DeepStyle: Learning User Preferences for Visual Recommendation

Qiang Liu, Shu Wu, Liang Wang. SIGIR 2017.

발표자 김형준

Overview

- **Introduction**
- **DeepStyle**
- **Experiments**
- **Conclusion**

Introduction

- **Visual information plays an important role in understanding user behaviors.**
  - “Seeing is believing” (?)
  - Especially in domains such as clothes, jewelries, house decorations, etc.

- **Models considering visual features for personalized recommendation.**
  - VBPR (5/17) : extends BPR and incorporate visual features
  - Sherlock : use categorical information.
    - Transfer visual features to categorical features.
    - one embedding for each category

Introduction

- **Fails to capture different "styles" of items**
  - One category of items are assigned to one cluster.
  - Items with **different styles** cannot be distinguished!
  - Suit pants + jeans ↔ suit pants + leather shoes

Introduction

- **Item consists of two components : style and category**

$$item = style + category \ .$$

  - Obtain the **style features** of an item by eliminating the corresponding categorical information.

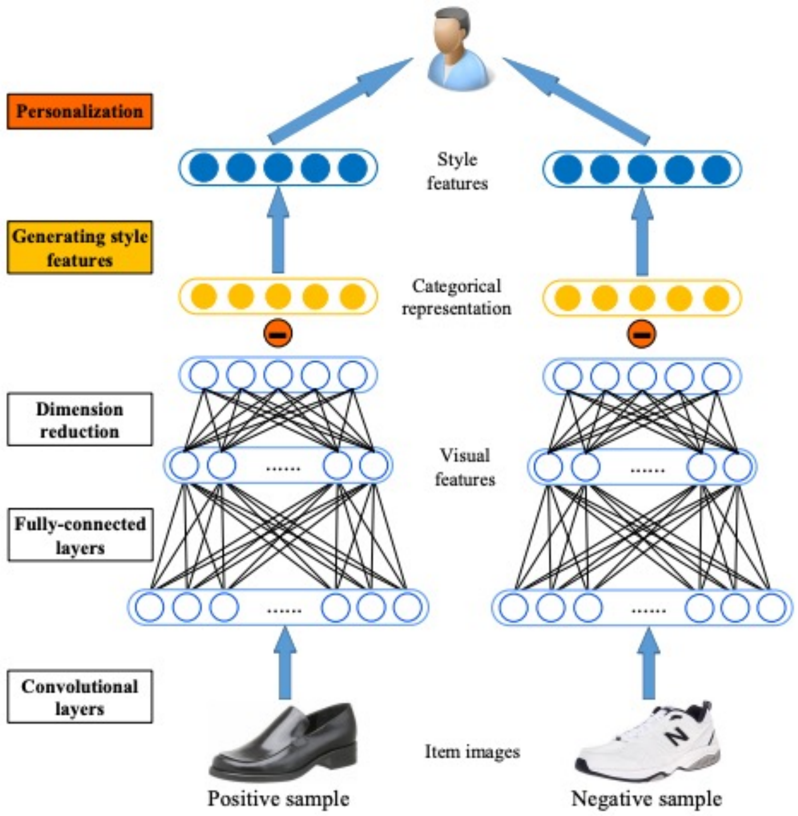
$$style = item - category$$

DeepStyle

- **Notation**
  - U : set of users
  - I : set of items
    - Item i is associated with an image.
  - I^u : set of items selected by user u
  - l_i : specific category that item i belongs.

