

# 第三章、推荐系统的冷启动问题

讲师：武永亮



# 教学目标

---

- 了解冷启动问题
- 理解如何利用用户注册信息
- 理解如何选择合适的物品启动用户的兴趣
- 理解利用物品的内容信息
- 理解发挥专家的作用

# 目录

---

1 冷启动问题简介

2 利用用户注册信息

3 选择合适的物品启动用户的兴趣

4 利用物品的内容信息

5 发挥专家的作用

# 冷启动问题简介

---

- 推荐系统需要根据用户的历史行为和兴趣预测用户未来的行为和兴趣，因此大量的用户行为数据就成为推荐系统的重要组成部分和先决条件。对于很多像百度、当当这样的网站来说，这或许不是个问题，因为它们目前已经积累了大量的用户数据。但是对于很多做纯粹推荐系统的网站（比如Jinni和Pandora），或者很多在开始阶段就希望有个性化推荐应用的网站来说，如何在没有大量用户数据的情况下设计个性化推荐系统并且让用户对推荐结果满意从而愿意使用推荐系统，就是冷启动的问题。

# 冷启动问题简介

---

- 冷启动问题（ cold start ）主要分3类。
  - **用户冷启动** 用户冷启动主要解决如何给新用户做个性化推荐的问题。
  - **物品冷启动** 物品冷启动主要解决如何将新的物品推荐给可能对它感兴趣的用户这一问题。
  - **系统冷启动** 系统冷启动主要解决如何在一个新开发的网站上（还没有用户，也没有用户行为，只有一些物品的信息）设计个性化推荐系统，从而在网站刚发布时就让用户体验到个性化推荐服务这一问题。

# 冷启动问题简介

- 对于这3种不同的冷启动问题，有不同的解决方案
  - 提供非个性化的推荐 非个性化推荐的最简单例子就是热门排行榜，我们可以给用户推荐热门排行榜，然后等到用户数据收集到一定的时候，再切换为个性化推荐。
  - 利用用户注册时提供的年龄、性别等数据做粗粒度的个性化。
  - 利用用户的社交网络账号登录（需要用户授权），导入用户在社交网站上的好友信息，然后给用户推荐其好友喜欢的物品。
  - 要求用户在登录时对一些物品进行反馈，收集用户对这些物品的兴趣信息，然后给用户推荐那些和这些物品相似的物品。
  - 对于新加入的物品，可以利用内容信息，将它们推荐给喜欢过和它们相似的物品用户。
  - 在系统冷启动时，可以引入专家的知识，通过一定的高效方式迅速建立起物品的相关度表。

