3.1 k近邻算法

k近邻算法（k-Nearest Neighbor Algorithm）

- 用于分类和回归的非参数方法
  - 对于每个输入实例x，在特征空间中找到k个最接近的训练实例$N_k(x)$
  - 对于回归问题：对x的预测基于k个实例的标签的平均值

$$\hat{y}(x) = \frac{1}{k} \sum_{x_i \in N_k(x)} y_i$$

  - 对于分类问题：做邻居间的多数投票

$$\hat{y}(x) = \text{majority vote } \{y_i, x_i \in N_k(x)\}$$

k近邻算法（k-Nearest Neighbor Algorithm）

- 广义版本
  - 定义输入实例x与其邻居x_i之间的相似度函数$s(x, x_i)$
  - 然后基于邻近相似性的加权标签平均值进行预测

$$\hat{y}(x) = \frac{\sum_{x_i \in N_k(x)} s(x, x_i) y_i}{\sum_{x_i \in N_k(x)} s(x, x_i)}$$

非参数KNN

- 没有参数需要学习
  - 实际上有N个参数：每个实例都是一个参数
  - 有N/k个有效参数
    - 直觉上讲，如果邻域不重叠，就会有N/k个邻域，每个邻域都对应一个参数
  - 超参数k
    - 我们不能将训练集上的平方误差和作为选择k的标准，因为$k = 1$时结果总是最好的
    - 在验证集上调整k的取值

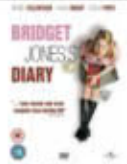
信息过滤

- 信息过滤处理用户可能感兴趣或有用的信息的传递
  - 推荐系统：以建议的形式进行信息过滤
  - 两种信息过滤方法
    - 基于内容的过滤
      - 推荐与用户喜欢的电影共享演员/导演/流派的电影
    - 协同过滤（Collaborative filtering）
      - 推荐与用户有类似兴趣的用户喜欢的电影

http://recommender-systems.org/information-filtering/

3.2 基于近邻算法的协同过滤

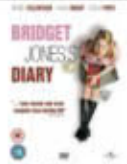
一个空的评分条目

	 Die Hard	 Mission: Impossible	 GoldenEye	 Casino Royale	 Titanic	 Notting Hill	 Bridget Jones's Diary	 Love Actually
Boris	★★★☆☆	★★★★☆	★★★★★			★☆☆☆☆		?
Dave		★★★★★	★★★★★	★★★★★				★☆☆☆☆
Will		★★☆☆☆			★★★★★	★★★★★	★★★☆☆	★★★★☆
George	★★★★☆	★★★★★	★★★★☆	★★★★☆				★★☆☆☆

- 你觉得它的评分会是多少?

3.2 基于近邻算法的协同过滤

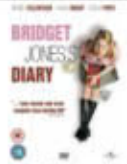
评分预测

	 Die Hard	 Mission: Impossible	 GoldenEye	 Casino Royale	 Titanic	 Notting Hill	 Bridget Jones's Diary	 Love Actually
Boris	★★★☆☆	★★★★☆	★★★★★			★☆☆☆☆		?
Dave		★★★★★	★★★★★	★★★★★				★☆☆☆☆
Will		★★☆☆☆			★★★★★	★★★★★	★★★☆☆	★★★★☆
George	★★★★☆	★★★★★	★★★★☆	★★★★☆				★★☆☆☆

