Collaborative Deep Learning and Its Variants for Recommender Systems

Hao Wang

Joint work with Naiyan Wang, Xingjian Shi, and Dit-Yan Yeung

香 港 科 技 大 學
THE HONG KONG UNIVERSITY OF SCIENCE AND TECHNOLOGY

Recommender Systems

Rating matrix:

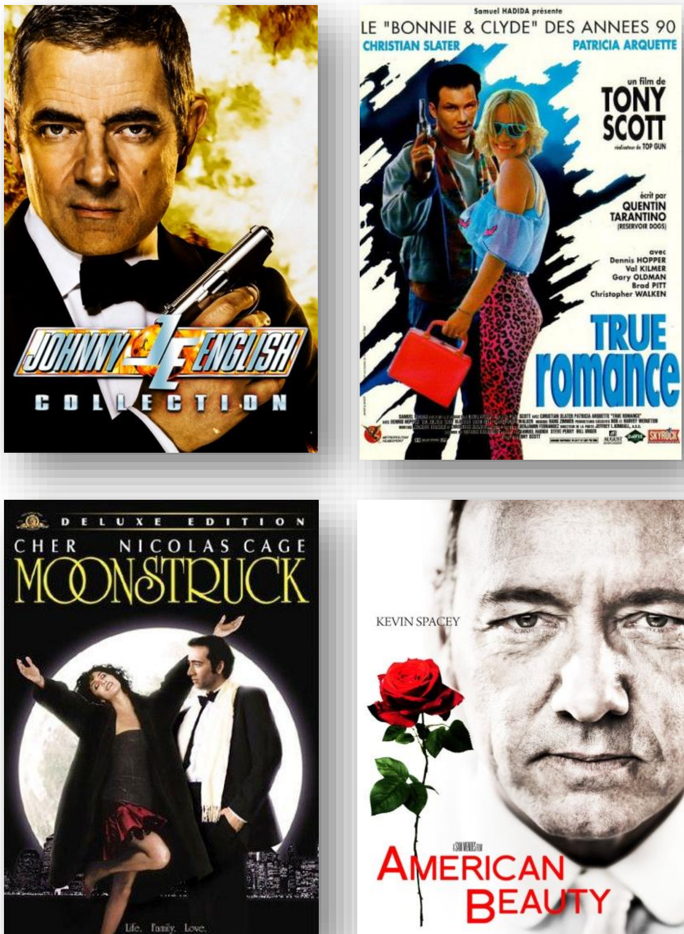
movie \ user	1	2	3	4	5
1	✓	?	?	?	?
2	✓	?	?	✓	?
3	?	?	✓	?	?
4	?	✓	?	?	✓
5	✓	?	?	?	?

Matrix completion

Observed preferences: ✓

To predict: ?

Recommender Systems with Content

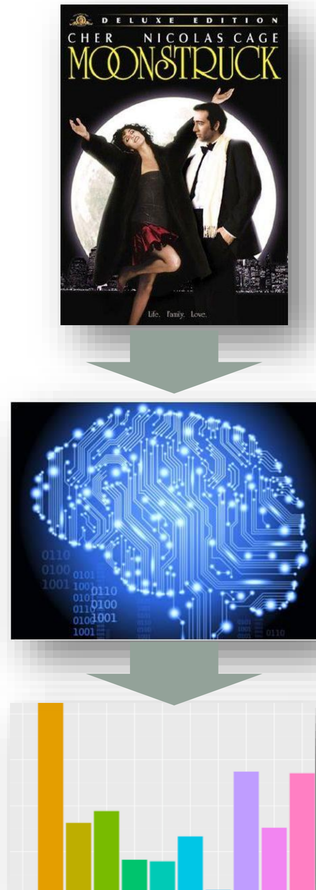


Content information:
Plots, directors, actors, etc.

movie \ user	1	2	3	4	5
1	✓	?	?	?	?
2	✓	?	?	✓	?
3	?	?	✓	?	?
4	?	✓	?	?	✓
5	✓	?	?	?	?

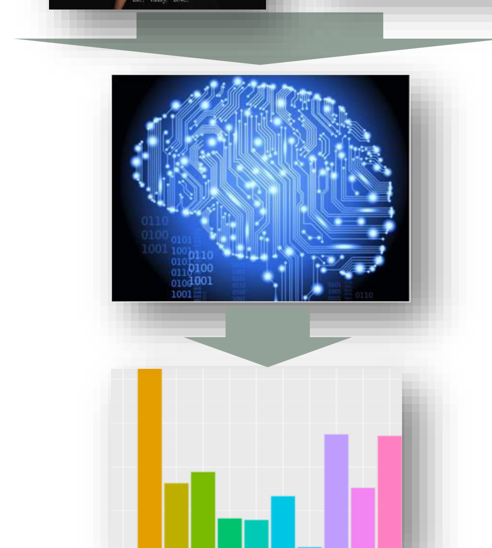
Modeling the Content Information

Handcrafted features



Automatically learn features

movie \ user	1	2	3	4	5
1	✓	?	?	?	?
2	✓	?	?	✓	?
3	?	?	✓	?	?
4	?	✓	?	?	✓
5	✓	?	?	?	?



Automatically learn features and adapt for ratings

Modeling the Content Information

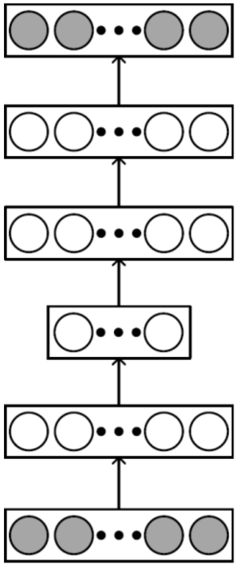
1. Powerful features for content information

Deep learning

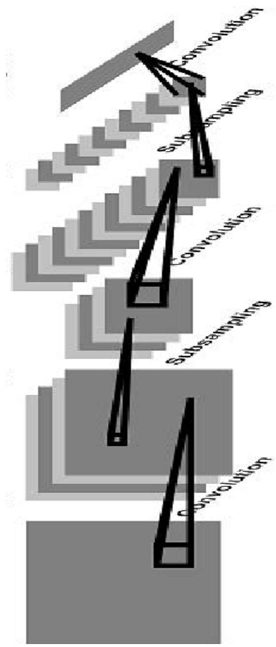
2. Feedback from rating information Non-i.i.d.

Collaborative deep learning

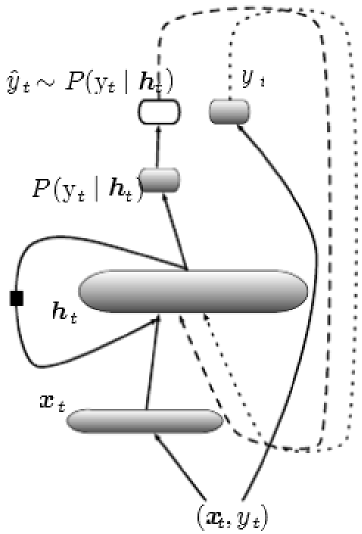
Deep Learning



Stacked denoising autoencoders



Convolutional neural networks



Recurrent neural networks

Typically for i.i.d. data

Modeling the Content Information

1. Powerful features for content information

Deep learning

2. Feedback from rating information → Non-i.i.d.

Collaborative deep learning (CDL)

Contribution

- **Collaborative deep learning:**

  *** deep learning for non-i.i.d. data**

  *** joint representation learning and collaborative filtering**

Contribution

- Collaborative deep learning
- Complex target:
  * beyond targets like classification and regression
  * to complete a low-rank matrix

Contribution

- Collaborative deep learning
- Complex target
- First hierarchical Bayesian models for deep hybrid recommender system

Related Work

- Not hybrid methods (ratings only)

  RBM (single layer, Salakhutdinov et al., 2007)

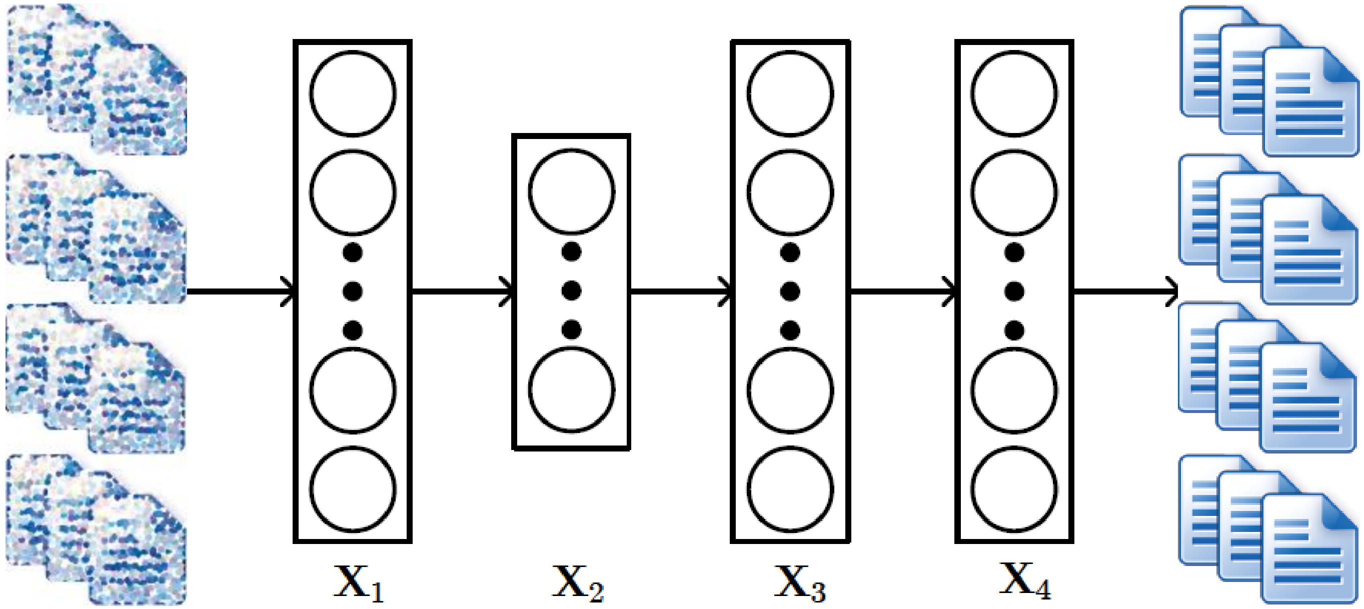
  I-RBM/U-RBM (Georgiev et al., 2013)

- Not using Bayesian modeling for joint learning

  DeepMusic (van den Oord et al., 2013)

  HLDBN (Wang et al., 2014)

Stacked Denoising Autoencoders (SDAE)



Corrupted input

Clean input

SDAE solves the following optimization problem:

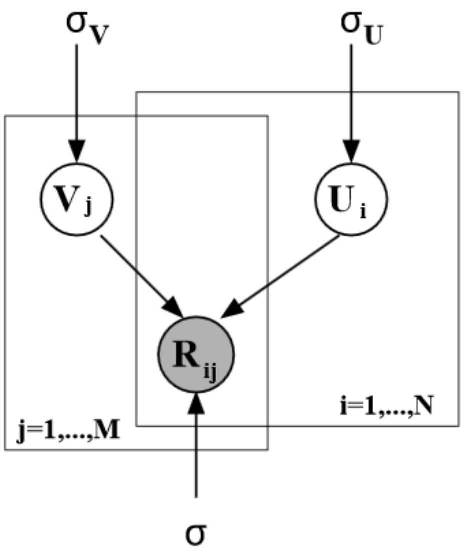
$$\min_{\{\mathbf{W}_l\},\{\mathbf{b}_l\}} \|\mathbf{X}_c - \mathbf{X}_L\|_F^2 + \lambda \sum_l \|\mathbf{W}_l\|_F^2,$$

where λ is a regularization parameter and $\|\cdot\|_F$ denotes the Frobenius norm.

[Vincent et al. 2010]

Probabilistic Matrix Factorization (PMF)

Graphical model:



Notation:

- V_j latent vector of item j
- U_i latent vector of user i
- R_{ij} rating of item j from user i

Generative process:

$$p(U|\sigma_U^2) = \prod_{i=1}^{N} \mathcal{N}(U_i|0, \sigma_U^2 \mathbf{I}) \qquad p(V|\sigma_V^2) = \prod_{j=1}^{M} \mathcal{N}(V_j|0, \sigma_V^2 \mathbf{I})$$

$$p(R|U, V, \sigma^2) = \prod_{i=1}^{N} \prod_{j=1}^{M} \left[\mathcal{N}(R_{ij}|U_i^T V_j, \sigma^2) \right]^{I_{ij}}$$

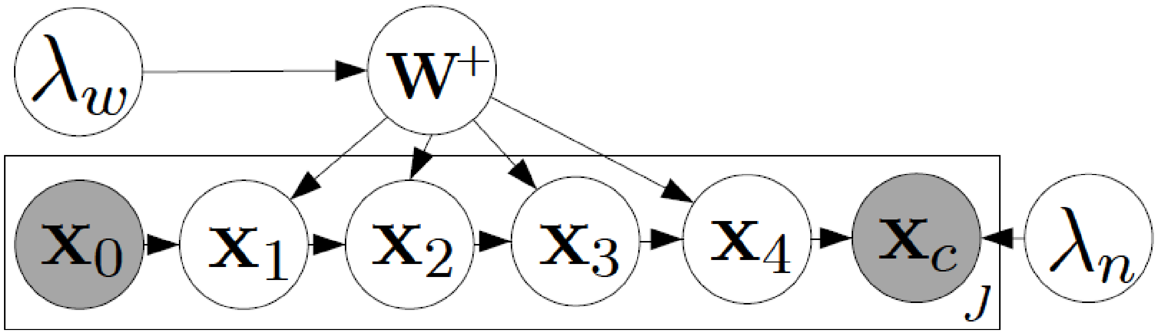
Objective function if using MAP:

$$E = \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{M} I_{ij} (R_{ij} - U_i^T V_j)^2 + \frac{\lambda_U}{2} \sum_{i=1}^{N} \| U_i \|_{Fro}^2 + \frac{\lambda_V}{2} \sum_{j=1}^{M} \| V_j \|_{Fro}^2$$

[Salakhutdinov et al. 2008]

Probabilistic SDAE

Graphical model:



Generative process:

$$\mathbf{W}_{l,*n} \sim \mathcal{N}(0, \lambda_w^{-1}\mathbf{I}_{K_l})$$

$$\mathbf{b}_l \sim \mathcal{N}(0, \lambda_w^{-1}\mathbf{I}_{K_l})$$

$$\mathbf{X}_{l,j*} \sim \mathcal{N}(\sigma(\mathbf{X}_{l-1,j*}\mathbf{W}_l + \mathbf{b}_l), \boxed{\lambda_s^{-1}\mathbf{I}_{K_l}})$$

$$\mathbf{X}_{c,j*} \sim \mathcal{N}(\mathbf{X}_{L,j*}, \lambda_n^{-1}\mathbf{I}_B)$$

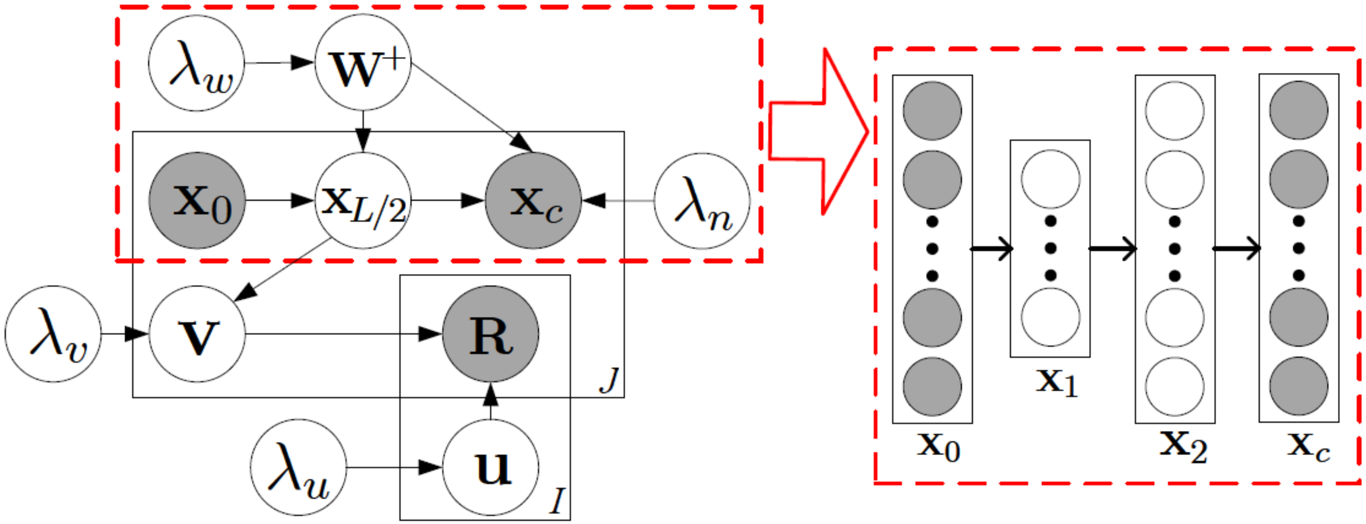
Generalized SDAE

Notation:

- $\mathbf{X}_0$ corrupted input
- $\mathbf{X}_c$ clean input
- $\mathbf{W}^+$ weights and biases

Collaborative Deep Learning (CDL)

Graphical model:



Collaborative deep learning | **SDAE**

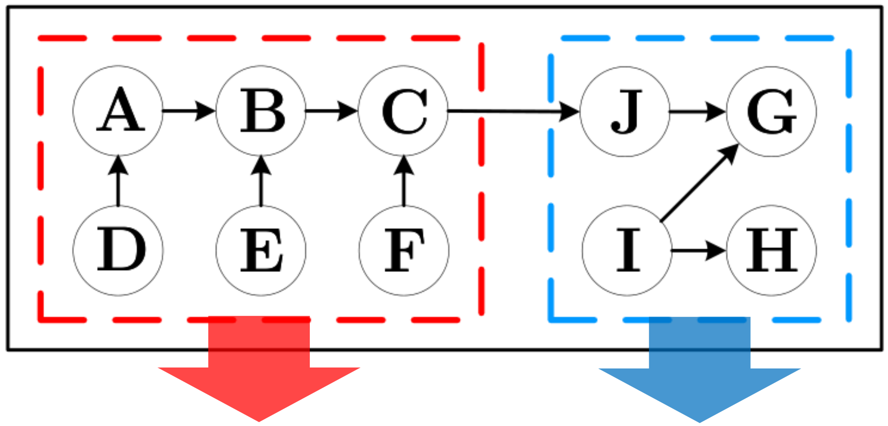
Two-way interaction

- More powerful representation
- Infer missing ratings from content
- Infer missing content from ratings

Notation:

- $\mathbf{R}$ rating of item j from user i
- $\mathbf{v}$ latent vector of item j
- $\mathbf{u}$ latent vector of user i
- $\mathbf{x}_0$ corrupted input
- $\mathbf{x}_c$ clean input
- $\mathbf{W}^+$ weights and biases
- $\mathbf{x}_{L/2}$ content representation

A Principled Probabilistic Framework



Perception Component   Task-Specific Component

Perception Variables: A B C D E F

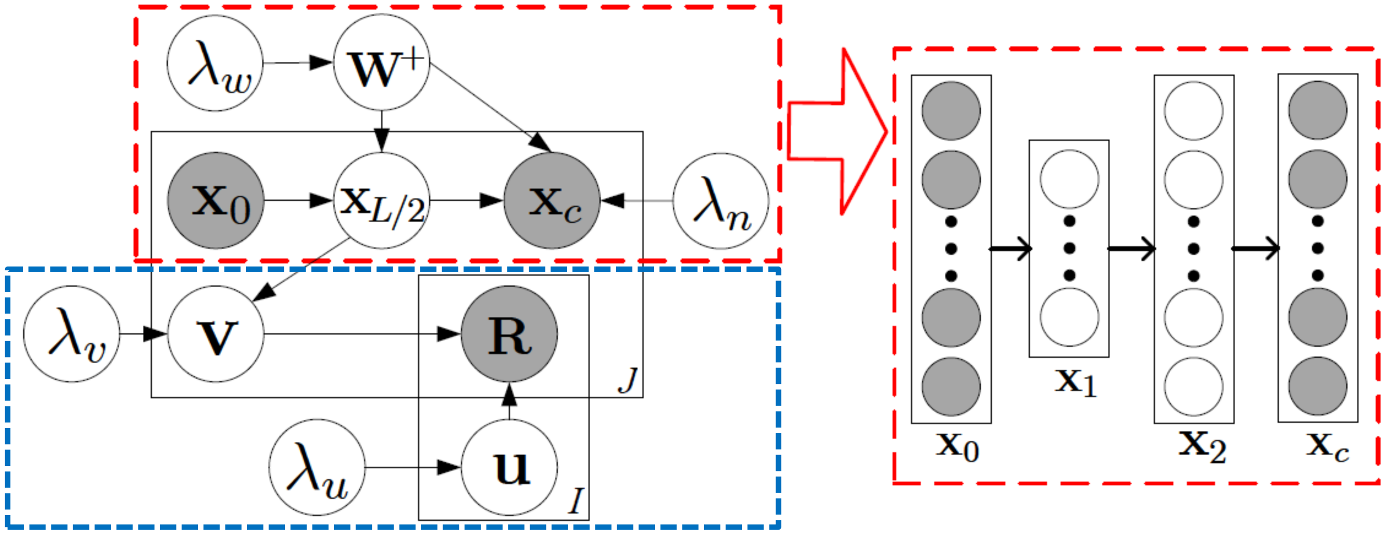
Task Variables: G H I

Hinge Variables: J

[Wang et al. TKDE 2016]

CDL with Two Components

Graphical model:



Collaborative deep learning **SDAE**

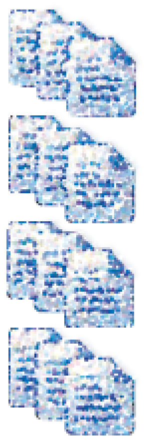
Two-way interaction

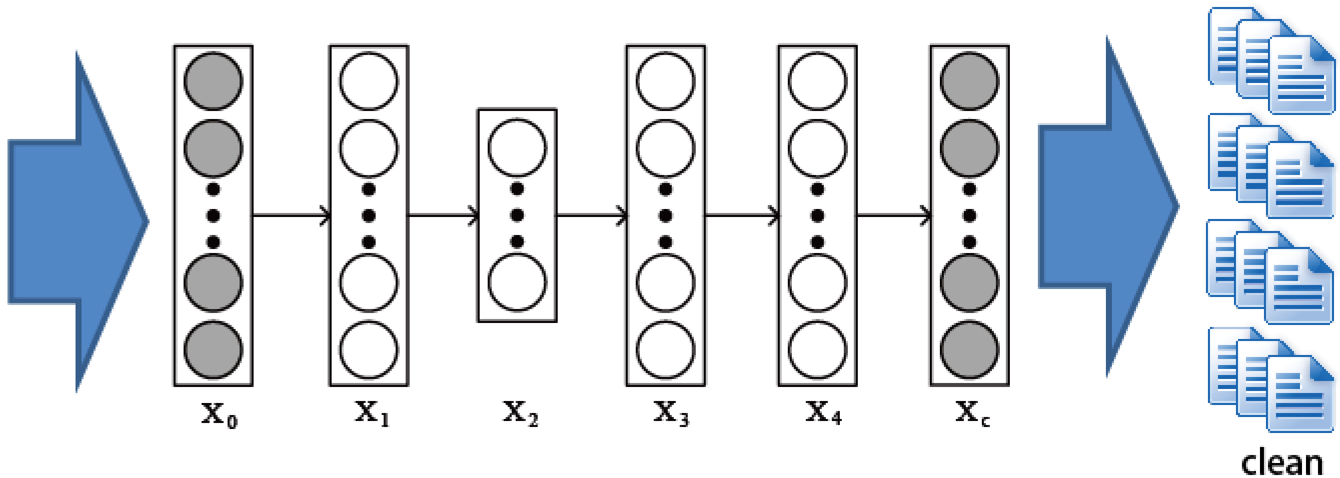
- More powerful representation
- Infer missing ratings from content
- Infer missing content from ratings

Notation:

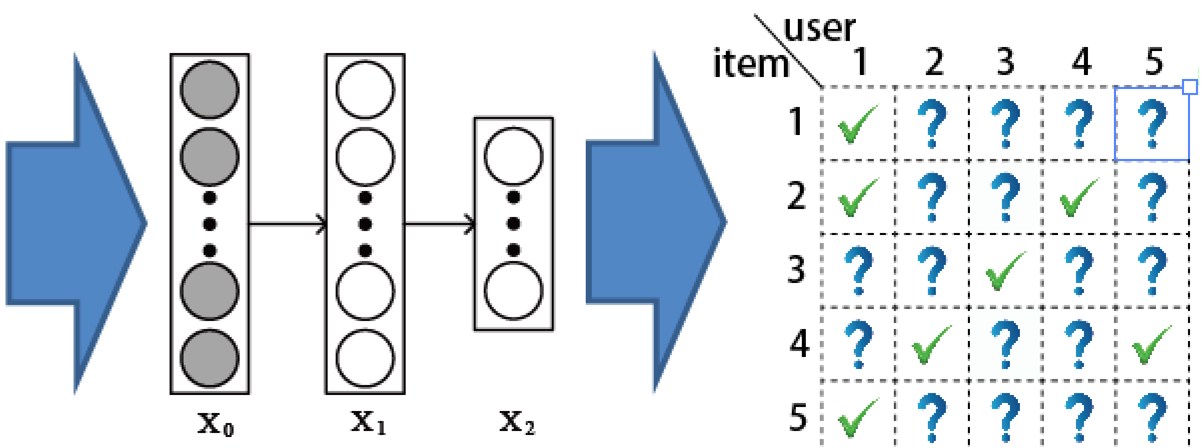
- $\mathbf{R}$ rating of item j from user i
- $\mathbf{v}$ latent vector of item j
- $\mathbf{u}$ latent vector of user i
- $\mathbf{x}_0$ corrupted input
- $\mathbf{x}_c$ clean input
- $\mathbf{W}^+$ weights and biases
- $\mathbf{x}_{L/2}$ content representation

Collaborative Deep Learning

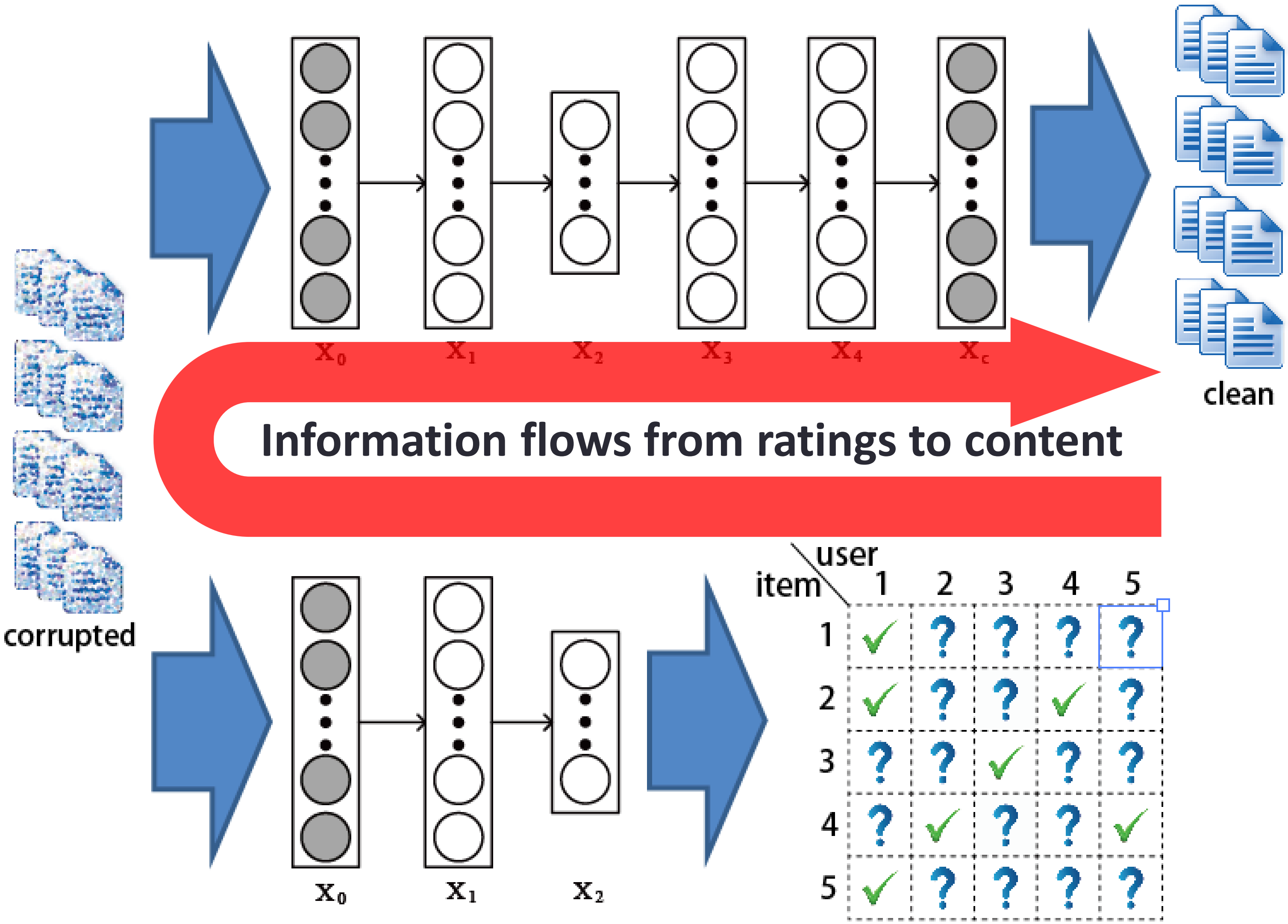




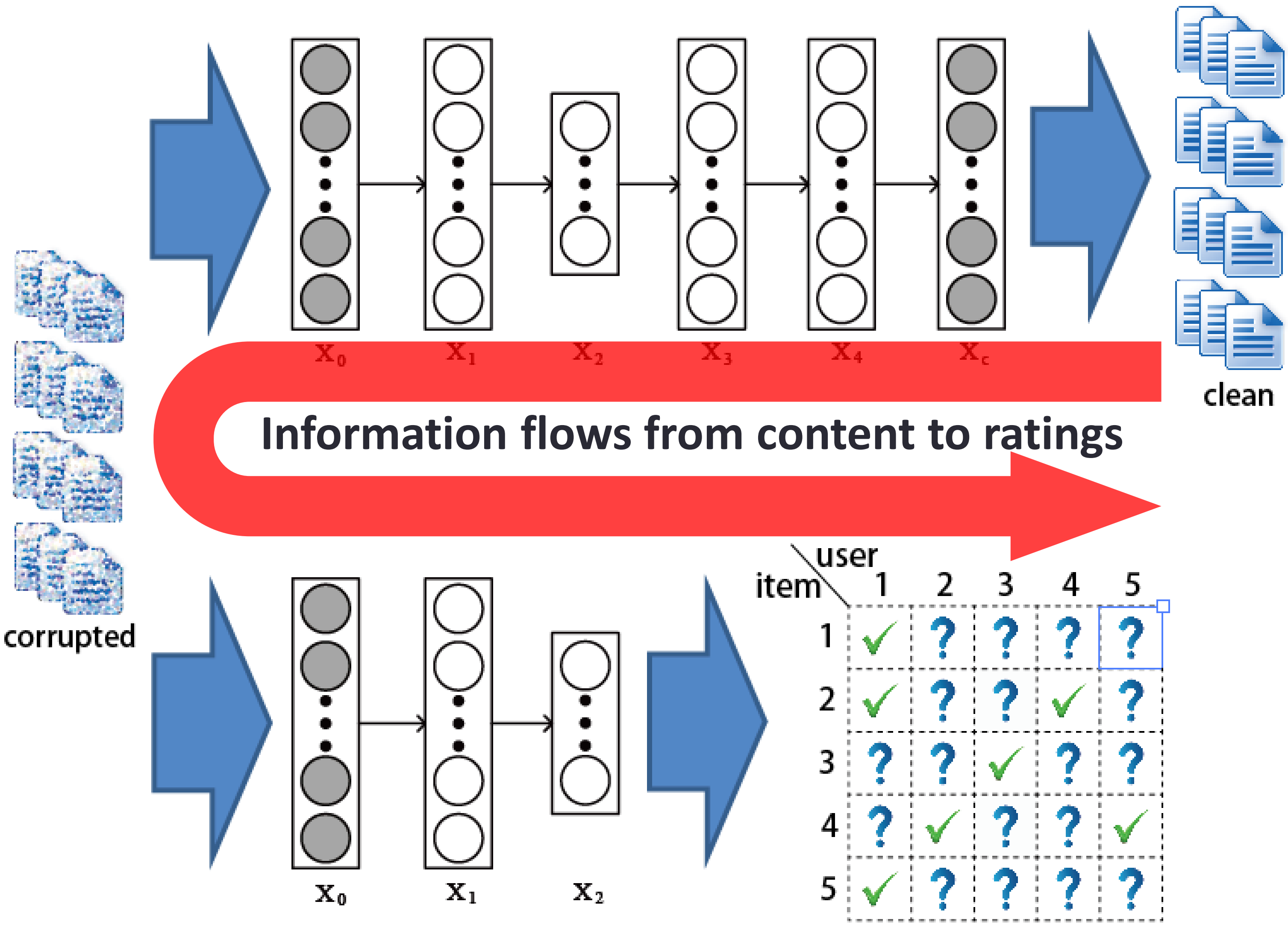
Neural network representation for degenerated CDL



