E-Commerce Recommender System

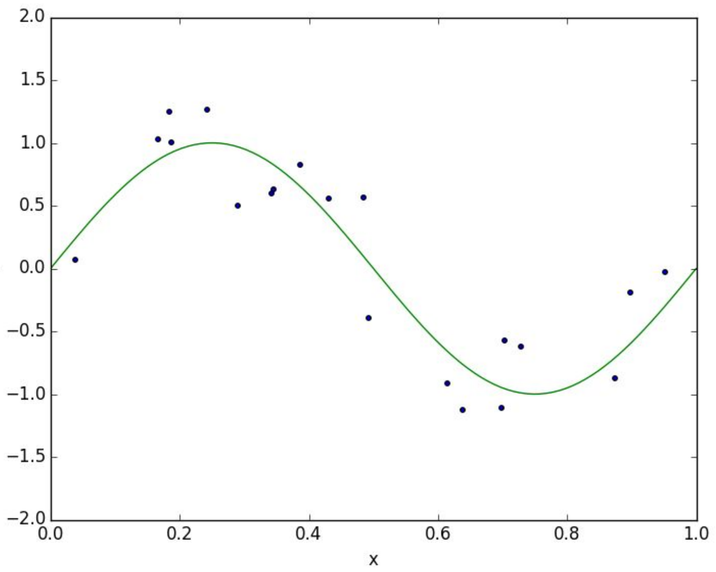
Florrie Cheng

VMware

Outline

- Brief Introduction to Machine Learning
- What is a Recommender System
- Data Preprocessing
- Main Algorithms
- Data Post-processing
- CTR Prediction
- Demo

Case: Linear Regression 线性回归



$$h(x) = wx + b$$

$$h(x) = w_1x_1 + w_2x_2 + \dots + w_nx_n + b$$

Polynomial Regression

$$h(x) = w_1x + w_2x^2 + \dots + w_nx^n + b$$

Error:

For single sample

$e = (y' - y)^2$ // y' is predicted, y is ground truth

Cost Function:

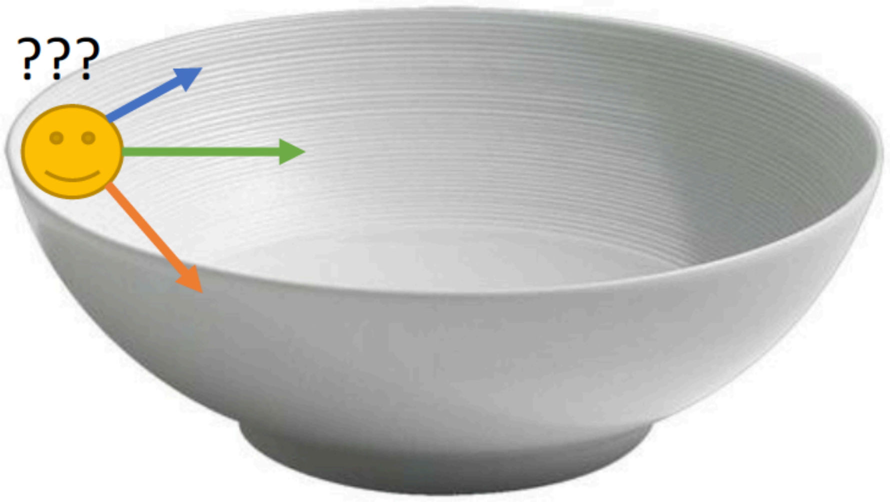
J = Avg(Sum(All sample error))

Optimization:

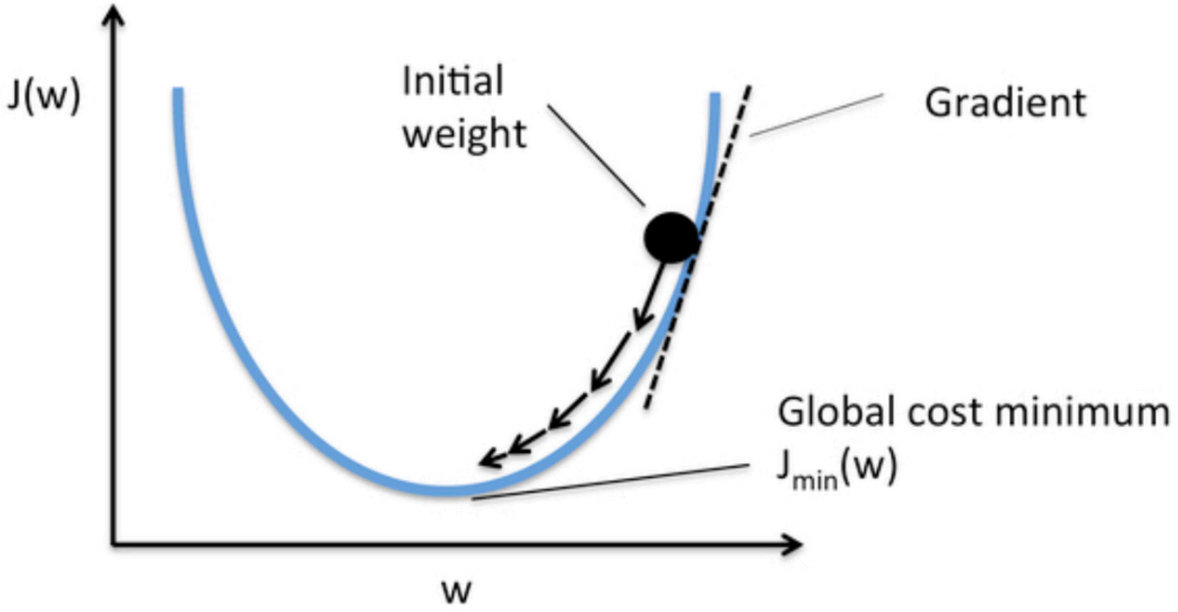
$$\theta_j := \theta_j - \alpha \frac{\partial}{\partial \theta_j} J$$

α = learning rate

Gradient Descent 梯度下降



Gradient Descent 梯度下降



$w_j = w_j - \alpha(y' - y)x_j$

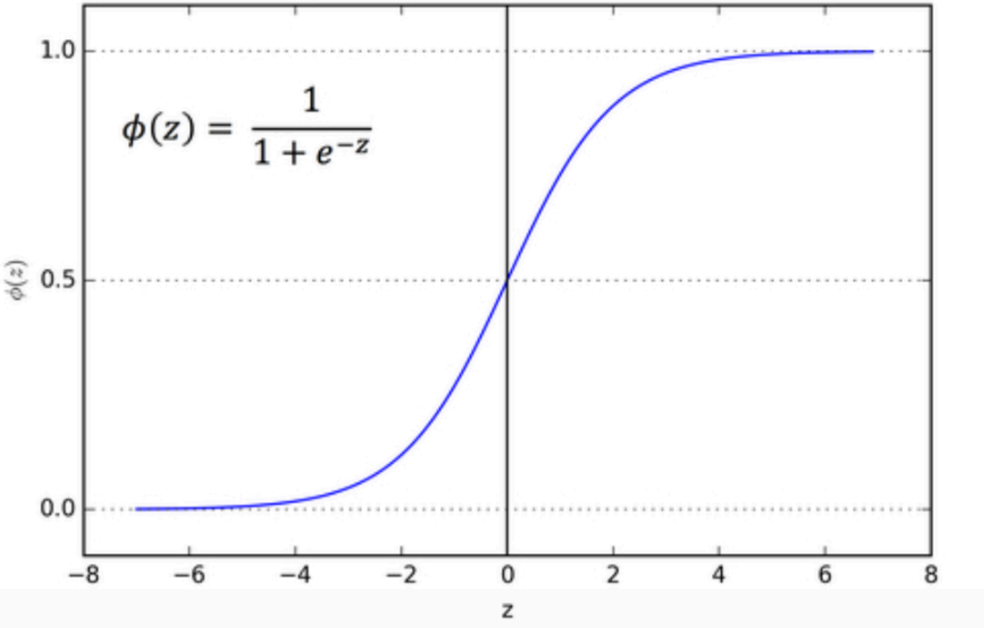
一次用一个样本更新每个w的值

Batch & Stochastic

一次用多个样本更新每个w的值

Logistic Regression

$$h(x) = \frac{1}{1+\exp(-w^T x)}$$



- $h(x) \in [0,1]$
- $y \in (-\infty, +\infty)$

$$sigmoid(z) = \sigma(z) = \frac{e^z}{1+e^z} = \frac{1}{1+e^{-z}}$$

激活函数

Softmax Regression

$$h_\theta(x^{(i)}) = \begin{bmatrix} p(y^{(i)} = 1 | x^{(i)}; \theta) \\ p(y^{(i)} = 2 | x^{(i)}; \theta) \\ \vdots \\ p(y^{(i)} = k | x^{(i)}; \theta) \end{bmatrix} = \frac{1}{\sum_{j=1}^{k} e^{\theta_j^T x^{(i)}}} \begin{bmatrix} e^{\theta_1^T x^{(i)}} \\ e^{\theta_2^T x^{(i)}} \\ \vdots \\ e^{\theta_k^T x^{(i)}} \end{bmatrix}$$