- 平均Dave和George在Love Actually上的评分
  - 预测 = (1+2)/2 = 1.5

3.2 基于近邻算法的协同过滤

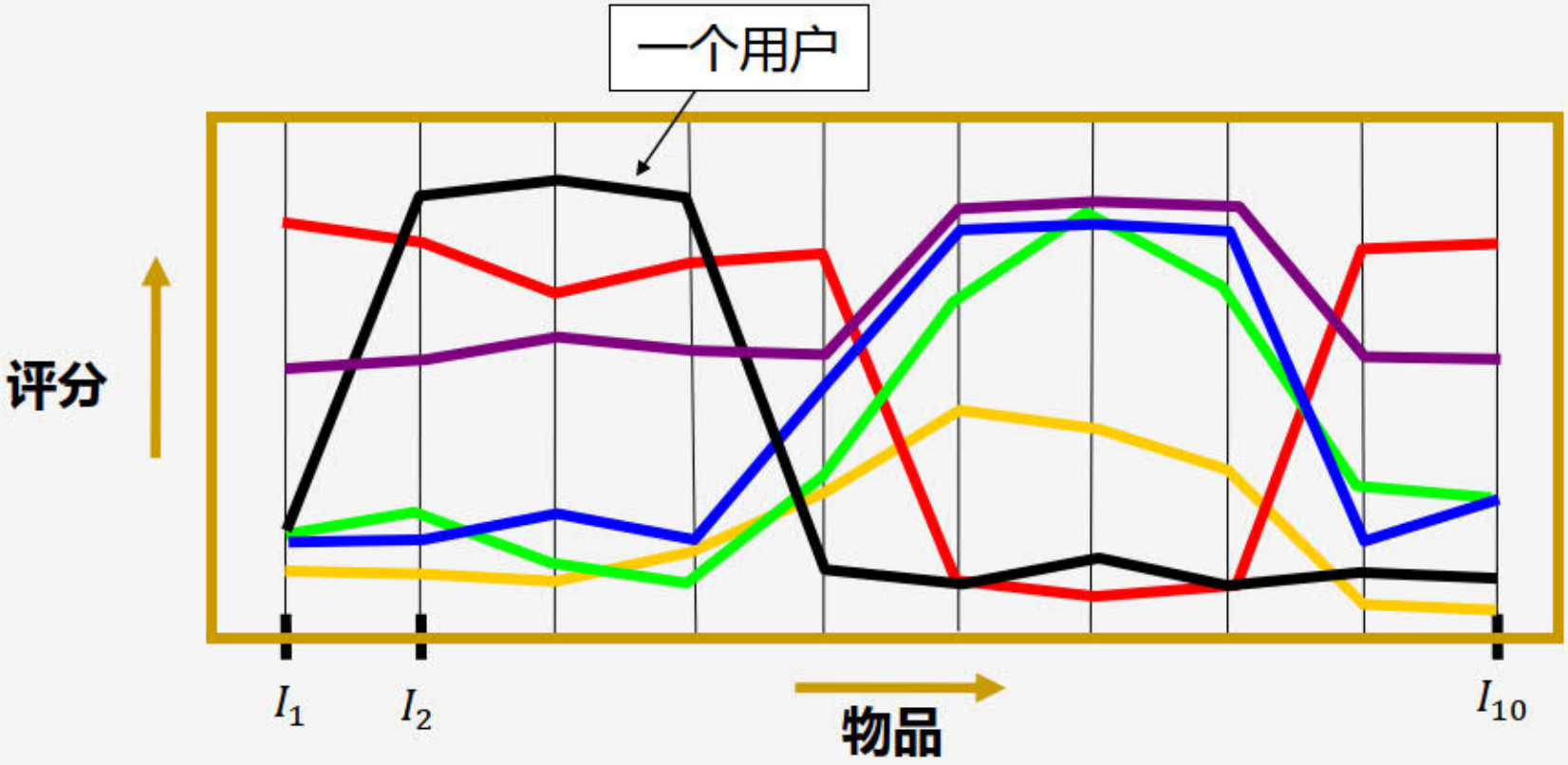
用于推荐的协同过滤

	 Die Hard	 Mission: Impossible	 GoldenEye	 Casino Royale	 Titanic	 Notting Hill	 Bridget Jones's Diary	 Love Actually
Boris	★★★☆☆	★★★★☆	★★★★★			★☆☆☆☆		?
Dave		★★★★★	★★★★★	★★★★★				★☆☆☆☆
Will		★★☆☆☆			★★★★★	★★★★★	★★★☆☆	★★★★☆
George	★★★★☆	★★★★★	★★★★☆	★★★★☆				★★☆☆☆

- 基于用户的基本KNN算法
  - 为每个目标用户提供推荐
    - 寻找类似用户
    - 基于类似用户推荐新物品

3.2 基于近邻算法的协同过滤

用户的相似度



- 每个用户的画像可以根据其选择的物品评级直接构建为向量

用户的相似度

- 两个用户a和b之间的相似度量
  - 余弦（角）相似度

$$s_u^{cos}(u_a, u_b) = \frac{u_a^{\top} u_b}{\| u_a \| \| u_b \|} = \frac{\sum_m x_{a,m} x_{b,m}}{\sqrt{\sum_m x_{a,m}^2 \sum_m x_{b,m}^2}}$$

  - 皮尔森相关性

$$s_u^{corr}(u_a, u_b) = \frac{\sum_m (x_{a,m} - \bar{x}_a)(x_{b,m} - \bar{x}_b)}{\sqrt{\sum_m (x_{a,m} - \bar{x}_a)^2 \sum_m (x_{b,m} - \bar{x}_b)^2}}$$

