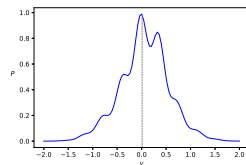
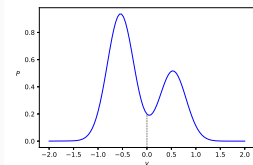
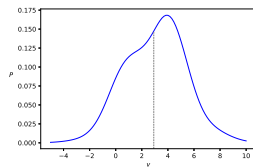
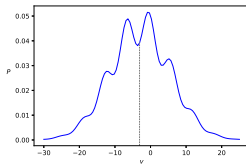
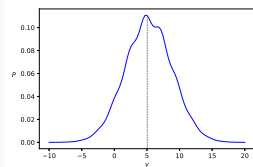
- Probability determination
- Probability sampling
- Conditional probability
- Feature detection for data classification
- Data regression

Probability determination

Riemann-Theta Boltzmann machine

RTBM $P(v)$ examples:

[Krefl, S.C., Haghighat, Kahlen '17]



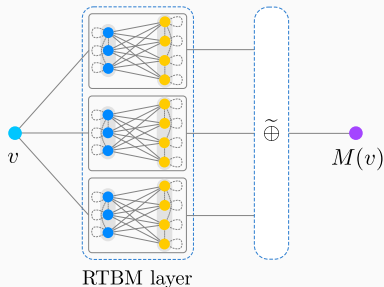
For different choices of parameters (with hidden sector in 1D or 2D).

Mixture model:

Expectation:

As long as the density is well enough behaved at the boundaries it can be learned by an RTBM mixture model.

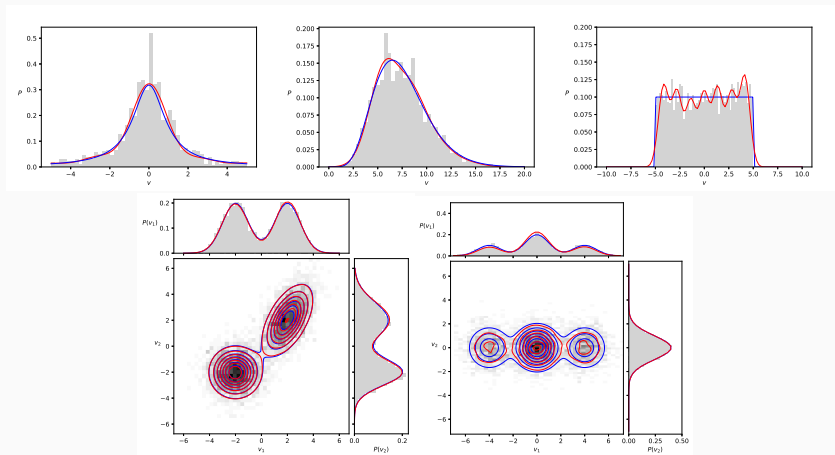
[Krefl, S.C., Haghighat, Kahlen '17]



Riemann-Theta Boltzmann machine

Examples:

[Krefl, S.C., Haghighat, Kahlen '17]



Top $N_v = 1$, $N_h = 3, 2, 3$, button $N_v = 2$, $N_h = 1$ (2x RTBM), 2.

Probability sampling

RTBM sampling algorithm

The probability for the visible sector can be expressed as:

$$P(v) = \sum_{[h]} P(v|h)P(h)$$

where $P(v|h)$ is a multivariate gaussian. The $P(v)$ sampling can be performed easily by:

- sampling $\mathbf{h} \sim P(h)$ using the RT numerical evaluation $\theta = \theta_n + \epsilon(R)$ with ellipsoid radius R so

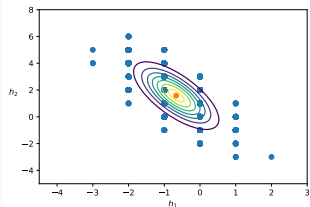
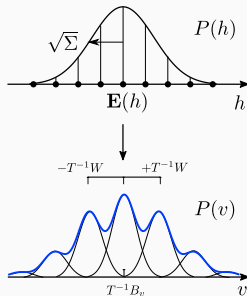
$$p = \frac{\epsilon(R)}{\theta_n + \epsilon(R)} \ll 1$$

is the probability that a point is sampled outside the ellipsoid of radius R , while

$$\sum_{[h](R)} P(h) = \frac{\theta_n}{\theta_n + \epsilon(R)} \approx 1$$

i.e. sum over the lattice points inside the ellipsoid.