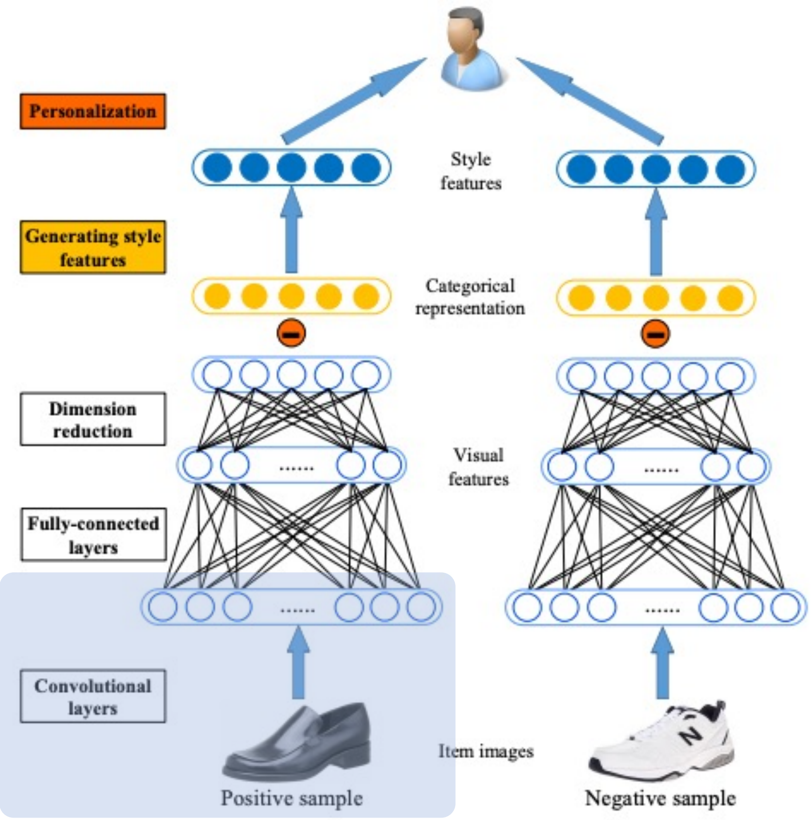
DeepStyle

- **Process**



DeepStyle

- **Process**
  - For each item i, feed the corresponding image into a deep CNN model.
    - 5 convolutional layers followed by 3 fully-connected layers.
    - Pre-trained on 1.2 million ImageNet images.

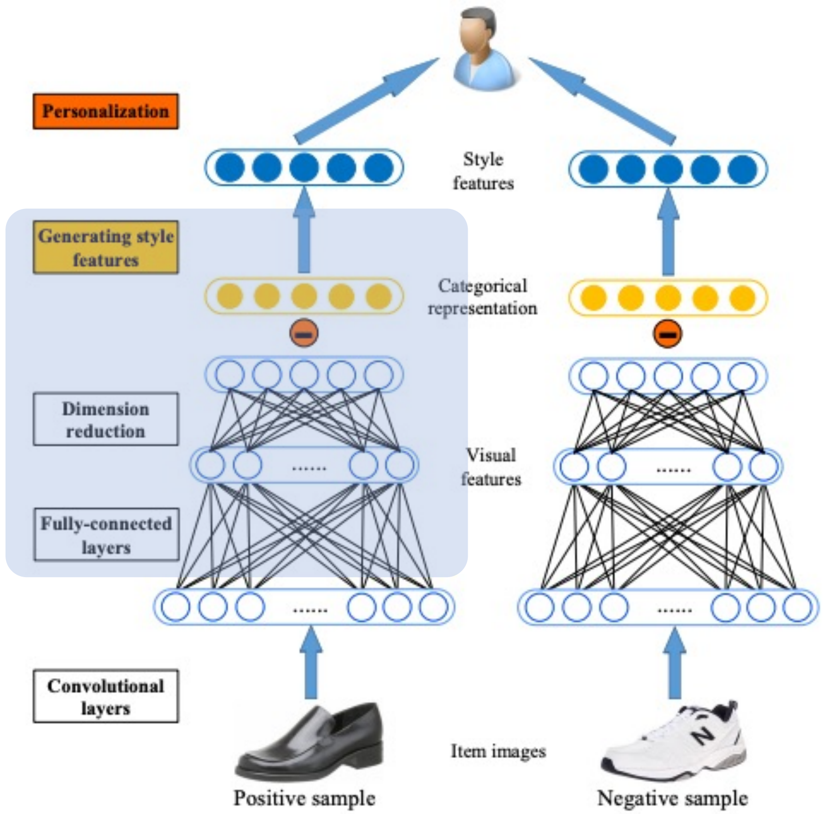


DeepStyle

- **Process**
  - For each item i, feed the corresponding image into a deep CNN model.
    - 5 convolutional layers followed by 3 fully-connected layers.
    - Pre-trained on 1.2 million ImageNet images.
  - Visual feature vector $v_i \in R^{4096}$ from the output layer of the CNN model.
  - Obtain item's **style features** by subtracting items' latent categorical representations from the visual features v_i.

$$\mathbf{s}_i = \mathbf{E}\mathbf{v}_i - \mathbf{l}_i \ ,$$

    - $\boldsymbol{s_i} \in R^d$: style feature of item i
    - $\boldsymbol{v_i} \in R^{4096}$: visual feature vector of item i
    - $\boldsymbol{E} \in R^{d \times 4096}$: matrix transferring visual features to lower dimensionality
    - $l_i \in R^d$: latent categorical representation of the corresponding category.

