

基于贝叶斯网研究自动作曲中音高的表示和推理

翁诗杰 李维华 丁海燕

(云南大学信息学院计算机科学与技术系 昆明 650091)

摘 要 自动作曲或称算法作曲是利用计算机进行自动或半自动的音乐创作过程。算法作曲的关键之一是生成音高。然而,不确定性是音乐本身固有的特征。贝叶斯网是不确定性知识的表示和推理的典型工具,已经成功应用到很多领域。在 MIDI 格式的基础上,利用贝叶斯网在算法作曲中生成音高,首先建立一个关于音高的贝叶斯网模型并基于此模型建立知识库。其次,基于贝叶斯网对音高进行推理,生成给定节拍处的每一个音的音高。实验表明,所提出的音高推理方法是可行的。

关键词 算法作曲,计算机音乐,MIDI 音乐系统,贝叶斯网,推理

中图分类号 TP181 **文献标识码** A

Representing and Reasoning Pitches in Algorithmic Composition Based on Bayesian Networks

WENG Shi-jie LI Wei-hua DING Hai-yan

(Department of Computer Science and Engineering, School of Information Science and Engineering, Yunnan University, 650091 Kunming, China)

Abstract Algorithmic composition is the partial or total automatic process of music composition by a computer. One of the challenges in algorithmic composition is to create pitches. However, uncertainty is an intrinsic feature of music. Bayesian network (BN) is an effective and popular framework for representing and reasoning knowledge under uncertainty, and BNs have been successfully applied to a variety of problems. Based on MIDI format to create pitches in algorithmic composition, we firstly built a model about pitches with BNs and a knowledge base based on the model. Moreover, based on bayesian inference, the pitch of every note at each pat could be created. A preliminary experiment demonstrates empirically that this method for pitch inference is feasible.

Keywords Algorithmic composition, Computer music, MIDI musical system, Bayesian network, Inference

1 引言

自动作曲或称算法作曲^[1]是试图使用某个形式化的过程,研究如何使人在利用计算机进行音乐创作时的介入程度达到最小。然而,自动生成符合人们的听觉审美又充满变化的音乐是一个复杂而困难的过程。Midi 音乐格式的出现使得音乐的计算机存储、合成以及自动创作成为可能。有许多学者在自动作曲方面进行了大量研究并取得了阶段性成果。目前,在自动作曲方面主要使用的方法有马尔科夫链^[2-4]、规则知识库^[5-8]、遗传算法^[9-11]、神经网络^[12]等。

马尔科夫模型^[2-4]二维转换表使用像函数一样的一个转换表来依次选择音符,其自变量是当前的音符,而函数值则是下一个要出现音符的可能性。但马尔科夫模型用二维的转换表来依次选择音符,在自动作曲中会导致由其生成的旋律线条比较随意,使得旋律的含义不够明确。其次,当维数增多时,转换表的大小呈指数增长,这就使得它难以管理。

此外,也有人基于规则的知识库^[5-8]系统来创作音乐,但是这种系统通常都包含大量的规则,例如 EbcioIgn 专门建立了一种基于规则的专家系统,大约有 350 条规则。虽然大多数的自动作曲系统都基于此方法,但是,在音乐领域,知识引

导机制的建立既困难又费时。其次,构造知识库系统的“音乐专家”往往很难找到。最后,收集所有的“例外规则”以及这些规则的先决条件,往往是非常复杂的。

基于遗传算法^[9-11]的算法作曲主要是构造适应函数,以此来评估及选择系统生成的旋律问题。但是,适应函数其实是很难准确定义的。于是,出现了让人代替适应函数来直接评估染色体的交互式的遗传算法。但所有和交互式的遗传算法 IGA 相关的方法主要都有主观性(人的评估决定了最后的结果)和不有效性(用户必须听到所有可能潜在的解,才能作出具有普遍性的评估)这两个弊端,这被称为适应函数的瓶颈问题。

人工神经网络^[12]已广泛地使用在音乐应用系统中,并且已取得相当大的成功。递归神经网络能够成功地获取一个旋律经过句的表层结构,并以这样获取的知识为基础,产生出新的旋律。但是,所生成的旋律缺乏音乐的全局连贯性,即它无法获取较高级的音乐特征。

