设计报告

**摘 要：**

本设计报告包括了MSN电视节目推荐系统的主要功能以及软件的设计和实现方法等等。

本系统通过用户的观看历史以及用户的个人信息来相应地推荐电视节目及其播放时间，并通过MSN为媒介来与用户来交流。用户可以通过MSN与聊天机器人进行交流，以获取信息。聊天机器人会将必要的信息传给概率分析器进行记忆和预测。概率分析器将使用数据挖掘的部分知识，从大量的数据中获取模式，来推测用户的喜好。电视节目的信息将从网络中获取和提炼。所有的信息将有序地存储在数据库中。

**关键词：**

MSN Chat Robot；概率分析器；推荐系统。

目 录

1. 概述 1

1.1 背景 1

1.2 系统简介 1

1.3 主要技术 1

1.3.1 MSN聊天机器人 1

1.3.2 数据库 1

1.3.3 概率分析器 2

1.3.4 信息获取 2

2. 系统总体设计 3

2.1 聊天机器人 3

2.1.1 模块简介 3

2.1.2 模块组成 3

2.1.3 模块联系 4

2.2 数据库模块 5

2.2.1 数据库中包含的表 5

2.2.2 数据库接口部分 6

2.3 概率分析器部分 9

2.3.1 决策树与概率分析器的区别 10

2.3.2 Bayes网络 10

2.3.3 运用策略 11

2.4 电视节目信息提取 12

3. 结论 15

# 概述

## 背景

当今社会飞速发展，我们正在逐渐被数据所淹没。虽然我们有计算机来帮助我们存储和处理数据，但是，要如何从无尽的数据中获取有用的知识仍是一个很新的课题。

大量的数据会提供大量的信息，但是在数据量增加的同时，伴随着人们对它的理解度的降低。隐藏在这些数据后的是具有潜在用处的信息，而这些信息却很少被显现出来或者被开发利用。而本项目是关于如何在数据中寻找模式，从而预测用户的行为，进行合理地推测。

在网络发展日益蓬勃的今天，一个能为用户提供贴心服务的系统自然是各企业的首选。其中，进行最让用户满意的推荐自然成了提高用户忠诚度的首选。数据挖掘理论不断发展，推荐系统也随之优化，本系统就是建立在一个世界闻名的数据挖掘平台Weka之上的。

另外，MSN是一款拥有广大用户群的聊天软件，它也提供丰富的接口。本系统将以MSN为媒介与用户进行交流。

相信基于MSN的推荐系统必将为企业带来效益。

## 系统简介

本系统是一款电视节目推荐系统。用户只需像平时聊天一样，加聊天机器人为好友，就可以与其对话，并得到相应的电视节目信息。

一旦添加为好友，聊天机器人将会把用户的观看节目历史信息记录下来，并以此为依据，推测用户的喜好，进行合理的推荐。用户提供的个人信息越丰富越准确，系统的推荐也就越精确。

推荐系统使用概率分析器来进行记录和预测，这是基于Bayes网络的一种方法。用户的各种信息都会存储在数据库中，便于长期读写。

电视节目的各类信息将会通过网络获取，尽量保证准确和全面。

## 主要技术

### MSN聊天机器人

采用MSN提供的接口设计出一个聊天机器人，它能获取用户的信息，还能向用户进行推荐。

### 数据库

采用Mysql，将数据进行存储。

### 概率分析器

基于Weka平台，采用Bayes网络作为工具进行机器学习和预测。

### 信息获取

从网络中获取电视节目的各类信息。

# 系统总体设计

## 聊天机器人

### 模块简介

ChatRobot模块通过MSN方式与用户进行交流，系统启动时，ChatRobot模块将登录MSN。所有在系统注册过并与ChatRobot成为好友的用户都可以MSN客户端与ChatRobot进行交流。ChatRobot将根据用户的信息作出相应的反应，并将反馈信息发送给用户。不过用户并不需要直接与ChatRobot进行通信，用户的请求将被电视终端接收并转换成一定格式的请求消息发送给ChatRobot，ChatRobot的反馈消息也将进行一定格式的封装。这样的处理比较方便ChatRobot和电视终端对消息的分析。