基于用户的KNN评分预测

- 预测从目标用户t与物品m之间的评分

根据和目标用户的相似度定义的权重

根据相似用户的评分进行预测（针对该用户的平均评分进行校正）

$$\hat{x}_{t,m} = \overline{x}_t + \frac{\sum_{u_a \in N_u(u_t)} s_u(u_t, u_a)(x_{a,m} - \overline{x}_a)}{\sum_{u_a \in N_u(u_t)} s_u(u_t, u_a)}$$

对目标用户的平均评分的修正

$$\overline{x}_t = \frac{1}{|I(u_t)|} \sum_{m \in I(u_t)} x_{t,m}$$

标准化，使评分在合理的范围内

基于物品的KNN评分预测

- 对于目标用户t的每个未评分物品m
  - 寻找类似物品{a}
  - 基于类似物品的集合{a}预测物品m的评分

物品的相似度

- 两个物品a和b之间的相似度量
  - 余弦（角）相似度

$$s_i^{cos}(i_a, i_b) = \frac{i_a^{\top} i_b}{\| i_a \| \| i_b \|} = \frac{\sum_u x_{u,a} x_{u,b}}{\sqrt{\sum_u x_{u,a}^2 \sum_u x_{u,b}^2}}$$

  - 调整后的余弦相似度（$\bar{x}_u$ 为第u个用户对所有物品的平均评分）

$$s_i^{adcos}(i_a, i_b) = \frac{\sum_u (x_{u,a} - \bar{x}_u)(x_{u,b} - \bar{x}_u)}{\sqrt{\sum_u (x_{u,a} - \bar{x}_u)^2 \sum_u (x_{u,b} - \bar{x}_u)^2}}$$

  - 皮尔森相关性（$\bar{x}_a$、 $\bar{x}_b$分别为所有用户对物品a和物品b的平均评分）

$$s_i^{corr}(i_a, i_b) = \frac{\sum_u (x_{u,a} - \bar{x}_a)(x_{u,b} - \bar{x}_b)}{\sqrt{\sum_u (x_{u,a} - \bar{x}_a)^2 \sum_u (x_{u,b} - \bar{x}_b)^2}}$$

Badrul Sarwar, George Karypis, Joseph Konstan, and John Riedl. “Item-Based Collaborative Filtering Recommendation Algorithms” WWW 2010.

基于物品的KNN评分预测

- 获取目标用户t评分过，且和物品i_a最相似的k个最近邻物品

排序位置

$$N_i(u_t, i_a) = \{i_b | r_i(i_a, i_b) < K^*, x_{t,b} \neq 0\}$$

选择K^*，使$|N_i(u_t, i_a)| = k$

- 预测目标用户未评分的物品i_a的评分

$$\hat{x}_{t,a} = \frac{\sum_{i_b \in N_i(u_t, i_a)} s_i(i_a, i_b) x_{t,b}}{\sum_{i_b \in N_i(u_t, i_a)} s_i(i_a, i_b)}$$

由于使用目标用户本身数据进行预测，因此无需修正用户的平均评分

实证研究

- Movielens数据集来自
  - https://grouplens.org/datasets/movielens/
  - 用户访问Movielens
    - 评分和收到电影的推荐
  - 数据集（ML-100k）
  - 100k个从1到5的评分
  - 943位用户，1682部电影（至少有一位用户评分）
  - 稀疏程度

$$1-\frac{\#\text{non-zero entries}}{\text{total entries}}=1-\frac{10^5}{943\times 1682}=93.69\%$$

Badrul Sarwar, George Karypis, Joseph Konstan, and John Riedl. “Item-Based Collaborative Filtering Recommendation Algorithms” WWW 2010.

