目 录

[第 一 章 背景介绍 2](#_Toc105219924)

[第 1 节 模板使用说明 2](#_Toc105219925)

[第 2 节 如何刷新目录 2](#_Toc105219926)

[第 二 章 正文要求说明 4](#_Toc105219927)

[第 1 节 字体和大小 4](#_Toc105219928)

[第 三 章 公式排版 5](#_Toc105219929)

[第 1 节 Microsoft Equation Editor 5](#_Toc105219930)

[第 2 节 MathType 5](#_Toc105219931)

[第 3 节 TeX/LaTeX 6](#_Toc105219932)

[第 四 章 图形和表格 7](#_Toc105219933)

[第 1 节 图形 7](#_Toc105219934)

[第 2 节 表格 7](#_Toc105219935)

[第 五 章 定理环境 8](#_Toc105219936)

[第 1 节 自定义定理环境 8](#_Toc105219937)

[第 2 节 已有环境 8](#_Toc105219938)

[第 3 节 自定义环境 8](#_Toc105219939)

[第 六 章 参考文献 10](#_Toc105219940)

[第 1 节 创建引文项 10](#_Toc105219941)

[第 2 节 其他的参考文献管理软件 10](#_Toc105219942)

[第 3 节 文献格式 10](#_Toc105219943)

[第 七 章 引用问题 12](#_Toc105219944)

[第 1 节 公式编号 12](#_Toc105219945)

[第 2 节 图表和章节标题 12](#_Toc105219946)

[第 3 节 定理环境 12](#_Toc105219947)

[致谢 13](#_Toc105219948)

在推荐系统中估计物品曝光度

周易

学号：13300180065

专业：数学与应用数学

摘要：协同过滤是一种常用的推荐方法，通过分析用户的偏好和物品（比如电影，歌曲，游戏等）的属性，来发现出未知的“用户-物品”关系。在处理隐式反馈数据的过程中，传统的方法[1]将所有的数据都考虑进去处理，然而这是有失偏颇的，因为每个用户的视野和精力是有限的，有些物品根本没曝光在用户眼前，因此对应那些的数据是难以反应用户偏好的。因此，一种通过概率方法估计物品曝光度的混合模型[2]被提了出来，该方法把曝光与否看成模型中的一个隐变量，更加能反映客观现实也更具有可解释性。然而我在实践中发现该模型在一些隐式反馈数据集上表现不好，本文分析了原因，并改进了该模型，从而可以处理各种类型的隐式反馈数据集,并在一些新的数据集上对该模型做了实验。其次，推荐理由是衡量一个推荐系统的好坏的重要指标，好的推荐理由可以让用户更信任你的推荐，本文给出了一个基于该模型的推荐理由的生成方法。

关键字：协同过滤，概率图，混合模型，EM方法，推荐系统

# 引言

为合适的用户推荐合适的物品，在这个信息量飞速增长的时代是一个非常重要的问题。一方面，面对过于庞大和冗杂的信息库，用户节约了自己亲自去搜索和试验位置事物的时间，可以直接获得一个经过分析历史数据而得到的个性化推荐列表；另一方面，对拥有自己产品和大量用户的公司来说，更好的满足了客户需求，因此能够取得更好的商业表现，赚取更多的利润。协同过滤就是一种出色的推荐方法。

具体地说，使用协同过滤的推荐系统面对的是这样一个问题：观测到一组用户和物品交互的信息，可以是评分、点击、浏览等，我们的目标是从中推断出用户的偏好，然后借助分析得到的用户偏好来为他们推荐物品。

