NETFLIX

Restricted Boltzmann Machines

Markus Pawellek

January 24, 2019

Outline

Problem

Problem: Collaborative Filtering – Movie Ratings

	Star Trek	The Matrix	Van Helsing	Harry Potter	The Hobbit
James T. Kirk	1	1	$\times$	0	$\times$
Trinity	$\times$	1	0	1	1
Anna Valerious	$\times$	$\times$	1	$\times$	0
Severus Snape	0	1	0	1	0
Thorin Oakenshield	1	1	1	$\times$	0

Problem: Collaborative Filtering – Movie Ratings

	Star Trek	The Matrix	Van Helsing	Harry Potter	The Hobbit
James T. Kirk	1	1	$\times$	0	$\times$
Trinity	$\times$	1	0	1	1
Anna Valerious	$\times$	$\times$	1	$\times$	0
Severus Snape	0	1	0	1	0
Thorin Oakenshield	1	1	1	$\times$	0

Goal:

Problem: Collaborative Filtering – Movie Ratings

	Star Trek	The Matrix	Van Helsing	Harry Potter	The Hobbit
James T. Kirk	1	1	$\times$	0	$\times$
Trinity	$\times$	1	0	1	1
Anna Valerious	$\times$	$\times$	1	$\times$	0
Severus Snape	0	1	0	1	0
Thorin Oakenshield	1	1	1	$\times$	0

Goal:

- approximately represent a complex probability distribution

Problem: Collaborative Filtering – Movie Ratings

	Star Trek	The Matrix	Van Helsing	Harry Potter	The Hobbit
James T. Kirk	1	1	$\times$	0	$\times$
Trinity	$\times$	1	0	1	1
Anna Valerious	$\times$	$\times$	1	$\times$	0
Severus Snape	0	1	0	1	0
Thorin Oakenshield	1	1	1	$\times$	0

Goal:

- approximately represent a complex probability distribution
- learn probability distribution based on given samples

Problem: Collaborative Filtering – Movie Ratings

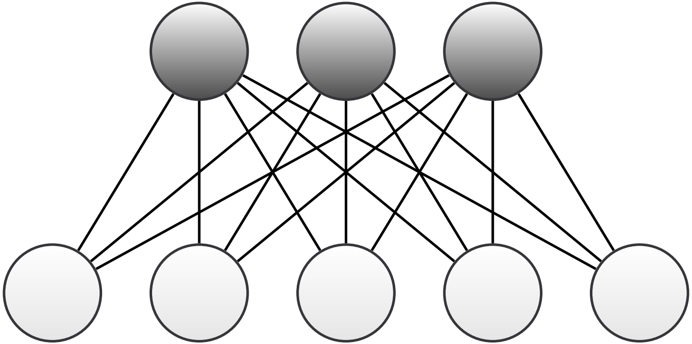
	Star Trek	The Matrix	Van Helsing	Harry Potter	The Hobbit
James T. Kirk	1	1	$\times$	0	$\times$
Trinity	$\times$	1	0	1	1
Anna Valerious	$\times$	$\times$	1	$\times$	0
Severus Snape	0	1	0	1	0
Thorin Oakenshield	1	1	1	$\times$	0

Goal:

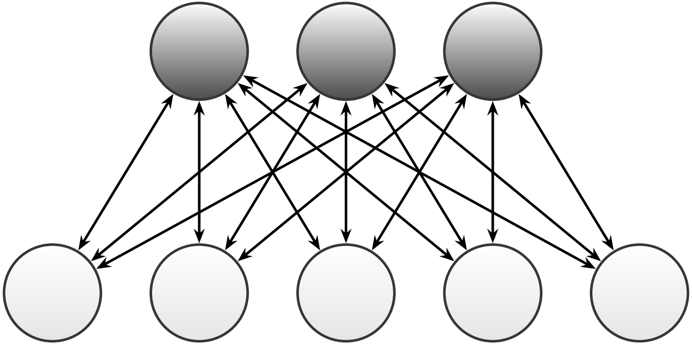
- approximately represent a complex probability distribution
- learn probability distribution based on given samples
- make predictions based on learned parameters