实验设置

- 把数据拆分成训练集($x\%$)和测试集($(100-x)\%$)
  - 可重复T次，结果取平均值
- 评估指标
  - 平均绝对误差（MAE）

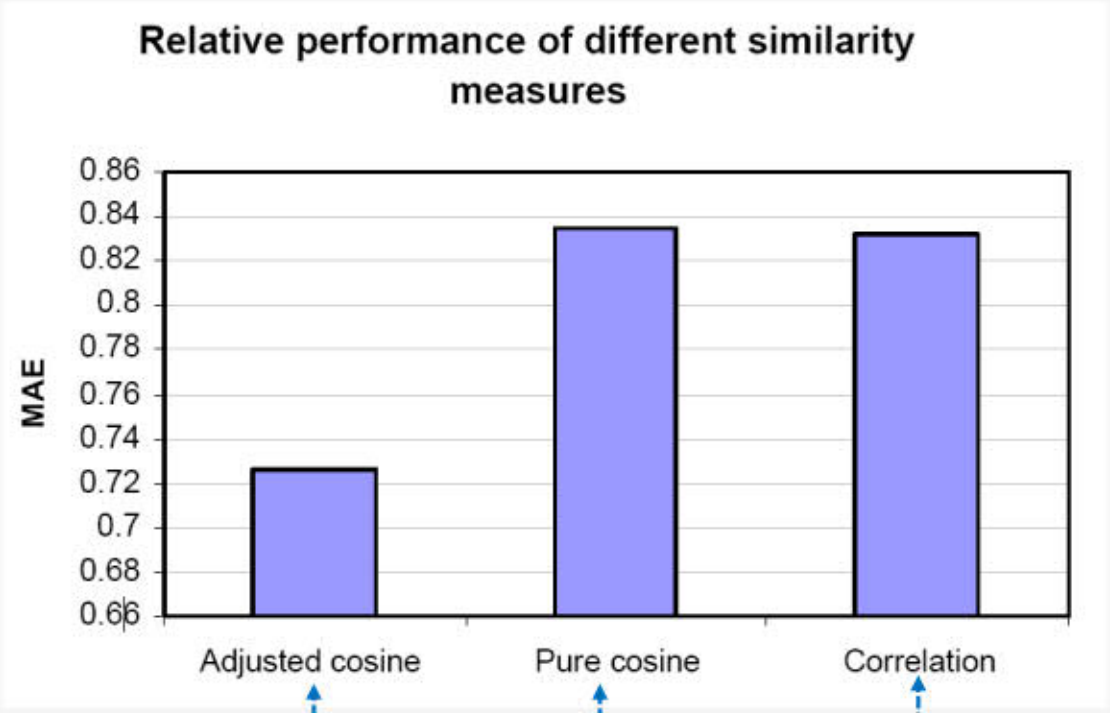
$$\text{MAE} = \frac{1}{|D_{\text{test}}|} \sum_{(u,i,r)\in D_{\text{test}}} |r - \hat{r}_{u,i}|$$

  - 均方根误差（RMSE）

$$\text{RMSE} = \sqrt{\frac{1}{|D_{\text{test}}|} \sum_{(u,i,r)\in D_{\text{test}}} \left(r - \hat{r}_{u,i}\right)^2}$$

