ICMR 2018 Tutorial
June 11, 2018
Yokohama, Japan

Recommendation Technologies for Multimedia Content

Xiangnan He
National University of Singapore

Hanwang Zhang
Nanyang Technological University

Tat-Seng Chua
National University of Singapore

Slides are available: http://comp.nus.edu.sg/~xiangnan/icmr18-recsys.pdf

Outline of Tutorial

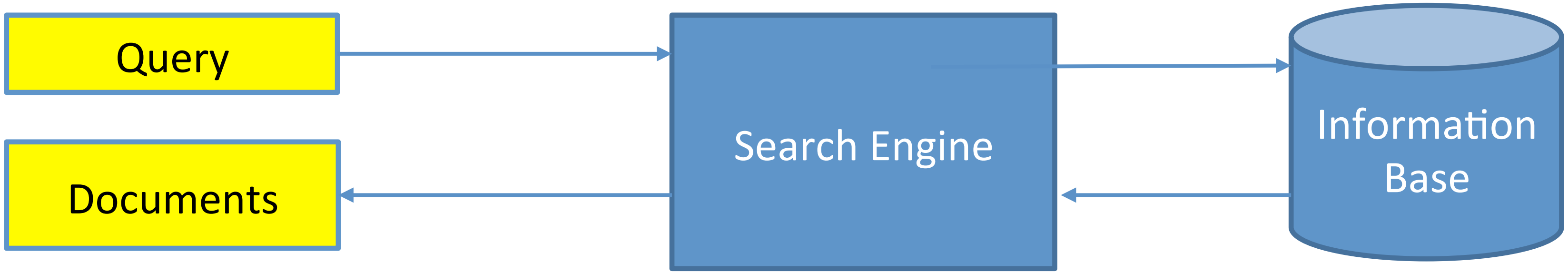
- Background (Xiangnan, 10 mins)
- Basics & Advances in Recommendation (Xiangnan, 50 mins)
- Visually-aware Product Recommendation (Xiangnan, 30 mins)
- Break (15 mins)
- Visual Representation (Hanwang, 45 mins)
- Image/Video Recommendation (Hanwang, 25 mins)
- Conclusion (Hanwang, 5 mins)

Slides: http://comp.nus.edu.sg/~xiangnan/icmr18-recsys.pdf

Retrieval vs. Recommendation

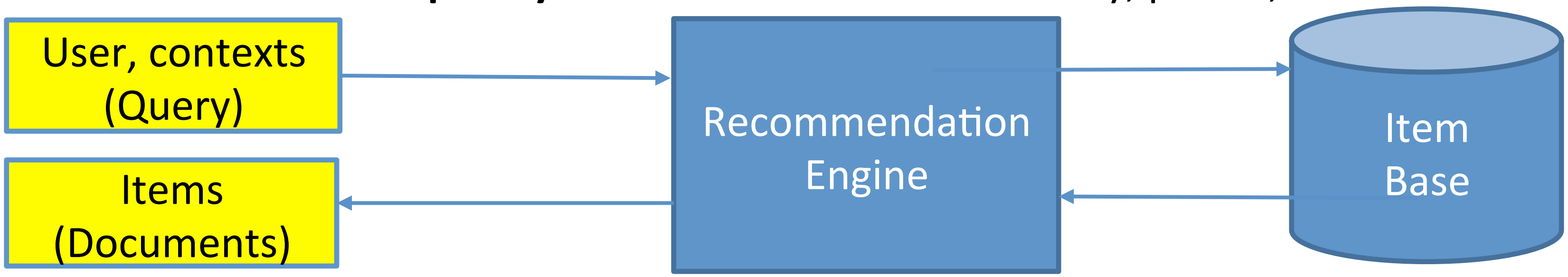
- Retrieval is information pull:
  - User pulls desired information by making a specific request

User intent is **explicitly** reflected in query



- Recommendation is information push:
  - System pushes desired information to a user by guessing her interest

User intent is **implicitly** reflected in interaction history, profile, contexts etc.



Slides: http://comp.nus.edu.sg/~xiangnan/icmr18-recsys.pdf

Importance of Recommendation

- Retrieval mostly exists in search engines
- But, recommendation exists everywhere…
  - When you search for a product <= Ad recommendation

Top result is usually an ad

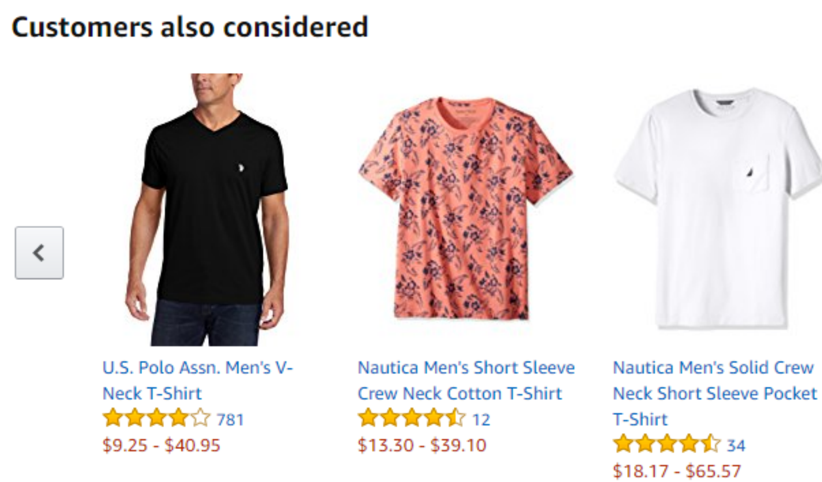
Search results of Taobao

Slides: http://comp.nus.edu.sg/~xiangnan/icmr18-recsys.pdf

Importance of Recommendation

- Retrieval mostly exists in search engines
- But, recommendation exists everywhere…
  - When you search for a product => Ad recommendation
  - When you open a product page => Product recommendation

Besides it:



Screenshot of Amazon

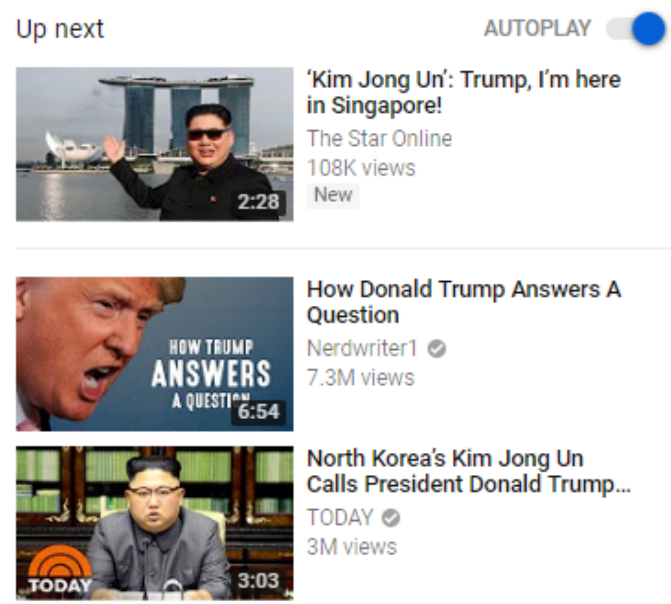
Slides: http://comp.nus.edu.sg/~xiangnan/icmr18-recsys.pdf

Importance of Recommendation

- Retrieval mostly exists in search engines
- But, recommendation exists everywhere…
  - When you search for a product => Ad recommendation
  - When you open a product page => Product recommendation
  - When you watch a video => Video Recommendation

President Trump reads letter from Kim Jong Un (YouTube)

Besides it:



Slides: http://comp.nus.edu.sg/~xiangnan/icmr18-recsys.pdf

Importance of Recommendation

- Retrieval mostly exists in search engines
- But, recommendation exists everywhere…
  - When you search for a product => Ad recommendation
  - When you open a product page => Product recommendation
  - When you watch a video => Video Recommendation
  - When you read a news => News recommendation
  - When you book a flight => Hotel Recommendation
  - When you use social network => Friend Recommendation
  - When you are hungry => Restaurant Recommendation
  - When you open any webpage/app, there maybe a recommendation list.

… …

Slides: http://comp.nus.edu.sg/~xiangnan/icmr18-recsys.pdf

Value of Recommender System (RecSys)

- RecSys has become a major monetization tool for customer-oriented online services
  - E.g., E-commerce, News Portal, Social Networks, etc.
- Ad systems are technically supported by recommendation solutions.
  - The key is Click-Through Rate (CTR) prediction
- Some statistics:
  - YouTube homepage: 60%+ clicks [Davidson et al. 2010]
  - Netflix: 80%+ movie watches, 1billion+ value/year [Gomze-Uribe et al 2016]
  - Amazon: 30%+ page views [Smith and Linden, 2017]

Slides: http://comp.nus.edu.sg/~xiangnan/icmr18-recsys.pdf

Why RecSys + MultiMedia?

- Multimedia contents are prevalent in Web, generated and consumed in a fast speed.
  - YouTube, Pinterest, Snap etc.

  => Need dedicated RecSys for multimedia contents (e.g., images, videos)
- A picture is worth a thousand words
  - Visual signal is crucial to attract users perform an action.
  - Many non-multimedia items are affiliated with image/video for better explanation, e.g., news, products, ads

  => Need enhanced RecSys to incorporate visual signal

Slides: http://comp.nus.edu.sg/~xiangnan/icmr18-recsys.pdf

Some Useful Resources

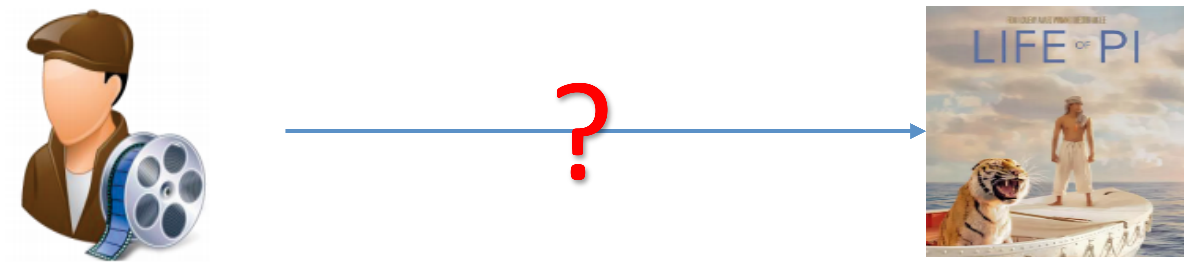
- Recent challenges:
  - MediaEval 2018 on content-based movie recommendation:
    http://www.multimediaeval.org/mediaeval2018/content4recsys/
  - ACM MM Challenge 2018 on social media headline prediction:
    https://social-media-prediction.github.io/PredictionChallenge/
- Datasets:
  - Pinterest images: https://sites.google.com/site/xueatalphabeta/academic-projects
  - Amazon products (with images): http://jmcauley.ucsd.edu/data/amazon/

Outline of Tutorial

- Background (Xiangnan, 10 mins)
- **Basics & Advances in Recommendation (Xiangnan, 50 mins)**
  - **Traditional Shallow Learning Methods**
  - **Recent Deep Learning Methods**
- Visually-aware Product Recommendation (Xiangnan, 30 mins)
- Break (15 mins)
- Visual Representation (Hanwang, 45 mins)
- Visual Recommendation (Hanwang, 25 mins)
- Conclusion (Hanwang, 5 mins)

Problem Formulation

- Recommendation solves a matching problem.



User Profile (query):
- User ID
- Rating history
- Age, Gender
- Income level
- Time of the day

…….

Item Profile (document):
- Item ID
- Description
- Category
- Price
- Image

…….

Challenge: no overlap between user features and item features
Matching can't be done on the superficial feature level!

Slides: http://comp.nus.edu.sg/~xiangnan/icmr18-recsys.pdf

Collaborative Filtering

- Collaborative Filtering (CF) is the most well-known technique for recommendation.

  *"CF makes predictions (**filtering**) about a user's interest by collecting preferences information from many users (**collaborating**)" ---Wikipedia*

- Math formulation: matrix completion problem

User	Movie	Rating
Alice	Titanic	5
Alice	Notting Hill	3
Alice	Star Wars	1
Bob	Star Wars	4
Bob	Star Trek	5
Charlie	Titanic	1
Charlie	Star Wars	5
...	...	...

Input Tabular data

→

User \ Movie	TI	NH	SW	ST	...
A	5	3	1	?	...
B	?	?	4	5	...
C	1	?	5	?	...
...	...	...	...	...	...

Rating Matrix
(Interaction Matrix)

Slides: http://comp.nus.edu.sg/~xiangnan/icmr18-recsys.pdf

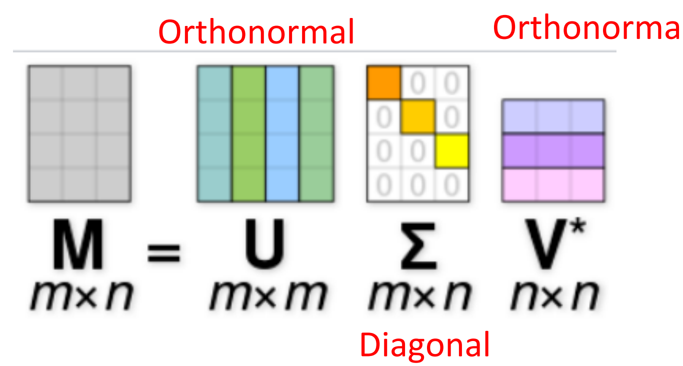
Solving Matrix Completion

- Singular Value Decomposition (SVD) is the most well-known technique for matrix completion

Movie

User	TI	NH	SW	ST	...
A	5	3	1	?	...
B	?	?	4	5	...
C	1	?	5	?	...
...	...	...	...	...	...

Rating Matrix



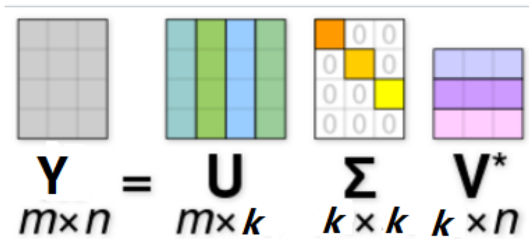
Steps to use SVD for CF:

1. Impute missing data to 0 in **Y**
2. Solving the SVD problem
3. Using only *K* dimensions in **U** and **V** to obtain a low rank model to estimate **Y**

Figure adopted from: https://en.wikipedia.org/wiki/Singular-value_decomposition

Slides: http://comp.nus.edu.sg/~xiangnan/icmr18-recsys.pdf

SVD is Suboptimal for CF



- In essence, SVD is solving the problem:

$$\arg\min_{\mathbf{U},\Sigma,\mathbf{V}} (\mathbf{Y} - \mathbf{U}\Sigma\mathbf{V}^T)^2$$

$$= \arg\min_{\mathbf{U},\Sigma,\mathbf{V}} \sum_{i=1}^{m}\sum_{j=1}^{n} (y_{ij} - (\mathbf{U}\Sigma\mathbf{V}^T)_{ij})^2$$

Label

Model Prediction

Training instance

- Several Implications (weaknesses):
  1. Missing data has the same weight as observed data (>99% sparsity)
  2. No regularization is enforced – easy to overfit

Slides: http://comp.nus.edu.sg/~xiangnan/icmr18-recsys.pdf

Adjust SVD for CF

- The "SVD" model in the context of recommendation:

$$\hat{y}_{ui} = \mathbf{v}_u^T \mathbf{v}_i$$

User latent vector   Item latent vector

- Regularized Loss function:

$$L = \underbrace{\sum_u \sum_i w_{ui}(y_{ui} - \hat{y}_{ui})^2}_{\text{Prediction error}} + \lambda \underbrace{\left(\sum_u ||\mathbf{v}_u||^2 + \sum_i ||\mathbf{v}_i||^2\right)}_{\text{L2 regularizer}}$$

- This method is also called *Matrix Factorization* (MF) in RecSys:
  - It represents a user and an item as a latent vector (ID embedding).
  - The interaction between user and item is modelled using inner product (measure how much user latent "preferences" match with item "properties"
  - Besides L2 regularized loss, other loss can also be used, e.g., cross-entropy, margin-based pairwise loss, etc.

Slides: http://comp.nus.edu.sg/~xiangnan/icmr18-recsys.pdf

Why MF Can Capture CF

Item latent vector: v_i^I

User latent vector: v_u^U

Movie

User	TI	NH	SW	ST	...
A	5	3	1	?	...
B	?	?	4	5	...
C	1	?	5	?	...
...	...	...	...	...	...

Prediction score:

$$\hat{y}_{ui} = \mathbf{v}_u^T \mathbf{v}_i$$

Train the model on all observed interactions by sharing user embedding and item embedding

Latent Embedding space:

Similar users should have similar embeddings

Similar items should have similar embeddings

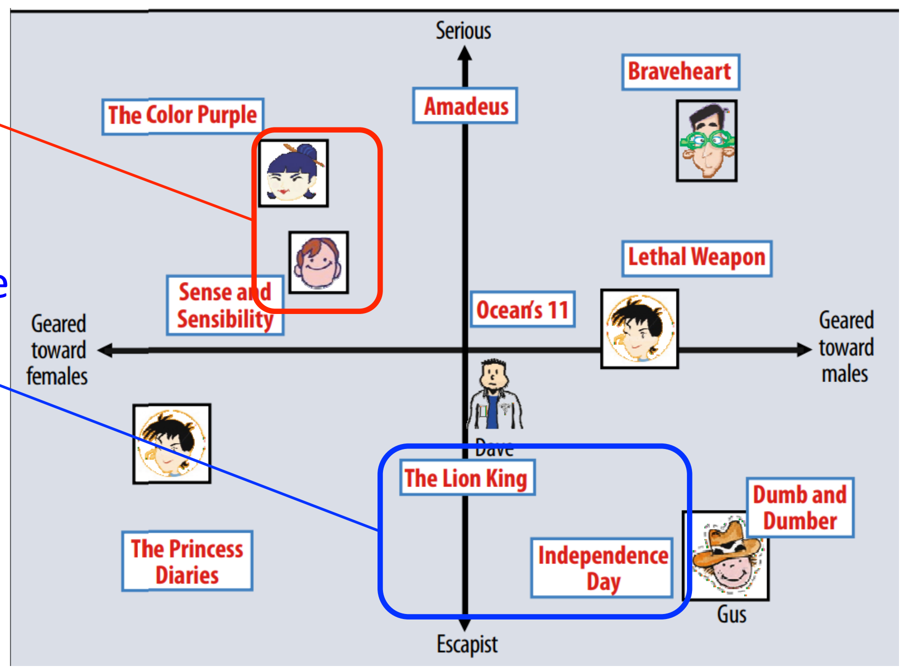
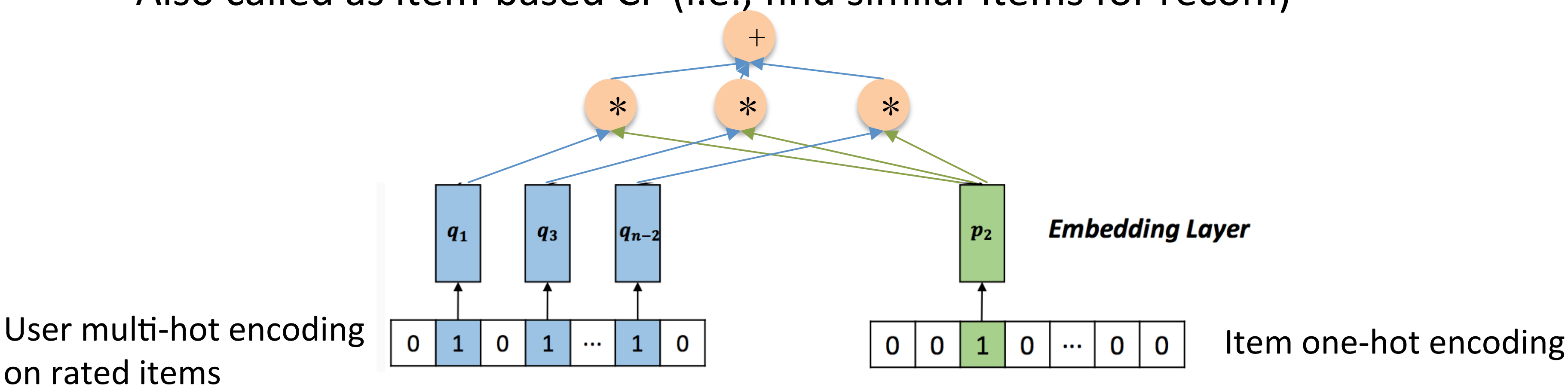


Figure adopted from: https://datajobs.com/data-science-repo/Recommender-Systems-%5BNetflix%5D.pdf

Slides: http://comp.nus.edu.sg/~xiangnan/icmr18-recsys.pdf

Factored Item Similarity Model (Kabbur et al., KDD'14)

- MF encodes a user with an ID, and projects it to embedding.
  - Also called as user-based CF (i.e., find similar users for recom)
- Another more meaningful encoding is to use rated items of the user.
  - Also called as item-based CF (i.e., find similar items for recom)



⇔ user representation

$$\hat{y}_{ui} = (\sum_{j \in \mathcal{R}_u} \mathbf{q}_j)^T \mathbf{v}_i$$

Can be interpreted as the **similarity** between item i and j

Items rated by u

Slides: http://comp.nus.edu.sg/~xiangnan/icmr18-recsys.pdf

SVD++: Fusing User-based and Item-based CF (Koren, KDD'08)

- MF (user-based CF) represents a user as her ID.
  - Directly projecting the ID into latent space
- FISM (item-based CF) represents a user as her interacted items.
  - Projecting interacted items into latent space
- SVD++ fuses the two types of models in the latent space:

$$\hat{y}_{ui} = \underbrace{(\mathbf{v}_u + \sum_{j \in \mathcal{R}_u} \mathbf{q}_j)}_{\text{User representation in latent space}}^T \mathbf{v}_i$$

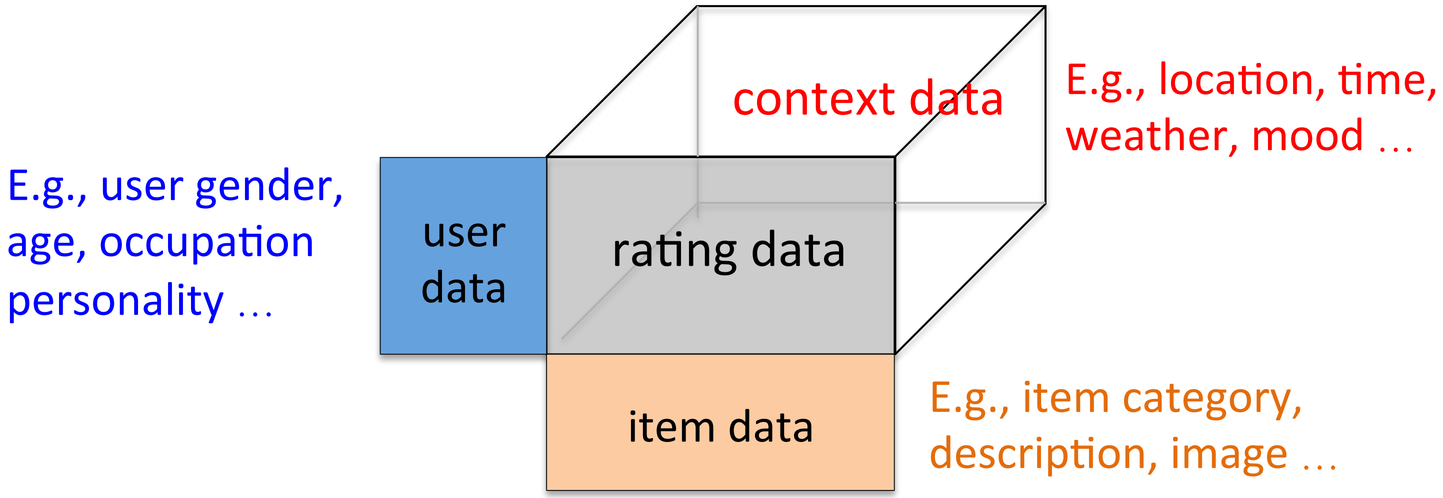
  - This is the best single model for rating prediction in the Netflix challenge.

Note: the normalization terms are discarded for clarity.

Slides: http://comp.nus.edu.sg/~xiangnan/icmr18-recsys.pdf

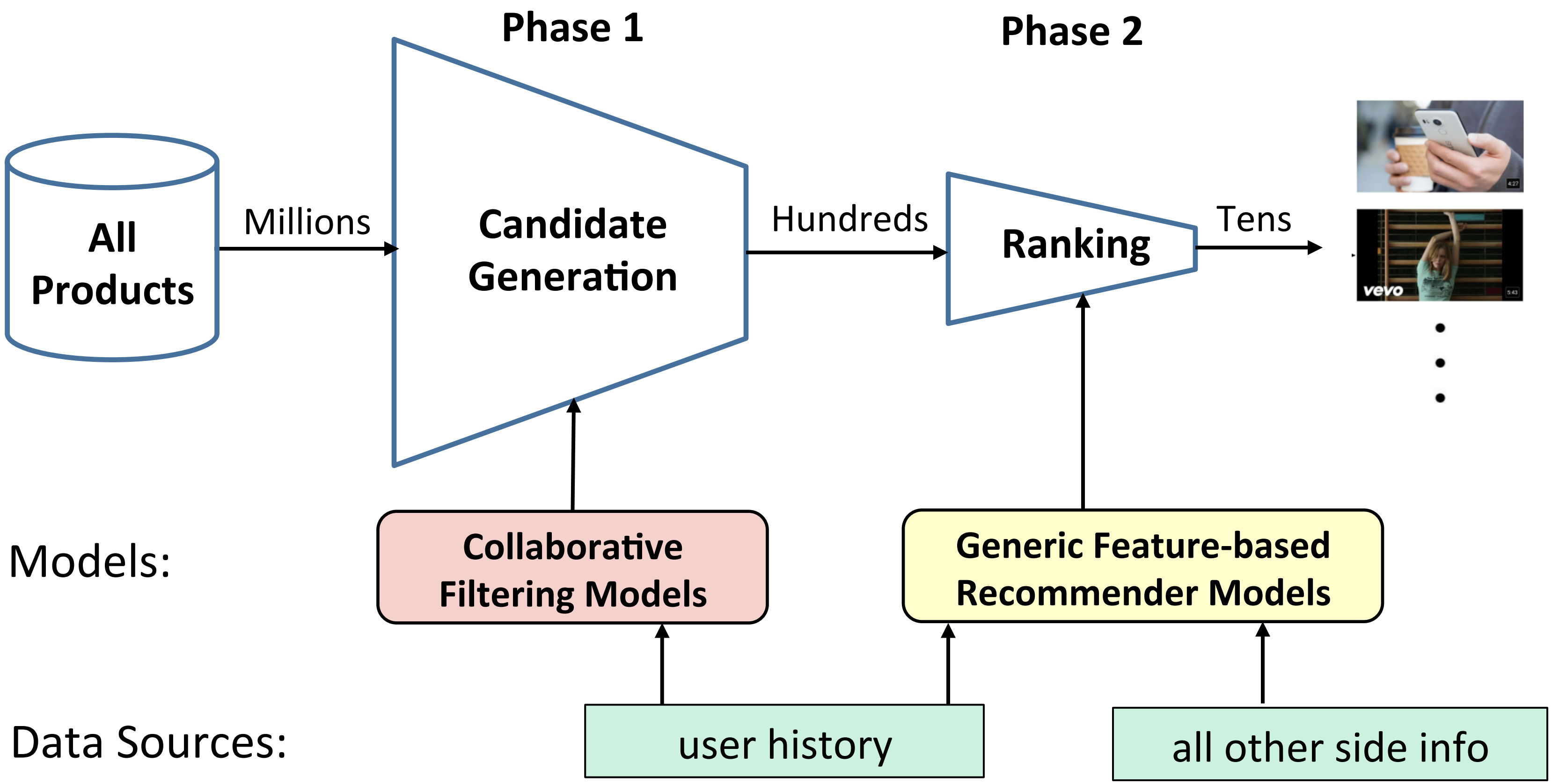
Generic Feature-based Recommendation

- CF utilizes only the interaction matrix only to build the predictive model.
- How about other information like user/item attributes and contexts?
- Example data used for building a RecSys:



Slides: http://comp.nus.edu.sg/~xiangnan/icmr18-recsys.pdf

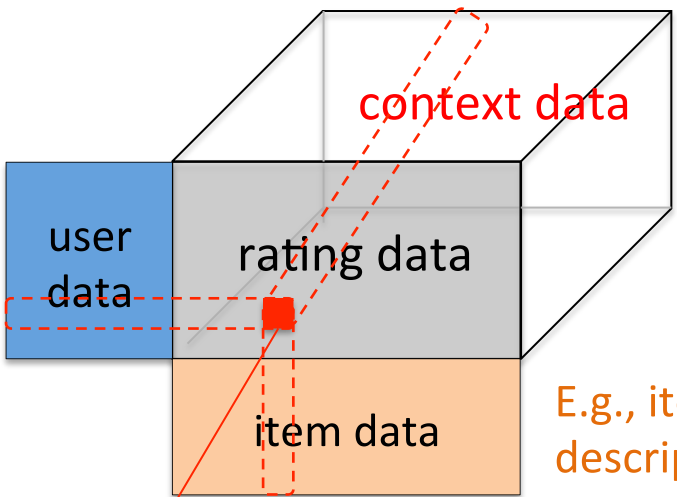
Morden RecSys Architecture



Slides: http://comp.nus.edu.sg/~xiangnan/icmr18-recsys.pdf

Generic Feature-based Recommendation

E.g., user gender, age, occupation personality …



E.g., recent history, location, time, weather, mood …

E.g., item category, description, image …

One-hot encoding

Input Features:
1. Categorical features:
user/item ID, bag-of-words, historical features…
2. Numerical features:
textual/visual embeddings, converted features (e.g. TFIDF, GBDT)…

Each row encodes all info for a rating

Feature vector **x**

	User				Movie					Other Movies rated					Target y	
	A	B	C	...	TI	NH	SW	ST	...	TI	NH	SW	ST	...		
$\mathbf{x}^{(1)}$	1	0	0	...	1	0	0	0	...	0.3	0.3	0.3	0	...	5	$y^{(1)}$
$\mathbf{x}^{(2)}$	1	0	0	...	0	1	0	0	...	0.3	0.3	0.3	0	...	3	$y^{(2)}$
$\mathbf{x}^{(3)}$	1	0	0	...	0	0	1	0	...	0.3	0.3	0.3	0	...	1	$y^{(3)}$
$\mathbf{x}^{(4)}$	0	1	0	...	0	0	1	0	...	0	0	0.5	0.5	...	4	$y^{(4)}$
$\mathbf{x}^{(5)}$	0	1	0	...	0	0	0	1	...	0	0	0.5	0.5	...	5	$y^{(5)}$
$\mathbf{x}^{(6)}$	0	0	1	...	1	0	0	0	...	0.5	0	0.5	0	...	1	$y^{(6)}$
$\mathbf{x}^{(7)}$	0	0	1	...	0	0	1	0	...	0.5	0	0.5	0	...	5	$y^{(7)}$

Sparse Predictive Model

Slides: http://comp.nus.edu.sg/~xiangnan/icmr18-recsys.pdf

FM: Factorization Machine (Rendle, ICDM'10)

- FM is inspired from previous factorization models
- It represents each feature an embedding vector, and models the second-order feature interactions:

Only nonzero features are considered

$$\hat{y}(\mathbf{x}) = w_0 + \sum_{i=1}^{p} w_i x_i + \sum_{i=1}^{p} \sum_{j>i}^{p} < \mathbf{v}_i, \mathbf{v}_j > \boxed{x_i x_j}$$

First-order: Linear Regression

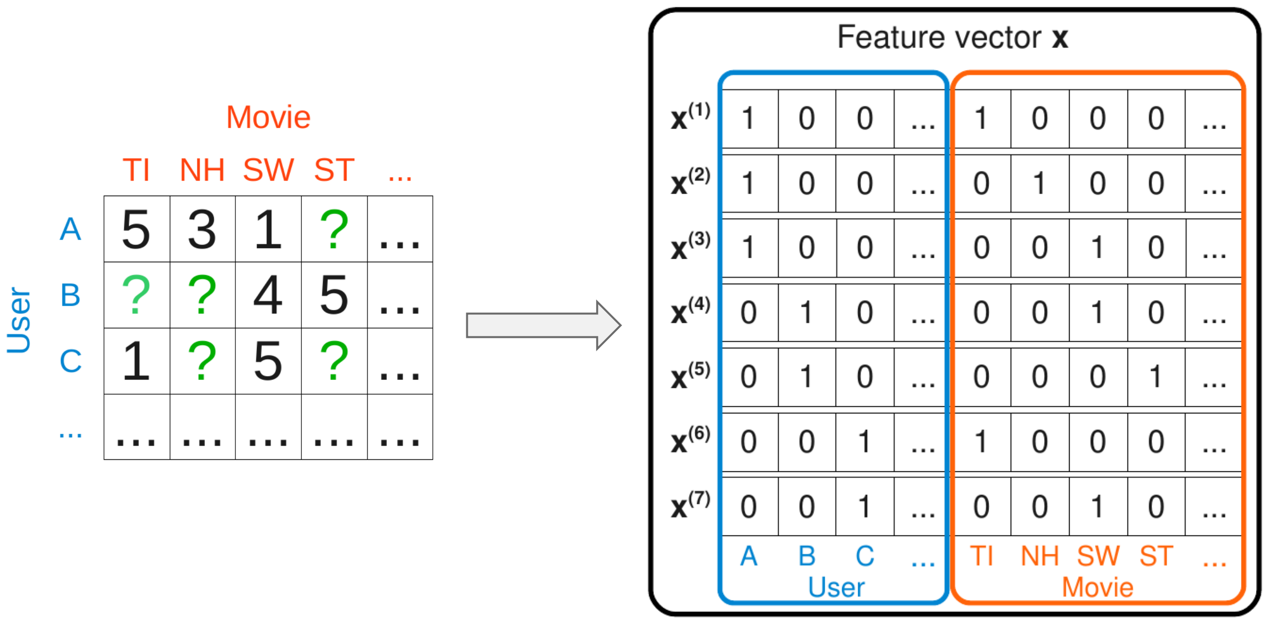
Second-order: pair-wise interactions between features

  - Note: self-interaction is not included: ~~$< \mathbf{v}_i, \mathbf{v}_i >$~~.
- FM allows easy feature engineering for recommendation, and can mimic many existing models (that are designed for a specific task) by inputting different features.
  - E.g., MF, SVD++, timeSVD (Koren, KDD'09), PITF (Rendle, WSDM'10) etc.

