Product Jacobi-Theta Boltzmann machines with score matching

Andrea Pasquale, Daniel Krefl, Stefano Carrazza, Frank Nielsen

27th October 2022, ACAT 2022, Bari

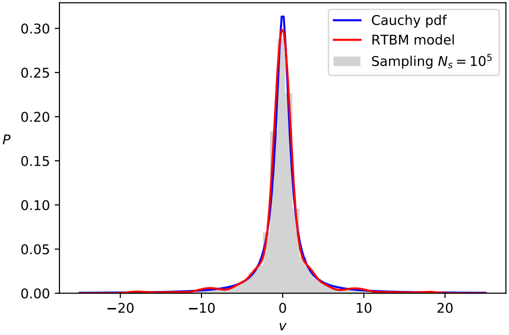
Introduction

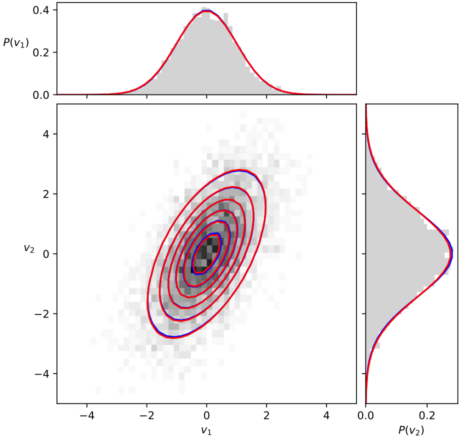
Introduction

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 started from Boltzmann Machines (BM).

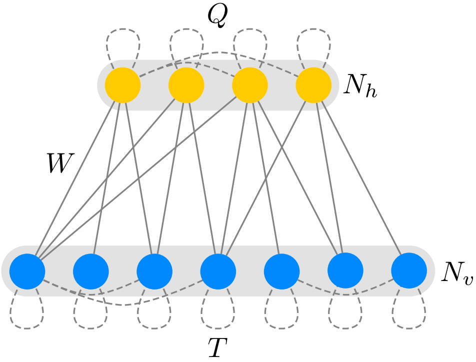




Theory

Boltzmann Machine (BM)

Graphical representation



Main Features:

- Visible sector with N_v nodes
- Hidden sector with N_h nodes
- Binary valued states **{0,1}** all the nodes
- Connection matrices Q, T and W between the nodes

This can be viewed as a statistical system with the following energy for a given state (v, h):

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

The probability of finding the system in the state v can be computed by marginalizing h

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

Density estimation using BM?

Can we use BM to perform density estimation?

Density estimation using BM?

Can we use BM to perform density estimation?

- Theoretically **yes**, since $P(v)$ is a **parametric** density.

Density estimation using BM?

Can we use BM to perform density estimation?

- Theoretically **yes**, since $P(v)$ is a **parametric** density.
- Practically **no** for a generic BM.

Density estimation using BM?

Can we use BM to perform density estimation?

- Theoretically **yes**, since $P(v)$ is a **parametric** density.
- Practically **no** for a generic BM.

Density estimation is possible only using Restricted BM (RBM).

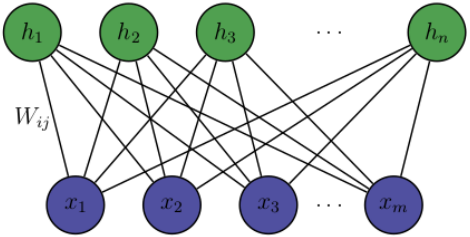


Figure 1: Graphical description of Restricted Boltzmann Machine

A RBM has trivial visible to visible and hidden to hidden units connections $(Q = T = 0)$

A new approach to BM

Can we keep the inner sector couplings non-trivial, but the machine solvable?

A possible solution is to go from binary values states to continuos/quantized values.

If we let $v_i \in \mathbb{R}$ and $h_j \in \mathbb{R}$ we get a *Continuous Boltzmann machine*

$$P(v) = \frac{e^{-\frac{1}{2}v^t(T-WQ^{-1}W^t)v+B_h^tQ^{-1}W^tv-\frac{1}{2}B^tA^{-1}B+\frac{1}{2}B_h^tQ^{-1}B_h}}{(2\pi)^{\frac{N_v}{2}}\sqrt{\det\Big((T-WQ^{-1}W^t)^{-1}\Big)}} ,$$

which is essentially a multivariate Gaussian $\Rightarrow$ trivial model.

If instead we let $v_i \in \mathbb{R}$ and $h_j \in \mathbb{Z}$ we get the following

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

non-trivial closed form solution under mild constraints!

Riemann-Theta BM

[Krefl et al., 2017]

Lets take a closer look at the probability density

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

Multivariate gaussian modulated by the functions θ which are known as Riemann-Theta functions:

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

Therefore we called this model Riemann-Theta BM or RTBM.

