Park Kibum (Polar)

VASP (2021)

Deep Variational Autoencoder with Shallow Parallel Path for Top-N Recommendation (VASP)

CONTENTS

Background

Introduction

Solutions

VASP 구조

Reference

VASP

Background

논문 선정 배경 및 간단한 모델 소개

01 Background

MovieLens

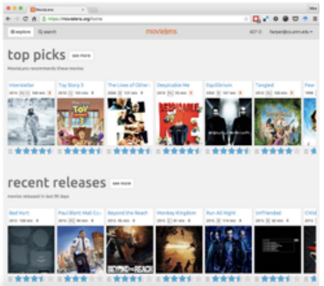
Edit

Introduced by F. Maxwell Harper et al. in The MovieLens Datasets: History and Context

The **MovieLens** datasets, first released in 1998, describe people's expressed preferences for movies. These preferences take the form of tuples, each the result of a person expressing a preference (a 0-5 star rating) for a movie at a particular time. These preferences were entered by way of the MovieLens web site1 — a recommender system that asks its users to give movie ratings in order to receive personalized movie recommendations.

Source: The MovieLens Datasets: History and Context

Homepage



Source: http://files.grouplens.org/papers/...

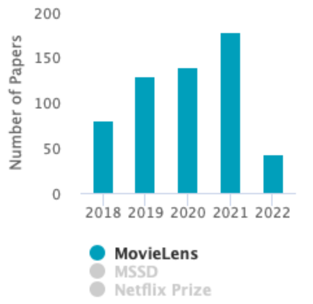
Benchmarks

Edit

Trend	Task	Dataset Variant	Best Model	Paper	Code
	Recommendation Systems	MovieLens 1M	GLocal-K		
	Recommendation Systems	MovieLens 20M	VASP		
	Recommendation Systems	MovieLens 10M	Bayesian timeSVD++ flipped		
	Recommendation Systems	MovieLens 100K	GHRS		
	Link Prediction	MovieLens 25M	PEAGAT		

Show all 10 benchmarks

Usage



License

Edit

Custom

출처 : https://paperswithcode.com/dataset/movielens

01 Background

Rank	Model	nDCG@100↑	nDCG@10	Recall@20	Recall@50	Recall@100	HR@10	Recall@10	Recall@2	RMSE	Extra Training Data	Paper	Code	Result
1	**VASP**	0.448		0.414	0.552						×	Deep Variational Autoencoder with Shallow Parallel Path for Top-N Recommendation (VASP)		
2	**H+Vamp Gated**	0.44522		0.41308	0.55109						×	Enhancing VAEs for Collaborative Filtering: Flexible Priors & Gating Mechanisms		
3	**RecVAE**	0.442		0.414	0.553						×	RecVAE: a New Variational Autoencoder for Top-N Recommendations with Implicit Feedback		
4	**RaCT**	0.434		0.403	0.543						×	Towards Amortized Ranking-Critical Training for Collaborative Filtering		
5	**Mult-VAE PR**	0.426		0.395	0.537						×	Variational Autoencoders for Collaborative Filtering		
6	**EASE**	0.420		0.391	0.521						×	Embarrassingly Shallow Autoencoders for Sparse Data		

Rank	Model	nDCG@100	nDCG@10	Recall@20↑	Recall@50	Recall@100	HR@10	Recall@10	Recall@2	RMSE	Extra Training Data	Paper	Code	Result	Y
1	**Multi-Gradient Descent**			0.418							×	Multi-Gradient Descent for Multi-Objective Recommender Systems			2
2	**VASP**	0.448		0.414	0.552						×	Deep Variational Autoencoder with Shallow Parallel Path for Top-N Recommendation (VASP)			2
3	**RecVAE**	0.442		0.414	0.553						×	RecVAE: a New Variational Autoencoder for Top-N Recommendations with Implicit Feedback			2
4	**H+Vamp Gated**	0.44522		0.41308	0.55109						×	Enhancing VAEs for Collaborative Filtering: Flexible Priors & Gating Mechanisms			2
5	**RaCT**	0.434		0.403	0.543						×	Towards Amortized Ranking-Critical Training for Collaborative Filtering			2

출처 : https://paperswithcode.com/sota/collaborative-filtering-on-movielens-20m

VASP

Introduction

논문의 문제제기와 핵심 포인트

02 Introduction

최근 동향

많은 아이템에서 **빠른** 시간 학습하여 **실시간으로** 추천

02 Introduction

최근 동향

Embarrassingly Shallow Autoencoders for Sparse Data*

Harald Steck
Netflix
Los Gatos, California
hsteck@netflix.com

ABSTRACT

Combining simple elements from the literature, we define a linear model that is geared toward sparse data, in particular implicit feedback data for recommender systems. We show that its training objective has a closed-form solution, and discuss the resulting conceptual insights. Surprisingly, this simple model achieves better ranking accuracy than various state-of-the-art collaborative-filtering approaches, including deep non-linear models, on most of the publicly available data-sets used in our experiments.