The Model

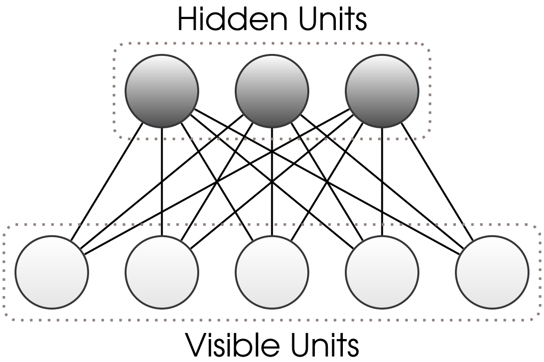
The Model: Idea



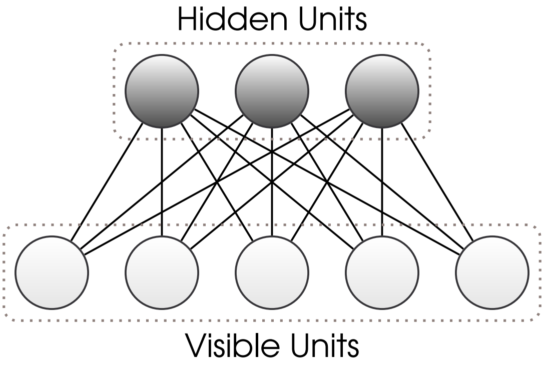
The Model: Idea



The Model: Idea

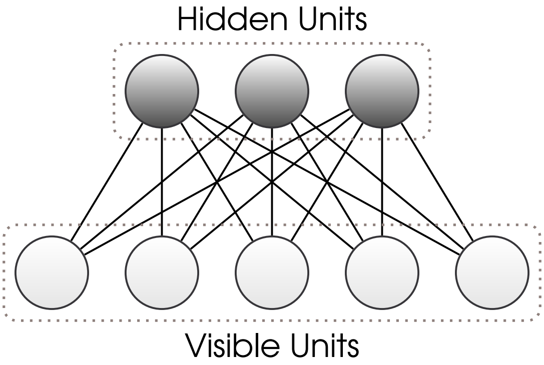


The Model: Idea



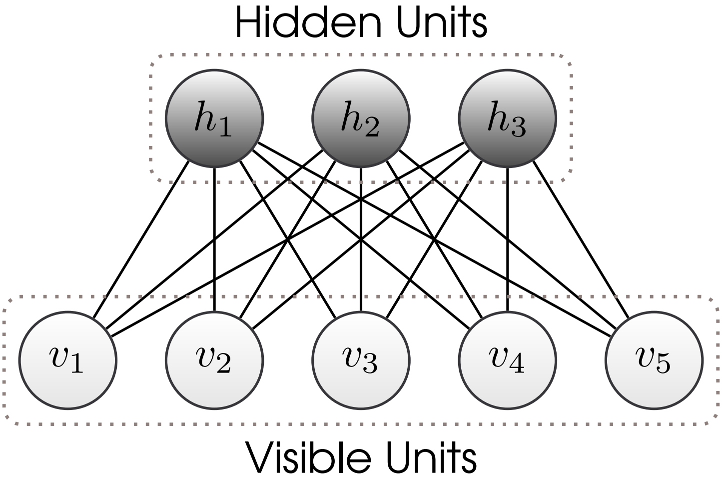
- units are divided into two subsets

The Model: Idea

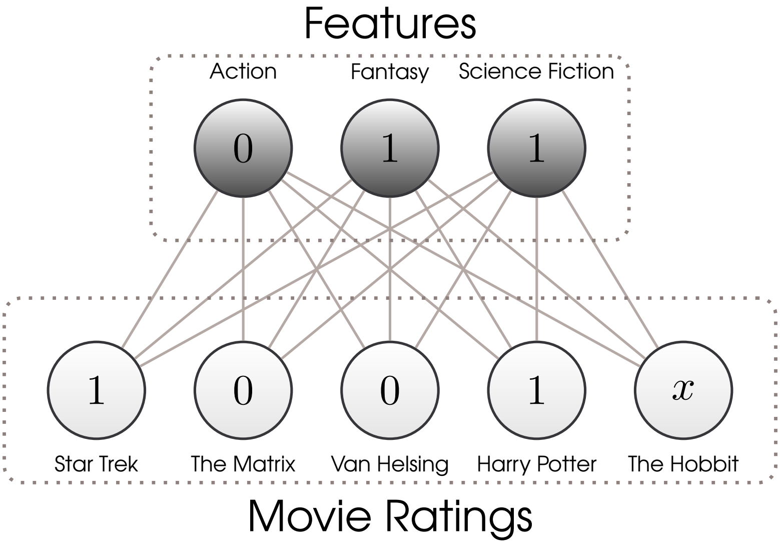


- units are divided into two subsets
- only connections between hidden and visible units are allowed

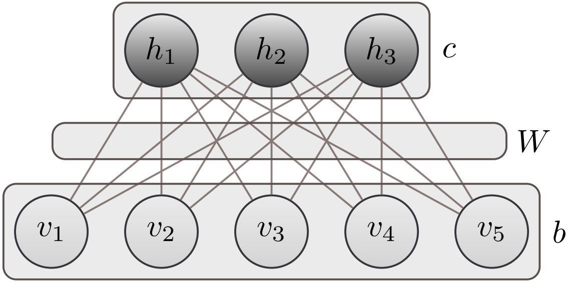
The Model: Idea – Inputs



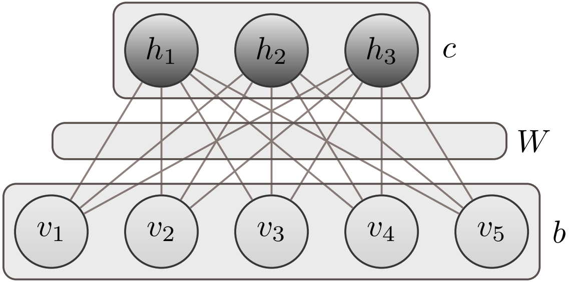
The Model: Idea – Example



The Model: Parameters



The Model: Parameters



$$v \in V \coloneqq \{0,1\}^n \qquad h \in H \coloneqq \{0,1\}^m \qquad \vartheta \coloneqq (W, b, c) \in \mathbb{R}^{(n\times m)+n+m}$$

The Model: Probability Distribution and Energy

$$p[\vartheta]\colon V \times H \to [0,1] \qquad p[\vartheta](v,h) := \frac{e^{-E[\vartheta](v,h)}}{Z(\vartheta)}$$

The Model: Probability Distribution and Energy

$$p[\vartheta]\colon V \times H \to [0,1] \qquad p[\vartheta](v,h) := \frac{e^{-E[\vartheta](v,h)}}{Z(\vartheta)}$$

$$E[\vartheta]\colon V \times H \to \mathbb{R} \qquad E[\vartheta](v,h) := -v^{\mathrm{T}}Wh - v^{\mathrm{T}}b - h^{\mathrm{T}}c$$

The Model: Probability Distribution and Energy

$$p[\vartheta]\colon V \times H \to [0,1] \qquad p[\vartheta](v,h) := \frac{e^{-E[\vartheta](v,h)}}{Z(\vartheta)}$$