Slides: http://comp.nus.edu.sg/~xiangnan/icmr18-recsys.pdf

Matrix Factorization with FM

- Input: 2 variables <user (ID), item (ID)>.



With this input, FM is identical to MF with bias:

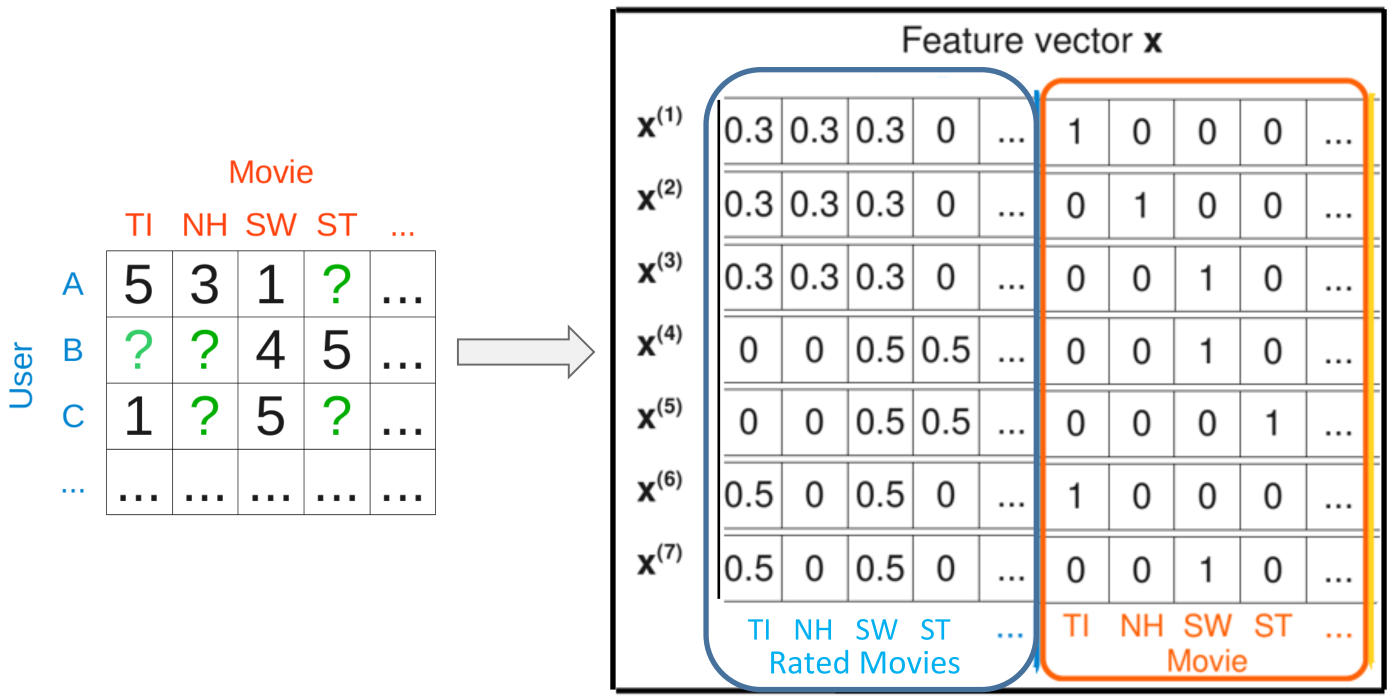
$$\hat{y}(\mathbf{x}) = w_0 + w_u + w_i + \underbrace{< \mathbf{v}_u, \mathbf{v}_i >}_{\text{MF}}$$

MF

Slides: http://comp.nus.edu.sg/~xiangnan/icmr18-recsys.pdf

Factored Item Similarity Model with FM

- Input: 2 variables <user (historical items ID), item (ID)>.



Further input user ID into FM will resume SVD++

With this input, FM subsumes FISM with additional terms:

$$\hat{y}(\mathbf{x}) = bias + \underbrace{\sum_{j \in \mathcal{R}_u} < \mathbf{v}_j, \mathbf{v}_i >}_{\text{FISM}} + \sum_{j \in \mathcal{R}_u, j' > j} < \mathbf{v}_j, \mathbf{v}_{j'} >$$

FISM

Slides: http://comp.nus.edu.sg/~xiangnan/icmr18-recsys.pdf

Explicit Feedback vs. Implicit Feedback

Explicit Feedback

Movie

User	TI	NH	SW	ST	...
A	5	3	1	?	...
B	?	?	4	5	...
C	1	?	5	?	...
...	...	...	...	...	...

Ratings

Implicit Feedback

Movie

User	TI	NH	SW	ST	...
A	1	1	1	?	...
B	?	?	1	1	...
C	1	?	1	?	...
...	...	...	...	...	...

Watches, Clicks, Purchases ...

Explicit Feedback conveys user preference explictly:

- Higher scores carry positive signal
- Lower scores carry negative signal

Implicit Feedback conveys user preference implicitly:

- Observed interactions do not mean positive signal
- Unobserved interactions do not mean negative signal

Slides: http://comp.nus.edu.sg/~xiangnan/icmr18-recsys.pdf

Rating Prediction vs. Ranking

- Old work on recommendation optimize L2 loss:

$$L = \sum_u \sum_i w_{ui}(y_{ui} - \hat{y}_{ui})^2 + \lambda(\sum_u ||\mathbf{v}_u||^2 + \sum_i ||\mathbf{v}_i||^2)$$

  - But many empirical evidence show that:

    A lower error rate does not lead to a good ranking performance…

  - Possible Reasons:
    1) Discrepancy between error measure (e.g., RMSE) and ranking measure.
    2) Observation bias – users tend to rate on the items they like.

- Modern work on recommendation optimize pairwise ranking loss:

sigmoid

Positive prediction Negative prediction

$$L_{BPR} = \arg\max_{\Theta} \sum_{(u,i,j) \in \mathcal{R}_B} \ln \sigma(\hat{y}_{ui} - \hat{y}_{uj}) - \lambda ||\Theta||^2$$

Pairwise training examples: u prefers i over j

  - Known as the *Bayesian Personalized Ranking* loss (Rendle, UAI'09).
  - It optimizes relative ranking between two items, rather than absolute scores

Slides: http://comp.nus.edu.sg/~xiangnan/icmr18-recsys.pdf

References

- https://en.wikipedia.org/wiki/Collaborative_filtering
- Xiangnan He, Hanwang Zhang, Min-Yen Kan, and Tat-Seng Chua. Fast matrix factorization for online recommendation with implicit feedback. In SIGIR 2016.
- Yehuda Koren, and Robert Bell. Advances in collaborative filtering. Recommender systems handbook. Springer, Boston, MA, 2015. 77-118.
- Santosh Kabbur, Xia Ning, and George Karypis. Fism: factored item similarity models for top-n recommender systems. In KDD 2013.
- Yehuda Koren. Factorization meets the neighborhood: a multifaceted collaborative filtering model. In KDD 2018.
- Steffen Rendle. Factorization machines. In ICDM 2010.
- Yehuda Koren. Collaborative filtering with temporal dynamics. Communications of the ACM 53, no. 4 (2010): 89-97.
- Steffen Rendle, and Lars Schmidt-Thieme. Pairwise interaction tensor factorization for personalized tag recommendation. In WSDM 2010.
- Immanuel Bayer, Xiangnan He, Bhargav Kanagal, and Steffen Rendle. A generic coordinate descent framework for learning from implicit feedback. In WWW 2017.
- Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. Neural collaborative filtering. In WWW 2017.
- Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. BPR: Bayesian personalized ranking from implicit feedback. In UAI 2009.

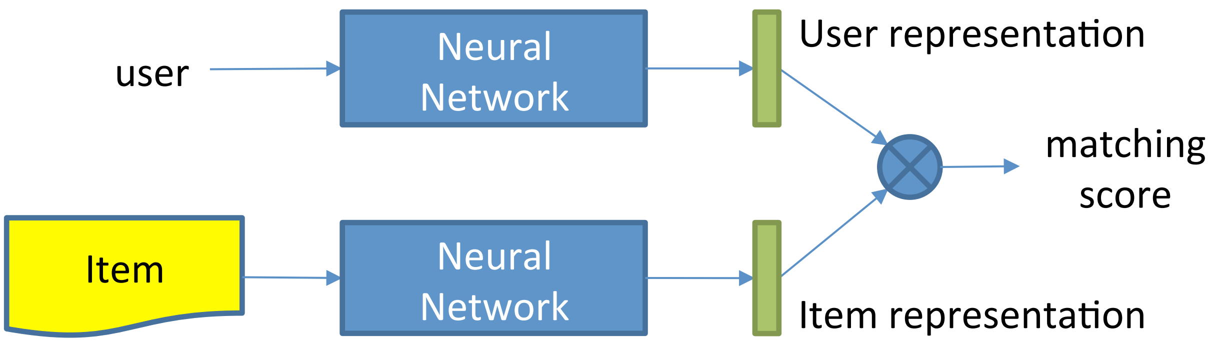
Outline of Tutorial

- Background (Xiangnan, 10 mins)
- Basics & **Advances in Recommendation (Xiangnan, 50 mins)**
  - Traditional Shallow Learning Methods
  - **Recent Deep Learning Methods**
    - **1) Deep Collaborative Filtering methods**
    - **2) Deep Feature-based Recommender Models**
- Visually-aware Product Recommendation (Xiangnan, 30 mins)
- Break (15 mins)
- Visual Representation (Hanwang, 45 mins)
- Image/Video Recommendation (Hanwang, 25 mins)
- Summary (Hanwang, 5 mins)

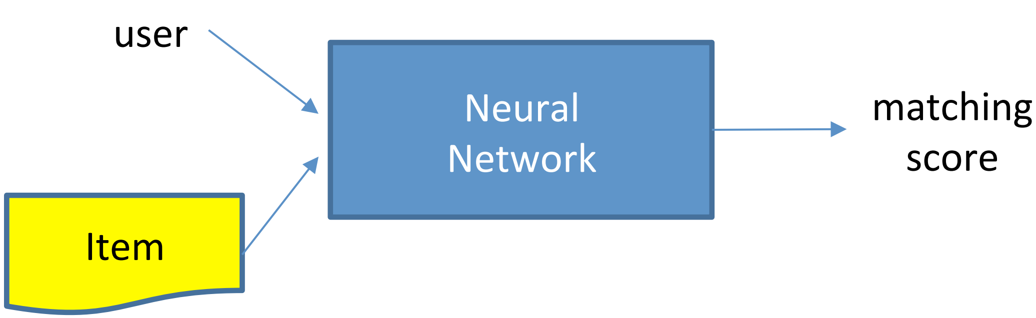
Slides: http://comp.nus.edu.sg/~xiangnan/icmr18-recsys.pdf

Deep Learning Models for CF

- Methods of representation learning



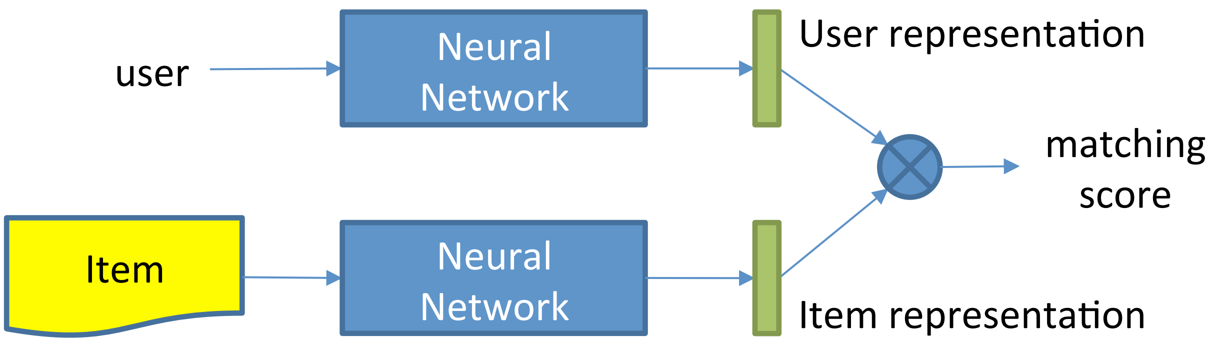
- Methods of matching function learning



Slides: http://comp.nus.edu.sg/~xiangnan/icmr18-recsys.pdf

Next …

- Methods of representation learning



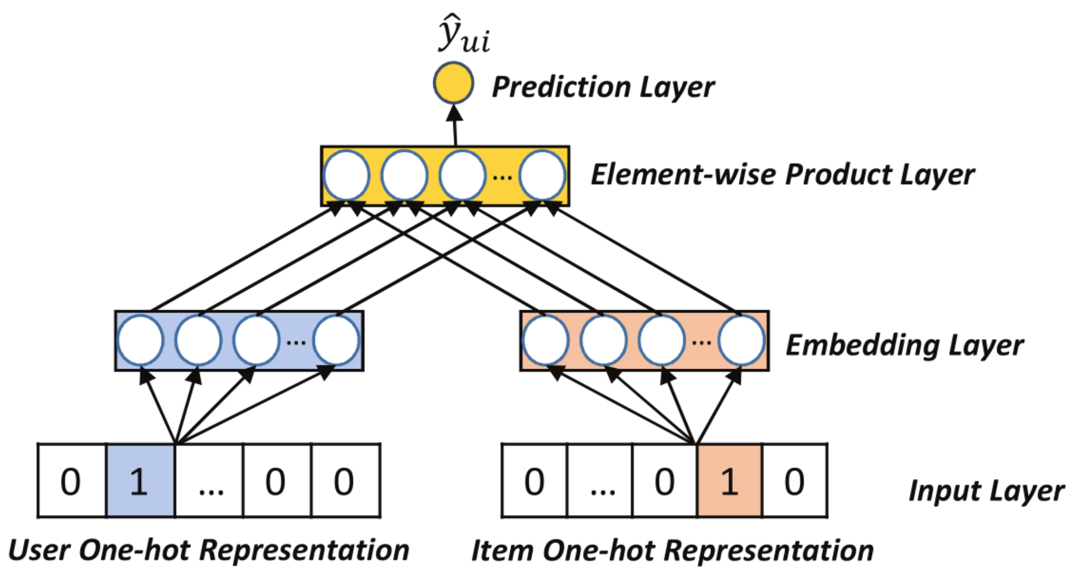
- **DeepMF**: Deep Matrix Factorization (Xue et al, IJCAI'17)
- **AutoRec**: Autoencoders Meeting CF (Sedhain et al, WWW'15)
- **CDAE**: Collaborative Denoising Autoencoder (Wu et al, WSDM'16)

Slides: http://comp.nus.edu.sg/~xiangnan/icmr18-recsys.pdf

Matrix Factorization as a Neural Network (Wang et al, SIGIR'17)

- **Input**: user -> ID (one-hot), item -> ID (one-hot).
- **Representation Function**: linear embedding layer.
- **Matching Function**: inner product.

$$f_{MF}(u, i|\mathbf{p}_u, \mathbf{q}_i) = \mathbf{p}_u^\top \mathbf{q}_i = \sum_{k=1}^{K} p_{uk} q_{ik},$$



Slides: http://comp.nus.edu.sg/~xiangnan/icmr18-recsys.pdf

Deep Matrix Factorization (Xue et al, IJCAI'17)

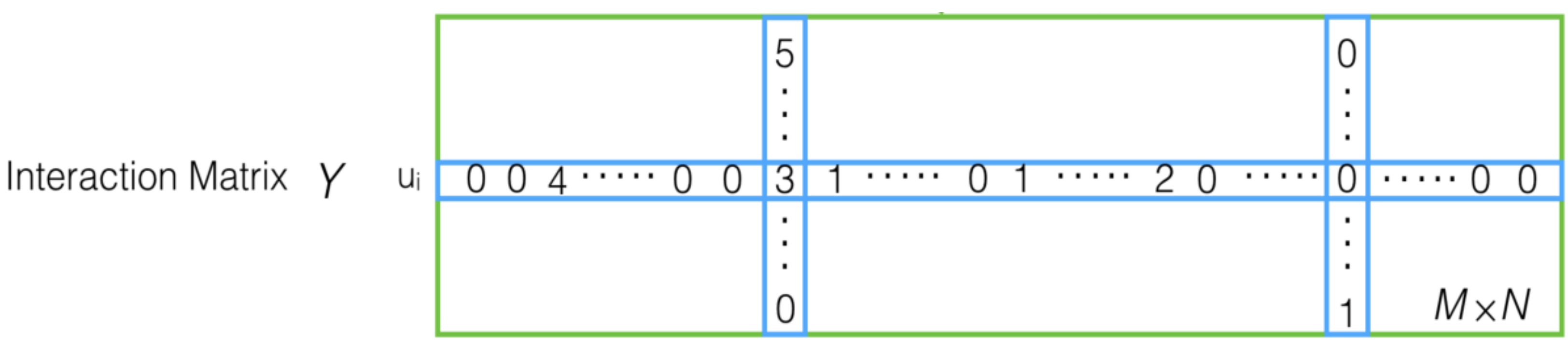
- **Input**:

  user -> items that she has rated (multi-hot), i.e., row vector of Y

  indicates the user's global preference

  item -> users who have rated it (multi-hot), i.e., column vector of Y

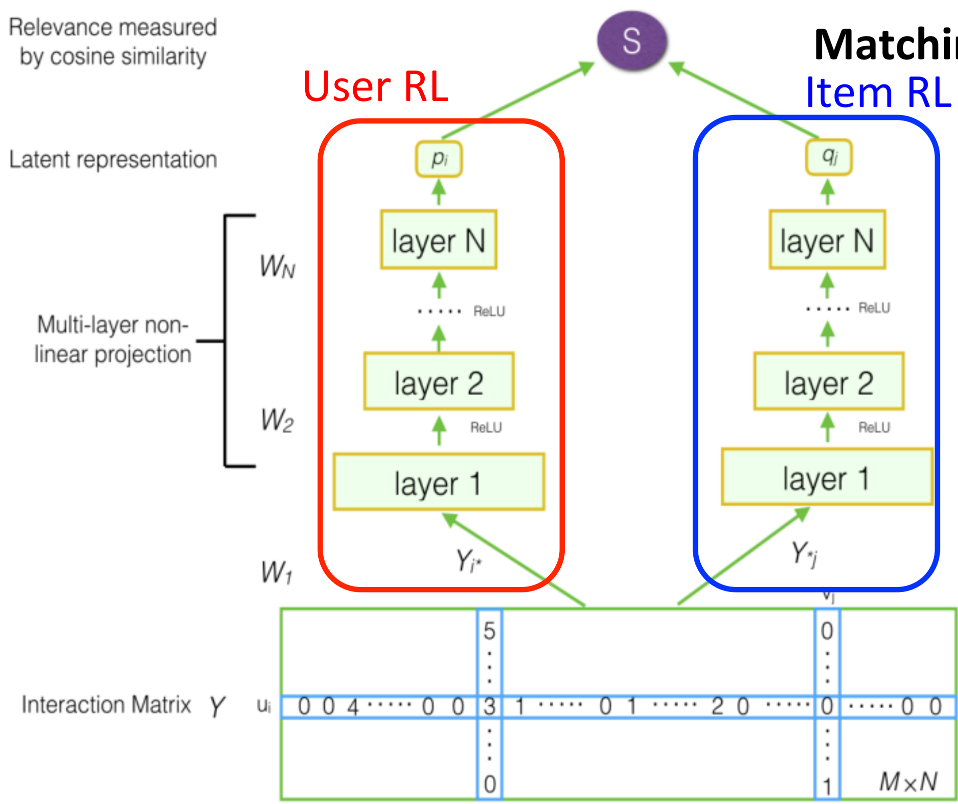
  indicates the item's rating profile.



Slides: http://comp.nus.edu.sg/~xiangnan/icmr18-recsys.pdf

Deep Matrix Factorization (Xue et al, IJCAI'17)

- **Representation Function**:
  - Multi-Layer Perceptron



Matching Function: cosine similarity

$$cosine(p_i, q_j) = \frac{p_i^T q_j}{||p_i|| \, ||q_j||}$$

Slides: http://comp.nus.edu.sg/~xiangnan/icmr18-recsys.pdf

AutoRec (Sedhain et al, WWW'15)

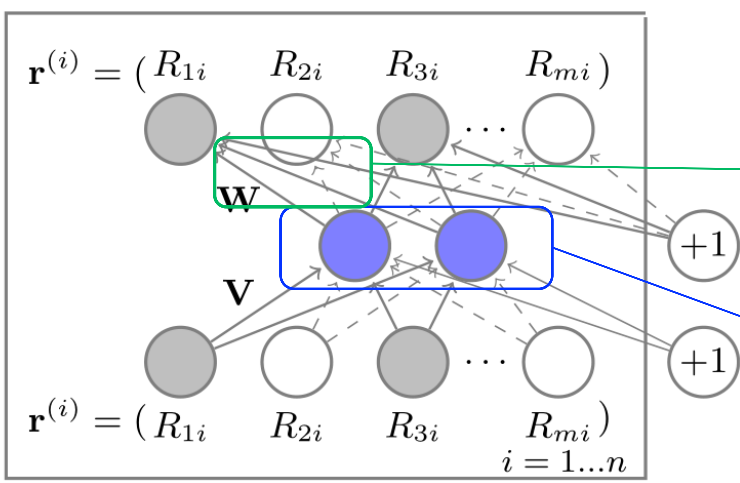
- **Input: (similar to DeepMF)**

  user -> historically rated items (user-based autoencoder).

  item-> ID
- **Representation Function:** Multi-Layer Perceptron
- **Matching Function**: inner product

Input reconstruction: $h(\mathbf{r};\theta) = f\left(\mathbf{W} \cdot g(\mathbf{V}\mathbf{r} + \boldsymbol{\mu}) + \mathbf{b}\right)$

$$\min_{\theta} \sum_{i=1}^{n} ||\mathbf{r}^{(i)} - h(\mathbf{r}^{(i)};\theta))||_{\mathcal{O}}^{2} + \frac{\lambda}{2} \cdot (||\mathbf{W}||_F^2 + ||\mathbf{V}||_F^2),$$



Output weights denote item representation

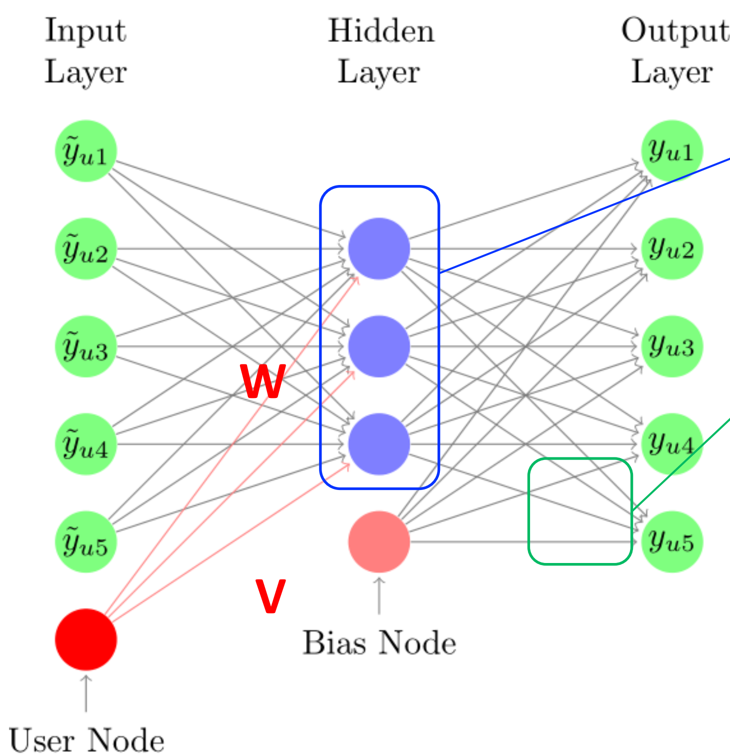
Hidden neurons denote user representation

user-based autoencoder

Slides: http://comp.nus.edu.sg/~xiangnan/icmr18-recsys.pdf

Collaborative Denoising Auto-Encoder (Wu et al, WSDM'16)

- **Input:**
  user -> ID & historically rated items (similar to SVD++)
  item -> ID
- **Representation Function**: Multi-Layer Perceptron



Hidden neurons are user representation:

$$z_u = h\left(\boldsymbol{W}^\top \tilde{\boldsymbol{y}}_u + \boldsymbol{V}_u + \boldsymbol{b}\right),$$

Weights of output layer are item representation

$$\hat{y}_{ui} = f\left(\boldsymbol{W}_i'^\top \boldsymbol{z}_u + b_i'\right),$$

Slides: http://comp.nus.edu.sg/~xiangnan/icmr18-recsys.pdf

Short Summary

- Either ID or history is used as the profile of user/item
- Models with history as input are more expressive, but are also more expensive to train.

- The Auto-Encoder architecture is essentially identical to

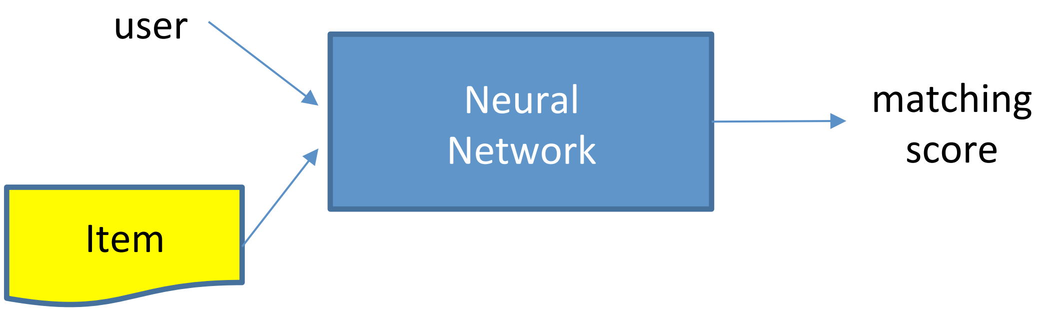
  MLP (representation learning) + MF (matching function).

  Nonlinear (MLP)   Linear (MF)

Slides: http://comp.nus.edu.sg/~xiangnan/icmr18-recsys.pdf

Next...

- Methods of matching function learning:



  - Based on **Neural Collaborative Filtering (NCF)** framework:
    - **NeuMF**: Neural Matrix Factorization (He et al, WWW'17)
    - **NNCF**: Neighbor-based NCF (Bai et al, CIKM'17)
    - **ConvNCF**: Convolutional NCF (He et al, IJCAI'18)

Slides: http://comp.nus.edu.sg/~xiangnan/icmr18-recsys.pdf

Why Using Neural Networks to Learn the Matching Function?

- The simple choice of inner product can limit the ***expressiveness*** of an embedding-based matching model.

$$\hat{y}_{ui} = \mathbf{U}_i^T \mathbf{V}_j \simeq cos(\mathbf{U}_i, \mathbf{V}_j)$$

(E.g., assuming a unit length)

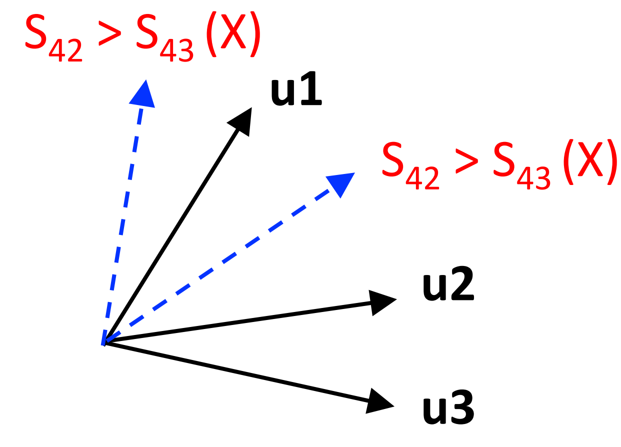
- Example:

	i_1	i_2	i_3	i_4	i_5
u_1	1	1	1	0	1
u_2	0	1	1	0	0
u_3	0	1	1	1	0
u_4	1	0	1	1	1

sim(u1, u2) = 0.5

sim(u3, u1) = 0.4
sim(u3, u2) = 0.66

sim(u4, u1) = 0.6 *****
sim(u4, u2) = 0.2 *
sim(u4, u3) = 0.4 ***



Jaccard Similarity: $s_{ij} = \frac{|\mathcal{R}_i| \cap |\mathcal{R}_j|}{|\mathcal{R}_i| \cup |\mathcal{R}_j|}$

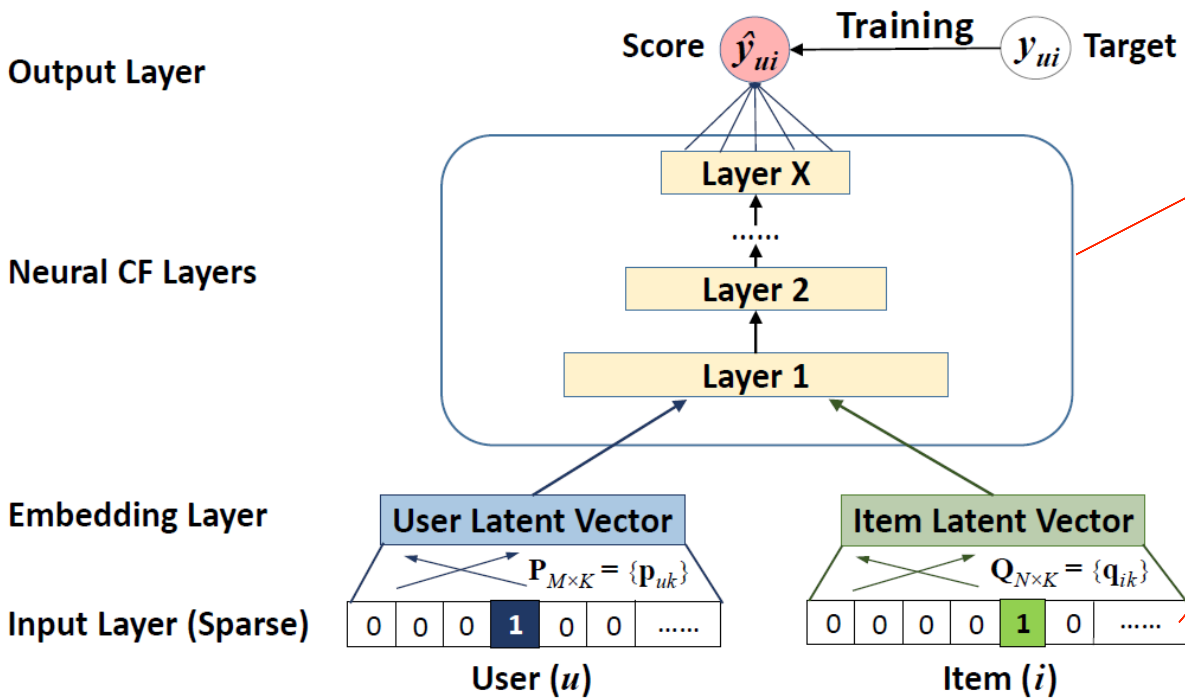
(He et al, WWW'17)

Slides: http://comp.nus.edu.sg/~xiangnan/icmr18-recsys.pdf

Neural Collaborative Filtering Framework (He et al, WWW'17)

- NCF is a general framework that replaces the inner product with a neural network to learn the matching function. $\hat{y}_{ui} = f(\mathbf{p}_u, \mathbf{q}_i)$

NN-based Matching function



Matching function: design whatever layers as you like.