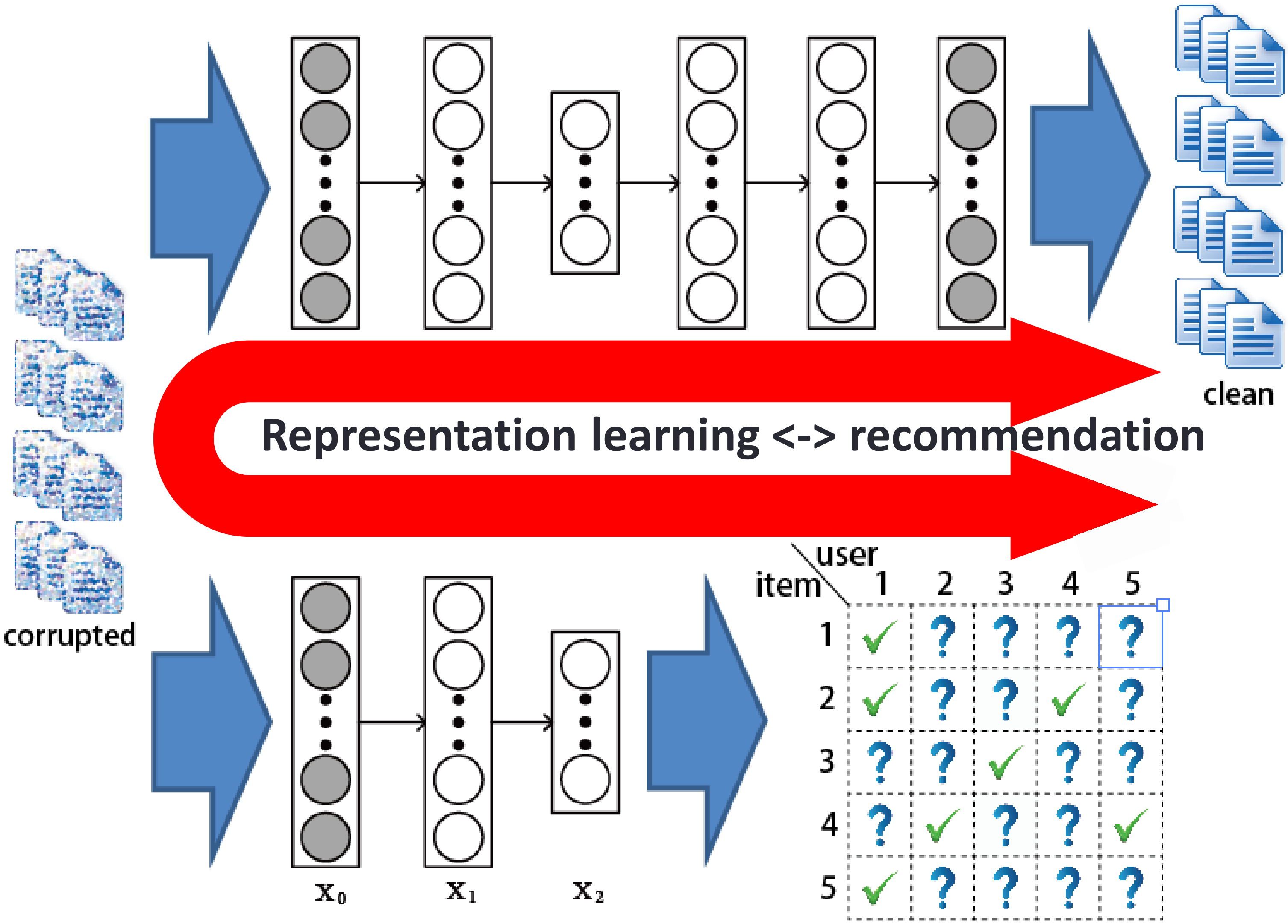
Collaborative Deep Learning



Collaborative Deep Learning



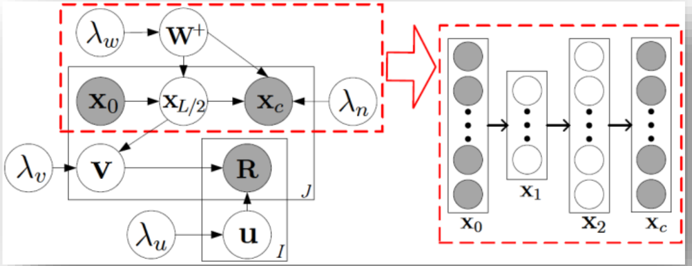
Collaborative Deep Learning



Learning

maximizing the posterior probability is equivalent to maximizing the joint log-likelihood

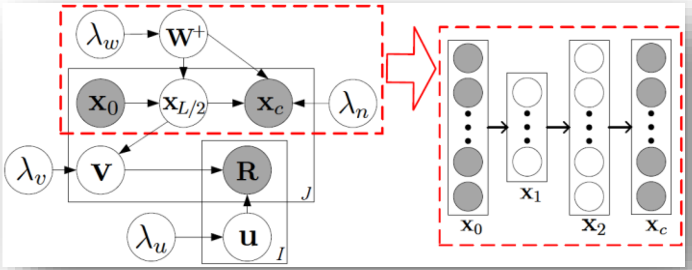
$$
\begin{aligned}
\mathscr{L} = & -\frac{\lambda_u}{2}\sum_i \|\mathbf{u}_i\|_2^2 - \frac{\lambda_w}{2}\sum_l (\|\mathbf{W}_l\|_F^2 + \|\mathbf{b}_l\|_2^2) \\
& -\frac{\lambda_v}{2}\sum_j \|\mathbf{v}_j - \mathbf{X}^T_{\frac{L}{2},j*}\|_2^2 - \frac{\lambda_n}{2}\sum_j \|\mathbf{X}_{L,j*} - \mathbf{X}_{c,j*}\|_2^2 \\
& -\frac{\lambda_s}{2}\sum_l \sum_j \|\sigma(\mathbf{X}_{l-1,j*}\mathbf{W}_l + \mathbf{b}_l) - \mathbf{X}_{l,j*}\|_2^2 \\
& -\sum_{i,j} \frac{\mathbf{C}_{ij}}{2}(\mathbf{R}_{ij} - \mathbf{u}_i^T \mathbf{v}_j)^2.
\end{aligned}
$$



Learning

Prior (regularization) for user latent vectors, weights, and biases

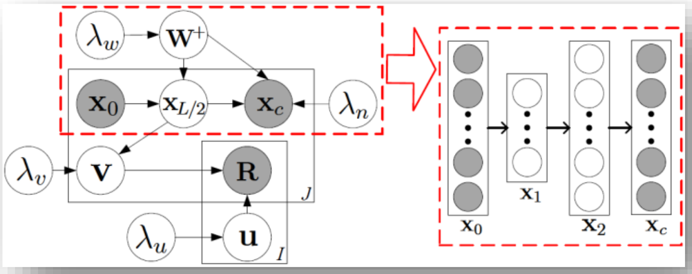
$$
\begin{aligned}
\mathscr{L} = & \boxed{-\frac{\lambda_u}{2}\sum_i \|\mathbf{u}_i\|_2^2 - \frac{\lambda_w}{2}\sum_l (\|\mathbf{W}_l\|_F^2 + \|\mathbf{b}_l\|_2^2)} \\
& - \frac{\lambda_v}{2}\sum_j \|\mathbf{v}_j - \mathbf{X}_{\frac{L}{2},j*}^T\|_2^2 - \frac{\lambda_n}{2}\sum_j \|\mathbf{X}_{L,j*} - \mathbf{X}_{c,j*}\|_2^2 \\
& - \frac{\lambda_s}{2}\sum_l \sum_j \|\sigma(\mathbf{X}_{l-1,j*}\mathbf{W}_l + \mathbf{b}_l) - \mathbf{X}_{l,j*}\|_2^2 \\
& - \sum_{i,j} \frac{\mathbf{C}_{ij}}{2}(\mathbf{R}_{ij} - \mathbf{u}_i^T \mathbf{v}_j)^2 .
\end{aligned}
$$



Learning

Generating item latent vectors from content representation with Gaussian offset

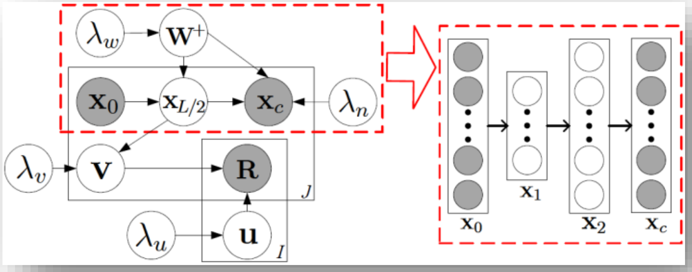
$$
\begin{aligned}
\mathscr{L} = & -\frac{\lambda_u}{2}\sum_i \|\mathbf{u}_i\|_2^2 - \frac{\lambda_w}{2}\sum_l \left(\|\mathbf{W}_l\|_F^2 + \|\mathbf{b}_l\|_2^2\right) \\
& \boxed{-\frac{\lambda_v}{2}\sum_j \|\mathbf{v}_j - \mathbf{X}^T_{\frac{L}{2},j*}\|_2^2} - \frac{\lambda_n}{2}\sum_j \|\mathbf{X}_{L,j*} - \mathbf{X}_{c,j*}\|_2^2 \\
& -\frac{\lambda_s}{2}\sum_l\sum_j \|\sigma(\mathbf{X}_{l-1,j*}\mathbf{W}_l + \mathbf{b}_l) - \mathbf{X}_{l,j*}\|_2^2 \\
& -\sum_{i,j}\frac{\mathbf{C}_{ij}}{2}(\mathbf{R}_{ij} - \mathbf{u}_i^T\mathbf{v}_j)^2.
\end{aligned}
$$



Learning

'Generating' clean input from the output of probabilistic SDAE with Gaussian offset

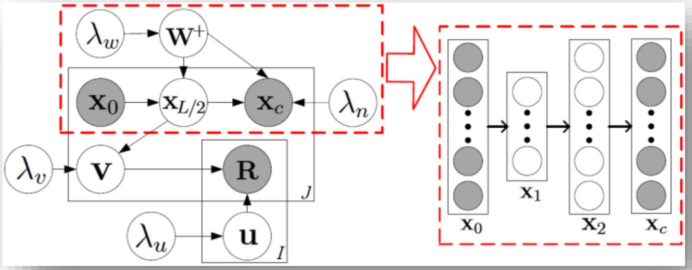
$$
\begin{aligned}
\mathscr{L} = &-\frac{\lambda_u}{2}\sum_i \|\mathbf{u}_i\|_2^2 - \frac{\lambda_w}{2}\sum_l (\|\mathbf{W}_l\|_F^2 + \|\mathbf{b}_l\|_2^2) \\
&- \frac{\lambda_v}{2}\sum_j \|\mathbf{v}_j - \mathbf{X}_{\frac{L}{2},j*}^T\|_2^2 - \boxed{\frac{\lambda_n}{2}\sum_j \|\mathbf{X}_{L,j*} - \mathbf{X}_{c,j*}\|_2^2} \\
&- \frac{\lambda_s}{2}\sum_l \sum_j \|\sigma(\mathbf{X}_{l-1,j*}\mathbf{W}_l + \mathbf{b}_l) - \mathbf{X}_{l,j*}\|_2^2 \\
&- \sum_{i,j} \frac{\mathbf{C}_{ij}}{2}(\mathbf{R}_{ij} - \mathbf{u}_i^T \mathbf{v}_j)^2.
\end{aligned}
$$



Learning

Generating the input of Layer l from the output of Layer l-1 with Gaussian offset

$$\mathscr{L} = -\frac{\lambda_u}{2}\sum_i \|\mathbf{u}_i\|_2^2 - \frac{\lambda_w}{2}\sum_l (\|\mathbf{W}_l\|_F^2 + \|\mathbf{b}_l\|_2^2)$$
$$-\frac{\lambda_v}{2}\sum_j \|\mathbf{v}_j - \mathbf{X}_{\frac{L}{2},j*}^T\|_2^2 - \frac{\lambda_n}{2}\sum_j \|\mathbf{X}_{L,j*} - \mathbf{X}_{c,j*}\|_2^2$$
$$\boxed{-\frac{\lambda_s}{2}\sum_l \sum_j \|\sigma(\mathbf{X}_{l-1,j*}\mathbf{W}_l + \mathbf{b}_l) - \mathbf{X}_{l,j*}\|_2^2}$$
$$-\sum_{i,j} \frac{\mathbf{C}_{ij}}{2}(\mathbf{R}_{ij} - \mathbf{u}_i^T\mathbf{v}_j)^2.$$



Learning

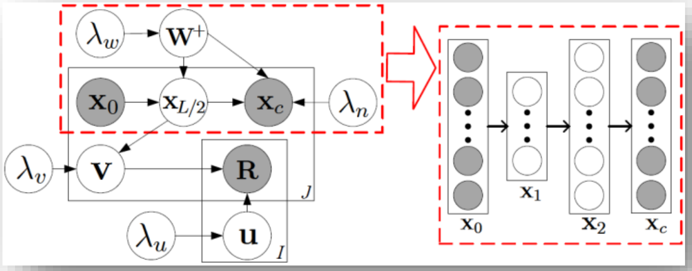
measures the error of predicted ratings

$$\mathscr{L} = -\frac{\lambda_u}{2}\sum_i \|\mathbf{u}_i\|_2^2 - \frac{\lambda_w}{2}\sum_l (\|\mathbf{W}_l\|_F^2 + \|\mathbf{b}_l\|_2^2)$$

$$-\frac{\lambda_v}{2}\sum_j \|\mathbf{v}_j - \mathbf{X}^T_{\frac{L}{2},j*}\|_2^2 - \frac{\lambda_n}{2}\sum_j \|\mathbf{X}_{L,j*} - \mathbf{X}_{c,j*}\|_2^2$$

$$-\frac{\lambda_s}{2}\sum_l\sum_j \|\sigma(\mathbf{X}_{l-1,j*}\mathbf{W}_l + \mathbf{b}_l) - \mathbf{X}_{l,j*}\|_2^2$$

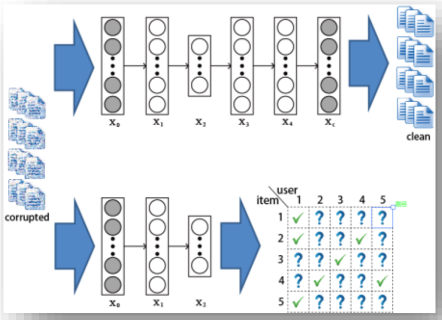
$$\boxed{-\sum_{i,j}\frac{\mathbf{C}_{ij}}{2}(\mathbf{R}_{ij} - \mathbf{u}_i^T\mathbf{v}_j)^2}.$$