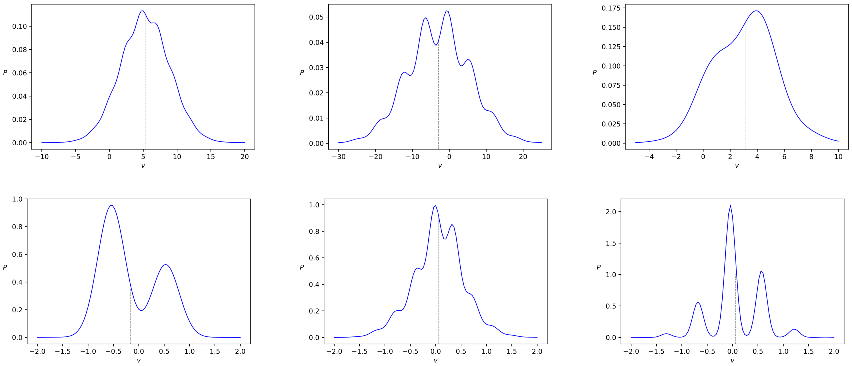


Figure 1: A few examples of $P(v)$ for different parameters

Properties of RTBMs

Learning

[Krefl et al., 2017]

With a RTBM we can perform density estimation to learn the parameters via maximum likelihood. That is, for N samples x_i from an unknown probability density we take the cost function

$$\mathcal{C} = -\sum_{i=1}^{N} \log P(x_i)$$

and we solve the optimization problem:

$$\underset{Q,T,W,B_h,B_v}{\arg\min} \quad \mathcal{C}$$

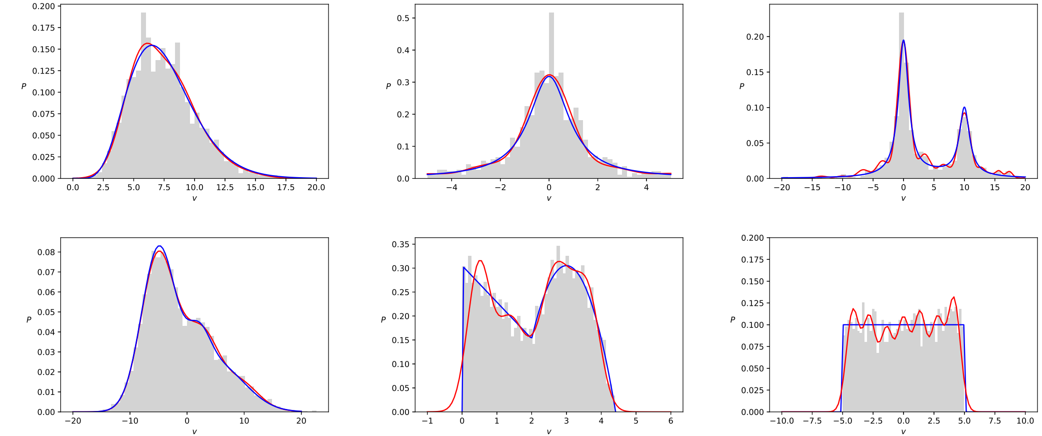
Gradient based techniques

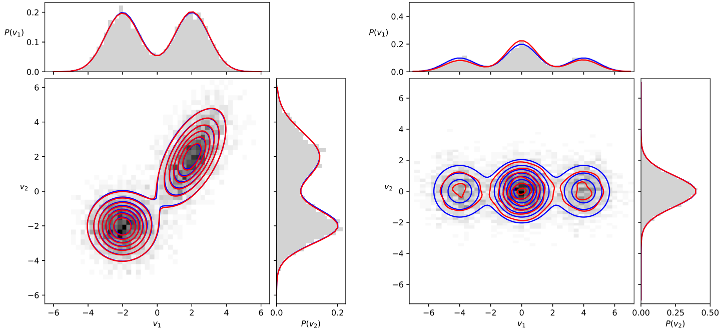
Since $P(v)$ is fully analytical we can solve the optimization problem using gradient or non-gradient based techniques[1].

[1] In this work we are using the CMA-ES optimizer.

Examples

[Krefl et al., 2017]





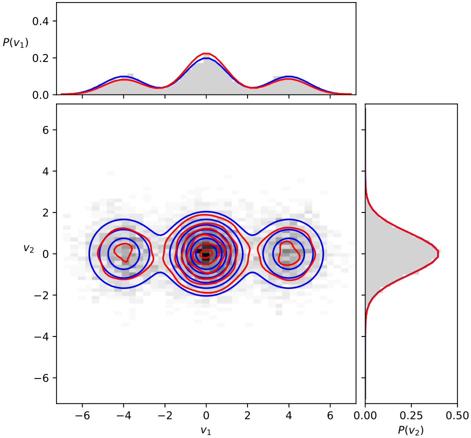
- Blue line: real distribution
- Red line: RTBM
- Histogram: sample from real distribution

Properties of RTBM

[Carrazza and Krefl, 2018]

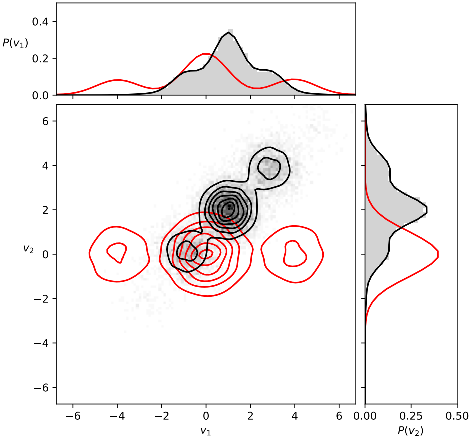
Over the last years we discoverd that the RTBM possesses a lot of nice properties.