- 被训练的参数为一个矩阵

- 假设最终输出为y

  y 是一个k维向量，$y_i \in [0,1]$，$\mathrm{sum}(y_0 \ldots y_k) = 1$

  k = 分类的总数

- 解决多分类问题

E-Commerce Recommender System

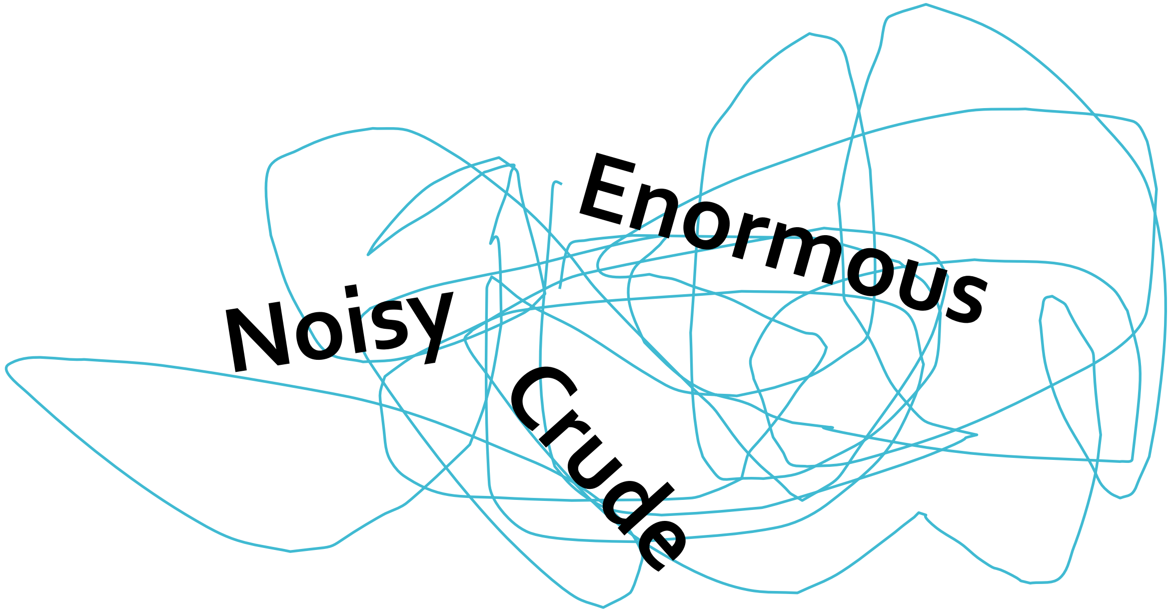
电商推荐系统

根据用户购买商品的历史数据，商品信息，用户信息，等等一切信息，为用户推荐他可能会感兴趣的商品

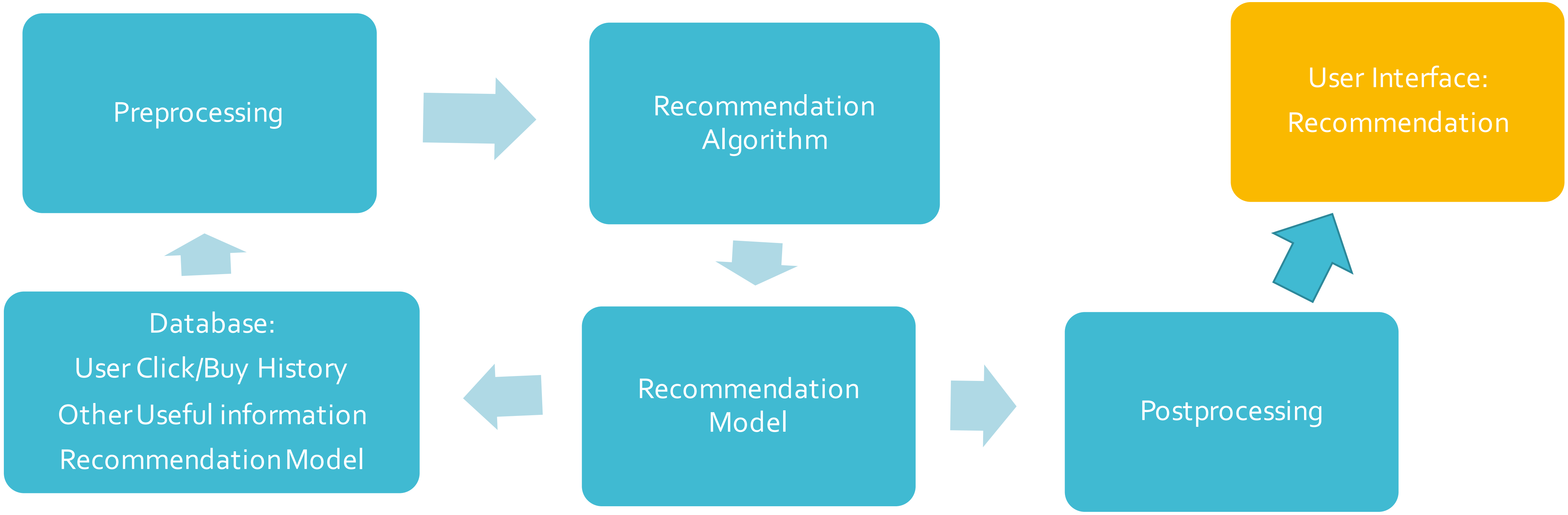
要求

相关性：用户感兴趣
实时性：快速推荐
推荐范围：包含新用户
有效性：确实对用户有利

Problems



General Architecture



Preprocessing

Down Sampling

stochastic

most recent data

Alignment

Normalization

$$x_i^{normed} = \frac{x_i - \mu}{\sigma}$$

均值

方差

Remove noise or outlier

Missing value

K-Nearest Neighbor 均值

Algorithms

Content-based Filtering

Collaborative Filtering

Word2Vec

Topic Model

Content-based Filtering

Popularity-based

Category-based

Keyword-based

User-information-based

……

Algorithms

Content-based Filtering

Collaborative Filtering

Word2Vec

Topic Model