3.2 基于近邻算法的协同过滤

相似度量的表现

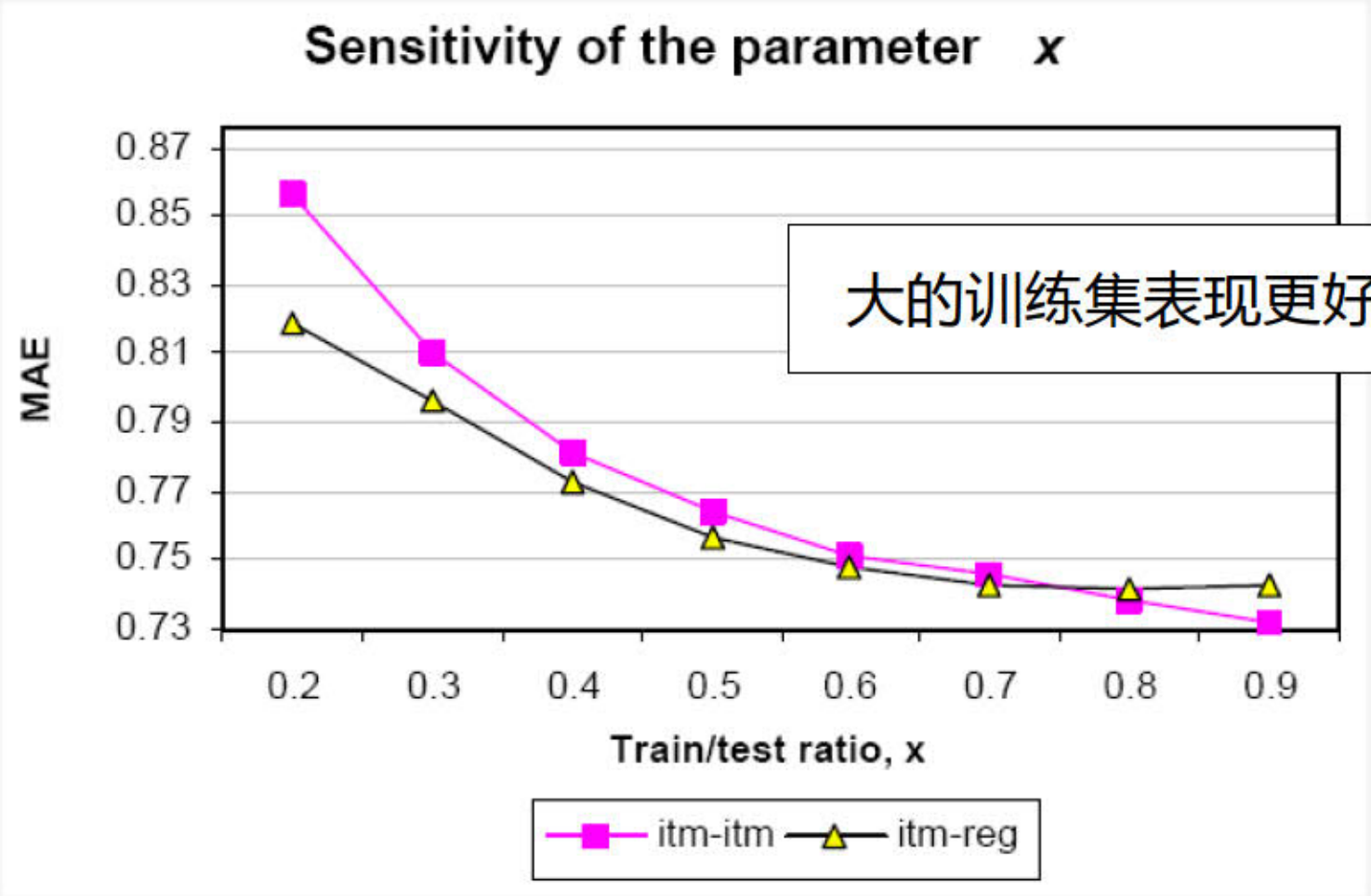


$$s_i^{adcos}(i_a, i_b) = \frac{\sum_u (x_{u,a} - \overline{x}_u)(x_{u,b} - \overline{x}_u)}{\sqrt{\sum_u (x_{u,a} - \overline{x}_u)^2 \sum_u (x_{u,b} - \overline{x}_u)^2}}$$

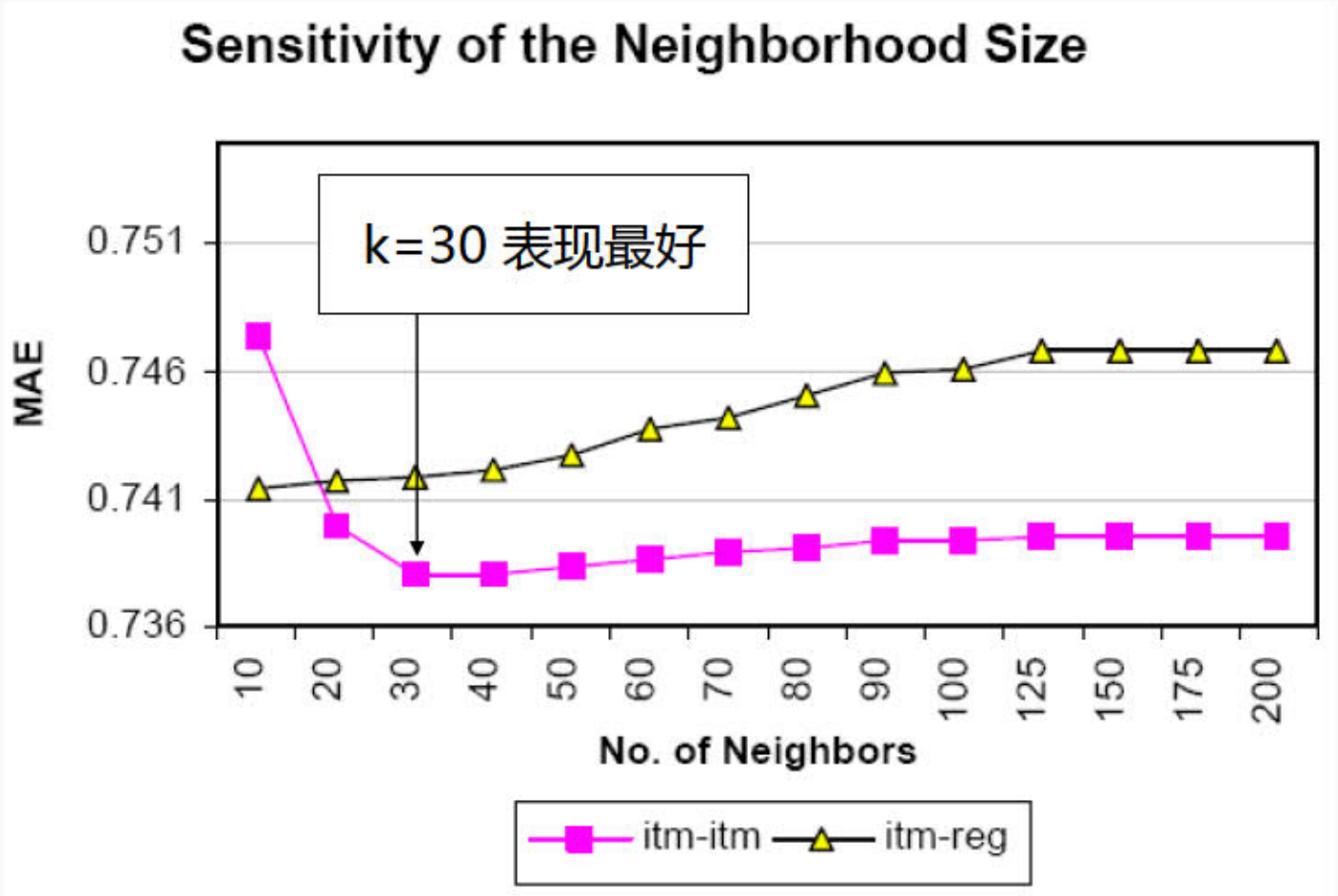
$$s_i^{cos}(i_a, i_b) = \frac{\sum_u x_{u,a} x_{u,b}}{\sqrt{\sum_u x_{u,a}^2 \sum_u x_{u,b}^2}}$$