贝叶斯网是目前不确定性知识表示和推理的重要工具^[13],已经广泛应用到人工智能的大部分领域并为其提供直观、有效、可靠的方法。然而,目前基于贝叶斯网研究自动作曲主要集中在音乐的实时系统^[12,14,15]。本文所做的工作是借

翁诗杰(1988-),男,硕士生,主要研究方向为数据与知识工程,E-mail:cqzwqwsj@163.com;李维华(1977-),女,副教授,主要研究方向为数据与知识工程;丁海燕(1974-),女,讲师,主要研究方向为数据库及相关技术。

助贝叶斯网对不确定性知识的表示和推理的优势,对音高进行建模和分析,并生成合理的音高。采用贝叶斯网来表达处理音乐,可以利用其擅长的不确定推理,生成符合人们的听觉审美又充满变化的音乐。理论上可以自由地在前后任意一个音符上提取证据并推理。这样既可解决二维转换表在生成音符的随意性问题,又可解决 n 维转换表难以管理的问题。其次,只需结合音乐规则,建立简单的知识库,就可为机器学习提供依据。从而可以避免规则系统创建不必要的繁琐的、高代价的知识规则库。最后,也不用像遗传算法一样去主观地寻找适应函数,本文根据提取的证据,为推理提供数据,更具有客观性。理论及实验证明,将贝叶斯网应用到音乐的推理上是可行的。

2 计算机音乐的模型创建

为了方便计算机对音乐进行处理以及设定计算机的推理规则,需要分析音乐库中音乐的特征,如调式、音符分布等。音乐是一个听觉的概念,为了使计算机更好地分析它,我们将对音乐进行形式化的定义。

2.1 建立音乐的先验知识库

在钢琴中一共有 88 个键,对于人声的音高,通常是在 E^2 至 c^3 之间,一共有 40 个半音音符。对应的是以首调唱名法为基础的表示方法,以前面冠以负号表示低音,前面冠以 1 表示高音。考虑到可歌唱性及人的音域范围,这里取最低音为 -3,最高音为 21,用所选取的此范围内的 20 个自然音级个体组成的集合作为音高论域,即:

$P_h = \{-3, -4, -5, -6, -7, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 21\}$, 在一个八度音阶中有 12 个半音, P_i 表示每一个音到第一级音符的音程。即 $P_i = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11\}$ 。在自然大调中, $P_n = \{0, 2, 4, 5, 7, 9, 11\}$ 。MIDI 码是从计算机易处理的角度来考虑的,对应的是这 20 个音高个体的 MIDI 音高,可表示为:

$$P_m = \{P | P = M + C; M \in P_i\} \quad (1)$$

$$C = \begin{cases} (6 + T_k), & T_k \in 2N - 1 \\ (12 + T_k) \% 12, & T_k \in 2N \end{cases} \quad (2)$$

$$P_c = \begin{cases} 58 + 2 \times P_{post} + 12 \times P_{pre}, & P_{post} = 1, 2, 3 \\ 57 + 2 \times P_{post} + 12 \times P_{pre}, & P_{post} = 4, 5, 6, 7 \end{cases} \quad (3)$$

$$P_s = ((P - C) - 60) \% 12 \quad (4)$$

在式(1)中, P 为用 MIDI 码来表示的绝对音高, P_c 为调号为 C 调 P_h 对应的 MIDI 音高的集合,可表示为:

$P_c = \{52, 53, 55, 57, 59, 60, 62, 64, 65, 67, 69, 71, 72, 74, 76, 77, 79, 81, 83, 84\}$, $C \in [0, 11]$, $C \in N$, 表示不同调式升高的半音数。在式(2)中, $T_k \in [-6, 5]$ 中的整数,分别对应 12 个调,表示每个调中升记号或降记号的数量。其中正整数表示升记号数量,负整数表示降记号数量,0 表示 C 调。 C 为对应调式的音阶在 MIDI 码上的偏移量。在式(3)中, P_{pre} 为 P_h 的前缀, P_{post} 为 P_h 的后缀。在式(4)中,为了简化模型,我们将每一个音到第一级音符的音程 P_i 对 12 取余数,将音程差控制在一个八度里。

