

AutoRec:Autoencoders Meets Collaborative Filtering

Recommender Systems
Collaborative Filtering
Autoencoders

추천 시스템 Boot Camp 4기
강석우

Abstract

Collaborative Filtering을 위한 autoencoder 프레임워크인 AutoRec 모델
compact(소형)고 효율적인 학습이 가능한 모델이다.

MovieLens와 Netflix 데이터셋 에서 실험을 했다. (biased MF, RBM-CF, LLORMA)
SOTA를 이뤘다.

Introduction

CF 모델은 개인화 추천 제공을 위해 아이템에 대한 유저의 선호에 대한 데이터를 활용한다.

Netflix 챌린지로 MF, Neighborhood 방법 등과 함께 다양한 CF 기법이 제안 되었다.

본 논문에서 제안하는 **AutoRec**은 비전과 스피치 분야에서 딥러닝으로 어느 정도 성공한 **Autoencoder** 구조를 CF에 대입했다.

AutoRec이 다른 CF에 비해 표현력이 있고 계산적으로 우월한 면이 있다고 주장한다.

이러한 점을 실험을 통해서 현재 **SOTA** 보다 좋은 성능을 보이는 점을 증명한다.

The AutoRec Model

Rating 을 기반으로 하는 CF에서는 m 으로 해당하는 유저 수, n 에 해당하는 아이템 수, 그리고 유저와 아이템으로 부분적으로 채워져 있는 Rating 매트릭스 R 이 있다.

$$R \in \mathbb{R}^{m \times n}$$

각 유저 u 는 부분적으로 관측된 벡터 $\mathbf{r}^{(u)} = (R_{u1}, \dots, R_{un}) \in \mathbb{R}^n$

각 아이템 i 는 부분적으로 관측된 벡터 $\mathbf{r}^{(i)} = (R_{1i}, \dots, R_{mi}) \in \mathbb{R}^m$

목표는 아이템-기반(유저-기반)의 오토인코더를 사용해서 부분적으로 관측된 \mathbf{r} 들을 입력으로 낮은 차원의 **latent** 공간으로 투영하고 다시 \mathbf{r} 을 재현해서 추천에 사용하는 것이다.

The AutoRec Model

주어진 rating 들의 집합 S 에 벡터들 \mathbf{r} 에 대해 $\min_{\theta} \sum_{\mathbf{r} \in S} \|\mathbf{r} - h(\mathbf{r}; \theta)\|_2^2$ 를 최소화

h 는 입력 \mathbf{r} 을 재현하고 $h(\mathbf{r}; \theta) = f(\mathbf{W} \cdot g(\mathbf{V}\mathbf{r} + \boldsymbol{\mu}) + \mathbf{b})$ k 차원의 히든 계층을 가진 auto-associative neural network (데이터 패턴을 기억하는 모델)다.

item-based AutoRec 모델이 아이템 수 만큼 만들
져서 제공된다. 부분적으로 관측된 데이터 \mathbf{r} 만 사
므로 정규화로 오버피팅을 해결하려고 했다.

item-based AutoRec의 목적 함수는 아래와 같다

$$\min_{\theta} \sum_{i=1}^n \|\mathbf{r}^{(i)} - h(\mathbf{r}^{(i)}; \theta)\|_2^2 + \frac{\lambda}{2} \cdot (\|\mathbf{W}\|_F^2 + \|\mathbf{V}\|_F^2)$$

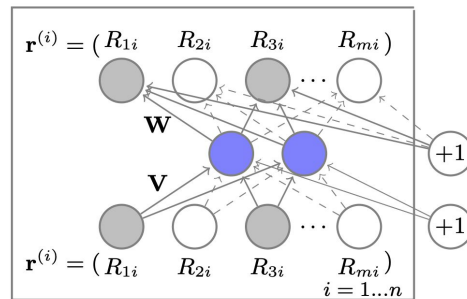
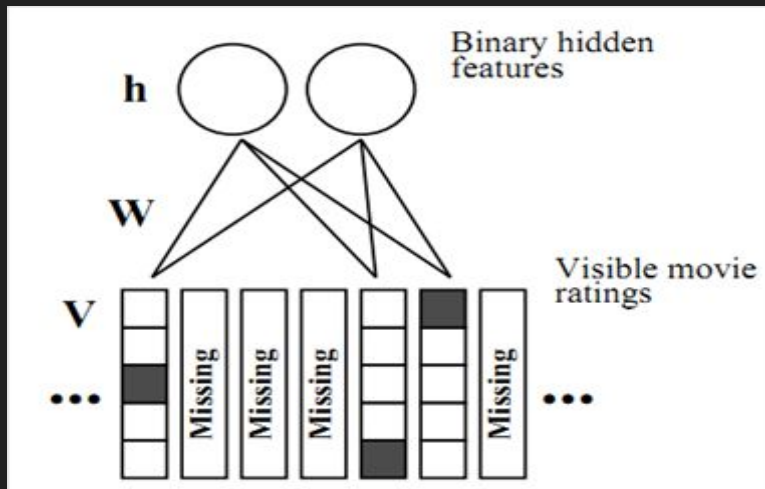