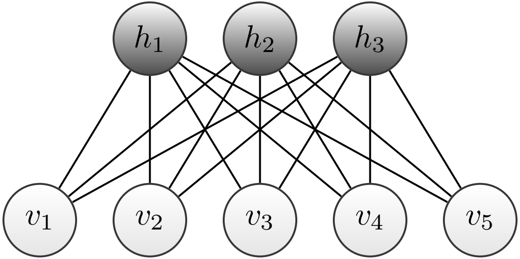
$$E[\vartheta]\colon V \times H \to \mathbb{R} \qquad E[\vartheta](v,h) := -v^{\mathrm{T}}Wh - v^{\mathrm{T}}b - h^{\mathrm{T}}c$$

$$Z(\vartheta) := \sum_{v\in V}\sum_{h\in H} e^{-E[\vartheta](v,h)}$$

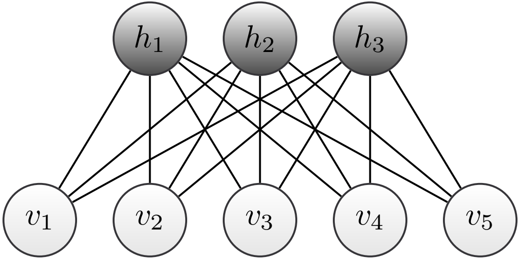
The Model: Probability Distribution for Visible Units

$$p[\vartheta]\colon V \to [0,1] \qquad p[\vartheta](v) := \sum_{h \in H} p[\vartheta](v,h)$$

The Model: Posterior Probability



The Model: Posterior Probability



$$p[\vartheta](h|v) = \prod_{j=1}^{m} p[\vartheta]\left(h_j = 1|v\right)$$

Learning

Learning: Maximum Likelihood Estimation

$$\mathcal{S} \in V^s \qquad \mathcal{L}[\mathcal{S}] \colon \mathbb{R}^{n \times m + n + m} \to \mathbb{R} \qquad \mathcal{L}[\mathcal{S}](\vartheta) := \frac{1}{s} \sum_{k=1}^{s} \ln p[\vartheta]\,(\mathcal{S}_k)$$

- maximize the product of probabilities of given samples
- equivalent to maximizing log-likelihood function

Learning: Gradient Ascent

$$\nabla_W \mathcal{L}[\mathcal{S}](\vartheta) = \frac{1}{s}\sum_{k=1}^{s} \mathbb{E}_\vartheta\left[\mathcal{V}\mathcal{H}^{\mathrm{T}} \middle| \mathcal{S}_k\right] - \mathbb{E}_\vartheta\left[\mathcal{V}\mathcal{H}^{\mathrm{T}}\right]$$

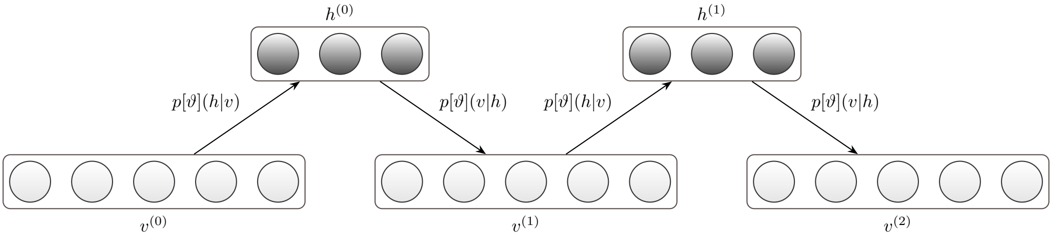
- use stochastic gradient ascent with minibatches

Learning: Gradient Ascent

$$\nabla_W \mathcal{L}[\mathcal{S}](\vartheta) = \frac{1}{s}\sum_{k=1}^{s} \mathbb{E}_\vartheta\left[\mathcal{V}\mathcal{H}^{\mathrm{T}}\middle|\mathcal{S}_k\right] - \mathbb{E}_\vartheta\left[\mathcal{V}\mathcal{H}^{\mathrm{T}}\right]$$

