비즈니스 Case Study를 통한

추천 시스템 구현

5강

컨텍스트를 활용한 추천 모델

Factorization Machine

5강

컨텍스트를 활용한 추천 모델

Factorization Machine

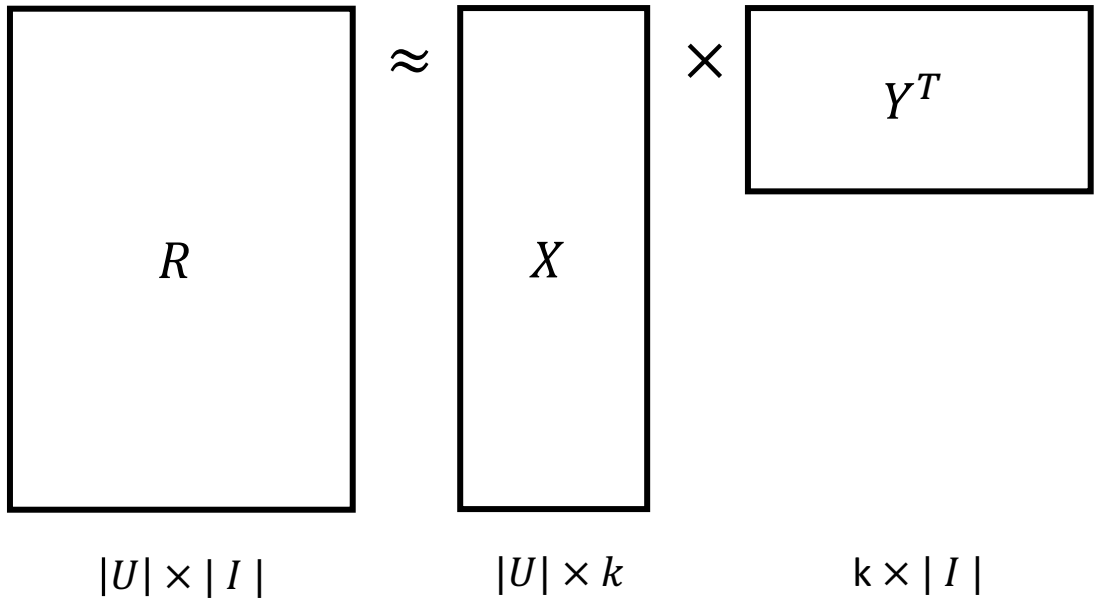
Contents

1. Matrix Factorization의 한계

Matrix Factorization 복습

$$R \approx X \times Y^T = \hat{R}$$

R 과 최대한 유사하게 $\hat{R}$ 을 추론하는 것이 목표



Objective Function

$$\min_{X,Y} \sum_{observed\ r_{u,i}} (r_{u,i} - x_u^T y_i)^2 + \lambda(||x_u||_2^2 + ||y_i||_2^2)$$

평점 예측

$$\widehat{r_{u,i}} = x_u^T y_i$$

$$\widehat{r_{u,i}} = \mu + b_u + b_i + x_u^T y_i$$

MF with Implicit Feedback

Real-world에서는 평점과 같은 Explicit Feedback보다 Implicit Feedback이 훨씬 많음

사용자의 클릭, 구매, 동영상 시청 → 추천 시스템의 평점으로 사용하기엔 부적합

MF Approach를 적용하기 위해 explicit한 평점 대신 confidence와 preference 개념을 사용함

Explicit

$$\min_{X,Y} \sum_{observed\, r_{u,i}} (r_{u,i} - x_u^T y_i)^2 + \lambda(||x_u||_2^2 + ||y_i||_2^2)$$

Implicit

$$\min_{X,Y} \sum_{observed\, r_{u,i}} c_{u,i}(p_{u,i} - x_u^T y_i)^2 + \lambda(||x_u||_2^2 + ||y_i||_2^2)$$

Matrix Factorization의 한계

추천 시스템의 데이터는 주로 유저 - 아이템의 상호작용 데이터로 이루어져 있음

개별 유저와 아이템을 피쳐로 사용하여 모델을 학습하고 추천을 수행

문제는 유저, 아이템 ID 외에 다른 부가적인 정보를 사용할 수 없다는 것

2차원 매트릭스라는 구조적인 한계로 인해 유저나 아이템의 메타데이터를 사용할 수 없음

유저의 데모그래픽, 아이템의 카테고리/태그 등을 피쳐로 활용할 수 있다면?