Collaborative Filtering 协同过滤

根据用户购买、浏览行为计算用户或商品之间的相似度进行推荐

- user-based：计算用户相似度，推荐同类用户购买过的其他商品
- item-based：计算商品相似度，推荐用户已购买过商品的同类商品

工业界一般用item-based，因为用户数量>>商品数量，计算用户相似性难度更大

Item-based

商品之间的相似度：Jaccard Similarity

$$J(A,B) = \frac{|\mathrm{set}(A) \cap \mathrm{set}(B)|}{|\mathrm{set}(A) \cup \mathrm{set}(B)|}$$

商品A与B的相似性 = 购买它们的用户集合的相似性

Item-based

Let $S_{i,j}$ denote the similarity between item i and j

$$d_{u,i} = \begin{cases} 1 & \text{if user } u \text{ recently purchased item } i \\ 0 & \text{otherwise} \end{cases}$$

- 计算用户 u 和商品 i 的相关性 score

$$score(u,i) = \frac{\sum_{j \in \text{neighbors(i)}} d_{u,j} s_{j,i}}{\sum_{j \in \text{neighbors(i)}} s_{j,i}}$$

Item-based

- 对每个用户u，选取与他相关性最大的k个商品进行推荐
- 优点：算法简单易懂，推荐效果优良，结合了人类智能
- 缺点: 忽视商品之间隐含的关系
  - 例如:一年之内的数据，购买了苹果电脑的人可能不会买戴尔电脑，反之亦然。在协同过滤中苹果电脑和戴尔电脑的相似度就变得很低。
  - 经常被同一个用户购买的两个商品可能毫无关联：例如沐浴露和零食