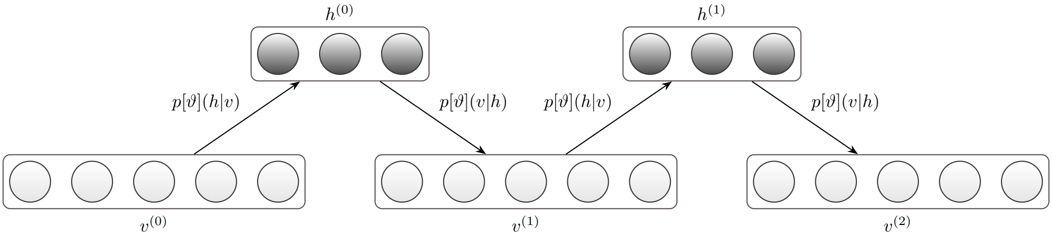
- use stochastic gradient ascent with minibatches
- evaluating the gradient introduces problems

Learning: Gibbs Sampling



- to estimate $\mathbb{E}_\vartheta\left[\mathcal{V}\mathcal{H}^{\mathrm{T}}\right]$ perform Gibbs sampling

Learning: Gibbs Sampling



- to estimate $\mathbb{E}_\vartheta\left[\mathcal{V}\mathcal{H}^\mathrm{T}\right]$ perform Gibbs sampling
- slow because it has to reach equilibrium

Learning: Contrastive Divergence

- abort Gibbs Sampling after $v^{(k)}$ and $h^{(k)}$ are computed

Learning: Contrastive Divergence

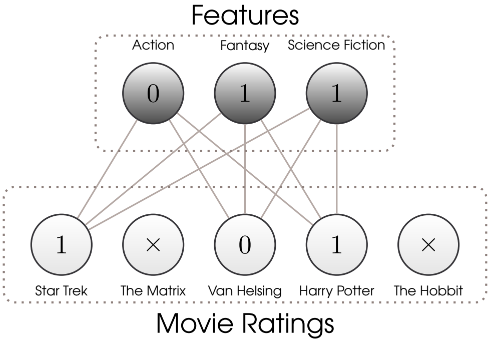
- abort Gibbs Sampling after $v^{(k)}$ and $h^{(k)}$ are computed
- approximate the expectation value

Learning: Contrastive Divergence

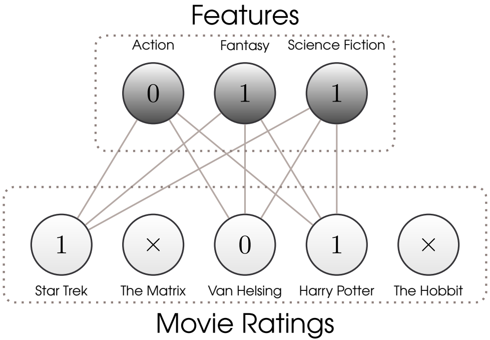
- abort Gibbs Sampling after $v^{(k)}$ and $h^{(k)}$ are computed
- approximate the expectation value

$$\mathbb{E}_{\vartheta}\left[\mathcal{V}\mathcal{H}^{\mathrm{T}}\right] \approx v^{(k)}{h^{(k)}}^{\mathrm{T}}$$