### 模块组成

ChatRobot主要由3部分组成，MSN Communication，Input Queue和Output Queue，MSN Communication部分负责处理MSN通信所需的一切工作，包括switch board的建立、撤销，消息的发送、接收等；Input Queue负责对用户的消息进行分析，将相应的请求封装成Packet存入工作队列中；Output Queue负责从工作队列中取出Packet，根据Packet的类型向系统的概率分析器模块发送请求。下面是每个组成部分的详细介绍：

 （1）MSN Communication部分：

 ChatRobot的主要部分，实现时使用了一个开源的类库JML。JML支持多种MSNP协议，并且将MSN通信所需的底层操作都封装起来了，使用起来非常方便。JML为MSN通信需要处理的事件如消息的发送、接收，好友的添加、删除，switch board的建立、撤销都提供的接口。MSN Communication这一部分的实现就是将这些接口一一实现，从而定制ChatRobot的行为。当然，在这些事件中，消息接收事件是最重要的。当消息接收事件触发时，本部分将对接收的消息进行分析，即以一定格式从消息中提取出数据，这些数据将用于构造一个Packet，Packet最后被放入工作队列中，供Input Queue部分进行处理。

如果该消息是switch board建立后用户发送过来的第一个消息，本部分还会将switch board保存在一个哈希表中，同时与概率分析器模块建立一个socket连接，该socket连接也将保存在哈希表中。这样，每个用户都有专门的socket连接和switch board负责通信。

 Input Queue部分：

 一个负责处理Packet的独立线程，它与MSN Communication直接之间的同步通过信号量完成。信号量初始化为0，每当MSN Communication向工作队列添加一个Packet，信号量将加1；Input Queue线程的主体是一个死循环，它首先尝试获得一个信号量，若信号量为0，即工作队列为空，线程将睡眠，直至工作队列被添加Packet后线程被唤醒。接下来本部分从工作队列中取出一个Packet，根据Packet的类型和内容构造相应的请求消息并发送给概率分析器模块进行处理。工作队列的互斥访问用过锁保证。

 Output Queue部分：

 一个负责接收概率分析器发回的反馈消息的独立线程，对于推荐节目的请求，Input Queue发送给概率分析器模块后，概率分析器会分析并得到一个推荐节目列表，本部分负责接收这一节目列表并将其通过MSN消息的形式发送给用户。Output Queue线程的主体同样是一个死循环，它尝试接收概率分析器发送过来的消息，若消息为空，线程将被阻塞直至接收到消息。线程被唤醒后从接收到的消息中提取出节目信息并发送给用户。

### 模块联系

ChatRobot模块内部各部分的联系以及ChatRobot模块与其他模块的联系可以通过以下的图表说明：



## 数据库模块

### 数据库中包含的表

1.TV\_info 用来存放电视节目的具体信息，表TV\_info的结构是：



2.user\_info 用来存放用户的具体信息，表user\_info的结构是：



3.Pro\_info 用来存放电视节目的一些具体信息，表Pro\_info的结构是：



4.buffer 用来存放用户对某个电视节目的评分，表buffer的结构是：



5.history 用来存放用户看过的电视节目，表history的结构是：



6.hate 用来存放用户讨厌的电视节目，表hate的结构是：



7.favorite 用来存放用户喜欢的电视节目，表favorite的结构是：



### 数据库接口部分

Database类：

Public static int get\_TVID(String channel,String showname);

 通过确定的channel和showname得到某个电视节目的TV\_ID。

 参数：channel,showname

 返回int类型的TV\_ID

Public static Timestamp getCurtime()

 得到当前的时间

 参数无

 返回Timestamp 类型的时间值

Public static int[] getRandomID()

 得到一个随机的TV\_ID，使用时机在用户最开始使用系统的时候或者是用户信息不全的时候

 参数无

 返回一个int[]类型的TV\_ID；

Public static void getTV()

 得到两个输出信息，分别是一当前时间为准，用户在当前时间内五分钟左右会开始，或者已经开始的节目的list，另外一个信息室两个小时到一个小时之间的节目

 参数无

 返回值无

Public static void add\_history(String userID,TV\_key\_words TV\_keys,double score)