从策略上来说，协同过滤可以有两种思路。一种是“邻居方法”（Neighborhood method），邻居方法专注于计算用户与用户之间、物品与物品之间的关系。比如，在电子商务中，得到商品之间的相似度之后，可以为用户推荐与他们曾经购买过的具有相似属性的物品。在电影评价估计中，得到用户之间的相似度之后，在估计一个用户对一部他没有看过的电影的时候，可以选取一部分和他思想相近、兴趣相投的用户，利用他们对这部电影的评价来估计他对这部电影的喜好。另一种是”隐向量方法”（Latent Factor method），这种方法和矩阵的奇异值分解有很大关系，奇异值分解在信息检索问题中被认为是一个很好的得到隐语义向量（latent semantic factors）的方法。类似的，它把得到的观测值看做一个“用户-物品”矩阵，然后将推荐看做一个矩阵的分解问题，分解得到的是一组用户向量和一组物品向量，这些向量刻画的是用户和物品的各种属性。在估计某用户是否喜欢某个物品的时候，只需要将该用户对应的向量和该物品对应的向量做点积，把这个乘积当做用户对物品的“偏好”的估计。在这篇文章使用的是“隐向量方法”。

从数据性质上来说，协同过滤可以分为从显式反馈数据中推荐和从隐式反馈数据中推荐。最方便而且高质量的数据是显式反馈数据，显式反馈数据是指用户自己本身记录的对物品的评价。比如说豆瓣网用户对不同书籍的评分，比如Netflix利用一个大拇指朝上或朝下的按钮，对它的用户收集的在对观看过的电影和电视剧的评价。推荐系统可以通过用户反映的喜欢或者讨厌，推断出用户的偏好。但是显式反馈数据往往比较难以获取，因为他们需要用户的参与才能采集到，需要用户自己参与进对物品的评价，相对而言是一种“昂贵”的数据。另一种更加容易得到的，范围更广的数据是隐式反馈数据。隐式反馈数据，往往是用户行为的副产品，包括用户的点击、对商品的浏览、过往的购买记录和观看电视剧的总时长等。从隐式反馈中推断用户的偏好往往比从显式反馈更难，这是和隐形反馈数据本身的性质有关。

隐式反馈的数据的特点决定了它们不能直接使用成熟的、被广泛研究的处理显式反馈数据的方法。隐式反馈数据的主要性质有以下：

* 没有负面反馈。通过观察用户的行为，我们可以推测出用户大概会喜欢某些选项。比如某个用户在在线音乐平台上点击了一首摇滚乐团五月天的歌曲，这个数据可以反应这个用户有可能喜欢摇滚类型的音乐或者是五月天的粉丝。但你不能通过他没有点击另一首古典音乐就断定该用户不喜欢这种类型的音乐或这位音乐家。因为有可能是那位古典音乐的音乐人相对商业化成功的五月天来说影响力比较小，曝光度很小，用户根本没有发现这个音乐，所以没有点击。没有负面反馈的这个性质，导致我们必须考虑一整个“用户-数据”矩阵，包括用户的正面反馈和空白反馈，并且从看上去是“空白信息”的反馈中，发掘负面的反馈。
* 数值信息不能代表用户喜好。在显式反馈数据中，用户明确地表达了他们的喜好，比如给非常喜欢的电视剧打五分，给不喜欢的电视剧打一分。但是在隐式反馈数据中，用户喜好不能直接从数值信息上看出来。隐式反馈数据往往代表着签到信息，购买记录，观看时长，用户行为的频率等。比如一个用户看两部电视剧《人民的名义》和《生活大爆炸》，观察到他每周花在《人民的名义》上的时间长于《生活大爆炸》。但这个信息并不能有力地证明该用户更喜欢《人民的名义》，因为也许这个用户的习惯就是每周看一集在看的剧集，但是《人民的名义》每集的长度就是长于《生活大爆炸》。

为了处理上述的问题，加权的矩阵分解方法[1](Weighted Matrix Factorization, 简写为WMF)引入了启发式的想法，给予用户的各种行为不同的信心(Confidence)，对那些“空白信息”给予很低的信心，并把用户行为看做一个二值化的决定（即非0即1的数据），从而把问题处理成一个回归的问题。但是这种方法是存在问题的是它把所有的数据都以同一种模式处理。