Learning

If λ_s goes to infinity, the likelihood simplifies to

$$
\begin{aligned}
\mathscr{L} = & -\frac{\lambda_u}{2}\sum_i \|\mathbf{u}_i\|_2^2 - \frac{\lambda_w}{2}\sum_l (\|\mathbf{W}_l\|_F^2 + \|\mathbf{b}_l\|_2^2) \\
& -\frac{\lambda_v}{2}\sum_j \|\mathbf{v}_j - f_e(\mathbf{X}_{0,j*}, \mathbf{W}^+)^T\|_2^2 \\
& -\frac{\lambda_n}{2}\sum_j \|f_r(\mathbf{X}_{0,j*}, \mathbf{W}^+) - \mathbf{X}_{c,j*}\|_2^2 \\
& -\sum_{i,j} \frac{\mathbf{C}_{ij}}{2}(\mathbf{R}_{ij} - \mathbf{u}_i^T \mathbf{v}_j)^2,
\end{aligned}
$$



Update Rules

For U and V, use block coordinate descent:

$$\mathbf{u}_i \leftarrow (\mathbf{V}\mathbf{C}_i\mathbf{V}^T + \lambda_u \mathbf{I}_K)^{-1}\mathbf{V}\mathbf{C}_i\mathbf{R}_i$$

$$\mathbf{v}_j \leftarrow (\mathbf{U}\mathbf{C}_i\mathbf{U}^T + \lambda_v \mathbf{I}_K)^{-1}(\mathbf{U}\mathbf{C}_j\mathbf{R}_j + \boxed{\lambda_v f_e(\mathbf{X}_{0,j*}, \mathbf{W}^+)^T})$$

For W and b, use a modified version of backpropagation:

$$\nabla_{\mathbf{W}_l} \mathscr{L} = -\lambda_w \mathbf{W}_l$$

$$\boxed{-\lambda_v \sum_j \nabla_{\mathbf{W}_l} f_e(\mathbf{X}_{0,j*}, \mathbf{W}^+)^T (f_e(\mathbf{X}_{0,j*}, \mathbf{W}^+)^T - \mathbf{v}_j)}$$

$$-\lambda_n \sum_j \nabla_{\mathbf{W}_l} f_r(\mathbf{X}_{0,j*}, \mathbf{W}^+)(f_r(\mathbf{X}_{0,j*}, \mathbf{W}^+) - \mathbf{X}_{c,j*})$$

$$\nabla_{\mathbf{b}_l} \mathscr{L} = -\lambda_w \mathbf{b}_l$$

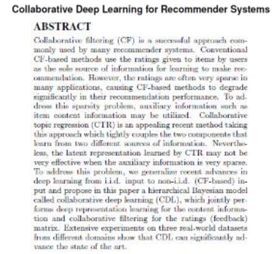
$$\boxed{-\lambda_v \sum_j \nabla_{\mathbf{b}_l} f_e(\mathbf{X}_{0,j*}, \mathbf{W}^+)^T (f_e(\mathbf{X}_{0,j*}, \mathbf{W}^+)^T - \mathbf{v}_j)}$$

$$-\lambda_n \sum_j \nabla_{\mathbf{b}_l} f_r(\mathbf{X}_{0,j*}, \mathbf{W}^+)(f_r(\mathbf{X}_{0,j*}, \mathbf{W}^+) - \mathbf{X}_{c,j*})$$

Datasets

	citeulike-a	citeulike-t	Netflix
#users	5551	7947	407261
#items	16980	25975	9228
#ratings	204987	134860	15348808

Content information

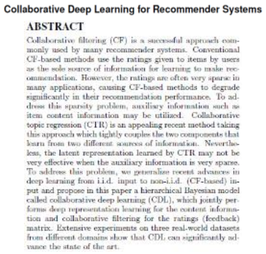


Collaborative Deep Learning for Recommender Systems

ABSTRACT

Collaborative filtering (CF) is a successful approach commonly used by many recommender systems. Conventional CF-based methods use the ratings given to items by users as the sole source of information for learning to make recommendation. However, the ratings are often very sparse in many applications, causing CF-based methods to degrade significantly in their recommendation performance. To address this sparsity problem, auxiliary information such as item content information may be utilized. Collaborative topic regression (CTR) is an appealing recent method taking this approach which tightly couples the two components that learn from two different sources of information. Nevertheless, the latent representation learned by CTR may not be very effective when the auxiliary information is very sparse. To address this problem, we generalize recent advances in deep learning from i.i.d. input to non-i.i.d. (CF-based) input and propose in this paper a hierarchical Bayesian model called collaborative deep learning (CDL), which jointly performs deep representation learning for the content information and collaborative filtering for the ratings (feedback) matrix. Extensive experiments on three real-world datasets from different domains show that CDL can significantly advance the state of the art.

Titles and abstracts

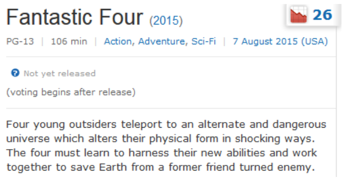


Collaborative Deep Learning for Recommender Systems

ABSTRACT

Collaborative filtering (CF) is a successful approach commonly used by many recommender systems. Conventional CF-based methods use the ratings given to items by users as the sole source of information for learning to make recommendation. However, the ratings are often very sparse in many applications, causing CF-based methods to degrade significantly in their recommendation performance. To address this sparsity problem, auxiliary information such as item content information may be utilized. Collaborative topic regression (CTR) is an appealing recent method taking this approach which tightly couples the two components that learn from two different sources of information. Nevertheless, the latent representation learned by CTR may not be very effective when the auxiliary information is very sparse. To address this problem, we generalize recent advances in deep learning from i.i.d. input to non-i.i.d. (CF-based) input and propose in this paper a hierarchical Bayesian model called collaborative deep learning (CDL), which jointly performs deep representation learning for the content information and collaborative filtering for the ratings (feedback) matrix. Extensive experiments on three real-world datasets from different domains show that CDL can significantly advance the state of the art.

Titles and abstracts



Fantastic Four (2015)

26

PG-13 | 106 min | Action, Adventure, Sci-Fi | 7 August 2015 (USA)

Not yet released

(voting begins after release)

Four young outsiders teleport to an alternate and dangerous universe which alters their physical form in shocking ways. The four must learn to harness their new abilities and work together to save Earth from a former friend turned enemy.

Movie plots

[Wang et al. KDD 2011]
[Wang et al. IJCAI 2013]

Evaluation Metrics

Recall:

$$\text{recall@}M = \frac{\text{number of items that the user likes among the top } M}{\text{total number of items that the user likes}}$$

Mean Average Precision (mAP):

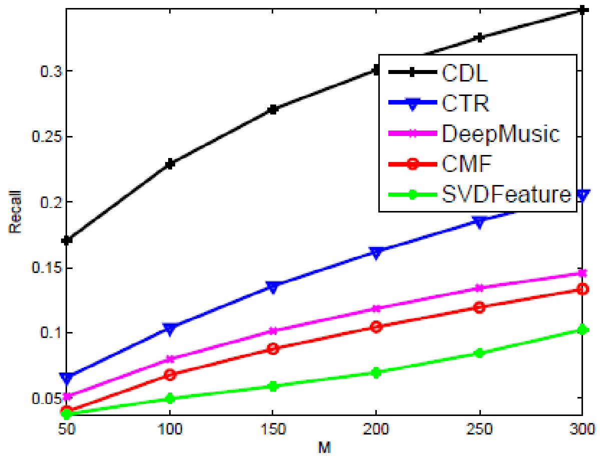
$$mAP = \frac{\sum_{q=1}^{Q} AveP(q)}{Q}$$

$$AveP = \frac{\sum_{k=1}^{n} (P(k) \times rel(k))}{\text{number of relevant items}}$$

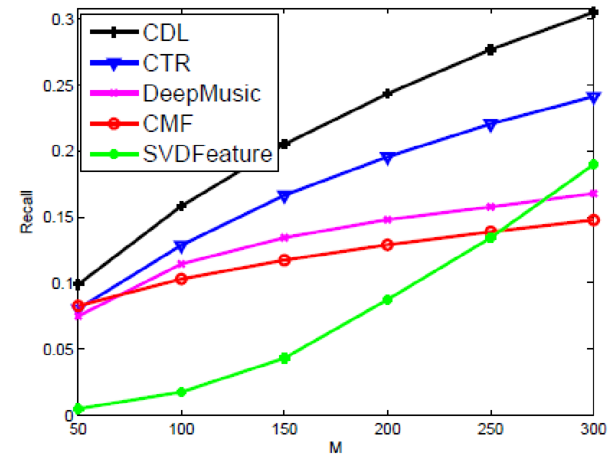
Higher recall and mAP indicate better recommendation performance

Recall@M

When the ratings are **very sparse**:

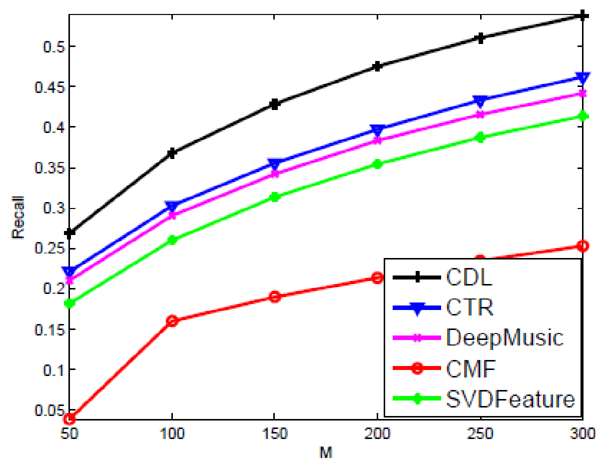


***citeulike-t*, sparse setting**

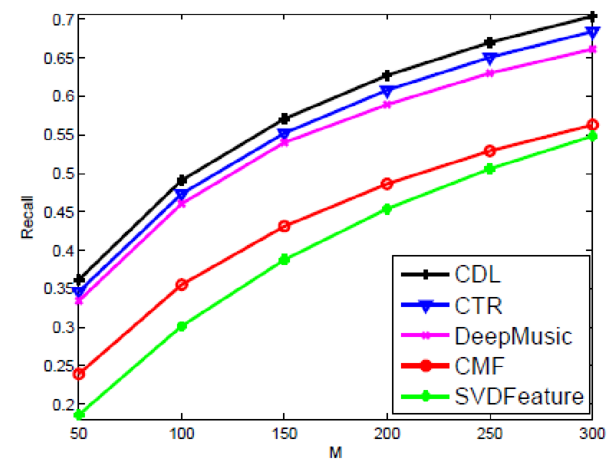


***Netflix*, sparse setting**

When the ratings are **dense**:



***citeulike-t*, dense setting**



***Netflix*, dense setting**

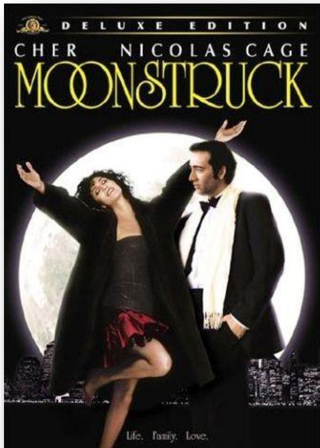
Mean Average Precision (mAP)

	citeulike-a	*citeulike-t*	*Netflix*
CDL	**0.0514**	**0.0453**	**0.0312**
CTR	0.0236	0.0175	0.0223
DeepMusic	0.0159	0.0118	0.0167
CMF	0.0164	0.0104	0.0158
SVDFeature	0.0152	0.0103	0.0187

Exactly the same as Oord et al. 2013, we set the cutoff point at 500 for each user.

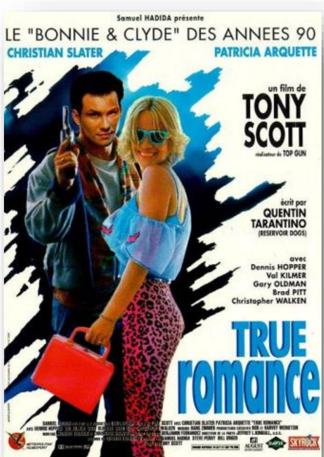
A relative performance boost of about 50%

Example User



Moonstruck

Romance Movies



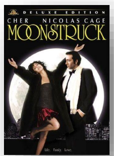
True Romance

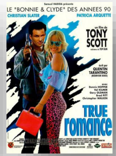
# training samples	2
Top 10 recommended movies by **CTR**	Swordfish
	A Fish Called Wanda
	Terminator 2
	A Clockwork Orange
	Sling Blade
	Bridget Jones's Diary
	Raising Arizona
	A Streetcar Named Desire
	The Untouchables
	The Full Monty

# training samples	2
Top 10 recommended movies by **CDL**	Snatch
	The Big Lebowski
	Pulp Fiction
	Kill Bill
	Raising Arizona
	The Big Chill
	Tootsie
	Sense and Sensibility
	Sling Blade
	Swinger

Precision: 20% VS 30%

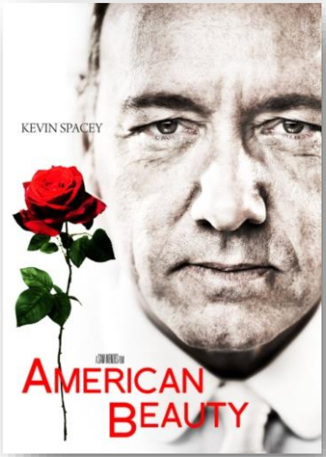
Example User





Johnny English

Action & Drama Movies



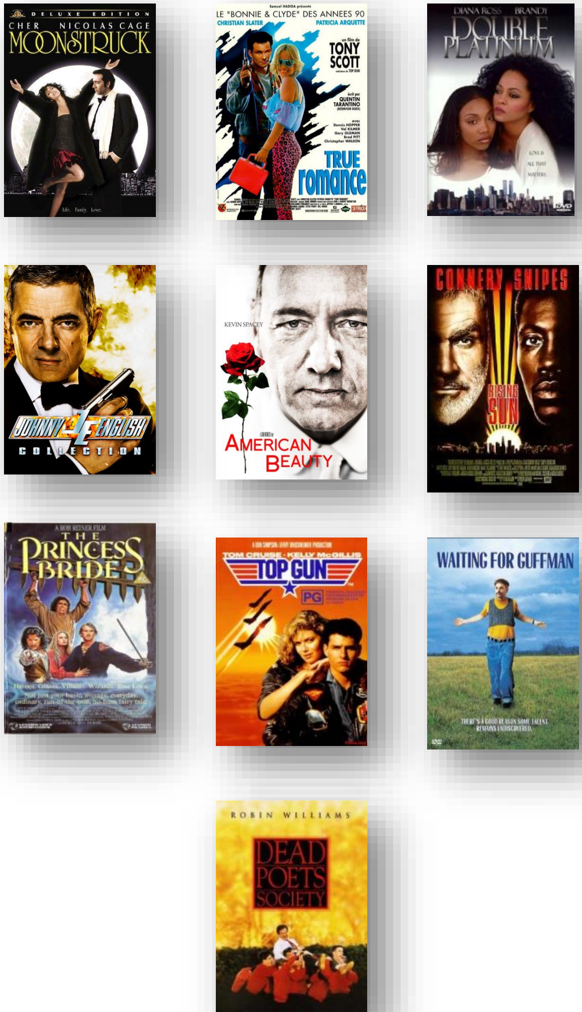
American Beauty

# training samples	4
Top 10 recommended movies by **CTR**	**Pulp Fiction**
	A Clockwork Orange
	Being John Malkovich
	Raising Arizona
	Sling Blade
	Swordfish
	A Fish Called Wanda
	Saving Grace
	The Graduate
	Monster's Ball

# training samples	4
Top 10 recommended movies by **CDL**	**Pulp Fiction**
	Snatch
	The Usual Suspect
	Kill Bill
	Momento
	The Big Lebowski
	One Flew Over the Cuckoo's Nest
	As Good as It Gets
	Goodfellas
	The Matrix

Precision: 20% VS 50%

Example User

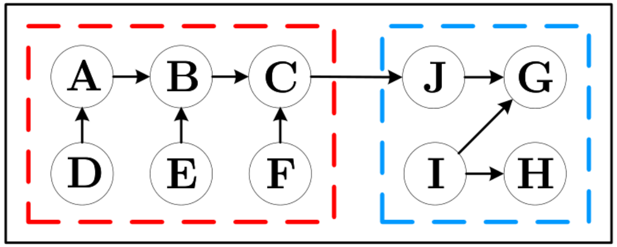


# training samples	10
Top 10 recommended movies by **CTR**	**Best in Snow**
	Chocolat
	Good Will Hunting
	Monty Python and the Holy Grail
	Being John Malkovich
	Raising Arizona
	The Graduate
	Swordfish
	Tootsie
	Saving Private Ryan