# 目录

---

1

冷启动问题简介

2

利用用户注册信息

3

选择合适的物品启动用户的兴趣

4

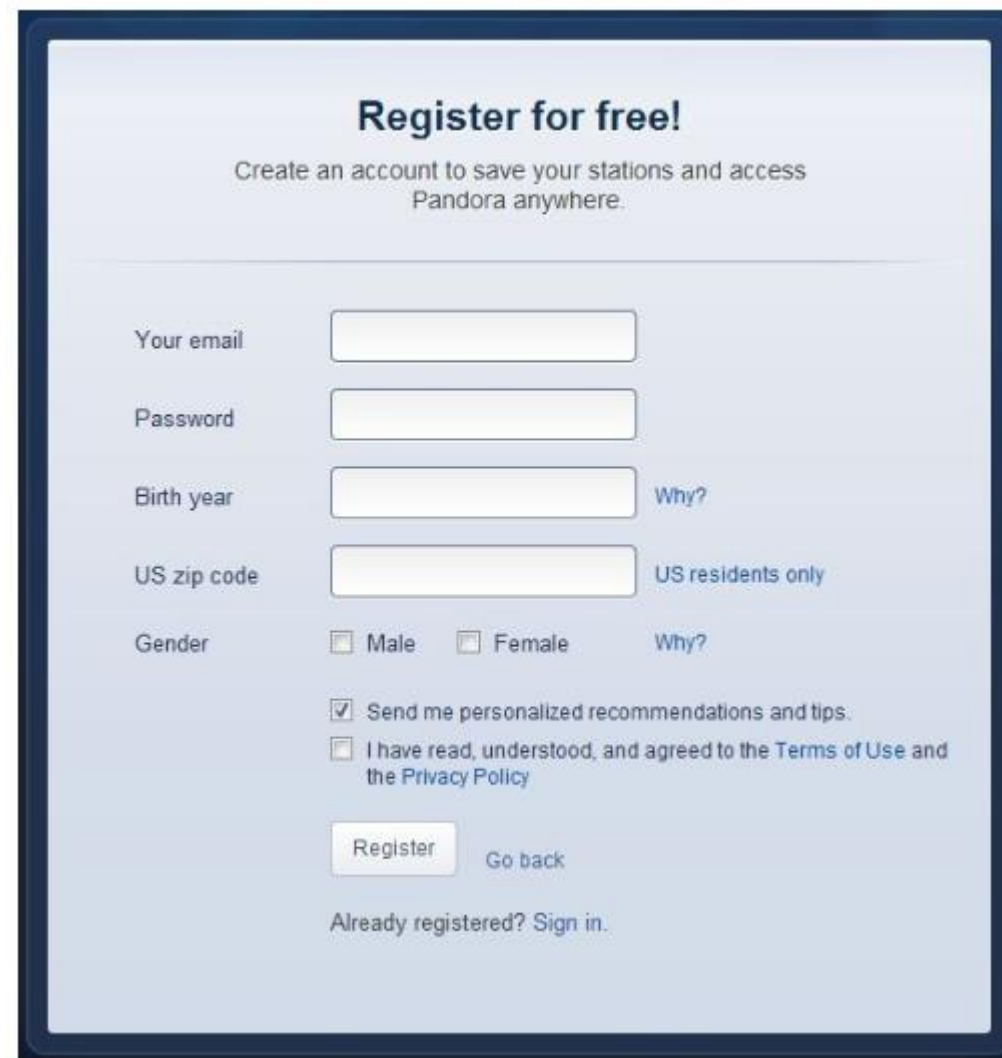
利用物品的内容信息

5

发挥专家的作用

# 利用用户注册信息

- 用户的注册信息分3种。
  - 人口统计学信息
  - 用户兴趣的描述
  - 从其他网站导入的用户站外行为数据



The image shows a registration form for Pandora. The title is "Register for free!" with a subtitle "Create an account to save your stations and access Pandora anywhere." The form includes input fields for "Your email", "Password", "Birth year", and "US zip code". There are also checkboxes for "Gender" (Male and Female) and a checkbox for "Send me personalized recommendations and tips." A link "Why?" is next to the "Birth year" and "Gender" fields. A link "US residents only" is next to the "US zip code" field. At the bottom, there is a checkbox for "I have read, understood, and agreed to the Terms of Use and the Privacy Policy" and a "Register" button. A "Go back" link is also present. At the very bottom, it says "Already registered? Sign in."

**Register for free!**  
Create an account to save your stations and access Pandora anywhere.

Your email

Password

Birth year  [Why?](#)

US zip code  [US residents only](#)

Gender ☐ Male ☐ Female [Why?](#)

☒ Send me personalized recommendations and tips.

☐ I have read, understood, and agreed to the [Terms of Use](#) and the [Privacy Policy](#)

[Go back](#)

Already registered? [Sign in.](#)



# 利用用户注册信息

- 通过用户注册时填写的人口统计学信息给用户提供粗粒度的个性化推荐
- 人口统计学特征包括年龄、性别、工作、学历、居住地、国籍、民族等，这些特征对预测用户的兴趣有很重要的作用。

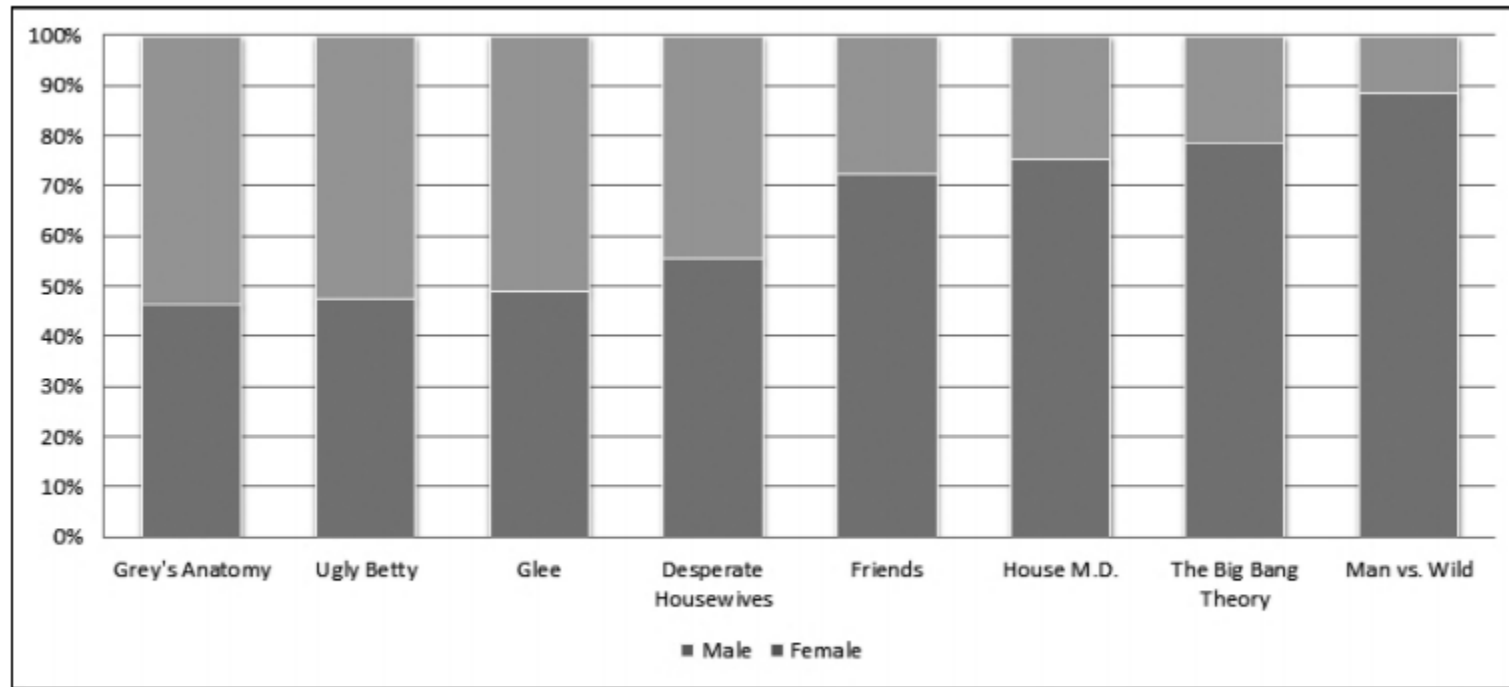


图3-2 IMDB中不同美剧的评分用户的性别分布

# 利用用户注册信息

- 基于注册信息的个性化推荐流程基本如下：
  - 获取用户的注册信息；
  - 根据用户的注册信息对用户分类；
  - 给用户推荐他所属分类中用户喜欢的物品。



# 利用用户注册信息

- 实际应用中也可以考虑组合特征。不过在使用组合时需要注意用户不一定具有所有的特征，因为一般的注册系统并不要求用户填写所有注册项。
- 基于用户注册信息的推荐算法其核心问题是计算每种特征的用户喜欢的物品。也就是说，对于每种特征 $f$ ，计算具有这种特征的用户对各个物品的喜好程度 $p(f, i)$ 。