새로운 유저, 아이템이 등장할 때 해당 피쳐를 활용해 Cold Start에 더 잘 대처할 수 있음

General Model Framework

Matrix Factorization 모델은 추천 문제된부터 파생된 모델

주어진 유저 - 아이템 상호작용 데이터에 가장 적합한 모델

주어진 피쳐의 Latent Factor를 추출하여 Factorization하는 원리를 차용하되, 좀 더 일반적인 문제를 풀 수 있도록 모델을 구성해보자

주어진 피쳐 X에 대해서 Y를 예측하는 prediction / classification 모델을 사용해보자

Linear / Logistic Regression

Support Vector Machine

2. Factorization Machine

Factorization Machine의 등장 배경

딥러닝이 등장하기 이전에 SVM은 머신러닝에서 가장 많이 사용되는 모델이었음

커널 공간을 활용하여 비선형 데이터셋에 대해서 높은 성능을 보임

그럼에도 CF 환경에서는 SVM보다 Matrix Factorization 계열의 모델이 더 높은 성능을 내었음

매우 sparse한 데이터에 대해서는 SVM은 좋은 성능을 내지 못함

하지만 MF 모델은 특별한 환경 혹은 데이터에만 적용할 수 있음

X: (유저, 아이템) → Y: (rating)으로 이루어진 데이터에 대해서만 적용함

➔ 이 둘의 장점을 결합할 수 없을까?

Factorization Machines

SVM과 Factorization Model의 장점을 결합한 FM을 소개한 논문

Factorization Machines

Steffen Rendle
Department of Reasoning for Intelligence
The Institute of Scientific and Industrial Research
Osaka University, Japan
rendle@ar.sanken.osaka-u.ac.jp

***Abstract*—In this paper, we introduce Factorization Machines (FM) which are a new model class that combines the advantages of Support Vector Machines (SVM) with factorization models. Like SVMs, FMs are a general predictor working with any real valued feature vector. In contrast to SVMs, FMs model all interactions between variables using factorized parameters. Thus they are able to estimate interactions even in problems with huge sparsity (like recommender systems) where SVMs fail. We show that the model equation of FMs can be calculated in linear time and thus FMs can be optimized directly. So unlike nonlinear SVMs, a transformation in the dual form is not necessary and the model parameters can be estimated directly without the need of any support vector in the solution. We show the relationship to SVMs and the advantages of FMs for parameter estimation in sparse settings.**

On the other hand there are many different factorization models like matrix factorization, parallel factor analysis or specialized

variable interactions (comparable to a polynomial kernel in SVM), but uses a factorized parametrization instead of a dense parametrization like in SVMs. We show that the model equation of FMs can be computed in linear time and that it depends only on a linear number of parameters. This allows direct optimization and storage of model parameters without the need of storing any training data (e.g. support vectors) for prediction. In contrast to this, non-linear SVMs are usually optimized in the dual form and computing a prediction (the model equation) depends on parts of the training data (the *support vectors*). We also show that FMs subsume many of the most successful approaches for the task of collaborative filtering including biased MF, SVD++ [2], PITF [3] and FPMC [4].

In total, the advantages of our proposed FM are:

Factorization Machines

매우 Sparse한 데이터에 대해서 높은 예측 성능을 보임 (SVM의 한계 극복)

