CF-NADE:
A Neural Autoregressive Approach to Collaborative Filtering

Yin Zheng, Bangsheng Tang, Wenkui Ding, Hanning Zhou

Hulu LLC.

Beijing, China

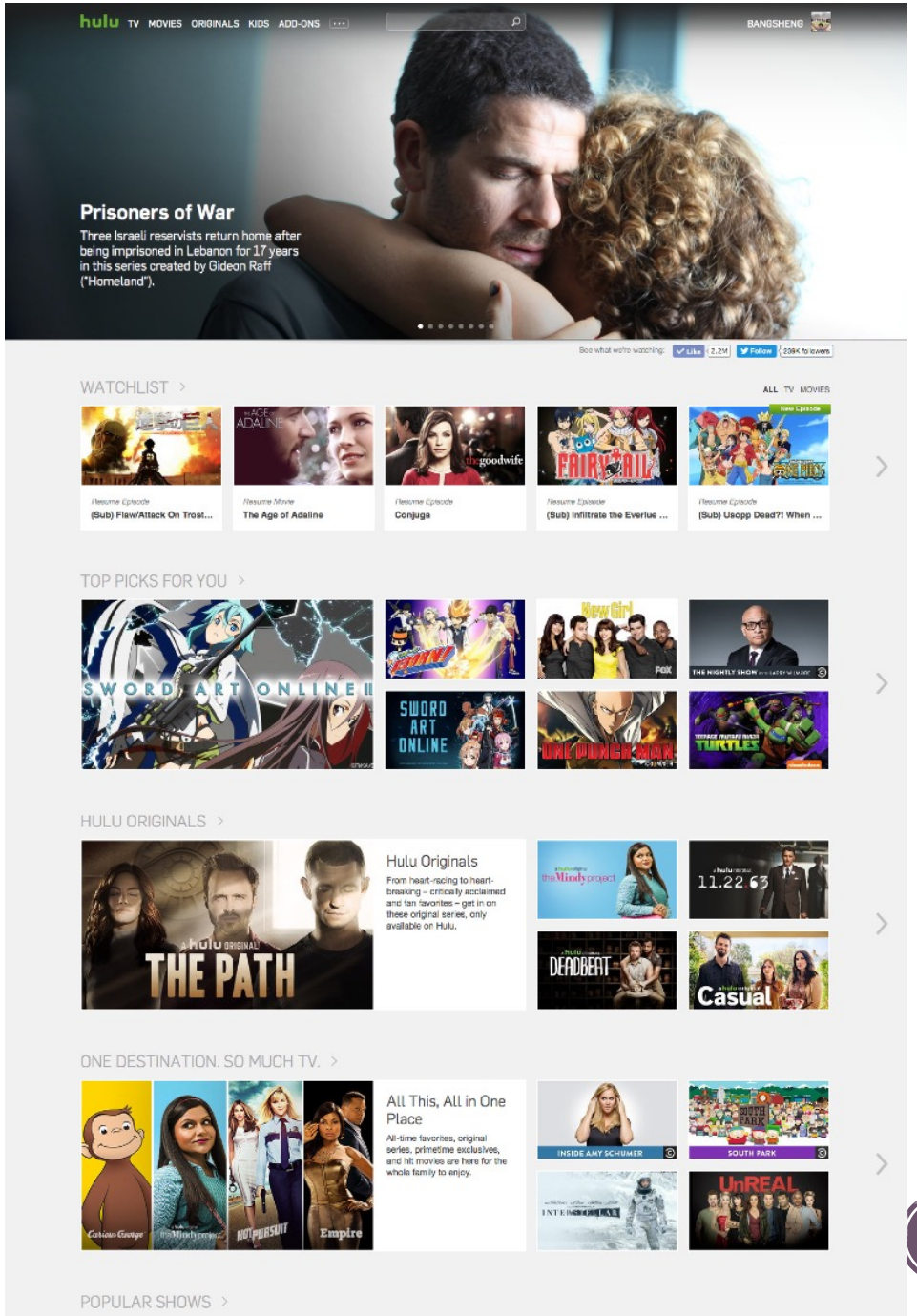
https://sites.google.com/site/zhengyin1126/

Outline

- Motivation
- Related Work
- CF-NADE
  - The basic model
  - Sharing parameters
  - Scalability
- Ordinal Cost in Collaborative Filtering
- Deep CF-NADE
- Experimental Results

Motivation

- Collaborative Filtering (CF):
  - Predict users' preferences
    - Previous behavior
    - Similar users
    - One's preferences do not change enormously
- Core of Recommender Systems
  - Hulu:
    - Personalized Homepage
    - Personalized Masthead
    - Watch list
    - Top picks for you
    - Related Shows
    - …

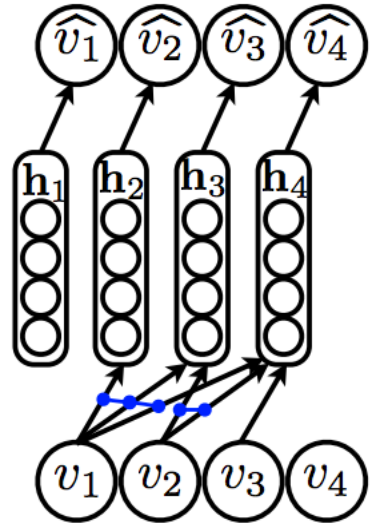


Related Work: CF methods

- Matrix Factorization based CF
  - SVD, SVD++
  - BiasMF (Koren et al., 2009)
  - PMF(Mnih & Salakhutdinov, 2007), BPMF(Salakhutdinov & Mnih,2008)
  - Poisson MF (Gopalan et al., 2013,2014)
  - LLORMA (Lee et al., 2015)
  - …
- Neural Network based CF
  - RBM-CF (Salakhutdinov, Mnih and Hinton, 2007)
  - AutoRec (Sedhain et al., 2015)

