

# NETFLIX

# Restricted Boltzmann Machines

Markus Pawellek

January 22, 2019

# Outline

Problem

The Model

Learning

Inference

Implementation

Results

Going Further

# Problem

# Problem: Collaborative Filtering – Movie Ratings

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

# Problem: Collaborative Filtering – Movie Ratings

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

Goal:

# Problem: Collaborative Filtering – Movie Ratings

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

Goal:

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



# Problem: Collaborative Filtering – Movie Ratings

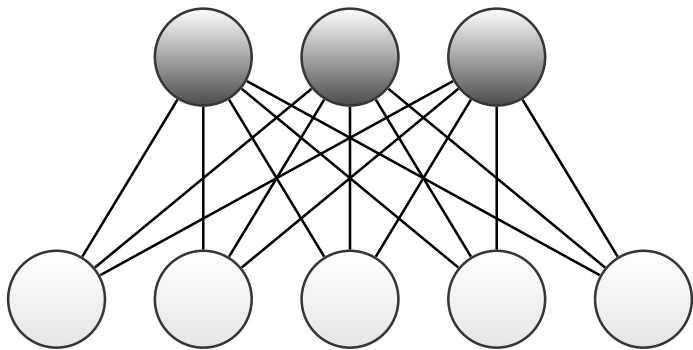
	Star Trek	The Matrix	Van Helsing	Harry Potter	The Hobbit
James T. Kirk	1	1	×	0	×
Trinity	×	1	0	1	1
Anna Valerious	×	×	1	×	0
Severus Snape	0	1	0	1	0
Thorin Oakenshield	1	1	1	×	0

Goal:

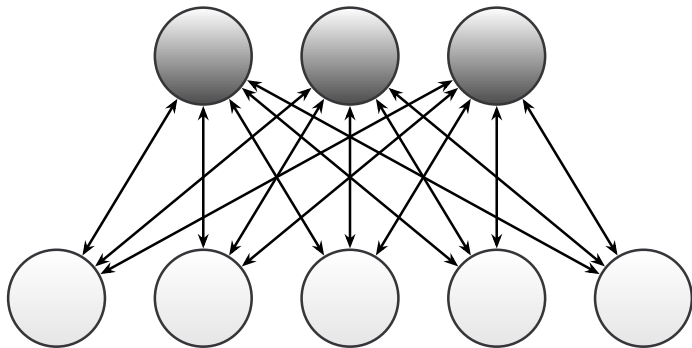
- ▶ approximately represent a complex probability distribution
- ▶ learn probability distribution based on given samples
- ▶ make predictions based on learned parameters