Sampling



We can easily extract samples from $P(v)$ using numerical evaluation of θ functions

Affine transformations



$P(v)$ stays in the same distribution under affine transformations

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

Limitations

Limitations

Despite the promising results there is one major issue:

The learning process can become slow for ($N_h > 5$).

It is not possible to use RTBMs for:

- Complicated low-dimensional models
- High dimensional models ($N_h \geq N_v$)

Why this happens?

The computation of θ and its derivatives is a bottleneck for increasing value of N_h [2].

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

[2] The computational times increase exponentially due to the computation of the exponential of a $N_h \times N_h$ matrix required by the θ function.

Improving the RTBM

Factorizability

[Pasquale et al., 2022 in preparation]

To speed up the calculation we can exploit the following property of the θ function:

Factorizability

Under the assumption that Ω is a complex **diagonal** $N \times N$ matrix, whose imaginary part is positive definite, the Riemann-Theta function $\theta(z, \Omega)$ factorizes

$$\theta(z, \Omega) = \prod_{i=1}^{N} \theta(z_i, \Omega_{ii}) \tag{1}$$

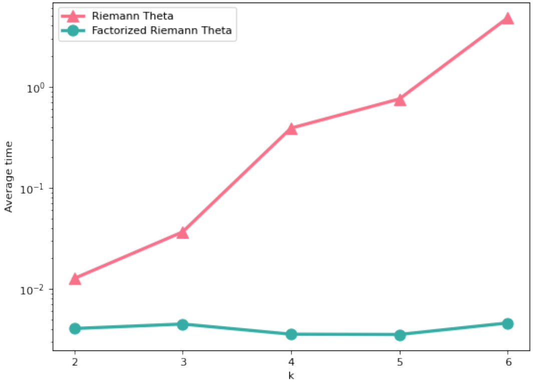


Figure 2: Average time to compute the RT using Deconinck et al., 2002

- Important speed-up as we increase N
- Computational times almost costant

Factorizability

[Pasquale et al., 2022 in preparation]

Can we exploit the previous property directly on $P(v)$?

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

We have to compute two θ function:

- $\boxed{\tilde{\theta}(B_h^t + v^t W | Q)}$ can we take Q diagonal? **Yes**
- $\boxed{\tilde{\theta}(B_h^t - B_v^t T^{-1} W | Q - W^t T^{-1} W)}$ $Q - W^t T^{-1} W$ diagonal? ***not feasible***

Obsv: the second term is a normalization term.

Idea: Can we select a specific cost function to avoid computing that term?

Score Matching

[Lyu, 2012]

A particular parameter learning methodology that can address this issue is *Score Matching*, which is based on the Fisher divergence.

$$D_F(p||q_\xi) = \int p(x) \left| \frac{\nabla_x\ p(x)}{p(x)} - \frac{\nabla_x\ q(x,\xi)}{q(x,\xi)} \right|^2 dx\ ,$$

which is slightly different from the Kullback-Leiber divergence:

$$D_{KL}(p||q_\xi) = \int p(x) \log \frac{p(x)}{q(x,\xi)} dx\ ,$$

We will show in the next slide that the normalization terms cancel out since

$$D_F \propto \frac{d^{(i)}}{dx^{(i)}} \log q(x,\xi)$$

Therefore if we start by assuming that Q is diagonal the learning process will involve only the computation of 1d θ functions!

Score Matching

[Pasquale et al., 2022 in preparation]

We can simplify the expression for D_F under the assumption that our model $q(x,\theta)$ is sufficiently regular, which is the case of the RTBM.

$$D_F(p||q_\theta) = \int p(x)\Big(\big|\nabla_x \log q(x,\theta)\big|^2 + 2\Delta_x \log q(x,\theta)\Big) + \text{const}$$

$$\approx \boxed{\textstyle\sum_{i=1}^N \left|\nabla_{v_i} \log q(v_i,\theta)\right|^2 + 2\Delta_{v_i} \log q(v_i,\theta)} + \text{const}\,.$$

$$\approx \mathcal{C}_F$$

We call $\mathcal{C}_F$ Fisher cost function and it is particularly useful for **non-normalizable** models.

Applications

Applications

In the following slides we show how the RTBM perform using empirical datasets.

- We start from low dimensional datasets and we want to generate more data: **data augmentation**.
- To quantify the quality of the samples generated we use a 2-D implementation of the Kolmogorov-Smirnov known as Fasano-Franceschini test.[3].
- We compare the performance of this model with state-of-the-art density estimation techniques such as Kernel Density Estimation [4] (KDE) and Normalizing Flows[5] (NF).

[3] For more details about this multi-dimensional generalization of the KS test see Fasano and Franceschini, 1987.

[4] We use a gaussian kernel where the optimal bandwidth is selected after a hyperoptimization using a grid search cross validation.