2.2 建立音乐知识库的分层贝叶斯网

定义 1^[13] 贝叶斯网是满足如下性质的有向无环图。

1) 一组随机变量构成网中结点。

2) 从结点 X 指向结点 Y 的有向边表示 X 直接影响 Y 。每一个结点 X 与其非后代结点独立于 X 的父亲节点集。

3) 每一个结点有一张条件概率表(CPT),表示其父结点对它的影响。

BN 以乘积的方式表示了随机变量的联合概率分布:

$$P(x_1, \dots, x_n) = \prod_{i=1}^n P(x_i | Parents(x_i))$$

基于上述贝叶斯网的定义,我们将根据已有的 MIDI 音乐库,基于数理统计的知识建立一个基于 MIDI 音乐的贝叶斯网知识库。统计时,将所有音都移到同一个八度进行处理,且都转换为 C 调统一处理。即 $P_i \in [60, 71]$, $P_i \in N$ 。

音乐知识库将对每一个音级进行统计学习,统计旋律中每一个音级音符后续音符出现的概率,用 $P(note | known)$ 表示。然后对应生成 12 个朴素贝叶斯网。其中 $note$ 为目标音符(见图 1)。

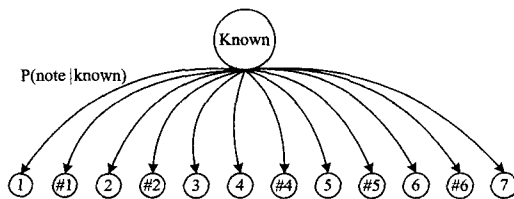


图 1 音乐知识库的分层贝叶斯网

2.3 确定 MIDI 音乐文件的调式并填充音符的先验知识库

在 MIDI 音乐中,音乐的移调命令为十六进制的“ff 59 02”^[16]。对于有移调的命令的 MIDI 文件,我们可以根据 MIDI 音乐的头文件直接得到调式,并直接按已知的调式对音符进行统计。但是,并非所有的 MIDI 音乐都有移调命令。在这种情况下无法直接得到 MIDI 音乐的调式。在 MIDI 标准中,音乐的开始为十六进制命令“90”,音乐的关闭为十六进制命令“80”。我们将根据音符的打开与关闭得到音乐的 P_n 值,并按以下算法计算音乐的调式。

算法 1 给定了一个 MIDI 音乐文件,自动分析出该音乐的调式。首先,若 MIDI 文件中有移调命令,则可根据移调命令直接确定出该首音乐的调式(步骤 1)。其次,若没有移调符号,则分别按 12 个调遍历一首 MIDI 音乐的每一个音乐,并记录存在升降符号的音符数量。最后,找到升降符号最少的这个调,即为这首 MIDI 音乐的调。

算法 1 analyseTone(Midi)

输入: Midi

输出: T

```

① if find “ff 59 02”
    return T
else
② for each y in Tk do
③     if (command == 90)
④         Ps ← ((Pm - C) - 60) % 12;
⑤         search(Ps, Pn)
⑥         if (!(Ps ∈ Pc)
                Ny ++;
            end if
        Ny ← Min(Ny);
        T ← Tk;
    end if
end for
end else
⑦ return T
    
```

算法 1 中, search() 可在时间复杂度 $O(n)$ 内计算得到, N_y 可在 $O(n)$ 内计算得到, 因此最坏情况下算法 1 的执行情况为 $O(n^2)$ 。

音乐的 3 要素: 音高, 时值, 力度。在本文中, 只考虑音高的处理, 假定时值与力度已经生成。为了便于分析, 每个音符音高取值有 12 种情况(假定在同一个八度内)。每一个音符的 12 种取值都有各自的先验概率, 可以从对 MIDI 音乐库的统计中获得。在知道了音乐的调式 T 之后, 我们将以 C 调统计音阶中的音符出现在音乐中的概率和在知道前面的一个音符后, 后续音符出现的概率。

2.4 对音乐旋律线进行建模

因为音乐具有律动性, 在创作音乐之前, 需要绘制一条旋律曲线, 并用一个矩阵来记录它, 即 $P_{i,j} = (0 < i < w, 0 < j < l)$ 。矩阵的初始值为 0, 曲线经过的地方, 将值设为 1。即:

$$M_{i,j} = \begin{cases} 1, & (i,j) \in Line \\ 0, & (i,j) \notin Line \end{cases} \quad (5)$$

