

魯東大學

本科毕业设计

基于贝叶斯推理的汉语双音节词语义预测研究

姓 名	郝轩廷
学 院	信息与电气工程学院
专 业	计算机科学与技术
年 级	2015
学 号	20152203026
指导教师	张志旺

2019 年 5 月 18 日

独 创 声 明

本人郑重声明：所提交的毕业论文（设计），是本人在指导老师的指导下，独立进行研究工作所取得的成果，成果不存在知识产权争议。尽我所知，除文中已经注明引用的内容外，本论文（设计）不含任何其他个人或集体已经发表或撰写过的作品成果。对本文的研究做出重要贡献的个人和集体均已在文中以明确方式标明。

此声明的法律后果由本人承担。

作者签名：_____

二〇一九年五月十八日

毕业论文（设计）使用授权声明

本人完全了解鲁东大学关于收集、保存、使用毕业论文（设计）的规定。

本人愿意按照学校要求提交论文（设计）的印刷本和电子版，同意学校保存论文（设计）的印刷本和电子版，或采用影印、数字化或其它复制手段保存论文（设计）；同意学校在不以营利为目的的前提下，建立目录检索与阅览服务系统，公布论文（设计）的部分或全部内容，允许他人依法合理使用。

（保密论文在解密后遵守此规定）

论文作者：_____

二〇一九年五月十八日

毕业设计开题报告

姓 名	郝轩廷	学院	信息与电气工程学院	年级	2015	学号	20152203026
题 目	基于贝叶斯推理的汉语双音节词语义研究						
课题来源	教师推荐	课题类别	应用实践研究				
<p>选题意义（包括科学意义和应用前景，研究概况，水平和发展趋势，列出主要参考文献目录）：</p> <p>近年来，新出现的双音节复合汉语词汇的出现为 NLP 领域带来了新挑战。对这个问题分析及提出有效的解决问题的方法变得越来越重要。贝叶斯网络起源于上世纪 80 年代对于概率的相关研究，将成为概率推理的重要领域。贝叶斯网络是基于概率论相关的理论描述随机变量之间的关系有向无环图结构，有效的降低了概率推理的时间复杂度，让概率推理能应用到更多的实际问题中。构造贝叶斯网络的方法因问题而定，在推理双音节复合词的实际问题中，我们利用汉语的 13 个相关属性构建有向无环图，进行概率推理来对这个实际问题进行分析并评估性能。</p> <p>主要参考文献：</p> <p>[1] 张连文，郭海鹏. 贝叶斯网引论[M]. 北京：科学出版社, 2006.</p> <p>[2] Lobna Boucaala,Afif Masmoudi,Faiez Gargouri,Ahmed Rebai. Improving algorithms for structure learning in Bayesian Networks using a new implicit score[G]. Expert Systems with Applications 2010(37).</p> <p>[3] Guangxia Gao, Zhiwang Zhang. Prediction of Chinese word-formation patterns using the layer-weighted semantic graph-based KFP-MCO classifier [G]. Expert Systems with Applications 2016(39).</p>							
<p>研究主要内容和预期结果（说明具体研究内容和拟解决的关键问题，预期结果和形式，如在理论上解决哪些问题及其价值，或应用的可能性及效果）：</p> <p>在推理双音节复合词的实际问题中，利用汉语的 13 个相关属性构建有向无环图，进行概率推理来对这个实际问题进行分析并评估性能。在构建贝叶斯网络时，需要大量的查找概率表，相应的具有高效的查找速度的数据结构变得重要起来。在仅仅依靠数据集自动建立贝叶斯网络的情形可能越来越常见，因为这是个 NP 难问题，目前主要采用迭代改进算法，其中的主要代表是 k2 算法。准备实现基本的语义推测，建立贝叶斯网络结构及贝叶斯网络的随机推理等相关内容。</p>							

拟采取的研究方法和技术路线（包括理论分析、计算，实验方法和步骤及其可行性论证，可能遇到的问题和解决方法，以及研究的进度与计划）：

研究方法：

使用语言：C++

技术路线：

基于算法设计中，可能对工程化不熟悉，以及相关的数学概念及算法编程技巧不熟悉，需要尽快熟悉相关的编程技巧。对于贝叶斯相关的概率论知识不够熟悉，尽快熟悉相关的概率论概念，熟悉相关公式的使用。

扩展内容中，完成 MSWT，k2，随机推理的相关算法。

研究计划：

3 月：熟悉基本的相关概念和编程技巧

4 月 1 日-4 月 15 日：完成基本的程序设计

4 月 15 日-5 月：完成扩展内容，并完成相关的收尾工作

指导教师意见（对论文选题的意义、应用性、可行性、进度与计划等内容进行评价，填写审核结果：同意开题、修改后再开题、不同意开题）：

论文选题符合专业培养目标，能够达到综合训练目标，本课题具有一定的研究价值，对于学生能力提高有利，实验设计合理，能够在规定时间完成，故同意开题。

签名：

年 月 日

学院毕业论文（设计）领导小组意见：

同 意 开 题

（签章）

年 月 日

毕业设计结题报告

姓 名	郝轩廷	学院	信息与电气工程学院	年级	2015	学号	20152203026
题 目	基于贝叶斯推理的汉语双音节词语义研究						
课题来源	教师推荐	课题类别	应用实践研究				
<p>本课题完成情况介绍（包括研究过程、实验过程、结果分析、存在的问题及应用情况等。）</p> <p>贝叶斯网络基于概率推理和因果推理，具有很大的研究意义。将贝叶斯网络应用到双音节词的语义推测中，让我认识到了处理多维数据且数据属性取值多的情况下，应用到贝叶斯网络中，处理这样的笛卡尔积问题需要基于优秀的数据结构，对这个问题进行了简要的讨论。在基于数据集自学习贝叶斯网络中，接触相关的两个算法，其中在应用 k2 算法中，思考如何消除算法使用的先验知识，想到了另一种迭代改进算法模拟退火，可以通过模拟退火不断的扰乱当前的顺序，在获得一个最佳顺序的情况下，再获得一个最优结构。变量消元算法的学习中，又接触了不定序动态规划，和关于随机推理的一些简要概念。</p>							
<p>指导教师评语：</p> <p>本文以贝叶斯网络为工具对双音节词的语意推测问题进行了分析，并以这个问题为基本点进行了纵向的扩展，但研究内容不够深入。论文语言表达流畅，格式符合要求。题目有一定难度，工作量一般，工作具有一定的实践指导意义。</p> <div>签名： 年 月 日</div>							
<p>学院毕业论文（设计）领导小组意见：</p> <div>同 意 结 题</div> <div>（公章） 年 月 日</div>							

毕业设计成绩评定表

学院（公章）：信息与电气工程学院

学号：20152203026

姓 名	郝轩廷	总成绩	
题 目	基于贝叶斯推理的汉语双音节词语义推测研究		
评 阅 人 评 语			
	签名：年 月 日		
答 辩 小 组 评 语			
	答辩成绩：组长签名：年 月 日		

注：总成绩=答辩成绩（100%）。总成绩由百分制转换为五级制，填入本表相应位置。

目 录

1 贝叶斯网络简介	2
1.1 贝叶斯网络背景	2
1.2 贝叶斯网络的基本概念	2
1.3 贝叶斯网络的前景	4
2 汉语双音节词的语义分析	4
3 需求分析	5
4 程序结构设计	7
4.1 数据预处理	7
4.2 建立贝叶斯网络	7
4.3 训练	9
4.4 推理	10
4.5 评估性能	10
5 扩展	10
5.1 不具备先验知识的情况下进行贝叶斯网络学习	10
5.1.1 MSWT 算法	11
5.1.2 k2 算法	15
5.1.3 利用模拟退火消除 k2 算法中使用的先验知识	19
5.2 使用变量消元进行精确推断	20
6 相关代码	23
6.1 二级 map 描述条件概率表	23
6.2 “条件概率树”描述条件概率表	23
6.3 哈希表描述条件概率表	24
6.4 MSWT 相关代码	24
6.5 k2 相关代码	25
6.6 模拟退火消除 k2 先验知识的相关代码	26
7 结束语	27
参考文献	28
致 谢	29

基于贝叶斯推理的汉语双音节词语义预测研究

郝轩廷

(鲁东大学信息与电气工程学院)

摘要:近年来,新出现的双音节复合汉语词汇为 NLP 领域带来了新挑战。对这个问题分析及提出有效的解决问题的方法变得越来越重要。贝叶斯网络起源于上世纪 80 年代对于概率的相关研究,是概率推理的重要领域。贝叶斯网络是基于概率论相关的理论描述随机变量之间的关系的有向无环图结构,有效的降低了概率推理的时间复杂度,让概率推理能应用到更多的实际问题中。构造贝叶斯网络的方法因问题而定,在推理双音节复合词的实际问题中,我们利用汉语的 13 个相关属性构建有向无环图,进行概率推理来对这个实际问题进行分析并评估性能。在构建贝叶斯网络时,需要大量的建立和查找概率表,相应的具有高效的查找速度的数据结构变得重要起来。同时,在仅仅依靠数据集自动建立贝叶斯网络在实际情况中可能更常见,因为这是个 NP 难题,目前主要采用迭代改进算法解决这个问题,其中的主要代表是 k2 算法等,本文实现了其中的一部分算法。