可能改进的办法：

- 训练集中去掉被绝大多数人购买过的商品
- 寻找machine learned features

Algorithms

Content-based Filtering

Collaborative Filtering

Word2Vec

Topic Model

Word2Vec

最初用于自然语言处理：将单词转化为向量

在推荐系统中，可用于将一个商品转化为拥有若干feature的向量

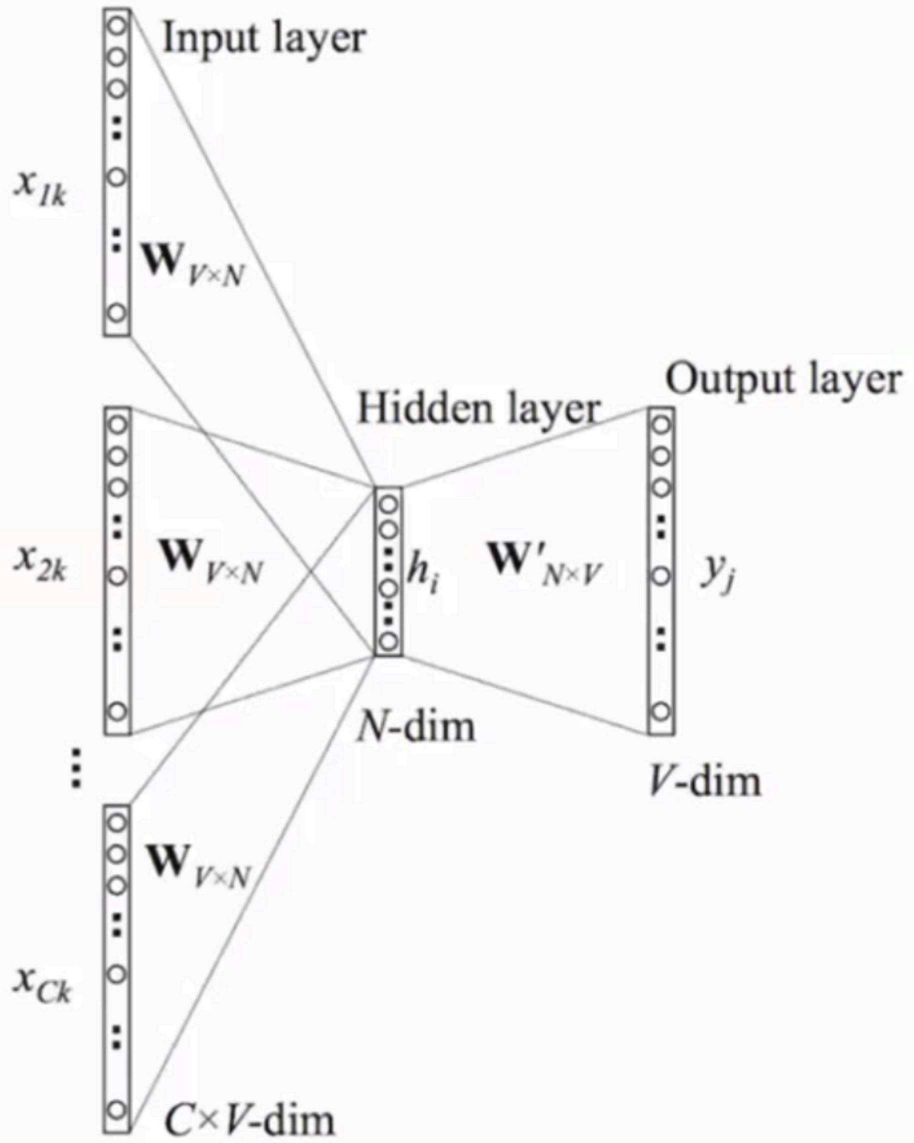
Word2Vec

对于每个用户购买的所有商品：

CBOW: 选取一个作为output，剩下作为input

Skip-Gram: 选取一个作为input，剩下作为output

- CBOW model 的网络结构
  - 2 层 feedforward neural network
  - 第 1 层没有 activation function
  - 第 2 层是 Softmax
  - 目标函数是 maximize log-likelihood
  - 用 One-hot representation 把每一个商品表示成一个向量，其中
    - V ： 商品的数量
    - C ： session 里的 rest 商品数量



- CBOW model
  - Let $D = \{(x_{j,1}, x_{j,2}, ..., x_{j,C_j}, y_j) | j = 1, ..., n\}$

    $x_{j,1}, x_{j,2}, ..., x_{j,C_j}$: 一个 session 里其他所有的商品

    y_j: 一个 session 里的 response 商品
  - 计算 hidden layer

    $$\boxed{h_j^T = \left(\frac{1}{C}\sum_{k=1}^{C_j} x_{j,k}^T\right) W}$$
  - 计算 output

    $$\boxed{o_j = \mathrm{softmax}(h_j W^{'})}$$
  - Objective function

    $$\boxed{\arg_{W,W'} \max \sum_{j}^{n} \log\left(\mathrm{softmax}(h_j W^{'}) y_j\right)}$$

W是一个V ✖ N的矩阵，每一行对应一个商品的表示

利用item-kNN做后续商品推荐

Algorithms

Content-based Filtering

Collaborative Filtering

Word2Vec

Topic Model

Topic Model: Latent Semantic Index

- 建立 item 和 user 的矩阵

$$X_{i,u} = \begin{cases} 1.0 & \text{if user } u \text{ purchased item } i \\ 0.0 & \text{otherwise} \end{cases}$$