WMF借鉴了处理显式反馈数据的方法，但是过高地估计了没有点击的物品的权重，因为对于绝大多数用户没有点击的数据来说，用户都是因为没有看到他们才没有点击，只有其中一小部分是因为不喜欢才没有点击。于是另一种想法[2]摒弃之前的启发式的观点，采取直接的概率手段，对物品是否曝光在用户面前进行了建模，同时继承了之前的把用户行为二值化的想法。把因为用户不喜欢一个物品而没有点击和用户因为没有发现这个物品而没有点击区分了开来，因此建立了一个混合模型，称为模拟曝光率的矩阵分解方法（Exposure Matrix Factorization, 简写为 ExpoMF）。

ExpoMF更能反映现实的情况，实验的效果也更加优异。但是原方法在处理二值化的数据，比如签到、购买等数据的时候比较优异，但在反映表示频率，点击次数，时长的这种类型隐性数据的时候效果不够理想，于是我将[1]中的信心概念重新引入到ExpoMF中，因为信心虽然不应该直接用作为回归的权重，但数值的相对大小可以帮助我们进行”曝光度“的建模，补充了ExpoMF，使得它可以处理新的数据类型。

我将ExpoMF模型处理了来自在线音乐平台的音乐的点击数据，电影评价网站的的电影观看数据和在线游戏公司的游戏购买数据。在实验中，ExpoMF在衡量推荐效果上的各个标准下，都超过了目前为止最主流的方法——WMF[1]。

本文接下来的部分按以下顺序展开：在第二章背景知识中回顾采用隐向量方法的协同过滤模型，在第三章中描述一整套ExpoMF模型。在第四章中，详细地描述了在各种数据集上的实验过程、评价体系和实验结果。在第五章中给出了一个生成推荐理由的方法。在第六章中给出了结论。

# 背景知识

在这一章中，在第一节介绍了隐向量方法以及它是如何处理显式反馈数据的。第二节介绍了如何通过类比，处理隐式反馈数据。第三节介绍了，如何从概率的角度来看待隐向量方法。

## 协同过滤的隐向量方法

隐向量方法把推荐系统内的所有的用户(user)和物品(item)映射到一个K维的实向量空间上：每个物品对应于一个向量，这个向量叫做物品的隐向量，这个向量的每个坐标反映一个属性维度。每个用户对应于一个向量,这个向量叫做用户的隐向量，刻画了用户在各个属性上的感兴趣的程度。

使用相应的两个向量的内积去刻画用户u和物品i的相关程度——用户对物品的总体的感兴趣的程度。

我们把观察到的数据记为矩阵R，矩阵中每个数据为。每个数据条目记录的相应的用户行为信息。上述的想法可以描述为：

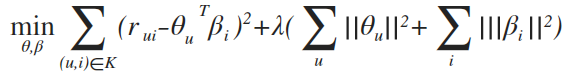


因为，观察到的数据矩阵是非常稀疏的，有很多空缺值。早期的方法是去填补空缺值，去做矩阵的分解，然而这样做的代价非常昂贵，以矩阵的奇异值分解为例，SVD分解的时间复杂度是三次方级别的，当数据量变大的时候，计算代价会变得非常大。早期的方法面对的另一个问题是，面对如此稀疏的数据真的值得我们去填补空缺值吗？

于是，当Netflix——美国的一个在线影视公司——举办[Netflix Prize](http://www.netflixprize.com/)比赛的时候，给出一个巨大的数据集，让全世界的参赛者帮他们解决预测电视电影评分预测的问题的时候，早期的方法就陷入了困境。于是参赛者想出了直接在观察到的数据集上进行隐向量的学习，同时引入正则化项避免过拟合的问题，这个想法成为后来的主流方法。从而，原问题被理解成了一个在观察值上训练用户隐向量和物品隐向量的回归问题。