关键词: 汉语双音节词, 贝叶斯网络, 高效数据结构, 迭代改进算法, k2 算法

Bayesian network based prediction of Chinese semantic word-formation patterns

Hao Xuanting

(School of Information and Electrical Engineering, Ludong University)

Abstract: In recent years, the emergence of new disyllabic compound Chinese vocabulary has brought new challenges to the field of NLP. It is more and more important to analyze this problem and propose effective methods to solve it. Bayesian networks originated from the research on probability in the 1980s and will become an important field of probabilistic reasoning. Bayesian network is a directed acyclic graph structure based on the theory of probability to describe the relationship between random variables, which effectively reduces the time complexity of probabilistic reasoning and enables probabilistic reasoning to be applied

to more practical problems. The method of constructing bayesian network is problem dependent. In the practical problem of reasoning two-syllable compound words, we use 13 related attributes of Chinese to construct directed acyclic graph and carry out probabilistic reasoning to analyze this practical problem and evaluate its performance. When building a bayesian network, a large number of probability tables need to be looked up, and the corresponding data structure with efficient speed becomes important. It is likely to be more and more common to establish bayesian networks automatically only by relying on data sets, because this is a NP hard problem. Currently, iterative improvement algorithm is mainly adopted, in which k2 algorithm is the main representative.

Key words: compound word, Bayesian network, efficient data structure, Iterative improvement algorithm, k2 algorithm

1 贝叶斯网络简介

1.1 贝叶斯网络背景

贝叶斯网络起源于上世纪 80 年代，属于一种机器学习模型。贝叶斯网络发展起源于概率统计与图论领域，用于进行概率推理和对数据分析。贝叶斯网是一种描述随机变量关系的图论模型，其主要目的是进行概率推理。概率推理中，理论上只需要一个相关变量联合概率分布。但是，这种情况下推理复杂度相对于变量个数呈指数级的增长，所以提出贝叶斯网络的主要目的是降低推理的时间复杂度，能让概率推理应用到更多的实际问题中。

1.2 贝叶斯网络的基本概念

概率论^[1]的相关概念：

随机变量 X, Y ：

相互独立

$$P(X, Y) = P(X)P(Y) \quad (1)$$

随机变量 X, Y, Z ：

X 和 Y 在给定 Z 的取值时相互条件独立

$$P(X, Y|Z) = P(X|Z)P(Y|Z) \quad (2)$$

条件概率公式及链式规则：

随机变量 X, Y 的联合概率分布可以分解为

$$P(X, Y) = P(X)P(Y|X) \quad (3)$$

拓展到 n 个变量的联合分布时可以分解为

$$P(X_1, X_2, \dots, X_n) = P(X_1)P(X_2|X_1) \dots P(X_n|X_1, \dots, X_{n-1}) \quad (4)$$

贝叶斯公式，随机变量 H, E :

$P(H=h)$ 为先验概率， $P(H=h|E=e)$ 为后验概率:

$$P(H = h|E = e) = \frac{P(H=h)P(E = e|H = h)}{P(E=e)} \quad (5)$$

图论的相关概念:

父结点: 从结点 X 到结点 Y 有一个有向边, 称 X 是 Y 的父结点。

子结点: 上述情况中, Y 是 X 的子结点。

邻居结点: 一个结点的所有父结点和子结点称为它的邻居结点。

根结点: 没有父结点的结点。

叶结点: 没有子结点的结点。

祖先结点: 当前结点的父结点及其父结点的祖先结点, 其中根结点无祖先结点。

后代结点: 当前结点的子结点及其子结点的后代结点, 其中叶结点无后代结点。

环: 有向图中, 某结点是它自己的祖先结点, 则存在环。

有向无环图 (DAG): 不含环的有向图。

示例如下:

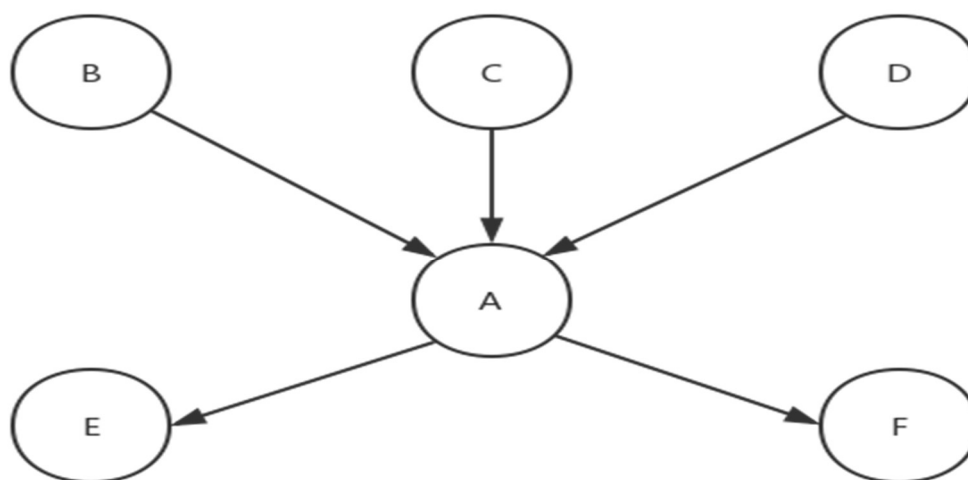


图 1 贝叶斯网络示例

图 1 中：E, F 是叶结点，A 是叶结点 E, F 的父结点，B, C, D 是 A 的父结点，是 E, F 的祖先结点。

贝叶斯网络是一个有向无环图（DAG）。有向无环图中的有向边表示了变量间的依赖和独立关系，有向边的弧头是父属性结点，弧尾是子属性结点，反应了直接依赖关系，并一定程度上反映了随机变量间的因果关系。在这样的有向无环图中也反映这样一个特性，属性间只具有直接相关的关系，即一个属性只与它的直接前驱和直接后继有关系，而其余的祖先属性与后代属性不在具有影响的关系。其中每个属性结点都有一个概率表来表示属性间关系在当前数据集中的相关概率，根属性结点为边缘分布表，非根属性为条件概率表，在“量”上反应属性间的依赖关系，用来进行概率推断及变量消元。每个属性结点的取值数量，取值不同，可以通过数据预处理来进行初始化，在对数据有了基础性的认识后进行贝叶斯网络的训练。

联合概率的分解降低了概率模型中推理计算的复杂度。使用贝叶斯网络虽没有进一步降低复杂度，但它使概率推理变得更加清晰直观。

1.3 贝叶斯网络的前景

贝叶斯网络可以应用在模拟计算生物学，生物信息学，基因调控网络，蛋白质结构，基因表达分析，医学，文件分类，信息检索，决策支持系统，工程学，游戏与法律相关，数据结合，图像处理。

2 汉语双音节词的语义分析

根据汉语构词^[2]特点，不同语义构词模式的影响因素可归纳为以下三个部分：

语素义项的类型，从《汉语词库》同义词构词规律来看，汉字及其语素义项的数量是相对稳定的。大多数新词来源于已有的汉字，它们的新含义是通过重新组合不同汉字的含义来解释的。因此，语素义是构词模式的主要影响因素，语素义类型可以从汉语词库中获得。对于构成复合词的所有汉字，根据语义范围将其意义类型分为三类，分别为大类、中类和小类。构成一个层次树在分类层次结构中，大类、中类和小类的数量分别为 12、94、1428。对于一个复合词，其含义可以表示为三层的二叉树，是图 2 中完全二叉树的子树。

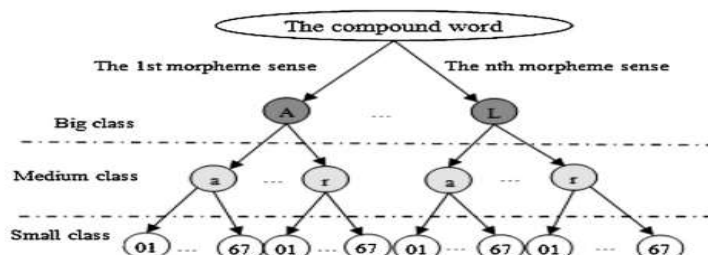


Fig. 1. The hierarchies tree of sense types for the Chinese compound word.

Table 1
The letters and their meanings of big class for sense types.