$p(f, i)$  可以简单地定义为物品 $i$ 在具有 $f$ 的特征的用户中的热门程度：

$$p(f, i) = |N(i) \cap U(f)|$$

其中  $N(i)$  是喜欢物品 $i$ 的用户集合， $U(f)$  是具有特征 $f$ 的用户集合。

# 利用用户注册信息

- 上面这种定义可以比较准确地预测具有某种特征的用户是否喜欢某个物品。但是，在这种定义下，往往热门的物品会在各种特征的用户中都具有比较高的权重。因此，我们可以将  $p(f, i)$  定义为喜欢物品  $i$  的用户中具有特征  $f$  的比例：

$$p(f, i) = \frac{|N(i) \cap U(f)|}{|N(i)| + \alpha}$$

- 这里分母中使用参数  $\alpha$  的目的是解决数据稀疏问题。

# 利用用户注册信息

---

- 有两个推荐系统数据集包含了人口统计学信息
  - BookCrossing数据集
  - Lastfm数据集
- BookCrossing数据集包含用户对图书的行为信息，包含：
  - BX-Users.csv，包含用户的ID、位置和年龄。
  - BX-Books.csv，包含图书的ISBN、标题、作者、发表年代、出版社和缩略。
  - BX-Book-Ratings.csv，包含用户对图书的评分信息。

# 利用用户注册信息

- 根据BookCrossing数据集研究一下年龄对用户喜欢图书的影响。我们研究两类用户，一类是小于25岁的，一类是大于50岁的
  - 首先，我们利用  $p(f, i)$  统计了这两部分用户最经常看的书

表3-1 年轻用户和老年用户经常看的图书的列表

## 年轻人（小于25岁）

Wild Animus, Rich Shapero, 2004, Too Far

The Lovely Bones: A Novel, Alice Sebold, 2002, Little, Brown

Harry Potter and the Sorcerer's Stone (Harry Potter (Paperback)), J. K. Rowling, 1999, Arthur A. Levine Books

The Catcher in the Rye, J.D. Salinger, 1991, Little, Brown

The Da Vinci Code, Dan Brown, 2003, Doubleday

## 老年人（大于50岁）

Wild Animus, Rich Shapero, 2004, Too Far

The Da Vinci Code, Dan Brown, 2003, Doubleday

The Lovely Bones: A Novel, Alice Sebold, 2002, Little, Brown

A Painted House, John Grisham, 2001, Dell Publishing Company

Angels & Demons, Dan Brown, 2001, Pocket Star

# 利用用户注册信息

- 根据BookCrossing数据集研究一下年龄对用户喜欢图书的影响。我们研究两类用户，一类是小于25岁的，一类是大于50岁的
  - 计算了年轻用户比例最高的5本书和老年用户比例最高的5本书

$$p(f,i) = \frac{|N(i) \cap U(f)|}{|N(i)| + \alpha}$$

表3-2 年轻用户比例最高的5本书和老年人比例最高的5本书

## 年轻人（小于25岁）

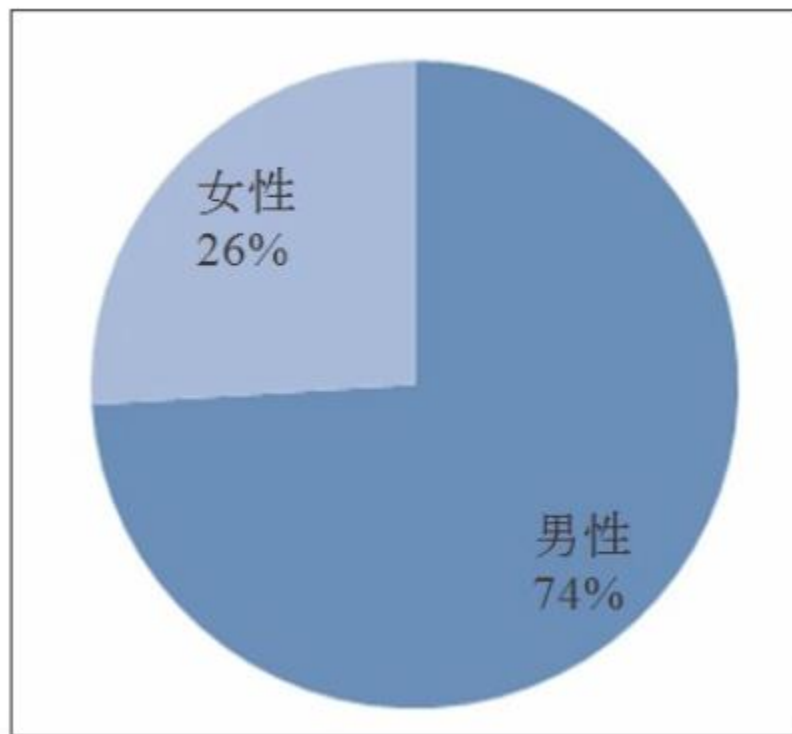
The Perks of Being a Wallflower, Stephen Chbosky, 1999, MTV  
The Catcher in the Rye, J.D. Salinger, 1991, Little, Brown  
And Then There Were None : A Novel, Agatha Christie, 2001, St. Martin's Paperbacks  
Chicken Soup for the Teenage Soul (Chicken Soup for the Soul), Jack Canfield, 1997, Health Communications  
The Giver (21st Century Reference), LOIS LOWRY, 1994, Laure Leaf