对于一个显性反馈数据集的带正则化项的最小二次误差模型如下：

这里K是指被观察到的、已经被记录的数据对的集合（比如所有的用户评分）。

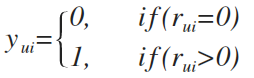


公式 1

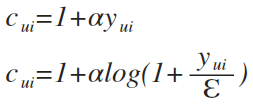
## 处理隐式反馈数据的隐向量方法

当面对隐性反馈数据的时候，情况变得不再相同。为了克服隐性反馈数据的特性，加权的矩阵分解方法[1](Weighted Matrix Factorization)被提出，这也是目前最主流、效果优秀的方法。这个模型采用的是简单的启发式想法，给予不同的数据点不同信心，作为回归的系数。

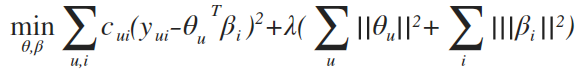
WMF首先将原始的数据二值化：



然后根据不同的原始数据，给每个数据点不同的信心（Confidence）作为回归的权重。具体的信心函数因数据的种类而异，原始数据的值越大，我们相对的确信用户对喜欢那个物品的信心就越大。常见的信心函数可以选择为线性函数对数函数：



和处理显式反馈的方法做类比，WMF把处理隐式反馈数据，归结为如下的回归问题：

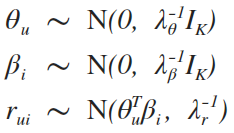


公式 2

## 隐向量分解的概率解释

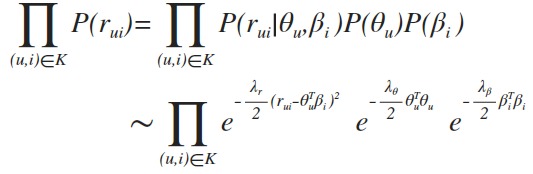
用隐向量分解矩阵，也可以从概率的角度出发，得到的结论和之前一致，但是是一种不同的诠释，相应的方法称为概率矩阵分解（Probabilistic matrix factorization，简写为PMF）[4]。

概率矩阵分解，从生成数据的模型的角度看，假设用户向量和物品向量两个特定的分布中生成，分别代表着用户的偏好和物品的属性。然后，观察值在从一个期望为的分布中生成。具体地说，一个正态分布的矩阵分解可以写成：

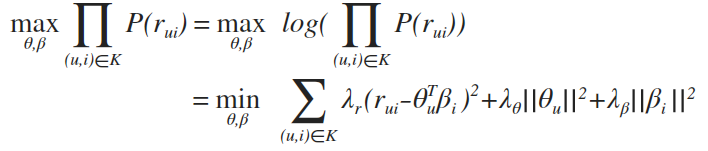


这里，N代表正态分布，K是隐向量的维度，代表K维单位阵，是超参数（hyper parameter）。

由以上的模型，我们可以计算观察值的出现的概率，针对本章第一节的显式反馈数据，我们有：



采用最大似然估计的想法，最大化上述概率，我们就可以得到：



这就是本章第一节的公式1。

同理，针对隐性反馈数据，PMF可以改写成：



这里，第二节中描述的“信心”。类似地，利用最大似然估计去估计参数，我们可以得到公式二。

# ExpoMF模型

在这章中给出ExpoMF的描述。第一节给出模型的介绍，第二节给出估计曝光度的方法，第三节给出模型的推断算法，第四节给出推荐理由的生成方法。

## 模型描述

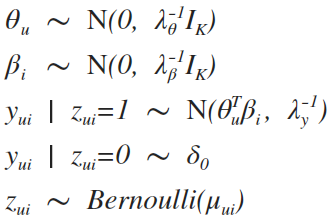
在推荐系统中，每个观测数据，都可以用用户u=1,2,3..U和物品i=1,2,3…I的组合来标记。

ExpoMF是用于处理隐式反馈数据。隐式反馈数据也有不同种类，如果是反应点击次数，观看时间长短的隐式反馈数据，在这里就用可以用表示。按照处理隐式反馈数据的方法，将其二值化（0或1），沿用之前的记号，记为，另一种本身就是0-1型的隐式反馈数据，比如是否观看某电影的记录，电子游戏的购买记录等，就记为记为。