Letter	Meanings
A	“人(people/human/person)”
B	“物(thing/matter/object)”
C	“时间(time), 空间(space)”
D	“抽象事物(abstraction/abstraction entity)”
E	“特征(feature/characteristic)”
F	“动作(movement/action/motion)”
G	“心理活动(psychological activity)”
H	“活动(activity/event)”
I	“现象(appearance/phenomenon), 状态(status/state/condition)”
J	“关联(correlation/relevance/association)”
K	“助语(auxiliary language)”
L	“敬语(word of respect)”

图 2 双音节词简介

其中，大类以大写英语字母“A”~“L”标记，中类以小写英文字母“a”~“r”进行标记，而小类则以数字“01”~“67”标记。

这里我们以双音节词接受为例，第一个语素为“Hc07”其中“H”为大类，“c”为中类，“07”为小类，而第二个语素“Je14”中 大类为“J”，中类为“e”，小类为“14”。

词语的用法或意义类型，词语的意义与不同的语境之间存在着必然的联系，因此词语的语义类型也是构词模式的重要影响因素。

语素或词的词义也受语素或词的词性制约，语素和词的词性类型可以从上述汉语词的同义词典或标注的语义语料库中获得。

综上所述，将上述三类影响因素作为词汇分类的特征属性。从以上三个来源提取的对应数据构成汉语语义构词模式的数据集。

3 需求分析

利用对两个字的词素义大类(WB)，词素义中类(WM)，词素义小类(WS)，词性大类(PosB)，词性小类(PosS)共十个属性进行标记，对这两个字组成的词汇的语义大类(TWB)，语义中类(TWM)，语义小类(TWS)进行标记(取值)推断。

首要问题是构建贝叶斯网络。构建贝叶斯网络的方法有两种：一是相关的专家手动构造，二是通过数据分析构造贝叶斯网络。双音节词推断这个实际问题中，我们首要采

用相关专家构建的贝叶斯网络进行推断。

手动构建贝叶斯网络相关步骤如下：

(1) 选择一个描述相关问题的随机变量 $\{X_1, X_2, \dots, X_n\}$

(2) 设定一个变量顺序 $\langle X_1, X_2, \dots, X_n \rangle$

(3) 从一个空图出发，按照上述的变量顺序逐个将变量加入集合 A 中。

(4) 在加入变量 X_i 时，A 中变量包括 X_1, X_2, \dots, X_{i-1}

(4.1) 利用实际问题，在这些变量中选择一个尽可能小的子集 B，使假设“给定 B， X_i 与 A 中变量独立”成立。

(4.2) 从 B 中每一个结点添加一条指向 X_i 的有向边。

利用这样一个实际问题的先验知识手动构建一个具有 13 个结点的有向无环图如下。

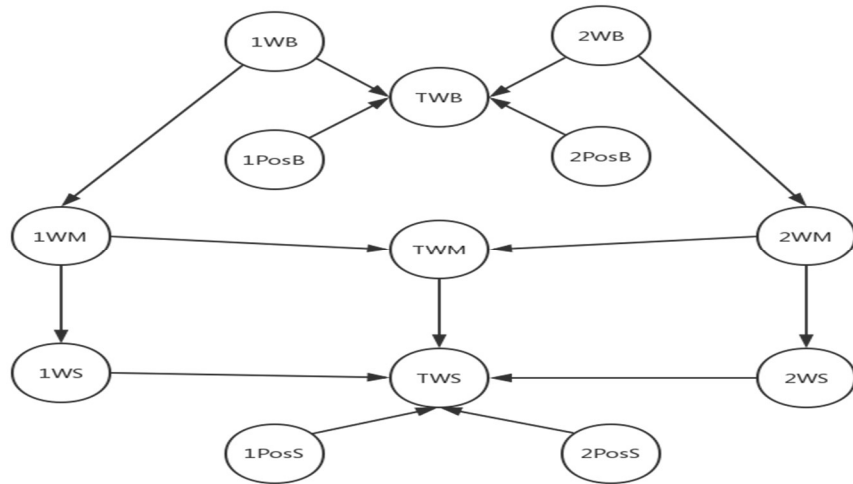


图 3 基于双音节词问题的贝叶斯网络

在双音词这个实际问题，通过先验知识构建如上的贝叶斯网络。而后的主要内容为训练相关节点的条件概率分布表和边缘概率分布表，利用其余的 10 个字相关的属性结点推测目标词汇的词汇语义大类（TWB），中类（TWM），小类（TWS）的标记再与测试集中准确标记进行对比评估模型的准确率。

其中主要通过有标记的训练集训练目标结点的条件概率表，主要使用的公式如下：

TWB 的条件概率公式：

$$P(TWB|1stWB, 2ndWB, 1stPosB, 2ndPosB) = \frac{P(TWB, 1stWB, 2ndWB, 1stPosB, 2ndPosB)}{P(1stWB, 2ndWB, 1stPosB, 2ndPosB)} \quad (6)$$

再根据变量间的相互独立的关系，上述公式可以变为

$$\begin{aligned} & P(TWB|1stWB, 2ndWB, 1stPosB, 2ndPosB) \\ &= \frac{P(TWB, 1stWB, 2ndWB, 1stPosB, 2ndPosB)}{P(1stWB)P(2ndWB)P(1stPosB)P(2ndPosB)} \end{aligned} \quad (7)$$

如上述所示，同理 TWM 的条件概率公式

$$P(TWM|TWB, 1stWM, 2ndWM) = \frac{P(TWM, TWB, 1stWM, 2ndWM)}{P(TWB)P(1stWM)P(2ndWM)} \quad (8)$$

如上述所示 TWS 的条件公式

$$\begin{aligned} & P(TWS|TWM, 1stWS, 2ndWS, 1stPosS, 2ndPosS) \\ &= \frac{P(TWS, TWM, 1stWS, 2ndWS, 1stPosS, 2ndPosS)}{P(TWM)P(1stWS)P(2ndWS)P(1stPosS)P(2ndPosS)} \end{aligned} \quad (9)$$

4 程序结构设计

程序设计主要基于 C++ 语言的面向过程思想进行设计。将程序的执行过程分为五个部分

GetDataset: 数据预处理

InitProblem: 初始化双音节词问题所需的相关变量

ConstructBN: 构建贝叶斯网络预设表项

Training: 训练边缘分布表及条件概率表

Inferencing: 推断并判断准确率

4.1 数据预处理

主要处理的问题是当前数据集中的属性的取值个数与取值的种类，利用制表符切割每一个样例字符串，获取对数据的基本认识，应为无论是联合概率分布还是条件概率分布的数量复杂度上都是笛卡尔积问题，数据的组合数量呈指数增长。双音节词实际问题中某几个属性的概率表组合数可能呈万级甚至是百万级。所以缩小数据量的大小，先对数据集进行预处理，在预处理的基础上获取了当前数据集的大背景再利用贝叶斯网络进行推断。

4.2 建立贝叶斯网络

当前双音节词问题中，贝叶斯网络在给定一个固定的贝叶斯结构后，主要的问题集中在训练条件概率表中。目前主要采用的方法是精确推理中的枚举推理。即枚举每个样

例后进行数据计数。问题又集中到如何快速的查询条件概率表，生成一个条件概率表是一个笛卡尔积连乘问题，而在这个问题中每个属性的取值量都在 10 以上，再因为属性的个数很多，所以运算量很大，这样就需要一个好的描述条件概率表的数据结构，用来快速的条件概率表中进行数据的查询计数。

这样约束查找问题的复杂度，设属性的个数为 n , 每个属性的取值个数为 N , 一个普通条件概率表的每一次查找速度应趋于 $O(N^n)$ 。

总的设计条件概率表思路，数据结构中第一级表项表示当前属性结点的条件属性节点取值，第二级表项表示当前属性结点的取值和对应的计数和概率等表项。

当前的实现方法中以一个 2 级 $\text{map}^{[3]}$ 为基础来描述条件概率表， map 在实现过程中利用红黑树保证数据间有序和查找效率，其在第一级表项上的查找速度应该为 $O(\log N^n)$, 现有情况下仍存在很多的冗余可以修改。首先在第二级表项的数据结构设计中通过对目标属性的取值进行标记顺序以哈希表的形式实现，此时二级表项的查找复杂度应趋于 $O(1)$ 。

在对第二级表项的更改的基础上，可以对第一级表项实现更多可能的修改，以降低时间复杂度，加快建表，查找的速度。

建立条件概率表生成表项时，是一个笛卡尔积^[4]问题，想到用 $\text{dfs}^{[5]}$ 生成一个树形结构。

暂时称这个描述条件概率表的数据结构为条件概率树，其本质上相对于普通的条件概率表查找速度优化到 $O(n)$, 原则上利用的算法设计思想是利用空间换取时间。每一层都是相同的条件属性，对每一层的条件属性取值设置标记（一种类似于哈希表的设计思想），这样的树一直向下延伸到目标节点将其标记为“叶子”结点，最后叶子结点的子结点为对应这条路径上属性对应的取值的概率，这样每一次的查找速度就是树的高度（层数）趋于 $O(n)$, 大大优化了查找速度。