$$s_i^{corr}(i_a, i_b) = \frac{\sum_u (x_{u,a} - \overline{x}_a)(x_{u,b} - \overline{x}_b)}{\sqrt{\sum_u (x_{u,a} - \overline{x}_a)^2 \sum_u (x_{u,b} - \overline{x}_b)^2}}$$

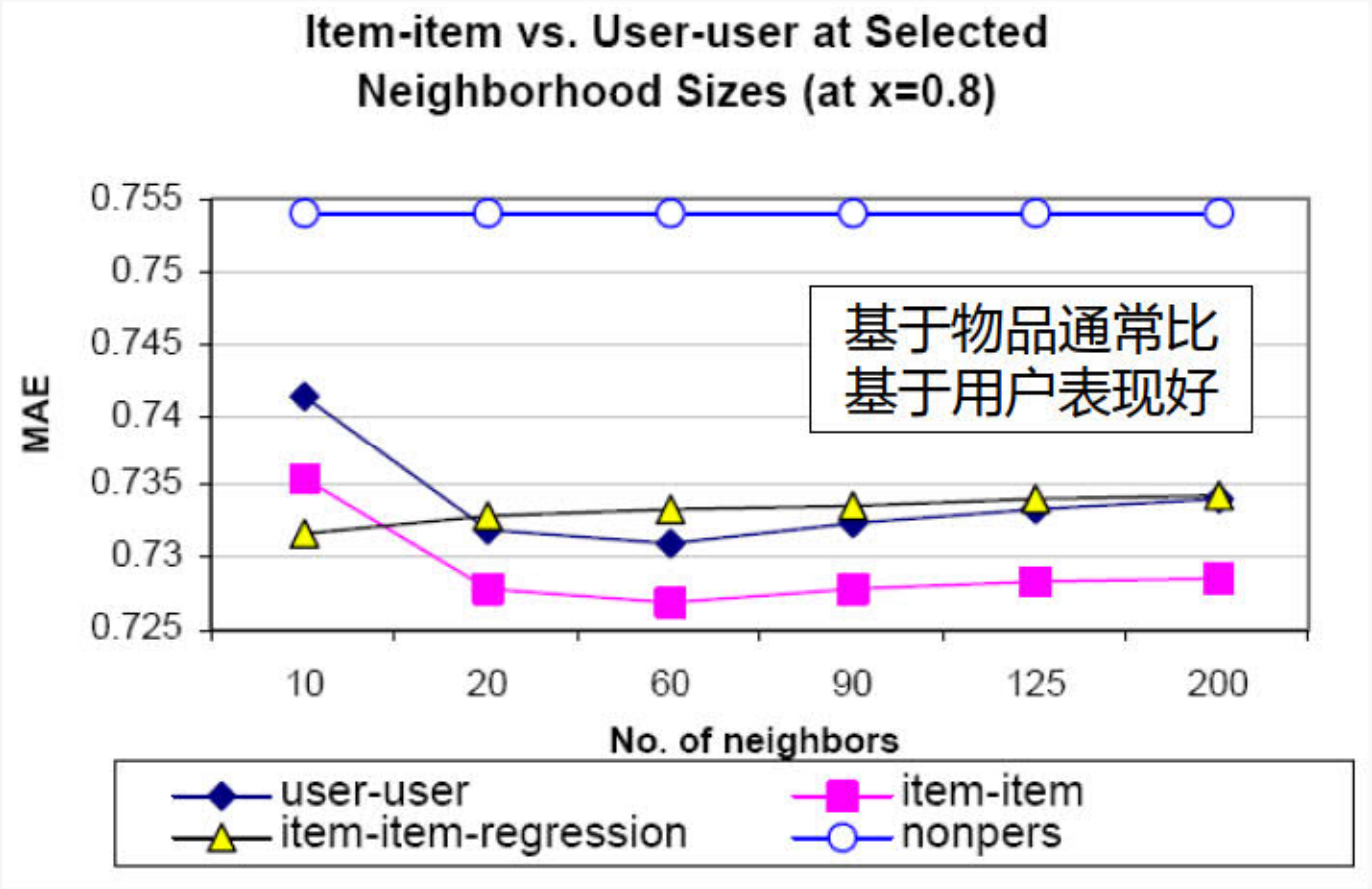
训练/测试比例的敏感度


Sensitivity of the parameter x
MAE
0.87
0.85
0.83
0.81
0.79
0.77
0.75
0.73
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
Train/test ratio, x
itm-itm
itm-reg
大的训练集表现更好

邻居大小k的敏感度


Sensitivity of the Neighborhood Size
k=30 表现最好
MAE
0.751
0.746
0.741
0.736
10
20
30
40
50
60
70
80
90
100
125
150
175
200
No. of Neighbors
itm-itm
itm-reg

基于物品 vs. 基于用户



- 物品-物品相似度通常更加稳定和客观

基于KNN的方法总结

- 直接，有高度可解释性
- 没有参数学习
  - 只有一个超参数k可以调整
  - 无法通过学习得到改善
- 效率可能是一个严重的问题
  - 当用户/项目编号很大时
  - 当有大量的用户-物品评分时
- 我们可能需要一个参数化和可学习的模型

3.3 矩阵分解

矩阵分解（Matrix Factorization）

R

用户

物品

1		3			5			5		4	
		5	4	?		4			2	1	3
2	4		1	2		3		4	3	5	
	2	4		5			4			2	
		4	3	4	2					2	5
1		3		3			2			4	

$$\hat{r}_{u,i} = p_u^{\top} q_i$$

$\approx$

用户 u

.1	-.4	.2
-.5	.6	.5
-.2	.3	.5
1.1	2.1	.3
-.7	2.1	-2
-1	.7	.3

P

$\cdot$

i 物品

1.1	-.2	.3	.5	-2	-.5	.8	-.4	.3	1.4	2.4	-.9
-.8	.7	.5	1.4	.3	-1	1.4	2.9	-.7	1.2	-.1	1.3
2.1	-.4	.6	1.7	2.4	.9	-.3	.4	.8	.7	-.6	.1

Q

基本的矩阵分解模型

- 用户u在物品i上的预测

$$\hat{r}_{u,i} = p_u^\top q_i$$

- 损失函数

$$\mathcal{L}(u, i, r_{u,i}) = \frac{1}{2}\left(r_{u,i} - p_u^\top q_i\right)^2$$

- 训练目标

$$\min_{P,Q} \sum_{r_{u,i} \in R} \frac{1}{2}(r_{u,i} - p_u^\top q_i)^2 + \frac{\lambda}{2}(\| p_u \|^2 + \| q_i \|^2)$$

- 梯度

$$\frac{\partial \mathcal{L}(u, i, r_{u,i})}{\partial p_u} = (p_u^\top q_i - r_{u,i}) q_i + \lambda p_u$$

$$\frac{\partial \mathcal{L}(u, i, r_{u,i})}{\partial q_i} = (p_u^\top q_i - r_{u,i}) p_u + \lambda q_i$$

带偏置的矩阵分解模型

- 用户u在物品i上的预测

$$\hat{r}_{u,i} = \mu + b_u + b_i + p_u^{\top} q_i$$

全局偏置　用户偏置　物品偏置　用户-物品交互

- 训练目标

$$\min_{P,Q} \sum_{r_{u,i} \in D} \frac{1}{2}(r_{u,i} - (\mu + b_u + b_i + p_u^{\top} q_i))^2 + \frac{\lambda}{2}(\| p_u \|^2 + \| q_i \|^2 + b_u^2 + b_i^2)$$