Related Work: NADE based Models

- NADE (Larochelle and Murray 2011):
  - A tractable distribution estimator
  - For Fixed Length Binary Vectors
  - Tied parameters for networks
  - O(DH)
  - Practical for High dimensional data



NADE model
Source: Larochelle and Murray 2011

NADE
Larochelle and Murray, AISTATS 2011

Real NADE
Uria, Murray, Larochelle, NIPS 2013

DocNADE
Larochelle and Lauly NIPS 2012

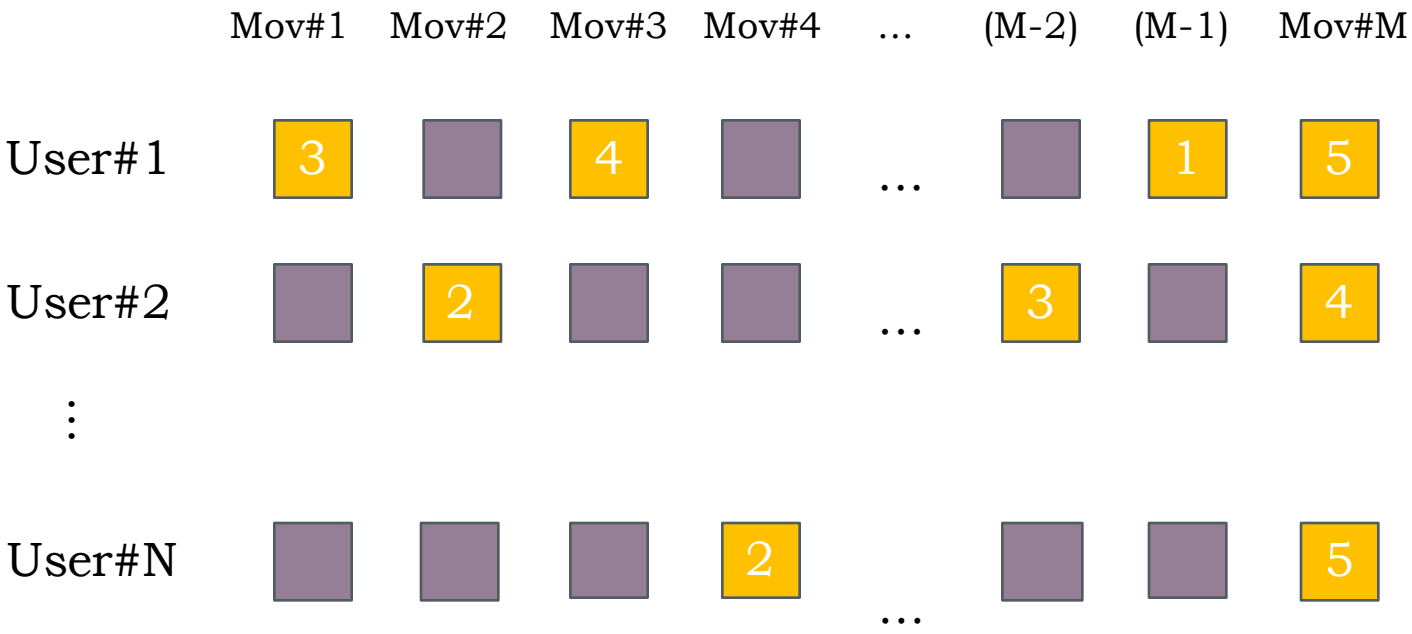
Fixation NADE
Zheng, Zemel, Zhang Larochelle, IJCV 2014

Deep NADE
Uria, Murray, Larochelle, ICML 2014

Supervised DocNADE
Zheng, Zhang and Larochelle, CVPR,2014

Deep DocNADE
Zheng, Zhang, Larochelle, tPAMI, 2015

CF-NADE: The basic model



Unrated Movie

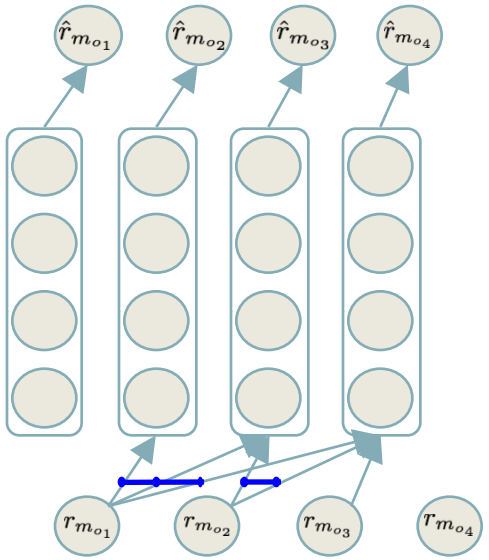
r Rated Movie with rating equals to r in {1,2,3,4,5}

For User u we denote it as $\mathbf{r}^u = (r^u_{m_{o_1}}, r^u_{m_{o_2}}, \ldots, r^u_{m_{o_D}})$

CF-NADE tries to model $p(\mathbf{r}) = \prod_{i=1}^{D} p\left(r_{m_{o_i}} | \mathbf{r}_{m_{o<i}}\right)$

CF-NADE: The basic model

$$p(\mathbf{r}) = \prod_{i=1}^{D} p\left(r_{m_{o_i}} | \mathbf{r}_{m_{o<i}}\right)$$