선형 복잡도를 가지므로 수십억 개의 트레이닝 데이터에 대해서도 빠르게 학습함

모델의 학습하는 파라미터의 개수도 linear함

FM은 일반적인 Supervised Learning 모델로 Regression, Classification, Ranking에 모두 활용 가능함

General Predictor → 주어진 input X에 대해서 Y를 예측하는 Task

Matrix Factorization과 다르게 일반적인 실수 피쳐를 모델의 input으로 활용하여 예측함

MF와 비교했을 때 유저, 아이템 ID 외에 다른 부가 정보들을 모델의 피쳐로 사용할 수 있음

Prediction Under Sparsity

유저의 영화에 대한 평점 데이터는 대표적인 High Sparsity 데이터

유저-아이템 매트릭스에서 다루던 Sparse Matrix와는 다른 의미

평점 데이터 = { (유저1, 영화2, 5), (유저3, 영화1, 4), (유저2, 영화3, 1), … }

일반적인 CF 문제의 입력 데이터와 같음

위 평점 데이터를 일반적인 예측 문제의 Input 데이터로 바꾼다면, 전체 유저와 아이템의 개수만큼의 feature vector가 필요하게 됨

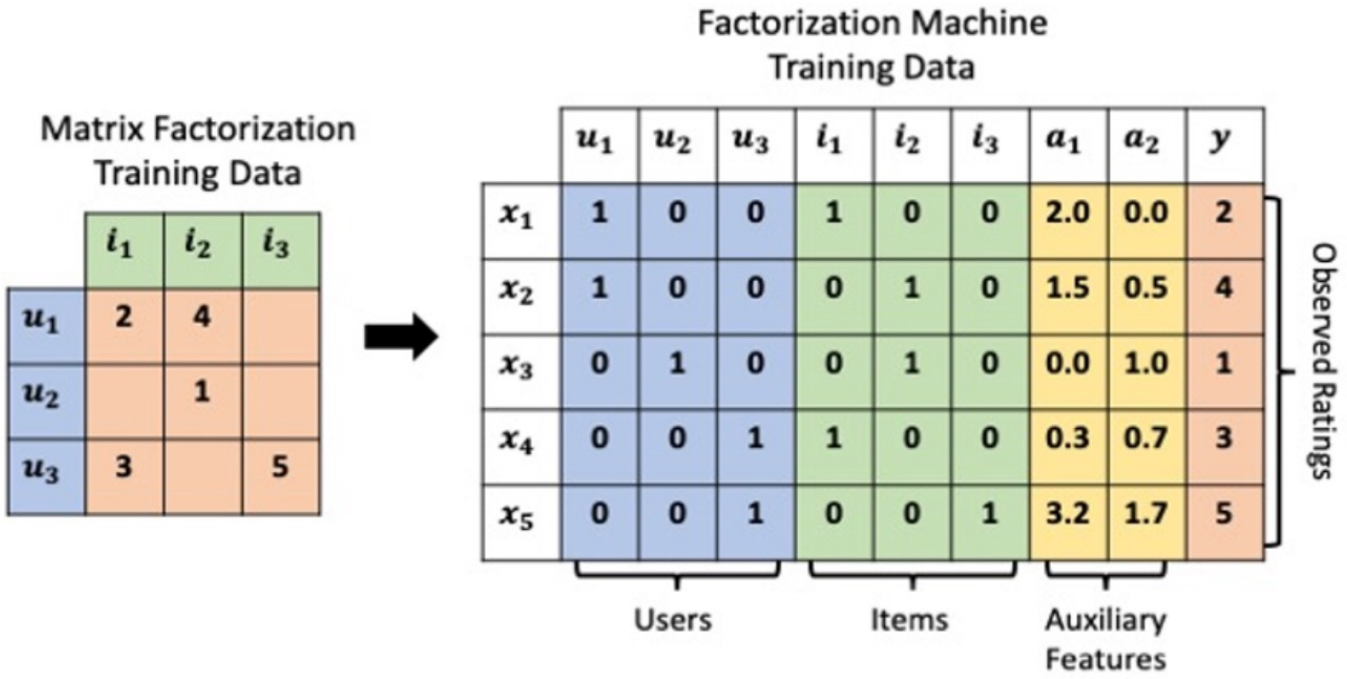
(유저1, 영화2, 5) => [1, 0, 0, …, 0, 1, 0, 0, …, 5]

(유저3, 영화1, 4) => [0, 0, 1, …, 1, 0, 0, 0, …, 4]