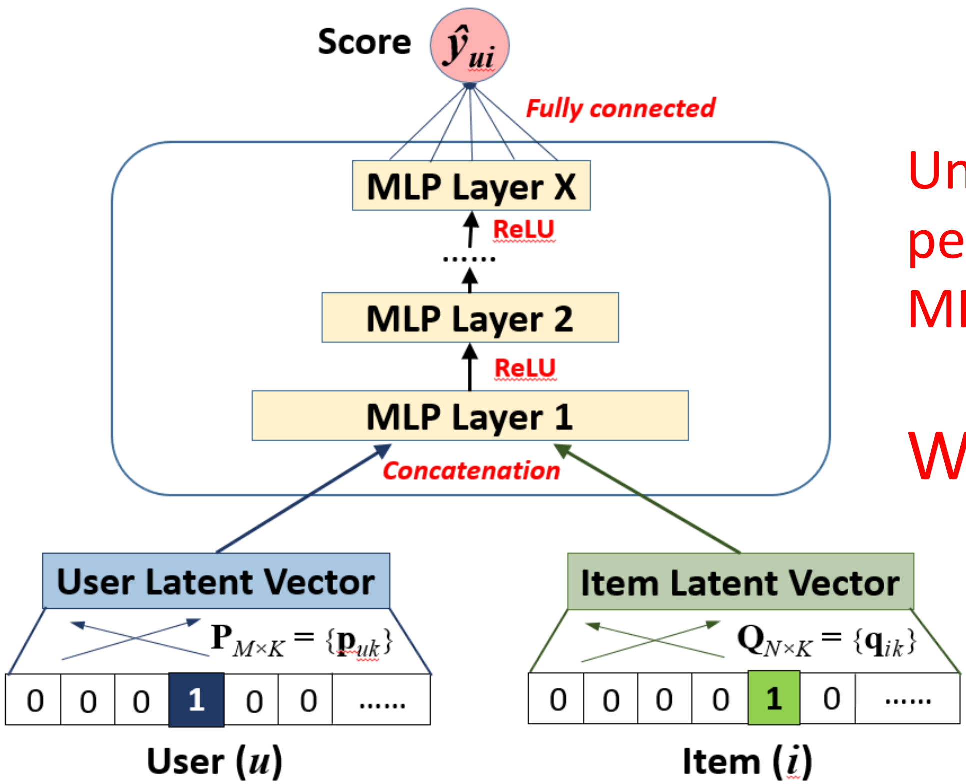
Input: design whatever features as you want.

Figure 2: Neural collaborative filtering framework

Slides: http://comp.nus.edu.sg/~xiangnan/icmr18-recsys.pdf

Multi-Layer Perceptron for CF

- The most intuitive idea is to use a Multi-Layer Perceptron as the matching function.



Unfortunately, MLP doesn't perform well and underperforms MF.

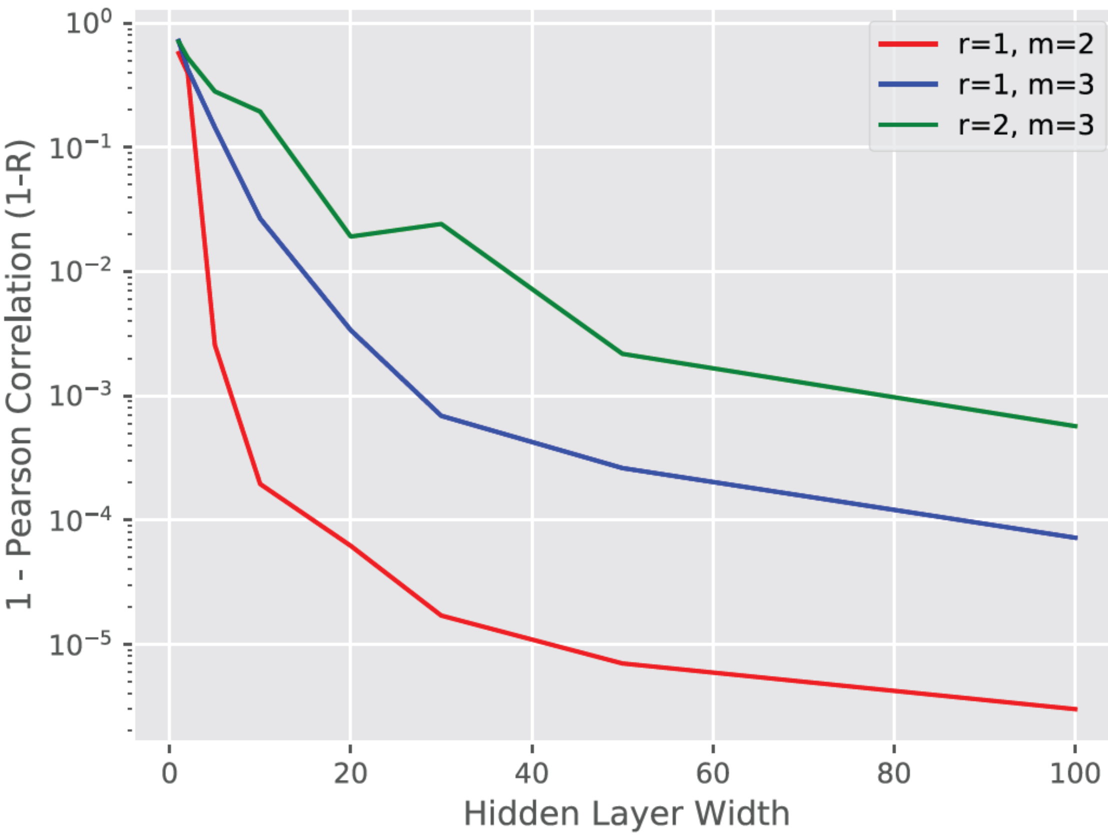
Why?

(He et al, WWW'17)

Slides: http://comp.nus.edu.sg/~xiangnan/icmr18-recsys.pdf

MLP is Weak in Capturing Low-Rank Relation (Beutel et al, WSDM'18)

Setting: Generating low-rank data, and using one-layer MLP to fit it.
r: rank size; m: data dimension (2 -> matrix; 3 -> 3D tensor).



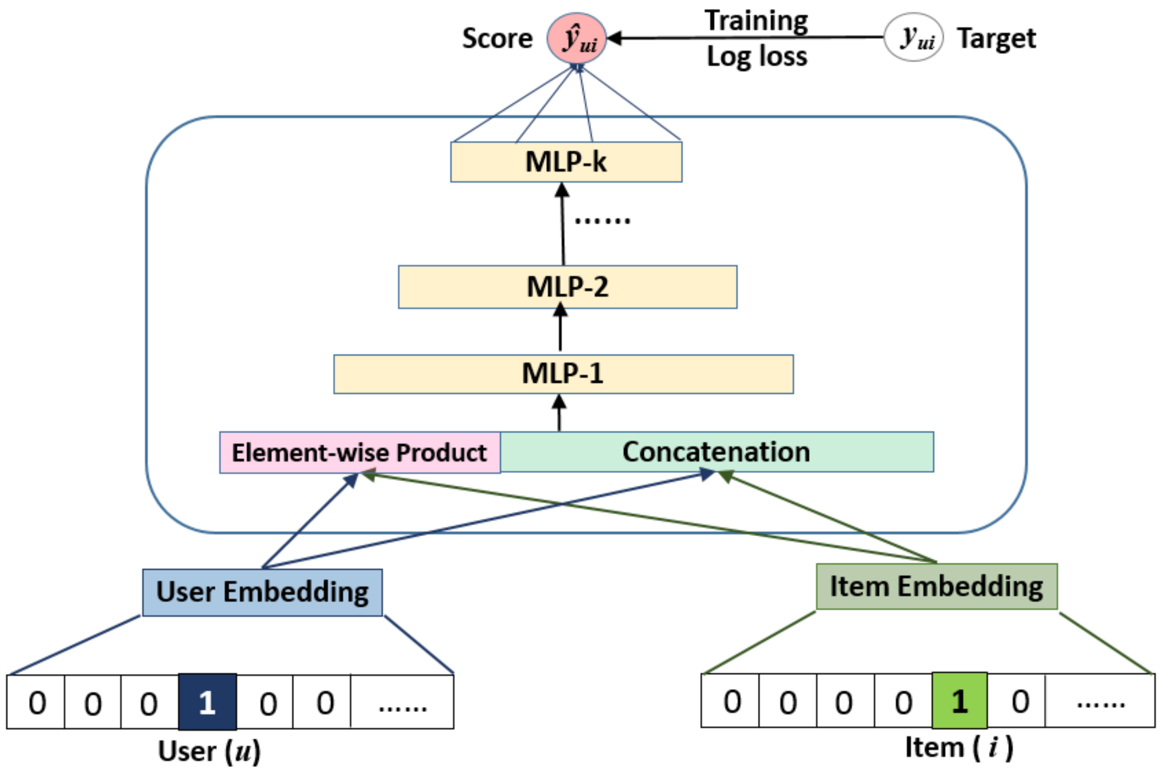
We have to design more effective models to make DNN work for CF!

MLP can learn to approximate the low-rank relation, but is inefficient in doing so.

Slides: http://comp.nus.edu.sg/~xiangnan/icmr18-recsys.pdf

NeuMF: Neural Matrix Factorization (He et al, WWW'17)

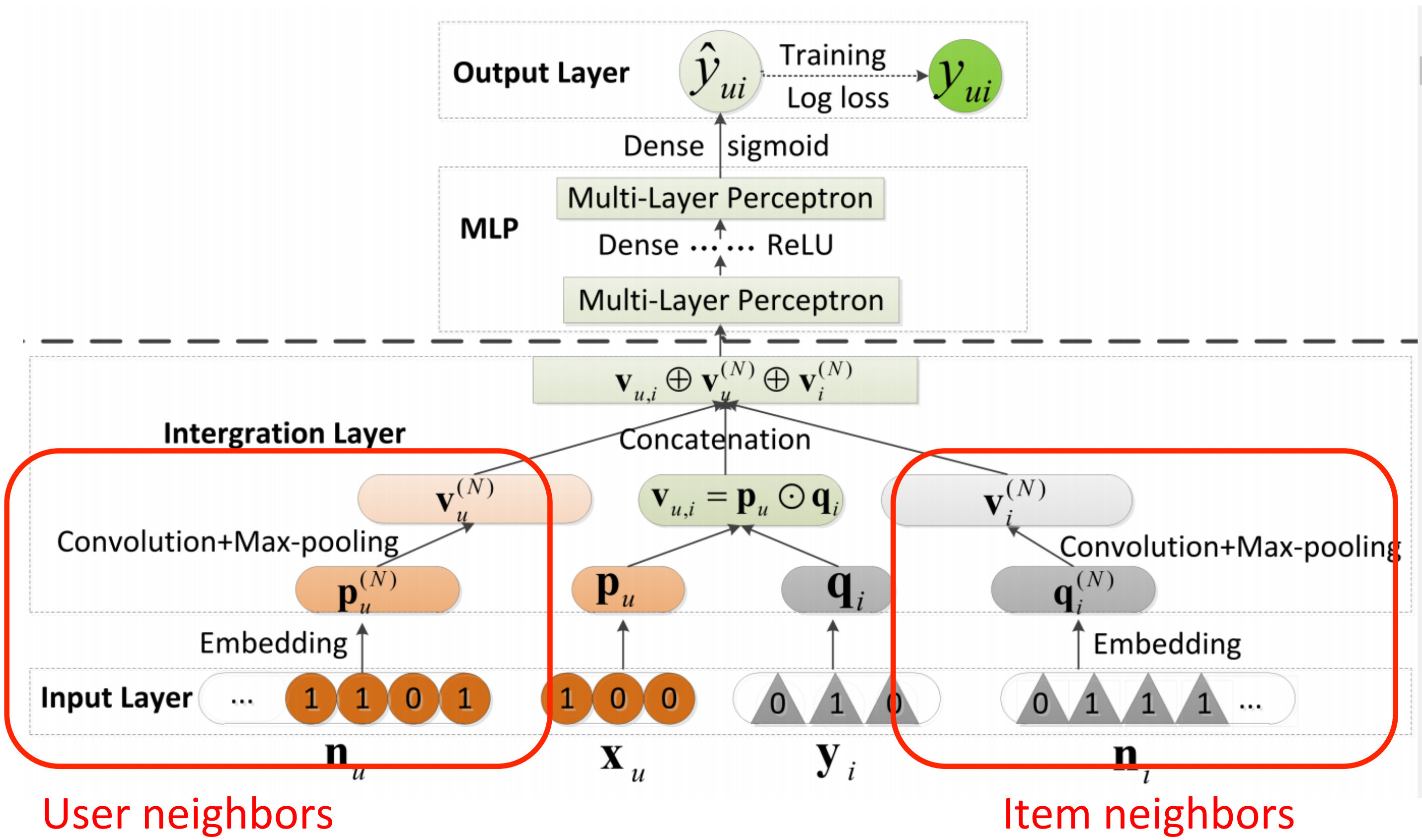
- NeuMF unifies the strengths of MF and MLP in learning the matching function:
  - MF uses inner product to capture the low-rank relation
  - MLP is more flexible in using DNN to learn the matching function.



Slides: http://comp.nus.edu.sg/~xiangnan/icmr18-recsys.pdf

NNCF: Neighbor-based NCF (Bai et al, CIKM'17)

- Feeding user and item neighbors into the NCF framework
  - Direct neighbors or community neighbors are considered.



Slides: http://comp.nus.edu.sg/~xiangnan/icmr18-recsys.pdf

Experiment Evidence

Datasets	#Interaction	# Users	#Items	Sparsity
Delicious	437,593	1,867	69,223	99.66%
MovieLens	1,000,209	3,706	6,040	95.53%

Performance Comparison on Item Recommendation (%)

Datasets	**Delicious**		**MovieLens**	
Models	HR@5	NDCG@5	HR@5	NDCG@5
ItemPop	5.41	3.22	31.49	20.18
ItemKNN	59.69	55.90	45.01	30.14
MF-BPR	73.77	74.11	51.03	36.21
NeuMF	85.53	80.68	56.55	38.30
NNCF	87.31	84.58	62.00	42.21

CF method is better than non-personalized method

Model-based CF is better than memory-based CF

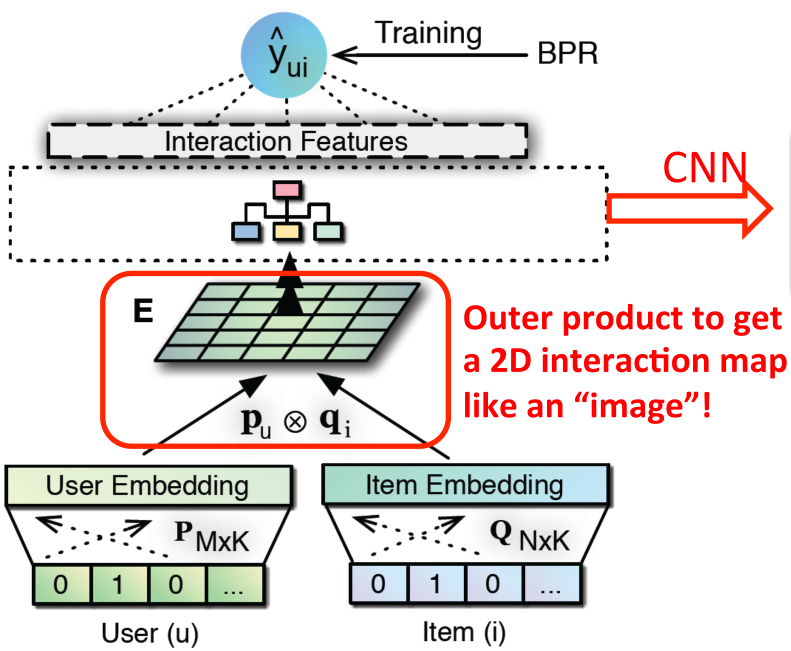
Deep NCF models are better than shallow MF models by a large margin.

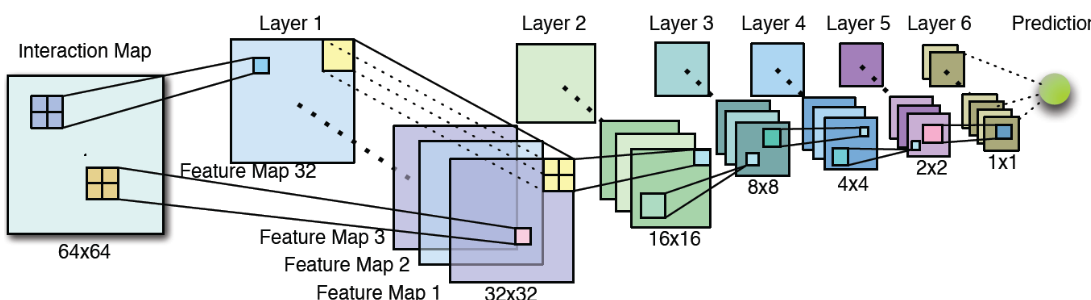
(Bai et al, CIKM'17)

Slides: http://comp.nus.edu.sg/~xiangnan/icmr18-recsys.pdf

Convolutional NCF (He et al, IJCAI'18)

- Although fully connected layers are popular in learning the matching function, they have too many parameters.
- Recently, we propose to use the **locally connected CNN** to build deeper NCF models.





- 2 Fully Connected Layers: > 10M parameters
- **6 Convolutional Layers: 20K parameters, but achieve better performance!**

Slides: http://comp.nus.edu.sg/~xiangnan/icmr18-recsys.pdf

Experiment Evidence

Datasets	#Interactions	#Users	#Items	Sparsity
Yelp	730,791	25,815	25,677	99.89%
Gowalla	1,249,703	54,156	52,400	99.95%

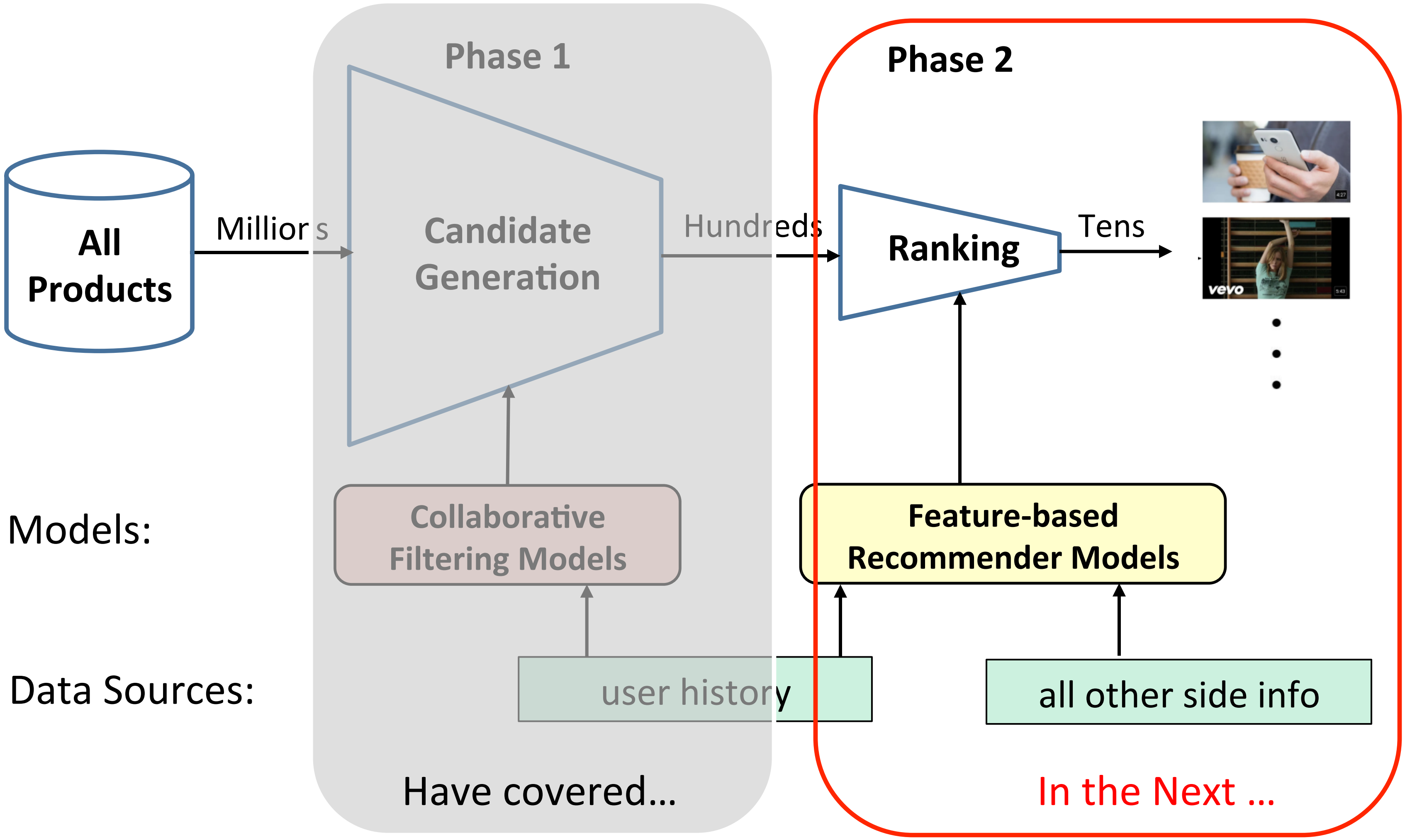
Datasets	Gowalla		Yelp	
Models	HR@5	NDCG@5	HR@5	NDCG@5
ItemPop	20.03	10.99	7.10	3.65
MF-BPR	62.84	48.25	17.52	11.04
MLP	63.59	48.02	17.66	11.03
IRGAN	63.89	49.58	18.61	11.98
NeuMF	67.44	53.19	18.81	11.89
ConvNCF	69.14	54.94	19.06	12.09

ConvNCF are better than NeuMF and MLP with much fewer parameters.

(He et al, IJCAI'18)

Slides: http://comp.nus.edu.sg/~xiangnan/icmr18-recsys.pdf

Morden RecSys Architecture



Slides: http://comp.nus.edu.sg/~xiangnan/icmr18-recsys.pdf

Recall: Input to Feature-based Models

Feature vector **x**

	User				Movie					Other Movies rated					Target y	
	A	B	C	...	TI	NH	SW	ST	...	TI	NH	SW	ST	...		
$\mathbf{x}^{(1)}$	1	0	0	...	1	0	0	0	...	0.3	0.3	0.3	0	...	5	$y^{(1)}$
$\mathbf{x}^{(2)}$	1	0	0	...	0	1	0	0	...	0.3	0.3	0.3	0	...	3	$y^{(2)}$
$\mathbf{x}^{(3)}$	1	0	0	...	0	0	1	0	...	0.3	0.3	0.3	0	...	1	$y^{(3)}$
$\mathbf{x}^{(4)}$	0	1	0	...	0	0	1	0	...	0	0	0.5	0.5	...	4	$y^{(4)}$
$\mathbf{x}^{(5)}$	0	1	0	...	0	0	0	1	...	0	0	0.5	0.5	...	5	$y^{(5)}$
$\mathbf{x}^{(6)}$	0	0	1	...	1	0	0	0	...	0.5	0	0.5	0	...	1	$y^{(6)}$
$\mathbf{x}^{(7)}$	0	0	1	...	0	0	1	0	...	0.5	0	0.5	0	...	5	$y^{(7)}$

Raw features:

1. Categorical features

   One-hot encoding on ID features

2. Continuous features

   E.g., time, frequency.

   Need feature normalization

Transformed features:

1. Categorical features

   Cross features are important (e.g., AND (A=true, B=true))

2. Continuous features

   E.g., outputs of other models like visual embeddings.

Slides: http://comp.nus.edu.sg/~xiangnan/icmr18-recsys.pdf

Importance of Cross Feature

An example that cross feature (feature interaction) is important for prediction:

Task: predict customer *income.*

Two input variables:

1) occupation = {banker, engineer,...}

2) Level = {junior, senior}

Facts:

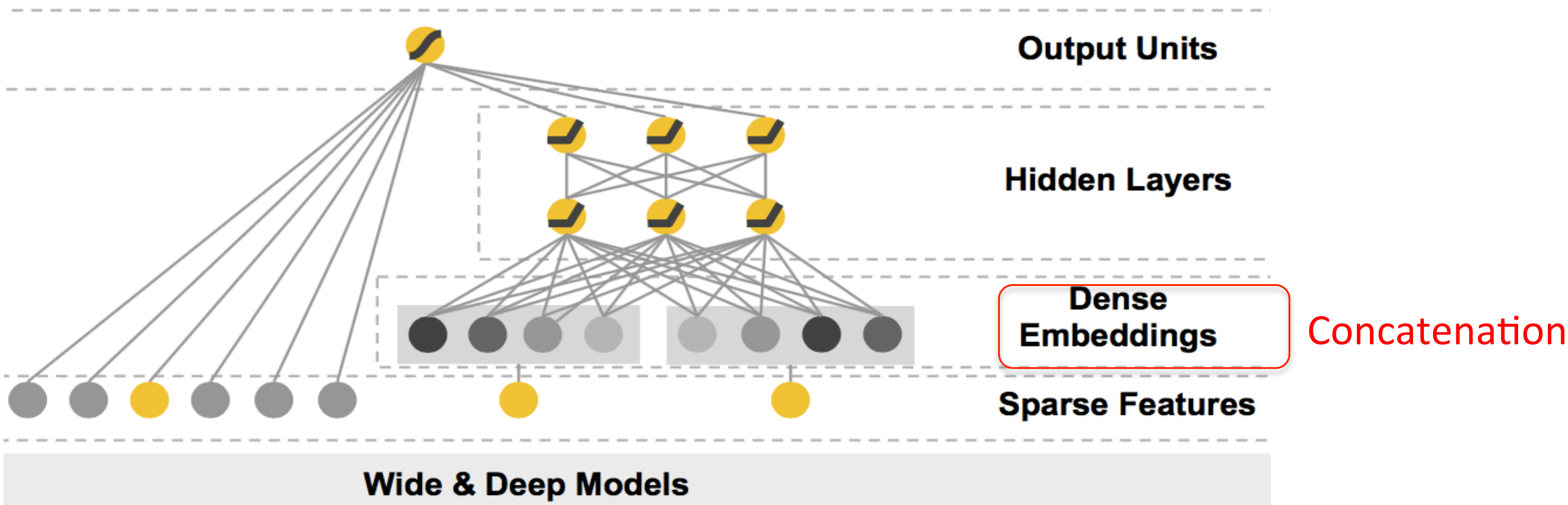
***income*(junior, banker) < income(junior, engineer)**

but,

Income(senior, banker) > income(senior, engineer)

Slides: http://comp.nus.edu.sg/~xiangnan/icmr18-recsys.pdf

Wide&Deep (Cheng et al, Recsys'16)

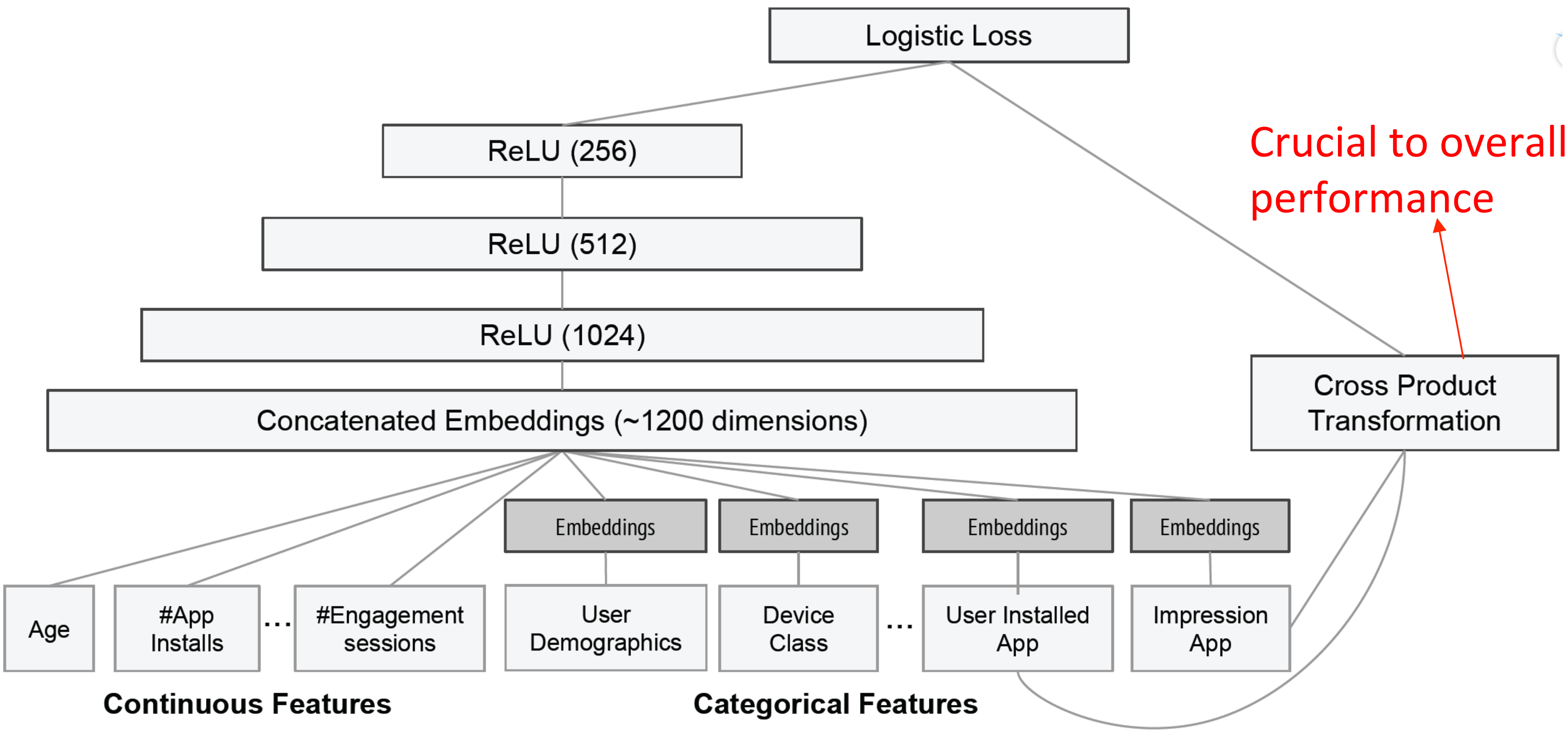


- The wide part is linear regression for memorizing seen feature interactions, which requires careful engineering on cross features.
  E.g., *AND(gender=female, language=en)* is 1 iff both single features are 1

- The deep part is DNN for generalizing to unseen feature interactions.
  Cross feature effects are captured in an implicit way.