## 老年人（大于50岁）

The No. 1 Ladies' Detective Agency (Today Show Book Club #8), Alexander McCall Smith, 2003, Anchor  
A Painted House, John Grisham, 2001, Dell Publishing Company  
The Da Vinci Code, Dan Brown, 2003, Doubleday  
Deception Point, Dan Brown, 2002, Pocket  
A Thief of Time (Joe Leaphorn/Jim Chee Novels), Tony Hillerman, 1990, HarperTorch

# 利用用户注册信息

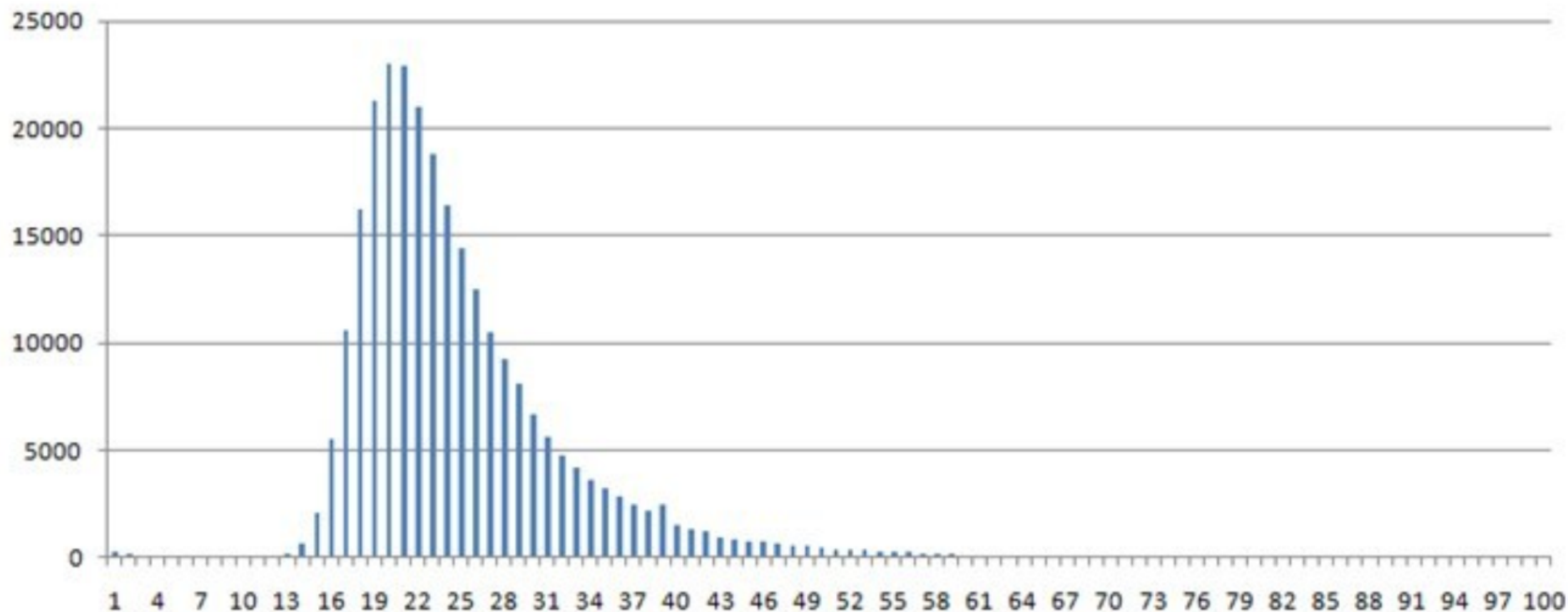
- Lastfm数据集包含了更多的用户人口统计学信息，包括用户的性别、年龄和国籍。