# training samples	10
Top 10 recommended movies by **CDL**	**Good Will Hunting**
	Best in Show
	The Big Lebowski
	A Few Good Men
	Monty Python and the Holy Grail
	Pulp Fiction
	The Matrix
	Chocolat
	The Usual Suspect
	CaddyShack

Precision: 50% VS 90%

Summary: Collaborative Deep Learning



- Non-i.i.d (collaborative) deep learning
- With a complex target
- First hierarchical Bayesian models for hybrid deep recommender system
- Significantly advance the state of the art

Marginalized CDL

Transformation to latent factors

CDL:

$$\mathscr{L} = -\frac{\lambda_u}{2}\sum_i \|\mathbf{u}_i\|_2^2 - \boxed{\frac{\lambda_v}{2}\sum_j \|\mathbf{v}_j - f_e(\mathbf{X}_{0,j*}, \mathbf{W}^+)^T\|_2^2}$$
$$- \boxed{\frac{\lambda_n}{2}\sum_j \|f_r(\mathbf{X}_{0,j*}, \mathbf{W}^+) - \mathbf{X}_{c,j*}\|_2^2} - \sum_{i,j}\frac{\mathbf{C}_{ij}}{2}(\mathbf{R}_{ij} - \mathbf{u}_i^T\mathbf{v}_j)^2$$

Reconstruction error

Transformation to latent factors

Marginalized CDL:

$$\mathscr{L} = -\frac{\lambda_u}{2}\sum_i \|\mathbf{u}_i\|_2^2 - \boxed{\frac{\lambda_v}{2}\sum_j \|\mathbf{v}_j\mathbf{P}_1 - \mathbf{X}_{0,j*}\mathbf{W}_1\|_2^2}$$
$$- \boxed{\sum_j \|\widetilde{\mathbf{X}}_{0,j*}\mathbf{W}_1 - \overline{\mathbf{X}}_{c,j*}\|_2^2} - \sum_{i,j}\frac{\mathbf{C}_{ij}}{2}(\mathbf{R}_{ij} - \mathbf{u}_i^T\mathbf{v}_j)^2$$

Reconstruction error

[Li et al., CIKM 2015]

Collaborative Deep Ranking

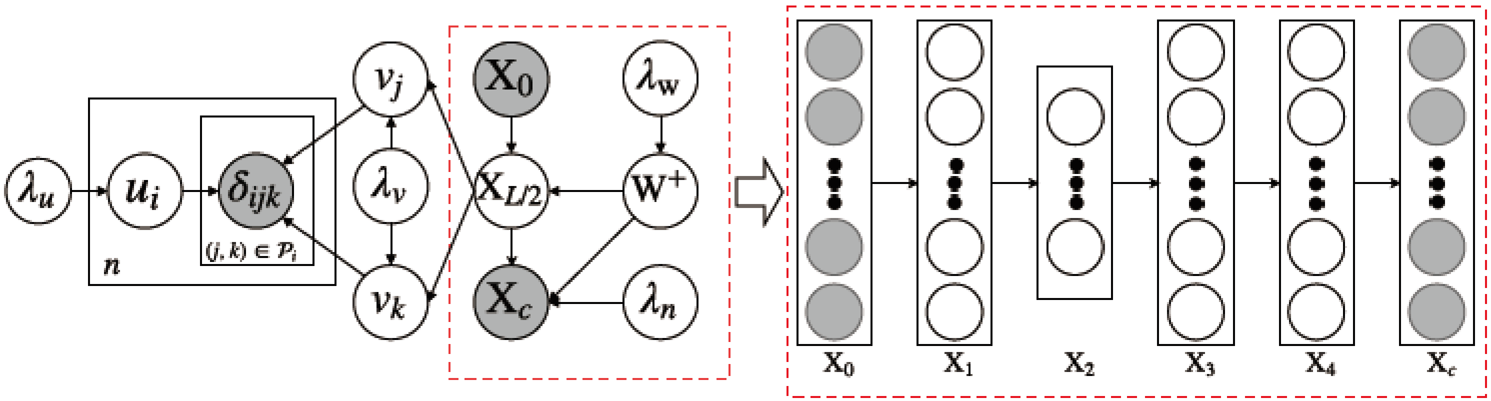


Fig. 1. The graphic model of CDR. SDAE with $L = 4$ is presented inside the dashed rectangle. Note that W^+ denotes the set of weight matrices and bias vectors of all layers.

[Ying et al., PAKDD 2016]

Generative Process: Collaborative Deep Ranking

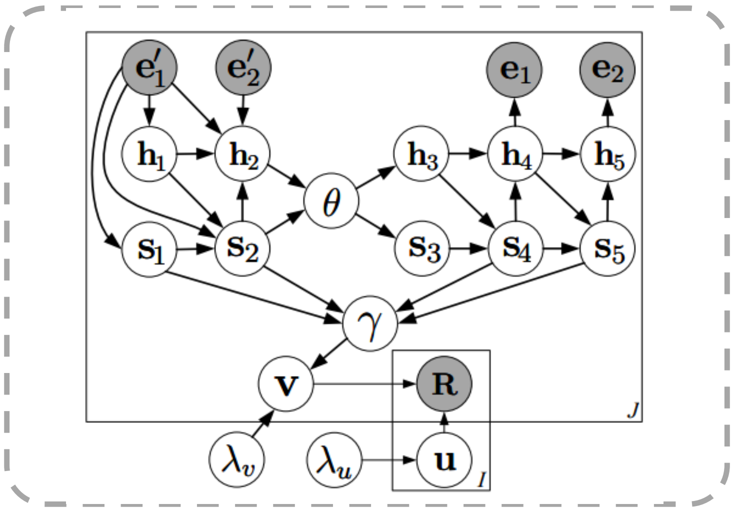
1. For each layer l of the SDAE network,
   (a) For each column q, draw the weight matrix and bias vector W_l^+, draw $W_{l,*q}^+ \sim \mathcal{N}(0, \lambda_w^{-1} I_{K_l})$.
   (b) For each row j of X_l, draw $X_{l,j*} \sim \mathcal{N}(\sigma(X_{l-1,j*} W_l + b_l), \lambda_s^{-1} I_{K_l})$
2. For each item j,
   (a) Draw a clean input $X_{c,j*} \sim \mathcal{N}(X_{L,j*}, \lambda_n^{-1} I_m)$
   (b) Draw a latent item offset vector $\epsilon_j \sim \mathcal{N}(0, \lambda_v^{-1} I_K)$and then set the latent item vector to be:
   $$v_j = \epsilon_j + X^T_{\frac{L}{2},j*}$$
3. For each user i,
   (a) Draw user factor vector $u_i \sim \mathcal{N}(0, \lambda_u^{-1} I_K)$
   (b) For each pair-wise preference $(j,k) \in \mathcal{P}_i$, where $\mathcal{P}_i = \{(j,k) : r_{ij} - r_{ik} > 0\}$, draw the estimator,
   $$\delta_{ijk} \sim \mathcal{N}(u_i^T v_j - u_i^T v_k, c_{ijk}^{-1})$$

CDL Variants

CDL Variants	Venue	Year
Deep CF	CIKM	2015
CD Ranking	PAKDD	2016
CF Networks	DLRS	2016
Collaborative KB Embedding	KDD	2016
AskGRU	RecSys	2016
ConvMF	RecSys	2016
Collaborative DAE	WSDM	2016
Collaborative Recurrent AE	NIPS	2016
DeepCoNN	WSDM	2017
Collaborative Metric Learning	WWW	2017
Additional SDAE	AAAI	2017

More details in http://wanghao.in/CDL.htm

Beyond Bag-of-Words: Documents as Sequences

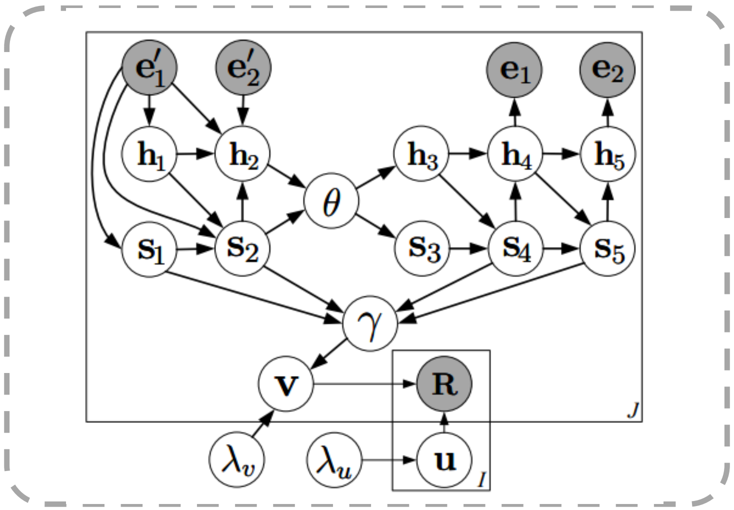


"Collaborative recurrent autoencoder: recommend while learning to fill in the blanks" [Wang et al., NIPS 2016a]

Motivation:

- A more natural way, take in one word at a time, model documents as sequences
- Jointly model preferences and sequence generation under the BDL framework

Beyond Bag-of-Words: Documents as Sequences



"Collaborative recurrent autoencoder: recommend while learning to fill in the blanks" [Wang et al., NIPS 2016a]

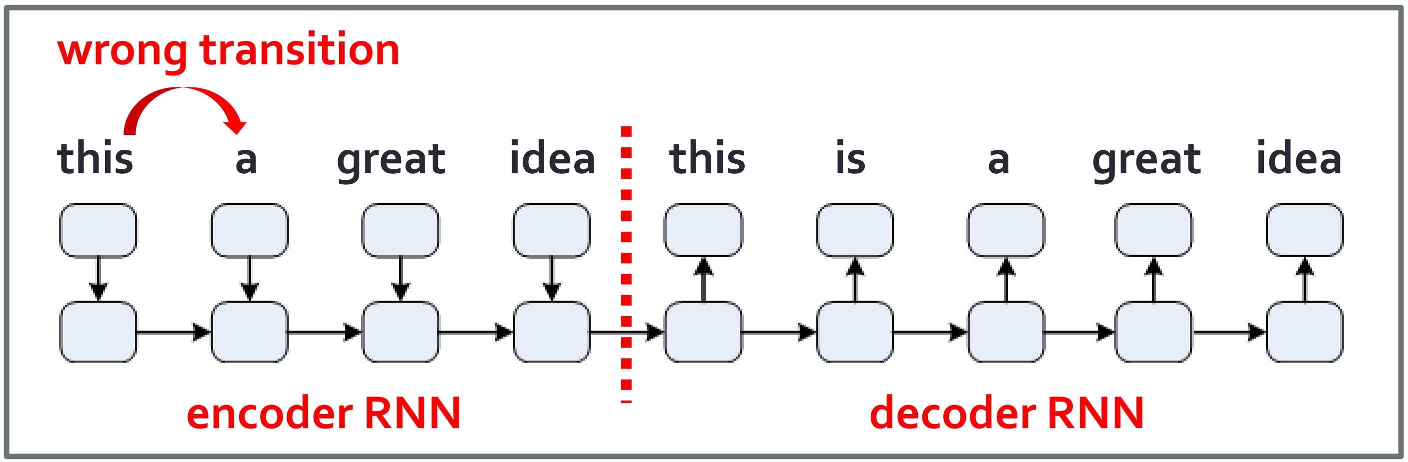
Main Idea:

- Joint learning in the BDL framework
- Wildcard denoising for robust representation

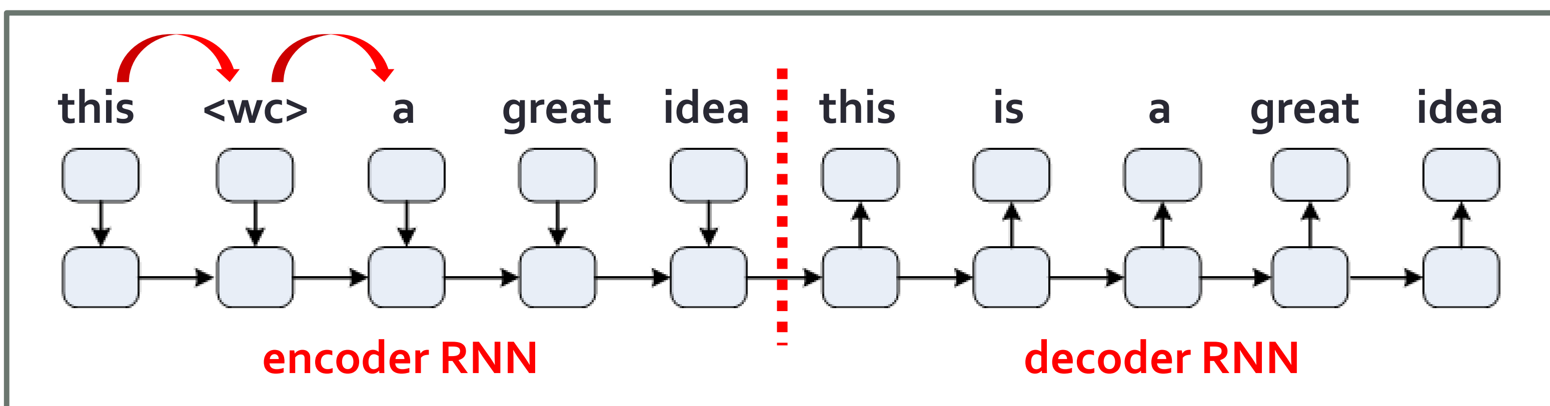
Wildcard Denoising

Sentence: This ~~is~~ a great idea. -> This is a great idea.

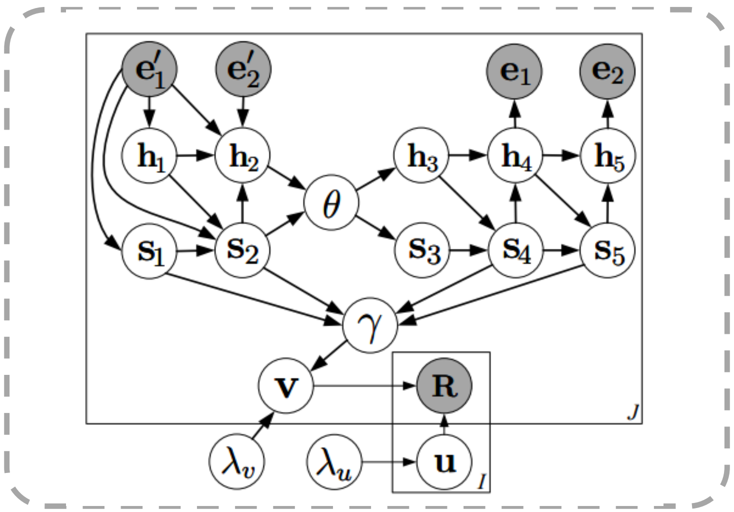
Direct Denoising:



Wildcard Denoising:



Documents as Sequences



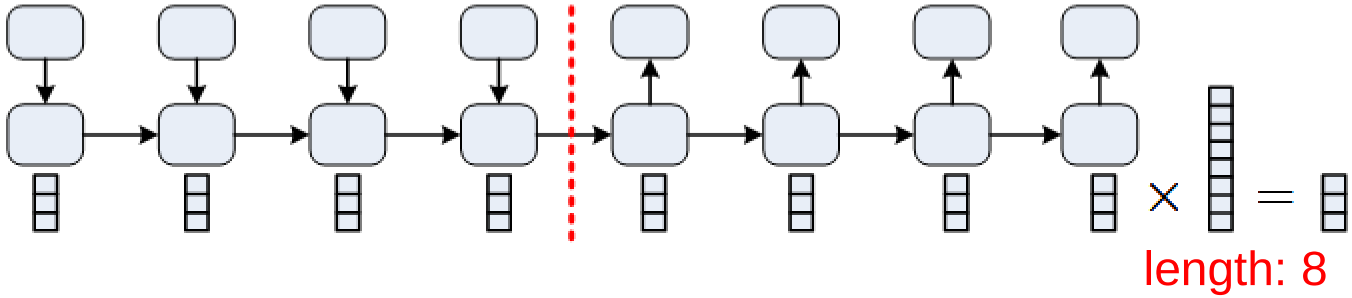
"Collaborative recurrent autoencoder: recommend while learning to fill in the blanks" [Wang et al., NIPS 2016a]

Main Idea:

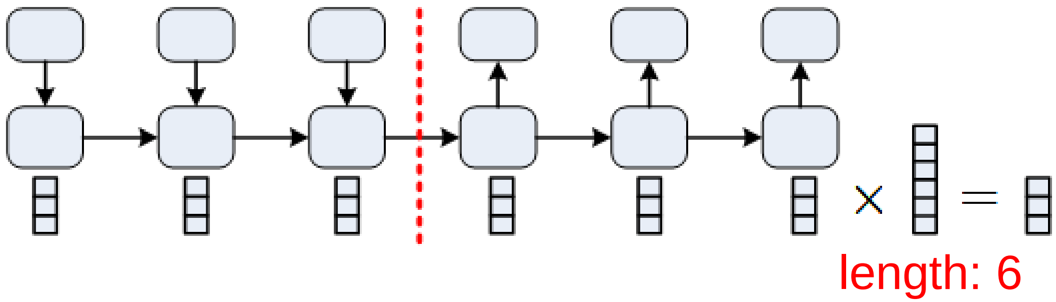
- Joint learning in the BDL framework
- Wildcard denoising for robust representation
- Beta-Pooling for variable-length sequences

Is Variable-Length Weight Vector Possible?