Slides: http://comp.nus.edu.sg/~xiangnan/icmr18-recsys.pdf

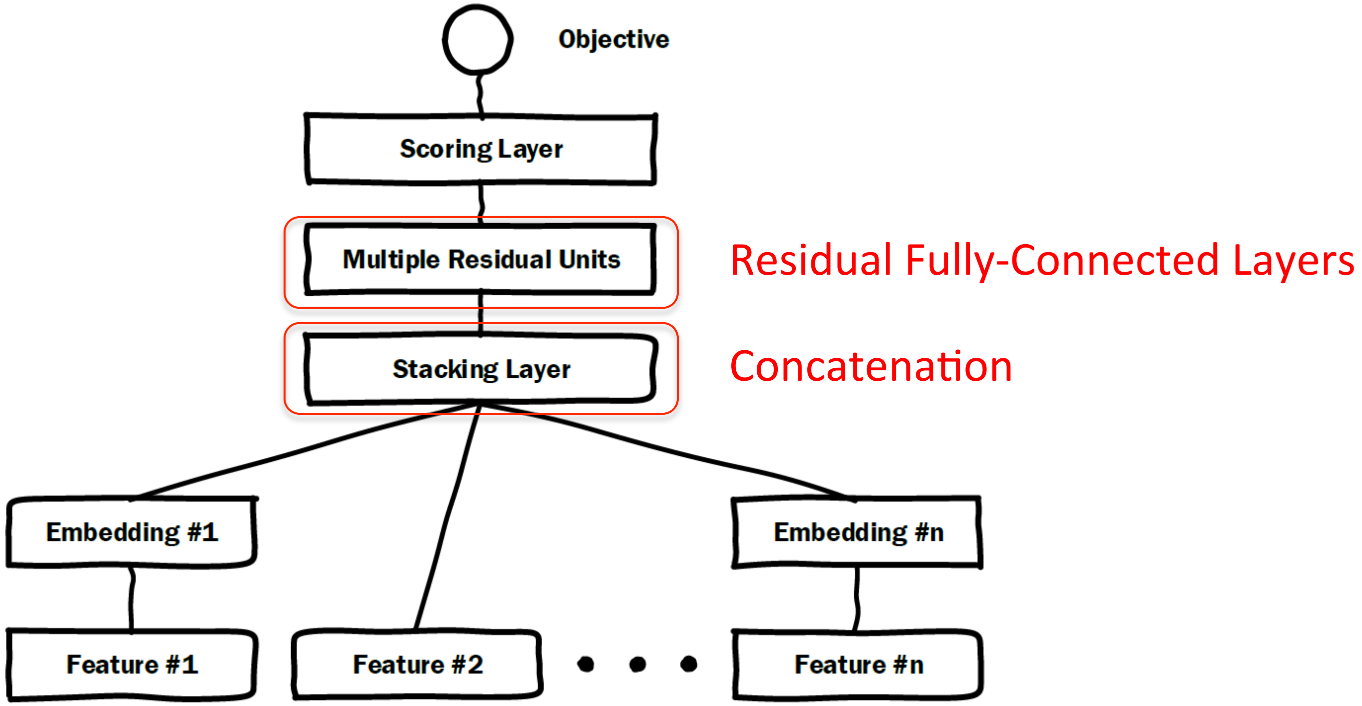
Wide&Deep for Google App Recommendation (Cheng et al, Recsys'16)



Slides: http://comp.nus.edu.sg/~xiangnan/icmr18-recsys.pdf

Deep Crossing (Shan et al, KDD'16)

Microsoft's Sponsor Search Solution in 2016:



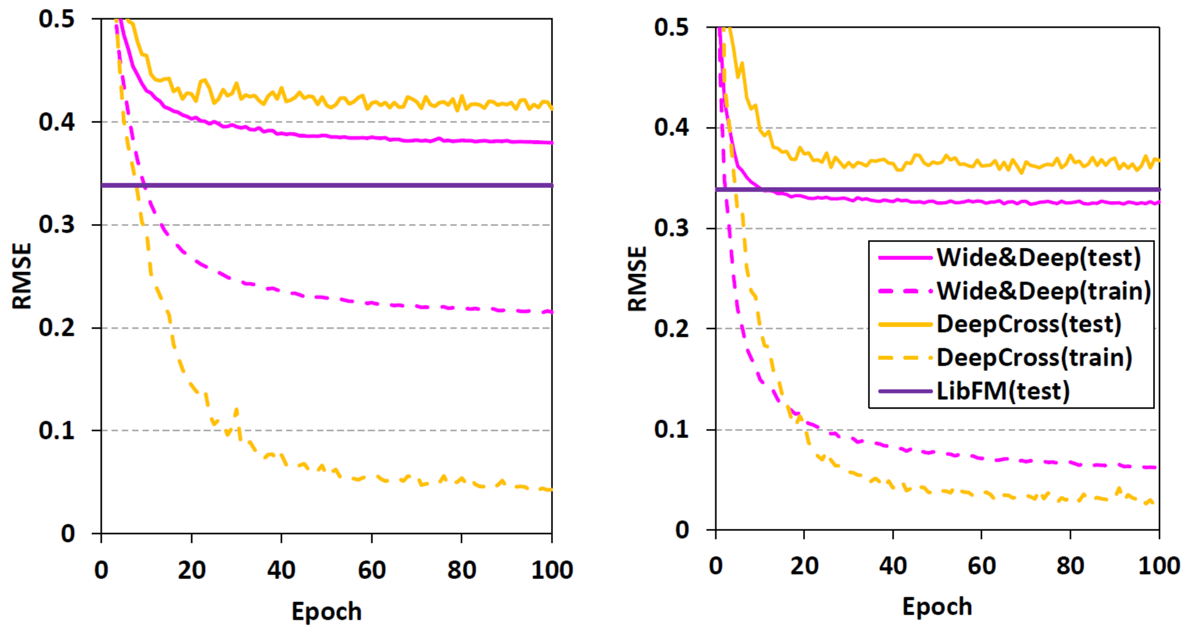
The use of residual layers makes the network be deep.

Slides: http://comp.nus.edu.sg/~xiangnan/icmr18-recsys.pdf

Empirical Evidence

- However, when only raw features are used, both DL models don't perform well in learning unseen feature interactions.

Solid line: testing loss; Dashed line: training loss



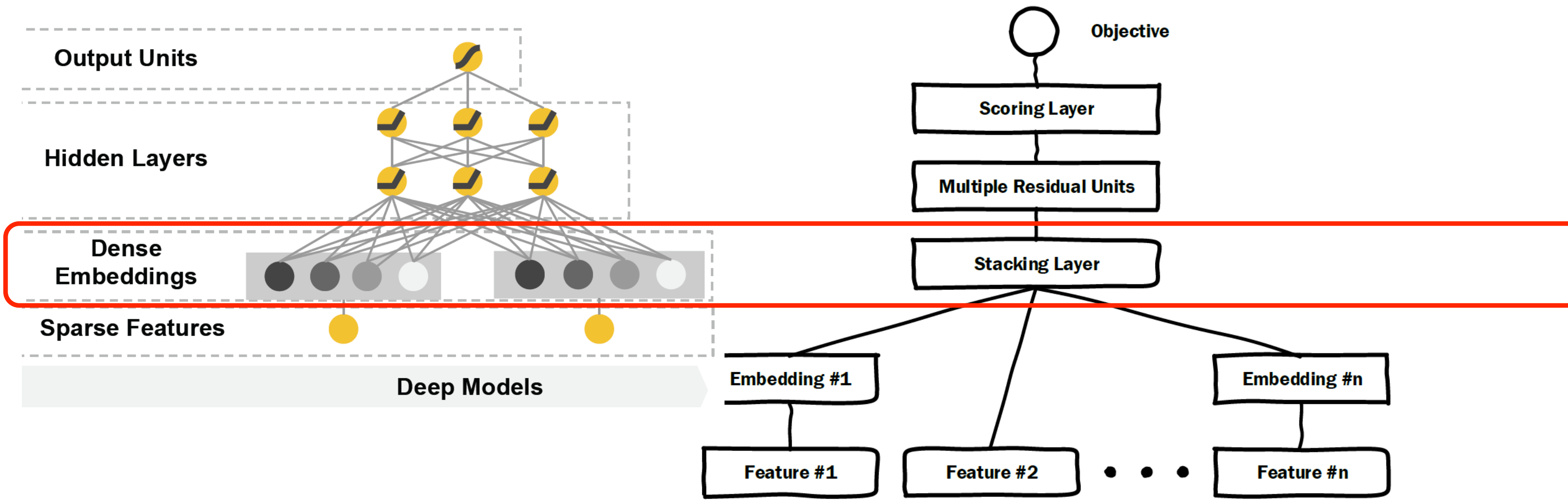
(a) Random initialization (b) FM as pre-training

(He and Chua, SIGIR'17)

Slides: http://comp.nus.edu.sg/~xiangnan/icmr18-recsys.pdf

Why MLP is Ineffective?

Besides optimization difficulties, one reason might be model design:

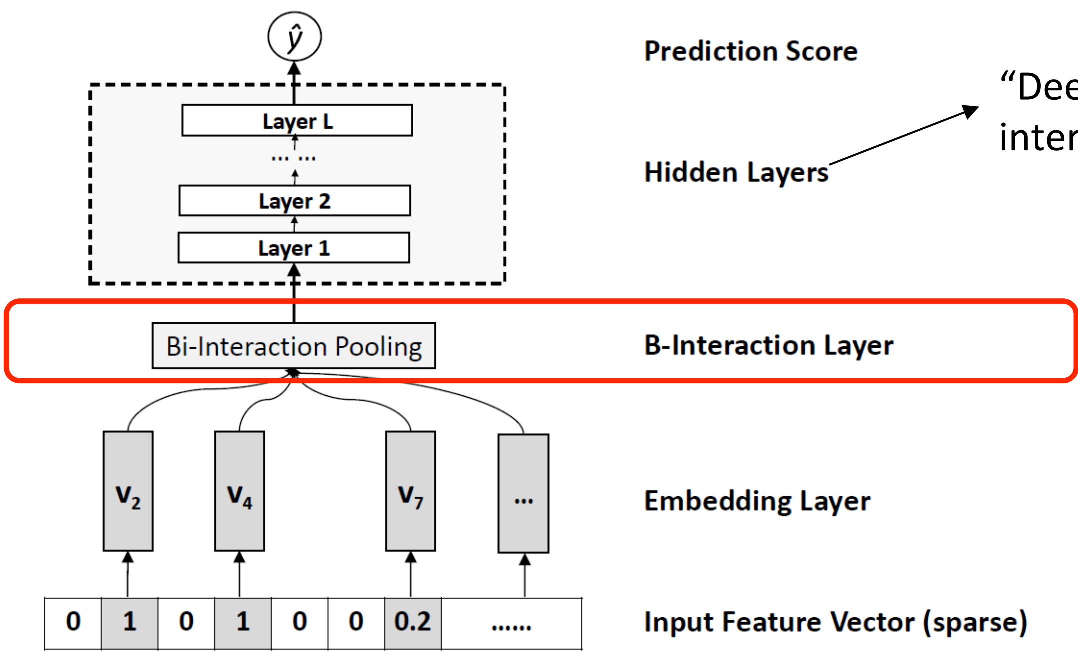


1. Embedding concatenation carries little information about feature interactions in the low level!
2. The structure of Concat+MLP is ineffective to learn the multiplicative relation (Beutel et al, WSDM'18).

Slides: http://comp.nus.edu.sg/~xiangnan/icmr18-recsys.pdf

NFM: Neural Factorization Machine (He and Chua, SIGIR'17)

- Inspired by FM, NFM models pairwise interactions between feature embeddings with multiplication.



"Deep layers" learn **higher-order** feature interactions only, being much easier to train.

Bilinear Interaction Pooling:

$$f_{BI}(\mathcal{V}_x) = \sum_{i=1}^{n} \sum_{j=i+1}^{n} x_i \mathbf{v}_i \odot x_j \mathbf{v}_j,$$

Figure 2: Neural Factorization Machines model (the first-order linear regression part is not shown for clarity).

Slides: http://comp.nus.edu.sg/~xiangnan/icmr18-recsys.pdf

Experiment Evidence

Task #1: Context-aware App Usage Prediction
- Frappe data: instance #: 288,609, feature #: 5,382

Task #2: Personalized Tag Recommendation
- MovieLens data: Inst #: 2,006,859, Feat #: 90,445

Table: Parameter # and testing RMSE at embedding size 128

	Frappe		MovieLens	
Method	**Param#**	**RMSE**	**Param#**	**RMSE**
Logistic Regression	5.38K	0.5835	0.09M	0.5991
FM	1.38M	0.3385	23.24M	0.4735
High-order FM	2.76M	0.3331	46.40M	0.4636
Wide&Deep (3 layers)	4.66M	0.3246	24.69M	0.4512
DeepCross (10 layers)	8.93M	0.3548	25.42M	0.5130
Neural FM (1 layer)	**1.45M**	**0.3095**	**23.31M**	**0.4443**

1. Shallow embedding methods learn interactions, better than simple linear models

2. Deep embedding methods:
Wide&Deep = Concat+3 layers
DeepCross = Concat+10 layers

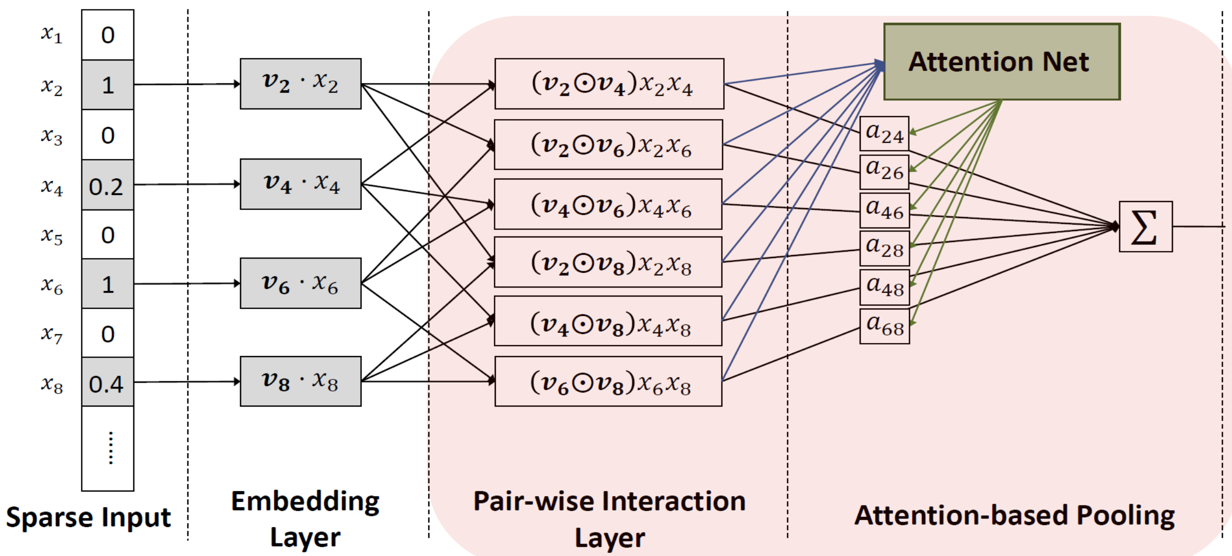
3. Our methods:
Neural FM = BI pooling + 1 layer
Shallower but outperforming existing deeper methods with less parameters.

Codes: github.com/hexiangnan/neural_factorization_machine

Slides: http://comp.nus.edu.sg/~xiangnan/icmr18-recsys.pdf

AFM: Attentional Factorization Machine (Xiao et al, IJCAI'17)

- Neural FM treats all second-order feature interactions as contributing equally.
- Attentional FM uses an attention network to learn the weight of a feature interaction.



$$f_{ABI}(\mathcal{V}_x) = \sum_{i=1}^{n} \sum_{j=i+1}^{n} (x_i \mathbf{v}_i \odot x_j \mathbf{v}_j) a_{ij}$$

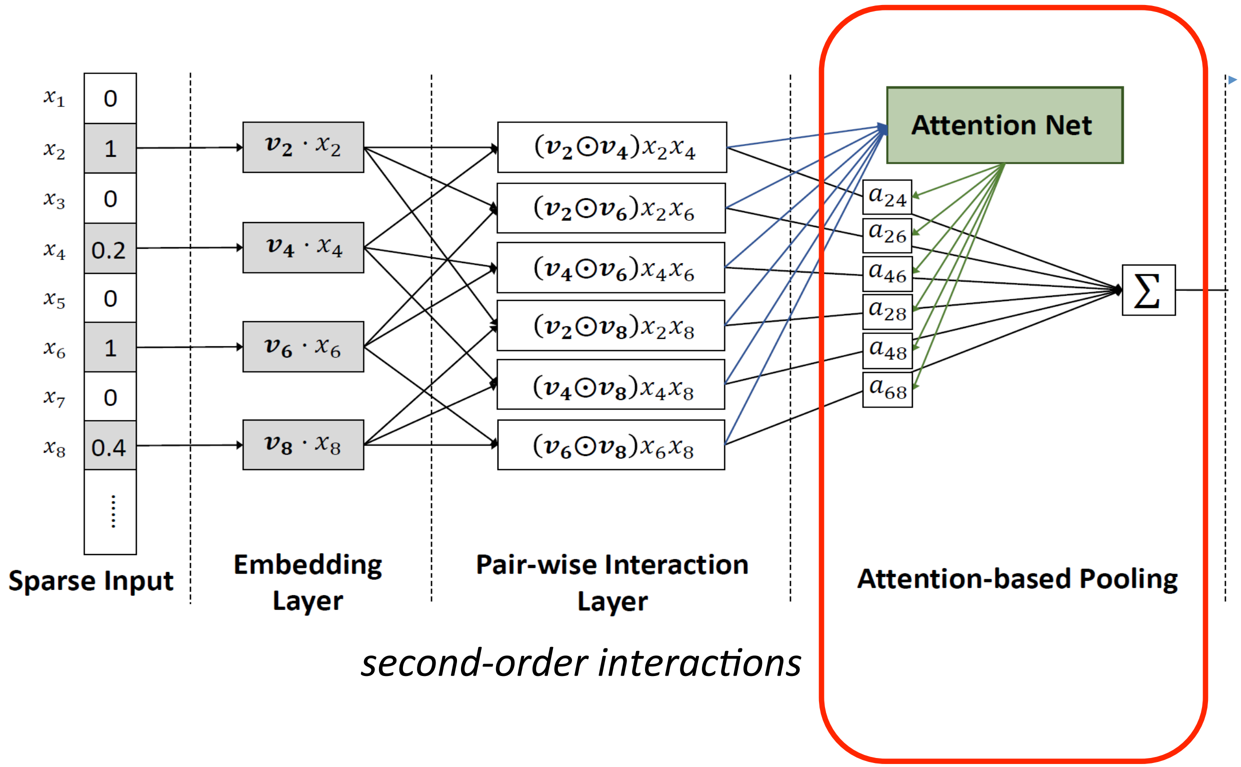
$$a'_{ij} = \mathbf{h}^T ReLU(\mathbf{W}(\mathbf{v}_i \odot \mathbf{v}_j) x_i x_j + \mathbf{b}),$$

$$a_{ij} = \frac{\exp(a'_{ij})}{\sum_{(i,j)\in \mathcal{R}_x} \exp(a'_{ij})},$$

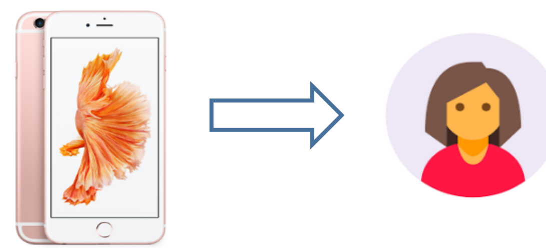
Slides: http://comp.nus.edu.sg/~xiangnan/icmr18-recsys.pdf

Explaining Recommendation with AFM

The attention scores can be used to select the most predictive second-order feature interactions as explanations.



Example: explainable recommendation with second-order cross features:



<Female, Age 20>
<Age 20, iPhone>
<Female, Color Pink>
......

Slides: http://comp.nus.edu.sg/~xiangnan/icmr18-recsys.pdf

Experiment Evidence

Task #1: Context-aware App Usage Prediction
- Frappe data: instance #: 288,609, feature #: 5,382

Task #2: Personalized Tag Recommendation
- MovieLens data: Inst #: 2,006,859, Feat #: 90,445

Table: Parameter # and testing RMSE at embedding size 128

	Frappe		MovieLens	
Method	**Param#**	**RMSE**	**Param#**	**RMSE**
Logistic Regression	5.38K	0.5835	0.09M	0.5991
FM	1.38M	0.3385	23.24M	0.4735
High-order FM	2.76M	0.3331	46.40M	0.4636
Wide&Deep (3 layers)	4.66M	0.3246	24.69M	0.4512
DeepCross (10 layers)	8.93M	0.3548	25.42M	0.5130
Neural FM (1 layer)	**1.45M**	**0.3095**	**23.31M**	**0.4443**
Attentional FM (0 layer)	**1.45M**	**0.3102**	**23.26M**	**0.4325**

AFM without hidden layers can be better than NFM with 1 hidden layer.

Codes: github.com/hexiangnan/attentional_factorization_machine

Slides: http://comp.nus.edu.sg/~xiangnan/icmr18-recsys.pdf

Short Summary

- ✓ Deep methods for collaborative filtering
  - User/Item representation learning
  - Matching function learning
- ✓ Deep methods for feature-based recommendation
  - Cross features are important to encode
  - DNN can effectively learn feature interactions, but require careful design
- It remains challenging to do explainable learning on high-order feature interaction.

Slides: http://comp.nus.edu.sg/~xiangnan/icmr18-recsys.pdf

Reference

- Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. Neural collaborative filtering. In WWW 2017.
- Xiangnan He, Xiaoyu Du, Xiang Wang, Feng Tian, Jinhui Tang, and Tat-Seng Chua. Out Product-based Neural Collaborative Filtering. In IJCAI 2018.
- Xiangnan He, and Tat-Seng Chua. Neural factorization machines for sparse predictive analytics. In SIGIR 2017.
- Jun Xiao, Hao Ye, Xiangnan He, Hanwang Zhang, Fei Wu, and Tat-Seng Chua. Attentional factorization machines: Learning the weight of feature interactions via attention networks. In IJCAI 2017.
- Ting Bai, Ji-Rong Wen, Jun Zhang, and Wayne Xin Zhao. A Neural Collaborative Filtering Model with Interaction-based Neighborhood. In CIKM 2017.
- Paul Covington, Jay Adams, and Emre Sargin. Deep neural networks for youtube recommendations. In Recsys 2016.
- Xiang Wang, Xiangnan He, Liqiang Nie, and Tat-Seng Chua. Item silk road: Recommending items from information domains to social users. In SIGIR 2017.
- Hong-Jian Xue, Xin-Yu Dai, Jianbing Zhang, Shujian Huang, and Jiajun Chen. Deep matrix factorization models for recommender systems. IJCAI 2017.
- Suvash Sedhain, Aditya Krishna Menon, Scott Sanner, and Lexing Xie. Autorec: Autoencoders meet collaborative filtering. In WWW 2015.
- Yao Wu, Christopher DuBois, Alice X. Zheng, and Martin Ester. Collaborative denoising auto-encoders for top-n recommender systems. In WSDM 2016.
- Heng-Tze Cheng, Levent Koc, Jeremiah Harmsen, Tal Shaked, Tushar Chandra, Hrishi Aradhye, Glen Anderson et al. Wide & deep learning for recommender systems. In DLRS 2016.
- Ying Shan, T. Ryan Hoens, Jian Jiao, Haijing Wang, Dong Yu, and J. C. Mao. Deep crossing: Web-scale modeling without manually crafted combinatorial features. In KDD 2016.
- Alex Beutel, Paul Covington, Sagar Jain, Can Xu, Jia Li, Vince Gatto, and Ed H. Chi. 2018. Latent Cross: Making Use of Context in Recurrent Recommender Systems. In WSDM 2018.

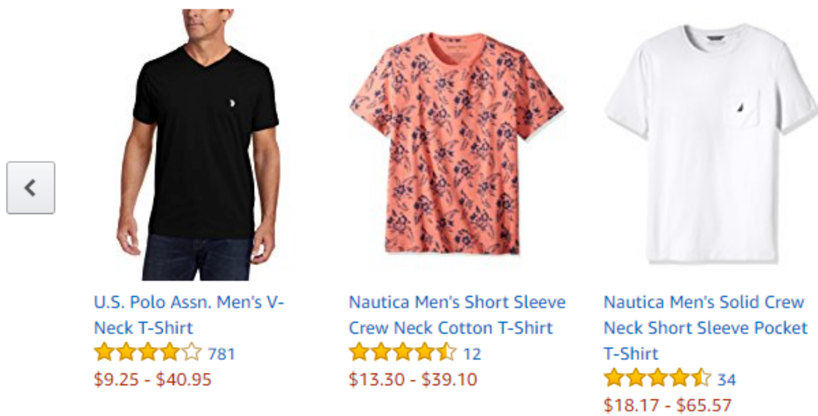
Slides: http://comp.nus.edu.sg/~xiangnan/icmr18-recsys.pdf

Outline of Tutorial

- Background (Xiangnan, 10 mins)
- Basics & Advances in Recommendation (Xiangnan, 50 mins)
- Visually-aware Product Recommendation (Xiangnan, 30 mins)
- Break (15 mins)
- Visual Representation (Hanwang, 45 mins)
- Image/Video Recommendation (Hanwang, 25 mins)
- Conclusion (Hanwang, 5 mins)

Problem Formulation

I have purchased:



Will I like this one?

Each item = ID + image

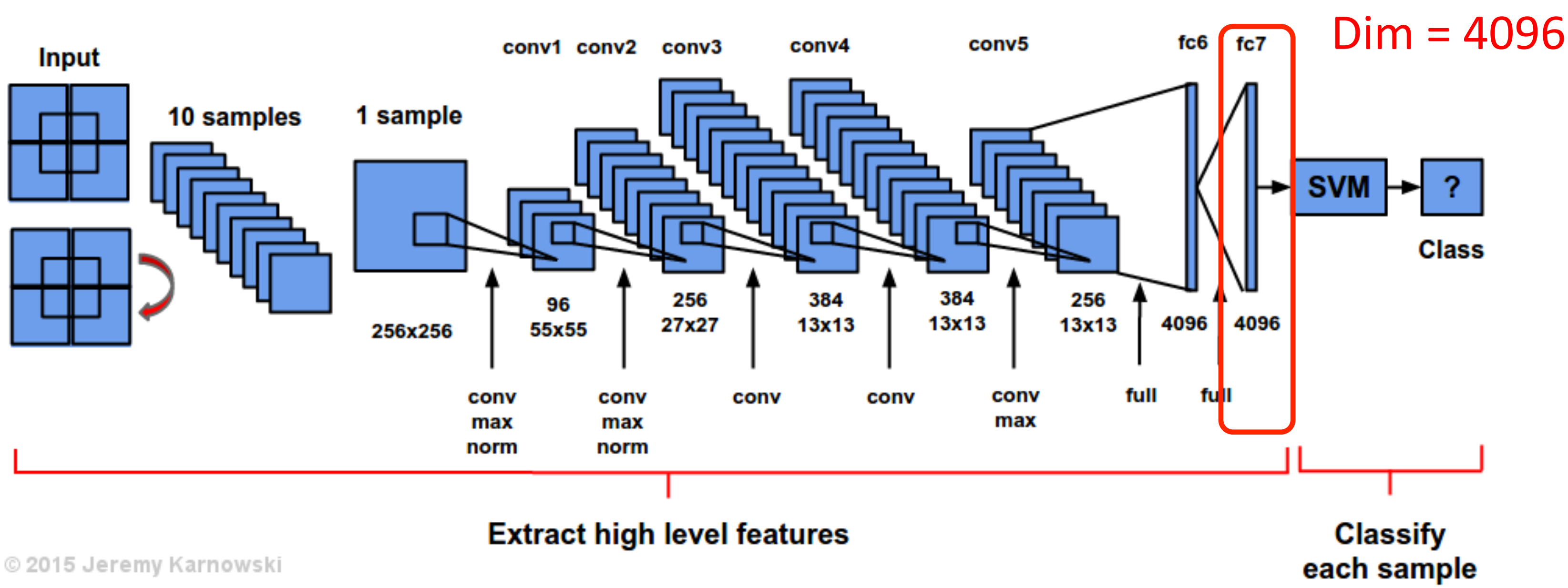
Two key questions:

1. How to understand image?
2. How to integrate image feature into CF model?

Slides: http://comp.nus.edu.sg/~xiangnan/icmr18-recsys.pdf

Image Understanding

- Traditional (low-level) image features:
  - Pixels, Color histograms
  - SIFT descriptors
  - Speeded Up Robust Features (SURF)
- Gap between low-level features and real semantics.
- Recent work uses deep CNN as feature extractor.



Slides: http://comp.nus.edu.sg/~xiangnan/icmr18-recsys.pdf

Collaborative Filtering with CNN Features

- Let $\mathbf{f}_i$ be CNN features for image i:
  - Usually of thousands dimension: AlexNet: 4096, ResNet: 2048
- MF predicts user rating on image i:

$$\hat{y}_{ui} = < \mathbf{p}_u, \mathbf{f}_i > = \mathbf{p}_u^T \mathbf{f}_i$$

User preference on image CNN features

- Problem:
  - $\mathbf{p}_u$ has to be of the same dimension as $\mathbf{f}_i$
  - Too big for CF latent space: too many parameters => overfitting

    E.g., 100 million users * 4096 * 8 B = 3.28 TB
  - Typically, the dimension of CF latent space is hundreds (128, 256) at most.

Slides: http://comp.nus.edu.sg/~xiangnan/icmr18-recsys.pdf

Deep CNN Features => CF Latent Space

- An intuitive solution is to do dimension reduction on CNN features, e.g., PCA
- However, it will lose signal in CNN features.
  - The objective of dimension reduction is not recommendation.
- Solution: learning a transformation matrix to do the projection based on user-item interactions:

$$\hat{y}_{ui} = \mathbf{p}_u^T(\mathbf{E}\mathbf{f}_i)$$

Transformation matrix that projects CNN features to CF latent space

  - **E** is optimized for the recommendation task.