接下来，对于每个组合(u,i)，我们引入一个隐变量，用这个变量去描述，用户u是否暴露在物品i面前：若，则代表用户u已经看到了物品i，之后的用户行为，就可以由他的偏好和物品的属性解释，和标准的方法类似的，用一个K维向量来描述用户u的偏好，用一个向量K维向量来量化物品i的属性，用它们的内积反映这个用户行为的程度；若，则代表用户根本没有看到物品i，因此后面他不应该会对物品做任何动作（点击，购买等）。

根据物品是否暴露在用户面前来做区分，构成了一个混合模型：

这里，表示，而表示我们对该物品是否会暴露在用户面前的一个先验估计，其余符号均沿用之前的含义。上述模型的概率图表示在图1(a)中给出。



在开始计算这个ExpoMF模型之前，我们手上的数据是一个用户-物品兴趣矩阵，每个 标记着用户u是否对物品i感兴趣，0代表不感兴趣，1代表感兴趣。ExpoMF区别于其他矩阵分解的方法的地方在于，它有一个暴露矩阵，运用这个暴露矩阵Z把兴趣矩阵Y的分解看做一个混合模型。这个两个矩阵之间是由联系的。如果，，因为你如果对一物品表现出了兴趣（发生了购买、点击），那么这个物品一定已经被你观察到了。但如果，那么对我们来说是未知的，这也是我们需要去建立模型模拟的关键点。从实践的角度说，因为兴趣矩阵Y往往是非常稀疏的，所以绝大多数的都是未知的。