- 梯度更新

$$\delta = r_{u,i} - (\mu + b_u + b_i + p_u^{\top} q_i)$$
$$\mu \leftarrow \mu + \eta\delta$$
$$b_u \leftarrow (1 - \eta\lambda) b_u + \eta\delta$$
$$b_i \leftarrow (1 - \eta\lambda) b_i + \eta\delta$$
$$p_u \leftarrow (1 - \eta\lambda) p_u + \eta\delta q_i$$
$$q_i \leftarrow (1 - \eta\lambda) q_i + \eta\delta p_u$$

动态变化的多个原因

- 物品方面的影响
  - 物品受欢迎程度不断变化
  - 季节性原因影响物品的受欢迎程度
- 用户方面的影响
  - 用户改变了他们的品味
  - 暂时的，短期偏置
  - 浮动的评分等级
  - 在一个家庭里实际评分人的改变

解决动态变化问题

- 因子模型方便地允许分别处理不同原因
- 我们观察到的变化：
  - 个人用户的评分等级$b_u(t)$
  - 个人物品的受欢迎程度$b_i(t)$
  - 用户偏好$p_u(t)$

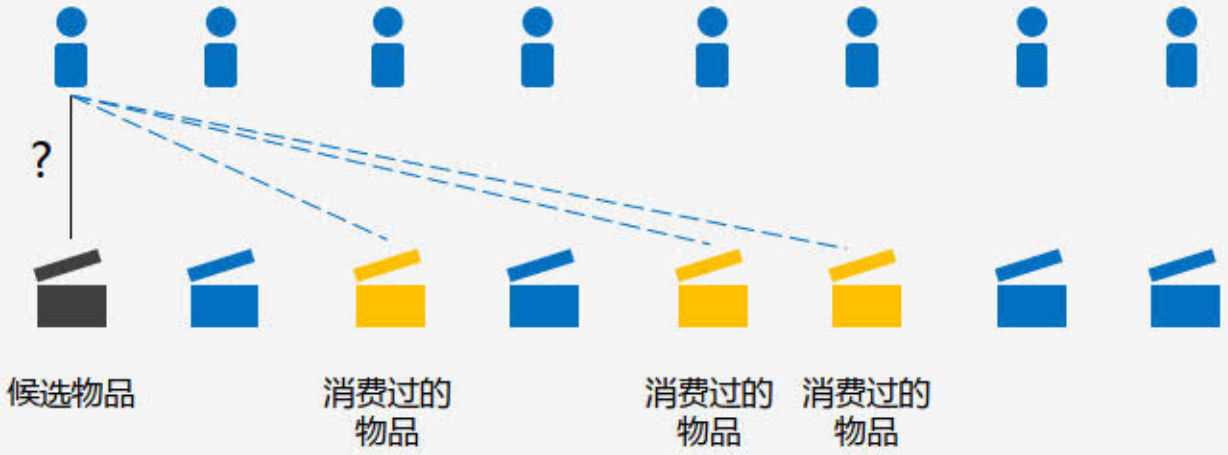
$$r_{u,i}(t) = \mu + b_u(t) + b_i(t) + p_u(t)^{\top} q_i$$

- 设计指南
  - 物品显示较慢的时间变化
  - 用户表现出频繁和突然的变化
  - 因子$p_u(t)$的建模成本很高
  - 通过大量参数化函数来获得灵活性

3.3 矩阵分解

基于邻居（相似性）的矩阵分解

- 假设：用户以前消费过的物品反映了她的品味
- 从这些“类似的”物品中获取未知评分（物品-物品变体）



3.3 矩阵分解

基于邻居的矩阵分解：SVD++

$$\hat{r}_{u,i} = b_{u,i} + q_i^{\top}(p_u + |N(u)|^{-\frac{1}{2}} \sum_{j \in N(u)} y_j)$$

- 需要估计用户u的评分物品i
- 偏置
- 物品向量
- 用户向量
- 用户之前评分的物品的集合
- 为了表示邻居相似度的物品向量

- 每个物品有两个潜在向量
  - 本身的物品向量q_i
  - 用于估计候选物品和目标用户之间的相似度的向量y_i

Koren, Yehuda. "Factorization meets the neighborhood: a multifaceted collaborative filtering model." KDD, 2008.

k-近邻算法总结

- 协同过滤是推荐系统中的重要机器学习技术，主要探索学习数据点之间的相似性，进而对目标标签做出预测
- KNN算法是最简单的非参数化模型
- 矩阵分解是最基础的双线性模型。参数向量的内积操作是双线性模型中最常使用的操作，这时对一边参数向量求梯度，得到另一边参数向量的取值
- 因子分解机是一种通过向量内积来直接学习两两特征交互对标签预测的影响的通用模型