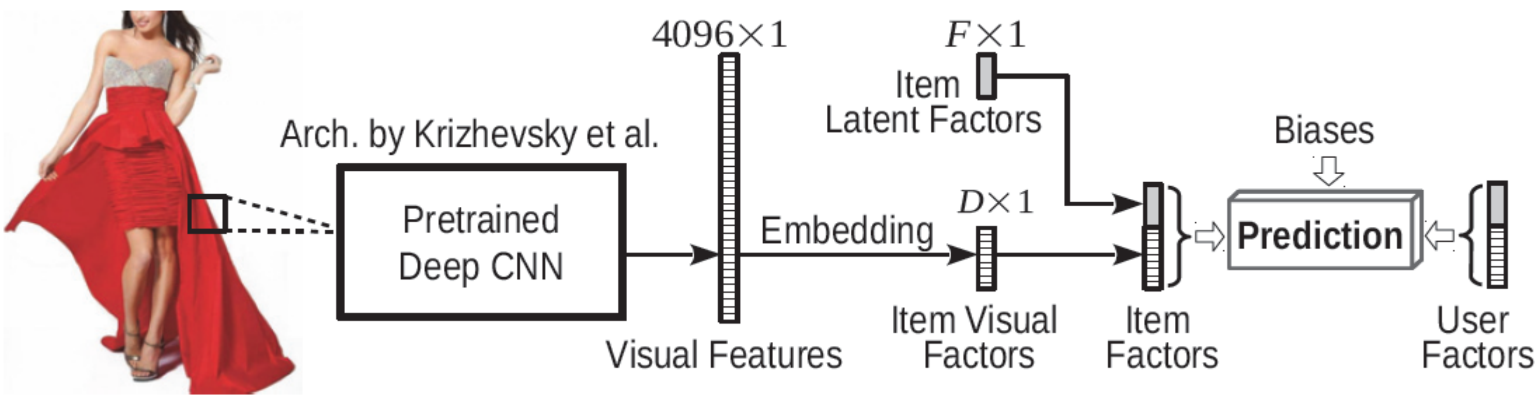
Slides: http://comp.nus.edu.sg/~xiangnan/icmr18-recsys.pdf

VBPR: Visual Bayesian Personalized Ranking (He et al, AAA'16)

- The first visually-aware recommendation method based on Deep CNN features:

$$\hat{y}_{ui} = b_u + b_i + \mathbf{v}_u^T \mathbf{v}_i + \mathbf{p}_u^T (\mathbf{E}\mathbf{f}_i)$$

Bias terms   CF prediction   Image feature-based Prediction



Slides: http://comp.nus.edu.sg/~xiangnan/icmr18-recsys.pdf

VBPR: Visual Bayesian Personalized Ranking (He et al, AAA'16)

$$\hat{y}_{ui} = b_u + b_i + \mathbf{v}_u^T \mathbf{v}_i + \mathbf{p}_u^T (\mathbf{E}\mathbf{f}_i)$$

- To learn model parameters, VBPR optimizes BPR pairwise loss:

sigmoid

Positive prediction

Negative prediction

$$L_{BPR} = \arg\max_{\Theta} \sum_{(u,i,j)\in\mathcal{R}_B} \ln \sigma(\hat{y}_{ui} - \hat{y}_{uj}) - \lambda ||\Theta||^2$$

Pairwise training examples: u prefers i over j

- Pairwise loss on the model:

$$\hat{y}_{ui} - \hat{y}_{uj} = \cancel{b_u} + b_i + \mathbf{v}_u^T \mathbf{v}_i + \mathbf{p}_u^T (\mathbf{E}\mathbf{f}_i) - [\cancel{b_u} + b_j + \mathbf{v}_u^T \mathbf{v}_j + \mathbf{p}_u^T (\mathbf{E}\mathbf{f}_j)]$$

$$= (b_i - b_j) + \mathbf{v}_u^T (\mathbf{v}_i - \mathbf{v}_j) + \mathbf{p}_u^T \mathbf{E}(\mathbf{f}_i - \mathbf{f}_j)$$

Margin on item bias

Margin on item embeddings

Margin on projected image features

Slides: http://comp.nus.edu.sg/~xiangnan/icmr18-recsys.pdf

Experimental Results

AUC score on personalized ranking

Dataset	Setting	(a) RAND	(b) MP	(e) BPR-MF	(f) VBPR	improvement f vs. best	improvement f vs. e
Amazon Women	All Items	0.4997	0.5772	0.7020	**0.7834**	9.4%	11.6%
	Cold Start	0.5031	0.3159	0.5281	**0.6813**	2.1%	29.0%
Amazon Men	All Items	0.4992	0.5726	0.7100	**0.7841**	9.1%	10.4%
	Cold Start	0.4986	0.3214	0.5512	**0.6898**	1.6%	25.1%
Amazon Phones	All Items	0.5063	0.7163	0.7918	**0.8052**	1.2%	1.7%
	Cold Start	0.5014	0.3393	0.5346	0.6056	-4.2%	13.3%
Tradesy.com	All Items	0.5003	0.5085	0.6198	**0.7829**	26.3%	26.3%
	Cold Start	0.4972	0.3721	0.5241	**0.7594**	44.9%	44.9%

1. RAND < MP (popularity) < BPR-MF < VBPR
2. VBPR has more improvements for cold-start items
3. Visual signal are more important for Clothing than Phones.

Slides: http://comp.nus.edu.sg/~xiangnan/icmr18-recsys.pdf

Visualization on Transformed CNN Features

$$\hat{y}_{ui} = \mathbf{p}_u^T(\mathbf{E}\mathbf{f}_i)$$

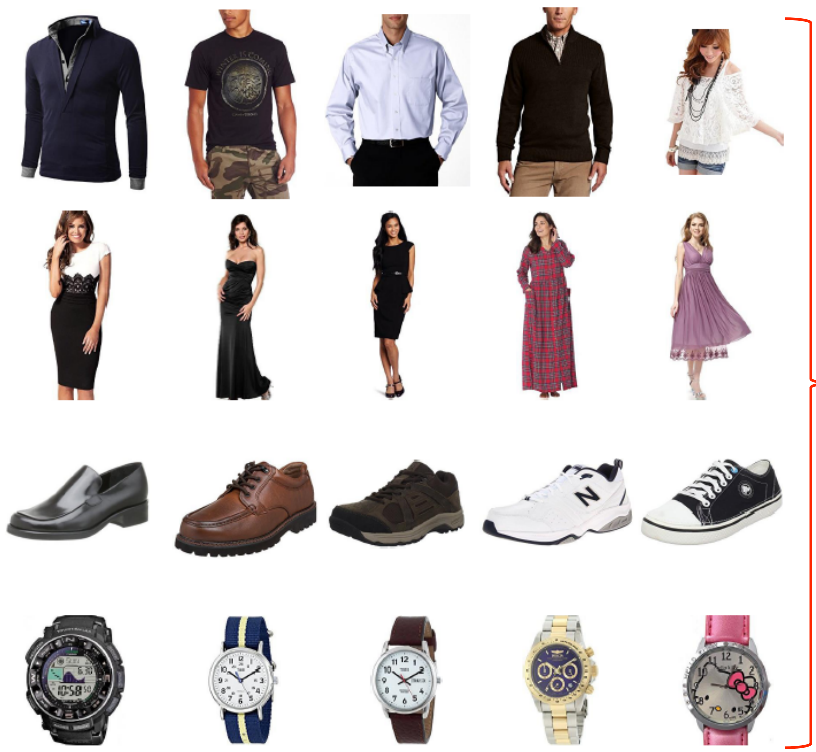
2D visualization (with t-SNE) on Amazon women dataset.

Items of the same subcategory are mapped to nearby locations.

DeepStyle (Liu et al, SIGIR'17)

- Drawback of VBPR: the transformed features mainly encode the category info, rather than the style info.
  - Example categories: ups, dresses, shoes, watches …
  - Example styles: casual, athletic, formal …

Each row is a cluster based on learned features.



Each cluster mixes items of different styles

Slides: http://comp.nus.edu.sg/~xiangnan/icmr18-recsys.pdf

DeepStyle (Liu et al, SIGIR'17)

- Item representation should be two-facet:

  item = *category* + *style*

- User rating on a fashion product is mainly determined by its style, rather than category.

- But how do we get style representation for an image?
  - A dedicated style-CNN needs labeled data.

- Solution: learn it from user-item interaction data!

Slides: http://comp.nus.edu.sg/~xiangnan/icmr18-recsys.pdf

DeepStyle (Liu et al, SIGIR'17)

Representation: *item* = *style* + *category*

Embedding: $\mathbf{Ef}_i = \mathbf{s}_i + \mathbf{c}_i$

Transformed CNN features | Style vector | Category vector

⇨ $$\mathbf{s}_i = \mathbf{Ef}_i - \mathbf{c}_i$$

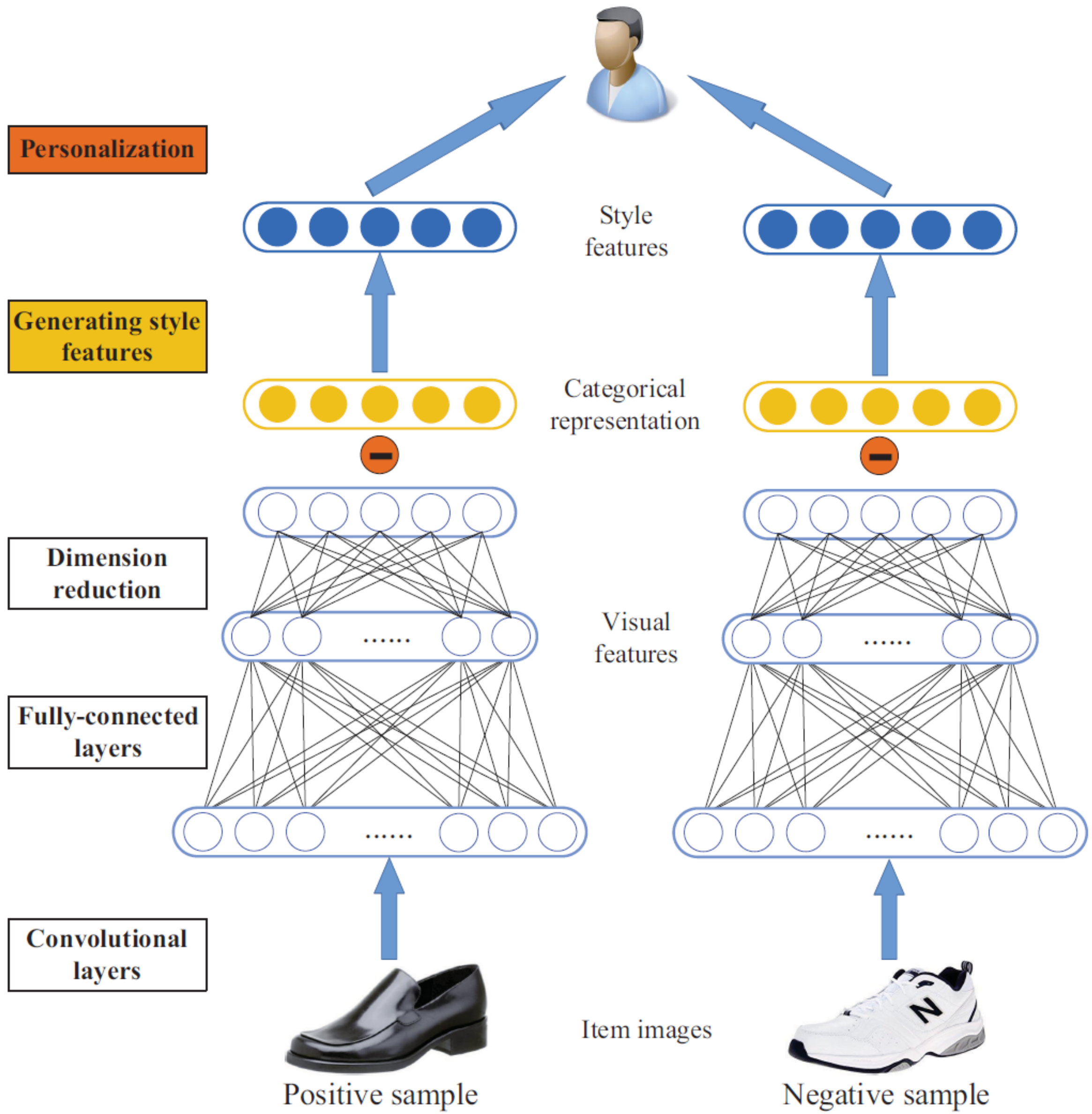
Prediction model:

$$\hat{y}_{ui} = b_u + b_i + \mathbf{v}_u^T \mathbf{v}_i + \mathbf{p}_u^T (\mathbf{Ef}_i - \mathbf{c}_i)$$

Remove the effect of category info in prediction

Slides: http://comp.nus.edu.sg/~xiangnan/icmr18-recsys.pdf

DeepStyle (Liu et al, SIGIR'17)



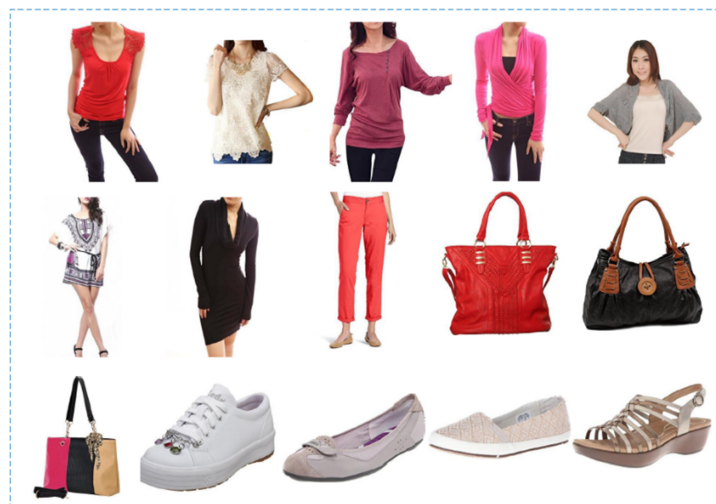
Experimental Results

AUC score on personalized ranking

dataset	setting	BPR	VBPR	DeepStyle
Clothing	warm-start	0.6243	0.7441	**0.7961**
	cold-start	0.5037	0.6915	**0.7317**
Home	warm-start	0.5848	0.6845	**0.7155**
	cold-start	0.5053	0.6140	**0.6396**

Room of improvement by learning better image representation.

Visualization on clusters of item style vector: $\mathbf{s}_i = \mathbf{E}\mathbf{f}_i - \mathbf{c}_i$



Each cluster mixes item of different categories, but they follow the same style!

Slides: http://comp.nus.edu.sg/~xiangnan/icmr18-recsys.pdf

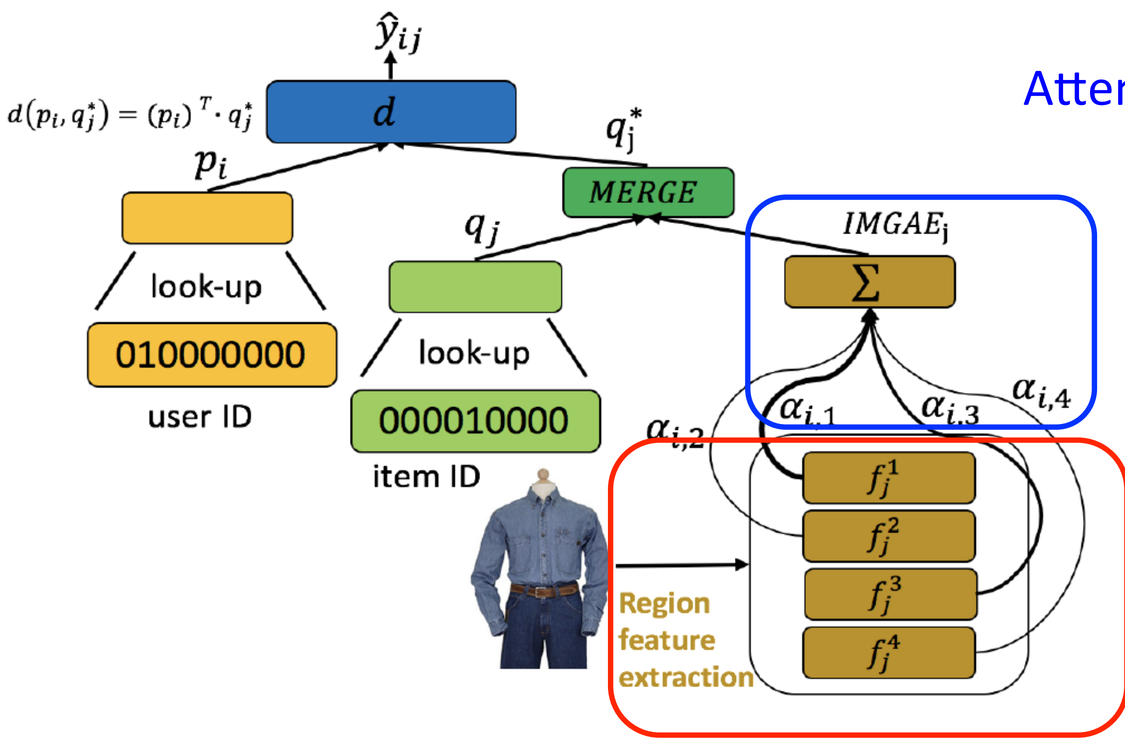
Visually Explainable CF (Chen et al, 2018)

- Users care about different visual features even on the same item.

Slides: http://comp.nus.edu.sg/~xiangnan/icmr18-recsys.pdf

Visually Explainable CF (Chen et al, 2018)

- Uncover region-level user preference with attention
  - Different regions of an image affect a user differently.
  - Originated from Attentive CF (Chen et al, SIGIR'17)



(a) Visually explainable collaborative filtering (VECF)

Attentive image representation:

$$\mathbf{IMAGE}_i = \sum_{r=1}^{R} \alpha_{u,i,r} \cdot \mathbf{f}_r^i$$

$$\alpha_{u,i,r} = \mathrm{softmax}(\mathbf{w}^T \mathbf{p}_u + \mathbf{h}^T \mathbf{q}_i + b)$$

Divide image by 14*14, each region is fed into a pre-trained VGG to get a 512-dim vector.

Slides: http://comp.nus.edu.sg/~xiangnan/icmr18-recsys.pdf

Visually Explainable CF + User Reviews (Chen et al, 2018)

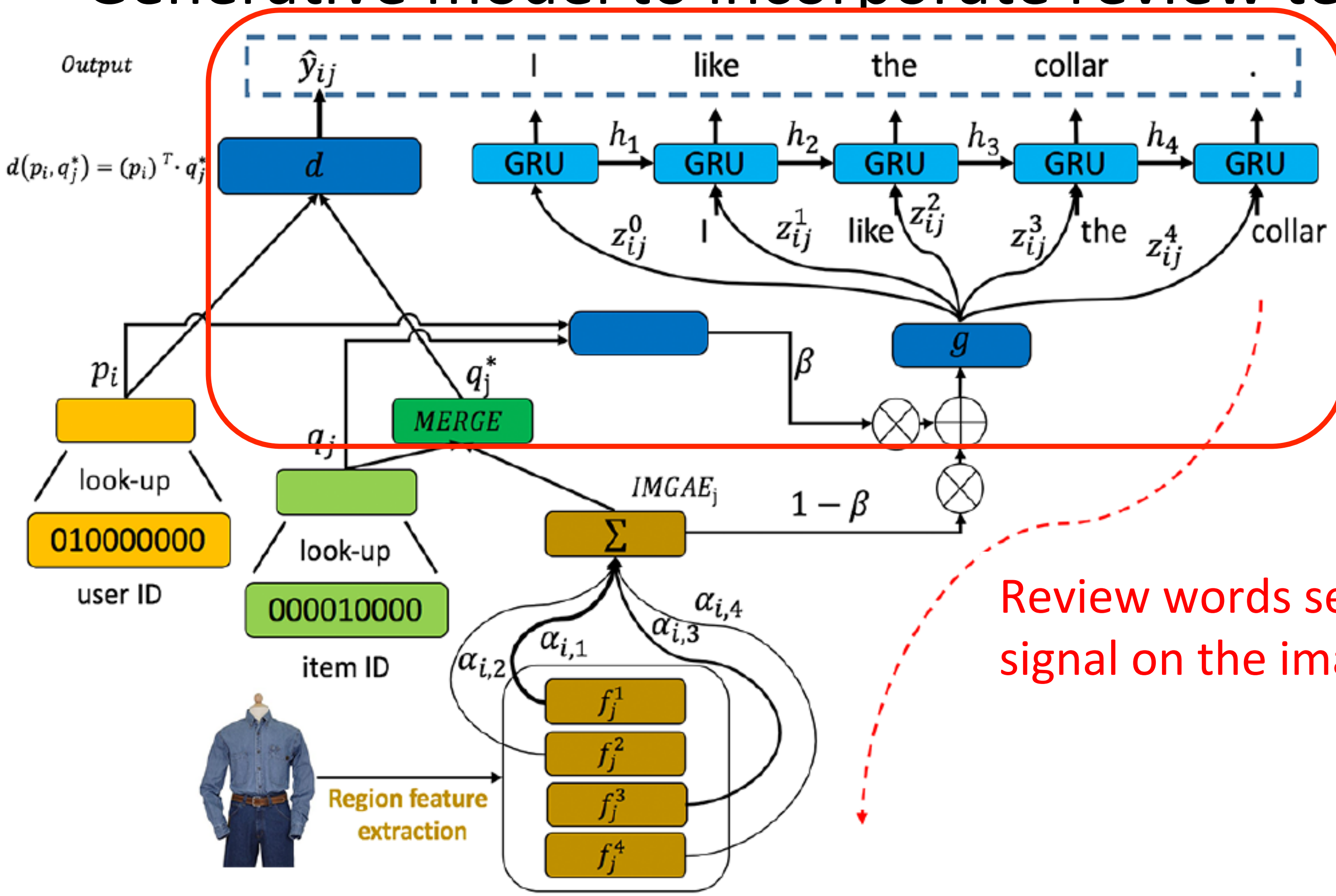
- Attention Net can learn user preference on regions, but may not be accurate in doing this.
- User review may help!

Some review words can be aligned with image regions

Slides: http://comp.nus.edu.sg/~xiangnan/icmr18-recsys.pdf

Visually Explainable CF + User Reviews (Chen et al, 2018)

- Generative model to incorporate review text:



Put in user vector, item vector and attentive image vector into GRU to generate review words.

Review words serve as a supervision signal on the image attention net.

(b) Review-enhanced visually explainable collaborative filtering (Re-VECF)

Slides: http://comp.nus.edu.sg/~xiangnan/icmr18-recsys.pdf

Visually Explainable CF + User Reviews (Chen et al, 2018)

- Learning model in a multi-task way:

$$l_2 = \delta \sum_{(i,j)} \sum_{t=1}^{l_{ij}} \log\ p(w_{ij}^t | \boldsymbol{w}_{ij}^{1:t-1}, \boldsymbol{z}_{ij}^{t-1})$$

Task #1: review generation
(error of generating each word)

$$+ (1-\delta)\left(\sum_{i \in \boldsymbol{u}} \sum_{j \in \boldsymbol{v}_+^i} \log \hat{y}_{ij} + \sum_{i \in \boldsymbol{u}} \sum_{j \in \boldsymbol{v}/\boldsymbol{v}_+^i} \log(1-\hat{y}_{ij}) \right) - \lambda ||\Theta||_F^2$$

Task #2: user preference prediction
(point-wise cross entropy loss)

Slides: http://comp.nus.edu.sg/~xiangnan/icmr18-recsys.pdf

Experimental Results

Dataset	Men			Women		
Measure@5(%)	F_1	HR	NDCG	F_1	HR	NDCG
BPR	1.209	3.901	0.740	0.897	3.342	0.611
HFT	1.242	4.243	0.757	0.915	3.371	0.631
VBPR	1.361	4.261	0.773	0.929	3.402	0.648
VECF	1.378	4.373	0.791	0.948	3.523	0.669
Re-VECF	**1.442**	**4.803**	**0.846**	**0.985**	**3.587**	**0.712**

HFT(review) vs. VBPR (image)

VECF (Visually Explainable CF) vs. Re-VECF (Review-enhanced VECF)

Image Attention with Review

#	Target Item	Historical Records	Textual Review	Visual Explanation VECF	Visual Explanation Re-VECF
2			***this is a really comfortable v-neck. i found that the size and location of the v are just right for me.*** i'm 5'8 & #34, but 200 lbs (and dropping :))		
3			***Great leggings. perfect for fly fishing or hunting or running.*** just perfect anytime you are cold!		
4			The socks on the shoes are a perfect fit for me. ***first time with a shoe with the speed laces and i like them a lot***		
5			Really like these socks! they are really thick woolen socks and are good for cold days. ***they cover a good portion of your feet as they go a little (halfway) above the calf muscle area.***		
6			***I like the front pocket~.*** Very cool!		

Slides: http://comp.nus.edu.sg/~xiangnan/icmr18-recsys.pdf

Short Summary

- The quality of image features is crucial for visually-aware recommendation.
  - Attention over regions is helpful

- After projecting image features to embedding space, it becomes a standard recommendation problem.
  - Any advanced preference learning techniques introduced in Part I can be used.

Reference

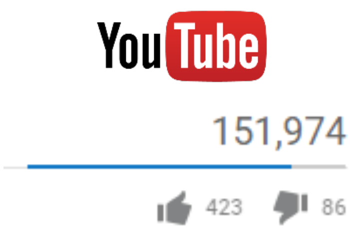
- He, Ruining, and Julian McAuley. "VBPR: Visual Bayesian Personalized Ranking from Implicit Feedback." In AAAI 2016.
- Chen, Jingyuan, Hanwang Zhang, Xiangnan He, Liqiang Nie, Wei Liu, and Tat-Seng Chua. "Attentive collaborative filtering: Multimedia recommendation with item- and component-level attention." In SIGIR 2017.
- Liu, Qiang, Shu Wu, and Liang Wang. "DeepStyle: Learning User Preferences for Visual Recommendation." In SIGIR 2017.
- Chen, Xu, Yongfeng Zhang, Hongteng Xu, Yixin Cao, Zheng Qin, and Hongyuan Zha. "Visually Explainable Recommendation." *arXiv preprint arXiv:1801.10288* (2018).
- Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E. Hinton. "Imagenet classification with deep convolutional neural networks." In NIPS 2012.
- Simonyan, Karen, and Andrew Zisserman. "Very deep convolutional networks for large-scale image recognition." In ICLR 2015.

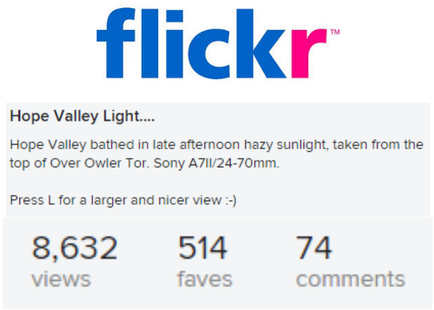
Outline of Tutorial

- Background (Xiangnan, 10 mins)
- Basics & Advances in Recommendation (Xiangnan, 50 mins)
- Visually-aware Product Recommendation (Xiangnan, 30 mins)
- Break (15 mins)
- Visual Representation (Hanwang, 45 mins)
- Image/Video Recommendation (Hanwang, 25 mins)
- Conclusion (Hanwang, 5 mins)

Social Multimedia Data

- ***Interaction with*** MM Content on Social Networks





淘宝网
Taobao.com

买家 [登录后可见]	淘宝价	数量	付款时间	款式和型号
就**0	¥58	1	2015-12-01 15:52:35	颜色分类:红蓝格纹
t**9	专属优惠	1	2015-12-01 15:51:04	颜色分类:红蓝格纹
相**8	专属优惠	1	2015-12-01 15:49:45	颜色分类:红蓝格纹

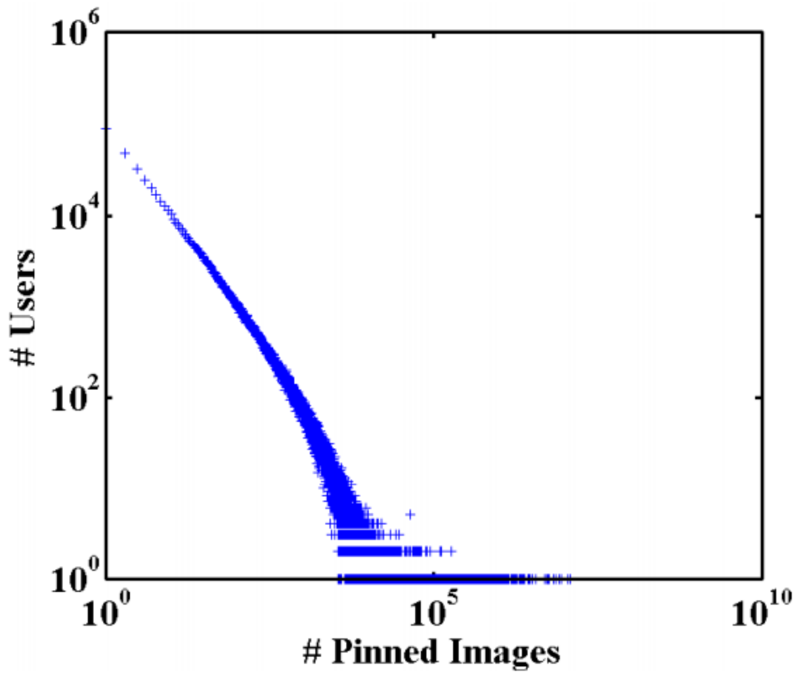
Slides: http://comp.nus.edu.sg/~xiangnan/icmr18-recsys.pdf

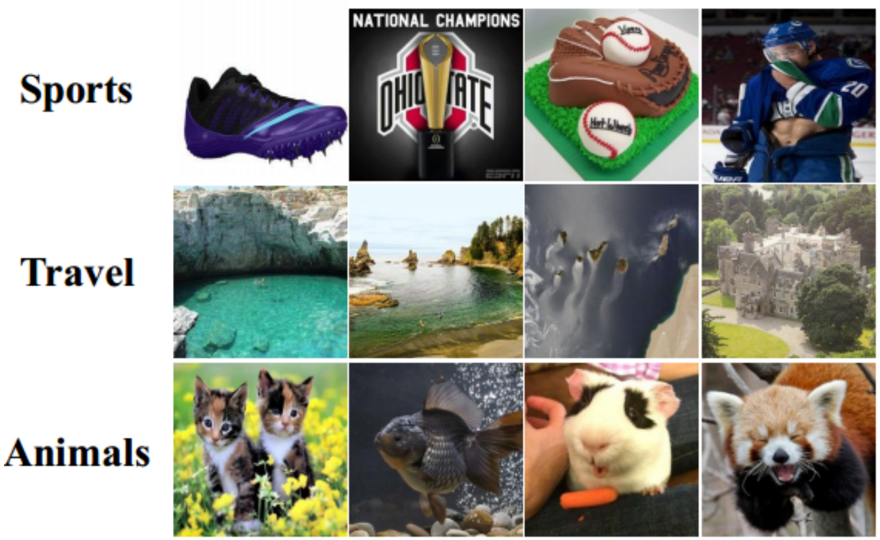
Challenges

- Large, sparse Image-User matrix

1M Flickr Image and 30K Tags: 99.7% sparsity

1M Pinterest Image and 10K Users: 99.1% sparsity





Sparse $\quad$ $X \approx U'V$ $\quad$ Diverse

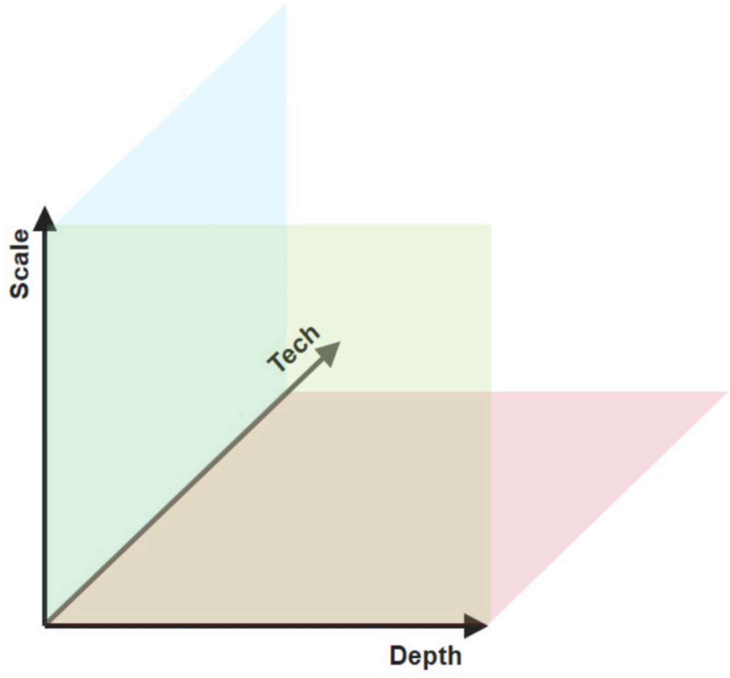
Slides: http://comp.nus.edu.sg/~xiangnan/icmr18-recsys.pdf

Outline

- State-of-the-art visual representations (45min)
  - Image, object feature
  - Video feature
  - Dynamic representation (visual attention)
- Deep embedding visual representation into user-item matrix (25min)
  - Weakly, semi-supervised learning