One Option for the conditionals

$$p\left(r_{m_{o_i}} = k | \mathbf{r}_{m_{o<i}}\right) = \frac{\exp\left(s^k_{m_{o_i}}\left(\mathbf{r}_{m_{o<i}}\right)\right)}{\sum_{k'=1}^{K} \exp\left(s^{k'}_{m_{o_i}}\left(\mathbf{r}_{m_{o<i}}\right)\right)}$$

$$s^k_{m_{o_i}}\left(\mathbf{r}_{m_{o<i}}\right) = b^k_{m_{o_i}} + \mathbf{V}^k_{m_{o_i},:}\mathbf{h}\left(\mathbf{r}_{m_{o<i}}\right)$$

$$\mathbf{h}\left(\mathbf{r}_{m_{o<i}}\right) = \mathbf{g}\left(\mathbf{c} + \sum_{j<i} \mathbf{W}^{r_{m_{o_j}}}_{:,m_{o_j}}\right)$$

About the order of the ratings

1. Should be predefined, e.g. *Timestamps*
2. A random order works well in practice
   - Different order is an different instantiation of CF-NADE for the same user
   - Key to extend CF-NADE to a deep model

CF-NADE: The basic model

Training Objective

$$-\log p(\mathbf{r}) = -\sum_{i=1}^{D} \log p\left(r_{m_{o_i}} | \mathbf{r}_{m_{o<i}}\right)$$

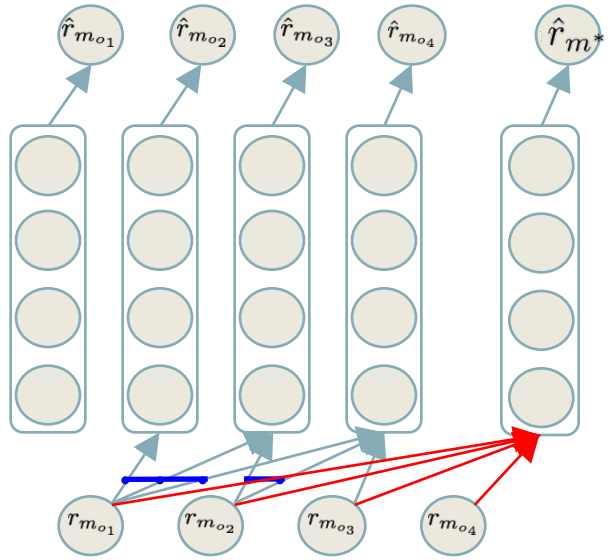
Testing Phase

$$\hat{r}_{m*} = \mathbb{E}_{p(r_{m*}=k|\mathbf{r})}[k]$$

where

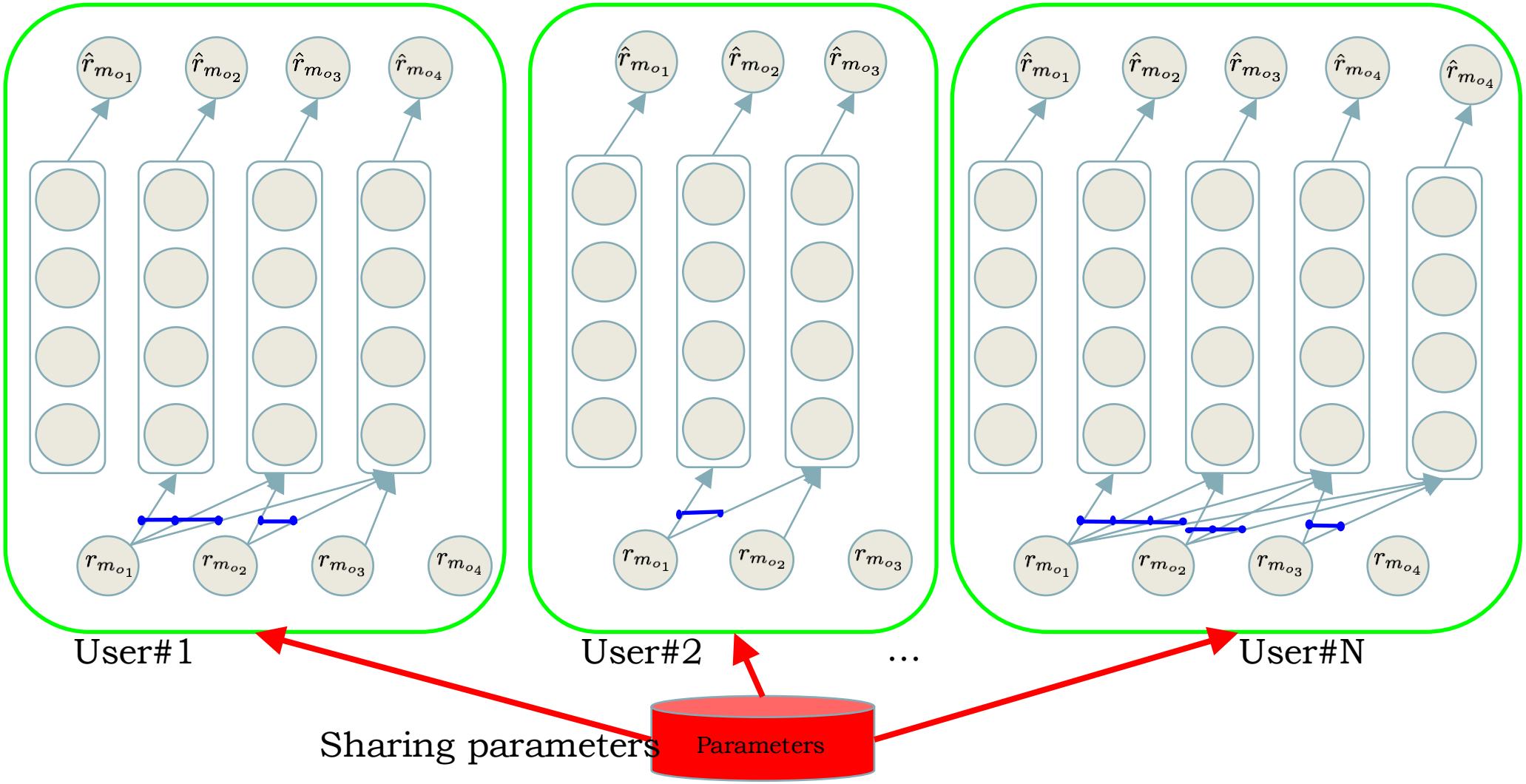
$$s^k_{m*}(\mathbf{r}) = b^k_{m*} + \mathbf{V}^k_{m*,:}\mathbf{h}(\mathbf{r})$$