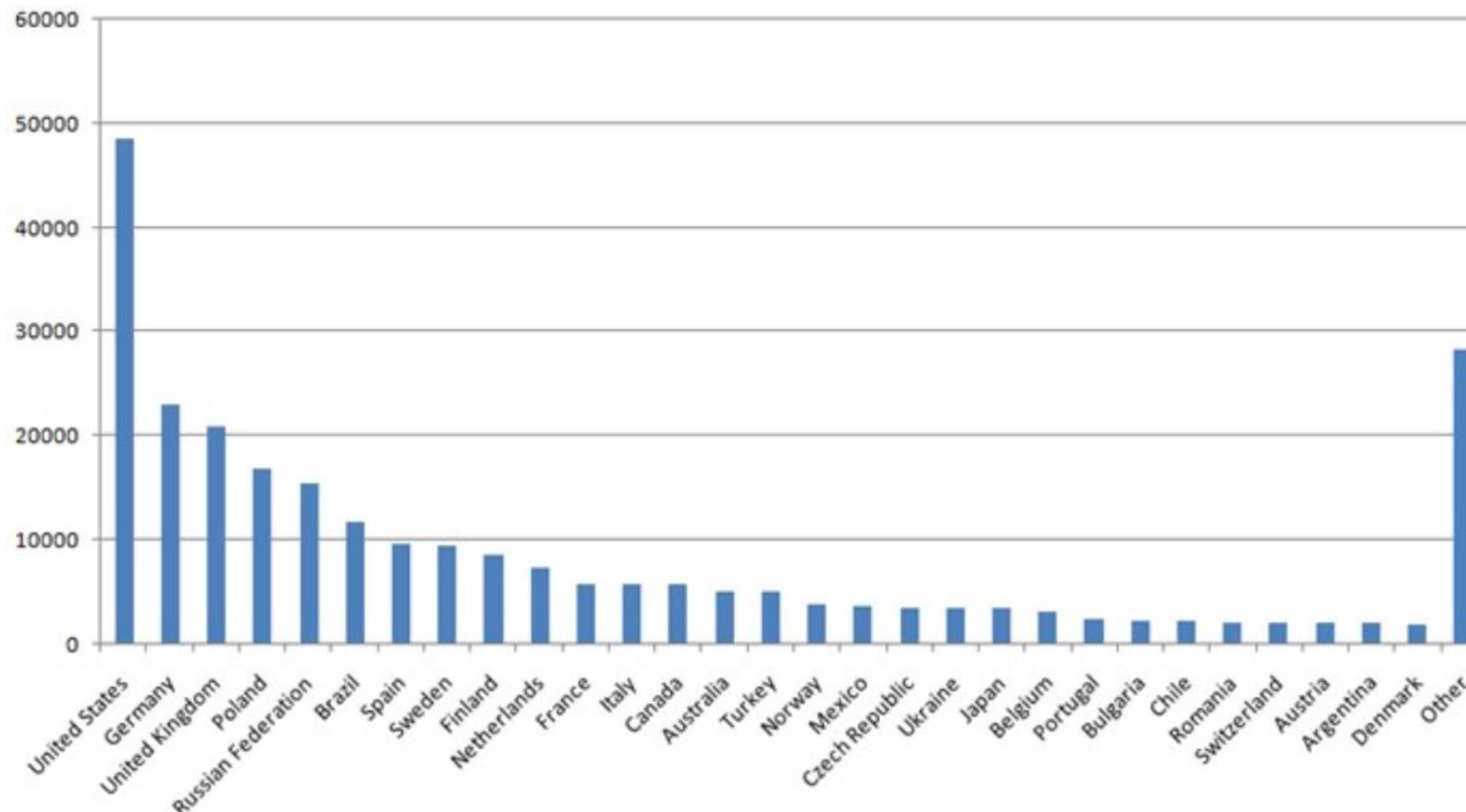
# 利用用户注册信息

- Lastfm数据集包含了更多的用户人口统计学信息，包括用户的性别、年龄和国籍。



# 利用用户注册信息

- Lastfm数据集包含了更多的用户人口统计学信息，包括用户的性别、年龄和国籍。



# 利用用户注册信息

---

我们准备用该数据集对比一下使用不同的人口统计学特征预测用户行为的精度。这里，我们将数据集划分成10份，9份作为训练集，1份作为测试集。然后，我们在训练集上利用  $p(f,i) = |N(i) \cap U(f)|$  计算每一类用户对物品的兴趣程度  $p(f,i)$ 。然后在测试集中给每一类用户推荐  $p(f,i)$  最高的10个物品，并通过准确率和召回率计算预测准确度。同时，我们也会计算推荐的覆盖率来评测推荐结果。

# 利用用户注册信息

---

- 我们按照不同的粒度给用户分类，对比了4种不同的算法。
  - MostPopular 给用户推荐最热门的歌手。
  - GenderMostPopular 给用户推荐对于和他同性别的用户最热门的歌手，这里我们将用户分成男女两类。
  - AgeMostPopular 给用户推荐对于和他同一个年龄段的用户最热门的歌手，这里我们将10岁作为一个年龄段，将用户按照不同的年龄段分类。
  - CountryMostPopular 给用户推荐对于和他同一个国家的用户最热门的歌手。
  - DemographicMostPopular 给用户推荐对于和他同性别、年龄段、国家的用户最热门的歌手。

# 利用用户注册信息

- 中MostPopular粒度最粗，而DemographicMostPopular算法的粒度最细。一般说来，粒度越细，精度和覆盖率也会越高。
- DemographicMostPopular > CountryMostPopular > AgeMostPopular > GenderMostPopular > MostPopular

表3-3 4种不同粒度算法的召回率、准确率和覆盖率

方 法	召 回 率	准 确 率	覆 盖 率
MostPopular	4.81%	2.36%	0.018%
GenderMostPopular	4.95%	2.43%	0.027%
AgeMostPopular	5.04%	2.47%	0.062%
CountryMostPopular	5.58%	2.73%	0.80%
DemographicMostPopular	6.00%	2.94%	3.85%

# 目录

---

1

冷启动问题简介

2

利用用户注册信息

3

选择合适的物品启动用户的兴趣

4

利用物品的内容信息

5

发挥专家的作用

# 选择合适的物品启动用户的兴趣

- 解决用户冷启动问题的另一个方法是在新用户第一次访问推荐系统时，不立即给用户展示推荐结果，而是给用户提供一些物品，让用户反馈他们对这些物品的兴趣，然后根据用户反馈给提供个性化推荐。



**Recommendations** | Wish List | Favorites | Ratings | Pulse | People

Rate at least 10 movies to jumpstart your recommendations! Your recommendations update with more titles once each day you sign into Jinni.



# 选择合适的物品启动用户的兴趣

- 解决用户冷启动问题的另一个方法是在新用户第一次访问推荐系统时，不立即给用户展示推荐结果，而是给用户提供一些物品，让用户反馈他们对这些物品的兴趣，然后根据用户反馈给提供个性化推荐。

## Rate Movies!

To jumpstart your **recommendations** and **Movie Personality** profile, rate movies for one of the 12 types of movie watchers below. You'll probably find a few types that fit.





# 选择合适的物品启动用户的兴趣

- 解决用户冷启动问题的另一个方法是在新用户第一次访问推荐系统时，不立即给用户展示推荐结果，而是给用户提供一些物品，让用户反馈他们对这些物品的兴趣，然后根据用户反馈给提供个性化推荐。