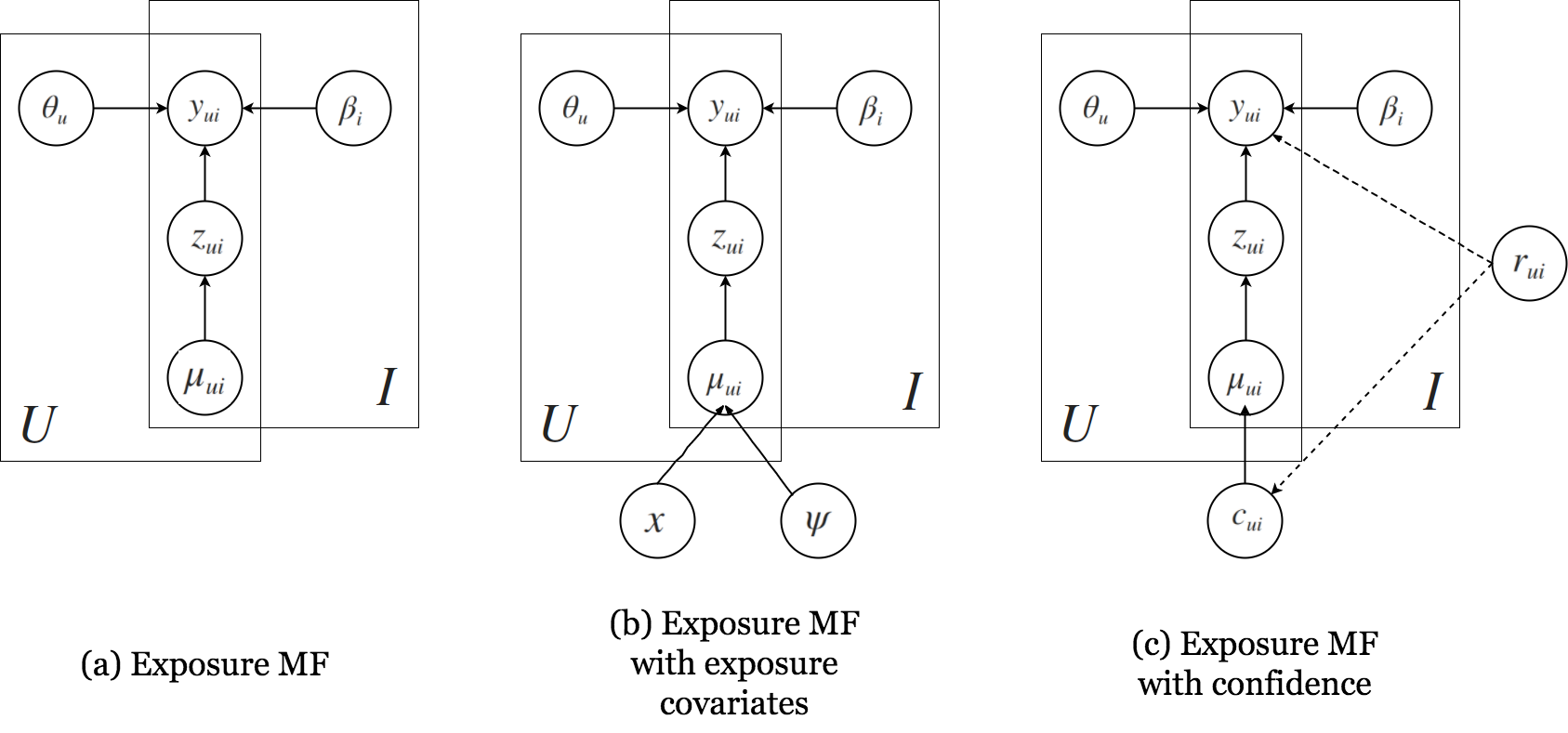


图 1：ExpoMF的概率图表示。一条带箭头的、从a到b的实线段表示，在模型中b的生成依赖于a。一条带箭头的、从a到b的虚线段表示，在模型中b是a数据处理后的结果。

ExpoMF模型推导出的概率公式如下：



公式 3

和WMF类似，ExpoMF也会选择性地降低”空缺值“对模型的影响，但是区别在于不是利用方差项来降低影响，而是通过判断是否该物品是否暴露在用户面前，来降低”空缺值“的影响。

公式3显示这是一个混合模型，如果把所有的都固定为，那么我们重新发现了标准的矩阵分解方法。

在进行更进一步的描述之前，先来看看为什么这个模型对隐式反馈数据起作用。对上述的公式3 取对数函数后，我们有：



公式 4

设想一个情景，当一个用户u对物品i有很大的偏好的时候（），但没有发生用户行为（），那么我们可以发现公式4右边的第二项会变得很非常小，甚至是负数，这会成为一个惩罚项，迫使选择一个非常小的值。换句话说，这个模型认为：如果你喜欢一个东西，但我们发现你没有去点击它，那么很有可能是因为你没看见它。

## 为曝光度建模

之前在第一节中的讨论没有涉及到如何选择和学习先验估计。

估计曝光度的方法当然是和你目前掌握了什么信息有关。下面根据掌握信息的不同，提出三种可能的方法。前两种来自[2]，第三种由本文提出。

1. Per-Item ：当没有任何外部信息的时候，最简单的估计方法，就是统计每个物品的在所有用户面前的曝光度，作为。
2. Exposure covariates：当推荐带有文本或者地理信息的物品时候，可以首先将文本信息通过自然语言处理处理成L维的向量，每个维度表示文本内容的主题，或者将地理信息进行聚类成L个分类，得到一个包含分类信息的L维向量。值得注意的是，这里的L维和隐向量方法的K维没有任何关系，不一定要取相同的值。把这个向量记做，在两种情形下，都应该被规范到单位长度。在模型中，我们对如下函数进行学习：