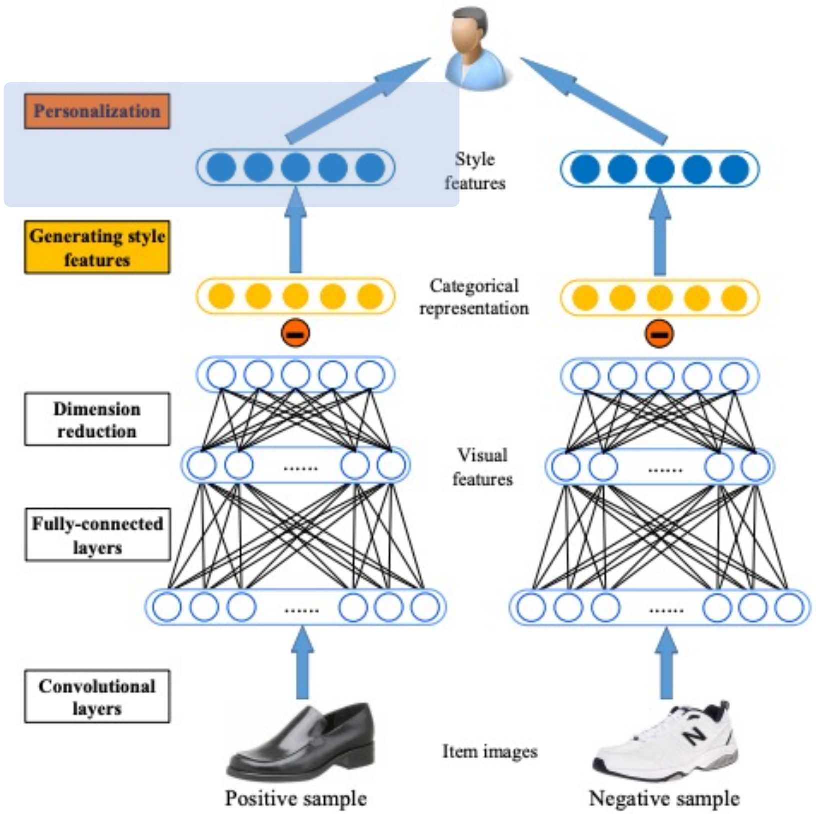


DeepStyle

- **Process**
  - Prediction of user u on item i (adapting BPR) :

$$\hat{y}_{u,i} = (\mathbf{p}_u)^T (\mathbf{s}_i + \mathbf{q}_i) \ ,$$
$$= (p_u)^T q_i + (p_u)^T s_i$$

    - $p_u \in R^d$: latent representation of user u
    - $q_i \in R^d$: latent representation of item i



INTELLIGENT DATA SYSTEMS LABORATORY

DeepStyle

- **Process (training)**
  - For user u, with an arbitrary **negative sample** i',
    - $i' \notin I^u$
  - We need to maximize $\boldsymbol{P(u, i > i') = g(\hat{y}_{u,i} - \hat{y}_{u,i'})}$ (BPR-framework)
    - where $g(x) = \frac{1}{(1+e^{-x})}$
  - Incorporating the negative log-likelihood, we minimize the objective function J :

$$J = \sum_{u,i} \ln \left(1 + e^{-\left(\hat{y}_{u,i} - \hat{y}_{u,i'}\right)}\right) + \lambda \|\theta\|^2 ,$$

DeepStyle

- **Process (training)**
  - For user u, with an arbitrary **negative sample** i',
    - $i' \notin I^u$
  - We need to maximize $\boldsymbol{P(u, i > i')} = \boldsymbol{g(\hat{y}_{u,i} - \hat{y}_{u,i'})}$ (BPR-framework)
    - where $g(x) = \frac{1}{(1+e^{-x})}$
  - Incorporating the negative log-likelihood, we minimize the objective function J :

$$J = \sum_{u,i} \ln \left(1 + e^{-\left(\hat{y}_{u,i} - \hat{y}_{u,i'}\right)}\right) + \lambda \|\theta\|^2 ,$$

Regularization term

  - Employ Stochastic Gradient Descent (SGD).

INTELLIGENT DATA SYSTEMS LABORATORY

Experiments

- **Amazon Dataset**
  - **Clothing** dataset : Clothing, Shoes, Jewelry
    - Consists of 74 categories.
  - **Home** dataset : Home, Kitchen
    - Consists of 86 categories.
  - 80% for training and 20% for testing.
  - Users with less than 5 records and more than 100 records were removed.

- **Warm-start vs Cold-start**
  - Cold-start : items with less than 5 records during training.

- **Metric : Area Under the ROC Curve (AUC)**
  - Larger the AUC value, the better the performance.

$$AUC = \frac{1}{|\mathcal{U}|} \sum_{u \in \mathcal{U}} \frac{1}{|set\left(i \in \mathcal{I}^u, i' \notin \mathcal{I}^u\right)|} \sum_{i \in \mathcal{I}^u, i' \notin \mathcal{I}^u} \delta\left(p_{u,i} > p_{u,i'}\right),$$

Experiments

- **State-of-the-art methods : BPR, VBPR, Sherlock**
  - BPR : method for modeling implicit feedbacks.
  - VBPR : incorporates **visual features** based on BPR.
  - Sherlock : extends VBPR and takes **categorical information** into consideration.