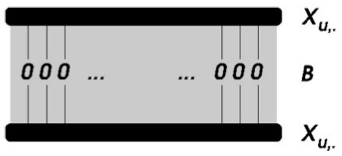


Figure 1: The self-similarity of each item is constrained to zero between the input and output layers.

Top-N 추천에 강한 EASE 모델의 등장

강하고 explainable하고
identity에 ovefitting

하지만 linear model이라는 한계 존재

02 Introduction

최근 동향

1	VASP	0.448		0.414	0.552			×	Deep Variational Autoencoder with Shallow Parallel Path for Top-N Recommendation (VASP)
2	H+Vamp Gated	0.44522		0.41308	0.55109			×	Enhancing VAEs for Collaborative Filtering: Flexible Priors & Gating Mechanisms
3	RecVAE	0.442		0.414	0.553			×	RecVAE: a New Variational Autoencoder for Top-N Recommendations with Implicit Feedback
4	RaCT	0.434		0.403	0.543			×	Towards Amortized Ranking-Critical Training for Collaborative Filtering
5	Mult-VAE PR	0.426		0.395	0.537			×	Variational Autoencoders for Collaborative Filtering
6	EASE	0.420		0.391	0.521			×	Embarrassingly Shallow Autoencoders for Sparse Data
7	Mult-DAE	0.419		0.387	0.524			×	Variational Autoencoders for Collaborative Filtering
8	HyperML		0.6404				0.8736	×	HyperML: A Boosting Metric Learning Approach in Hyperbolic Space for Recommender Systems
9	LRML		0.6152				0.8447	×	Latent Relational Metric Learning via Memory-based Attention for Collaborative Ranking
10	CML		0.5301		0.4665	0.6022	0.7764	×	Collaborative Metric Learning

AutoEncoder / VAE 연구 증가

02 Introduction

Problems

- RS have to scale to millions of active users and millions of items. Speed of training and recall are also increasingly important as available content often change dynamically and RS need to be able to react in real- time.

- We had to overcome several issues, most importantly, the overfitting towards identity.

- Traditionally, overfitting towards identity is addressed by using dropout in the input layer. However, this approach is not effective enough and is not enabling the usage of really deep architectures.

- It is more difficult to recommend niche items as they do not have many interactions. Higher loss for these items push recommender system to focus more on cold start and niche items.