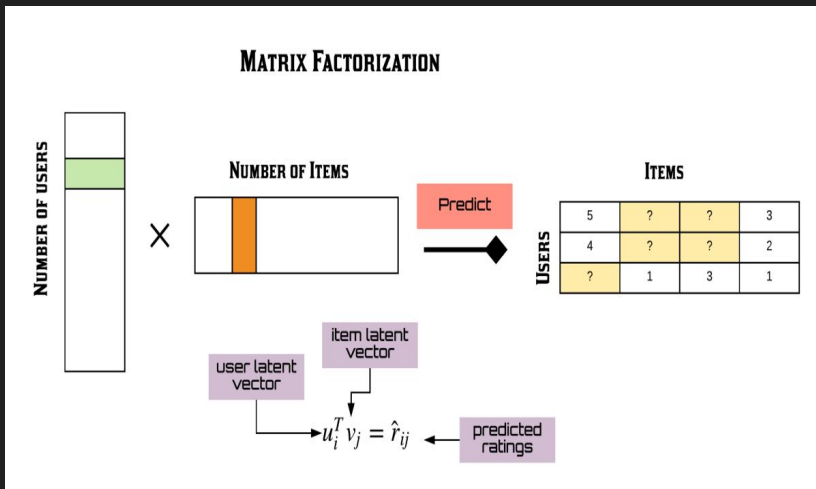
Figure 1: Item-based AutoRec model. We use plate notation to indicate that there are n copies of the neural network (one for each item), where \mathbf{W} and \mathbf{V} are tied across all copies.

Relative Works

AutoRec은 RBM-Based CF, Matrix Factorization 모델과 다르다.

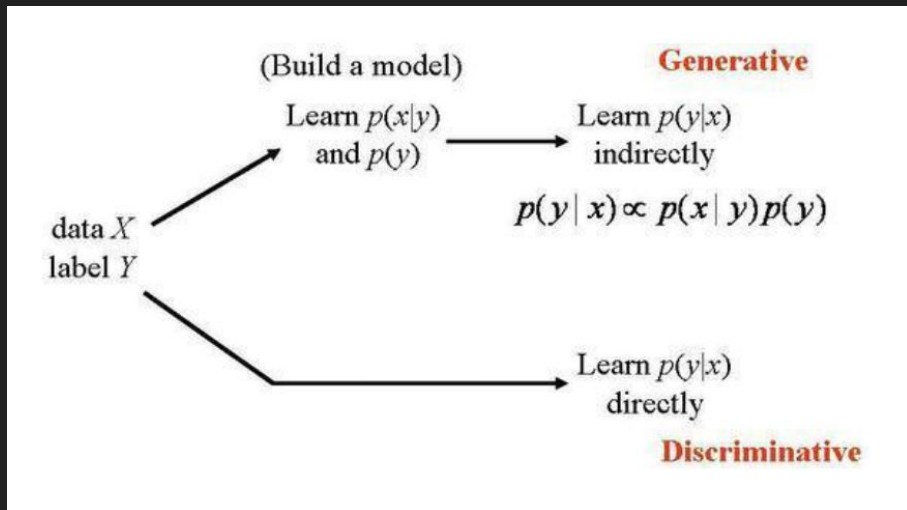


RBM은 generative, probabilistic 모델이다



MF 는 유저, 아이템 임베딩을 같이 임베딩 한다.

AutoEncoder(Discriminative) VS RBM(Generative)



Generative Model : $p(h = 0, 1) = s(Wx + b)$

Discriminative Model : $h = s(Wx + b)$

Experimental Evaluation

	ML-1M	ML-10M
U-RBM	0.881	0.823
I-RBM	0.854	0.825
U-AutoRec	0.874	0.867
I-AutoRec	0.831	0.782

(a)

$f(\cdot)$	$g(\cdot)$	RMSE
Identity	Identity	0.872
Sigmoid	Identity	0.852
Identity	Sigmoid	0.831
Sigmoid	Sigmoid	0.836

(b)

	ML-1M	ML-10M	Netflix
BiasedMF	0.845	0.803	0.844
I-RBM	0.854	0.825	-
U-RBM	0.881	0.823	0.845
LLORMA	0.833	0.782	0.834
I-AutoRec	0.831	0.782	0.823

(c)

테스트 케이스(전체 10%)의 Rating을 3을 기본으로 주어서 측정 했다.

- (a) I/U-AutoRec과 RBM 모델의 RMSE 성능 비교
- (b) Activation 함수의 linear와 non-linear 선택 에 따른 RMSE 성능 비교
- (c) I-AutoRec 과 Baseline 모델의 RMSE 성능 비교

I-RBM은 학습 후 일주일이 지나면 수렴하지 못한다고 한다.