Learning: Example

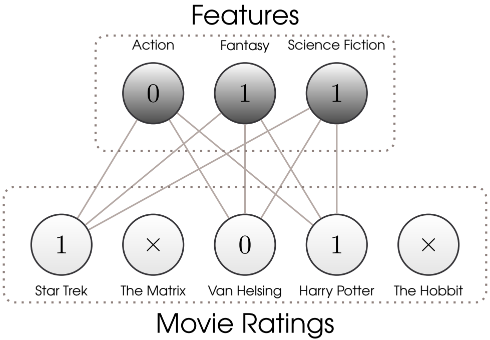


Learning: Example



- one RBM for every user with connections for rated movies

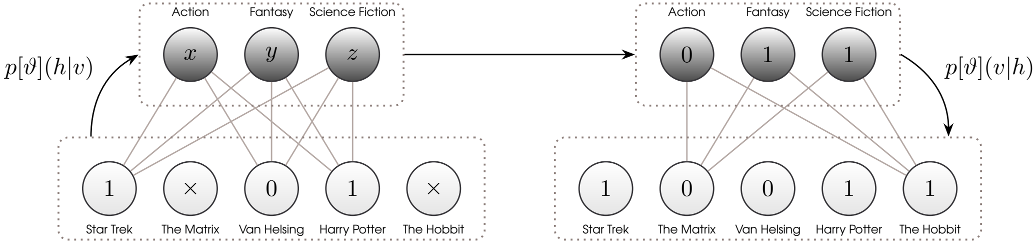
Learning: Example



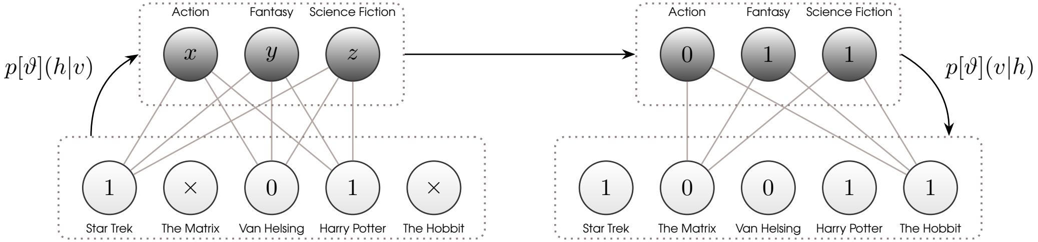
- one RBM for every user with connections for rated movies
- weights and biases off all RBM are tied together

Inference

Inference: Example

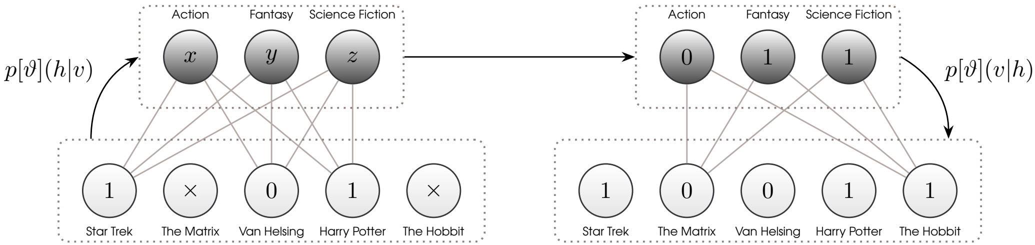


Inference: Example



- compute hidden values only for rated movies

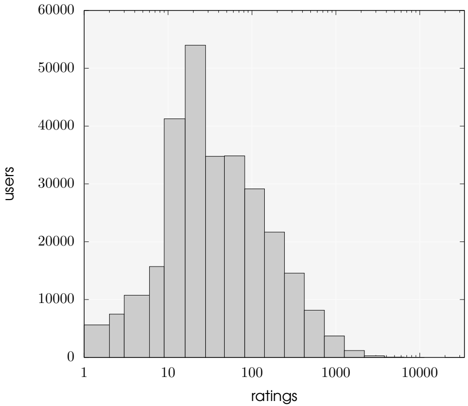
Inference: Example

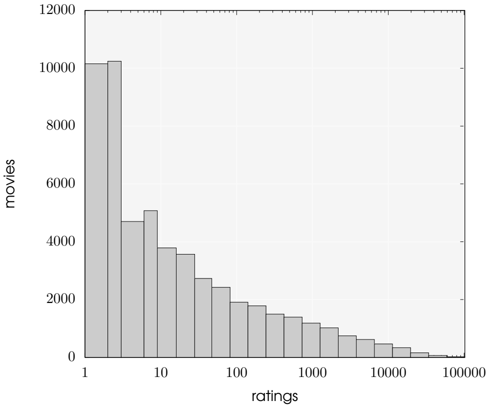


- compute hidden values only for rated movies
- compute visible values of unrated movies based on hidden values