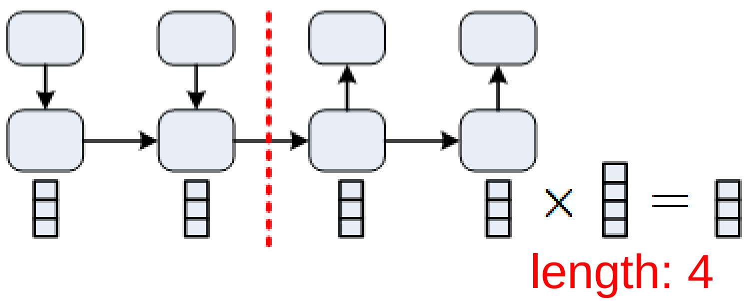
vector	length
sequence	4
weight	8



vector	length
sequence	3
weight	6



vector	length
sequence	2
weight	4



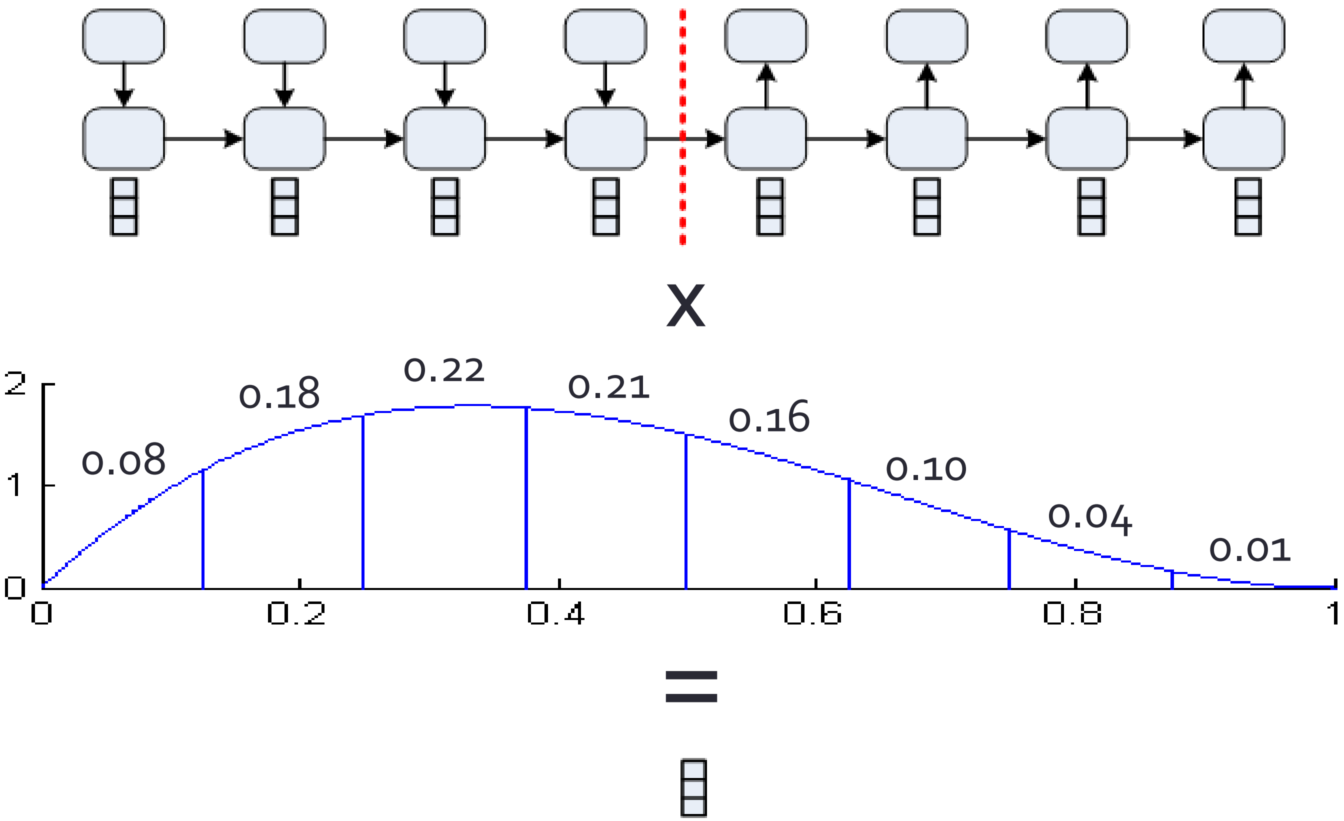
[Wang et al., NIPS 2016a]

Variable-Length Weight Vector with Beta Distributions

8 length-3 vectors

length-8 weight vector

one single vector

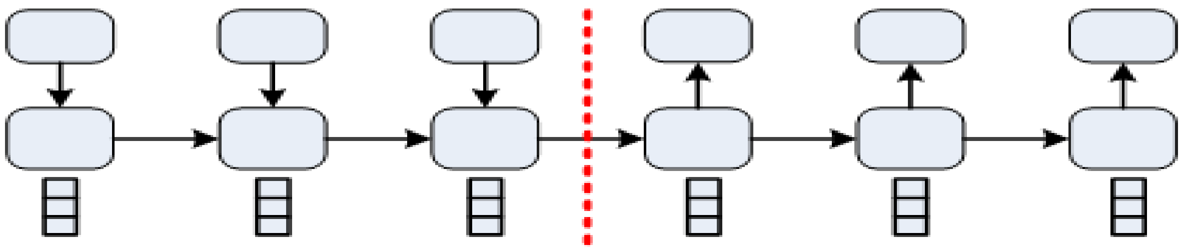


Use the area of the beta distribution to define the weights!

[Wang et al., NIPS 2016a]

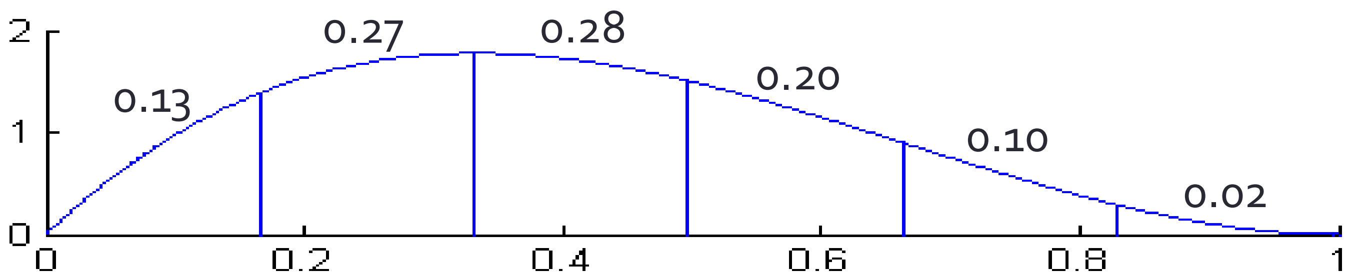
Variable-Length Weight Vector with Beta Distributions

6 length-3 vectors



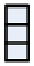
X

length-6 weight vector



=

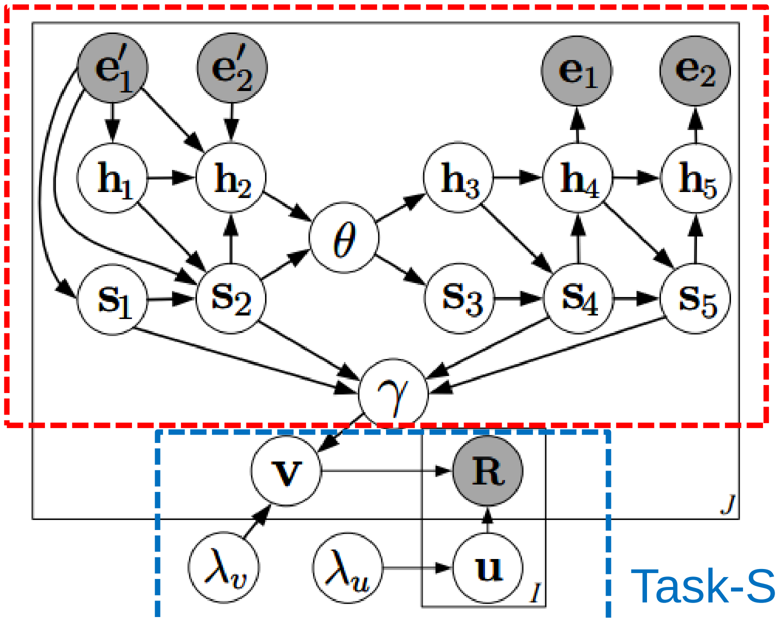
one single vector



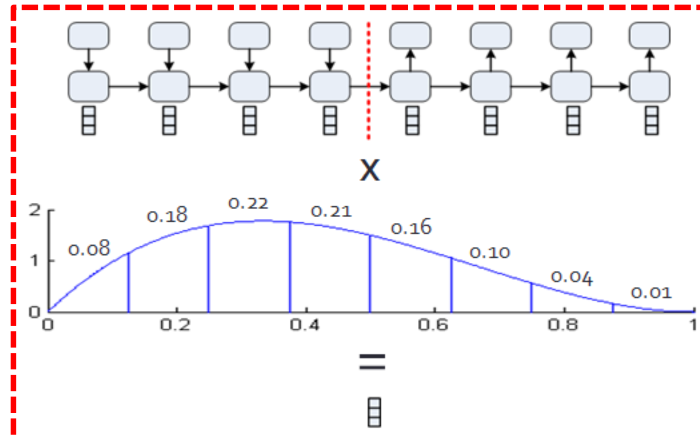
Use the area of the beta distribution to define the weights!

[Wang et al., NIPS 2016a]

Graphical Model: Collaborative Recurrent Autoencoder



Task-Specific Component



Perception Component

- Joint learning in the BDL framework
- Wildcard denoising for robust representation
- Beta-Pooling for variable-length sequences

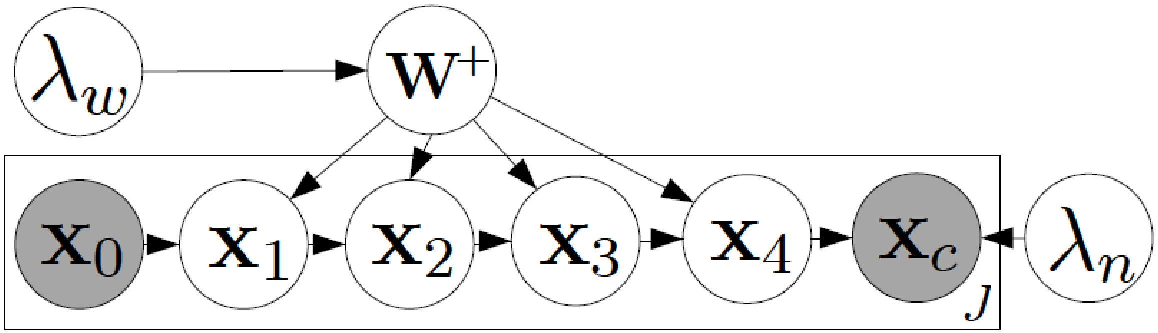
[Wang et al., NIPS 2016a]

Incorporating Relational Information

[Wang et al. AAAI 2015]
[Wang et al. AAAI 2017]

Probabilistic SDAE

Graphical model:



Generative process:

$$\mathbf{W}_{l,*n} \sim \mathcal{N}(0, \lambda_w^{-1}\mathbf{I}_{K_l})$$

$$\mathbf{b}_l \sim \mathcal{N}(0, \lambda_w^{-1}\mathbf{I}_{K_l})$$

$$\mathbf{X}_{l,j*} \sim \mathcal{N}(\sigma(\mathbf{X}_{l-1,j*}\mathbf{W}_l + \mathbf{b}_l), \boxed{\lambda_s^{-1}\mathbf{I}_{K_l}})$$

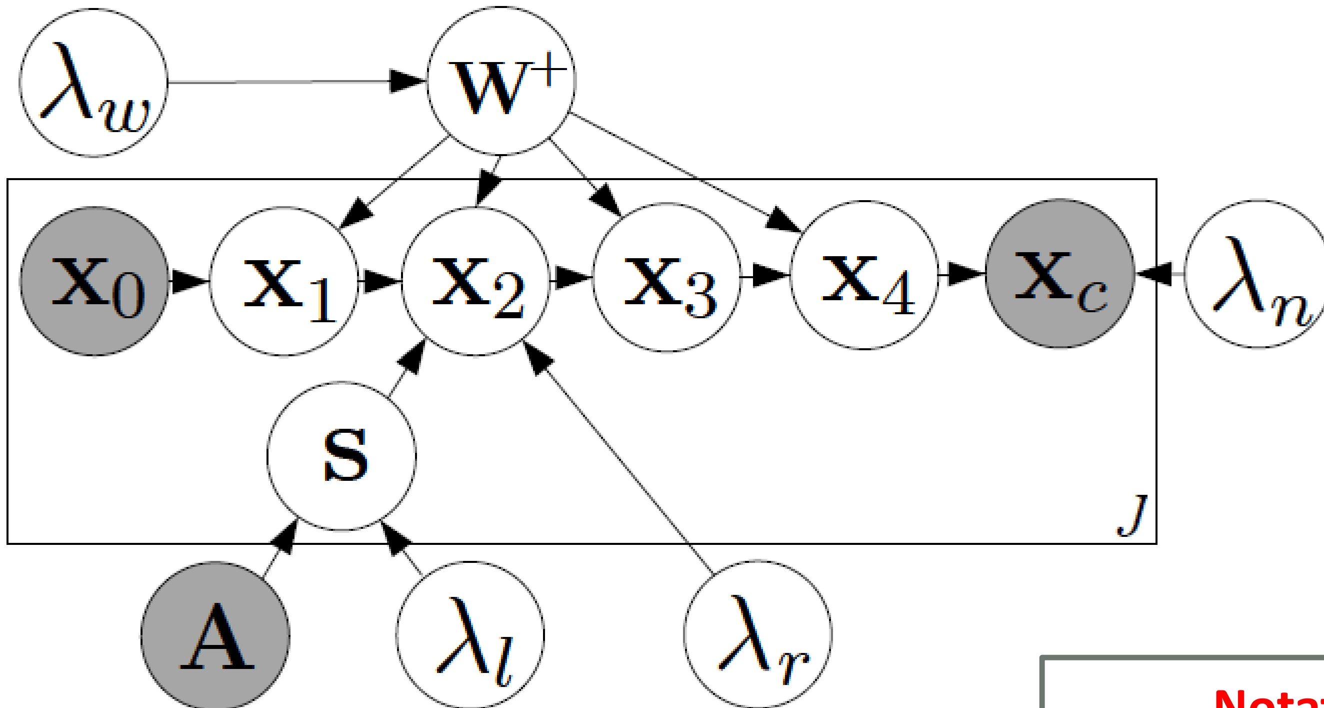
$$\mathbf{X}_{c,j*} \sim \mathcal{N}(\mathbf{X}_{L,j*}, \lambda_n^{-1}\mathbf{I}_B)$$

Generalized SDAE

Notation:

- $\mathbf{x}_0$ corrupted input
- $\mathbf{x}_c$ clean input
- $\mathbf{W}^+$ weights and biases