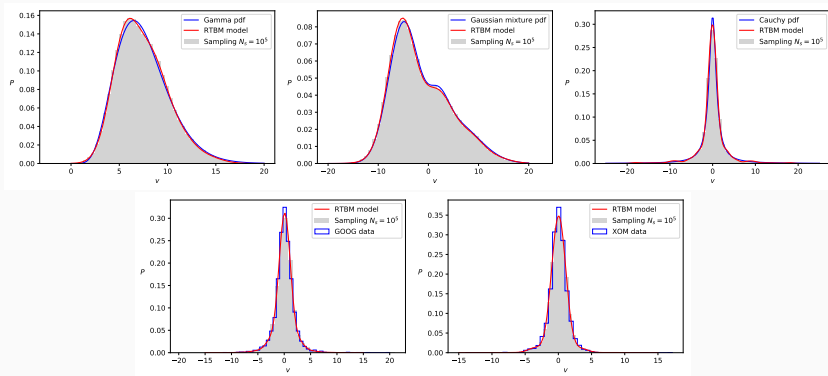
- then sampling $\mathbf{v} \sim P(v|\mathbf{h})$



Sampling examples

RTBM $P(v)$ sampling examples:

[S.C. and Krefl '18]



Top $N_v = 1$, $N_h = 2, 3$ (2x RTBM), 3, bottom $N_v = 1$, $N_h = 3$.

Sampling distance estimators

Distribution	$\chi^2_{\text{RTBM}}/N_{\text{bins}}$	$\text{MSE}_{\text{RTBM}}^{\text{sampling}}$	$\text{MSE}_{\text{pdf}}^{\text{sampling}}$	$\text{MSE}_{\text{RTBM}}^{\text{pdf}}$	KS distance
Gamma	0.02/50	$2 \cdot 10^{-5}$	$2.6 \cdot 10^{-5}$	$3.4 \cdot 10^{-4}$	0.01
Cauchy	0.12/50	$2.9 \cdot 10^{-4}$	$3.7 \cdot 10^{-4}$	$1.5 \cdot 10^{-3}$	0.02
Gaussian mixture	0.01/50	$6.7 \cdot 10^{-6}$	$1.4 \cdot 10^{-5}$	$9.3 \cdot 10^{-5}$	0.01
GOOG	0.10/50	$2.7 \cdot 10^{-4}$	$9.5 \cdot 10^{-3}$	$2.5 \cdot 10^{-4}$	0.02
XOM	0.09/50	$2.6 \cdot 10^{-4}$	$6.7 \cdot 10^{-3}$	$3.7 \cdot 10^{-4}$	0.02

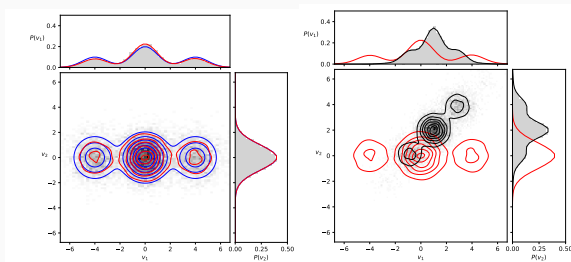
TABLE I: Distance estimators for the sampling examples in figures 3 and 4. Exact definitions for all distance estimators are given in section VII. The mean squared error (MSE) is taken between the sampling, the RTBM model and the underlying distribution (pdf). The Kolmogorov-Smirnov (KS) distance is shown in the last column of the table. For GOOG and XOM the empirical distribution is employed as underlying pdf.

Distribution	Mean	2nd moment	3th moment	4th moment
Gamma	7.43 (7.43) [7.49]	6.91 (6.89) [7.41]	10.03 (10.03) [13.79]	154 (153.23) [195.8]
Cauchy	-0.057 (-0.057) [-]	11.64 (11.64) [-]	-4.63 (-4.97) [-]	1749.8 (1753) [-]
Gaussian mixture	-1.48 (-1.48) [-1.31]	34.45 (34.45) [34.29]	134.35 (136.67) [131.78]	3558.7 (3571.8) [3569.1]
GOOG	0.06 (0.06) [0.08]	3.28 (3.23) [3.58]	1.52 (1.42) [6.04]	117 (108) [191]
XOM	0.02 (0.02) [0.03]	2.13 (2.15) [2.36]	-0.42 (-0.18) [1.44]	38.3 (40.2) [97.1]

TABLE II: Mean and central moments for the sampling data, the RTBM model (round brackets) and the underlying true distribution (square brackets). Note that the moments of the Cauchy distribution are either undefined or infinite. The given values correspond to the RTBM model approximation and its sampling, which are defined and finite, cf., 4. For the GOOG and XOM distributions the true moments (square brackets) are evaluated from the underlying empirical distribution.

Sampling examples with affine transformation

RTBM $P(v)$ sampling with affine transformation: [S.C. and Krefl '18]



For a rotation of $\theta = \pi/4$ and scaling of 2 ($N_v = 2$, $N_h = 2$).

