

# AutoRec : AutoEncoders Meet Collaborative Filtering

## ABSTRACT

오토인코더 프레임워크를 CF에 적용

## INTRODUCTION

CF는 아이템에 대한 유저의 선호 정보를 찾는데 초점을 맞춘다.

이 논문은 오토인코더 패러다임에 기반을 둔 CF 모델 AutoRec을 제안

AutoRec은 표현력과 계산에서 기존의 CF모델 보다 이점을 가짐.

## THE AUTOREC MODEL

점수 기반 CF에서 유저(m), 아이템(n)과 부분적으로 레이팅이 관측된 매트릭스를 가질 수 있다.

	u1	u2	u3	...	u9	u10
i1	1		1	...	1	
i2		0		...		
i3	2			...	3	
...	...	...	...	...	...	...
i11				...		1
i12		2	3	...		

각 유저(아이템)의 표현은 벡터 해당 행,열의 벡터로 표현 가능 (이후로는 유저 관점만 언급, 아이템도 동일)

## 오토인코더

이 작업에서는 추천의 목적으로 오토인코더가 입력을 위 행렬의 일부분(유저 벡터)을 받아, 낮은 차원의 잠재 공간으로 투영시키고, 그 후 없었던 레이팅을 채워서 재구성하는 것에 초점을 맞춘다.

$$\min_{\theta} \sum_{\mathbf{r} \in \mathbf{S}} \|\mathbf{r} - h(\mathbf{r}; \theta)\|_2^2, \quad (1)$$

오토 인코더 목적 함수

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

오토인코더 연산 과정

활성화 함수(f,g)와 파라미터(W,V,m,b)를 가짐.

## 차이점

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

관측된 값에 대해서만 계산, 정규화 텀 추가된 목적함수

관측된 값에 대해서만 역전파 - 보통의 MF 방법론에 적용되어 있음

오버피팅을 막기 위해 정규화 텀 사용

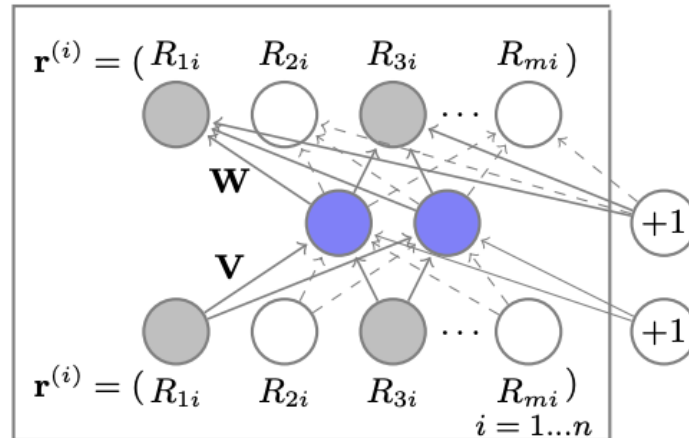


Figure 1: Item-based AutoRec model. We use plate notation to indicate that there are  $n$  copies of the neural network (one for each item), where  $\mathbf{W}$  and  $\mathbf{V}$  are tied across all copies.

음영있는 부분이 관측된 값, 실선 부분이 업데이트에 관여

## 비교

	AutoRec	RBM-CF
기반 모델	autoencoder	RBM(restricted Boltzmann machine)
학습 방법	RMSE 최소화 gradient 역전파 - 빠름	log likelihood 최대화 contrastive divergence 사용 - 느림
사용 평점	평점 종류에 관계 없음	이진 평점 (좋아요/싫어요)
파라미터 수	상대적으로 적음	상대적으로 많음
표현 학습	아이템, 유저의 별도 잠재 공간 사용 비선형성 포함 가능	아이템 유저 잠재 공간 공유 비선형성 포함 불가

## EXPERIMENTAL EVALUATION

아이템 / 유저 베이스 중에 뭐가 나온가?

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

(a)

(a) Comparison of the RMSE of I/U-AutoRec and RBM models.

아이템 베이스가 좀 더 나은 것을 볼 수 있음

평균 레이팅 수가 유저 기준보다는 아이템 기준이 많기 때문일 수 있다.

**AutoRec은 선형/비선형 활성화 함수에 따라 어떤 성능을 보였는가(f,g)?**

$f(\cdot)$	$g(\cdot)$	RMSE
Identity	Identity	0.872
Sigmoid	Identity	0.852
Identity	Sigmoid	<b>0.831</b>
Sigmoid	Sigmoid	0.836

(b)

(b) RMSE for I-AutoRec with choices of linear and nonlinear activation functions, Movielens 1M dataset.

복원 과정(g)에서 비선형 함수를 사용하는 것이 좋은 성능을 보였다.

시그모이드를 ReLU로 바꿔봤지만 성능이 더 안좋아졌다.

f : identity / g : sigmoid로 설정

**AutoRec에서 hidden unit의 수에 따라 어떤 성능을 보였는가?**

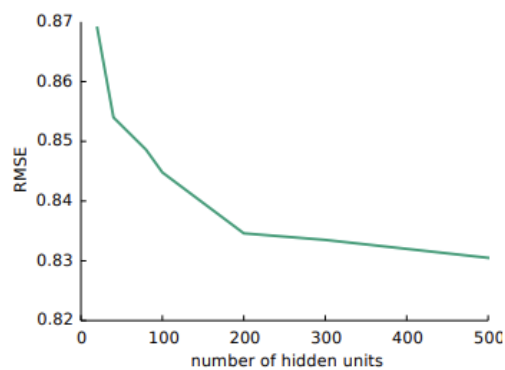


Figure 2: RMSE of I-AutoRec on Movielens 1M as the number of hidden units  $k$  varies.

유닛 수가 늘어날수록 성능 향상이 되긴 하지만, 그 폭이 줄어드는 것을 볼 수 있다.

k=500으로 설정

**AutoRec은 베이스라인에 비해 어떤 성능을 보이는가?**

	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	<b>0.782</b>	0.834
I-AutoRec	<b>0.831</b>	<b>0.782</b>	<b>0.823</b>

(c)

Comparison of I-AutoRec with baselines on MovieLens and Netflix datasets. We remark that I-RBM did not converge after one week of training

베이스 라인보다 전반적으로 높은 성능을 보임.

LLORMA는 복잡한 구조를 갖고 있으나, 단일 모델 아키텍처로 비슷하거나 능가한 점이 흥미로운 부분

### AutoRec의 층을 깊이 쌓는 것이 도움이 될까?

3층 (500, 250, 500) 로 설정하여 실험해보았을 때, RMSE가 0.831 → 0.827로 줄었다.