Relational SDAE: Graphical Model



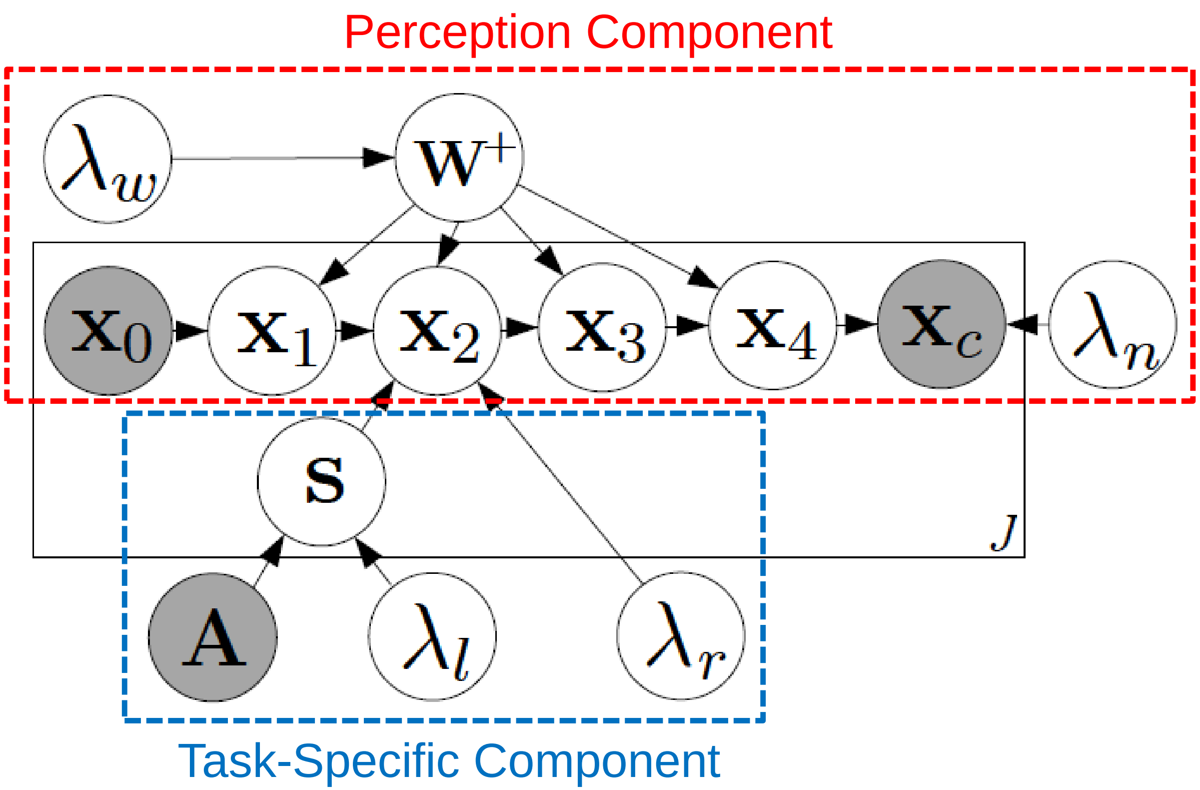
Notation:

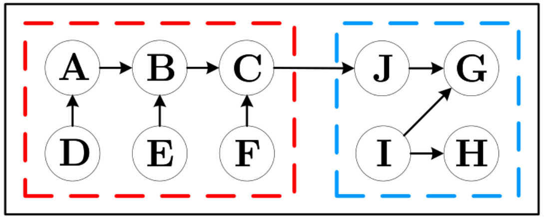
$\mathbf{x}_0$ corrupted input

$\mathbf{x}_c$ clean input

$\mathbf{A}$ adjacency matrix

Relational SDAE: Two Components





Relational SDAE: Generative Process

1. Draw the relational latent matrix $\mathbf{S}$ from a *matrix variate normal distribution*:

$$\mathbf{S} \sim \mathcal{N}_{K,J}(0, \mathbf{I}_K \otimes (\lambda_l \mathscr{L}_a)^{-1}).$$

2. For layer l of the SDAE where $l = 1, 2, \ldots, \frac{L}{2} - 1$,
   1. For each column n of the weight matrix $\mathbf{W}_l$, draw $\mathbf{W}_{l,*n} \sim \mathcal{N}(0, \lambda_w^{-1} \mathbf{I}_{K_l})$.
   2. Draw the bias vector $\mathbf{b}_l \sim \mathcal{N}(0, \lambda_w^{-1} \mathbf{I}_{K_l})$.
   3. For each row j of $\mathbf{X}_l$, draw

$$\mathbf{X}_{l,j*} \sim \mathcal{N}(\sigma(\mathbf{X}_{l-1,j*} \mathbf{W}_l + \mathbf{b}_l), \lambda_s^{-1} \mathbf{I}_{K_l}).$$

3. For layer $\frac{L}{2}$ of the SDAE network, draw the representation vector for item j from the product of two Gaussians (PoG):

$$\mathbf{X}_{\frac{L}{2},j*} \sim \mathrm{PoG}(\sigma(\mathbf{X}_{\frac{L}{2}-1,j*} \mathbf{W}_l + \mathbf{b}_l), \mathbf{s}_j^T, \lambda_s^{-1} \mathbf{I}_K, \lambda_r^{-1} \mathbf{I}_K).$$

Relational SDAE: Generative Process

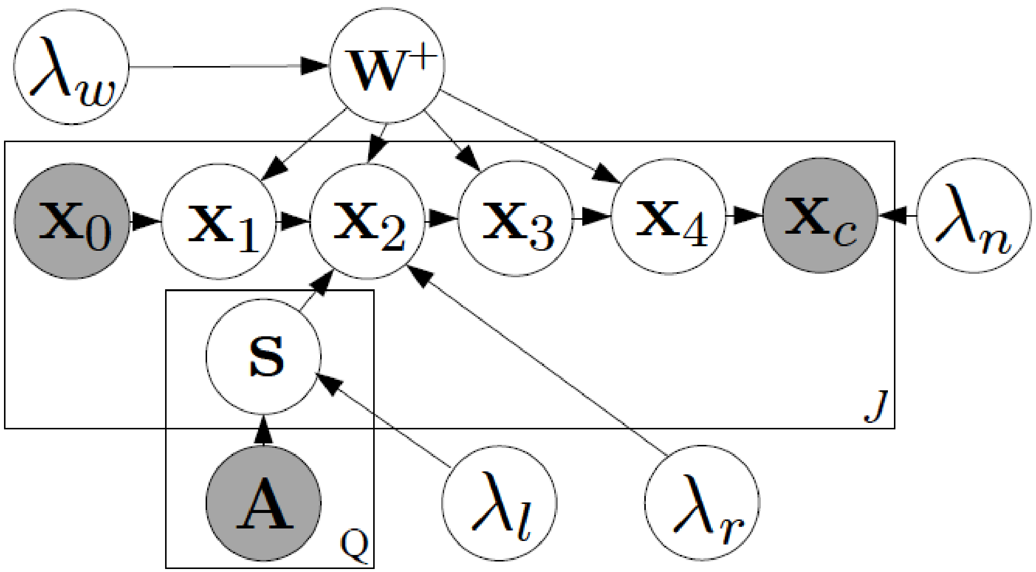
1. For layer l of the SDAE network where $l = \frac{L}{2} + 1, \frac{L}{2} + 2, \ldots, L$,
   1. For each column n of the weight matrix $\mathbf{W}_l$, draw $\mathbf{W}_{l,*n} \sim \mathcal{N}(0, \lambda_w^{-1}\mathbf{I}_{K_l})$.
   2. Draw the bias vector $\mathbf{b}_l \sim \mathcal{N}(0, \lambda_w^{-1}\mathbf{I}_{K_l})$.
   3. For each row j of $\mathbf{X}_l$, draw

$$\mathbf{X}_{l,j*} \sim \mathcal{N}(\sigma(\mathbf{X}_{l-1,j*}\mathbf{W}_l + \mathbf{b}_l), \lambda_s^{-1}\mathbf{I}_{K_l}).$$

2. For each item j, draw a clean input

$$\mathbf{X}_{c,j*} \sim \mathcal{N}(\mathbf{X}_{L,j*}, \lambda_n^{-1}\mathbf{I}_B).$$

Multi-Relational SDAE: Graphical Model



Product of Q+1 Gaussians

Multiple networks:
citation networks
co-author networks

Notation:

- $\mathbf{x}_0$ corrupted input
- $\mathbf{x}_c$ clean input
- $\mathbf{A}$ adjacency matrix

Relational SDAE: Objective Function

$$\mathscr{L} = \boxed{-\frac{\lambda_l}{2}\mathrm{tr}(\mathbf{S}\mathscr{L}_a\mathbf{S}^T)} - \boxed{\frac{\lambda_r}{2}\sum_j \|(\mathbf{s}_j^T - \mathbf{X}_{\frac{L}{2},j*})\|_2^2} - \frac{\lambda_w}{2}\sum_l(\|\mathbf{W}_l\|_F^2 + \|\mathbf{b}_l\|_2^2)$$
$$- \frac{\lambda_n}{2}\sum_j \|\mathbf{X}_{L,j*} - \mathbf{X}_{c,j*}\|_2^2 - \frac{\lambda_s}{2}\sum_l\sum_j \|\sigma(\mathbf{X}_{l-1,j*}\mathbf{W}_l + \mathbf{b}_l) - \mathbf{X}_{l,j*}\|_2^2.$$

Similar to the generalized SDAE, taking λ_s to infinity, the joint log-likelihood becomes:

$$\mathscr{L} = -\frac{\lambda_l}{2}\mathrm{tr}(\mathbf{S}\mathscr{L}_a\mathbf{S}^T) - \frac{\lambda_r}{2}\sum_j \|(\mathbf{s}_j^T - \mathbf{X}_{\frac{L}{2},j*})\|_2^2 - \frac{\lambda_w}{2}\sum_l(\|\mathbf{W}_l\|_F^2 + \|\mathbf{b}_l\|_2^2)$$
$$- \frac{\lambda_n}{2}\sum_j \|\mathbf{X}_{L,j*} - \mathbf{X}_{c,j*}\|_2^2,$$

Network **A** → Relational Matrix **S**

Relational Matrix **S** → Middle-Layer Representations

Update Rules

For $\mathbf{S}$:

$$\mathbf{S}_{k*}(t+1) \leftarrow \mathbf{S}_{k*}(t) + \delta(t)r(t)$$
$$r(t) \leftarrow \lambda_r \mathbf{X}^T_{\frac{L}{2},*k} - (\lambda_l \mathscr{L}_a + \lambda_r \mathbf{I}_J)\mathbf{S}_{k*}(t)$$
$$\delta(t) \leftarrow \frac{r(t)^T r(t)}{r(t)^T (\lambda_l \mathscr{L}_a + \lambda_r \mathbf{I}_J) r(t)}.$$

For $\mathbf{X}$, $\mathbf{W}$, and $\mathbf{b}$: Use Back Propagation.

From Representation to Tag Recommendation

Objective function:

$$\mathscr{L} = -\frac{\lambda_u}{2}\sum_i \|\mathbf{u}_i\|_2^2 - \frac{\lambda_v}{2}\sum_j \|\mathbf{v}_j - \mathbf{X}_{\frac{L}{2},j*}^T\|_2^2 - \sum_{i,j}\frac{c_{ij}}{2}(\mathbf{R}_{ij} - \mathbf{u}_i^T\mathbf{v}_j)^2,$$

where λ_u and λ_v are hyperparameters. c_{ij} is set to 1 for the existing ratings and 0.01 for the missing entries.

Algorithm

1. **Learning representation:**

   repeat

   Update $\mathbf{S}$ using the updating rules

   Update $\mathbf{X}$, $\mathbf{W}$, and $\mathbf{b}$

   until convergence

   Get resulting representation $\mathbf{X}_{\frac{L}{2},j*}$

2. **Learning $\mathbf{u}_i$ and $\mathbf{v}_j$:**

   Optimize the objective function $\mathscr{L}$

3. **Recommend tags to items according to the predicted R_{ij}:**

   $\mathbf{R}_{ij} = \mathbf{u}_i^T \mathbf{v}_j$

   Rank $\mathbf{R}_{1j}$,$\mathbf{R}_{2j}$,. . . ,$\mathbf{R}_{Ij}$

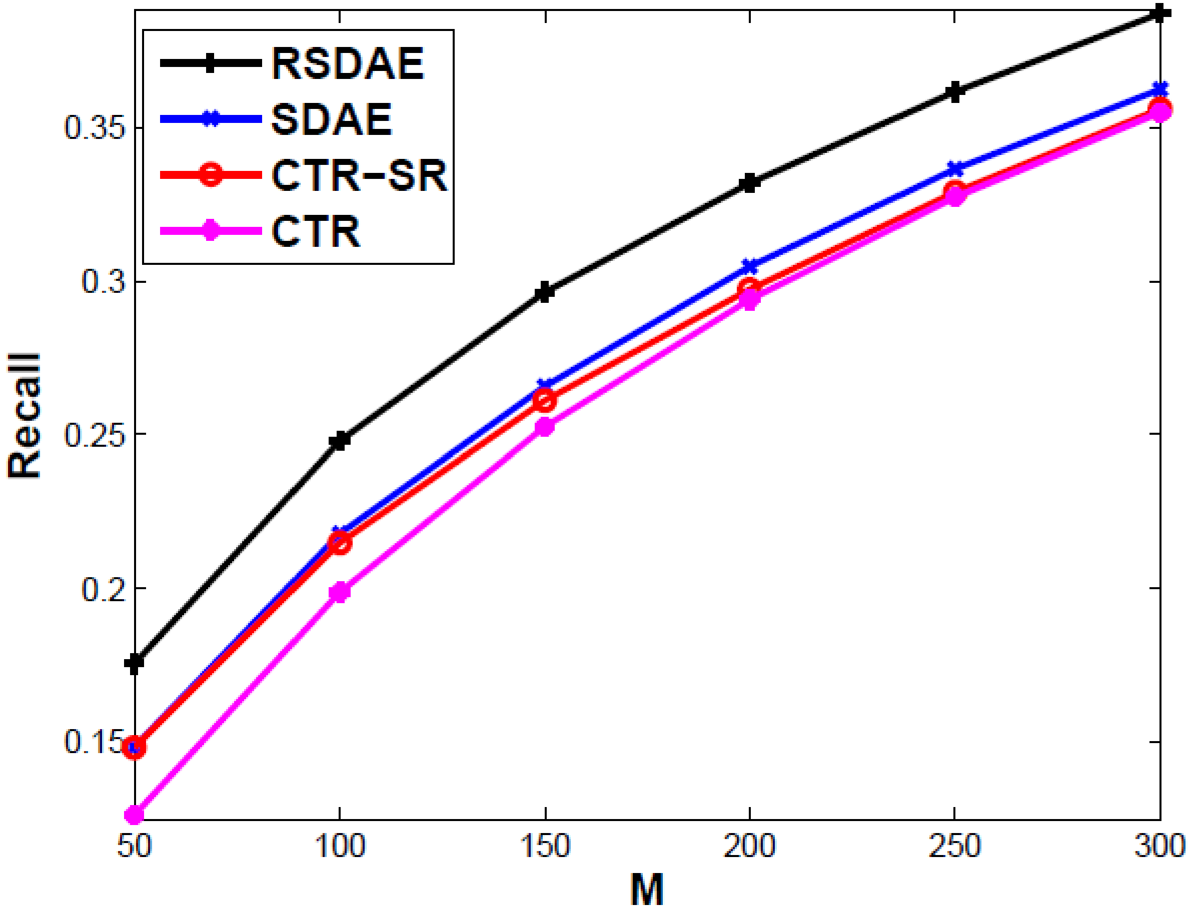
   Recommend tags with largest $\mathbf{R}_{ij}$ to item j

Datasets

Description of datasets

	citeulike-a	citeulike-t	movielens-plot
#items	16980	25975	7261
#tags	7386	8311	2988
#tag-item paris	204987	134860	51301
#relations	44709	32665	543621

Sparse Setting, *citeulike-a*



Case Study 1: Tagging Scientific Articles

An example article with recommended tags

Example Article	Title: Mining the Peanut Gallery: Opinion Extraction and Semantic Classification of Product Reviews			
	Top topic 1: language, text, mining, representation, semantic, concepts, words, relations, processing, categories			
Top 10 tags	SDAE	True?	RSDAE	True?
	1. instance	no	1. sentiment_analysis	no
	2. consumer	**yes**	2. instance	no
	3. sentiment_analysis	no	**3. consumer**	**yes**
	4. summary	no	4. summary	no
	5. 31july09	no	**5. sentiment**	**yes**
	6. medline	no	**6. product_review_mining**	**yes**
	7. eit2	no	**7. sentiment_classification**	**yes**
	8. l2r	no	8. 31july09	no
	9. exploration	no	**9. opinion_mining**	**yes**
	10. biomedical	no	**10. product**	**yes**