Slides: http://comp.nus.edu.sg/~xiangnan/icmr18-recsys.pdf

The Feature Evolution Space



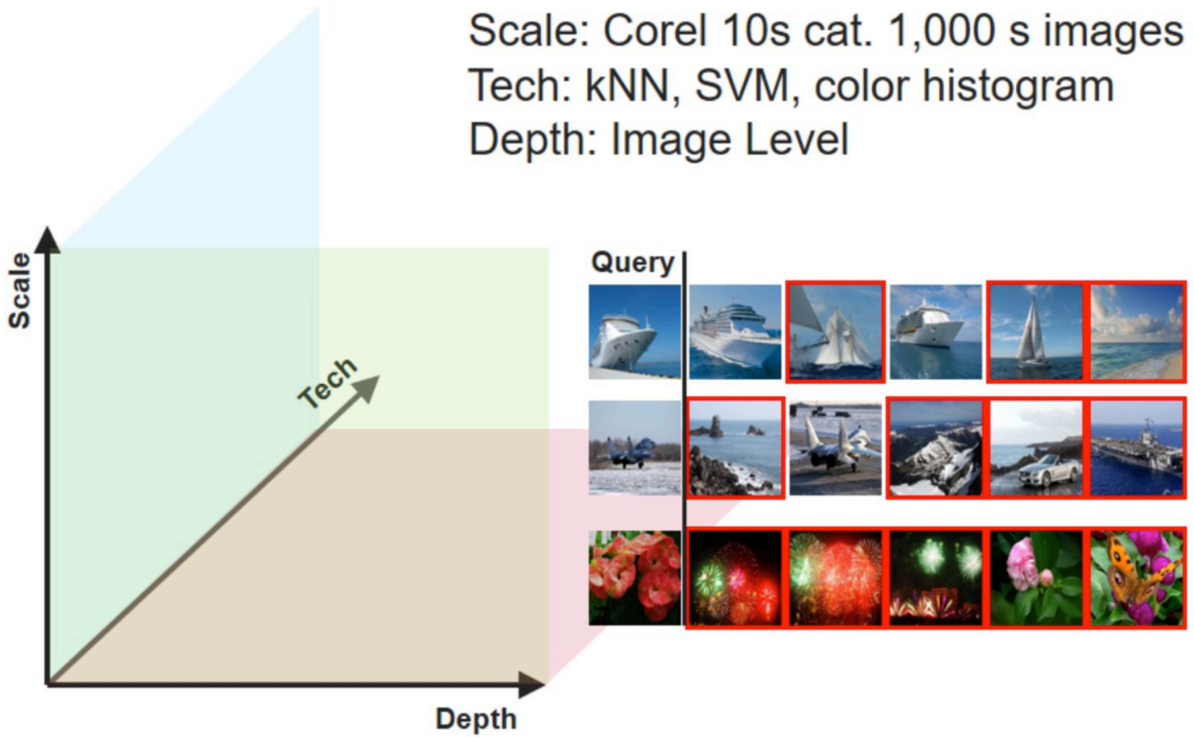
- Scale
  - Data, concept
- Depth
  - Content understanding
- Tech
  - Model

Slides: http://comp.nus.edu.sg/~xiangnan/icmr18-recsys.pdf

Near 2000: At the end of the early stage

Smeulders et al. TPAMI'00

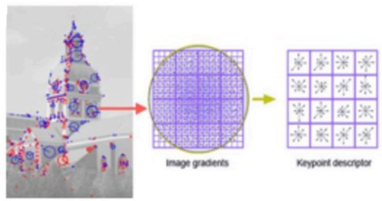
Scale: Corel 10s cat. 1,000 s images
Tech: kNN, SVM, color histogram
Depth: Image Level



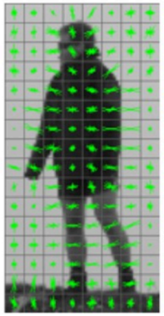
Slides: http://comp.nus.edu.sg/~xiangnan/icmr18-recsys.pdf

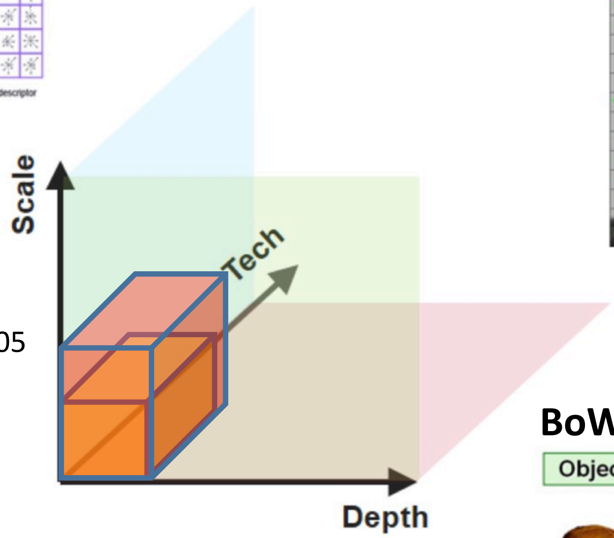
2000~2005: The Rise of Local Features

SIFT Lowe. IJCV'02

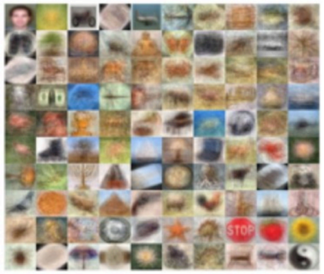


HoG Dalal&Triggs. CVPR'05

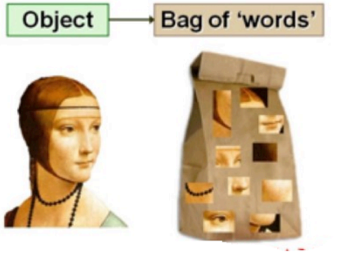




Caltech101 Fei-Fei. CVPR'05



BoW Sivic et al. ICCV'05



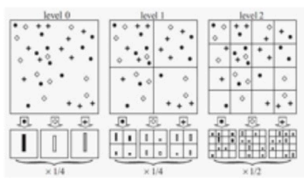
Slides: http://comp.nus.edu.sg/~xiangnan/icmr18-recsys.pdf

2005~2011: The Mature of Shallow Methods

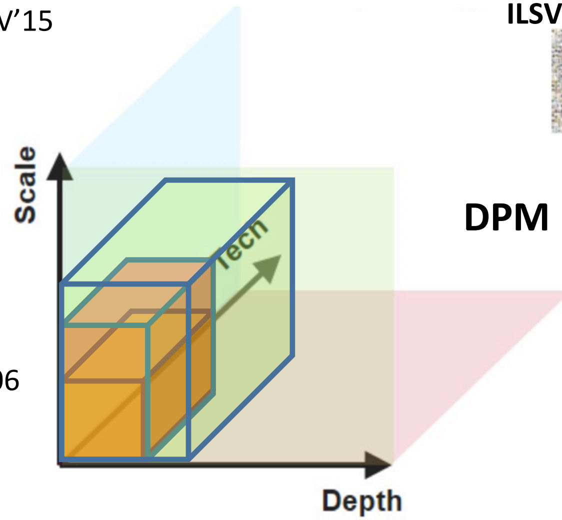
VOC Everingham et al. IJCV'15

Pattern Analysis, Statistical Modelling and Computational Learning

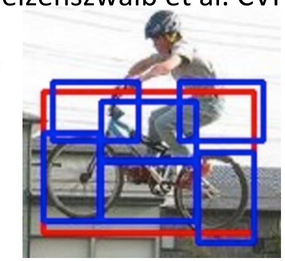
ILSVRC Deng et al. CVPR'09

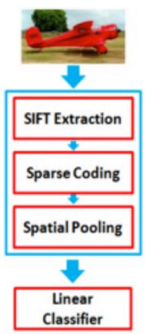


SPM Lazebnik et al. CVPR'06



DPM Felzenszwalb et al. CVPR'08





ScSP Yang et al. CVPR'09

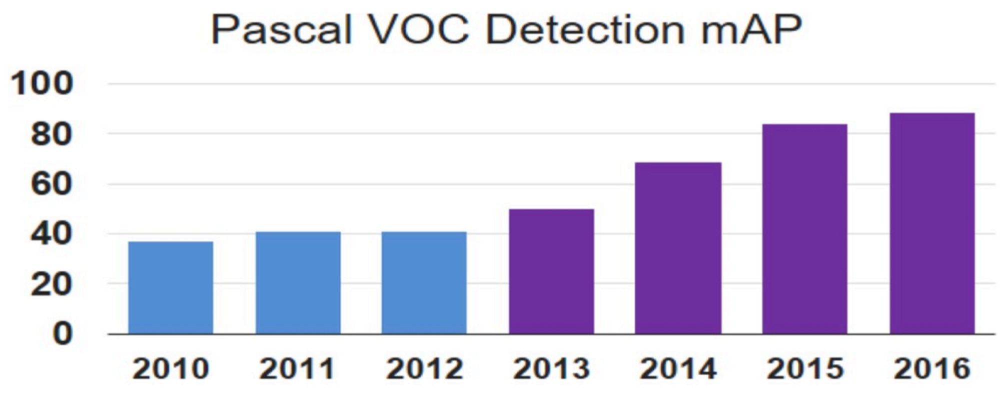
VLAD Jegou et al. CVPR'10

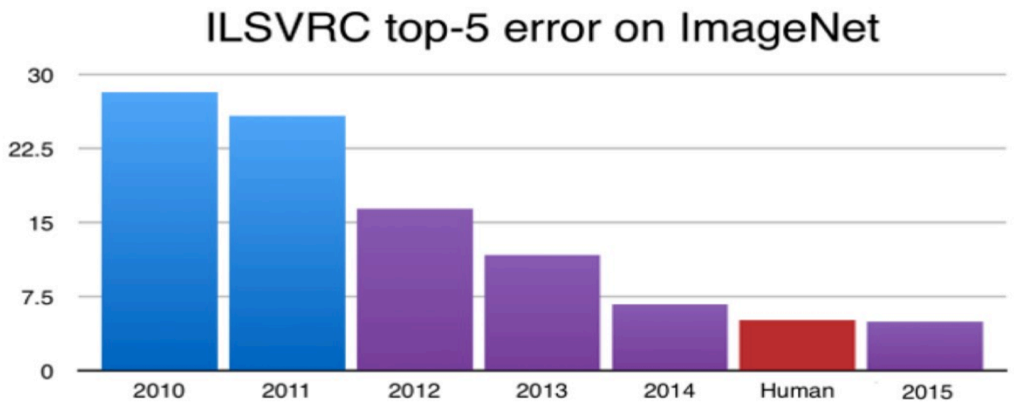
FisherVector Perronnin et al. ECCV'10

Tricks Chatfield et al. BMVC'11

Slides: http://comp.nus.edu.sg/~xiangnan/icmr18-recsys.pdf

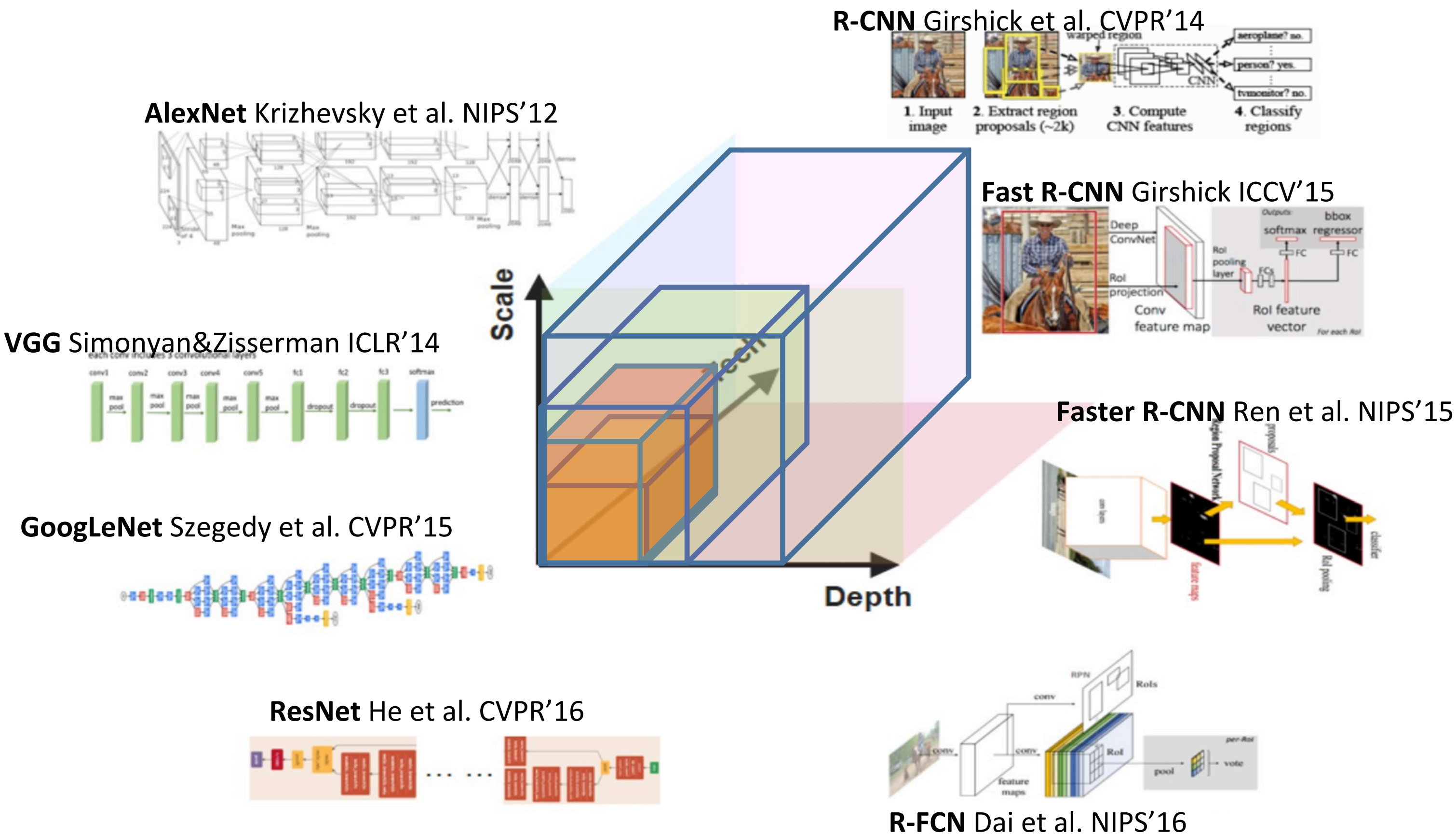
2012: The Revolution





Slides: http://comp.nus.edu.sg/~xiangnan/icmr18-recsys.pdf

2012~Now: The DL Republic

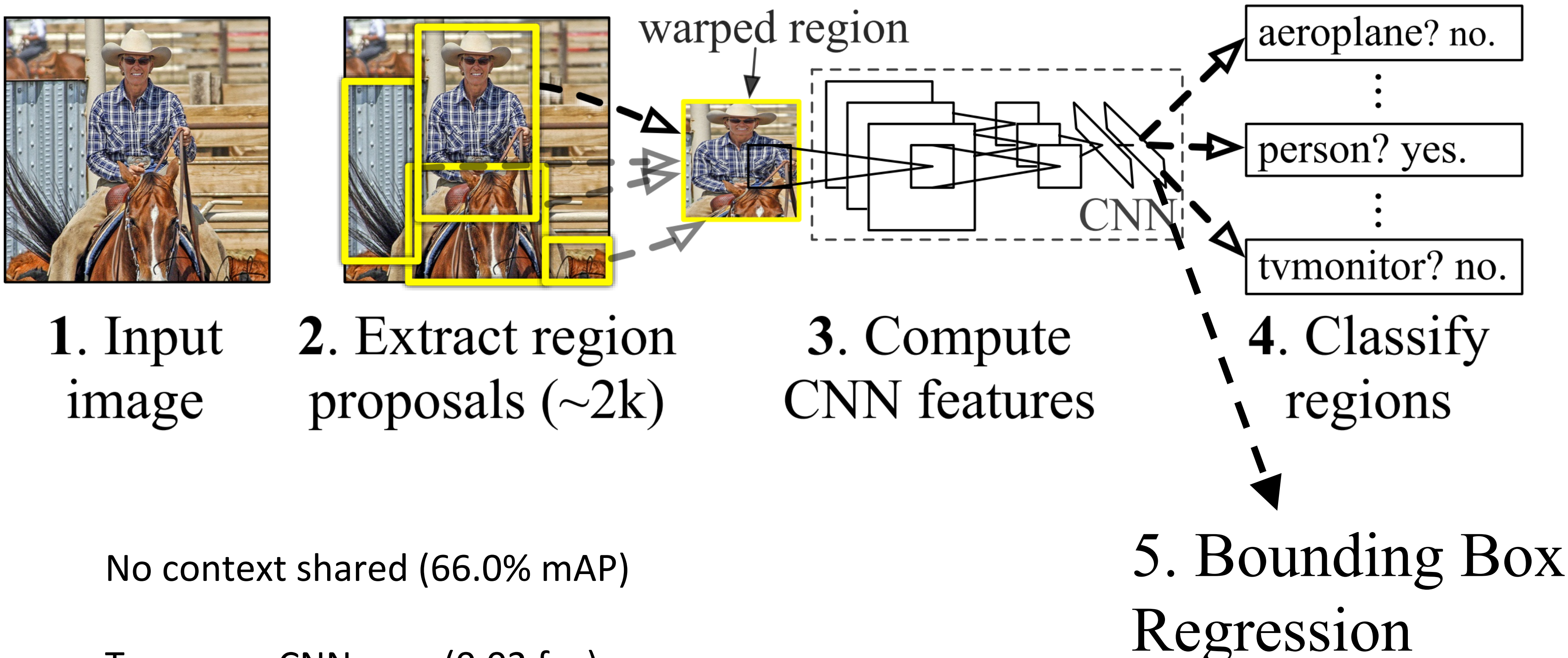


Slides: http://comp.nus.edu.sg/~xiangnan/icmr18-recsys.pdf

Object Detection

--- *On the way of sharing context features*

R-CNN [Girshick et al. CVPR'14]



No context shared (66.0% mAP)

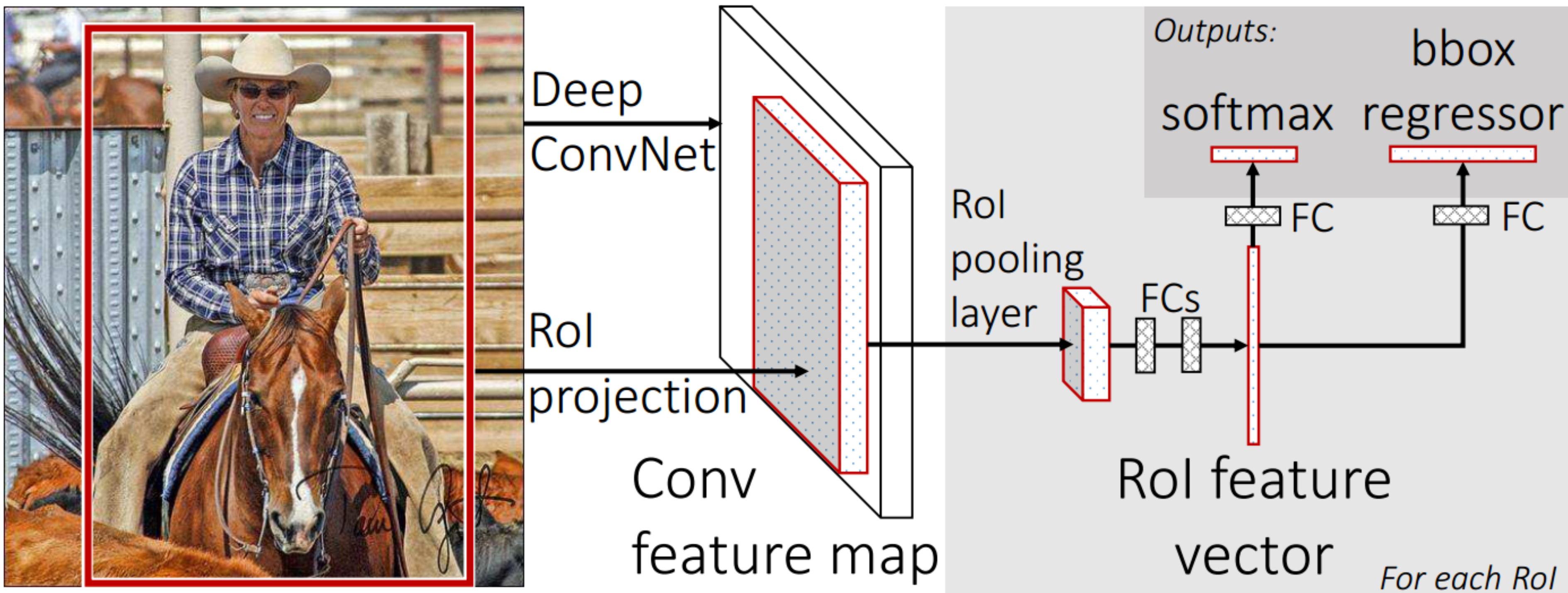
Too many CNN pass (0.02 fps)

Slides: http://comp.nus.edu.sg/~xiangnan/icmr18-recsys.pdf

Object Detection

--- *On the way of sharing context features*

Fast R-CNN [Girshick et al. ICCV'15]



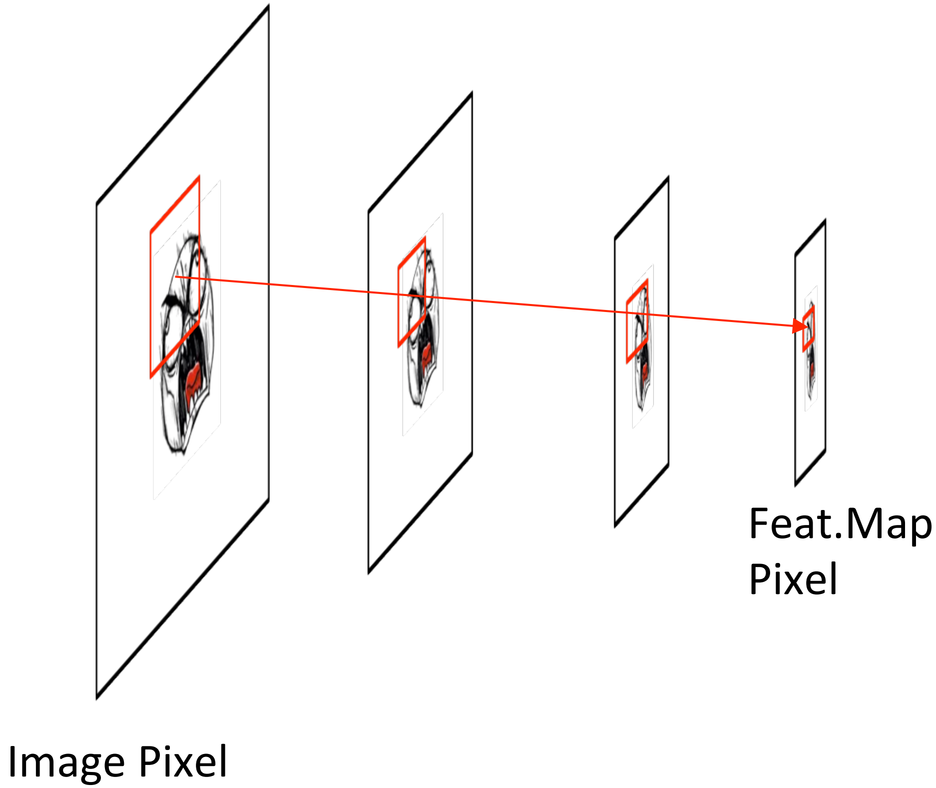
Few context shared (70.0% mAP)

Only one CNN pass but still requires 3rd party proposal (0.5 fps)

Slides: http://comp.nus.edu.sg/~xiangnan/icmr18-recsys.pdf

Object Detection

--- *On the way of sharing context features*



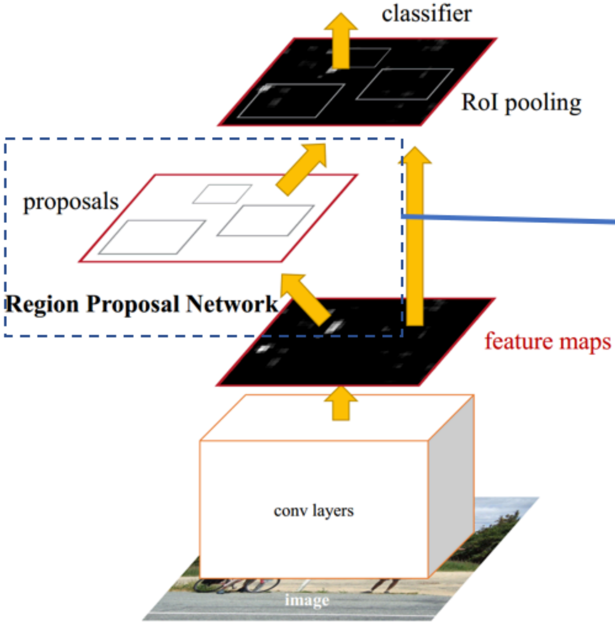
RoI pooling encapsulate context

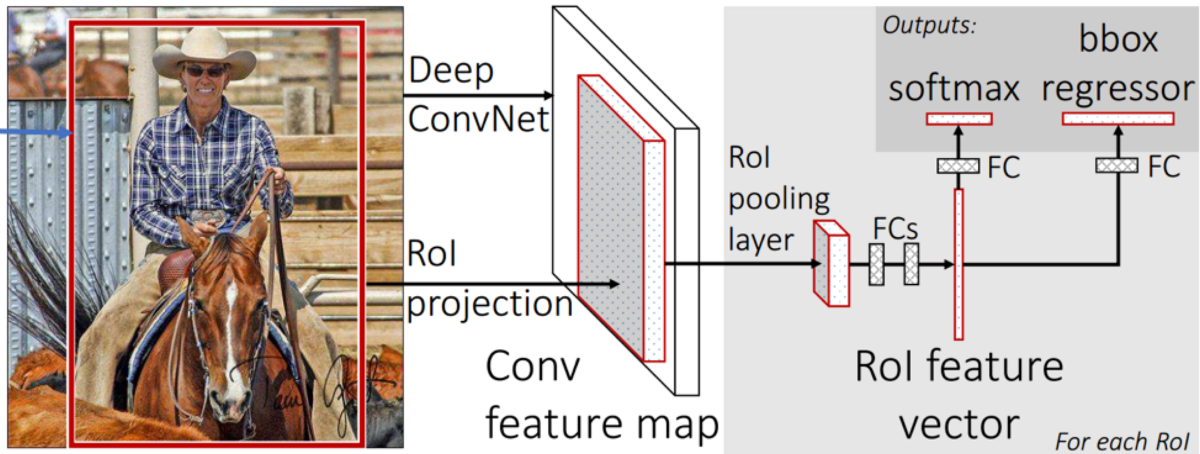
Slides: http://comp.nus.edu.sg/~xiangnan/icmr18-recsys.pdf

Object Detection

--- *On the way of sharing context features*

Faster R-CNN [Ren et al. NIPS'16]





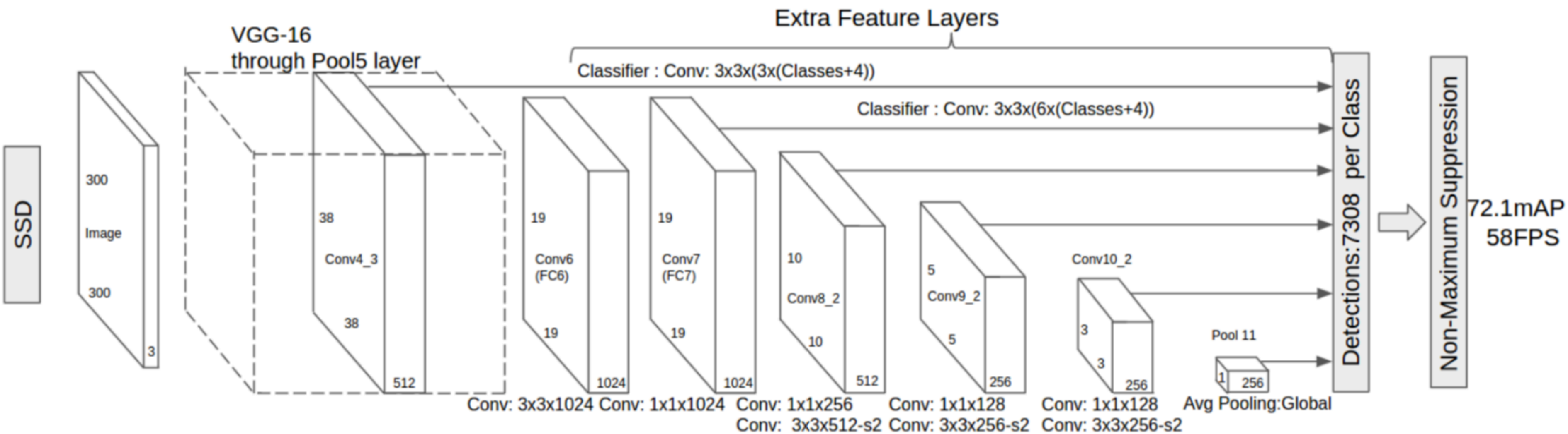
Context can help to know "where": conv to generate proposals (73.2% mAP, 5 fps)

Slides: http://comp.nus.edu.sg/~xiangnan/icmr18-recsys.pdf

Object Detection

--- *On the way of sharing context features*

SSD [Liu et al. ECCV'16] (see also YOLO9K [Redmon et al CVPR'17])



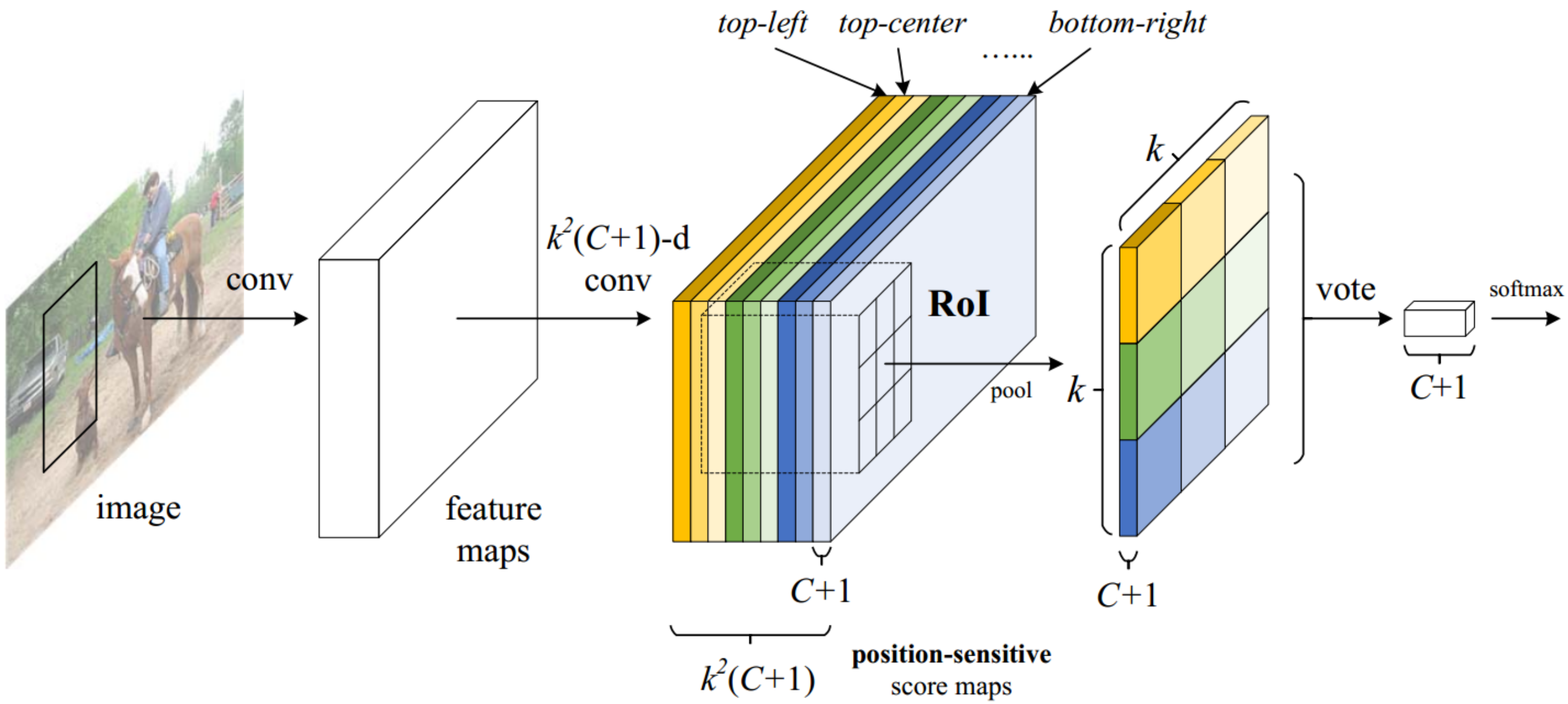
Context can help to know "where" and "what" simultaneously:
Conv at multi-layers to generate bbox and classes (75.1% mAP, 58 fps)

Slides: http://comp.nus.edu.sg/~xiangnan/icmr18-recsys.pdf

Object Detection

--- *On the way of sharing context features*

R-FCN[Dai et al. NIPS'16]



Context can help to know "where" and "what" **separately**:
Conv to generate bbox (RPN) and classes (RoI pooling)
80.5% mAP, 6fps

Slides: http://comp.nus.edu.sg/~xiangnan/icmr18-recsys.pdf

Message To-Go !