Prediction Under Sparsity

MF를 적용하기 위해서는 (user, item, feedback)으로만 이루어진 데이터를 사용함

→ FM의 경우 그 외의 다른 피쳐들을 제한없이 사용할 수 있음

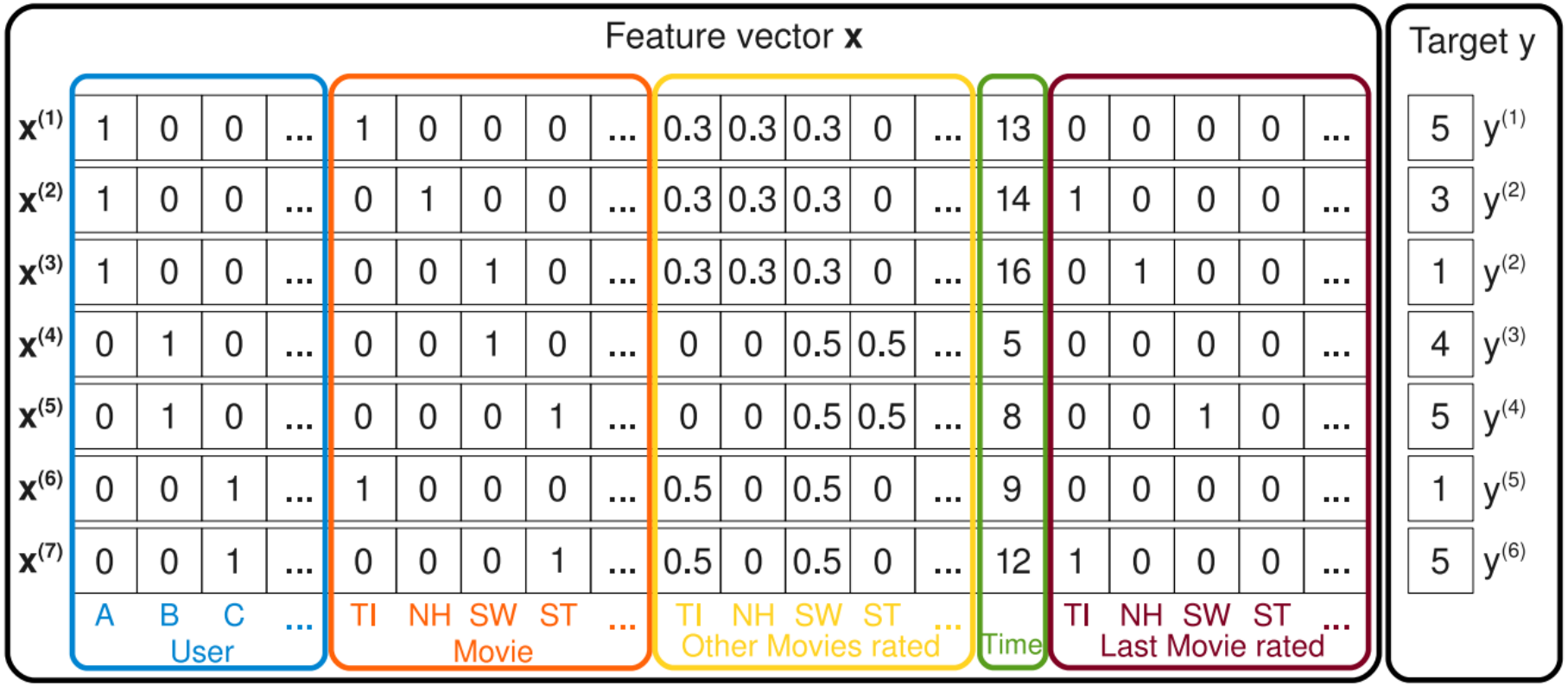


	u_1	u_2	u_3	i_1	i_2	i_3
u_1	2	4				
u_2		1				
u_3	3		5			

	u_1	u_2	u_3	i_1	i_2	i_3	a_1	a_2	y
x_1	1	0	0	1	0	0	2.0	0.0	2
x_2	1	0	0	0	1	0	1.5	0.5	4
x_3	0	1	0	0	1	0	0.0	1.0	1
x_4	0	0	1	1	0	0	0.3	0.7	3
x_5	0	0	1	0	0	1	3.2	1.7	5

Prediction Under Sparsity

부가 정보: 유저 정보, 아이템 정보, 컨텍스트 정보 (시간, 요일, 장소 등)



어떤 부가 정보를 사용하는가?

1. 유저 정보

데모 그래픽: 성별, 연령, 지역, 관심사

유저 행동 정보: 과거 선호 아이템 정보, 클릭 데이터

2. 아이템 정보

아이템의 메타 데이터

예시) 상품: 상품 카테고리, 브랜드, 출시일, 상품 이미지

3. 컨텍스트 정보

시간, 요일, 주중/주말, 현재 위치

Factorization Machine 수식

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

위 수식의 학습 파라미터는 아래와 같음

$$w_0 \in \mathbb{R}, \qquad w_i \in \mathbb{R}, \qquad \mathrm{v}_i \in \mathbb{R}^k$$

$<\cdot,\cdot>$ 는 두 벡터의 dot product를 의미함

$$< \mathrm{v}_i, \mathrm{v}_j > := \sum_{f=1}^{k} v_{i,f} \cdot v_{j,f}$$

Linear Model

$$\hat{y}(\mathrm{x}) = w_0 + \sum_{i=1}^{n} w_i x_i, \qquad w_i \in \mathbb{R}$$

FM은 기본적인 선형 다항 회귀 문제로부터 시작됨

학습 파라미터의는 입력 변수의 개수만큼 생김

하지만 Linear Model은 고차원의 데이터, 높은 Sparsity에 취약하고 변수간의 interaction을 무시하기 때문에 한계를 가짐

Interaction을 고려한 Polynomial Model

$$\hat{y}(\mathrm{x}) = w_0 + \sum_{i=1}^{n} w_i x_i, + \sum_{i=1}^{n} \sum_{j=i+1}^{n} w_{ij} x_i x_j \,, \qquad w_i, w_{ij} \in \mathbb{R}$$

Linear Model의 한계를 극복할 수 있으나 파라미터수가 급격하게 증가함

w_{ij} 파라미터의 경우 입력 변수의 제곱의 개수만큼 생김

과적합될 가능성도 높아짐

→ Factorization Machine의 경우 각 입력 변수를 저차원의 vector로 factorization하여 이를 해결함

Factorization Machines

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

위 수식의 학습 파라미터는 아래와 같음

$$w_0 \in \mathbb{R}, \qquad w_i \in \mathbb{R}, \qquad \mathrm{v}_i \in \mathbb{R}^k$$

$<\cdot,\cdot>$ 는 두 벡터의 dot product를 의미함

$$< \mathrm{v}_i, \mathrm{v}_j > := \sum_{f=1}^{k} v_{i,f} \cdot v_{j,f}$$

Factorization Machines

유저의 영화 평점 데이터에 대해서 유저 100명, 영화 100개가 존재할 때 FM으로 예측 모델을 만든다면

$x_1, \dots, x_{100}$은 유저, $x_{101}, \dots, x_{200}$은 영화에 대한 변수가 됨

모든 x_i는 0 또는 1의 binary feature

그렇다면 FM의 수식으로 3번 유저의 5번 영화에 대한 평점을 예측해보면

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

$$= w_0 + w_3 x_3 + w_{105} x_{105} + < \mathrm{v}_3, \mathrm{v}_{105} > x_3 x_{105}$$