也可以从分解二级 map 的查找时间复杂度来理解这个问题，实际上是把每一层的有序二分查找变成映射的哈希查找，这样对每一层的查找进行加速。

树形结构如图 4。

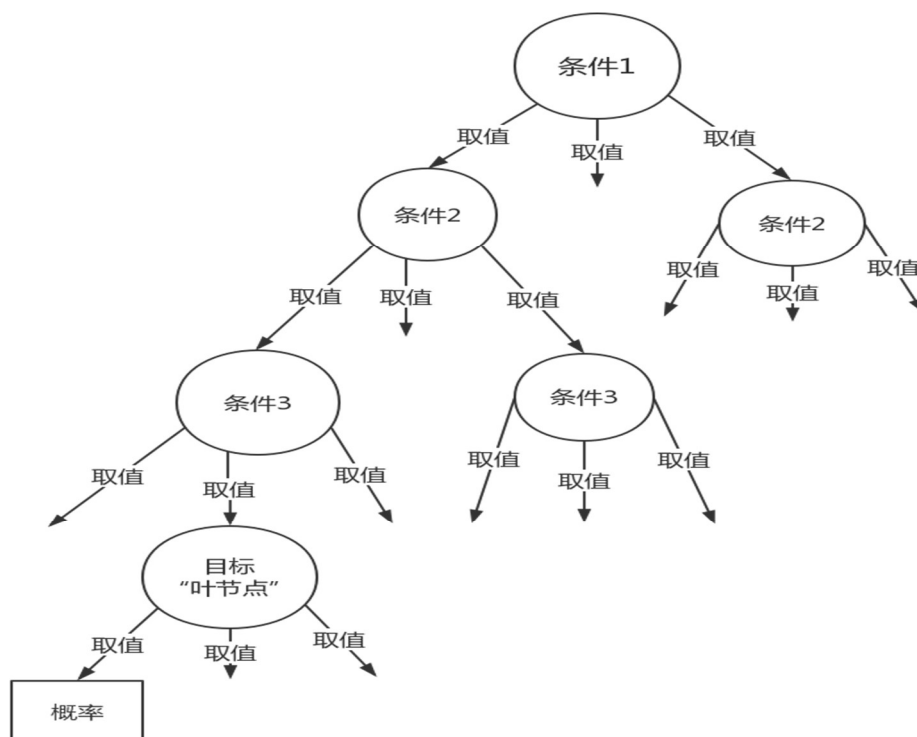


图 4 条件概率树

但是这样的数据结构中，仍可能存在查找速度过慢的问题，继续扩展可以将条件概率树上每一条的路径上的取值作为一个元素通过哈希技术融合成一张哈希表，也可以从 C++ 编程中通过 `map` 到 `unordered_map` 加上对条件概率取值融合成一个字符串来实现这个问题，这样的查找速度会大大减少，甚至总查找速度趋近于 $O(1)$ ，通过数据的测试，利用哈希技术是当前速度上的最优选择，但同时也相当于二级 `map` 结构使用了更多的空间。

在空间和时间上可以通过权衡选择最后使用的数据结构。基于当前的问题在测试后，哈希表的方式应是最优的实现概率表的方式。

4.3 训练

通过读入带标记的训练集进行概率训练，训练中利用上述的条件概率公式进行计算通过联合概率计算与条件属性的边缘概率计算最后得到的结果读入条件概率表。实现复杂的地方主要在对表项的查找与响应的读入，重点在计算的准确性上，不在于逻辑上的复杂。

在进行概率计算时有一处需要注意的地方，当所利用的条件概率公式中分母的个数

在训练集中为 0 时，不能直接去除以计数会造成零除错误，这时的条件概率应是 1 除以当前目标属性的取值个数。

4.4 推理

通过读入测试集，在相应的条件概率表表项中进行查找，以最基本的原则查找概率最大的相应项最为最后推断的结果。再与正确的标记进行对比，评估当前模型的准确率。

4.5 评估性能

当前模型的完全匹配的正确率，期望（使用的训练集，测试集在当前词库中的比例）为 8：2。

词汇大类标准取值: H 推断词汇大类取值H	词汇中类标准取值: f 推断词汇中类取值f	词汇小类标准取值: 4 推断词汇小类取值4
词汇大类标准取值: B 推断词汇大类取值B	词汇中类标准取值: a 推断词汇中类取值a	词汇小类标准取值: 13 推断词汇小类取值13
词汇大类标准取值: K 推断词汇大类取值K	词汇中类标准取值: o 推断词汇中类取值o	词汇小类标准取值: 29 推断词汇小类取值29
词汇大类标准取值: A 推断词汇大类取值A	词汇中类标准取值: c 推断词汇中类取值c	词汇小类标准取值: 3 推断词汇小类取值7
词汇大类标准取值: H 推断词汇大类取值H	词汇中类标准取值: a 推断词汇中类取值a	词汇小类标准取值: 2 推断词汇小类取值17
词汇大类标准取值: A 推断词汇大类取值A	词汇中类标准取值: j 推断词汇中类取值j	词汇小类标准取值: 1 推断词汇小类取值1
词汇大类标准取值: A 推断词汇大类取值A	词汇中类标准取值: e 推断词汇中类取值e	词汇小类标准取值: 20 推断词汇小类取值20
词汇大类标准取值: J 推断词汇大类取值J	词汇中类标准取值: b 推断词汇中类取值b	词汇小类标准取值: 2 推断词汇小类取值2
词汇大类标准取值: A 推断词汇大类取值A	词汇中类标准取值: f 推断词汇中类取值f	词汇小类标准取值: 2 推断词汇小类取值2
词汇大类标准取值: D 推断词汇大类取值D	词汇中类标准取值: j 推断词汇中类取值j	词汇小类标准取值: 1 推断词汇小类取值1
词汇大类标准取值: A 推断词汇大类取值A	词汇中类标准取值: e 推断词汇中类取值e	词汇小类标准取值: 2 推断词汇小类取值2
词汇大类标准取值: A 推断词汇大类取值A	词汇中类标准取值: f 推断词汇中类取值f	词汇小类标准取值: 5 推断词汇小类取值1
词汇大类标准取值: A 推断词汇大类取值A	词汇中类标准取值: a 推断词汇中类取值a	词汇小类标准取值: 4 推断词汇小类取值3
词汇大类标准取值: A 推断词汇大类取值A	词汇中类标准取值: f 推断词汇中类取值f	词汇小类标准取值: 2 推断词汇小类取值1
词汇大类标准取值: A 推断词汇大类取值A	词汇中类标准取值: a 推断词汇中类取值a	词汇小类标准取值: 1 推断词汇小类取值1
词汇大类标准取值: A 推断词汇大类取值A	词汇中类标准取值: a 推断词汇中类取值a	词汇小类标准取值: 8 推断词汇小类取值1
词汇大类标准取值: A 推断词汇大类取值A	词汇中类标准取值: b 推断词汇中类取值b	词汇小类标准取值: 5 推断词汇小类取值1
词汇大类标准取值: D 推断词汇大类取值D	词汇中类标准取值: a 推断词汇中类取值a	词汇小类标准取值: 26 推断词汇小类取值26
词汇大类标准取值: F 推断词汇大类取值F	词汇中类标准取值: b 推断词汇中类取值b	词汇小类标准取值: 14 推断词汇小类取值14
词汇大类标准取值: I 推断词汇大类取值I	词汇中类标准取值: a 推断词汇中类取值a	词汇小类标准取值: 2 推断词汇小类取值2
词汇大类标准取值: K 推断词汇大类取值K	词汇中类标准取值: a 推断词汇中类取值a	词汇小类标准取值: 2 推断词汇小类取值2
词汇大类标准取值: A 推断词汇大类取值A	词汇中类标准取值: a 推断词汇中类取值a	词汇小类标准取值: 8 推断词汇小类取值8
词汇大类标准取值: A 推断词汇大类取值A	词汇中类标准取值: f 推断词汇中类取值f	词汇小类标准取值: 1 推断词汇小类取值1
词汇大类标准取值: A 推断词汇大类取值A	词汇中类标准取值: j 推断词汇中类取值j	词汇小类标准取值: 11 推断词汇小类取值1
词汇大类标准取值: A 推断词汇大类取值A	词汇中类标准取值: e 推断词汇中类取值e	词汇小类标准取值: 14 推断词汇小类取值4
词汇大类标准取值: A 推断词汇大类取值A	词汇中类标准取值: h 推断词汇中类取值h	词汇小类标准取值: 1 推断词汇小类取值4
词汇大类标准取值: D 推断词汇大类取值D	词汇中类标准取值: k 推断词汇中类取值a	词汇小类标准取值: 4 推断词汇小类取值1
词汇小类标准取值: 10 推断词汇小类取值1		

 当前模型的完全准确率: 0.553191
 当前模型的期望: 0.751773
 Running Time: 0.155s
 请按任意键继续. . .