Implementation

Implementation: MovieLens Dataset by GroupLens





- $\sim$ 27,000,000 ratings, $\sim$ 58,000 movies, $\sim$ 280,000 users

Implementation: Rating Database

	Star Trek	The Matrix	Van Helsing	Harry Potter	The Hobbit
James T. Kirk	1	1	$\times$	0	$\times$
Trinity	$\times$	1	0	1	1
Anna Valerious	$\times$	$\times$	1	$\times$	0
Severus Snape	0	1	0	1	0
Thorin Oakenshield	1	1	1	$\times$	0

Implementation: Rating Database

	Star Trek	The Matrix	Van Helsing	Harry Potter	The Hobbit
James T. Kirk	1	1	$\times$	0	$\times$
Trinity	$\times$	1	0	1	1
Anna Valerious	$\times$	$\times$	1	$\times$	0
Severus Snape	0	1	0	1	0
Thorin Oakenshield	1	1	1	$\times$	0

- implement as sparse matrix in CSR format

Implementation: Rating Database

	Star Trek	The Matrix	Van Helsing	Harry Potter	The Hobbit
James T. Kirk	1	1	$\times$	0	$\times$
Trinity	$\times$	1	0	1	1
Anna Valerious	$\times$	$\times$	1	$\times$	0
Severus Snape	0	1	0	1	0
Thorin Oakenshield	1	1	1	$\times$	0

- implement as sparse matrix in CSR format
- entries should be floating point

Implementation: Training and Testing

- same users in training and testing dataset

Implementation: Training and Testing

- same users in training and testing dataset
- divide ratings of users into training and testing dataset

Implementation: Training and Testing

- same users in training and testing dataset
- divide ratings of users into training and testing dataset
- learn biases and weights for every user only for rated movies

Going Further

Going Further: Tweak the Learning

- Contrastive Divergence Variants
- Momentum
- Weight Decay
- Different types of units

References

(1) Fischer, Asja and Christian Igel: *An introduction to restricted boltzmann machines*. LNCS, 7441:14–36, 2012.

(2) GroupLens: *Movielens dataset*, 2018. https://grouplens.org/datasets/movielens/latest/, visited on 2019-01-21.

(3) Harper, F. Maxwell und Joseph A. Konstan: *The MovieLens Datasets: History and Context*. ACM Trans. Interact. Intell. Syst., 5(4):19:1–19:19, Dezember 2015, ISSN 2160-6455. http://doi.acm.org/10.1145/2827872.

(4) Hinton, Geoffrey: *A practical guide to training restricted boltzmann machines: Version 1*. 2010. https://www.cs.toronto.edu/~hinton/absps/guideTR.pdf.

(5) Montúfar, Guido: *Restricted boltzmann machines: Introduction and review*. CoRR, abs/1806.07066, 2018. http://arxiv.org/abs/1806.07066.

(6) Murphy, Kevin P.: *Machine Learning: A Probabilistic Perspective*. MIT Press, 2012, ISBN 978-0-262-01802-9.

(7) Netflix: *Netflix prize*, 2009. https://www.netflixprize.com/index.html, visited on 2019-01-21.

(8) Netflix: *Netflix prize dataset*, 2009. https://archive.org/details/nf_prize_dataset.tar, visited on 2019-01-21.

(9) Netflix: *Netflix logo*, 2018. https://mms.businesswire.com/media/20150827005946/en/482959/5/etflix-Logo.jpg?download=1, visited on 2019-01-21.

(10) Oppermann, Artem: *Deep learning meets physics: Restricted boltzmann machines part i*, 2018. https://towardsdatascience.com/deep-learning-meets-physics-restricted-boltzmann-machines-part-i-6df5c4918c15, visited on 2019-01-22.

(11) Salakhutdinov, Ruslan, Andriy Mnih, and Geoffrey Hinton: *Restricted boltzmann machines for collaborative filtering*. Proceedings of the 24th international conference on Machine learning, pages 791–798, 2007.