$$\mathbf{h}(\mathbf{r}) = \mathbf{g}\left(\mathbf{c} + \sum_{j=1}^{D} \mathbf{W}^{r_{m_{o_j}}}_{:,m_{o_j}}\right)$$



A CF-NADE for each user

hulu



Sharing Parameters

hulu

The basic CF-NADE model

- different parameters for different ratings
- More parameters, less Optimization

$$\mathbf{h}\left(\mathbf{r}_{m_{o<i}}\right) = \mathbf{g}\left(\mathbf{c} + \sum_{j<i} \mathbf{W}^{r_{m_{oj}}}_{:,m_{oj}}\right)$$

$$s^k_{m_{oi}}\left(\mathbf{r}_{m_{o<i}}\right) = b^k_{m_{oi}} + \mathbf{V}^k_{m_{oi},:}\mathbf{h}\left(\mathbf{r}_{m_{o<i}}\right)$$

Sharing Parameters between Ratings

$$\mathbf{h}\left(\mathbf{r}_{m_{o<i}}\right) = \mathbf{g}\left(\mathbf{c} + \sum_{j<i} \mathbf{W}^{r_{m_{oj}}}_{:,m_{oj}}\right)$$

Sharing

$$\mathbf{h}\left(\mathbf{r}_{m_{o<i}}\right) = \mathbf{g}\left(\mathbf{c} + \sum_{j<i}\sum_{k=1}^{r_{m_{oj}}} \mathbf{W}^{k}_{:,m_{oj}}\right)$$

$$s^k_{m_{oi}}\left(\mathbf{r}_{m_{o<i}}\right) = b^k_{m_{oi}} + \mathbf{V}^k_{m_{oi},:}\mathbf{h}\left(\mathbf{r}_{m_{o<i}}\right)$$

Sharing

$$s^k_{m_{oi}}\left(\mathbf{r}_{m_{o<i}}\right) = \sum_{j\le k}\left(b^j_{m_{oi}} + \mathbf{V}^j_{m_{oi},:}\mathbf{h}\left(\mathbf{r}_{m_{o<i}}\right)\right)$$

Better Scalability

hulu

Too many parameters... $\mathbf{W}^k \in \mathbb{R}^{H \times M} \quad \mathbf{V}^k \in \mathbb{R}^{M \times H} \quad k \in \{1, 2, \ldots, K\}$

Netflix dataset, M=17700, H=500, K=5, then **89** million free parameters

Weight Decay and Dropout is OK, but...

Solution: Factorize W, V by a product of 2 low rank matrices

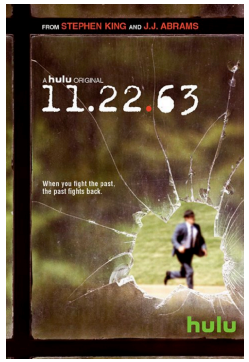
$$W^k_{i,m} = \sum_{j=1}^{J} B_{i,j} A^k_{j,m} \qquad V^k_{m,i} = \sum_{j=1}^{J} P^k_{m,j} Q_{j,i} \qquad J \ll H \quad J \ll M$$

e.g. J=50, M=17700, H=500, K=5, then only **9** million free parameters for Netflix dataset

Ordinal Cost for CF

Ordinal Nature of Rating data:

I give this show 4-star score!



$\mathcal{C}_{\text{reg}}$ **Regular Cost: Softmax**

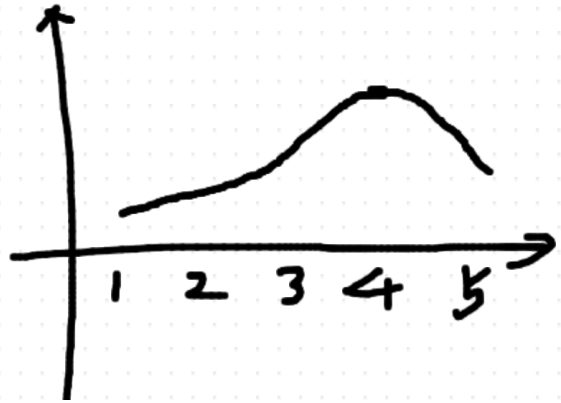
Score of rating k is the biggest

Rankings of Ratings

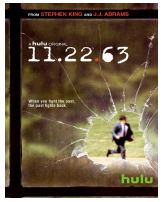
$$\mathbf{y}^{\text{down}} = (k, k-1, \ldots, 1)$$
$$\mathbf{y}^{\text{up}} = (k, k+1, \ldots, K)$$