## The Model

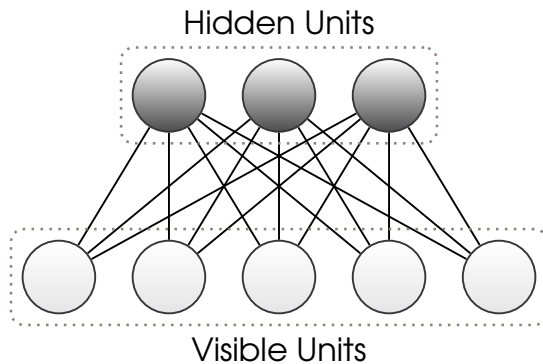
# The Model: Idea



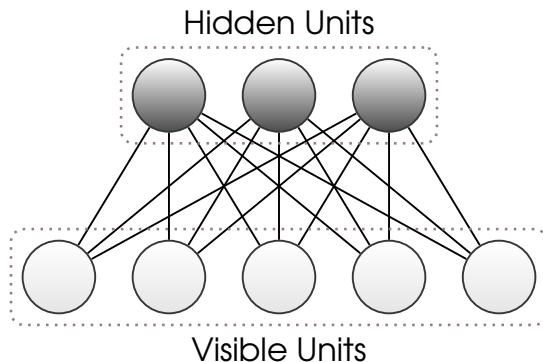
# The Model: Idea



# The Model: Idea

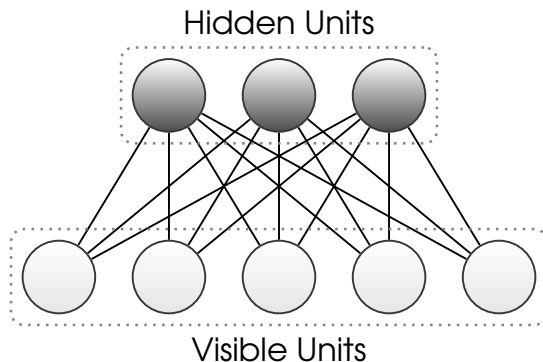


# The Model: Idea



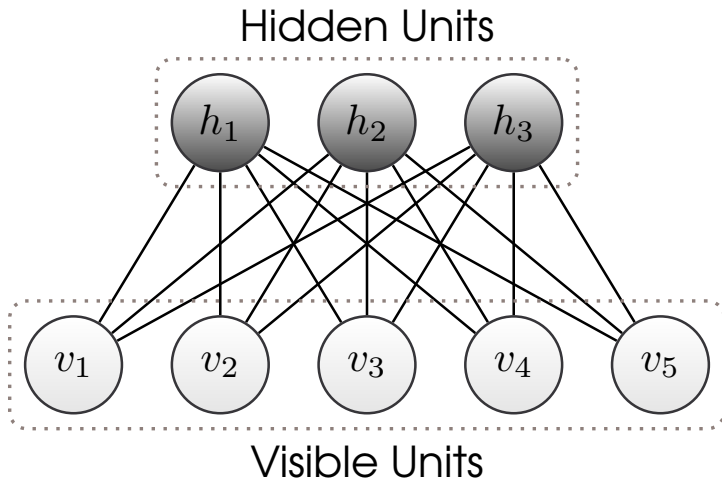
- units are divided into two subsets

# The Model: Idea



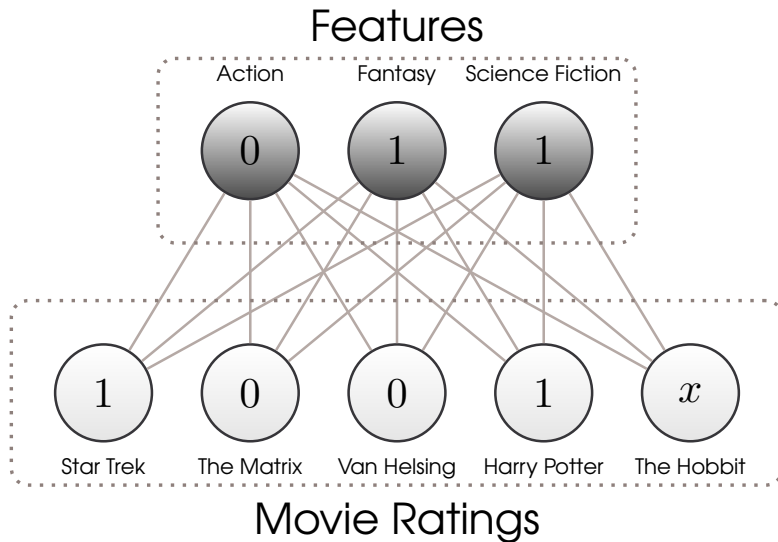
- ▶ units are divided into two subsets
- ▶ only connections between hidden and visible units are allowed

## The Model: Idea – Inputs

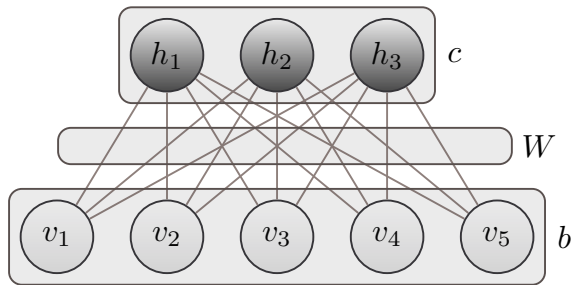




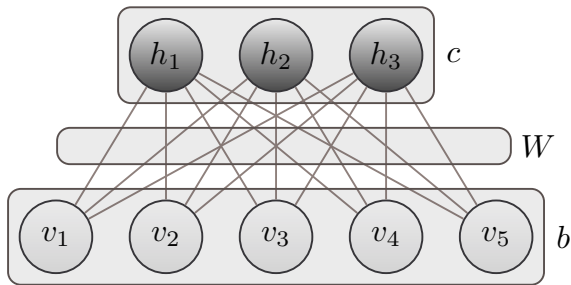
# The Model: Idea – Example



# The Model: Parameters



# The Model: Parameters



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

# The Model: Probability Distribution and Energy

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

# The Model: Probability Distribution and Energy

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

$$E[\vartheta]: V \times H \rightarrow \mathbb{R} \quad E[\vartheta](v, h) := -v^T W h - v^T b - h^T c$$

# The Model: Probability Distribution and Energy

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

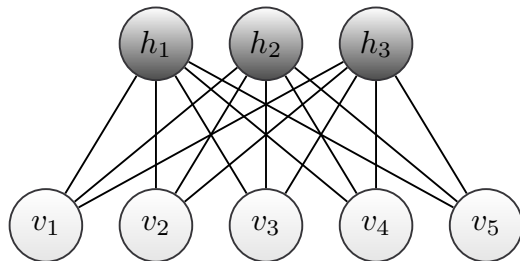
$$E[\vartheta]: V \times H \rightarrow \mathbb{R} \quad E[\vartheta](v, h) := -v^T W h - v^T b - h^T c$$