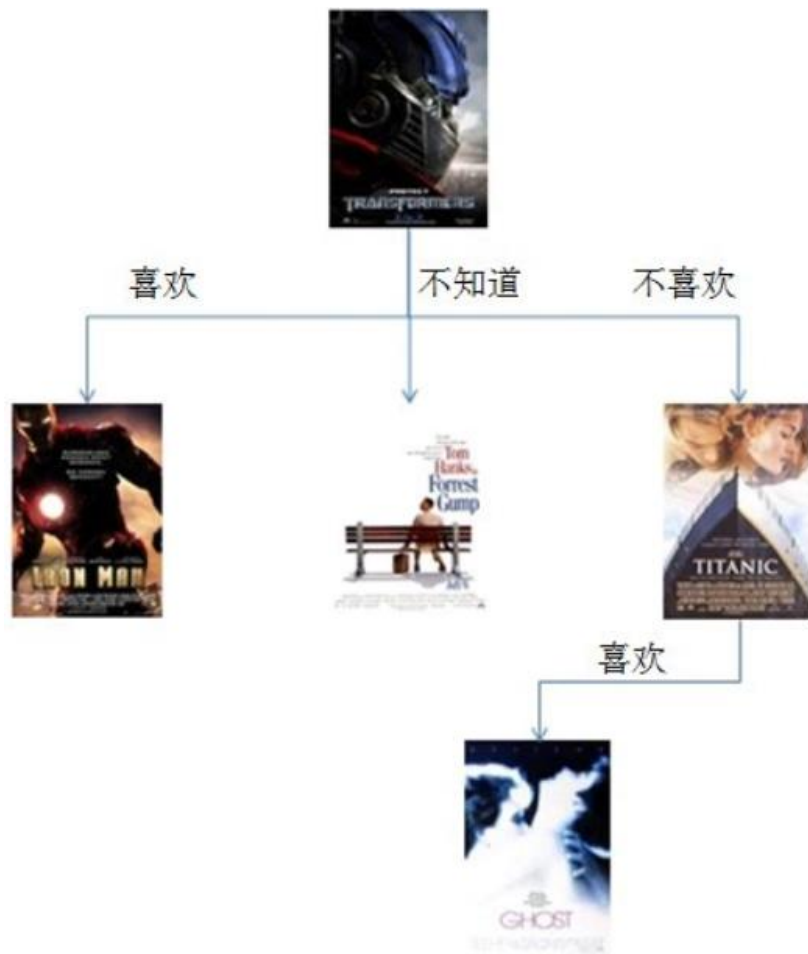
# 选择合适的物品启动用户的兴趣

---

- 能够用来启动用户兴趣的物品需要具有以下特点：
  - 比较热门 如果要用用户对一个物品进行反馈，前提是用户知道这个物品是什么东西。
  - 具有代表性和区分性 启动用户兴趣的物品不能是大众化或老少咸宜的，因为这样的物品对用户的兴趣没有区分性。
  - 启动物品集合需要有多多样性 在冷启动时，我们不知道用户的兴趣，而用户兴趣的可能性非常多，为了匹配多样的兴趣，我们需要提供具有很高覆盖率的启动物品集合，这些物品能覆盖几乎所有主流的用户兴趣。

# 选择合适的物品启动用户的兴趣

- 如何设计一个选择启动物品集合的系统呢？
  - 给定一群用户用这群用户对物品评分的方差度量这群用户兴趣的一致程度。



# 目录

---

1

冷启动问题简介

2

利用用户注册信息

3

选择合适的物品启动用户的兴趣

4

利用物品的内容信息

5

发挥专家的作用

# 利用物品的内容信息

- 物品冷启动需要解决的问题是如何将新加入的物品推荐给对它感兴趣的<sup>用户</sup>。物品冷启动在新闻网站等时效性很强的网站中非常重要，因为那些网站中时时刻刻都有新加入的物品，而且每个物品必须能够在第一时间展现给用户，否则经过一段时间后，物品的价值就大大降低了。
- UserCF和ItemCF算法
  - UserCF算法对物品冷启动问题并不非常敏感。
  - ItemCF算法的原理是给用户推荐和他之前喜欢的物品相似的物品。冷启动问题更严重。

# 利用物品的内容信息

- 物品的内容信息多种多样，不同类型的物品有不同的内容信息。

表3-4 常见物品的内容信息

图书	标题、作者、出版社、出版年代、丛书名、目录、正文
论文	标题、作者、作者单位、关键字、分类、摘要、正文
电影	标题、导演、演员、编剧、类别、剧情简介、发行公司
新闻	标题、正文、来源、作者
微博	作者、内容、评论

# 利用物品的内容信息

- 物品的内容可以通过向量空间模型表示，该模型会将物品表示成一个关键词向量。如果物品的内容是一些诸如导演、演员等实体的话，可以直接将这些实体作为关键词。但如果内容是文本的形式，则需要引入一些理解自然语言的技术抽取关键词。



图3-11 关键词向量的生成过程

# 利用物品的内容信息

- 对物品 $d$ ，它的内容表示成一个关键词向量如下：

$$d_i = \{(e_1, w_1), (e_2, w_2), \dots\}$$

- 如果物品是文本，我们可以用信息检索领域著名的TF-IDF公式计算词的权重：

$$w_i = \frac{\text{TF}(e_i)}{\log \text{DF}(e_i)}$$

- 如果物品是电影，可以根据演员在剧中的重要程度赋予他们权重。



# 利用物品的内容信息

- 在具体计算物品之间的内容相似度时，最简单的方法当然是对两两物品都利用上面的余弦相似度公式计算相似度，如下代码简单实现了这种方法：

```
function CalculateSimilarity(D)
    for di in D:
        for dj in D:
            w[i][j] = CosineSimilarity(di, dj)
    return w
```

- 但这种算法的时间复杂度很高。假设有 $N$ 个物品，每个物品平均由 $m$ 个实体表示，那么这个算法的复杂度是  $O(N^2m)$

# 利用物品的内容信息

- 在实际应用中，可以首先通过建立关键词—物品的倒排表加速这一计算过程，关于这一方法已经在前面介绍UserCF和ItemCF算法时详细介绍过了，所以这里直接给出计算的代码：

```
function CalculateSimilarity(entity-items)
    w = dict()
    ni = dict()
    for e,items in entity_items.items():
        for i,wie in items.items():
            addToVec(ni, i, wie * wie)
            for j,wje in items.items():
                addToMat(w, i, j, wie, wje)
    for i, relate_items in w.items():
        relate_items = {x:y/math.sqrt(ni[i] * ni[x]) for x,y in relate_items.items() }
```