[5] For the NF we employ an architecture with rational-quadratic coupling transforms as in Durkan et al., 2019 trained using maximum likelihood estimation

Uranium dataset

[Pasquale et al., 2022 in preparation]

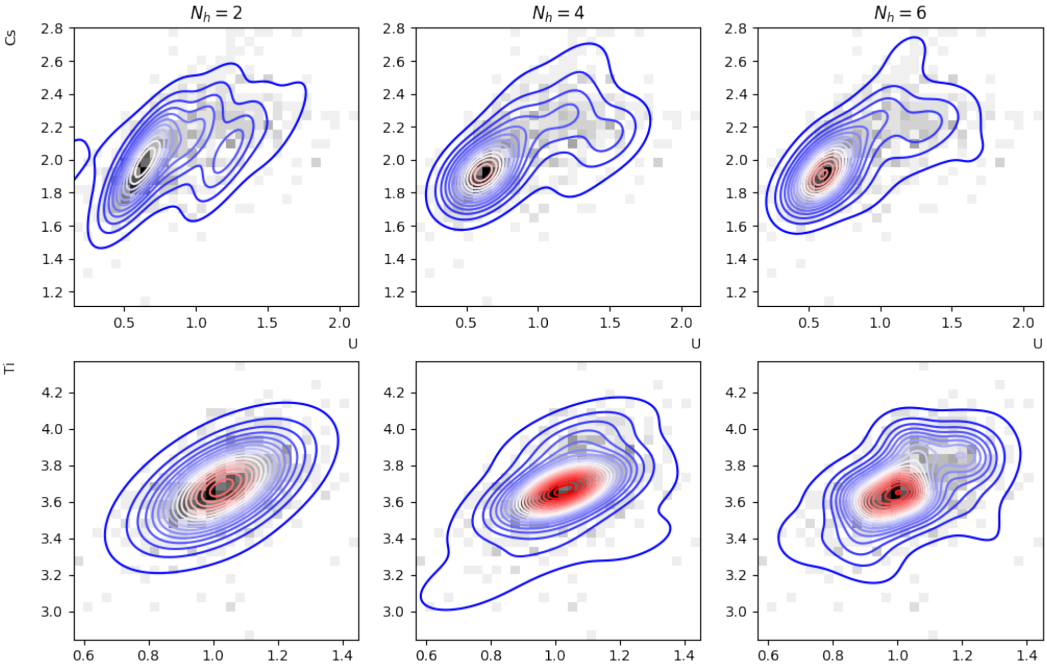
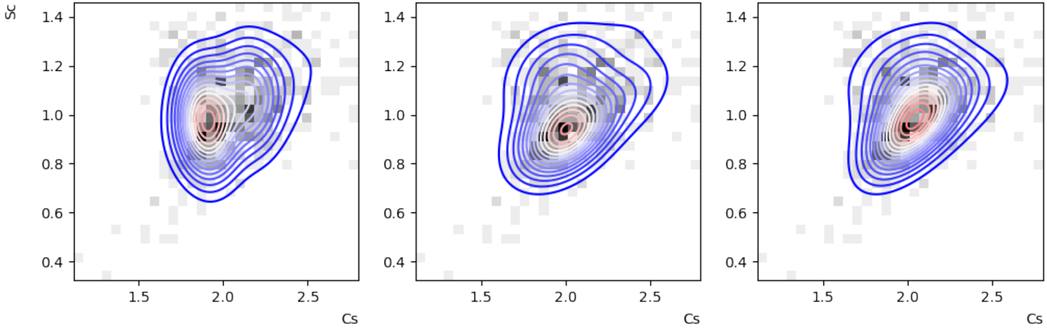
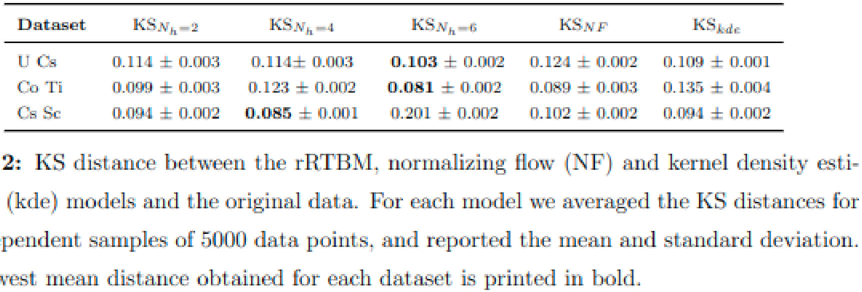


Figure 3: rRTBMs modelling the concentrations of Uranium and Cesium (first row), Cobalt and Titanium (second row) and, Cesium and Scandium (third row) for $N_h = 2, 4, 6$ (left,center,right). The rRTBM contours and histograms of the original data are shown.

Uranium dataset

[Pasquale et al., 2022 in preparation]





Dataset	$\mathrm{KS}_{N_h=2}$	$\mathrm{KS}_{N_h=4}$	$\mathrm{KS}_{N_h=6}$	KS_{NF}	KS_{kde}
U Cs	0.114 ± 0.003	0.114± 0.003	**0.103** ± 0.002	0.124 ± 0.002	0.109 ± 0.001
Co Ti	0.099 ± 0.003	0.123 ± 0.002	**0.081** ± 0.002	0.089 ± 0.003	0.135 ± 0.004
Cs Sc	0.094 ± 0.002	**0.085** ± 0.001	0.201 ± 0.002	0.102 ± 0.002	0.094 ± 0.002

Table 2: KS distance between the rRTBM, normalizing flow (NF) and kernel density estimation (kde) models and the original data. For each model we averaged the KS distances for 10 independent samples of 5000 data points, and reported the mean and standard deviation. The lowest mean distance obtained for each dataset is printed in bold.