$\mathcal{C}_{\text{ord}}$ **Ordinal Cost: Ranking loss**

$$k \succ k-1 \succ \ldots \succ 1$$
$$k \succ k+1 \succ \ldots \succ K$$



Ordinal Cost for CF



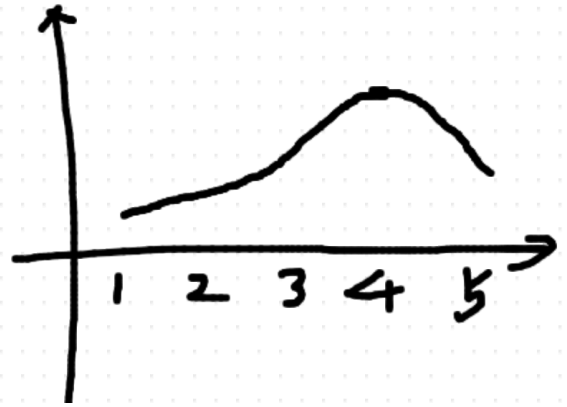
hulu

Ordinal Cost

$$p\left(r_{m_{o_i}} = k | \mathbf{r}_{m_{o<i}}\right) = \prod_{j=k}^{1} \frac{\exp(s^j_{m_{o_i}})}{\sum_{t=1}^{j} \exp(s^t_{m_{o_i}})} \prod_{j=k}^{K} \frac{\exp(s^j_{m_{o_i}})}{\sum_{t=j}^{K} \exp(s^t_{m_{o_i}})}$$

CF-NADE → LTR →

$$\begin{aligned} \mathbf{y}^{\text{down}} &= (k, k-1, \ldots, 1) \\ \mathbf{y}^{\text{up}} &= (k, k+1, \ldots, K) \end{aligned}$$



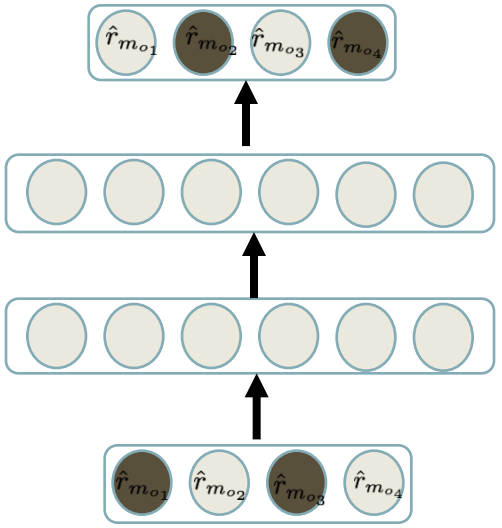
Hybrid Cost

$$\mathcal{C}_{\text{hybrid}} = (1-\lambda)\mathcal{C}_{\text{reg}} + \lambda\mathcal{C}_{\text{ord}}$$

Fen Xia, et al. Listwise approach to learning to rank: theory and algorithm. ICML 2008

Deep CF-NADE

About the orderings of rated items $-\log p(\mathbf{r}) = -\sum_{i=1}^{D} \log p\left(r_{m_{o_i}} | \mathbf{r}_{m_{o<i}}\right)$



Training over all possible orderings $\mathcal{C} = \mathop{\mathbb{E}}_{o \in \mathcal{O}} \sum_{i=1}^{D} -\log p\left(r_{m_{o_i}} | \mathbf{r}_{m_{o<i}}, o\right)$

- Given a context $\mathbf{r}_{m_{o<i}}$, CF-NADE be equally good at modeling $\mathbf{r}_{m_{o \geq i}}$

Cost Function is rewritten as $\mathcal{C} = \frac{D}{D-i+1} \sum_{j \geq i} -\log p\left(r_{m_{o_j}} | \mathbf{r}_{m_{o<i}}\right)$

- only relies on $\mathbf{r}_{m_{o<i}}$
- More hidden layers can be added as regular multiple layer neural networks

$$\mathbf{h}^{(l)}\left(\mathbf{r}_{m_{o<i}}\right) = \mathbf{g}\left(\mathbf{c}^{(l)} + \mathbf{W}^{(l)}\mathbf{h}^{(l-1)}\left(\mathbf{r}_{m_{o<i}}\right)\right)$$

Complexity: $O(K\hat{D}H + H^2L)$

Experiments

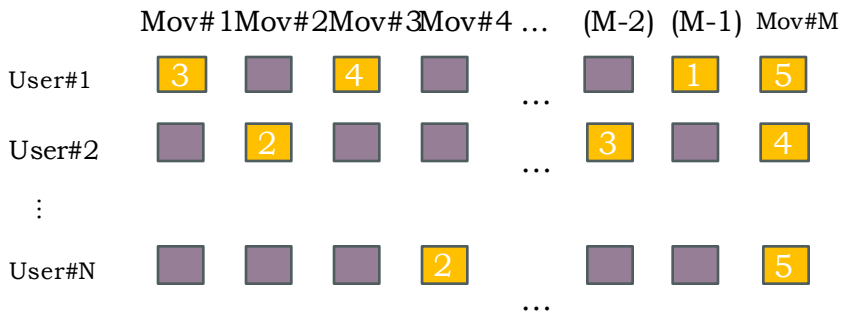
dataset	#Users	#Items	#Scales	#Ratings	type
MovieLen 1M	6040	3952	5	10^6	I-CF-NADE
MovieLen 10M	71567	10681	10	10^7	U-CF-NADE
Netflix	480189	17770	5	10^8	U-CF-NADE