Conditional probability

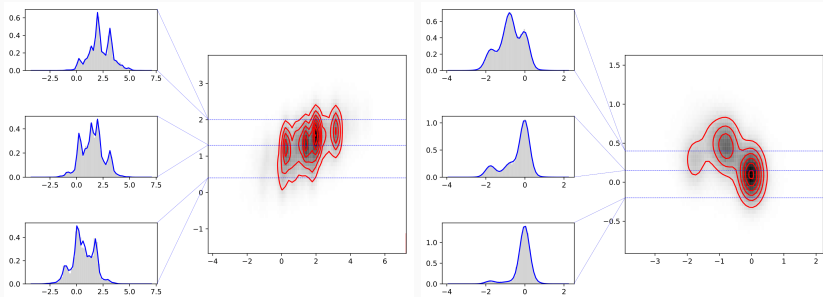
Conditional probability estimation

[Papaluca, S.C., Krefl '19 in preparation]

Considering a probability function $P(v)$ modelled by a RTBM, given some observed data d and some future outcome y , i.e. $v = (y, d)$:

$$P(y|d) = \frac{P(y, d)}{P(d)} = \frac{\sqrt{t_0}}{2\pi} e^{-\frac{1}{2}t_0 y^2 - B_0 y + \frac{1}{2t_0} B_0^2} \frac{\tilde{\theta}(B_h^t - v^t W | Q)}{\tilde{\theta}(B_h^t - r^t W | Q - \frac{W_0 W_0^t}{t_0})}$$

Examples in 2D:



Feature detection

Riemann-Theta Boltzmann machine

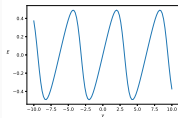
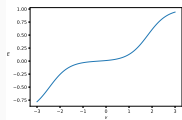
Feature detector:

New:

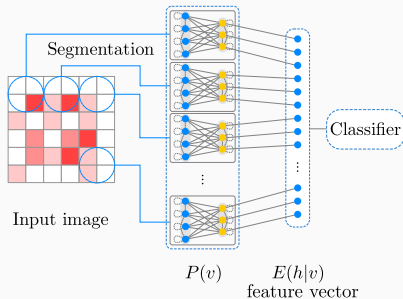
Conditional expectations of hidden states after training

$$E(h_i|v) = -\frac{1}{2\pi i} \frac{\nabla_i \tilde{\theta}(v^t W + B_h^t | Q)}{\tilde{\theta}(v^t W + B_h^t | Q)}$$

The detector is trained in probability mode and generates a feature vector.



[Krefl, S.C., Haghighat, Kahlen '17]
Similar to [Krizhevsky '09]



Feature detector example - jet classification

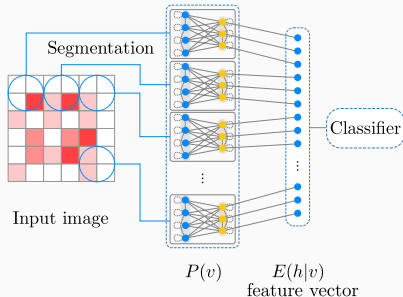
Jet classification:

[Krefl, S.C., Haghighat, Kahlen '17]
Data from [Baldi et al. '16, 1603.09349]

Discriminating jets from single hadronic particles and overlapping jets from pairs of collimated hadronic particles.

Data (images of 32x32 pixels)

- 5000 images for training
- 2500 images for testing



Classifier	Test dataset precision
Logistic regression (LR)	77%
RTBM feature detector + LR	83%

Data regression

Riemann-Theta Boltzmann machine

Theta Neural Network:

Idea:

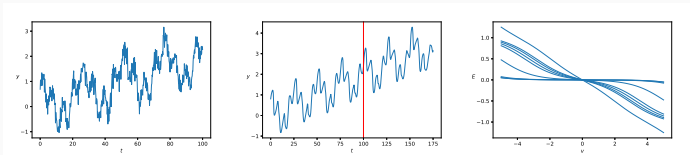
Use as activation function in a standard NN. The particular form of non-linearity is learned from data.

Key point:

smaller networks needed but Riemann-Theta evaluation is expensive.

Example (1:3-3-2:1):

$$y(t) = 0.02t + 0.5 \sin(t + 0.1) + 0.75 \cos(0.25t - 0.3) + \mathcal{N}(0, 1)$$

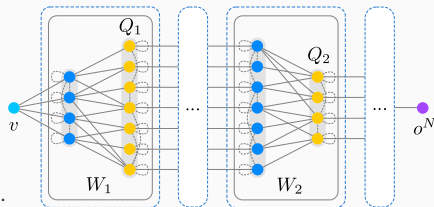


$y(t)$

TNN fit

TNN activations

[Krefl, S.C., Haghighat, Kahlen '17]



Conclusion

In summary:

- New BM architecture based on the Riemann-Theta function.
- Results are encouraging, several application opportunities.

For the future:

- Perform systematic benchmarks.
- Develop better optimization algorithms.
- Provide a more complete physics interpretation (if possible)

Thank you!