Faithful dataset

[Pasquale et al., 2022 in preparation]

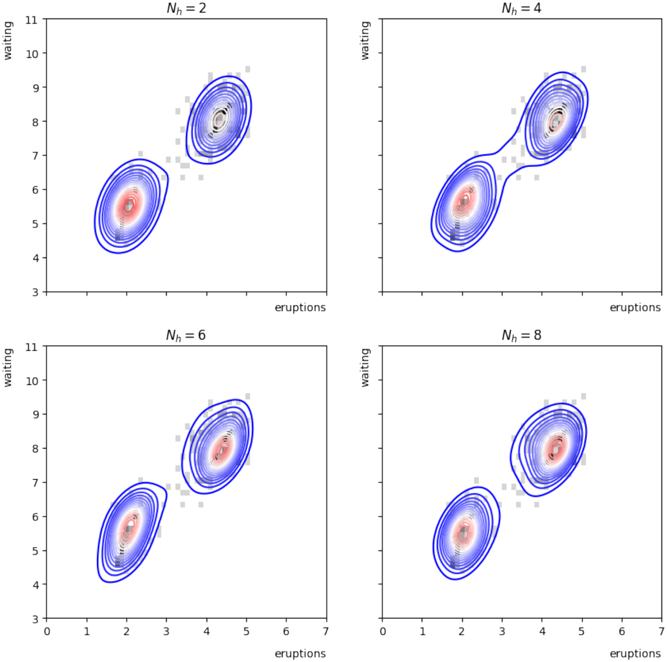


Figure 4: rRTBMs trained to model the waiting time between eruptions and the duration of the eruption for the Old Faithful geyser for $N_h = 2$ (top left), $N_h = 4$ (top right), $N_h = 6$ (bottom left) and $N_h = 8$ (bottom right). The curves correspond to the rRTBM model and the gray histogram is obtained from the original data with 30 bins.

Iris dataset

[Pasquale et al., 2022 in preparation]

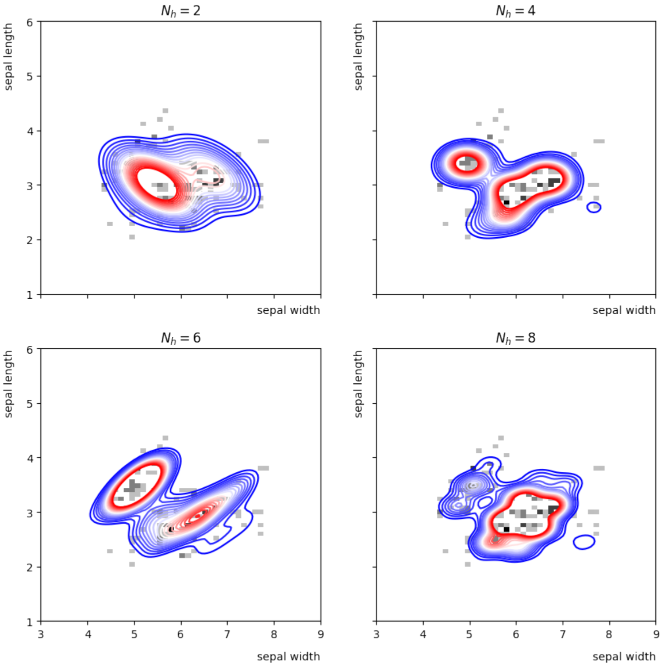
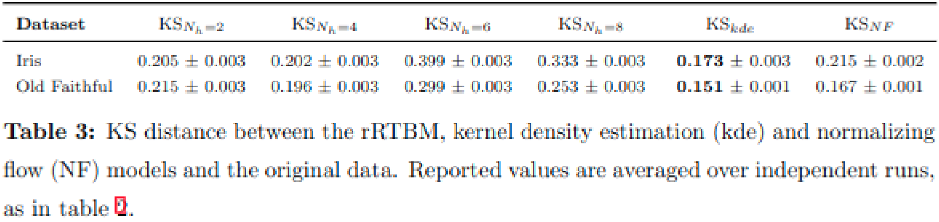


Figure 5: rRTBMs trained to model the joint distribution of sepal width and sepal length from the Iris dataset for $N_h = 2$ (top left), $N_h = 4$ (top right), $N_h = 6$ (bottom left) and $N_h = 8$ (bottom right). The curves correspond to the rRTBM model and the gray histogram is obtained from the original data with 30 bins.

Results

[Pasquale et al., 2022 in preparation]



Dataset	$\text{KS}_{N_h=2}$	$\text{KS}_{N_h=4}$	$\text{KS}_{N_h=6}$	$\text{KS}_{N_h=8}$	KS_{kde}	KS_{NF}
Iris	0.205 ± 0.003	0.202 ± 0.003	0.399 ± 0.003	0.333 ± 0.003	**0.173** ± 0.003	0.215 ± 0.002
Old Faithful	0.215 ± 0.003	0.196 ± 0.003	0.299 ± 0.003	0.253 ± 0.003	**0.151** ± 0.001	0.167 ± 0.001

Table 3: KS distance between the rRTBM, kernel density estimation (kde) and normalizing flow (NF) models and the original data. Reported values are averaged over independent runs, as in table 2.