根据这条用矩阵来记录的旋律线, 我们将统计出线的所有极值点 MP , 极值点的高度为 MPH , 这将作为我们在音乐推理中的“证据”来使用。

3 基于贝叶斯网生成音高

为了表达音乐的不确定性, 我们画一条表达音乐律动的曲线。这条曲线是人随机画成的。我们将其从中提取节点信息, 这些信息具有随机性。我们将其作为音乐推理的证据, 提取的证据越多, 越能够使音乐具有特定的风格。

1) 如图 2 所示, 绘制出一条旋律线, 并且假设已完成了对音乐节奏的分析处理。在这条旋律线中, 我们找出每一个极值点的高度 MPH , 即矩阵中的 Y 轴坐标。以所有的这些记录的高度为证据, 并用 B 表示。

$$B_i = MPH_i \quad (6)$$

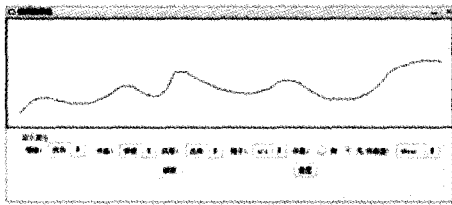


图 2 创作一条旋律线

极大值点与极小值点将直观地反映音的高低随时间律动的线条, 为了指导算法设计, 构建一个音高律动规则模型, 如式(7)~式(9)所示:

$$\Delta h_i = MPH(i) - MPH(i-1) \quad (7)$$

$$y = MPH(i-1) + \frac{\Delta h}{2} \quad (8)$$

$$S^2 = \sum_{k \in [i-1, i]} MPH_k^2 - y^2 \quad (9)$$

通过分析 S^2 的值, 系统将推出音乐的副歌部分, 在副歌部分音乐通常出现较大的转折。我们取旋律线中 S^2 最大的一段旋律为副歌, 副歌与主歌的比例通常满足黄金比例^[17], 以此来确定副歌的长度。

$$\Delta P = |P(i) - P(i-1)| \in S_i \quad (10)$$

$$S_i = \{I | I \geq 0, I \leq I_{\max}\} \quad (11)$$

$$I_{\max} = \begin{cases} 7, & MP \in \text{主歌} \\ 12, & MP \in \text{副歌} \end{cases} \quad (12)$$

2) 之后系统将生成节奏对应的音高, 生成的音符之间的关系, 可以用一个贝叶斯网来表示, 如图 3 所示。

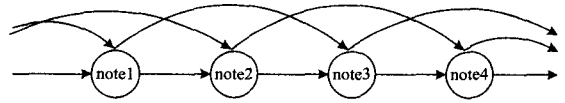


图 3 音与音之间的关系

图 4 中, “note”为待生成的音符, “fringe”为未知的音符, “b”为证据。“1 2 3 4 5”表示已经生成的音符 *known*。其中每一个 *fringe* 是未知的, 它们可以为音阶 12 个音中的任一个音符。并且每一个 *fringe* 的取值是彼此独立的, 先验概率为 $P(\text{fringe})$ 。待生成音符的先验概率受 *known* 音符的影响, 先验概率为 $P(\text{note} | \text{known})$ 。

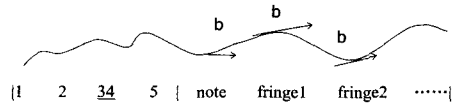


图 4 贝叶斯推理的最大可能解释问题示意图

算法 2 将根据证据来选择最优音符。首先, 确定待生成的音符是在主歌部分还是在副歌部分(步骤 1)。其次, 根据先验概率、证据和边缘概率, 运用贝叶斯公式生成音符。

算法 2 createNote(b, fringe)

输入: b, fringe

输出: note

① if (note ∈ 主歌)

 Note[] ← 主歌;

else

 Note[] ← 副歌;

end if

② for each note in note[]

$P_n \leftarrow P(\text{note} | \text{known});$

③ for each note in fringe;