Experiments

Table 1: Performance comparison on predicting users preferences on items measured by AUC. The dimensionality is $d = 10$ on both datasets.

dataset	setting	BPR	VBPR	Sherlock	DeepStyle
Clothing	warm-start	0.6243	0.7441	0.7758	**0.7961**
	cold-start	0.5037	0.6915	0.7167	**0.7317**
Home	warm-start	0.5848	0.6845	0.7049	**0.7155**
	cold-start	0.5053	0.6140	0.6322	**0.6396**

Experiments

Table 1: Performance comparison on predicting users preferences on items measured by AUC. The dimensionality is $d = 10$ on both datasets.

dataset	setting	BPR	VBPR	Sherlock	DeepStyle
Clothing	warm-start	0.6243	0.7441	0.7758	**0.7961**
	cold-start	0.5037	0.6915	0.7167	**0.7317**
Home	warm-start	0.5848	0.6845	0.7049	**0.7155**
	cold-start	0.5053	0.6140	0.6322	**0.6396**

- **Methods incorporating visual features outperform baseline method BPR.**
  - Larger improvements in cold-start setting.
  - Visual features can model **cold-start items** even when the observations are not enough.

Experiments

Table 1: Performance comparison on predicting users preferences on items measured by AUC. The dimensionality is $d = 10$ on both datasets.

dataset	setting	BPR	VBPR	Sherlock	DeepStyle
Clothing	warm-start	0.6243	0.7441	0.7758	**0.7961**
	cold-start	0.5037	0.6915	0.7167	**0.7317**
Home	warm-start	0.5848	0.6845	0.7049	**0.7155**
	cold-start	0.5053	0.6140	0.6322	**0.6396**

- **Methods modeling categorical factors (Sherlock, DeepStyle) have better performance than VBPR.**
  - It is important to consider categorical information.

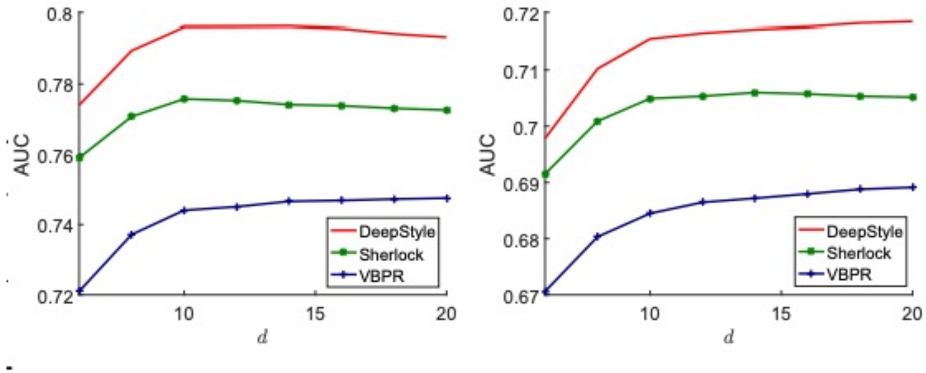
Experiments

Table 1: Performance comparison on predicting users preferences on items measured by AUC. The dimensionality is $d = 10$ on both datasets.

dataset	setting	BPR	VBPR	Sherlock	DeepStyle
Clothing	warm-start	0.6243	0.7441	0.7758	**0.7961**
	cold-start	0.5037	0.6915	0.7167	**0.7317**
Home	warm-start	0.5848	0.6845	0.7049	**0.7155**
	cold-start	0.5053	0.6140	0.6322	**0.6396**

- **DeepStyle have better performance then Sherlock.**
  - Learning style features of items is important.

Experiments



(a) Clothing. (b) Home.

- **Dimensionality sensitivity**
  - $d = [6, 8, 10, 12, 14, 16, 18, 20]$
  - After $d = 10$, DeepStyle is not very sensitive with the dimensionality.
  - Sherlock has tendency to overfit when the dimensionality is large.
    - One embedding for each category requires too many parameters.

Visualization

Visualization

Female

Male

Visualization

Formal/official Casual/school-look Old-style(?)

Visualization

- **Cluster covers a distinct style of clothing.**

Conclusion

- **DeepStyle : learning styles of items**
  - Subtracts categorical information from visual features of items generated by CNN.
  - With the learned style features combined with BPR framework, personalized recommendation can be performed.

Thank you 🤗