 将用户观看过的电视节目的历史记录写进数据库中的history表中，同一个用户的历史记录上限是30个,并于此同时根据用户的打分对指定的电视节目进行评分。

 参数是：String userID,TV\_key\_words,double score

 返回值无

public static void add\_like(String userID,TV\_key\_words TV\_keys,double score)

 将用户标记为喜欢的电视节目写入数据库中的favourite表中，如果有重复，保留新加入的，删除早插入的，如果该节目在该用户的hate lisit中出现，从该用户的hate list中将其删除。每个用户标记自己喜欢的节目的上限是30个，在每次插入的时候，要根据用户的打分对该电视节目进行评分。

 参数：String userID,TV\_key\_words TV\_keys,double score

 返回值无

Public static void add\_hate(String userID,TV\_key\_words TV\_keys,double score)

 将用户标记喜欢的电视节目插入到数据库中的hate表中，如果有重复，保留新加入的记录，删除后加入的记录，如果该节目在用户的favorite list 中出现，从该用户的favorite list 中将其删除。每个用户标记自己讨厌的节目的上限是30个，在每次插入的时候，要根据用户的打分对该电视节目进行评分。

 参数：String userID,TV\_key\_words TV\_keys,double score

 返回值无

Public static TV\_key\_words get\_TV\_key\_words(int TV\_ID)

 通过PK TV\_ID查询到电视的信息

 参数：int TV\_ID

 返回值：类型为TV\_key\_words类型的变量

Public static TV\_key\_words get\_TV\_key\_words(String time,String channel)

 通过从chatrobot的时间和电视台的名称，找到节目单中同一个频道的最接近这个时间但是已经在播放的节目，然后返回。

 参数：String time,String channel

 返回值：TV\_key\_words 类型的值

Public static String get\_user\_info(String userID)

 得到用户的所有信息，并整合成一个String返回

 参数：String userID

 返回值：String 类型的用户信息

Public static String update\_user\_info(String userID,String attribute,String content)

 对用户的信息的某个属性进行更新，属性的名字是attribute,内容是content。

 参数：String userID,String attribte,String content

 返回值：如果成功就返回update OK,否则不返回

Public static TV\_key\_words[] get\_high\_TV()

 根据用户对电视节目的评分，得到评分最高的几个节目

 参数无

 返回值：TV\_key\_words类型的数组

Public static Strng get\_TV\_info(TV\_key\_words TV\_keys)

 根据TV\_keys中的参数的值，选出相应的TV的信息，然后返回一个String

 参数：TV\_key\_words TV\_keys

 返回值：String类型的电视节目信息

Public static Boolean ishate(String userID,TV\_key\_words TV\_keys)

 判断用户是否讨厌一个节目

 参数：String userID,TV\_key\_words TV\_key

 返回值：如果讨厌返回true，反之返回false。

Public static String get\_history\_but\_not\_hate(String userID,String time)

 得到一个存在于用户历史记录中，但是不存在用户hate记录中的电视的信息的一个字符串

 参数：String userID,String time

 返回值:String

Public static String get\_hot\_but\_not\_hate(String userID,String time)

 得到评分较高同时不是用户讨厌的节目的一个字符串

 参数：String userID,String time

 返回值：String

Public static User\_info get\_user\_info\_class(String userID)

 将相应用户的所有信息整合成一个类型为User\_info的变量

 参数：String userID

 返回值：User\_info类型的变量

Public static void insert\_Pro\_info(Pro\_info pro\_info)

 将节目的信息插入到Pro\_info表中

 参数：Pro\_info pro\_info

 返回值：无

Public static Pro\_info select\_pro\_info(int TV\_ID)

 根据给出的TV\_ID得到想一个的电视节目的详细信息

 参数：int TV\_ID

 返回值：Pro\_info 类型的变量

public static int get\_match(double pro[])

 根据给出的相应类型的概率值，然后得出匹配度最佳的节目，然后返回

 参数：double pro[]，pro[]中存储的是相应类型的概率值

 返回值：int TV\_ID

Public static void add\_buffer(String userID,int TV\_ID,double score)

 临时将某个用户对某个电视节目的打分存储答TV.buffer中

 参数：String userID,int TV\_ID，double score

 返回值：无

Public static Collaboration\_learn[] get\_buffer()