④ if (note, fringe_i) ∈ b

$P_i \leftarrow P_i(i);$

⑤ for each note in fringe_{i+1}

⑥ if (fringe_i, fringe_{i+1}) ∈ b

$P_{i+1} \leftarrow P_i(i+1);$

end if

end for

end if

$P_{\text{total}} \leftarrow P_{\text{total}} + (P_n P_i \times P_{i+1});$

end for

⑦ $P[] = P_n * P_{\text{total}};$

end for

⑧ $P \leftarrow \text{Max}(P[]);$

⑨ note ← Note[P];

⑩ return note

算法 2 中, P_{total} 可在复杂度时间 $O(n^2)$ 内计算得到, $P[]$ 可在 $O(n)$ 内计算得到, 因此最坏情况下算法 2 的执行情况为 $O(n^3)$ 。

3) 在生成音符的过程中, 以生成音符与从旋律线分析得到的证据为依据, 对待生成音符进行推理并生成音高。因此, 生成某一个音符由已经生成的音符和旋律线决定。所以待生成音符 *note* 在已知音符与已知证据的条件下的条件概率为 $P(\text{note} | \text{known}, b)$ 。令 *fringe* 表示除了已知音符 *known* 和查

询音符 *note* 以外由所有未知音符组成的复合随机变量,则:

$$P(\text{note}|\text{known}, b) = a \sum_{\text{fringe}} P(\text{note}, \text{known}, b, \text{fringe})$$

定理 1 若 *note* 受 *known*, *b* 影响,在 *fringe* 条件独立的条件下,*note* 的条件概率等于 *note* 的概率与所有满足证据的 *fringe* 的组的概率乘积之和。

证明:

$$\begin{aligned} P(\text{note}|\text{known}, b) &= a \sum_{\text{fringe}} P(\text{note}, \text{known}, b, \text{fringe}) \\ &= a \sum_{\text{fringe}} P(b|\text{note}, \text{known}, \text{fringe}) P(\text{note}, \text{known}, \text{fringe}) \\ &= a \sum_{\text{fringe}} P(b|\text{note}, \text{known}, \text{fringe}) P(\text{note}, \text{known}) P(\text{fringe}) \\ &= a P(\text{note}, \text{known}) \sum_{\text{fringe}} P(b|\text{note}, \text{known}, \text{fringe}) P(\text{fringe}) \\ &= a P(\text{known}) P(\text{note}|\text{known}) \sum_{\text{fringe}} P(b|\text{note}, \text{known}, \text{fringe}) P(\text{fringe}) \\ &= a' P(\text{note}|\text{known}) \sum_{\text{fringe}} P(b|\text{note}, \text{known}, \text{fringe}) P(\text{fringe}) \end{aligned}$$

当假设音符满足证据时, $P(b|\text{note}, \text{known}, \text{fringe}) = 1$,

即:

$$P(\text{note}|\text{known}, b) = a' P(\text{note}|\text{known}) \sum_{\text{fringe}} P(\text{fringe})$$

反之, $P(\text{note}|\text{known}, b) = a' P(\text{note}|\text{known}) \sum_{\text{fringe}} 0 \times$

$P(\text{fringe}) = 0$

综上所述:

$$P(\text{note}|\text{known}, b) = \begin{cases} a' P(\text{note}|\text{known}) \sum_{\text{fringe}} P(\text{fringe}), & \Delta h_{\text{fringe}} \in b \\ 0, & \Delta h_{\text{fringe}} \notin b \end{cases} \quad (13)$$

当满足证据的条件时,由于 *fringe* 音的取值是相互独立的,将每一个 *fringe* 位置对应的概率相乘,即为满足证据的一种情况。由于每一个 *fringe* 位置的音符有多种取值,将满足证据的所有情况求和相加,即为目标音符 *note* 的概率。由于 *note* 有多种取值,以此类推,算出所有 *note* 的概率。在每一个概率前乘上对应的概率 $P(\text{note}|\text{known})$,将得到一个表示每一种可能取值的概率数组。乘上归一化因子 *a*,并取最大的这一种可能,作为待生成音符的取值。同理,按以上的方法,生成所有节奏上对应的音高。