$$= w_0 + w_3 + w_{105} + < \mathrm{v}_3, \mathrm{v}_{105} >$$

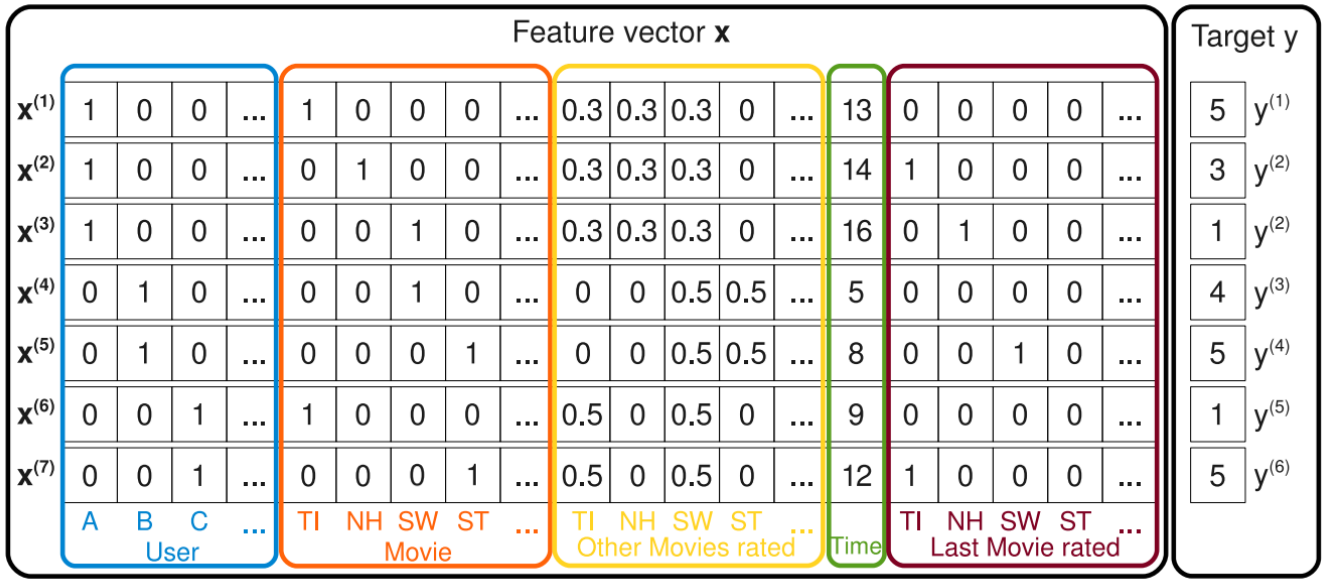
Factorization Machines

Sparsity한 환경에서 어떻게 예측을 잘하는가?

유저 A는 ST에 대한 평점이 없음 → v_A, v_{ST} FM 모델을 통해 학습되기 때문에 예측할 수 있음

유저 B, 유저 C의 ST에 대한 평점 데이터를 통해 v_{ST}가 학습이 됨

유저 A와 유저 B, C가 공유하는 영화 SW의 평점 데이터를 통해 v_A가 학습이 됨



Factorization Machines

Linear Complexity

$O(kn^2)$

$$\sum_{i=1}^{n}\sum_{j=i+1}^{n}\langle \mathbf{v}_i, \mathbf{v}_j \rangle x_i x_j$$

$$= \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\langle \mathbf{v}_i, \mathbf{v}_j \rangle x_i x_j - \frac{1}{2}\sum_{i=1}^{n}\langle \mathbf{v}_i, \mathbf{v}_i \rangle x_i x_i$$

$$= \frac{1}{2}\left(\sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{f=1}^{k} v_{i,f}\, v_{j,f}\, x_i x_j - \sum_{i=1}^{n}\sum_{f=1}^{k} v_{i,f}\, v_{i,f}\, x_i x_i\right)$$

$$= \frac{1}{2}\sum_{f=1}^{k}\left(\left(\sum_{i=1}^{n} v_{i,f}\, x_i\right)\left(\sum_{j=1}^{n} v_{j,f}\, x_j\right) - \sum_{i=1}^{n} v_{i,f}^2\, x_i^2\right)$$

$O(kn)$

$$= \frac{1}{2}\sum_{f=1}^{k}\left(\left(\sum_{i=1}^{n} v_{i,f}\, x_i\right)^2 - \sum_{i=1}^{n} v_{i,f}^2\, x_i^2\right)$$

Factorization Machines

General Predictor

Regression: $\hat{y}(\mathrm{x})$를 예측해야 하는 값으로 놓고 실측값 y와의 차이를 Least Square Error 등으로 두어 학습시키는 방법

영화 평점 1 ~ 5점을 예측하는 문제

Binary Classification: $\hat{y}(\mathrm{x})$를 0 / 1의 binary 값을 예측하도록 하며, 이 경우 hinge loss나 logit loss를 사용하여 학습

일반적인 분류 문제, CTR 예측 문제

Ranking: (x^a, x^b)의 pair에 대해서 $\hat{y}(\mathrm{x})$를 score로 사용하여 랭킹문제로 치환할 수 있으며, 이 때는 pairwise classification loss를 사용하여 학습