Topic Model: Latent Semantic Index

- Run SVD on X to perform low-rank approximation

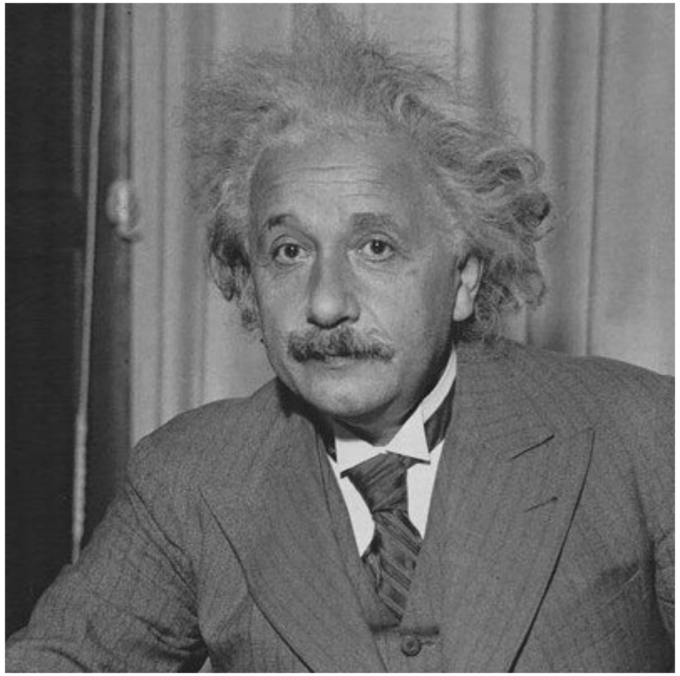
$$\boxed{X \approx U \Sigma V^\top}$$

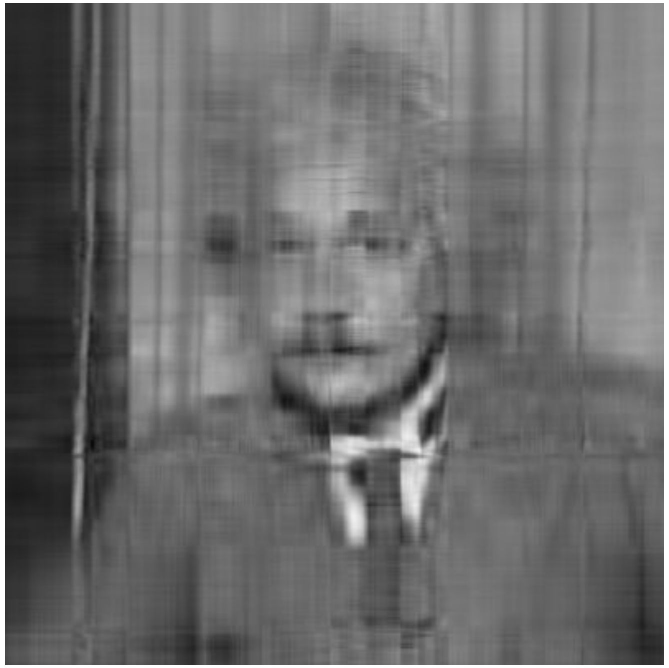
其中 U 是一个 $N \times K$ 的矩阵，并且每一列都是正交的

Σ 是一个 $K \times K$ 的对角阵

V 是一个 $L \times K$ 的矩阵， 并且每一列都是正交的

N 是商品数量，L 是用户数量，要求 $K \ll N$ 和 $K \ll L$





- 计算商品向量的内积

$$\boxed{XX^\top \approx U\Sigma V^\top (U\Sigma V^\top)^\top = U\Sigma (U\Sigma)^\top}$$

可以认为 $U\Sigma$ 表示每一个商品

- 去噪音
- 发现隐含的商品关系

- Probabilistic LSI
- Latent Dirichlet Allocation: LDA

Content-based Filtering: Category-Based

Recommend items based on its category

One of the most direct way

Usually act as backfill when there're not enough items recommended

Collaborative Filtering: Item-Based

Recommend items that are most similar to the known one

Should preprocess the data to compute similarity between each pair of items

Notable Performance dealing with large dataset

Word2Vec

CBOW Model

Transfer every item into a vector

Reveal potential features

Topic Model: Latent Semantic Index

SVD, Low-rank approximation

Reveal potential connection between items & remove noise

Hard to be described by probability

Postprocessing

Diversity Maximization

Re-ranking

Diversity Maximization

Postprocessing

Diversity Maximization

Re-ranking

Re-ranking

根据不同目标，重新排列推荐商品的顺序

- 最大化click through
- 最大化conversion
- 最大化profit
- 重点推销某类商品
- ……

CTR Prediction

CTR Prediction

- Click-through rate prediction
- Conversion rate prediction

收集被推荐用户的反馈，记录推荐商品是否被点击或购买

$$\mathrm{CTR}(\text{item } i) = \frac{\alpha_i}{\beta_i}$$

α_i : num of clicks on item i

β_i : num of sends on item i

Not Reliable Enough

- bayes smoothing within category

$$\mathrm{CTR}(\text{item } i) = \frac{\alpha_i + \lambda\eta_i}{\beta_i + \lambda\gamma_i}$$