 将buffer表中所有行的信息整合成为一个Collaboration\_learn类型的数组，然后返回

 参数：无

 返回值：一个Collaboration\_learn类型的数组

其它还是用到的类有：Collaboration\_learn,Pro\_info,TV\_key\_words,User\_info类，为数据定义了一个schema，方便存储。

## 概率分析器部分

在数据挖掘飞速发展的今天，推荐系统的重要性也越来越被各大企业所重视。对用户进行有针对性地推荐可以有效地提高用户的忠诚度，提高企业的竞争力。

推荐系统使用的模型有很多种，最为流行的有决策树和概率分析器。

### 决策树与概率分析器的区别

为了从一个实例集进行学习，我们很自然地想到分治法，从而想到决策树。一个决策树上的节点包含了对某个特定属性的测试。一般来说，在一个节点上的测试是将一个属性值与一个常量进行比较，或者使用一个包含一个或多个属性的函数公式进行比较。叶节点对所有到达叶子的实例给出一个分类，或是一组分类。

但是现有的决策树很少有支持模糊匹配的，而在推荐系统中，模糊匹配是十分重要的。在这一点上，概率分析器就做得较好。

### Bayes网络

本系统使用的概率分析器实际上是Bayes网络。

为Bayes网络建立一种学习算法的方法是定义两个组成部分：一个是对基于某个数据集的网络进行评估的评估函数，另一个是在所有可能的网络空间中搜索的搜索方法。

网络中的节点是预设的：每个属性各有一个（包括类属性）。学习网络结构等于是在可能的有向连线组空间进行搜索，对每组的条件概率表进行估计，并计算结果网络基于某个数据集的对数似然，以此作为对网络质量的衡量。

如果对数似然是基于训练数据集进行了最大化，增加更多的有向连线总是会获得较好的结果，造成最终的网络过度拟合。很多方法可用于解决这个问题。一种可能的方法是采用交叉验证法来估计拟合的良好度。第二种是根据参数数目，相应增加网络复杂度惩罚，即在所有概率列表中独立估计的总数目。 每个表中，独立概率的数目是表中所有概率的总个数减去最后一列中的个数，由于每一行概率的和应为1 ，因此最后一列的概率可根据其他列的值而导出。有两个较普及的衡量方法可用于评估网络质量：Akaike信息标准以及MDL度量。

现在来看用于贝叶斯网络学习的实际算法。一个简单而快速的学习算法称为K2起始于某个给定的属性（即节点）排序。然后，对每个节点依次进行处理，贪心地增加从先前处理过的节点指向当前节点的连线。每一个过程中都增加那些能使网络评分达到最高值的连线。当不再有改进时，便将注意力转向下一个节点。每一个节点的父辈数量可以被限制在一个预设的最大值范围内，这可作为一个防止过度拟合的附加机构，由于只考虑起始于前面已经处理过的节点的连线，并且顺序是固定的，这个顺序不会产程环。但是结果依赖于初始排序，因此采用不同的随机排序对算法进行多次运作是有意义的。

朴素贝叶斯分类器是这样一种网络，它的连线是由类属性指向其他每个属性。当为分类而建立网络时，使用这种网络作为搜索起点有时还是有帮助的。可以使用K2方案来实现，强制把类属性列为排列中的第一个属性并合理地设定初始连线。

利用这些算法就可以实现一个Bayes网络，Weka就是这样实现的。

### 运用策略

本推荐系统是根据用户的个人信息以及观看历史来进行推荐的，采用基于观看历史和基于用户信息两种方法。

#### 总体流程

推荐部分：

(1) Chatbot提供一个用户ID，向数据库索取用户的基本信息以及喜好。如果喜好信息不足，转2。否则，利用基于观看历史方法（每个用户有各自的一张表）进行推荐：向数据库索要评分较好的若干节目的ID以及关键词，返回概率较高的节目ID，转2。如果数据库中没有足够的评分信息，转3。

(2) 如果用户信息不足，转3。利用基于用户信息方法（一张总表）进行推荐：利用用户信息的相似度，得到用户喜欢的节目关键词。将此关键词给数据库，数据库进行匹配，返回匹配度较高的节目ID。转5。