FM으로 Matrix Factorization을 표현

Factorization Machines

i는 FM의 모든 입력 변수를 의미함

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

Matrix Factorization

u는 특정 유저, i는 특정 아이템을 의미함

$$\hat{y}(\mathrm{x}) = w_0 + w_u + w_i + < \mathrm{v}_u, \mathrm{v}_i >$$

FM으로 Matrix Factorization을 표현

유저의 영화 평점 데이터에 대해서 유저 100명, 영화 100개가 존재할 때 FM으로 예측 모델을 만든다면

$x_1, \dots, x_{100}$은 유저, $x_{101}, \dots, x_{200}$은 영화에 대한 변수가 됨

모든 x_i는 0 또는 1의 binary feature

그렇다면 FM의 수식으로 3번 유저의 5번 영화에 대한 평점을 예측해보면

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

$$= w_0 + w_3 x_3 + w_{105} x_{105} + < \mathrm{v}_3, \mathrm{v}_{105} > x_3 x_{105}$$

$$= w_0 + w_3 + w_{105} + < \mathrm{v}_3, \mathrm{v}_{105} >$$

Auxiliary Feature를 사용해보자

유저 100명, 영화 100개 외에 2개 유저 성별과 10개의 영화 장르의 정보도 추가로 사용한다면, 성별은 x_{201}, x_{202}, 장르는 $x_{203}, \ldots, x_{212}$로 입력변수를 추가할 수 있음

성별과 장르 모두 0 또는 1의 binary feature로 표현, 총 212개의 입력 변수가 존재함

그렇다면 FM의 수식으로 3번 유저(남자)의 5번 영화(1번 장르)에 대한 평점을 예측해보면,

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

$$= w_0 + w_3 x_3 + w_{105} x_{105} + w_{201} x_{201} + w_{203} x_{203} + < \mathrm{v}_3, \mathrm{v}_{105} > x_3 x_{105} + \cdots + < \mathrm{v}_{201}, \mathrm{v}_{203} > x_{201} x_{203}$$

$$= w_0 + w_3 + w_{105} + w_{201} + w_{203} + < \mathrm{v}_3, \mathrm{v}_{105} > + < \mathrm{v}_3, \mathrm{v}_{201} > + \cdots + < \mathrm{v}_{105}, \mathrm{v}_{203} > + < \mathrm{v}_{201}, \mathrm{v}_{203} >$$

우리가 수행할 추천 Task

1. 영화 평점 예측

기존에 MF나 다른 알고리즘은 유저와 아이템의 pair에 대한 평점 데이터를 사용해 평점을 예측함

유저, 아이템 ID외에 유저의 정보와 아이템의 정보를 추가로 사용하여 FM으로 평점을 예측해보자

유저: 성별, 연령, 지역 등

아이템: 영화의 장르, 출시연도 등

이 때 FM은 일반적인 Regression 모델처럼 사용되어 평점 값을 예측함

예측해야 하는 y label은 1 ~ 5점의 평점이고 Least Square Error를 최소화하는 방향으로 모델이 학습됨

우리가 수행할 추천 Task

2. 광고 CTR 예측

CTR 예측 문제는 유저가 주어진 아이템을 클릭할 확률을 예측하는 문제이다

예측해야 하는 y label은 0 또는 1이므로 binary classification 문제

FM의 output $\hat{y}(\mathbf{x})$을 sigmoid에 통과시키면 0과 1 사이의 출력되고, 이 값이 곧 예측 CTR이 됨

광고가 노출된 상황의 다양한 유저, 광고, 컨텍스트 피쳐를 FM의 입력 변수로 사용한다

유저 ID가 존재하지 않는 데이터도 다른 유저 피쳐나 컨텍스트 피쳐를 사용하여 예측할 수 있음

실제로 현업에서는 유저 ID를 피쳐로 사용하지 않는 경우가 많음

참고. log loss란?

Binary Classification에 사용하는 손실함수로 log loss를 최소화시키는 방향으로 모델이 학습됨

정보이론 관점에서 cross entropy라고 불리며 negative log likelihood로 표현됨

어떤 데이터가 0 또는 1로 예측될 확률을 $\hat{y}$, 실측값을 y라고 하면,

$$likelihood = \hat{y}^{y}(1-\hat{y})^{1-y}$$

$$log\ likelihood = y\log\hat{y} + (1-y)\log(1-\hat{y})$$

$$log\ loss = negative\ log\ likelihood = -y\log\hat{y} - (1-y)\log(1-\hat{y})$$

FM 지원 라이브러리

libFM: 저자가 공개한 C++ 라이브러리

fastFM: Academic Use의 FM 라이브러리

xLearn: high performance, easy-to-use, and scalable을 강조, 현업에서도 사용되며 online learning도 가능함

pytorch-fm: pytorch로 구현된 FM 계열의 라이브러리 사용 가능, FM으로부터 발전된 다양한 딥러닝 모델도 지원함

FM으로 Tensorflow로 구현
영화 평점과 CTR 예측하기

3. Field-aware Factorization Machine

Field-aware Factorization Machine

Factorization Model의 변형된 모델로서 더 높은 성능을 보이는 FFM 모델을 제안한 논문

Field-aware Factorization Machines for CTR Prediction

Yuchin Juan
Criteo Research*
Palo Alto, CA
yc.juan@criteo.com

Yong Zhuang
Dept. of ECE*
Carnegie Mellon Univ., USA
yong.zhuang22@gmail.com

Wei-Sheng Chin
Dept. of Computer Science
National Taiwan Univ., Taiwan
d01944006@csie.ntu.edu.tw