- The RTBM is the model with the lowest KS for the `uranium` dataset.
- In the other cases it is still competitive.
- We successfully generate more data (5000) starting from low-sized datasets (≈ 200 points)

Conclusion

Outlook

In summary

- The RTBM is a valid model to perform density estimation even when Q is diagonal
- Using score matching we are able to train efficiently using large values of N_h
- Open source code soon available here: `https://github.com/RiemannAI/theta`

For the future

- Speed up the computation of the θ by moving to a GPU pr FPGA implementation
- Possibility to use this mechanism to perform MC multi-dimensional integration for physics related problem

Thanks for listening!

Bibliography i

References

Carrazza, Stefano and Daniel Krefl (2018). “Sampling the Riemann-Theta Boltzmann Machine”. In: *Comput. Phys. Commun.* 256, p. 107464. doi: 10.1016/j.cpc.2020.107464. arXiv: 1804.07768 [stat.ML].

Deconinck, Bernard et al. (2002). *Computing Riemann Theta Functions*. arXiv: nlin/0206009 [nlin.SI].

Durkan, Conor et al. (2019). *Neural Spline Flows*. doi: 10.48550/ARXIV.1906.04032. url: https://arxiv.org/abs/1906.04032.

Fasano, G. and A. Franceschini (Mar. 1987). “A multidimensional version of the Kolmogorov–Smirnov test”. In: *Monthly Notices of the Royal Astronomical Society* 225.1, pp. 155–170. issn: 0035-8711. doi: 10.1093/mnras/225.1.155. eprint: https://academic.oup.com/mnras/article-pdf/225/1/155/18522274/mnras225-0155.pdf. url: https://doi.org/10.1093/mnras/225.1.155.

Krefl, Daniel et al. (May 2017). “Riemann-Theta Boltzmann machine”. In: *Neurocomputing* 388, pp. 334–345. issn: 0925-2312. doi: 10.1016/j.neucom.2020.01.011. url: http://dx.doi.org/10.1016/j.neucom.2020.01.011.

Bibliography ii

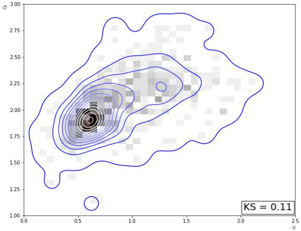
- Labrande, Hugo (Nov. 2015). “Computing Jacobi’s θ in quasi-linear time”. In: *Mathematics of Computation* 87. doi: `10.1090/mcom/3245`.
- Lyu, Siwei (2012). *Interpretation and Generalization of Score Matching*. arXiv: `1205.2629 [cs.LG]`.
- Pasquale et al., In preparation (2022). *Product Jacobi-Theta Boltzmann machines with score matching*.

Backup slides

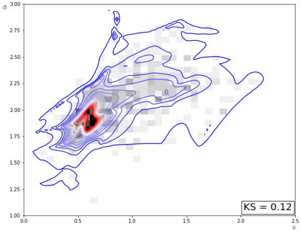
Overfitting problems

A common downside Goodness of Fit test is the fact that they do not take into account the complexity of the model.

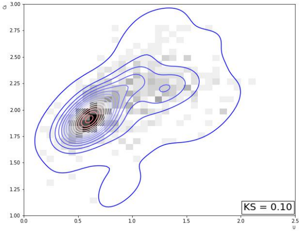
KDE



NFLOW



RTBM

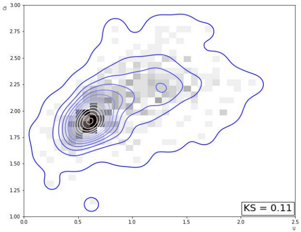


Is there a way to quantify the overfitting behavior of a model?

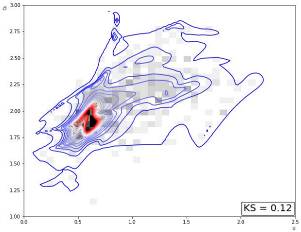
Overfitting problems

A common downside Goodness of Fit test is the fact that they do not take into account the complexity of the model.

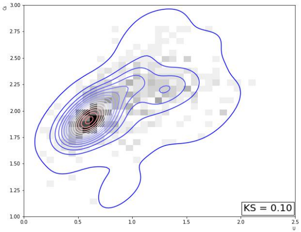
KDE



NFLOW



RTBM



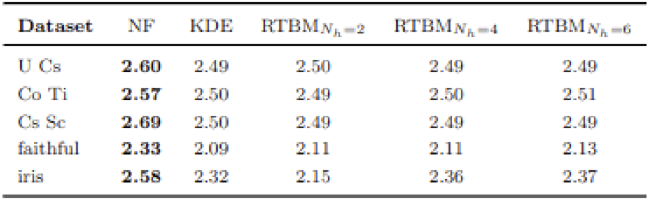
Is there a way to quantify the overfitting behavior of a model?

Fractal dimension!

What is the fractal dimension?