图 5 双音节词推理结果示例

当前模型推断的准确率应取决于训练集在双音节词中的代表性。下面会介绍通过当前训练集直接训练一个贝叶斯网络。

5 扩展

5.1 不具备先验知识的情况下进行贝叶斯网络学习

学习贝叶斯网络是一个 NP-难问题，这意味着没有多项式时间复杂度的算法能有效解决这个问题。而在不具备先验知识的情况下进行推理可能是现在贝叶斯网络推断中一个常见的问题，目前被广泛使用的方法一般是通过定义一个评分函数和一个有效的

搜索策略来进行启发式搜索^[6]搜索一个具有最高评分的贝叶斯网络结构。

在启发式搜索的框架确定后, 决定搜索结果的好坏往往取决于评分函数是否适应当前的问题。下面三种算法中基于不同的评分函数, 和不同的搜索思想用于从未知建立一个符合数据集的贝叶斯网络。

5.1.1 MSWT 算法

MSWT 算法的评分函数基于信息论知识。信息论是基于概率论的研究信息传输和信息处理的数学理论。下面简要介绍信息论的相关知识。

熵: 熵的定义:

$$H(X) = \sum_X P(X) \log \frac{1}{P(X)} = -\sum_X P(X) \log P(X) \quad (10)$$

熵^[7]是对随机变量的不确定性的度量。其中随机变量 X 的熵越大, 它的不确定性也越大。

条件熵: 条件熵是利用条件概率分布对熵的一个延申。设两个随机变量 X, Y, 若 Y 的取值为 y, 且条件概率为 $P(X|Y = y)$, 以此定义条件熵为

$$H(X|Y = y) = \sum_X P(X|Y = y) \log \frac{1}{P(X|Y = y)} \quad (11)$$

由熵的定义可知, 条件熵表示在已知 Y 的取值时, X 的不确定性。

互信息: 一定程度上可以将互信息理解成两个随机变量相互关联的程度, 且以“量”来衡量。

定义:

$$I(X_i; X_j) = H(X) - H(X|Y) = \sum_{x_i, x_j} P(x_i, x_j) \log \frac{P(x_i, x_j)}{P(x_i)P(x_j)} \quad (12)$$

基于信息论的相关概念, 开始最大权生成树的构造, 为当前随机变量两两设置权值, 基于随机变量间的互信息评分, 构成无向完全图。

从完全图到最大权生成树, 可以使用使用克鲁斯卡尔算法, 克鲁斯卡尔算法是一种基于边的贪心策略。

基于边贪心的 MSWT 算法的算法原理如下:

输入: 两两属性 (随机变量) 间的互信息

(1). 对于分布 $P(x)$, 对于所有 $i \neq j$, 计算联合概率分布 $P(X_i, X_j)$

(2).计算两个结点的互信息 $I(X_i, Y_j)$,并把这个计算后的结果作为连接两个属性结点的权值,要求属性结点间两两计算,构建无向完全图。

(3).计算最大权生成树:

(3.1)初始状态: n 个属性结点, 0 条边

(3.2)插入最大权重的互信息作为当前树的边

(3.3)找到下一个最大的互信息,然后继续加入到树中,要求加入后当前树中不能有环生成,如能加入删除这个互信息,否则不能加入这个边,继续收缩剩余的互信息。

(3.4)重复上述过程 3.2, 3.3 直到插入了 $n-1$ 条边(树的建立完成)

输出: 一个最大权生成树

在建树结束后,可以选择任意的结点作为根,从跟到叶子标记边的方向,构建一个有向无环图结构。

基于 80 个样例的数据集构建的贝叶斯网络如下:

```

-----生成MSWT-----
MSWT权重: 13.9526
9 13    1.92799
10 13   1.8166
8 13    1.47584
7 10    1.41053
6 9     1.34076
1 3     1.19807
2 7     1.14589
5 13    1.13223
1 9     1.09439
2 4     0.9185
9 11    0.269322
9 12    0.2225
-----观察贝叶斯网络(树)-----
输入树根的编号(1<= && <=13), 观察贝叶斯网络的结构:
1
输出
1stWB
1stPosB
1stWS
1stWM
1stPosS
2ndPosS
TWS
TWB
TWM
2ndWS
2ndWM
2ndWB
2ndPosB
.....
    
```

图 6 80 个样本产生的 MSWT

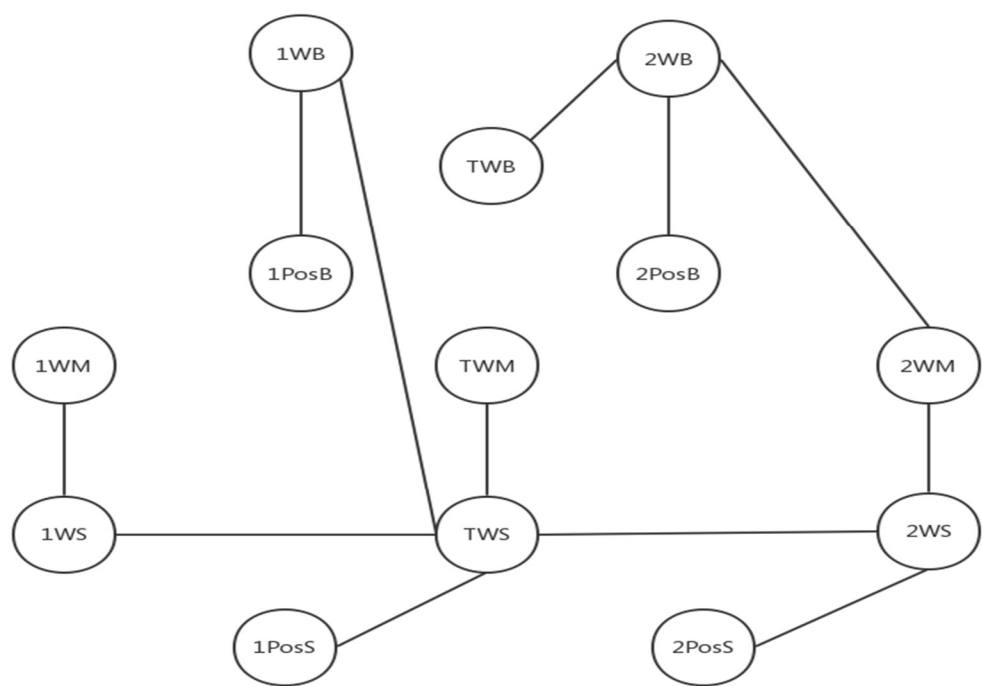


图 7 80 个样本产生的 MSWT

基于 800 个样例的数据集构建的贝叶斯网络。

生成MSWT		
MSWT权重: 10.0744		
10	13	1.327
9	13	1.28165
1	3	1.10504
7	10	1.02019
2	5	0.961365
2	7	0.913459
2	4	0.90612
7	8	0.813188
1	6	0.736001
1	5	0.680377
9	11	0.173013
10	12	0.157005
观察贝叶斯网络(树)		
输入树根的编号(1<= && <=13)，观察贝叶斯网络的结构:		
1		
输出		
1stWB		
1stPosB		
TWB		
2ndWB		
2ndPosB		
2ndWM		
TWM		
2ndWS		
2ndPosS		
TWS		
1stWS		
1stPosS		
1stWM		

图 8 800 个样本产生的 MSWT

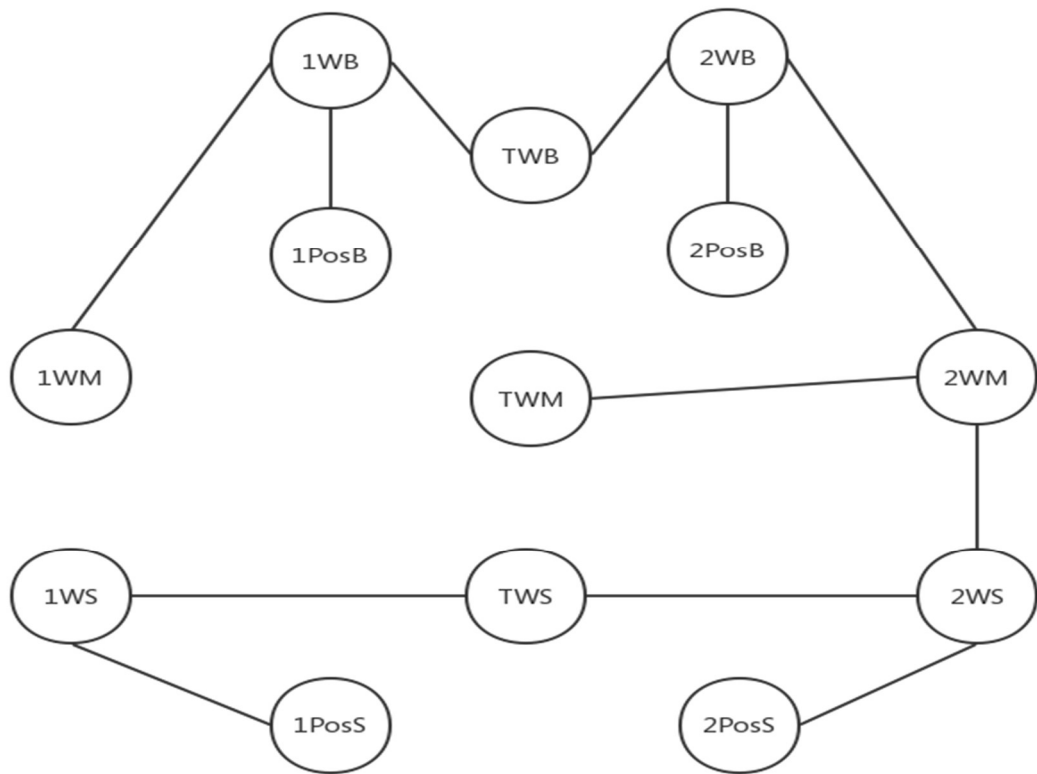


图 9 800 个样本产生的 MSWT

关于 MSWT 的性能:

基于 80 个样本数据集的推断问题

词汇大类标准取值: H	推断词汇大类取值H	词汇中类标准取值: f	推断词汇中类取值n	词汇小类标准取值: 4	推断词汇小类取值4
词汇大类标准取值: B	推断词汇大类取值D	词汇中类标准取值: a	推断词汇中类取值a	词汇小类标准取值: 13	推断词汇小类取值1
词汇大类标准取值: B	推断词汇大类取值B	词汇中类标准取值: o	推断词汇中类取值o	词汇小类标准取值: 29	推断词汇小类取值1
词汇大类标准取值: K	推断词汇大类取值E	词汇中类标准取值: c	推断词汇中类取值a	词汇小类标准取值: 13	推断词汇小类取值22
词汇大类标准取值: K	推断词汇大类取值E	词汇中类标准取值: a	推断词汇中类取值j	词汇小类标准取值: 3	推断词汇小类取值22
词汇大类标准取值: A	推断词汇大类取值A	词汇中类标准取值: a	推断词汇中类取值e	词汇小类标准取值: 2	推断词汇小类取值17
词汇大类标准取值: H	推断词汇大类取值D	词汇中类标准取值: j	推断词汇中类取值h	词汇小类标准取值: 1	推断词汇小类取值1
词汇大类标准取值: H	推断词汇大类取值D	词汇中类标准取值: j	推断词汇中类取值j	词汇小类标准取值: 20	推断词汇小类取值8
词汇大类标准取值: A	推断词汇大类取值A	词汇中类标准取值: e	推断词汇中类取值e	词汇小类标准取值: 2	推断词汇小类取值17
词汇大类标准取值: J	推断词汇大类取值D	词汇中类标准取值: e	推断词汇中类取值h	词汇小类标准取值: 2	推断词汇小类取值1
词汇大类标准取值: A	推断词汇大类取值A	词汇中类标准取值: b	推断词汇中类取值e	词汇小类标准取值: 1	推断词汇小类取值1
词汇大类标准取值: A	推断词汇大类取值A	词汇中类标准取值: f	推断词汇中类取值n	词汇小类标准取值: 2	推断词汇小类取值1
词汇大类标准取值: D	推断词汇大类取值D	词汇中类标准取值: j	推断词汇中类取值n	词汇小类标准取值: 4	推断词汇小类取值3
词汇大类标准取值: A	推断词汇大类取值A	词汇中类标准取值: e	推断词汇中类取值e	词汇小类标准取值: 2	推断词汇小类取值17
词汇大类标准取值: A	推断词汇大类取值A	词汇中类标准取值: a	推断词汇中类取值h	词汇小类标准取值: 1	推断词汇小类取值1
词汇大类标准取值: A	推断词汇大类取值A	词汇中类标准取值: f	推断词汇中类取值k	词汇小类标准取值: 8	推断词汇小类取值1
词汇大类标准取值: A	推断词汇大类取值A	词汇中类标准取值: a	推断词汇中类取值n	词汇小类标准取值: 5	推断词汇小类取值4
词汇大类标准取值: E	推断词汇大类取值E	词汇中类标准取值: b	推断词汇中类取值b	词汇小类标准取值: 26	推断词汇小类取值26
词汇大类标准取值: D	推断词汇大类取值D	词汇中类标准取值: a	推断词汇中类取值a	词汇小类标准取值: 14	推断词汇小类取值14
词汇大类标准取值: F	推断词汇大类取值H	词汇中类标准取值: b	推断词汇中类取值e	词汇小类标准取值: 2	推断词汇小类取值17
词汇大类标准取值: I	推断词汇大类取值E	词汇中类标准取值: a	推断词汇中类取值e	词汇小类标准取值: 2	推断词汇小类取值17
词汇大类标准取值: K	推断词汇大类取值E	词汇中类标准取值: c	推断词汇中类取值k	词汇小类标准取值: 8	推断词汇小类取值8
词汇大类标准取值: A	推断词汇大类取值A	词汇中类标准取值: a	推断词汇中类取值h	词汇小类标准取值: 1	推断词汇小类取值1
词汇大类标准取值: A	推断词汇大类取值A	词汇中类标准取值: f	推断词汇中类取值f	词汇小类标准取值: 11	推断词汇小类取值1
词汇大类标准取值: A	推断词汇大类取值A	词汇中类标准取值: j	推断词汇中类取值a	词汇小类标准取值: 14	推断词汇小类取值4
词汇大类标准取值: A	推断词汇大类取值A	词汇中类标准取值: e	推断词汇中类取值h	词汇小类标准取值: 1	推断词汇小类取值4
词汇大类标准取值: A	推断词汇大类取值A	词汇中类标准取值: h	推断词汇中类取值n	词汇小类标准取值: 4	推断词汇小类取值26
词汇大类标准取值: D	推断词汇大类取值D	词汇中类标准取值: k	推断词汇中类取值a	词汇小类标准取值: 10	推断词汇小类取值3

 当前模型的完全准确率: 0.234043
 当前模型的期望: 0.574468
 Running Time: 0.253s
 请按任意键继续. . .

图 10 基于 80 个样本产生的 MSWT 推理结果示例

可以看到与上文中的手动建立的贝叶斯网络模型准确率仍有差距。因为当前形成的

结构本质上是一个树形结构，每一个结点的父结点只有一个，不能更加贴切的反应属性间的关系。

5.1.2 k2 算法

k2 算法^[8]本质上是一个贪心的爬山搜索。k2 算法的评分基于贝叶斯网络模型关于数据集的概率。k2 算法的效率还是要基于评分系统的适应性。下面介绍一些基本概念。

相关变量：

D: 数据集。

Z: n 个离散属性的集合。

r_i : Z 中 X_i 属性有 r_i 中可能的取值。

B: 包含全部属性的贝叶斯网络。

π_i : 属性 X_i 的父属性结点的集合，共有 q_i 种实例化。

W_{ij} : π_i 所构成的组合中的第 j 种实例化。

N_{ijk} : 数据集 D 中 X_i 取值为 V_{ik} 并且 π_i 被实例化为 W_{ij} , 其中 $N_{ijk} = \sum_{k=1}^{r_i} N_{ijk}$ 。

不同的贝叶斯网络基于数据集的概率公式如下：

$$\frac{P(B_i|D)}{P(B_j|D)} = \frac{P(B_i,D)/P(D)}{P(B_j,D)/P(D)} = \frac{P(B_i,D)}{P(B_j,D)} \quad (13)$$

这样只需要计算 $P(B, D)$ 即可判断

$P(B, D)$ 的相关计算公式如下：

$$P(B, D) = \int_{\theta} P(D|\theta, B) P(\theta|B) P(B) d\theta$$

$$P(B, D) = P(B) \prod_{i=1}^n \prod_{j=1}^{q_i} \frac{(r_i-1)!}{(N_{ij}+r_i-1)!} \prod_{k=1}^{r_i} N_{ijk}! \quad (14)$$

第一个连乘符号通过 i 遍历每一个属性（随机变量），n 为属性的个数。

第二个连乘符号通过 j 遍历当前属性 X_i 所有可能的父属性组合实例， q_i 为父属性组合的数量。

第三个连乘符号通过 k 遍历当前属性 X_i 可能的取值， r_i 为当前属性的取值数量。

设不同的贝叶斯网络结构服从均匀分布，我们想以此找到最符合当前数据集的贝叶斯网络结构，即函数如下：

$$\max[P(B, D)] = c \prod_{i=1}^n \max_{\pi_i} [\prod_{j=1}^{q_i} \frac{(r_i-1)!}{(N_{ij}+r_i-1)!} \prod_{k=1}^{r_i} N_{ijk}!] \quad (15)$$

基于贪心的策略，可以将评分函数设置成

$$g(i, \pi_i) = \prod_{j=1}^{q_i} \frac{(r_i-1)!}{(N_{ij}+r_i-1)!} \prod_{k=1}^{r_i} N_{ijk}! \quad (16)$$

k2 算法基于贪心的策略，算法的核心思想是想通过对每一部分取最大值，来获得最后的最大值。

下面简要介绍 k2 算法的流程：

输入：一组属性，一个包含属性的顺序，一个父属性结点数量的上限，数据集

for(对所有属性结点进行搜索)

 设置当前属性的父属性结点集合为空;

 对当前的集合进行评分并赋值给 P_{old} ;