Chih-Jen Lin
Dept. of Computer Science
National Taiwan Univ., Taiwan
cjlin@csie.ntu.edu.tw

ABSTRACT

Click-through rate (CTR) prediction plays an important role in computational advertising. Models based on degree-2 polynomial mappings and factorization machines (FMs) are widely used for this task. Recently, a variant of FMs, field-aware factorization machines (FFMs), outperforms existing models in some world-wide CTR-prediction competitions. Based on our experiences in winning two of them, in this paper we establish FFMs as an effective method for classifying large sparse data including those from CTR prediction. First, we propose efficient implementations for training FFMs. Then we comprehensively analyze FFMs and compare this approach with competing models. Experiments show that FFMs are very useful for certain classification problems. Finally, we have released a package of FFMs for public use.

		Publisher	Advertiser
+80	−20	ESPN	Nike
+10	−90	ESPN	Gucci
+0	−1	ESPN	Adidas
+15	−85	Vogue	Nike
+90	−10	Vogue	Gucci
+10	−90	Vogue	Adidas
+85	−15	NBC	Nike
+0	−0	NBC	Gucci
+90	−10	NBC	Adidas

Table 1: An artificial CTR data set, where + (−) represents the number of clicked (unclicked) impressions.

In problem (1), λ is the regularization parameter, and in the loss function we consider the linear model:

Introduction

Factorization Machine은 General Predictor로 다양한 Task에서 좋은 성능을 보였으며, 특히 sparse 데이터로 이루어진 CTR 예측에서 좋은 성능을 보임

Field-aware Factorization Machine (FFM)은 FM을 발전시킨 모델로서 입력 변수를 Field로 나누어 Field에 따라 서로 다른 Latent Factor로 파라미터를 Factorize함

기존의 FM은 하나의 변수에 대해서 k개로 Factorize했으나 FFM은 f개의 필드에 대해 각각 k개로 Factorize함

Field는 모델을 설계할 때 정의하며 같은 의미를 갖는 변수들의 집합으로 설정함

유저: 성별, 디바이스, 운영체제

아이템: 광고, 카테고리

컨텍스트: 어플리케이션, 배너

Introduction

FFM은 2-degree Polynomial Model이나 FM보다 더 좋은 성능을 보임

CTR Prediction 데이터인 Criteo와 Avazu 데이터를 활용

일반적인 Stochastic Gradient Descent를 사용해 파라미터를 학습함

Adagrad를 사용하여 Learning Rate를 조절함

과적합을 피하기 위해서 학습 데이터의 Epoch는 딱 1번만 사용함

CTR 예측 모델은 Epoch를 1번만 사용하는 것이 일반적임

FFM의 원리

FFM은 추천 시스템을 위해 제안한 PITF 모델에서 아이디어를 얻음

TF: Tensor Factorization

PITF에는 (User, Item, Tag) 3개의 필드에 대해서 클릭을 예측하는데, 이 때 (User, Item), (Item, Tag), (User, Tag)에 대해서 서로 다른 latent factor를 정의해서 구함

→ 이를 일반화하여 여러 개의 필드에 대해서도 latent factor를 정의한 것이 FFM

CTR 예측에 사용되는 피쳐는 이보다 훨씬 다양한데, 피쳐의 개수만큼 필드를 정의하여 FFM 모델을 사용할 수 있음

FM과 FFM

광고 클릭 데이터가 존재하고 사용할 수 있는 피쳐는 Publisher, Advertiser, Gender 총 3개가 존재

FM의 경우는 필드는 존재하지 않고 하나의 변수에 factorization dimension(k)만큼의 파라미터를 학습

linear term인 w_i와 factorization term인 v_i

FM Formula

$$\hat{y}(\mathrm{x}) = w_0 + w_{ESPN} + w_{Nike} + w_{Male} + \mathrm{v}_{ESPN} \cdot \mathrm{v}_{Nike} + \mathrm{v}_{ESPN} \cdot \mathrm{v}_{Male} + \mathrm{v}_{Nike} \cdot \mathrm{v}_{Male}$$

Clicked	Publisher (P)	Advertiser (A)	Gender (G)
Yes	ESPN	Nike	Male

FM과 FFM

광고 클릭 데이터가 존재하고 사용할 수 있는 피쳐는 Publisher, Advertiser, Gender 총 3개가 존재할 때,
FFM은 각각의 피쳐를 필드 P, A, G로 정의함