02 Introduction

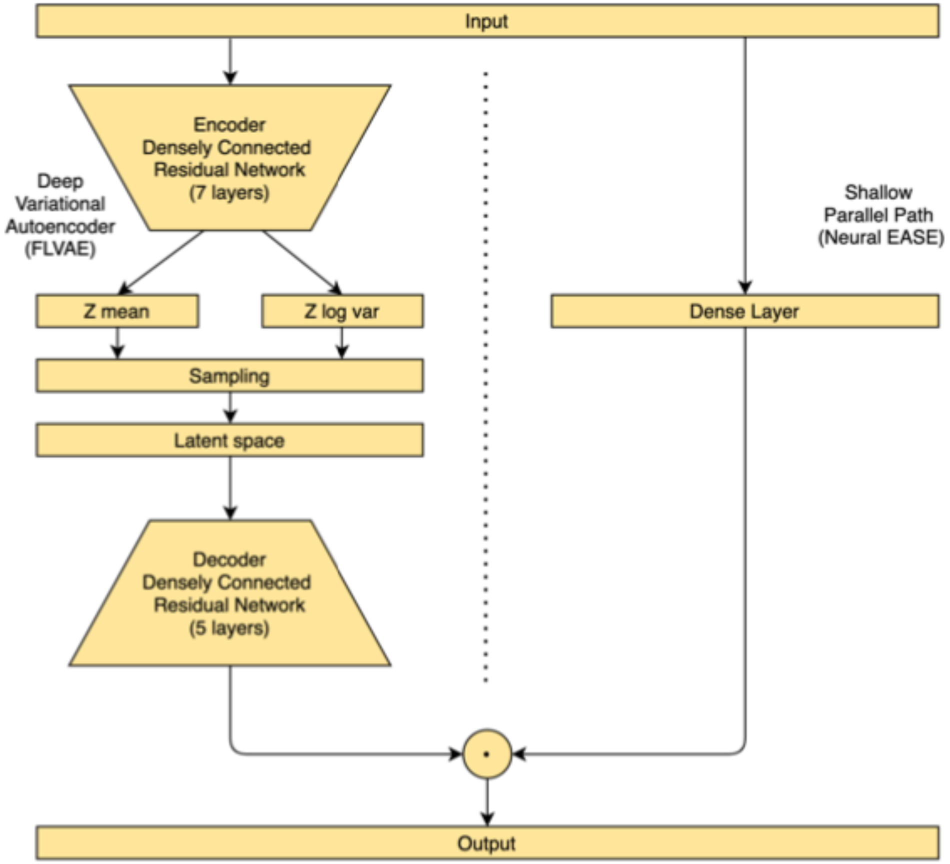


Fig. 1. VASP Architecture

Points

1. EASE + VAE
2. Focal Loss
3. Data Augmentation
4. Joint Learning

VASP

Solutions

Long-tail / Identity Overfitting / Ensemble

03 Solutions

Problem1 : Niche Item

$$FL(p_t) = -\alpha_t(1 - p_t)^{\gamma} log(p_t),$$

$$p_t = \begin{cases} \hat{x}_{ui} & if\ x_{ui} = 1 \\ 1 - \hat{x}_{ui} & otherwise \end{cases}$$

Long-tail Problem = Class imblance

CV / NLP에서 이미 많이 사용된 사례를 차용

1. Residual Networks
2. Focal Loss (FL)

Item attribute 대신 interaction을 활용

03 Solutions

Problem2 : Identity Overfitting

$$\mathbf{x}_u = [x_{u1}, x_{u2}, ..., x_{uI}]^T \in \mathbb{N}^I$$
$$\hat{\mathbf{x}}_u = [\hat{x}_{u1}, \hat{x}_{u2}, ..., \hat{x}_{uI}]^T \in \mathbb{N}^I$$

$$\hat{\mathbf{x}}_u = W \cdot \mathbf{x}_u,$$

W : I X I weight matrix

Diagonal of W is constrained to zero to prevent learning identity function between the input and the output.

03 Solutions

Problem3 : New Data Augmentation

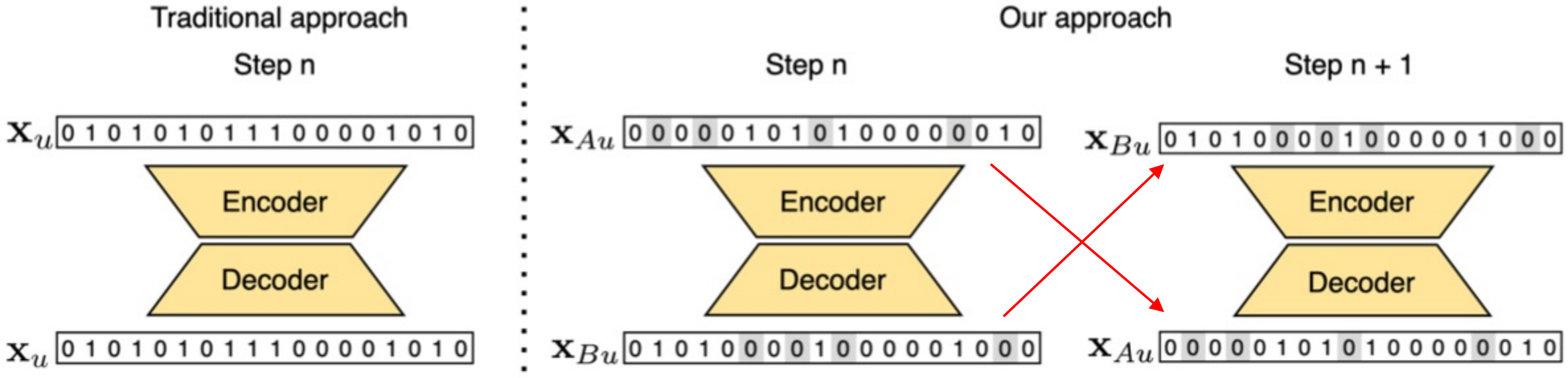


Fig. 2. Data augmentation to prevent overfitting towards identity

03 Solutions

Problem3 : New Data Augmentation

$$x_{Aui} = \begin{cases} 0 & if\ x_{ui} = 0 \\ 1 - x_{Bui} & otherwise \end{cases}$$

$$x_{Bui} = \begin{cases} 0 & if\ x_{ui} = 0 \\ 1 - x_{Aui} & otherwise \end{cases}$$

$$\sum_{i=1}^{I} x_{Aui} \approx \sum_{i=1}^{I} x_{Bui}$$

사용자에 대해 A와 B로 나눠서
$x_{B_{ui}}$를 학습할 때는 $x_{A_{ui}}$를 보여주고
$x_{A_{ui}}$를 학습할 때는 $x_{B_{ui}}$를 활용하여 학습