We can associate a continuum dimension to lines and surfaces.

What happens when we apply this to the surface generate from the previous models?



Dataset	NF	KDE	RTBM$_{N_h=2}$	RTBM$_{N_h=4}$	RTBM$_{N_h=6}$
U Cs	**2.60**	2.49	2.50	2.49	2.49
Co Ti	**2.57**	2.50	2.49	2.50	2.51
Cs Sc	**2.69**	2.50	2.49	2.49	2.49
faithful	**2.33**	2.09	2.11	2.11	2.13
iris	**2.58**	2.32	2.15	2.36	2.37

➔ We can quantify the overfitting in terms of the fractal dimension.

Gradients for RTBM

[Pasquale et al., 2022 in preparation]

$$\partial_{v_i} \log P(v) = -(Tv)_i - (B_v)_i + (WD)_i \,,$$
$$\partial^2_{v_i} \log P(v) = -T_{ii} + (WHW^t)_{ii} + (WD)_i^2 \,,$$

with D the normalized gradient and H the normalized hessian

$$(D)_i = \frac{\nabla_i \tilde{\theta}(B_h^t + v^t W|Q)}{\tilde{\theta}(B_h^t + v^t W|Q)} \,, \quad (H)_{ij} = \frac{\nabla_i \nabla_j \tilde{\theta}(B_h^t + v^t W|Q)}{\tilde{\theta}(B_h^t + v^t W|Q)} \,.$$

If Q is diagonal

$$\partial_{v_i} \log P(v) = -(Tv)_i - (B_v)_i + \sum_{j=1}^{N_h} \frac{\partial_{v_i} \tilde{\theta}((B_h^t + v^t W)_j | Q_{jj})}{\tilde{\theta}((B_h^t + v^t W)_j | Q_{jj})} W_{ji} \,,$$

$$\begin{aligned}\partial^2_{v_i} \log P(v) = & - T_{ii} + \sum_{j=1}^{N_h} \frac{\partial^2_{v_i} \tilde{\theta}((B_h^t + v^t W)_j | Q_{jj})}{\tilde{\theta}((B_h^t + v^t W)_j | Q_{jj})} W_{ji}^2 \\ & - \sum_{j=1}^{N_h} \frac{(\partial_{v_i} \tilde{\theta}((B_h^t + v^t W)_j | Q_{jj}))^2}{\tilde{\theta}((B_h^t + v^t W)_j | Q_{jj})} W_{ji} \,.\end{aligned}$$

The cost function can be evaluated using only 1d RT functions!

Efficient calculation of the 1d RT

[Pasquale et al., 2022 in preparation]

The computation of the RT and its derivatives is computationally challenging due to the infinite sum over an N-dimensional integer lattice $\mathbb{Z}^N$

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

For the multi-dimensional case we can obtain a numerical approximation by summing over an finite subset of lattice points.

For the 1d case there exist more efficient methods. A possibility is to truncate the series:

$$\theta(z, \Omega) \approx S_B(z, \Omega) = 1 + \sum_{0<n<B} q^{n^2} (e^{2\pi i n z} + e^{-2\pi i n z}) =: 1 + \sum_{0<n<B} v_n \,.$$

It can be shown (see Labrande, 2015) that v_n can be computed recursively, giving us a fast algorithm to evaluate the RT in dim 1.

$$v_{n+1} = q^{2n} v_1 v_n - q^{4n} v_{n-1} \,. \tag{2}$$

where $q = e^{i\pi\Omega}$.

What about the gradients?

[Pasquale et al., 2022 in preparation]

To speed up the learning process for the RTBM we would like to have a similar algorithm for the derivatives of the RT function. In our work he prove that this is possible:

$$\frac{d}{dz}\theta(z,\Omega) \approx U_B(z,\Omega) = \sum_{1<n<B} -4\pi n q^{n^2} \sin(2\pi n z) =: \sum_{1<n<B} w_n \ ,$$

$$\frac{d^2}{dz^2}\theta(z,\Omega) \approx V_B(z,\Omega) = \sum_{1<n<B} -8\pi^2 n^2 q^{n^2} \cos(2\pi n z) =: \sum_{1<n<B} \xi_n \ .$$

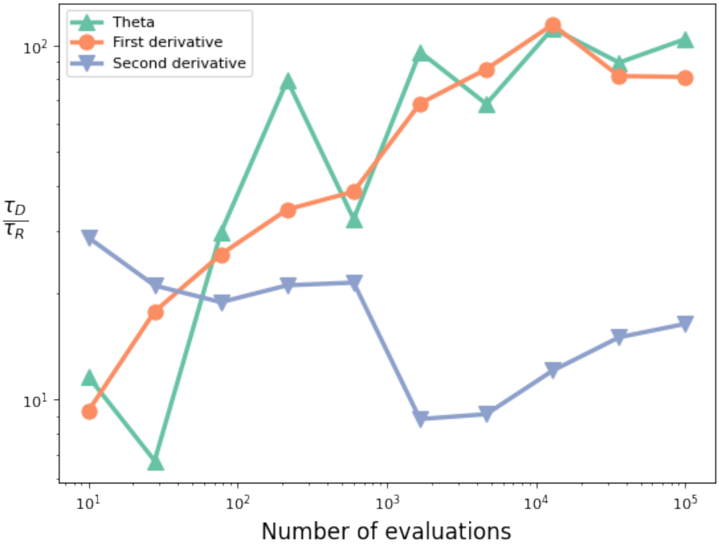
After a few mathematical passages it can be shown that there exist a recurrence to compute both w_n and ξ_n.