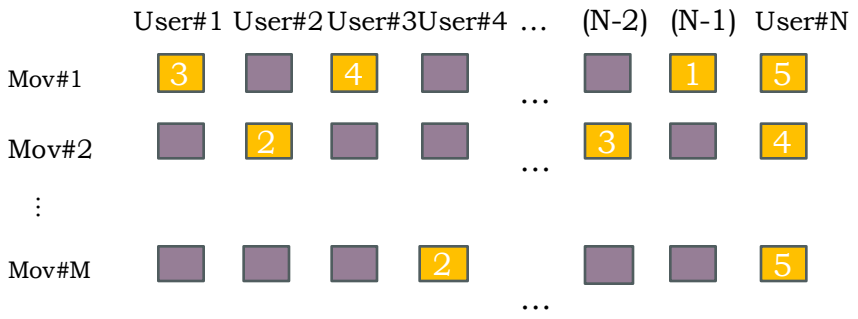
Train: 90% ratings
Test: 10% ratings

Metric

$$RMSE = \sqrt{\frac{\sum_{i=1}^{S}(r_i - \tilde{r}_i)^2}{S}}$$

User based CF-NADE (U-CF-NADE) VS Item based CF-NADE (I-CF-NADE)

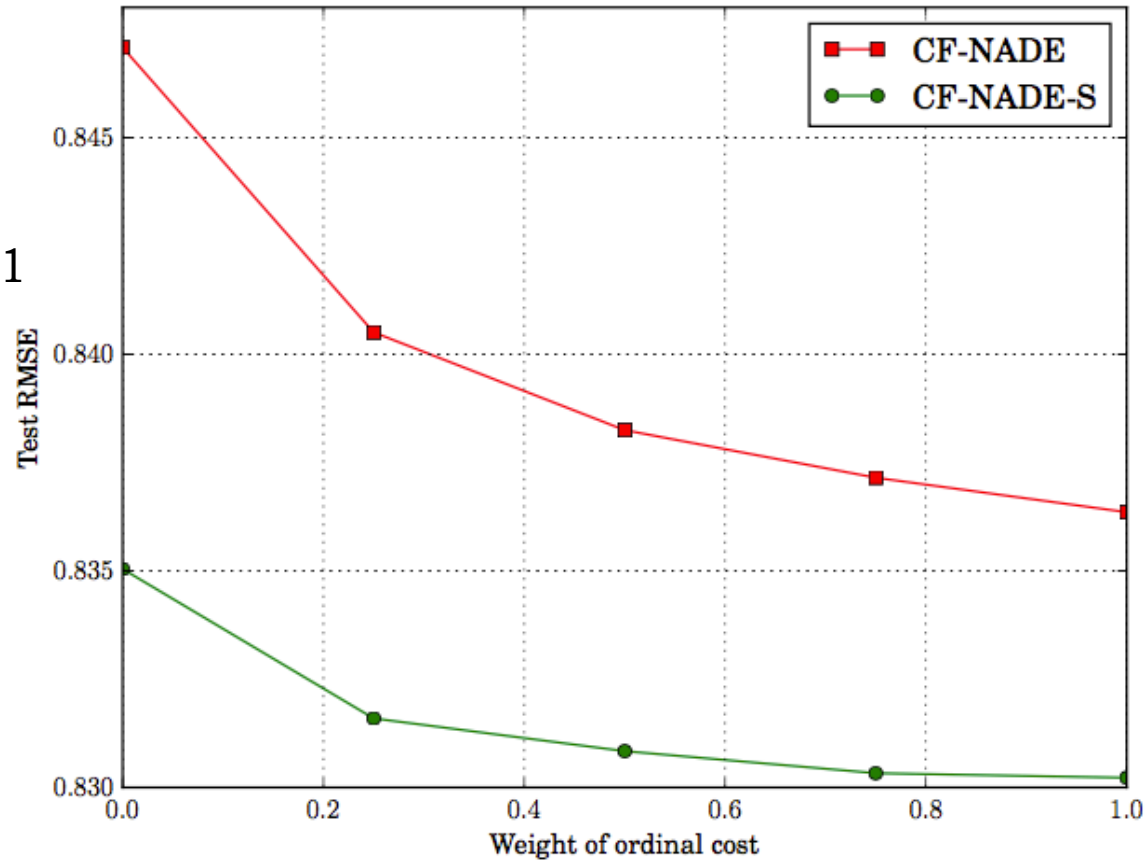




Experiments on MovieLens 1M

Impact of Ordinal Cost

- Hidden size = 500
- Weight decay = 0.015
- Adam Optimizer, learning rate 0.001



Experiments on MovieLens 1M

Compare with other baselines

- Hidden Size {250, 500, 750}
- Weight decay {0.015, 0.02}
- Learning rate {0.001, 0.0005, 0.0002}