사용자를 split할 때 반드시 반반이 되게 나눠줘야 함

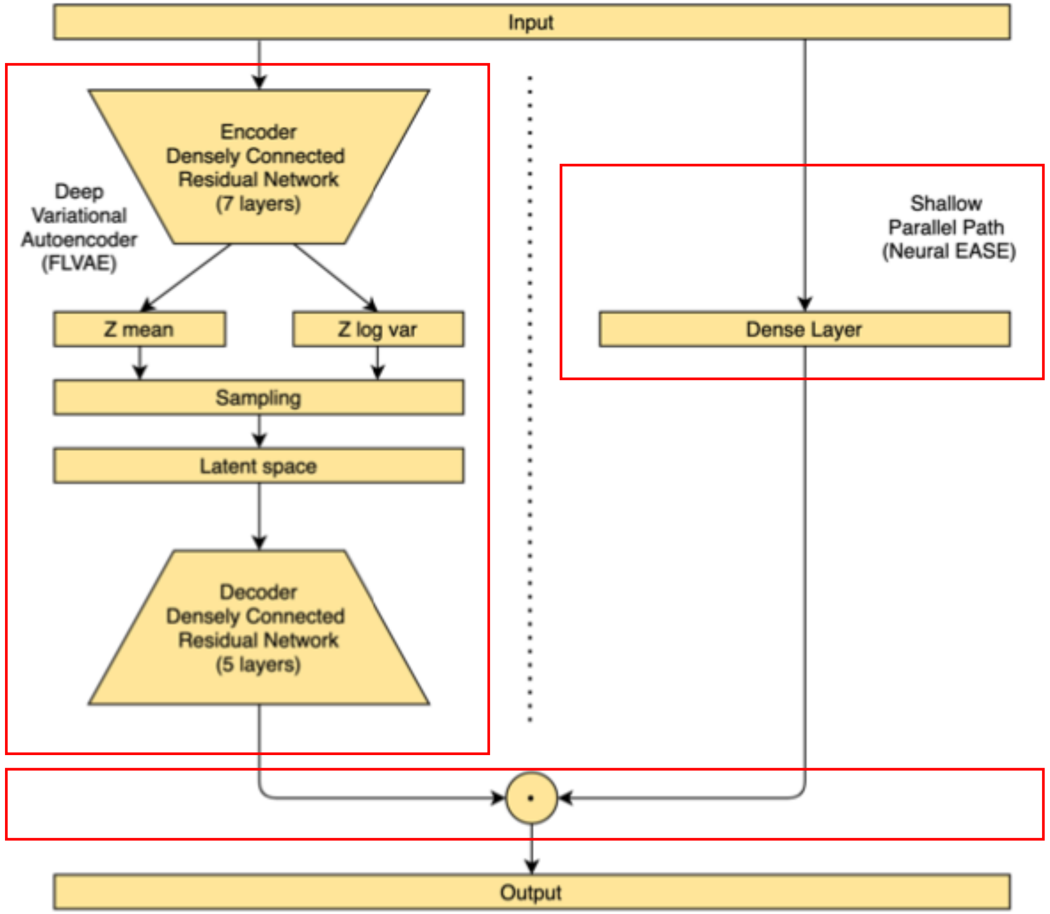
VASP

VASP 구조

VASP 모델 구조와 수식

04 VASP 구조

Focal Loss with VAE (FLVAE)



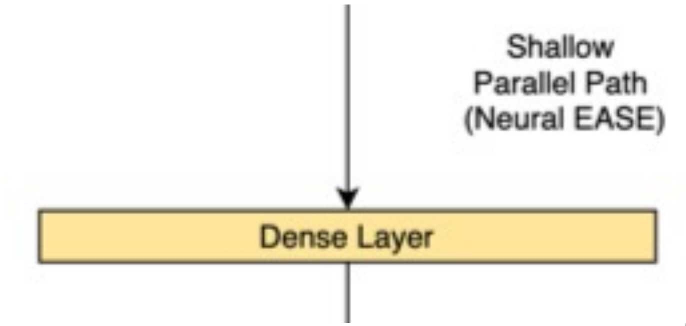
Neural EASE

VASP (Joint-learning by Hadamard product)