$$w_{n+1} = (n+1)\left[\frac{2\cos(2\pi z)}{n} q^{2n+1} w_n - \frac{q^{4n}}{n-1} w_{n-1}\right],$$

$$\xi_{n+1} = (n+1)^2\left[\frac{2\cos(2\pi z)}{n^2} q^{2n+1} \xi_n - \frac{1}{(n-1)^2} q^{4n} \xi_{n-1}\right].$$

Results

[Pasquale et al., 2022 in preparation]



Sampling algorithm

[Carrazza and Krefl, 2018]

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. $P(v)$ can be easily sampled using the following algorithm:

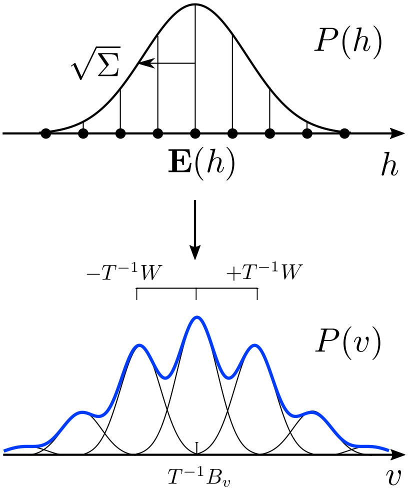
- sample $\mathbf{h} \sim P(h)$ using RT numerical evaluation $\theta = \theta_n + \epsilon(R)$ with ellipsoid radius R such that

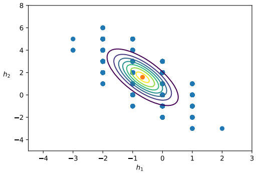
  $$p = \frac{\epsilon(R)}{\theta_n + \epsilon(R)} \ll 1$$

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

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

  is the sum over the lattice points inside the ellipsoid.
- sample $\mathbf{v} \sim P(v|\mathbf{h})$

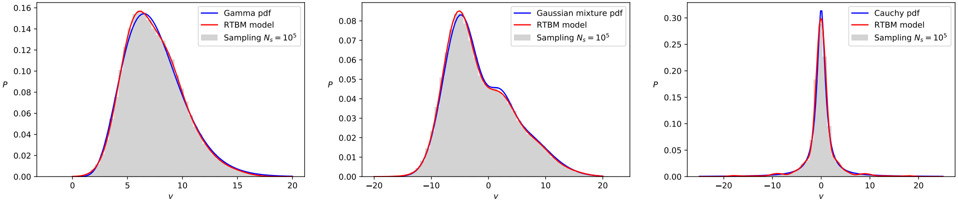




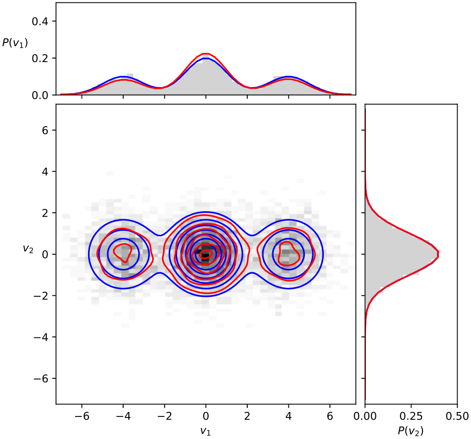
Sampling examples

[Carrazza and Krefl, 2018]

One dimensional case:



Multi-dimensional case



- Blue line: Real distribution
- Red line: RTBM
- Histogram: sample from RTBM

Affine Transformations

[Carrazza and Krefl, 2018]

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

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

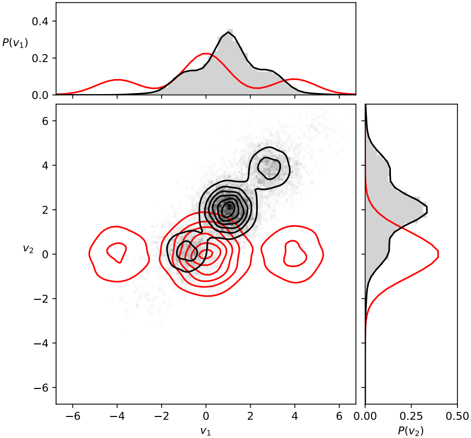
if the linear transformation A has a full column rank. The connection matrices and the biases of the transformed RTBM are given by:

$$T^{-1} \to AT^{-1}A^t\,, \quad B_v \to (A^+)^t B_v - Tb\,,$$
$$W \to (A^+)^t W\,, \quad B_h \to B_h - W^t b\,.$$

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

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

Example: rotation of $\theta/4$ and scaling of $1/2$ $(N_v = 2, N_h = 2)$



$P(v_1)$
0.4
0.2
0.0
6
4
2
v_2 0
−2
−4
−6
−6
−4
−2
0
2
4
6
v_1
0.00
0.25
0.50
$P(v_2)$