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

# The Model: Probability Distribution for Visible Units

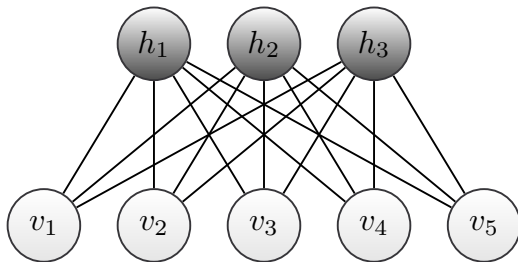
$$p[\vartheta]: V \rightarrow [0, 1] \quad p[\vartheta](v) := \sum_{h \in H} p[\vartheta](v, h)$$

# The Model: Posterior Probability





# The Model: Posterior Probability



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

# Learning

# Learning: Maximum Likelihood Estimation

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

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

# Learning: Gradient Ascent

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

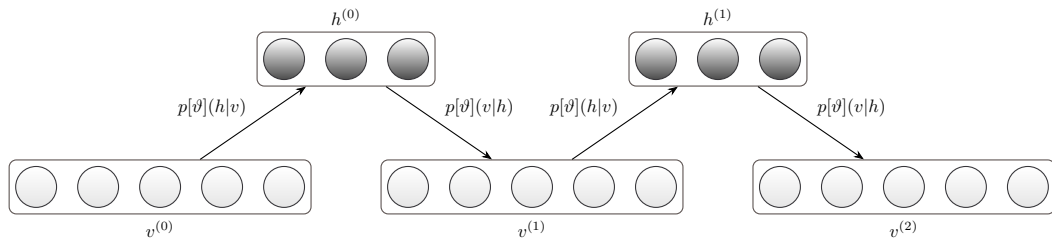
- ▶ use stochastic gradient ascent with minibatches

# Learning: Gradient Ascent

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

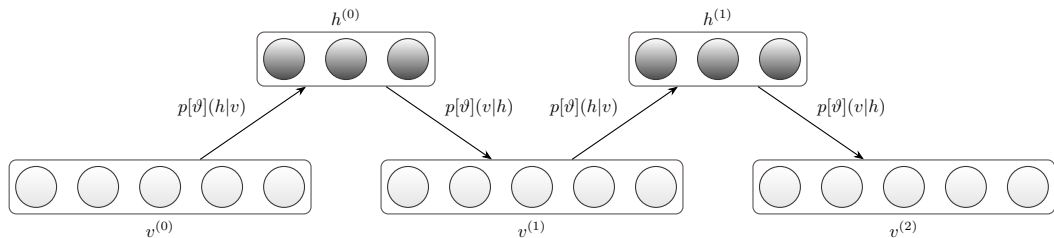
- ▶ use stochastic gradient ascent with minibatches
- ▶ evaluating the gradient introduces problems

# Learning: Gibbs Sampling



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

# Learning: Gibbs Sampling



- ▶ to estimate  $\mathbb{E}_{\vartheta} [\mathcal{V}\mathcal{H}^T]$  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

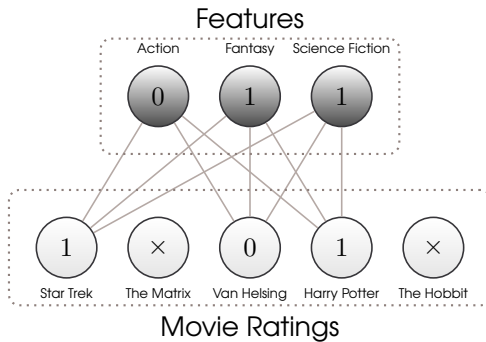
- ▶ abort Gibbs Sampling after  $v^{(k)}$  and  $h^{(k)}$  are computed
- ▶ approximate the expectation value

# Learning: Contrastive Divergence

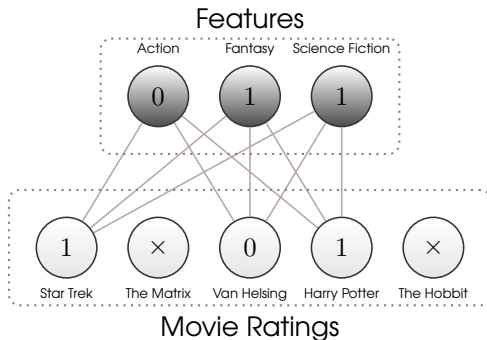
- ▶ abort Gibbs Sampling after  $v^{(k)}$  and  $h^{(k)}$  are computed
- ▶ approximate the expectation value