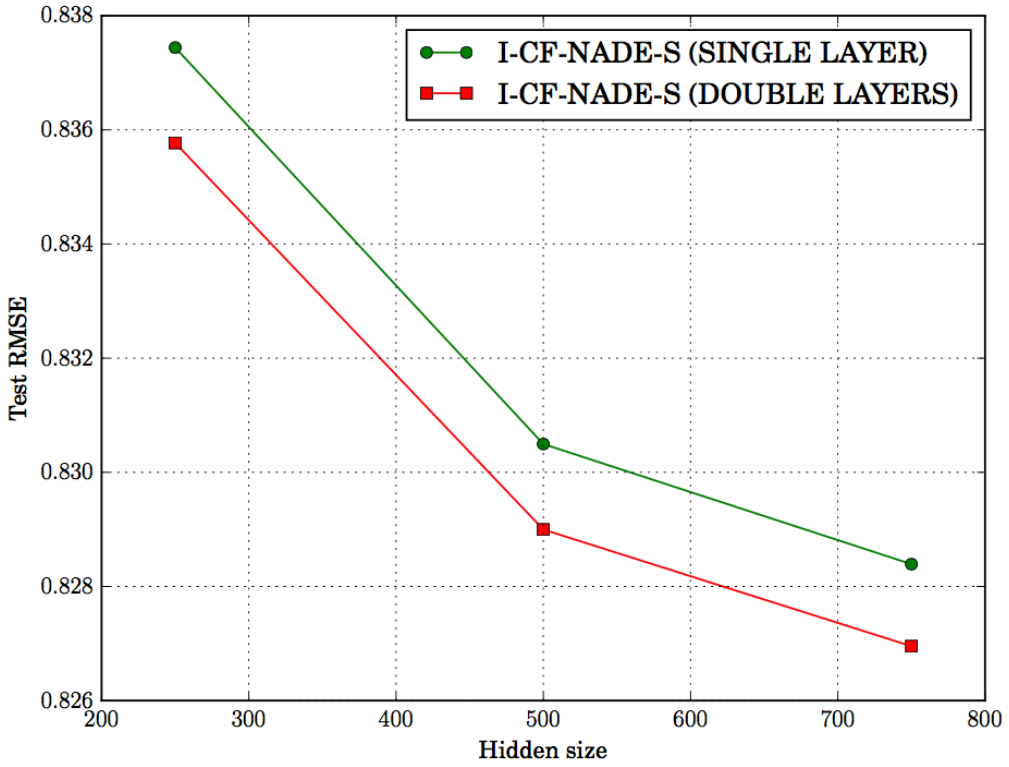


Table 1. Test RMSE of different models on MovieLens 1M.

METHOD	TEST RMSE
PMF†	0.883
U-RBM∗	0.881
U-AUTOREC (SEDHAIN ET AL., 2015)	0.874
LLORMA-GLOBAL (LEE ET AL., 2013)	0.865
I-RBM∗	0.854
BIASMF∗	0.845
NNMF (DZIUGAITE & ROY, 2015)	0.843
LLORMA-LOCAL (LEE ET AL., 2013)	0.833
I-AUTOREC (SEDHAIN ET AL., 2015)	0.831
U-CF-NADE-S (SINGLE LAYER)	0.850
U-CF-NADE-S (2 LAYERS)	0.845
I-CF-NADE-S (SINGLE LAYER)	0.830
I-CF-NADE-S (2 LAYERS)	**0.829**

†: Taken from (Dziugaite & Roy, 2015).

∗: Taken from (Sedhain et al., 2015).

Experiments on MovieLens 10M

hulu

Configurations

- Hidden Size 500
- Factorized U-CF-NADE, J=50
- Learning rate 0.0005
- Weight decay 0.015

Table 2. Test RMSE of different models on MovieLens 10M.

METHOD	TEST RMSE
U-AUTOREC (SEDHAIN ET AL., 2015)	0.867
I-RBM†	0.825
U-RBM†	0.823
LLORMA-GLOBAL (LEE ET AL., 2013)	0.822
BIASMF†	0.803
LLORMA-LOCAL (LEE ET AL., 2013)	0.782
I-AUTOREC (SEDHAIN ET AL., 2015)	0.782
U-CF-NADE-S (SINGLE LAYER)	0.772
U-CF-NADE-S (2 LAYERS)	**0.771**

†: Taken from (Sedhain et al., 2015).

Experiments on Netflix Dataset

hulu

Configurations

- Hidden Size 500
- Factorized U-CF-NADE, J=50
- Learning rate 0.0005
- Weight decay 0.001

Table 3. Test RMSE of different models on Netflix dataset.

METHODS	TEST RMSE
LLORMA-GLOBAL (LEE ET AL., 2013)	0.874
U-RBM†	0.845
BIASMF†	0.844
LLORMA-LOCAL (LEE ET AL., 2013)	0.834
I-AUTOREC (SEDHAIN ET AL., 2015)	0.823
U-CF-NADE-S (SINGLE LAYER)	0.804
U-CF-NADE-S (2 LAYERS)	**0.803**

†: Taken from (Sedhain et al., 2015).