这是模型需要学习的关于的协变量。是sigmoid函数。从可解释性的角度来说，可以被理解为用户平常花在各种主题的文本中的时间或者出现在各个地理位置的频率。

1. Confidence-Per-Item : 当推荐用户行为中带有频率和或者反映时间长短的物品的信息的时候，直接套用方法1效果会不够理想。原因是忽视了原始数据中反映用户行为的相对程度的值被忽视了，这导致对的估计不够准确。比如，对于音乐点击数据来说，一个频繁点击某首歌的用户行为，显然不应该和点击过一次某首歌的用户行为划上等号。为了量化不同的用户行为，重新引入[1]中的“信心函数”是有必要的，把用户行为的信心作为权重，计算物品的在所有用户面前的曝光度，作为。

## 模型推断算法

因为ExpoMF模型是一个带有隐变量混合模型，我们使用处理带有隐变量的模型的成熟方法——EM方法（expectation-maximization）去估计本模型的参数。EM方法可以被理解成两步，先在已有的模型中估计隐变量的值(E-Step)，然后，在知道隐变量之后，利用最大似然估计，估计模型的参数(M-step)。

* **E-step:**

这一步，需要估计的是暴露矩阵中的值的期望。

# 图形和表格

## 图形

一般说来，大家都会插入图片。有难度的是图片的编号，想说明的是图片的编号也不是手工完成的，如：

这个过程的基本步骤是，插入一个文本框，在文本框中插入图片，然后右键点击图片，选择题注，然后选取合适的即可，如上面就是新建的一个图的标签。这样做的目的是可以使图片以行间的方式插入到文章中，当然，如果是其他的类型，可以考虑其他的实现方案。

j0149627

图表 1.0这是一幅牛的图片

## 表格

插入表格与图片类似。当然，可以使用Excel预先作一个表格，然后导入进来，但是word本身也可以胜任一部分简单表格的绘制，如：

|  |  |  |  |
| --- | --- | --- | --- |
|  | 从不吸烟 | 偶尔吸烟 | 经常吸烟 |
| 患肺癌 | 10 | 200 | 40 |
| 不患肺癌 | 233 | 40 | 80 |

表格 2.1 一个简单的表格

加入编号，类似于图片。我们这里定义了表格的题注样式“题注表格”。如果你的题注不是该样式，选中后使用该样式即可。

# 定理环境

Word并不像TeX/LaTeX为我们提供了合适的定理环境，因此需要我们另想办法。

## 自定义定理环境

我们已经使用了“定理样式”作为定理排版的样式，如：

定理1.1. 对顶角相等。

如果大家需要其他的如引理，公理，定义等环境可以仿照定义。

定理1.2. 三边对应相等的三角形全等。

我们将这个过程也定义成了宏，在工具栏Theorem里面。书写过程如下：先写好定理本身，然后在该段落处放置光标，打开Theorem工具栏，点SetTheorem，即可见到效果。请尝试下面一个例子：

三角形内角和为。

## 已有环境

为了方便大家使用，我们定义了定义环境。如果大家有兴趣，可尝试下面的例子（请使用工具栏）：

定义2.1. GNU是GNU is not Unix。

请尝试：

GNU是GNU is not Unix。

## 自定义环境

为了方便大家自己做出类似的格式，这里简要叙述一下如何产生合适的环境。首先为该环境创建一个宏，可以直接复制已有的，如InsertTheoremHead，把名字改为需要的，把代码中的“定理”换为你需要的。否则，基本过程如下，先点击开始录制宏，然后插入你需要的标题，然后插入一个域，使用Ctrl + F9。输入STYLEREF 2 \s，到该与后面输入分隔符，一般是一个英文的句号，然后再插入一个域，内容为SEQ theorem \\* ARABIC \s 2。接着在后面加上一个英文句号和一个空格。最后选中前面输入所有内容，将样式设置为“定理标题”。结束宏的录制，执行看看效果。

然后我们再来做一个工具栏上的类似按钮。先新建宏，并点击工具栏的按钮，把那个命令拖入到Theorem工具栏上，关闭对话框后应开始录制宏。先运行一个名为SetTheoremBodyStyle的宏，将本行文字变为相应样式，然后按Home到行首，执行Insert\*\*\*Head的命令插入标题。结束宏的录制。