FFM의 경우는 하나의 변수에 대해서 필드 개수와 factorization 차원의 곱(fk)만큼의 파라미터를 학습

linear term인 w_i와 factorization term인 $\mathrm{v}_{i,f}$

FFM Formula

$$\hat{y}(\mathrm{x}) = w_0 + w_{ESPN} + w_{Nike} + w_{Male} + \mathrm{v}_{ESPN,A} \cdot \mathrm{v}_{Nike,P} + \mathrm{v}_{ESPN,G} \cdot \mathrm{v}_{Male,P} + \mathrm{v}_{Nike,G} \cdot \mathrm{v}_{Male,A}$$

Clicked	Publisher (P)	Advertiser (A)	Gender (G)
Yes	ESPN	Nike	Male

FFM Formula

FFM

$$\hat{y}(\mathrm{x}) = w_0 + \sum_{i=1}^{n} w_i x_i + \sum_{i=1}^{n} \sum_{j=i+1}^{n} < \mathrm{v}_{i,f_j}, \mathrm{v}_{j,f_i} > x_i x_j$$

$$w_0 \in \mathbb{R}, \qquad w_i \in \mathbb{R}, \qquad \mathrm{v}_{i,f} \in \mathbb{R}^k$$

FFM Formula

FFM

$$\hat{y}(\mathrm{x}) = w_0 + \sum_{i=1}^{n} w_i x_i + \sum_{i=1}^{n} \sum_{j=i+1}^{n} < \mathrm{v}_{i,f_j}, \mathrm{v}_{j,f_i} > x_i x_j$$

$$w_0 \in \mathbb{R}, \qquad w_i \in \mathbb{R}, \qquad \mathrm{v}_{i,f} \in \mathbb{R}^k$$

FM

$$\hat{y}(\mathrm{x}) = w_0 + \sum_{i=1}^{n} w_i x_i + \sum_{i=1}^{n} \sum_{j=i+1}^{n} < \mathrm{v}_i , \mathrm{v}_{j,} > x_i x_j$$

$$w_0 \in \mathbb{R}, \qquad w_i \in \mathbb{R}, \qquad \mathrm{v}_i \in \mathbb{R}^k$$

FM과 FFM 비교

FM의 Complexity는 $O(kn)$이고 학습 파라미터의 개수는 nk

n은 입력 변수의 개수, k는 factorization 차원

FFM의 경우 Complexity는 $O(kn^2)$이고 학습 파라미터의 개수는 nfk

f는 필드의 개수

factorization 차원의 경우 일반적으로 $k_{FFM} \ll k_{FM}$

FFM은 필드를 사용하기 때무에 FM보다 더 적은 차원의 latent factor로 factorization시켜도 됨

참고.

FFM의 경우 Complexity가 더 높지만 딥러닝에 적용하는 Computational Power를 고려하면 큰 이슈 없음

FFM의 필드 구성

Categorical Feature

FM 피쳐

label feat1:val1 feat2:val2 ···,

Yes P-ESPN:1 A-Nike:1 G-Male:1

FFM 피쳐

label field1:feat1:val1 field2:feat2:val2 ···

Yes P:P-ESPN:1 A:A-Nike:1 G:G-Male:1.

FFM의 필드 구성

Numerical Feature

Accepted	AR	Hidx	Cite
Yes	45.73	2	3
No	1.04	100	50,000

1. dummy field

numeric feature 한 개당 하나의 필드에 할당하고 real value를 사용

예시) Yes　AR:AR:45.73　Hidx:Hidx:2　Cite:Cite:3

2. discretize

numeric feature를 n개의 구간으로 나누어 binary value를 사용하고, n개의 변수를 하나의 필드에 할당

예시) Yes　AR:45:1　Hidx:2:1　Cite:3:1

FFM의 성능 비교

CTR 예측 문제이므로 성능 비교는 log loss를 사용함

	statistics			logloss			
Data set	# instances	# features	# fields	LM	Poly2	FM	FFM
KDD2010-bridge	20,012,499	651,166	9	0.27947	0.2622	0.26372	<u>0.25639</u>
KDD2012	149,639,105	54,686,452	11	0.15069	0.15099	0.15004	<u>0.14906</u>
phishing	11,055	100	30	0.14211	0.11512	<u>0.09229</u>	0.1065
adult	48,842	308	14	0.3097	0.30655	0.30763	<u>0.30565</u>
cod-rna (dummy fields)	331,152	8	8	0.13829	0.12874	<u>0.12580</u>	0.12914
cod-rna (discretization)	331,152	2,296	8	0.16455	0.17576	0.16570	<u>0.14993</u>
ijcnn (dummy fields)	141,691	22	22	0.20093	0.08981	0.07087	<u>0.0692</u>
ijcnn (discretization)	141,691	69,867	22	0.21588	0.24578	0.20223	<u>0.18608</u>

Table 4: Comparison between LM, Poly2, FM, and FFMs. The best logloss is underlined.

FFM 지원 라이브러리

libFFM: 저자가 공개한 C++ 라이브러리

xLearn: high performance, easy-to-use, and scalable을 강조, 현업에서도 사용되며 online learning도 가능함

pytorch-fm: pytorch로 구현된 FM 계열의 라이브러리 사용 가능, FM으로부터 발전된 다양한 딥러닝 모델도 지원함

xlearn FFM 라이브러리로 CTR 예측