η_i : average num of clicks on item within the same category as item i

γ_i : average num of sends on item within the same category as item i

- Logistic regression model

提取不同feature（用户信息，商品信息，环境信息，用户购买历史数据）

输出是该商品会被点击/转化 的概率

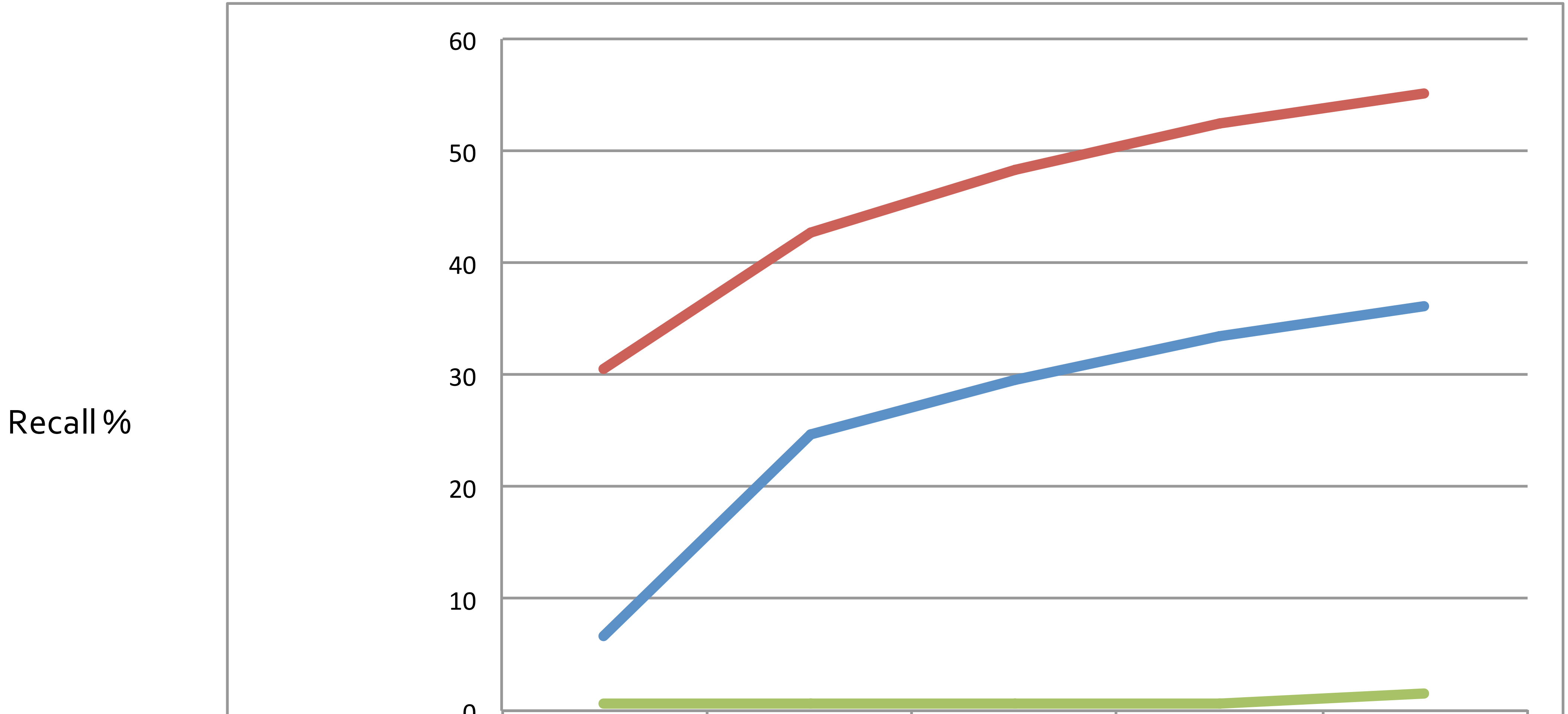
DEMO

Performance Analysis

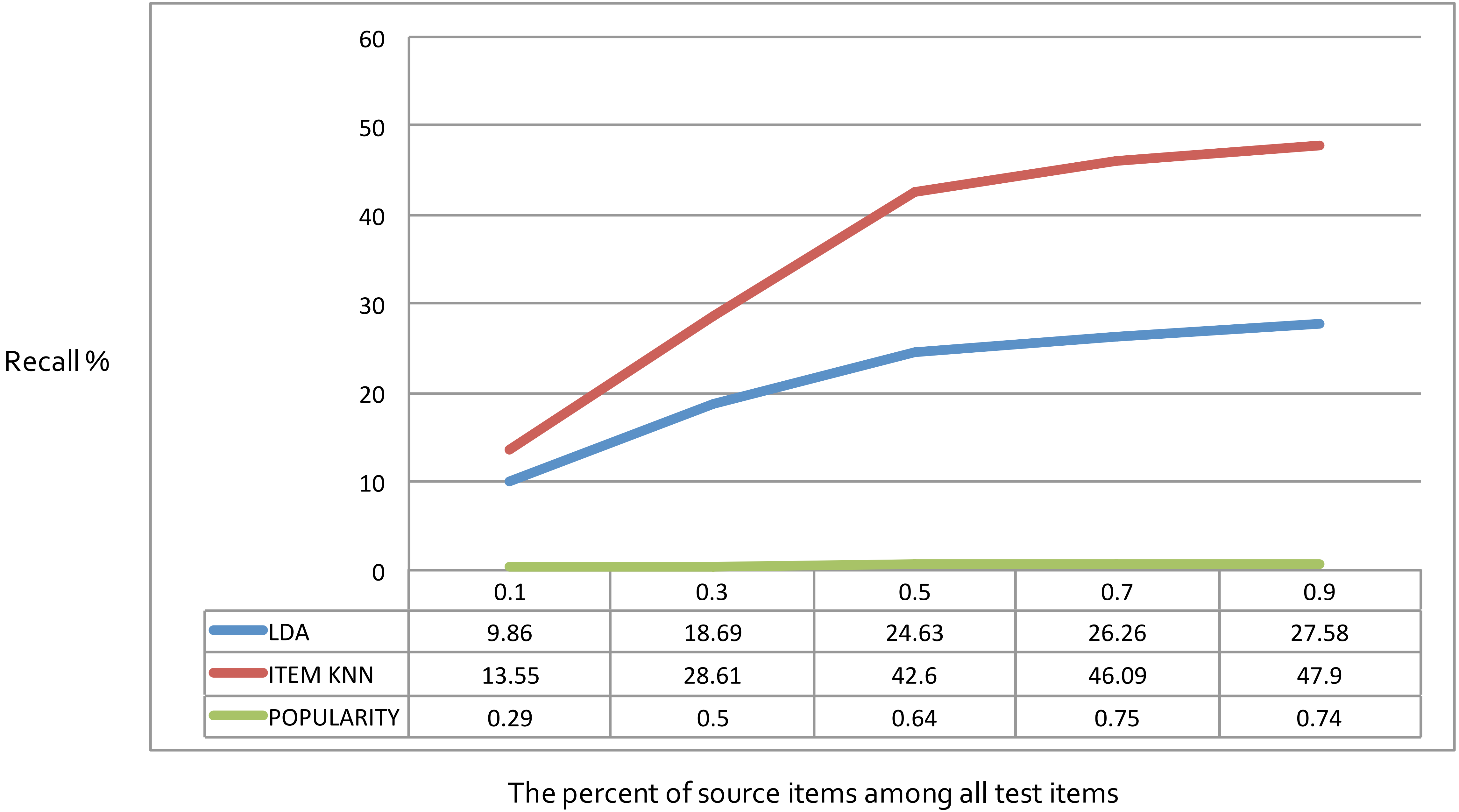
Recommendation Number 10

Source items : Test items 1:1

	Recall	Precision	Throughput (times per second)	Training Time
Popularity-based	0.64%	0.12%	7812500	<10 min
Item KNN	42.60%	7.33%	142	30 min – 60 min
LDA	24.63%	4.56%	233	20+ hours



	5	10	15	20	25
LDA	6.67	24.63	29.55	33.4	36.06
ITEM KNN	30.52	42.6	48.24	52.39	55.08
POPULARITY	0.64	0.64	0.67	0.67	1.44



	0.1	0.3	0.5	0.7	0.9
LDA	9.86	18.69	24.63	26.26	27.58
ITEM KNN	13.55	28.61	42.6	46.09	47.9
POPULARITY	0.29	0.5	0.64	0.75	0.74

Some Reflection

1. 不是越复杂的模型就越有效
2. 在开发机器学习应用中，依靠人工干预加入先验信息同样重要
3. 算法选择上，除了准确率，还需要考虑并行度
4. 有的时候得不到好的结果需要进行超参数调优

Reference

Sklearn的preprocessing库
http://scikit-learn.org/stable/modules/preprocessing.html

Word2vec tensorflow官方示例
https://github.com/tensorflow/tensorflow/blob/r0.12/tensorflow/examples/tutorials/word2vec/word2vec_basic.py

Sklearn Truncated SVD库
http://scikit-learn.org/stable/modules/generated/sklearn.decomposition.TruncatedSVD.html#sklearn.decomposition.TruncatedSVD

Sklearn LDA库
https://github.com/lda-project/lda

Demo代码地址
https://github.com/chyt123/capstone

天池大数据比赛：衣物搭配推荐（Mar. 1 截止）
https://tianchi.aliyun.com/getStart/introduction.htm?spm=5176.100066.0.0.2ee90339MYktOp&raceId=231575