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

# Learning: Example

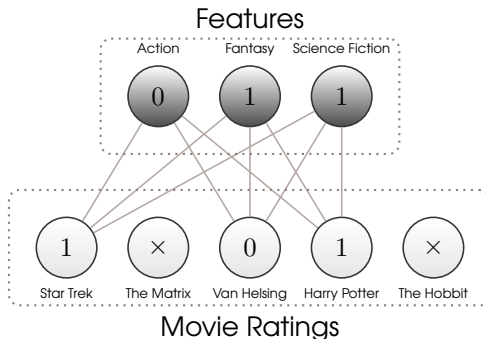


# Learning: Example



- ▶ one RBM for every user with connections for rated movies

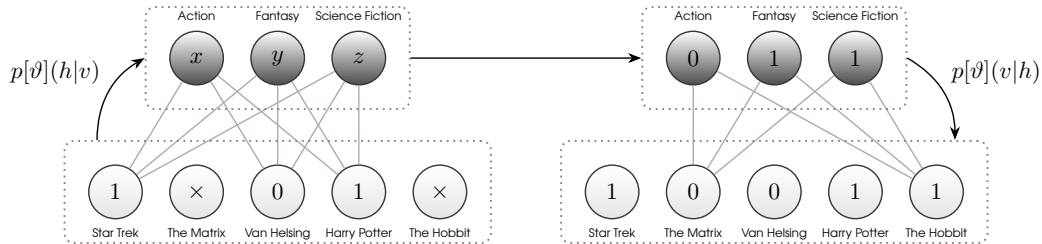
# Learning: Example



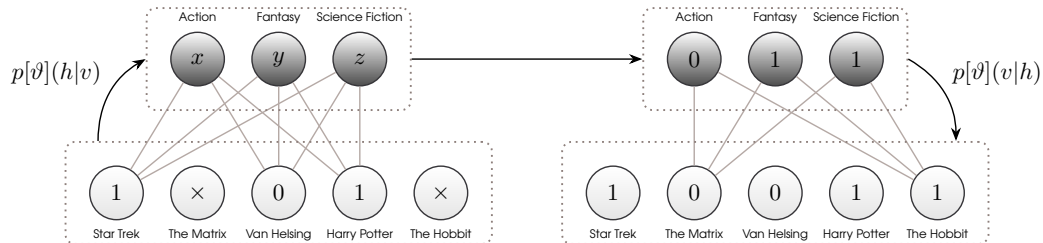
- ▶ one RBM for every user with connections for rated movies
- ▶ weights and biases of all RBM are tied together

# Inference

# Inference: Example



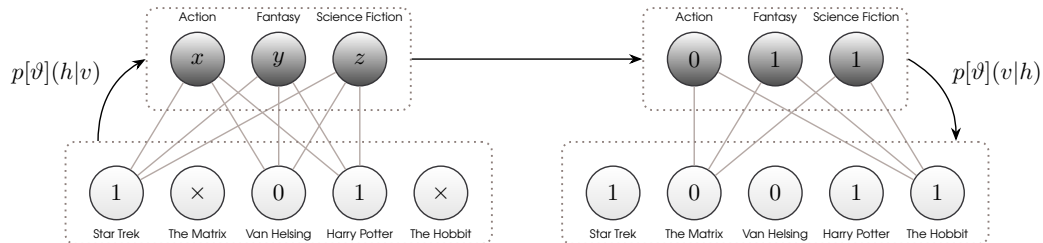
# Inference: Example



- 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

## Results

# Results

- ▶ RBMs are a powerful and versatile tool in machine learning

## Going Further

# Going Further: Tweak the Learning

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

# Going Further: Applications

- ▶ language modeling and document retrieval
- ▶ classification
- ▶ reducing dimensionality of data



# 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) Hinton, Geoffrey: *A practical guide to training restricted boltzmann machines: Version 1*. 2010.  
<https://www.cs.toronto.edu/~hinton/absps/guideTR.pdf>.
- (4) Montúfar, Guido: *Restricted boltzmann machines: Introduction and review*. CoRR, abs/1806.07066, 2018.  
<http://arxiv.org/abs/1806.07066>.
- (5) Murphy, Kevin P: *Machine Learning: A Probabilistic Perspective*. MIT Press, 2012, ISBN 978-0-262-01802-9.
- (6) Netflix: *Netflix prize*, 2009.  
<https://www.netflixprize.com/index.html>, visited on 2019-01-21.
- (7) Netflix: *Netflix prize dataset*, 2009.  
[https://archive.org/details/nf\\_prize\\_dataset.tar](https://archive.org/details/nf_prize_dataset.tar), visited on 2019-01-21.
- (8) Netflix: *Netflix logo*, 2018.  
<https://mms.businesswire.com/media/20150827005946/en/482959/5/etflix-Logo.jpg?download=1>, visited on 2019-01-21.
- (9) 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.
- (10) 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.