Precision: 10% VS 60%

Case Study 2: Tagging Movies (SDAE)

An example movie with recommended tags

Example Movie	Title: E.T. the Extra-Terrestrial	
	Top topic 1: crew, must, on, earth, human, save, ship, rescue, by, find, scientist, planet	
Top 10 recommended tags	SDAE	True tag?
	1. Saturn Award (Best Special Effects)	yes
	2. Want	no
	3. Saturn Award (Best Fantasy Film)	no
	4. Saturn Award (Best Writing)	yes
	5. Cool but freaky	no
	6. Saturn Award (Best Director)	no
	7. Oscar (Best Editing)	no
	8. almost favorite	no
	9. Steven Spielberg	yes
	10. sequel better than original	no

Precision: 30% VS 60%

Case Study 2: Tagging Movies (RSDAE)

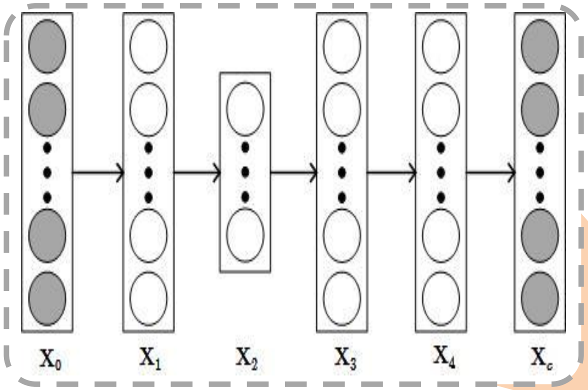
An example movie with recommended tags

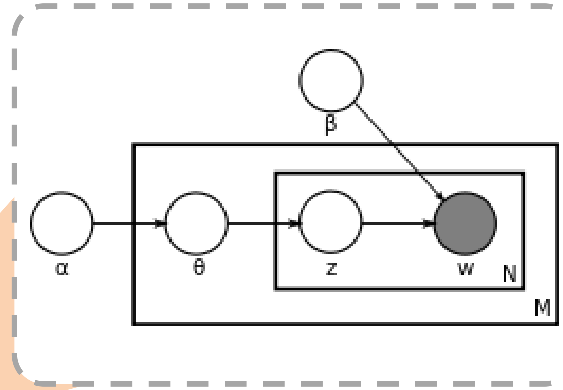
Example Movie	Title: E.T. the Extra-Terrestrial	
	Top topic 1: crew, must, on, earth, human, save, ship, rescue, by, find, scientist, planet	
Top 10 recommended tags	RSDAE	True tag?
	1. Steven Spielberg	yes
	2. Saturn Award (Best Special Effects)	yes
	3. Saturn Award (Best Writing)	yes
	4. Oscar (Best Editing)	no
	5. Want	no
	6. Liam Neeson	no
	7. AFI 100 (Cheers)	yes
	8. Oscar (Best Sound)	yes
	9. Saturn Award (Best Director)	no
	10. Oscar (Best Music - Original Score)	yes

Does not appear in the tag lists of movies linked to ‘E.T. the Extra-Terrestrial’

Very difficult to discover this tag

Relational SDAE as Deep Relational Topic Models





Perception component

Topic hierarchy

Task-Specific component

Inter-document relation

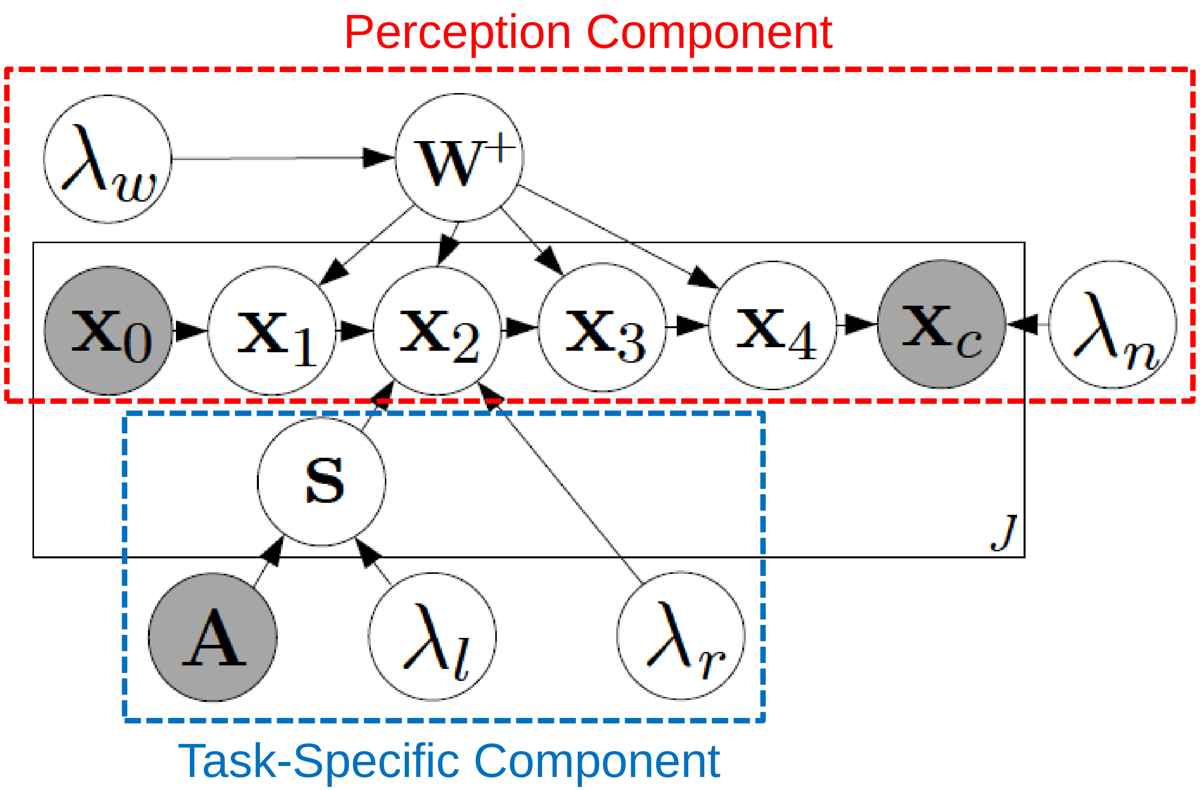
BDL-Based Topic Models

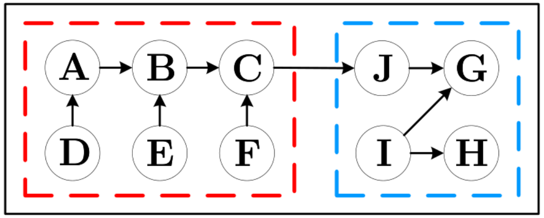
Unified into a probabilistic relational model

for relational deep learning

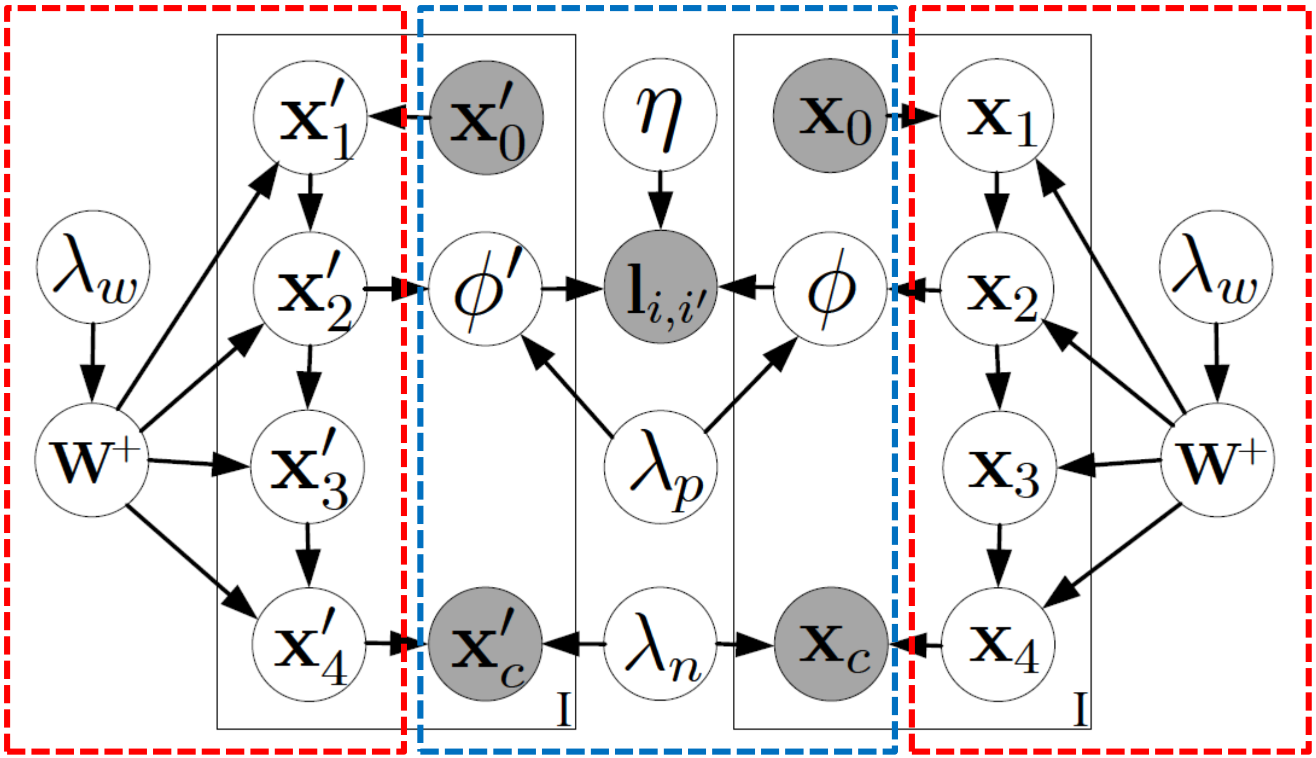
[Wang et al. 2015 (AAAI)]

(Recap) Relational SDAE: Two Components





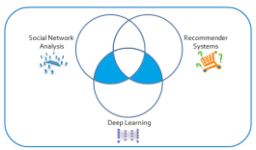
Using Relational Information as Observations



Probabilistic SDAE

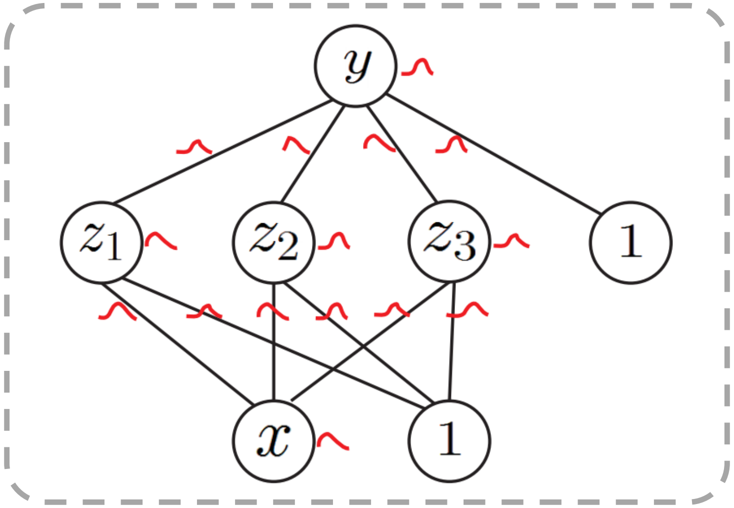
Modeling relation among nodes

[Wang et al. 2017 (AAAI)]



Be ‘Bayesian’ in Collaborative Deep Learning

Be Bayesian in BDL

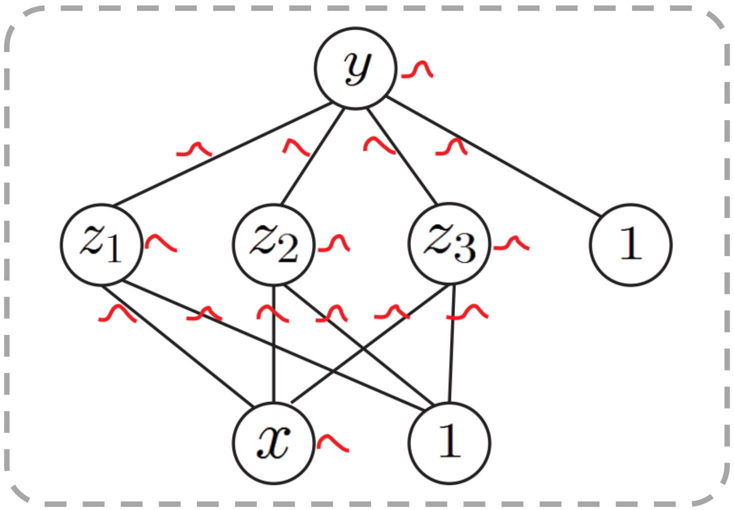


"Natural-Parameter Networks: A Class of Probabilistic Neural Networks"

Motivation:

- Uncertainty estimation for reinforcement learning, active learning, etc.
- Robust for insufficient data and noise
- More accurate prediction

Be Bayesian in BDL

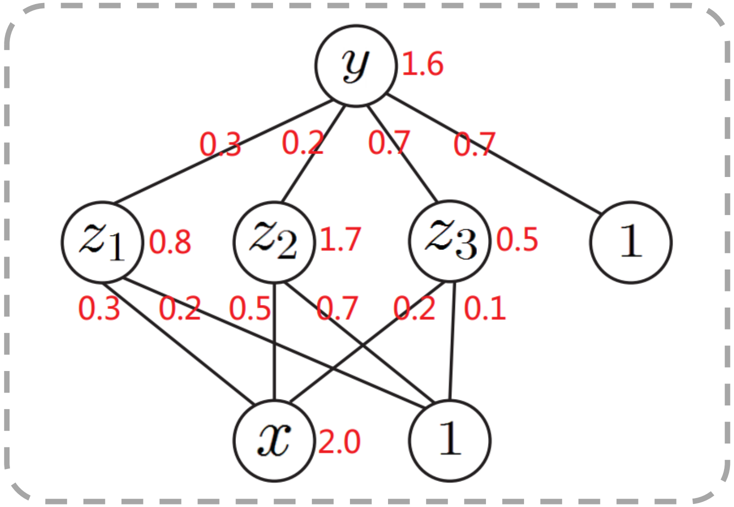


"Natural-Parameter Networks: A Class of Probabilistic Neural Networks"

What We Want:

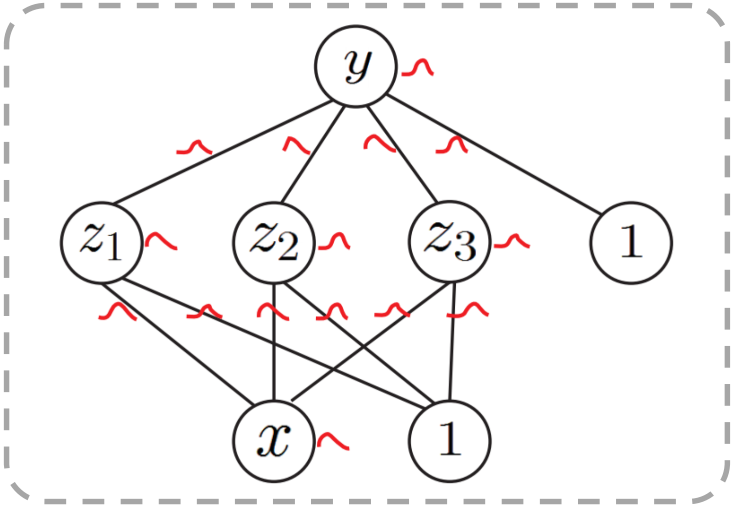
- Solvable via back propagation
- Sampling-free during both training and testing
- Intuitive and easy to implement

Weights/Neurons as Distributions



neural networks

weights/neurons as points



natural-parameter networks

weights/neurons as distributions

Take-home Messages

- Probabilistic graphical models for formulating both representation learning and inference/reasoning components
- Learnable representation serving as a bridge
- Tight, two-way interaction is crucial

Thanks!
Q&A