 设置循环标志位 OKToProceed 为 true;

 while(当 OKToProceed 为真&&当前父属性的数量<父属性结点的上限)

 让 Z 作为 Pred(Xi)(为顺序中 Xi 的前驱)- π_i 的属性，并在 π_i 中加入 Z;

 对当前的集合进行评分并赋值给 P_{new}

 if($P_{new} > P_{old}$)

$P_{old} = P_{new}$;

$\pi_i = \pi_i \cup \{Z\}$;

 else

 OKToProceed 设为 false;

 输出当前结点的父属性结点集

输出：每一个属性结点的父属性结点

对于评分函数，可以加上 \log （对数函数）以加快运行速度。

k2 算法中，for 循环对每一个属性进行遍历实现了贪心功能，其中 while 循环实现了爬山搜索。爬山搜索是一种迭代改进算法，它总是基于当前问题的最优解进行搜索，所以有可能陷入最优解。

对于 80 个样例的训练集的训练结果如下：

```

1 old评分 -1.06111e+09
-----
最后评分: -1.06111e+09

1stWB的父结点

*****
2 old评分 -1.06111e+09
2 new评分 -3732.11
-----
最后评分: -3732.11

2ndWB的父结点
1stWB

*****
3 old评分 -1.06111e+09
3 new评分 -nan(ind)
-----
最后评分: -1.06111e+09

1stPosB的父结点

*****
4 old评分 -1.06111e+09
4 new评分 -4957.51
4 new评分 -1078.86
4 new评分 -870.603
-----
最后评分: -870.603

2ndPosB的父结点
1stWB          2ndWB          1stPosB

*****

```

图 11 利用 k2 算法基于 80 个样本产生的贝叶斯网


```

5 new评分 -4130.81
5 new评分 -1127.45
5 new评分 -866.955
-----
最后评分: -866.955

TWB的父结点
1stWB          2ndWB          1stPosB
*****
6 old评分 -1.06111e+09
6 new评分 -3489.46
6 new评分 -574.168
6 new评分 -410.421
-----
最后评分: -410.421

1stWM的父结点
1stWB          2ndWB          1stPosB
*****
7 old评分 -1.06111e+09
7 new评分 -2633.15
7 new评分 -619.235
7 new评分 -410.975
-----
最后评分: -410.975

2ndWM的父结点
1stWB          2ndWB          1stPosB
*****
8 old评分 -1.06111e+09
8 new评分 -2432.76
8 new评分 -395.926
8 new评分 -265.864
-----
最后评分: -265.864

TWM的父结点
1stWB          2ndWB          1stPosB
*****

```

图 12 利用 k2 算法基于 80 个样本产生的贝叶斯网络

```

9 old评分 -1.06111e+09
9 new评分 -2216.11
9 new评分 -415.063
9 new评分 -326.548
-----
最后评分: -326.548

1stWS的父结点
1stWB          2ndWB          1stPosB
*****
10 old评分 -1.06111e+09
10 new评分 -3315.25
10 new评分 -702.411
10 new评分 -476.606
-----
最后评分: -476.606

2ndWS的父结点
1stWB          2ndWB          1stPosB
*****
11 old评分 -1.06111e+09
11 new评分 -4927.99
11 new评分 -667.111
11 new评分 -483.926
-----
最后评分: -483.926

1stPosS的父结点
1stWB          2ndWB          1stPosB
*****
12 old评分 -1.06111e+09
12 new评分 -nan(ind)
-----
最后评分: -1.06111e+09

2ndPosS的父结点
*****
13 old评分 -1.06111e+09
13 new评分 -2225.25
13 new评分 -348.695
13 new评分 -244.433
-----
最后评分: -244.433

```

图 13 利用 k2 算法基于 80 个样本产生的贝叶斯网络

5.1.3 利用模拟退火消除 k2 算法中使用的先验知识

k2 算法中仍使用了一定的先验知识，需要为属性结点预设一个顺序，而在实际问题中，如 MSWT 算法应用的出发点，我们可能对先验知识一无所知，这时我们可以在 k2 算法外层套上一个模拟退火算法，通过不断的扰乱当前的顺序再对整体的贝叶斯网络的评估来获取一个更加贴近最优情况的贝叶斯网络。

退火^[9]是一种冶金技术，将金属加热后逐步冷却，进行炼金。模拟退火算法是一种迭代改进算法，能让问题的解有机会逃脱陷入局部最优解的情况去搜索全局最优解。模拟退火的循环体中，通过扰动当前的解，产生一个邻接候选解，然后计算邻接解与当前解的能量差，如果能量差的正负符合想要改进当前解的方向，就设置新解为当前解。如果新解不优于当前解，就以一个当前温度与能量差成比例的概率判断是否接受这个邻接解为当前解，这个接受的可能性会随着温度的降低越来越小。这样初始时有较大可能会跳出局部最优解，而随着算法迭代次数的增加，相应接受较差解的可能性变小，算法的结果也会逐渐收敛。

模拟退火算法流程如下：

输入：一个起始温度，一个下降率

获取初始解，并获得当前解的值；

while(温度>0)

 扰动当前解获得邻接解，获得这个扰动后邻接解的值；

 计算邻接解与当前解的能量差；

 if(能量差>0)

 当前解=邻接解

 else

 设置当前概率 $P = \exp(-\frac{\text{能量差}}{\text{当前温度}})$

 if($P > \text{RANDOM}(0..1)$)

 当前解=邻接解

输出：一个近似最优解

基于 80 个样例数据集，进行 10 次模拟退火的结果（运算复杂度较高，不能完全进

行模拟退火)。

```

////////////////////////////////////
////////////////////////////////////最终输出////////////////////////////////////
////////////////////////////////////
1stWB 父属性结点:

*****
2ndWB 父属性结点:
1stWB      1stPosB      TWB
*****
1stPosB 父属性结点:

*****
2ndPosB 父属性结点:
1stWB      2ndWB      1stPosB
*****
TWB 父属性结点:

*****
1stWM 父属性结点:
1stWB      2ndWB      1stPosB
*****
2ndWM 父属性结点:
1stWB      2ndWB      1stPosB
*****
TWM 父属性结点:
1stWB      1stPosB      TWB
*****
1stWS 父属性结点:
TWB
*****
2ndWS 父属性结点:
1stWB      2ndWB      1stPosB
*****
1stPosS 父属性结点:

*****
2ndPosS 父属性结点:

*****
TWS 父属性结点:
TWB      1stWS      1stPosS
*****

```

图 14 利用模拟退火消除 k2 所需先验知识产生的贝叶斯网络

5.2 使用变量消元进行精确推断

利用变量消元算法^[10]进行精确推理的使用场景, 不同与双音节词的推理。双音节词的实际推理中是从父属性结点推理到子属性结点。而变量消元算法的使用是为了对贝叶斯网络中不同的证据变量, 不同的查询变量进行推断, 例如从结果到原因, 从原因到结果, 同一结果的不同原因。

变量消元的准备工作是要做好消元顺序的准备, 好的消元顺序会大大降低计算的时间复杂度。

一个相关的例子:



图 15 贝叶斯网络示例

如果计算

$$P(D) = \sum_{A,B,C} P(A, B, C, D) = \sum_{A,B,C} P(A)P(B|A)P(C|B)P(D|C) \quad (17)$$

假设所有的属性取值数量都为 2，按乘法的顺序计算， $P(A)$ 与 $P(B|A)$ 计算需要 4 次乘法，再与 $P(C|B)$ 计算需要 8 次乘法，再与 $P(D|C)$ 计算需要 16 次乘法。总共需要 28 次乘法，14 次加法。

为了降低时间复杂度，在上式的计算时，可以看到 $P(A)$ 和 $P(B|A)$ 与变量 A 相关，与 B 有关的是 $P(C|B)$ 和 $P(D|C)$ ，因此可以修改计算顺序如下。

$$P(D) = \sum_C P(D|C) \sum_B P(C|B) \sum_A P(A)P(B|A) \quad (18)$$

在这样的运算顺序下 $P(A)$ 和 $P(B|A)$ 相乘需要 4 次乘法，消去 A 需要 2 次加法，再与 $P(C|B)$ 相乘需要 4 次乘法，消去 B 需要 2 次加法，再与 $P(D|C)$ 相乘需要 4 次乘法，消去 C 需要 2 次加法，这样总共需要 12 次乘法，6 次加法。

对联合分布进行分解能降低复杂度，是因为可以通过合理的顺序进行计算局部化，只计算与当前属性相关的因子，这种方法降低的复杂度可能是指数级的。

本文中我采用了最大势搜索的方法获取消元顺序，另一种相关方法是最小缺边搜索。

最大势搜索也是一种贪心策略，算法流程类似于 Prim 算法，算法的基本思想如下：

(1)第一步任选一个属性结点，编号为 n。

(2)第 i 步中，选择相邻结点中被标号最多的未标号结点作为当前编号结点，编号为 n-i+1。

相关算法的流程如下：

输入：一个开始结点

for(所有结点)