最后自己看看效果。

# 参考文献

这章介绍如何创建参考文献。

## 创建引文项

其实很简单，在插入的地方点插入菜单，选择插入引用，然后选择索引和目录，换到引文选项页。但是该方案不大方便管理多方面的文献，因此我们常使用其他的软件辅助。

## 其他的参考文献管理软件

常见的有Reference Manager，Endnote，Biblioscape。这里就Endnote作一个介绍，因为它管理的文献可以以BibTeX的方式导出，方便在TeX/LaTeX里面使用。

### Reference Manager

与word的集成度高，即写即用。管理专业，能够依照要求生成漂亮的参考文献列表。据说不支持中文。

### Endnote

功能不如前者强大，但是可以导出为BibTeX能处理的bib文件。据说也不支持中文。

### Biblioscape

据说支持中文。

## 文献格式

所有参考文献均用尾注形式列在论文篇末，内容包括：主要负责人（作者，编者）, 文献题名。出版地，出版年份，起止页码。（如果文献是期刊杂志内的文章，则除要列出作者和题名外还要注明期刊名，出版时间，卷号或期号，起止页码）。

英文出版物见[2][3]，英文期刊见[1]，中文出版物见[4]，中文期刊见[5]。

建议参考文献按作者姓氏的字母排序。

注意“参考文献”不写成论文的一章.

# 引用问题

这里讨论如何引用一些编号。

## 公式编号

这个功能由MathType提供，其实公式编辑器也应该可以实现。引用的时候使用MathType工具栏上面的按钮比较方便，如。

## 图表和章节标题

使用插入引用中的交叉引用项，如前面图形和表格的表格 2.1和图表 1.0这是一幅牛的图片。

## 定理环境

尚未弄清楚怎么引用。

# 致谢

请对帮助过您完成论文的老师、同学致谢。也可以在此对您四年大学生活有重要帮助的人致谢。

**“致谢”本身不作为一章，致谢内容的字体大小不宜与作为标题的``致谢"两字的大小有很大的反差. 这一点尤其请使用word模板的同学注意。**一般说来, 杂志论文的致谢在文章正文结束、参考文献前（即本模板中它所处的位置）; 学位论文的致谢在最后一页，并宜单独成页; 书籍的致谢在序言结尾。

本模板由何力同学 ( 01级) 根据候捷《word排版艺术》一书在李湛同学 ( 01级 ) 的模板基础之上整理而成。在此，谨对这两位同学致以诚挚的感谢！不足之处，请大家斧正。欢迎大家对模板进行修改，特别, 我们欢迎尽量简单的新模板。相关事宜请和楼红卫老师或杜雅倩老师联系。

Y. Koren, R. Bell, and C. Volinsky. Matrix factorization techniques for recommender systems. Computer, 42(8): 30–37, Aug. 2009. ISSN 0018-9162.

* A. Mnih and R. Salakhutdinov. Probabilistic matrix factorization. In Advances in Neural Information Pro- cessing Systems, pages 1257–1264, 2007.

参考文献

#### Yifan Hu, Yehuda Koren, and Chris Volinsky, “Collaborative Filtering for Implicit Feedback” , 2008 Eighth IEEE International Conference on Data Mining，2008

#### Dawen Liang, Laurent Charlin, James McInerney, David M. Blei, “Modeling User Exposure in Recommendation” ,International Conference on World Wide Web , 2016

#### Y. Koren, R. Bell, and C. Volinsky. “Matrix factorization techniques for recommender systems.”, Computer, 42(8): 30–37, Aug. 2009. ISSN 0018-9162, 2009

#### A. Mnih and R. Salakhutdinov. “Probabilistic matrix factorization.”, In Advances in Neural Information Pro- cessing Systems, pages 1257–1264, 2007.

#### S.Funk,“NetflixUpdate:TryThisatHome”,http://sifter.org/~simon/journal/20061211.html ,2006

\