Fig. 1. VASP Architecture

04 VASP 구조

Neural EASE



$$\mathbf{x}_u = [x_{u1}, x_{u2}, ..., x_{uI}]^T \in \mathbb{N}^I$$
$$\hat{\mathbf{x}}_u = [\hat{x}_{u1}, \hat{x}_{u2}, ..., \hat{x}_{uI}]^T \in \mathbb{N}^I$$

$$\hat{\mathbf{x}}_u = W \cdot \mathbf{x}_u,$$

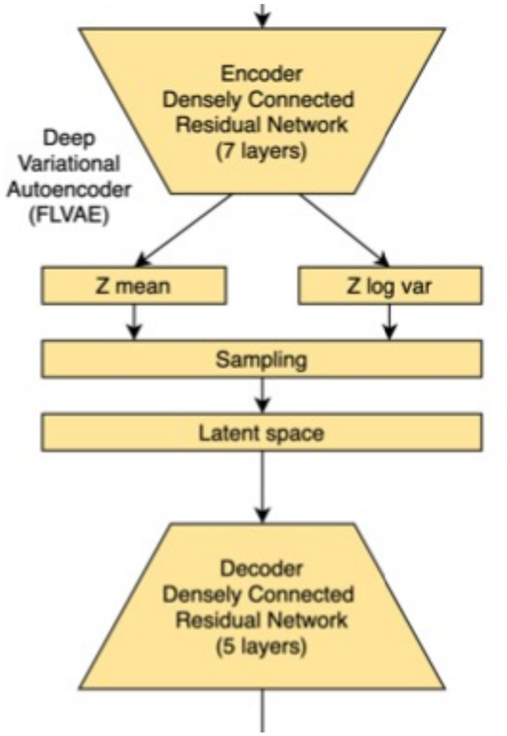
기존의 EASE를 single layer model로 구현

Embarrassingly shallow autoencoders for sparse data. Harald Steck 에 따르면 data와 prediction의 square loss는 closed-form이라 training objective

Diagonal of W is constrained to zero to prevent learning identity function between the input and the output.
→ a single-layer perceptron without bias nodes and with forced zeros on the diagonal

04 VASP 구조

Deep VAE (FLVAE)



$$\mathbf{z}_u \sim N(0, \mathbf{I}_k),$$
$$\pi(\mathbf{z}_u) \propto exp\{f_\theta(\mathbf{z}_u)\},$$
$$\mathbf{x}_u \sim Mult(N_u, \pi(\mathbf{z}_u))$$

$$FL(p_t) = -\alpha_t(1 - p_t)^\gamma log(p_t),$$

$$p_t = \begin{cases} \hat{x}_{ui} & if\ x_{ui} = 1 \\ 1 - \hat{x}_{ui} & otherwise \end{cases}$$

VAE는 $p(z_u|x_u) = \int p(x_u|z_u)p(z_u)dz$
를 최대화하는 것이 목적

04 VASP 구조

VASP : Joint-learning

$$m_n(\mathbf{x}_u) = \bigodot_{j=1}^{n} m_j = m_1(\mathbf{x}_u) \odot m_2(\mathbf{x}_u) \odot \ldots \odot m_n(\mathbf{x}_u)$$

$$m_n(\mathbf{x}_u) = \hat{x}_{nu} = \bigodot_{j=1}^{n} \hat{x}_{ju}$$

$$\hat{x}_{nu} \in \langle 0, 1 \rangle.$$

$$\begin{aligned} m_{VASP}(\mathbf{x}_u) &= m_{FLVAE}(\mathbf{x}_u) \odot m_{EASE}(\mathbf{x}_u) \\ &= \hat{\mathbf{x}}_{FLVAEu} \odot \hat{\mathbf{x}}_{NEASEu} \end{aligned}$$

Wide & Deep에서 사용된 아이디어를 차용

최종 output에는 sigmoid function 적용하여
0~1로 결과값이 나옴

Wide & Deep에서는 summation의 형태 (OR) 연산
VASP에서는 Hadamard product (AND) 연산

VASP

Reference

05 Reference

1. **Deep Variational Autoencoder with Shallow Parallel Path for Top-N Recommendation (VASP),** https://velog.io/@chwchong/Deep-Variational-Autoencoder-with-Shallow-Parallel-Path-for-Top-N-Recommendation-VASP

VASP

THANK YOU