 获取当前相邻结点中被标号最多的未标号结点

 标记这个结点，并入栈

for(与这个标记结点的相邻未标号结点)

更新它们的相邻结点数量

输出：一个消元顺序

注意在消元顺序的获得中，不应该消去证据变量和查询变量，应当通过一些编程手段获取正确的消元顺序。

关于变量消元的两个基本概念。分解：设 $F(X_1, X_2, \dots, X_n)$ 是变量 $\{X_1, X_2, \dots, X_n\}$ 的一个函数， $A=\{f_1, f_2, \dots, f_n\}$ 是一组函数，其中每个函数的变量是上述变量集合的一个子集。且 $F = \prod_{i=1}^m f_i$ ，则 A 是 F 的一个分解。消元：如 $G(X_2, \dots, X_n) = \sum_{X_1} F(X_1, X_2, \dots, X_n)$ 称为消元。

从上述内容可以得知，进行变量推理主要进行的是消元。如果直接从原函数中直接进行变量消元，计算的时间复杂度会相对于变量个数呈指数级增长。但是从分解中进行变量消元的时间复杂度往往比直接消元的时间复杂度小的多。

下面介绍变量消元的 VE 算法：

输入：一个贝叶斯网络，证据变量 E ，证据变量的取值 e ，查询变量 Q ，消元顺序 S

A =从贝叶斯网络获得一个联合概率的分解；

A 中设置证据变量的取值为 e

while($S \neq \emptyset$)

Z = S 中排在最前的变量，在 S 中删除 Z

A =Elim 函数消元重新计算

A 中所有因子相乘，得到查询变量 Q 的函数 $h(Q)$

输出： $\frac{h(Q)}{\sum_Q h(Q)}$

其中 Elim 函数如下：

输入：函数集合 A ，消元变量 Z

从 A 中提取且删除所有变量包括 Z 的函数因子，设这个函数因子组合为 f

$g=f$ 中函数因子的连乘

对 g 中变量 Z 进行消元

将 g 放入 A 中

输出： A

变量消元的思想还可以应用到求最大后验假设问题中，最大后验假设问题是一个不定序的动态规划问题。推理问题中，因为变量消元需要消耗很多空间，时间，更多的是随机推理方法。

6 相关代码

6.1 二级 map 描述条件概率表

```
//二级表项
struct CTableI{
    int cnt;
    double pro;
};
//二级表头
struct CTableH{
    int sumcnt;
    map<string, CTableI> curCTableI;
};
//条件概率表
map<vector<string>, CTableH> CTable;
```

6.2 “条件概率树” 描述条件概率表

```
struct Node{
    bool flag;//标记是否为叶节点（叶节点为路径上的目标结点）;判断是不是概率结点
    int value;//上一层分支的取值
    string name;//结点名称（属性名称）
    int id;//结点标号（属性标号）
    int conpro;//概率结点存储条件概率的容器
    int pro;//概率结点存储最终概率的容器
    vector<int> path;//节点存储的路径上的结点属性的取值
```

```
vector<Node> child;//孩子结点  
}
```

6.3 哈希表描述条件概率表

```
//string 中利用“编码”技术  
map<string,CPtableH> CPtable;
```

6.4 MSWT 相关代码

基于 C++ 的 MSWT 算法实现：

//并查集压缩路径+查找

```
int findfather(int u)  
{  
    if (father[u] == u)  
        return u;  
    else  
        return father[u] = findfather(father[u]);  
}
```

//kruskal获取最大生成树

```
void kruskal_MSWT()  
{  
    double ans = 0;sort(GV.begin(), GV.end(), cmp);  
    //并查集初始化  
    for (int id = 1; id <= 13; id++)  
        father[id] = id;  
    int curedgenum = 0;  
  
    for (int i = 0; i < GV.size(); i++)  
    {
```

```

int fa = findfather(GV[i].u);
int fb = findfather(GV[i].v);
if (fa != fb)
{
    ans += GV[i].wei;
    father[fa] = fb;
    MSWT.push_back(GV[i]);
    curedgenum++;

    if (curedgenum == n - 1)
        break;
}
}
}

```

6.5 k2 相关代码

```

void k2(int limit)
{
    for (int id = 1; id <= 13; id++)
    {
        Pa[id].clear();
        //父属性结点为空的特殊处理
        cout << id << " old评分 ";
        //double Pold = getscore(id, Pa[id]);
        double Pold = -INF;
        cout << Pold << endl;

        bool OKToProcced = true;
    }
}

```



```

while (OKToProcced == true && Pa[id].size() < limit && !Pre[id].empty())
{
    set<int> curPa = Pa[id];
    set<int>::iterator it = Pre[id].begin();
    curPa.insert(*it);
    Pre[id].erase(it);
    double Pnew = getscore(id, curPa);
    if (Pnew > Pold)
    {
        Pold = Pnew;
        Pa[id] = curPa;
    }
    else
        OKToProcced = false;
}
}
}

```

6.6 模拟退火消除 k2 先验知识的相关代码

```

void simulated_annealing(int limit)
{
    double MAXP = k2(limit, order);
    double t = T;
    double curP;
    double deltaE;

    int cnt = 1;
    while (t > 0)
    {

```

```
    if (cnt > 30)
        break;
    vector<int> curorder = swap_order();
    curP = k2(limit, curorder);
    deltaE = curP - MAXP;

    if (deltaE > 0)
    {
        MAXP = curP;
        order = curorder;
    }
    else
    {
        double p = exp(-deltaE / t);
        double r = ((double)(rand() % 100) / 100);

        if (p > r)
        {
            MAXP = curP;
            order = curorder;
        }
    }
    t *= a;
}
}
```

7 结束语

贝叶斯网络基于概率推理和因果推理，具有很大的研究意义，贝叶斯网络的知识宏大，在本文中，只讨论了其中有限的一小部分。将贝叶斯网络应用到双音节词的语义推

测中，让我认识到了处理多维数据且数据属性取值多的情况下，应用到贝叶斯网络中，处理这样的笛卡尔积问题需要基于优秀的数据结构，我对这个问题进行了简单的思考，但是限于知识匮乏，没能有什么真正具有意义解决方案，但是，我相信在对这个问题有了基础认识后，在以后的研究中，我会更上一层楼，相信能对这个问题提出大量的改进。在基于数据集自学习贝叶斯网络中，接触相关的两个算法，同时在应用 k2 算法中，思考如何消除算法使用的先验知识，想到了另一种迭代改进算法模拟退火，可以通过模拟退火不断的扰乱当前的顺序，在获得一个最佳顺序的情况下，再获得一个最优结构。变量消元算法的学习中，又接触了不定序动态规划，和关于随机推理的一些简要概念。

贝叶斯网络及贝叶斯推理的相关知识既有广度也有深度，希望我能以此为开始，继续对贝叶斯网络的相关问题进行研究。

参考文献

- [1] Cameron Davidson-Pilon. 贝叶斯方法[M]. 北京：人民邮电出版社, 2017.
- [2] Guangxia Gao, Zhiwang Zhang. Prediction of Chinese word-formation patterns using the layer-weighted semantic graph-based KFP-MCO classifier [G]. Expert Systems with Applications 2016(39).
- [3] 雷朋，宋丽华，张小峰. 面向对象 C++程序设计[M] 北京：清华大学出版社, 2014.
- [4] 张小峰，赵永升，杨洪勇，李秀芳. 离散数学[M]. 北京：清华大学出版社, 2016.
- [5] 王晓东. 计算机算法设计与分析[M]. 北京：电子工业出版社, 2012.
- [6] M.Tim Jones. Artificial Intelligence A Systems Appraoch[M]. 北京：电子工业出版社, 2010.
- [7] 张连文，郭海鹏. 贝叶斯网引论[M]. 北京：科学出版社, 2006.
- [8] Lobna Boucaala,Afif Masmoudi,Faiez Gargouri,Ahmed Rebai. Improving algorithms for structure learning in Bayesian Networks using a new implicit score[G]. Expert Systems with Applications 2010(37).
- [9] 胡凡，曾磊. 算法笔记[M]. 北京：机械工业出版社, 2016.
- [10] 周志华. 机器学习[M]. 北京：清华大学出版社, 2016.

致 谢

首次接触机器学习的相关编程，非常感谢我的指导老师张志旺老师对我的不厌其烦的指导。张志旺老师治学严谨，为人亲和。毕业设计中，我多次打扰老师，但是老师一直耐心、尽心辅导我进行毕业设计，在我困难时，对我进行鼓励和支持，知识上，授人以鱼且授人以渔。学生愚钝，没有老师的指导，没有我今天对贝叶斯网络相关知识的浅薄理解。同时我还要感谢我的同学，在我毕业设计时对我提供帮助，希望未来的学习工作中能与大学同学们继续共同进步。

4 年校园生活转瞬即逝，希望我以我的大学老师，同学为榜样，在未来的学习生活中继续前行。感谢我的所有的大学老师及我的大学同学，感谢 4 年来的教育与支持。再次感谢我的老师与同学对我的帮助！