# 利用物品的内容信息

---

- 内容过滤算法和协同过滤算法的优劣，我们在MovieLens和GitHub两个数据集上进行了实验。
  - MovieLens数据集，它也提供了有限的内容信息，主要包括电影的类别信息（动作片、爱情片等类别）
  - GitHub数据集包含代码开发者对开源项目的兴趣数据，它的用户是程序员，物品是开源工程，如果一名程序员关注某个开源工程，就会有一条行为记录。

# 利用物品的内容信息

- 内容过滤算法和协同过滤算法的优劣。
- 这两种算法融合一定能够获得比单独使用这两种算法更好的效果。

表3-5 MovieLens/GitHub数据集中几种推荐算法性能的对比

方 法	准 确 率	召 回 率	覆 盖 率	流 行 度
MovieLens				
Random	0.631%	0.305%	100%	4.3855
MostPopular	12.79%	6.18%	2.60%	7.7244
ItemCF	<b>22.28%</b>	<b>10.76%</b>	<b>18.84%</b>	<b>7.254526</b>
ContentItemKNN	6.78%	3.28%	19.06%	5.8481
GitHub				
Random	0.000985%	0.00305%	84.18%	0.9878
MostPopular	1.18%	4.36%	0.0299%	7.1277
ItemCF	2.56%	9.44%	33.71%	2.9119
ContentItemKNN	<b>6.98%</b>	<b>25.75%</b>	<b>34.44%</b>	<b>1.7086</b>

# 利用物品的内容信息

- 向量空间模型在内容数据丰富时可以获得比较好的效果。以文本为例，如果是计算长文本的相似度，用向量空间模型利用关键词计算相似度已经可以获得很高的精确度。但是，如果文本很短，关键词很少，向量空间模型就很难计算出准确的相似度。
- 在这种情况下，首先需要知道文章的话题分布，然后才能准确地计算文章的相似度。建立文章、话题和关键词的关系是**话题模型**（topic model）研究的重点。

# 利用物品的内容信息

- 话题模型的基本思想是，一个人在写一篇文档的时候，会首先想这篇文章要讨论哪些话题，然后思考这些话题应该用什么词描述，从而最终用词写成一篇文章。因此，文章和词之间是通过话题联系的。
- LDA中有3种元素，即文档、话题和词语。每一篇文档都会表现为词的集合，这称为词袋模型(bag of words)。每个词在一篇文章中属于一个话题。令 $D$ 为文档集合， $D[i]$ 是第 $i$ 篇文档。 $w[i][j]$ 是第 $i$ 篇文档中的第 $j$ 个词。 $z[i][j]$ 是第 $i$ 篇文档中第 $j$ 个词属于的话题。

# 利用物品的内容信息

- LDA的计算过程包括初始化和迭代两部分。首先要对 $z$ 进行初始化，而初始化的方法很简单，假设一共有 $K$ 个话题，那么对第 $i$ 篇文章中的第 $j$ 个词，可以随机给它赋予一个话题。同时，用 $NWZ(w, z)$ 记录词 $w$ 被赋予话题 $z$ 的次数， $NZD(z, d)$ 记录文档 $d$ 中被赋予话题 $z$ 的词个数。

```
foreach document i in range(0, |D|):  
    foreach word j in range(0, |D(i)|):  
        z[i][j] = rand() % K  
        NZD[z[i][j], D[i]]++  
        NWZ[w[i][j], z[i][j]]++  
        NZ[z[i][j]]++
```

# 利用物品的内容信息

- 在初始化之后，要通过迭代使话题的分布收敛到一个合理的分布上去。  
伪代码如下所示：

```
while not converged:
    foreach document i in range(0, |D|):
        foreach word j in range(0, |D(i)|):
            NWZ[w[i][j], z[i][j]]--
            NZ[z[i][j]]--
            NZD[z[i][j], D[i]]--
            z[i][j] = SampleTopic()
            NWZ[w[i][j], z[i][j]]++
            NZ[z[i][j]]++
            NZD[z[i][j], D[i]]++
```



# 利用物品的内容信息

- LDA可以很好地将词组合成不同的话题。

“Arts”	“Budgets”	“Children”	“Education”
NEW	MILLION	CHILDREN	SCHOOL
FILM	TAX	WOMEN	STUDENTS
SHOW	PROGRAM	PEOPLE	SCHOOLS
MUSIC	BUDGET	CHILD	EDUCATION
MOVIE	BILLION	YEARS	TEACHERS
PLAY	FEDERAL	FAMILIES	HIGH
MUSICAL	YEAR	WORK	PUBLIC
BEST	SPENDING	PARENTS	TEACHER
ACTOR	NEW	SAYS	BENNETT
FIRST	STATE	FAMILY	MANIGAT
YORK	PLAN	WELFARE	NAMPHY
OPERA	MONEY	MEN	STATE
THEATER	PROGRAMS	PERCENT	PRESIDENT
ACTRESS	GOVERNMENT	CARE	ELEMENTARY
LOVE	CONGRESS	LIFE	HAITI