Complexity of CF-NADE

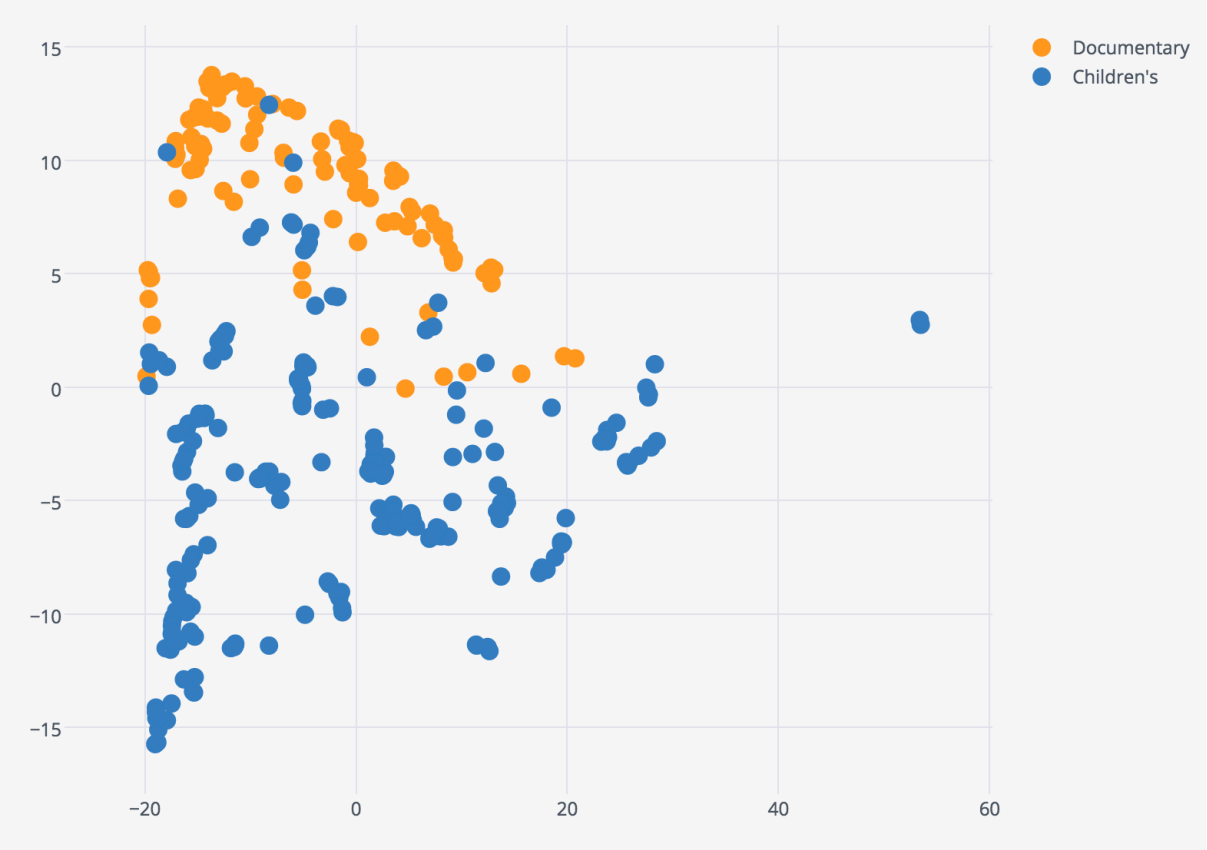
Table 4. Complexity of CF-NADE on different benchmarks

Datasets	# Layers	Train Time (second)	Test Time (second)	#Params (million)
MLens1M	1 layer	3.09	0.65	30.2
	2 layers	3.11	0.68	30.48
MLens10M	1 layer	134.76	31.72	10.78
	2 layers	135.62	32.73	10.98
Netflix	1 layer	1057.81	239.78	9.02
	2 layers	1064.33	243.79	9.19

Note that there are some rooms to speed up the implementation

The complexity is $O(K\hat{D}H + H^2L)$

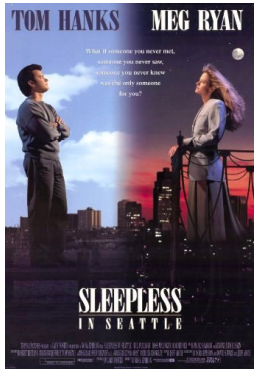
Visualization of the Learned Model



tSNE of the representations of movies on MovieLens 1M

Input Movies

Star Trek VI:
The Undiscovered Country

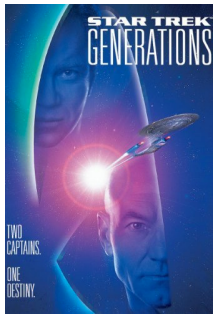


Sleepless in Seattle

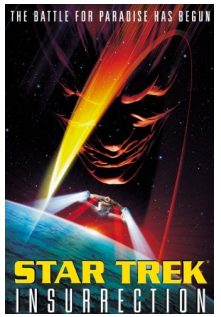
The Lion King

Top 6 most similar movies by CF-NADE on MovieLens 1M

Star Trek III:
The Search for Spock



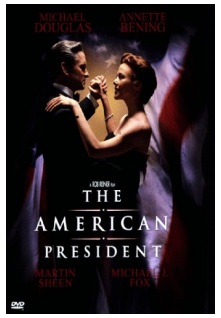
Star Trek:
Generations



Star Trek:
Insurrection

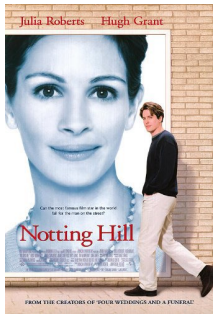
Star Trek:
First Contact

Star Trek IV:
The Voyage Home



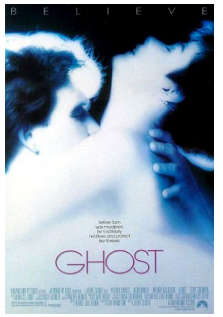
The American President

Pretty Woman

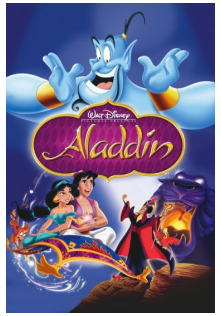


Notting Hill

A League of Their Own



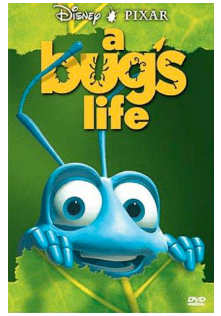
Ghost



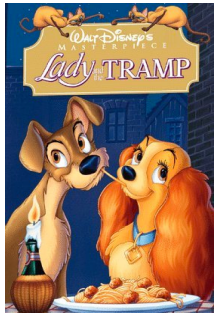
Aladdin

Beauty and
the Beast

The Little
Mermaid



A Bug's Life



Lady and
the Tramp

Conclusion

- CF-NADE:
  - An Efficient and Powerful architecture for CF tasks
  - Sharing parameters is a good way to improve the performance
  - Better scalability by factorization
  - Can be extended to a deep version
- Ordinal cost for rating data
  - Learning to rank
- Meaningful representations
- Source code at: https://github.com/Ian09/CF-NADE

Thanks! Q&A