- Image feature
  - Apply CNN
  - Take the last layer output

- Object feature
  - Apply CNN-based object detector
  - Get the boxes (or masks)
  - Take the RoI feature who generates them

Slides: http://comp.nus.edu.sg/~xiangnan/icmr18-recsys.pdf

Video Representation

- The best hand-crafted: iDT
- The best spatial-temporal convolution: (2+1)D + Two-Stream
- The best choice: Frame Pooling

Body up?

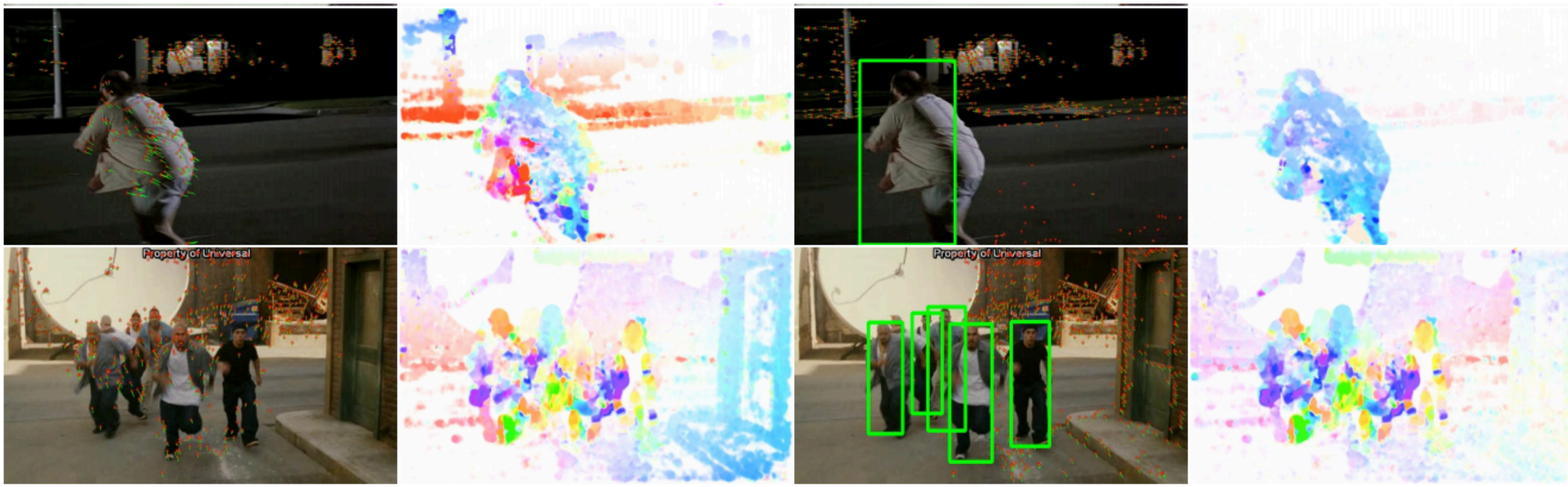
Kissed?

Weapon or arm forward?

Slides: http://comp.nus.edu.sg/~xiangnan/icmr18-recsys.pdf

Improved Dense Trajectories (iDT)

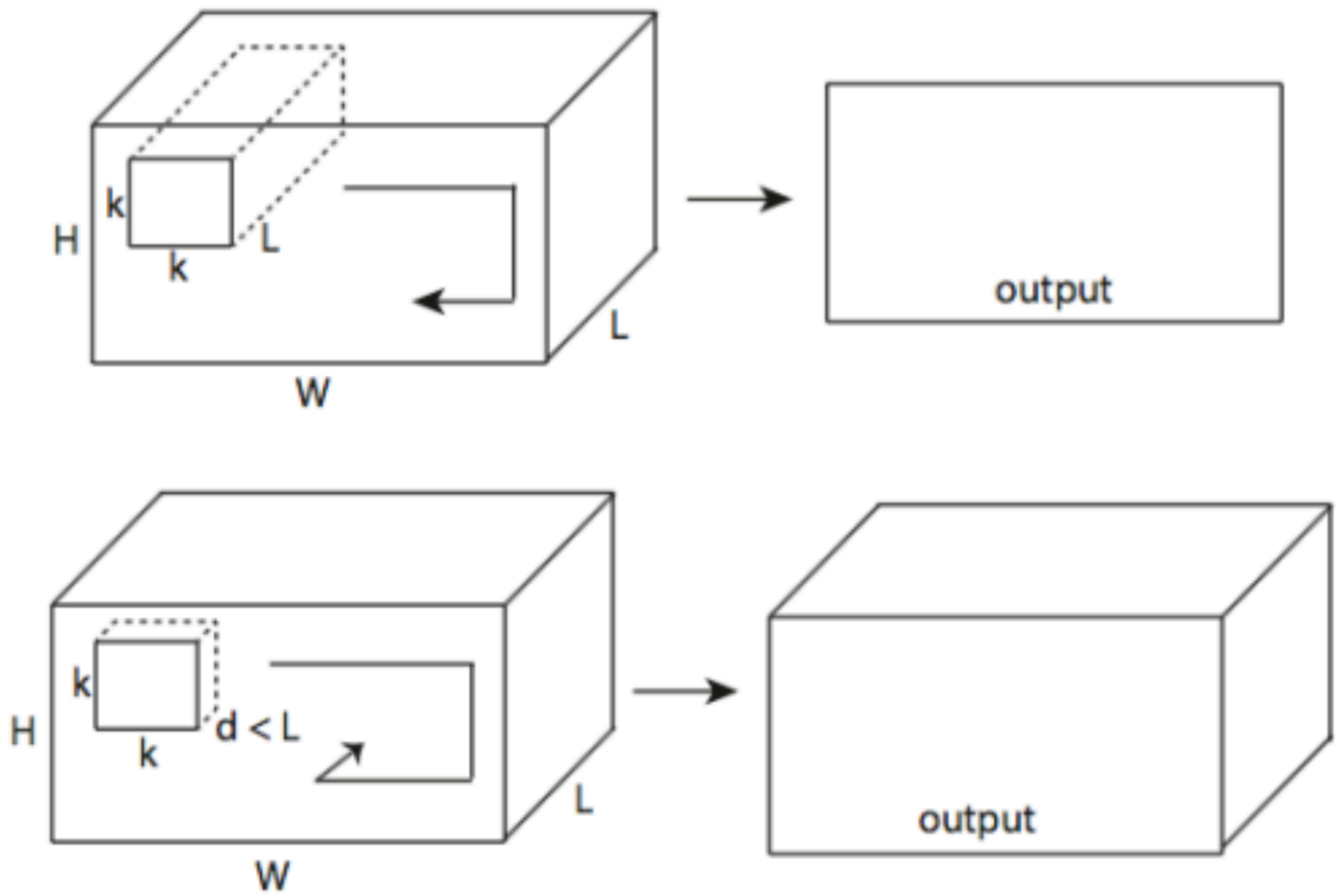
[Wang & Schimid. ICCV'13]



- Trajectory feature point + HOG descriptor + Fisher Vector Encoding + BoW ≅ 10K dimensions
- TRECVID MED challenge 2013 and THUMOS'13 action recognition challenge winners
- Slow and high-dimensional

Slides: http://comp.nus.edu.sg/~xiangnan/icmr18-recsys.pdf

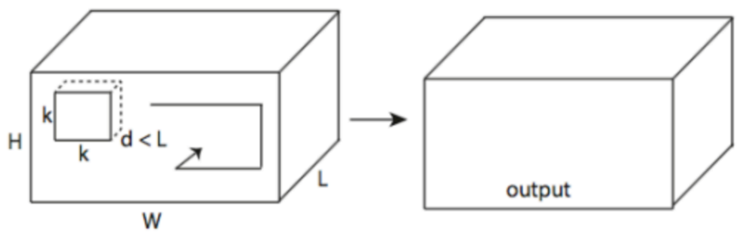
Spatial-Temporal Convolution



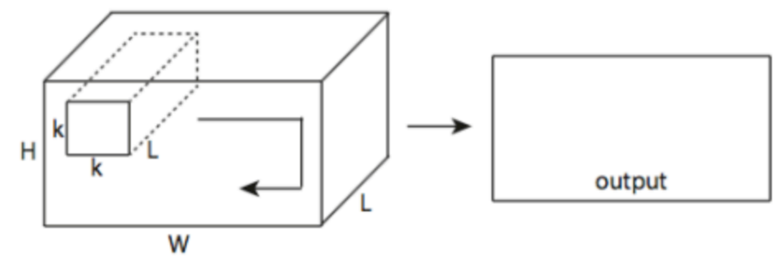
Slides: http://comp.nus.edu.sg/~xiangnan/icmr18-recsys.pdf

Pre-Training for Features

Sports 1M [Karpathy et al. CVPR'14]
Videos: 1.1M Class: 487



ImageNet [Russakovsky et al. IJCV'15]
Images: 1.2M Class: 1000

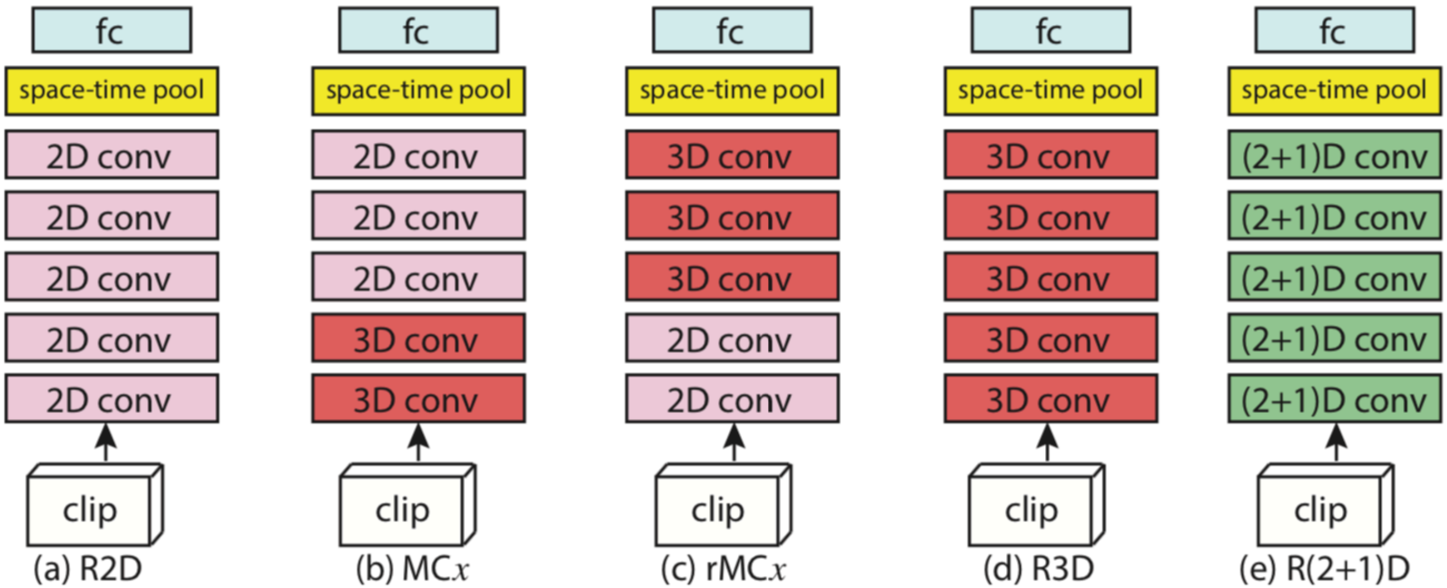


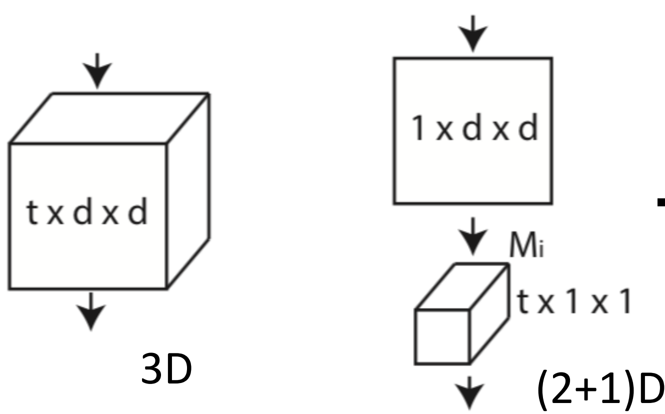
Boostrapping 3D from 2D [Mansimov et al. arXiv'15]
Boring video from ImageNet + Target video set

Slides: http://comp.nus.edu.sg/~xiangnan/icmr18-recsys.pdf

More ST Conv. Designs

[Tran et al. CVPR'18]





→ More nonlinear
Converge faster

Slides: http://comp.nus.edu.sg/~xiangnan/icmr18-recsys.pdf

More ST Conv. Designs

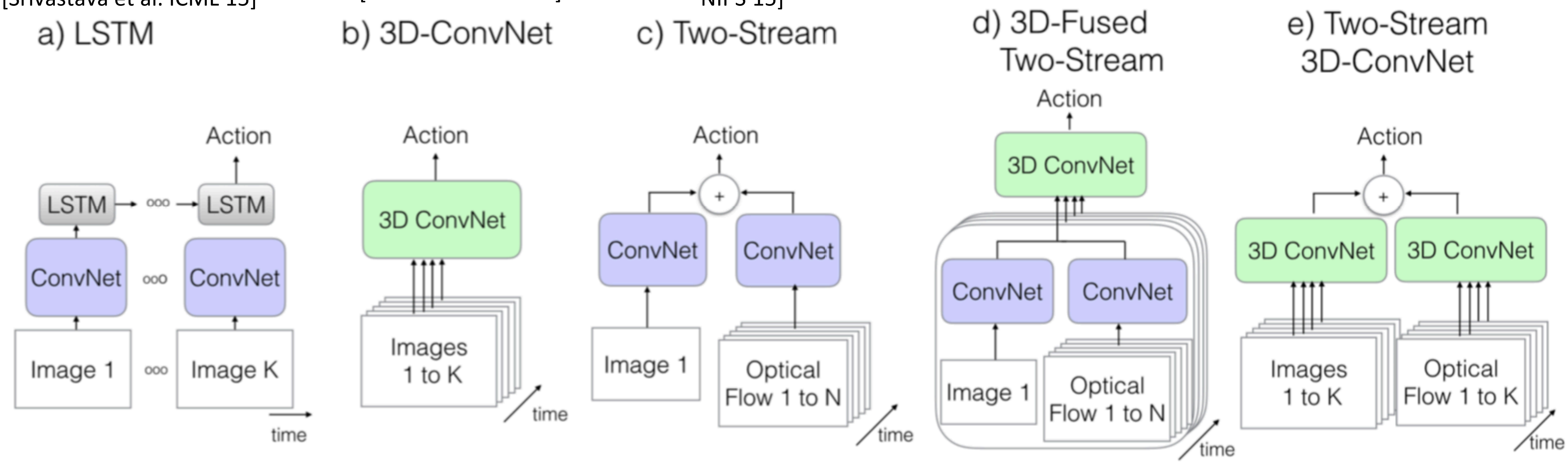
[Donahue et al. CVPR'15]
[Srivastava et al. ICML'15]

[Tran et al. CVPR'15]

[Simonyan&Zisserman. NIPS'15]

[Feichtenhofer et al. CVPR'16]

[Carreira &Zisserman. CVPR'17]



Slides: http://comp.nus.edu.sg/~xiangnan/icmr18-recsys.pdf

Message To-Go!

Performances on Sports 1M

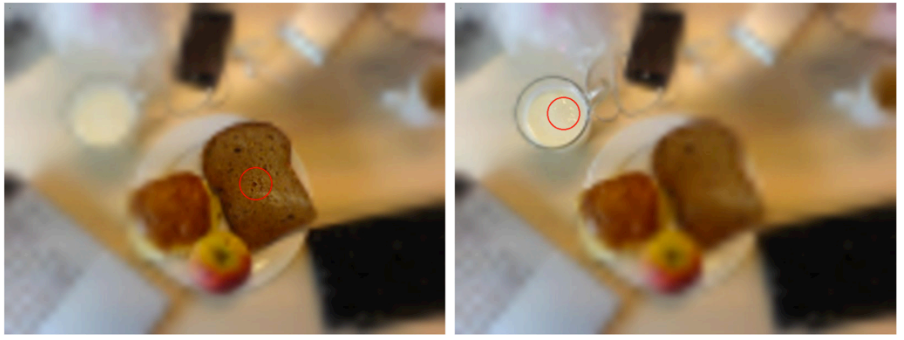
Method	C3D	2D+ Pooling	(2+1) D	Two-Stream	(2+1)D+ Two-Stream
Video@5	85.2%	90.4%	91.5%	87.4%	91.9%

- Image-level is very strong
- Video-level requires tricky pre-training
- If you are lazy, just use image-level+pooling

Slides: http://comp.nus.edu.sg/~xiangnan/icmr18-recsys.pdf

Visual Attention
--- dynamic visual feature

- Inspired from cognitive science, visual saliency is a special case with prior.



[Jiang et al. CVPR'16]

- Pioneer works on signal band-width savings[Mnih et al. NIPS'15]
- Great success in Machine Translation [Bahdanau et al. ICLR'15] and Image Captioning [Xu et al. ICML'15].

Slides: http://comp.nus.edu.sg/~xiangnan/icmr18-recsys.pdf

Dynamic Feature

Slides: http://comp.nus.edu.sg/~xiangnan/icmr18-recsys.pdf

Question?

Slides: http://comp.nus.edu.sg/~xiangnan/icmr18-recsys.pdf

Generic Formulations

- Get a (local, partial) feature $\boldsymbol{X_i}$
- Get a contextual information (env.) $\boldsymbol{h}$
- Calculate an attention probability $\boldsymbol{p_i} \propto F(\boldsymbol{X_i}, \boldsymbol{h})$
- Calculate the new feature $\boldsymbol{X}$
  - *Probabilistic (hard):* $\boldsymbol{X} =$ ***Monte Carlo*** $(\boldsymbol{p_i})$
  - *Deterministic (soft):* $\boldsymbol{X} = \sum \boldsymbol{p_i X_i}$

Slides: http://comp.nus.edu.sg/~xiangnan/icmr18-recsys.pdf

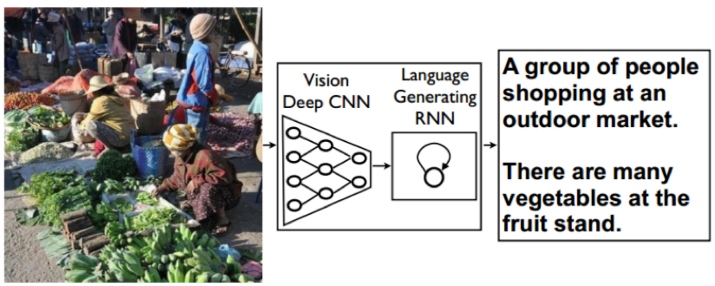
Notes

- Attention is not just "weighted sum" (soft), it is originally "discrete visual policy" (hard).
- Reinforcement learning is its natural solution.

Slides: http://comp.nus.edu.sg/~xiangnan/icmr18-recsys.pdf

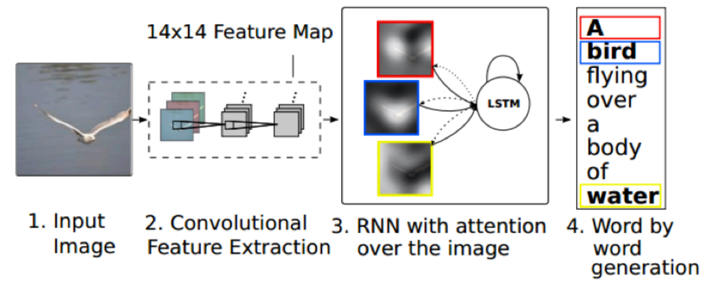
Shallow Visual Attention

GoogleNIC (Vinyals et al. 2014)



Encoder (Image→CNN→Vector) → ***Decoder*** (Vector→Word Seq.)

Attention (Xu et al. 2015)



Encoder (Image→CNN→ Dynamic Vector)→***Decoder*** (Dynamic Vector →Word Seq.)

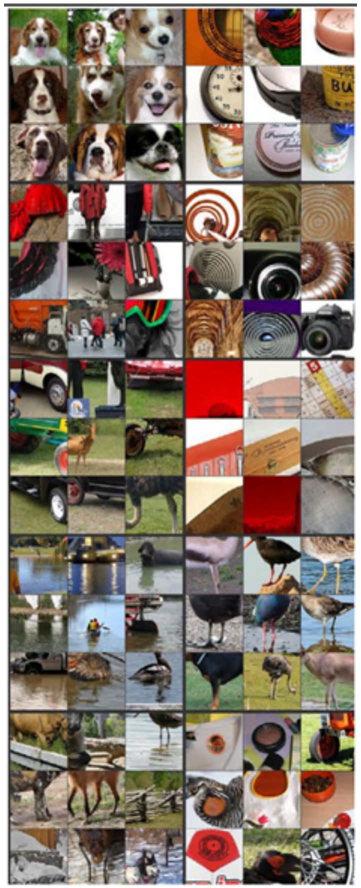
Limitations

1. Only spatial attention. What about various feature maps?
2. Only single layer. What about hierarchical attention?

Slides: http://comp.nus.edu.sg/~xiangnan/icmr18-recsys.pdf

CNN Revisit [Zeiler and Fergus. ECCV'14]

Layer3



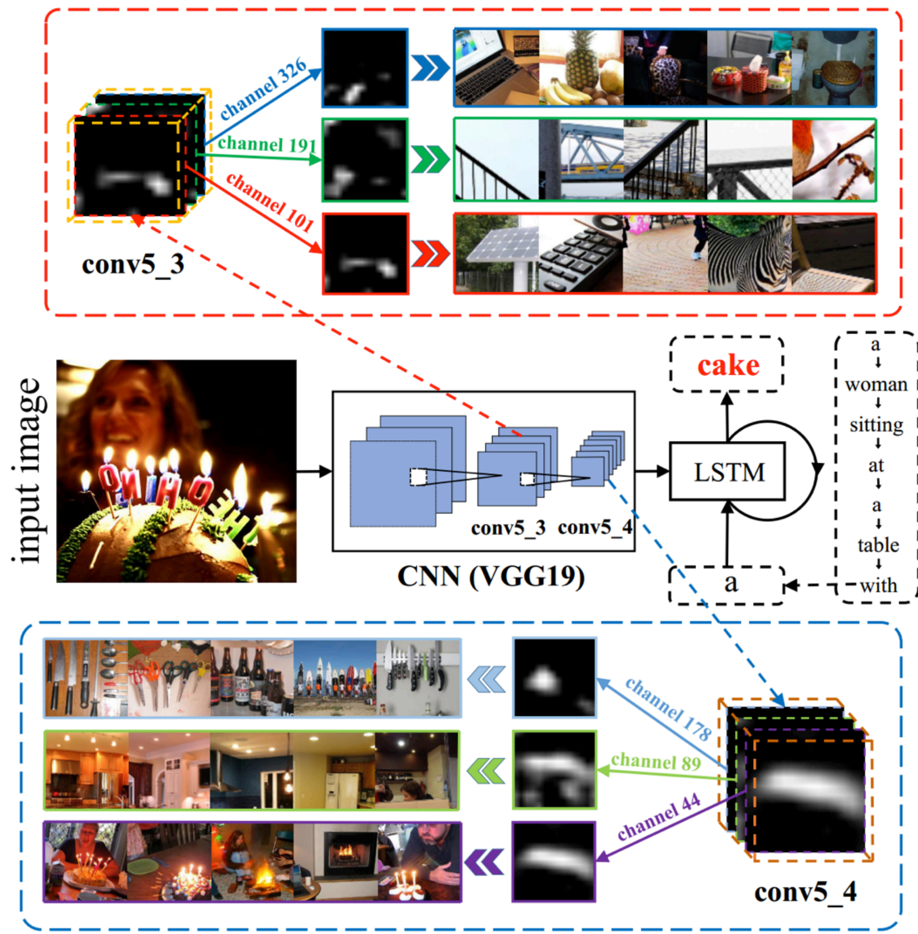
Layer4

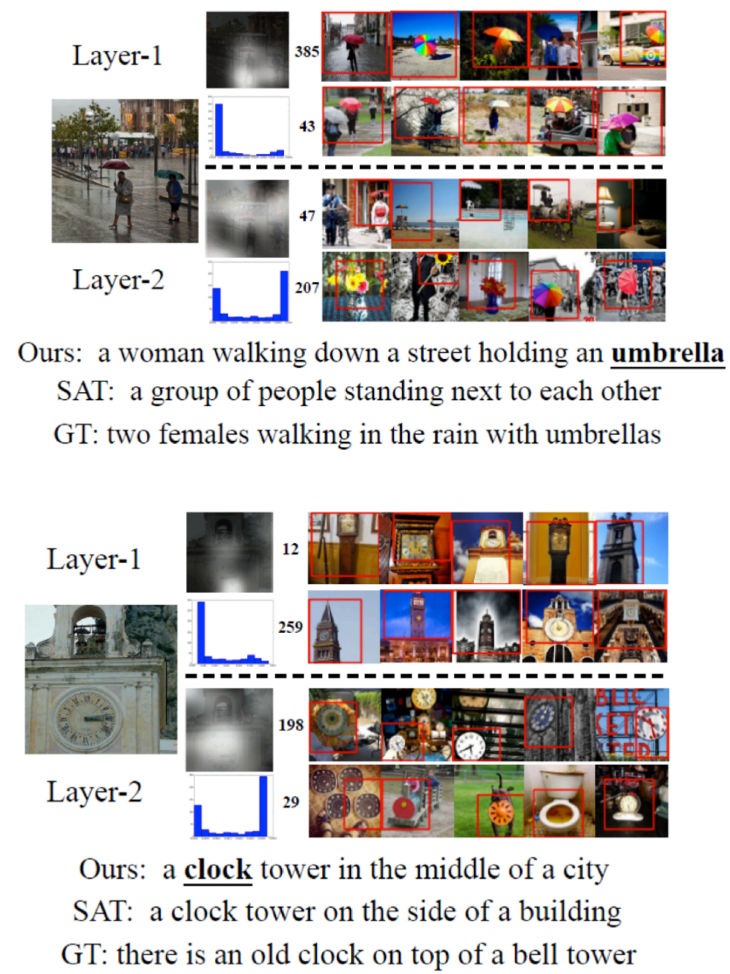
Layer5

Slides: http://comp.nus.edu.sg/~xiangnan/icmr18-recsys.pdf

Spatial- and Channel-wise Attention

[Chen et al. CVPR'17]

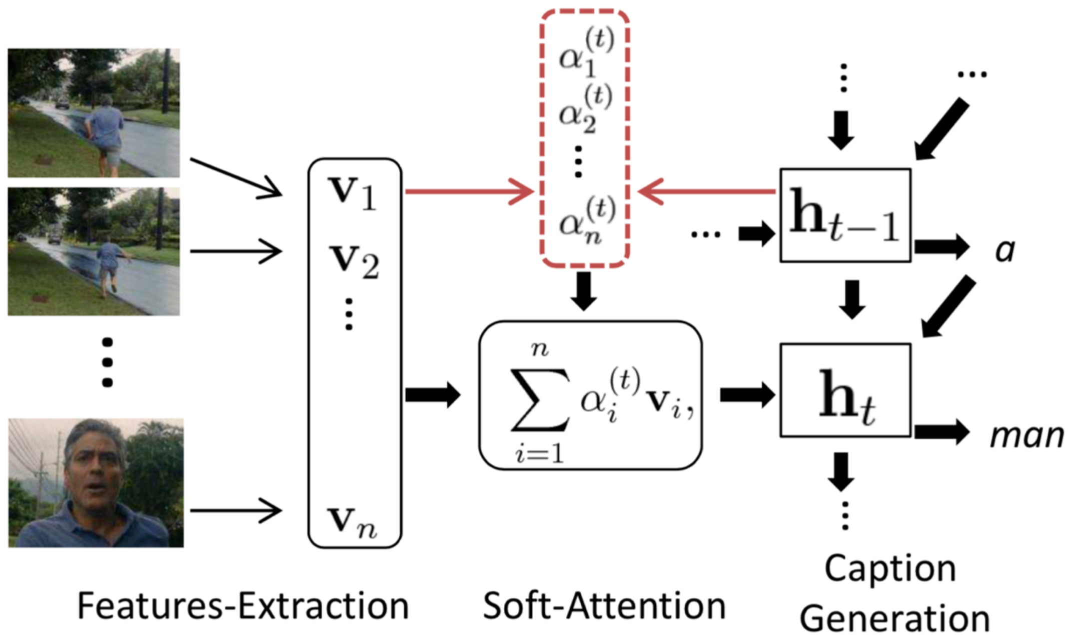




Slides: http://comp.nus.edu.sg/~xiangnan/icmr18-recsys.pdf

Temporal Attention in Video

[Yao et al. ICCV'15]



Slides: http://comp.nus.edu.sg/~xiangnan/icmr18-recsys.pdf

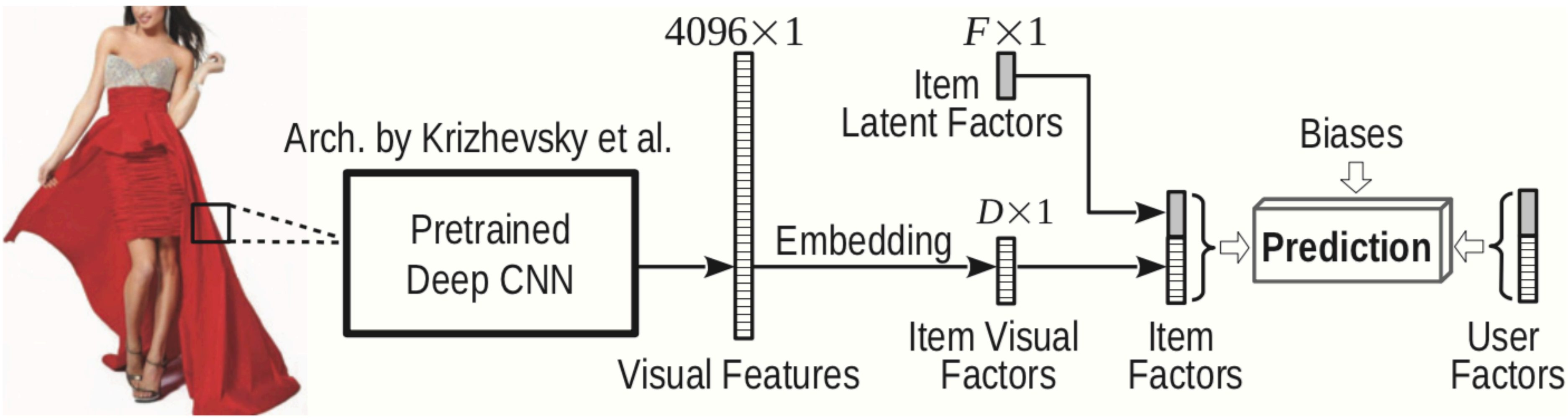
Outline

- State-of-the-art visual representations (35min)
  – Image, object feature
  – Video feature
  – Dynamic representation (visual attention)
- Deep embedding visual representation into user-item matrix (35min)
  – Weakly, semi-supervised learning

Slides: http://comp.nus.edu.sg/~xiangnan/icmr18-recsys.pdf

How to embed features into CF?