(3) 根据用户的观看历史直接推荐（但要去除hate的）。转4。

(4) 随机推荐。转5。

(5) 推荐近期新出的节目和最火的节目。

学习部分：

(1) Chatbot提供用户ID及观看历史、喜欢与否、评分，将观看历史及评分存入数据库，将喜好情况输入对应于该用户的表中。转2。

(2) 从数据库中得到该用户的基本信息及该节目的关键词输入总表进行学习。转3。

(3) 将用户喜欢的节目的tag保存起来。转4。

(4) 如果chatbot发现针对于该用户的推荐成功率太低，则从数据库中得到哪些属性缺少，由chatbot向用户询问。

#### 基于观看历史方法

在这种方法中，我们为每个用户进行个性化的推荐，即根据用户过去的喜好来推荐电视节目。

概率分析器的输入形式是ABC…->R，故我们输入电视节目的关键词作为ABC…，而将用户喜欢与否作为R。这样概率分析器就将用户的信息记忆了。

在进行推荐时，先向数据库索要评分较高的电视节目的关键词，将它们输入概率分析器后，我们得到用户喜欢和不喜欢的概率，将这两个概率相减，从而得到概率较高的电视节目并将其推荐。

#### 基于用户信息方法

对于一些新的用户，由于没有很多喜好信息，难以做出正确的预测。那么就采用将用户分组（虚拟）的方式根据相似的其他用户的喜好来进行预测。

Bayesian network的学习输入是用户的基本信息以及喜欢的电视节目的关键词。

预测时输入用户的基本信息，得到喜欢的电视节目关键词的概率。这是根据有相同生活背景的人品味相近的原则进行预测的。

由于Bayesian network的每一个节点都需要名词性属性，故要求用户信息都是选择性的信息，而非String类型，举个例子，职业只能在有限的几个职业中选择。

基于用户信息方法的优点是可以推荐那些和用户过去品味不太相同的节目，不局限。 这样有利于拓展用户的眼界，收看更多不同种类的节目。

## 电视节目信息提取

工具

Lucene

IKAnalyzer3.0

 Test on IKAnalyzer:

 sample code

 Test string: IK分词器Lucene Analyzer接口实现类 民生银行

 Result:

term=ik

term=分词器

term=分词

term=lucene

term=analyzer

term=接口

term=口实

term=实现

term=类

term=民生银行

term=民生

term=银行

结构

1.下载和解压缩文件

2.分析节目单

IKAnalyzer的分词功能依靠词典的支持，并且提供检测字符串中是否含有符合要求的词语的函数。

但是直接修改词典可能对分词性能有影响，在这里使用了一个关键词列表，先分词再查询关键词列表，工具会给出一个hit的判断。

比如，节目“电视剧：XXXX”，其中“电视剧”是一个分词结果，并且在关键词列表中有这个词，于是将“电视剧”作为节目属性。

同一属性的节目有多个不同的关键词，需要将关键词列表按属性分组，每个组有一个属性ID，在某一组内的任意一个关键词被找到则输出对应的属性结果。

3.网络搜索节目信息

有些节目不能根据标题来判断属性，要用网络搜索来得到。

程序目前实现的是给电视剧和电影搜索导演、演员、类型信息，通过给电影库网站发送查询请求，可以得到返回的html页面，然后对html页面使用正则表达式过滤标记语言，根据格式找到需要的信息。

4.数据输出

格式：

TV\_ID: INT

Channel: STRING

Date: DATE

Time: TIME

Title: STRING

Attribute1: INT (1或0，表示yes或no)

Attribute2: INT

…

AttributeN: INT

Director: STRING ARRAY

Actor: STRING ARRAY

Other Keyword: STRING ARRAY

# 结论

在数据挖掘中，计算机以电子化的形式存储数据，并且能自动地查询数据，或至少扩增数据。人们普遍相信，存在于数据中的模式能够被自动地找到，识别，确认并能用于预测。在最近几年，数据库急剧膨胀，正把数据挖掘带到新的商业运用技术的前沿。本系统获取的数据的总量是相当可观的，如果能从中发现某种模式，并通过该模式向用户提供有针对性的推荐，就能产生出一些效益。

同时，借助于MSN的广大用户群，我们也能够很快的取得市场，扩大用户覆盖率。

总之，相信本款基于MSN的推荐系统必将为企业带来盈利。