## 利用物品的内容信息

- 在使用LDA计算物品的内容相似度时，我们可以先计算出物品在话题上的分布，然后利用两个物品的话题分布计算物品的相似度。

$$D_{\text{KL}}(p \parallel q) = \sum_i p(i) \ln \frac{p(i)}{q(i)}$$

# 目录

---

1

冷启动问题简介

2

利用用户注册信息

3

选择合适的物品启动用户的兴趣

4

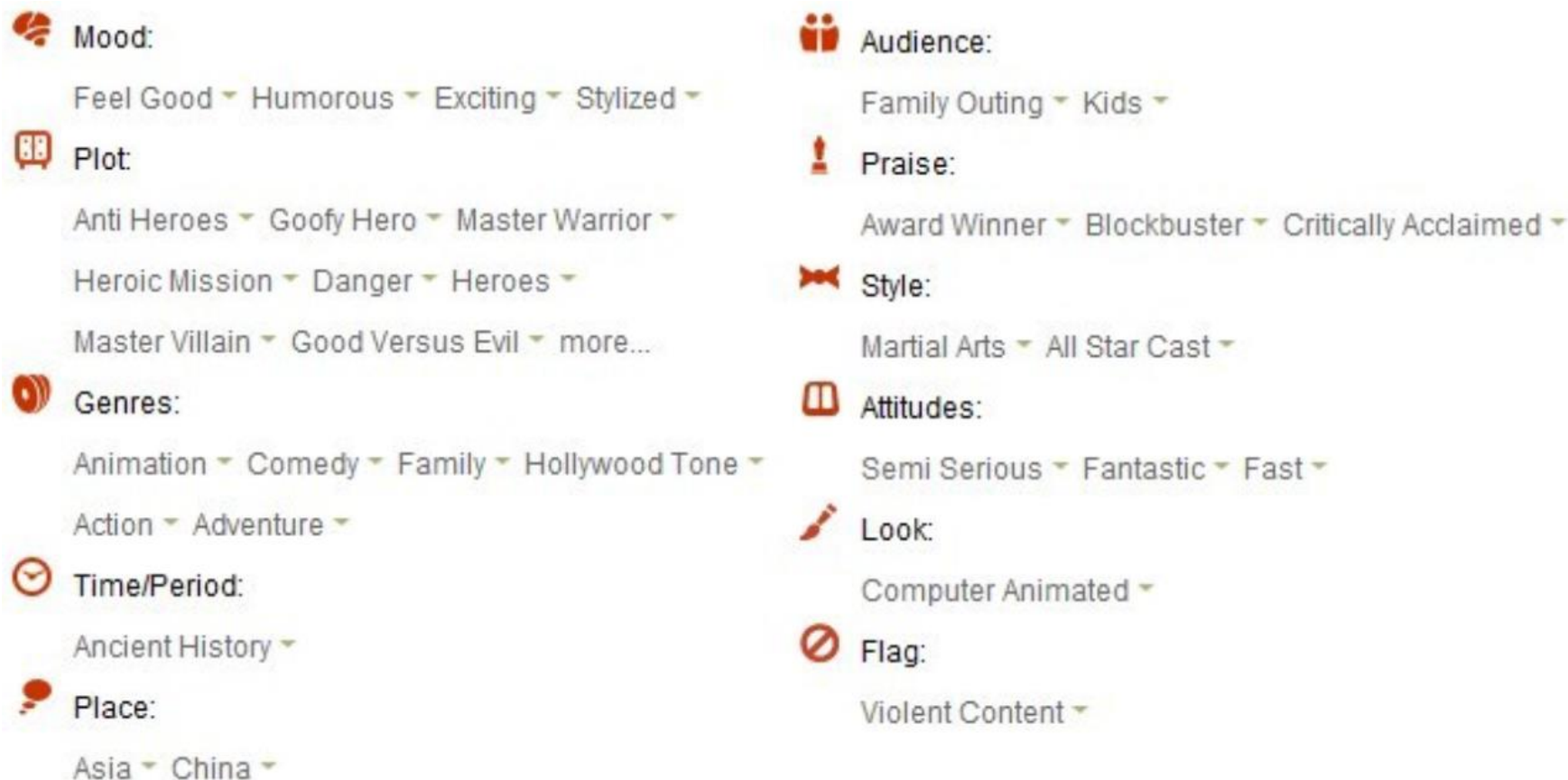
利用物品的内容信息

5

发挥专家的作用

# 发挥专家的作用

- 多推荐系统在建立时，既没有用户的行为数据，也没有充足的物品内容信息来计算准确的物品相似度。那么，为了在推荐系统建立时就让用户得到比较好的体验，很多系统都利用专家进行标注。



# 发挥专家的作用

- 这里的基因包括如下分类：
  - 心情 (Mood) 表示用户观看电影的心情，比如对于《功夫熊猫》观众会觉得很有趣，很兴奋。
  - 剧情 (Plot) 包括电影剧情的标签。
  - 类别 (Genres) 表示电影的类别，主要包括动画片、喜剧片、动作片等分类。
  - 时间 (Time/Period) 电影故事发生的时间。
  - 地点 (Place) 电影故事发生的地点。
  - 观众 (Audience) 电影的主要观众群。
  - 获奖 (Praise) 电影的获奖和评价情况。
  - 风格 (Style) 功夫片、全明星阵容等。
  - 态度 (Attitudes) 电影描述故事的态度。
  - 画面 (Look) 电脑拍摄的画面技术，比如《功夫熊猫》是用电脑动画制作的。
  - 标记 (Flag) 主要表示电影有没有暴力和色情内容。

# 发挥专家的作用

---

- Jinni在电影基因工程中采用了半人工、半自动的方式。
  - 首先，它让专家对电影进行标记，每个电影都有大约50个基因，这些基因来自大约1000个基因库。
  - 然后，在专家标记一定的样本后，Jinni会使用自然语言理解和机器学习技术，通过分析用户对电影的评论和电影的一些内容属性对电影（特别是新电影）进行自己的标记。
  - 同时，Jinni也设计了让用户对基因进行反馈的界面，希望通过用户反馈不断改进电影基因系统。

# 内容回顾

---

1 冷启动问题简介

2 利用用户注册信息

3 选择合适的物品启动用户的兴趣

4 利用物品的内容信息

5 发挥专家的作用

Thank you !