Visual BPR [He & McAuley. AAAI'16]



Slides: http://comp.nus.edu.sg/~xiangnan/icmr18-recsys.pdf

How to embed features into CF?

- Van den Oord et all. Deep content-based music recommendation. *NIPS*'13
- Geng et al. One of a Kind: User Profiling by Social Curation. MM'14
- Geng et al. Learning Image and User Features for Recommendation in Social Networks. ICCV'15
- Lei et al. Comparative deep learning of hybrid representations for image recommendations. CVPR'16
- Chen et al. Attentive Collaborative Filtering. SIGIR'17
- Gao et al. A Unified Personalized Video Recommendation via Dynamic Recurrent Neural Networks. MM'17
- Niu et al. Neural Personalized Ranking for Image Recommendation. WSDM'18
- Yu et al. Aesthetic-based Clothing Recommendation. WWW'18

Slides: http://comp.nus.edu.sg/~xiangnan/icmr18-recsys.pdf

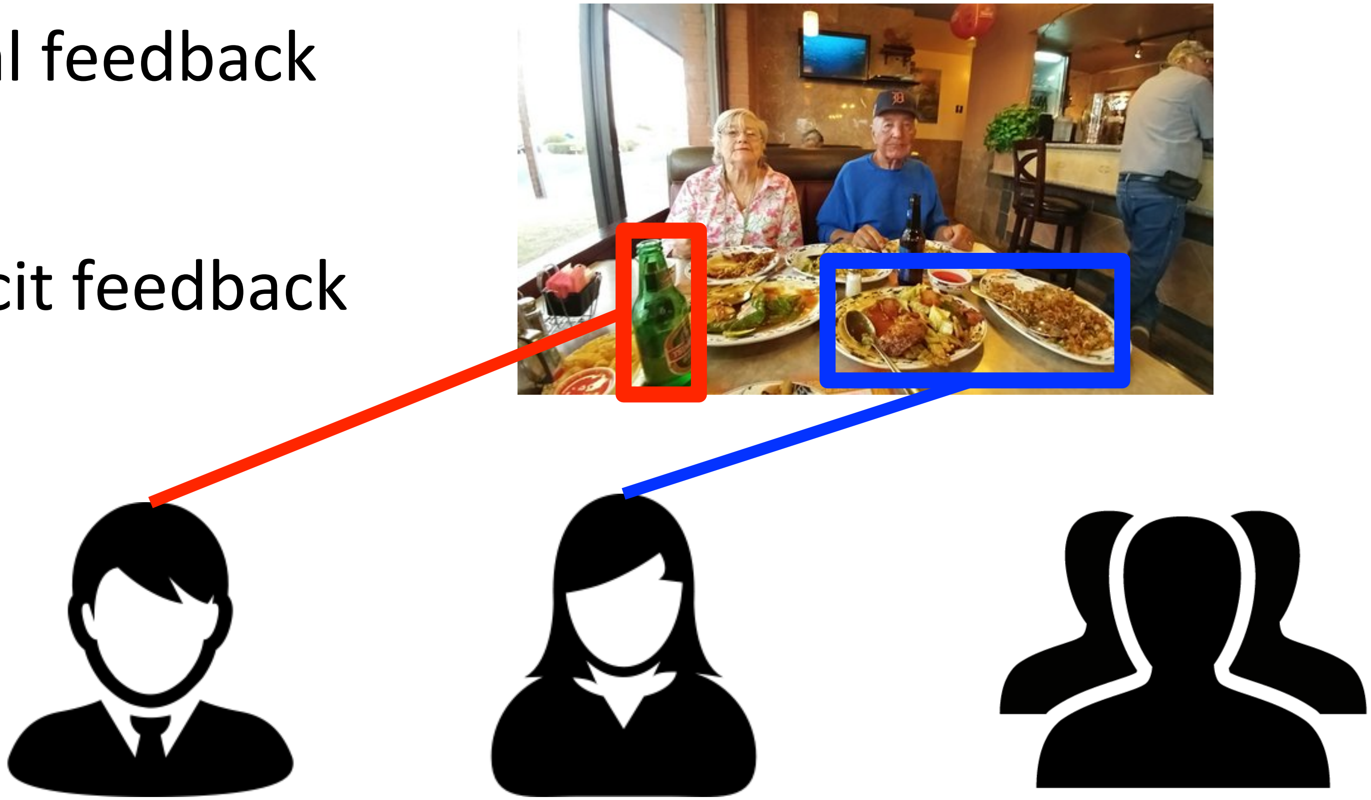
How to embed features into CF?

- Van den Oord et all. Deep content-based music recommendation. *NIPS*'13
- Geng et al. One of a Kind: User Profiling by Social Curation. MM'14
- Geng et al. Learning Image and User Features for Recommendation in Social Networks. ICCV'15
- Lei et al. Comparative deep learning of hybrid representations for image recommendations. CVPR'16
- Chen et al. Attentive Collaborative Filtering. SIGIR'17
- Gao et al. A Unified Personalized Video Recommendation via Dynamic Recurrent Neural Networks. MM'17
- Niu et al. Neural Personalized Ranking for Image Recommendation. WSDM'18
- Yu et al. Aesthetic-based Clothing Recommendation. WWW'18

Slides: http://comp.nus.edu.sg/~xiangnan/icmr18-recsys.pdf

Nature: Weakly Semi-Supervised Learning

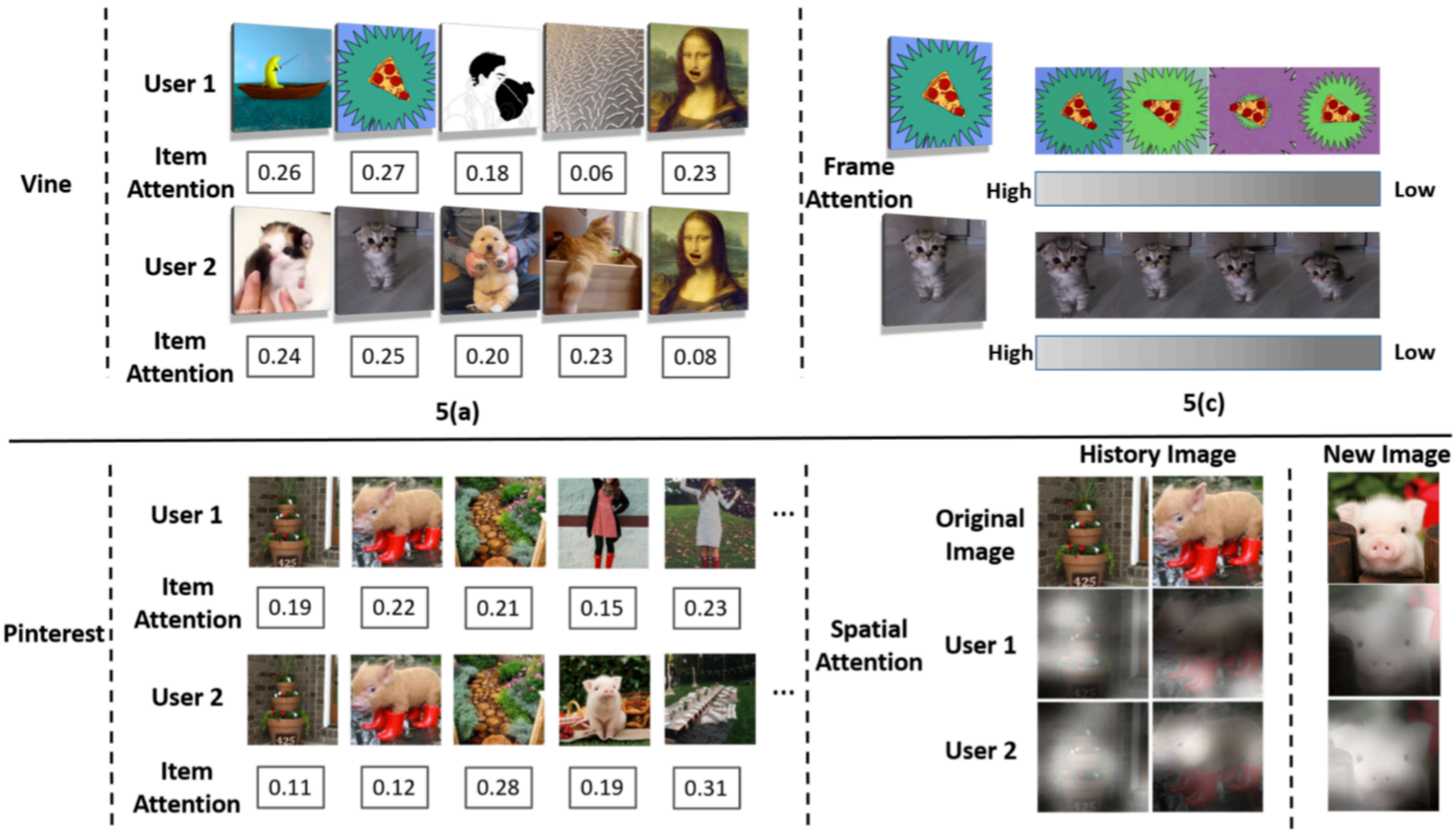
- Weak
  - Partial feedback
- Semi-
  - Implicit feedback



Slides: http://comp.nus.edu.sg/~xiangnan/icmr18-recsys.pdf

Towards "Weakly": Attentive CF

[Chen et al. SIGIR'17]



Slides: http://comp.nus.edu.sg/~xiangnan/icmr18-recsys.pdf

ACF: Attentive Collaborative Filtering (Chen et al, SIGIR'17)

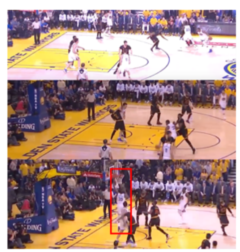
- **Input:**

  user -> ID & historical interacted items.

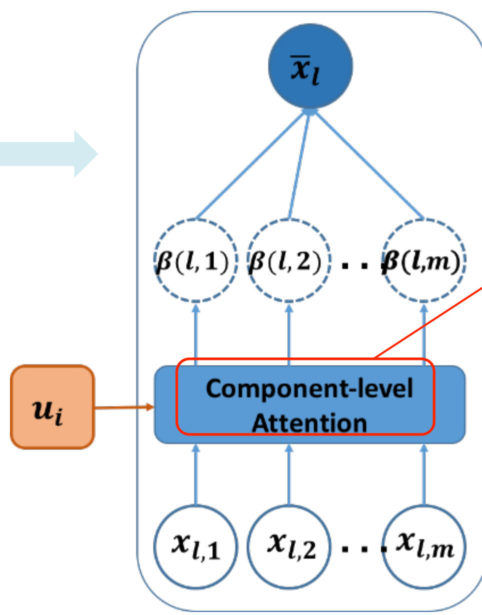
  Item -> ID & visual features.

- **Item Representation**:

  Component-level attention -> not all components contribute equally to an item's representation



A user's preference on different components of the item are not equal!



$$\overline{\mathbf{x}}_l = \sum_{m=1}^{|\{\mathbf{x}_{l*}\}|} \beta(i,l,m) \cdot \mathbf{x}_{lm}$$

$$\beta(i,l,m) = \frac{exp\,(b(i,l,m))}{\sum_{n=1}^{|\{\mathbf{x}_{l*}\}|} exp\,(b(i,l,n))}$$

$$b(i,l,m) = \mathbf{w}_2^T \phi(\mathbf{W}_{2u}\mathbf{u}_i + \mathbf{W}_{2x}\mathbf{x}_{lm} + \mathbf{b}_2) + \mathbf{c}_2$$

Slides: http://comp.nus.edu.sg/~xiangnan/icmr18-recsys.pdf

ACF: Attentive Collaborative Filtering (Chen et al, SIGIR'17)

- **Input:**

  user -> ID & historical interacted items.

  item -> ID & visual features.

- **User Presentation:**
  - Item-level attention -> not all historical items contribute equally to a user's representation

$$\hat{R}_{ij} = \left(\mathbf{u}_i + \sum_{l \in \mathcal{R}(i)} \alpha(i,l)\mathbf{p}_l\right)^T \mathbf{v}_j$$

Attention weights learned by a neural net ⬄ Attentive SVD++ model.

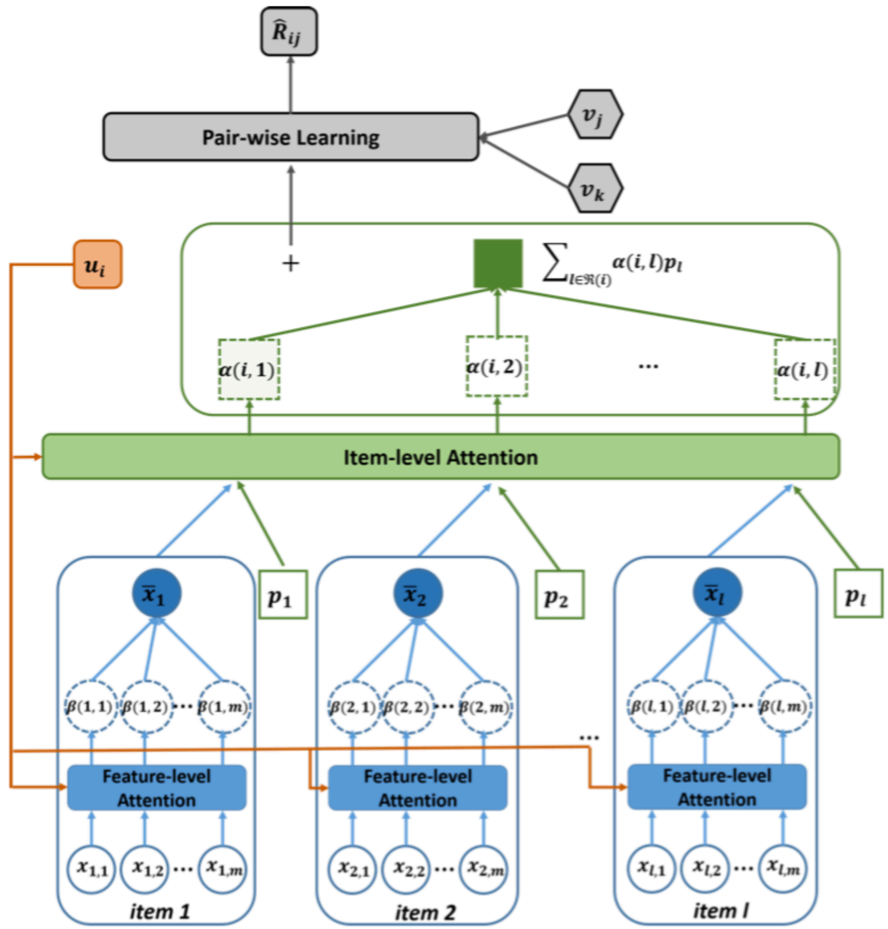
A user's preference on different items of history are not equal!

Slides: http://comp.nus.edu.sg/~xiangnan/icmr18-recsys.pdf

Attentive CF

$$\arg\min_{\mathbf{U},\mathbf{V},\mathbf{P},\Theta} \sum_{(i,j,k)\in\mathcal{R}_B} -\ln\sigma\left\{\left(\mathbf{u}_i + \sum_{l\in\mathcal{R}(i)} \alpha(i,l)\mathbf{p}_l\right)^T \mathbf{v}_j - \left(\mathbf{u}_i + \sum_{l\in\mathcal{R}(i)} \alpha(i,l)\mathbf{p}_l\right)^T \mathbf{v}_k\right\} + \lambda(||\mathbf{U}||^2 + ||\mathbf{V}||^2 + ||\mathbf{P}||^2),$$

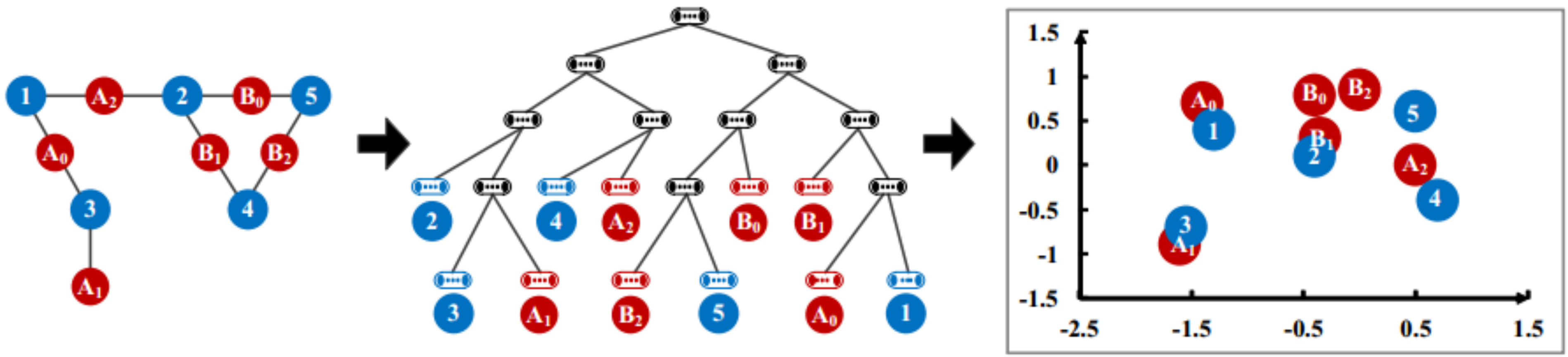
$$\hat{R}_{ij} = \overbrace{\mathbf{u}_i^T\mathbf{v}_j}^{latent\ factor\ model} + \underbrace{\sum_{l\in\mathcal{R}(i)} \alpha(i,l)\mathbf{p}_l^T\mathbf{v}_j}_{neighborhood\ model},$$



Slides: http://comp.nus.edu.sg/~xiangnan/icmr18-recsys.pdf

Towards "Semi-": Deep Tree-CF

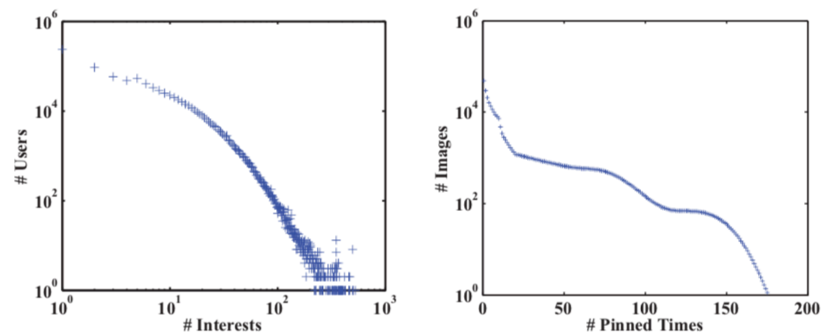
[Geng et al. ICCV'15]



(a) (b) (c)

$$J = \frac{1}{2m} \sum_{ij} \left(\mathbf{G}_{ij} - \frac{d_i d_j}{2m} \right) \delta(i,j)$$

0-1"likes" Avg"likes"

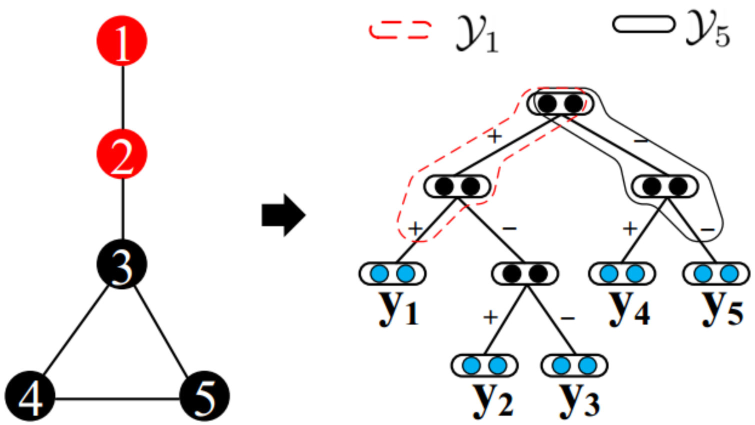


Pinterest: 100K users, 1M images

Slides: http://comp.nus.edu.sg/~xiangnan/icmr18-recsys.pdf

A Closer Look

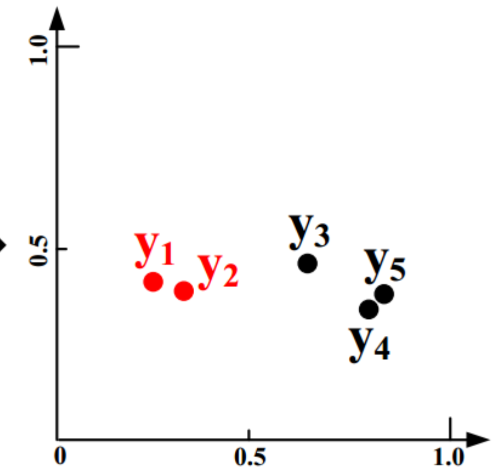
[Zhang et al. TOMM'16]



(a) (b)

Model	Feature
$\mathbf{y}_1$	$\mathcal{Y}_1$ $\mathcal{Y}_2$
$\mathbf{y}_2$	$\mathcal{Y}_1$ $\mathcal{Y}_2$ $\mathcal{Y}_3$
$\mathbf{y}_3$	$\mathcal{Y}_2$ $\mathcal{Y}_3$ $\mathcal{Y}_4$ $\mathcal{Y}_5$
$\mathbf{y}_4$	$\mathcal{Y}_3$ $\mathcal{Y}_4$ $\mathcal{Y}_5$
$\mathbf{y}_5$	$\mathcal{Y}_3$ $\mathcal{Y}_4$ $\mathcal{Y}_5$

(c)



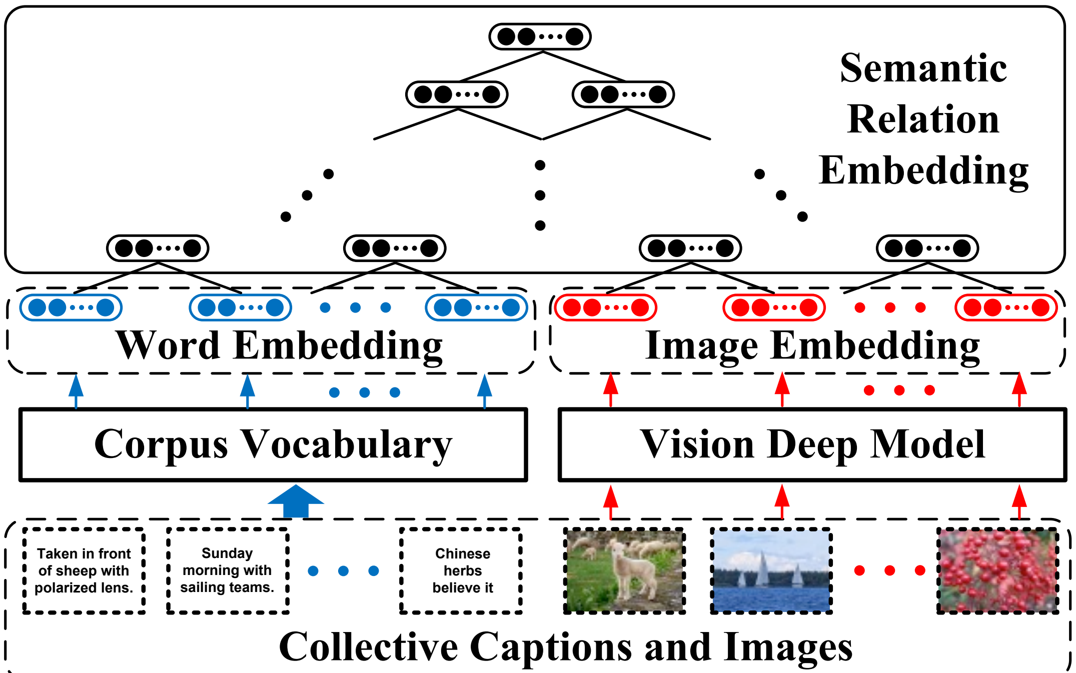
(d)

$$\min_{\mathbf{Y}} \sum_{ij} S_{ij} \log p(\mathbf{y}_i|\mathbf{y}_j)$$

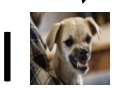
Instead of ***$\langle y_i, y_j \rangle$***, we use ***$\langle y_i, Y_j \rangle + \langle y_j, Y_i \rangle$***

Slides: http://comp.nus.edu.sg/~xiangnan/icmr18-recsys.pdf

Visualization by Image and Word(User)



$$\max_{\mathbf{Y}} \sum_{ij} S_{ij} \log p(\mathbf{y}_i|\mathbf{y}_j)$$

P("dog" | )

Flickr 1M Image-Caption Pairs
8K+ Words

Slides: http://comp.nus.edu.sg/~xiangnan/icmr18-recsys.pdf

Image-Word Embedding

Table 2: Performance (mAP%) of keyword-based retrieval on NUSWIDE and CCV.

Dataset/Method	S-CNN	R-Ours	HR-Ours	GoogLeNet
NUSWIDE	19.2	21.1	**24.3**	23.0
CCV	20.9	24.7	**36.2**	19.4

Slides: http://comp.nus.edu.sg/~xiangnan/icmr18-recsys.pdf

Word-Word Embedding

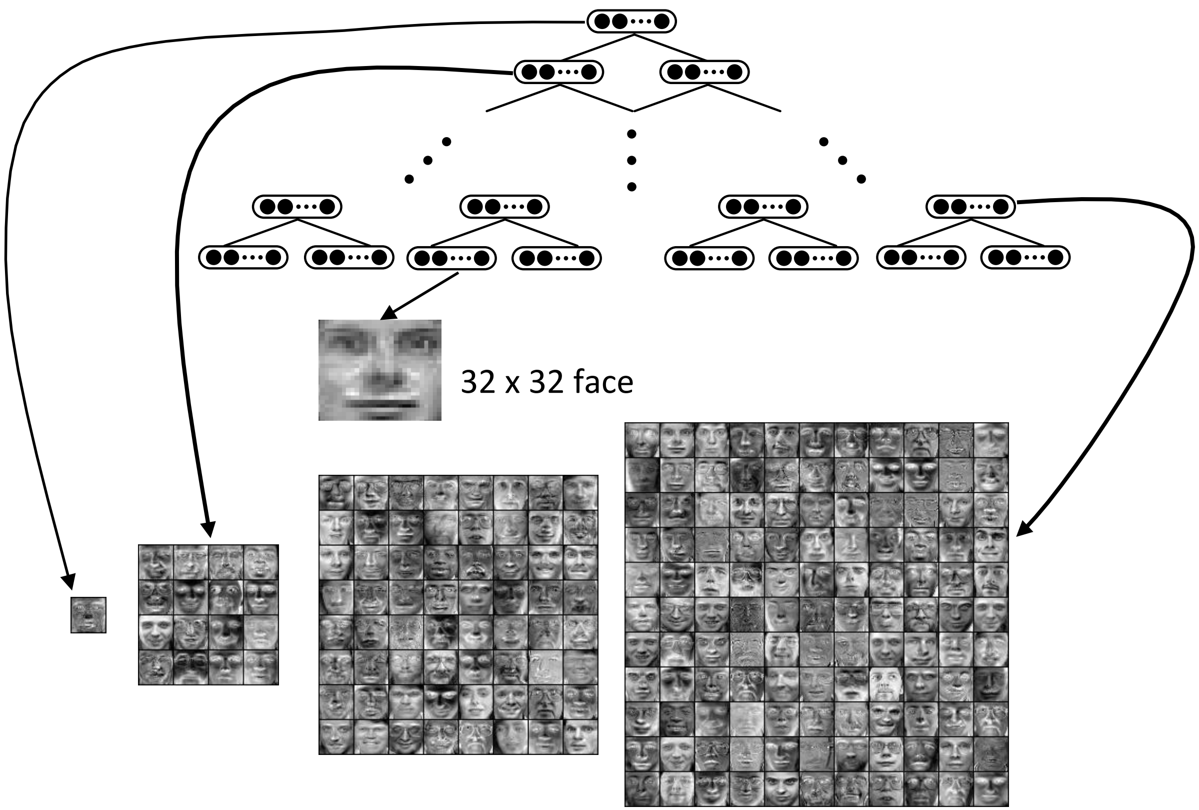
Dog	***foster, sleeping, service, cute, cat, puppy, stray, freshfields, spaniel, pug***	***Girl***	***dress, cute, hat, pink, boy, portrait, baby, shirt, birthday, blonde***
Morning* + *Drink	***drink, champagne, sky, juice, drinks, coffee, foggy, cold, soda, sun***	***Evening* + *Drink***	***evening, drink, wine, beer, bartender, alcohol, beverage, martini, liquor, bottle***

Table 1: Spearman correlation of 971 word-pair similarities computed by different methods and MEN human judgements. Our method learned from SBU is very close to Word2Vec learned from Google News.

Method	Word2Vec	S-CNN	Ours
MEN	**0.65**	0.36	0.64

Slides: http://comp.nus.edu.sg/~xiangnan/icmr18-recsys.pdf

Model Visualization: Why it works?



Slides: http://comp.nus.edu.sg/~xiangnan/icmr18-recsys.pdf

Slides: http://comp.nus.edu.sg/~xiangnan/icmr18-recsys.pdf

Slides: http://comp.nus.edu.sg/~xiangnan/icmr18-recsys.pdf

Slides: http://comp.nus.edu.sg/~xiangnan/icmr18-recsys.pdf

Summary

- Many multimedia tasks benefit from GOOD visual features, so does Recommendation!
- Image: Use fine-grained and dynamic features (RoI+Visual Attention).
- Video: Better choice is to follow image, but 3d-conv is worth trying.
- Please follow ``weakly semi-supervised learning ′′ in ML for inspiration.

Slides: http://comp.nus.edu.sg/~xiangnan/icmr18-recsys.pdf

Outline of Tutorial

- Background (Xiangnan, 10 mins)
- Basics & Advances in Recommendation (Xiangnan, 50 mins)
- Visually-aware Product Recommendation (Xiangnan, 30 mins)
- Break (15 mins)
- Visual Representation (Hanwang, 45 mins)
- Image/Video Recommendation (Hanwang, 25 mins)
- Conclusion (Hanwang, 5 mins)

Conclusion

- Recommendation becomes increasingly important in our daily life
  – Age of Information overload
- Multimedia Recommendation is rich area of research.
  ✓ Advanced recommendation technologies
  ✓ Advanced visual representation learning
  ✓ Representative visually-aware recommendation methods
- More research to be done in integrating state-of-the-art CV techniques into recommendation.

Slides: http://comp.nus.edu.sg/~xiangnan/icmr18-recsys.pdf

Challenges

- Model: data-driven + knowledge-driven
  - Most current methods are purely data-driven
  - Prior information (e.g., domain knowledge, symbolic knowledge) is helpful and should be integrated into data-driven learning in a principled way.
- Task: multiple criteria
  - Existing work have primarily focused on similarity (relevance)
  - Different scenarios may have different matching goals
  - Other criteria such as novelty, diversity, and explainability should be taken into consideration

Slides: http://comp.nus.edu.sg/~xiangnan/icmr18-recsys.pdf

Thanks!