4 实验

为了对生成的音乐进行评估,首先对音乐库中音高所占的比例(见表 1)与生成音乐的音符比例进行统计对比(见表 2)。此外,为了方便描述,将本文的算法 2 记为“Model-Pitches Evidence-Bayesian Network”算法(即“模型-音高证据-贝叶斯网”算法,简称为“M-PE-BN”算法),通过分析已有乐曲并以音符为单位的马尔可夫模型方法记为“Imitation-Note-Markov”算法(即“模仿-音符-马尔可夫”算法,简称“I-N-M 算法”^[17])。然后,我们使用 M-PE-BN 创作 1000 首歌曲,与 I-N-M 算法进行对比,并创建一个形式规则库来对生成的音乐进行客观的评价。最后,对于生成的音乐,我们将人工进行打分,得分高的乐曲将被存入音乐知识库,成为音乐系统的先验知识。实验结果表明使用贝叶斯方法可以较好地推理生成音

乐的音高。

表 1 先验音乐库中的音符分布

Ps	0	1	2	3	4	5
P(%)	14.1	0.08	13.7	0.01	14.1	14.2
Ps	6	7	8	9	10	11
P(%)	0.02	17.4	1.6	12.1	0.01	0.05

表 2 生成乐曲的音符分布

Ps	0	1	2	3	4	5
P(%)	12.1	0	14.6	0	16.1	15.6
Ps	6	7	8	9	10	11
P(%)	2e-3	21.4	0.4	14.5	4e-4	6e-3

用于客观评价的形式规则库:

1)“坏点”音符的百分比:如果乐句内相邻两个音符的音程在 8 度以上,则后面的音符判定为“坏点”音符。

2)评价音乐和谐的程度规则库 R_u ,如式(14)一式(16)所示。

$$I_b = \{0, 2, 3, 4\} \quad (14)$$

$$I_{bb} = \{0, 2, 3, 4, 5, 7\} \quad (15)$$

$$I_f = \{0, 2, 3, 4, 5, 7, 9, 10, 12\} \quad (16)$$

其中, I_b, I_{bb}, I_f 分别表示小节之内、小节之间、高潮过渡对应的音程样本值,我们用 $N_{i=1}$ 来记录一个乐段中小二度音程出现的次数, $N_{i=1}$ 的值与音乐和谐度成反比。这是因为该音程具有强烈的紧张度。和谐度的值为生成音符中和谐音符与所有生成音符的比值^[17]。

3)高潮的体现度:乐曲黄金分割点^[18]后的平均音高减去黄金分割点前的平均音高(计算时采用 MIDI 码)。

使用 M-PE-BN 算法生成 1000 首乐曲所需的时间为 3.2s。从表 2 可以看出,在音乐库的 100 首音乐中,生成音乐音符的比例与音乐库中音符的比例有较好的匹配程度,拥有音乐库中音乐一定的特性。表 3 中“e”表示音符的分布概率值较小。此外,从表 4 可以看出,生成的音乐拥有较为丰富的变化,引导生成的乐曲较为满足客观评价条件。

表 3 生成音符的横向分布

n	0	1	2	3	4	5	6	7	8	9	10	11
0	0.09	0	0.25	0	0.25	0.1	e	0.01	0	0.3	e	e
1	0	0	0	0	0	0	0	0	0	0	0	0
2	0.3	0	0.26	0	0.26	0.1	0	0.02	0	0.08	0	e
3	0	0	0	0	0	0	0	0	0	0	0	0
4	0.16	0	0.3	0	0.33	0	e	0.2	0	0.08	0	e
5	0	0	0.16	0	0.1	0.15	e	0.33	0.02	0.25	0	e
6	0.25	0	0.25	0	0	0	0	0.5	0	0	0	0
7	0.06	0	0.01	0	0.06	0.36	0	0.26	0	0.25	0	0
8	0	0	0	0	0	0	0	0	0	0	0	0
9	0.14	0	0.01	0	0.05	0.15	0	0.36	0	0.29	0	e
10	1.0	0	0	0	0	0	0	0	0	0	0	0
11	0.25	0	0.63	0	0.31	0.13	0	0.13	0	0.13	0	0

表 4 两种方法客观评价的比较

	M-PE-BN 算法	I-N-M 算法
坏点	0%	3.8%
和谐度	97.3%	87.4%
高潮体现度	15.3%	0.03%

结束语 本文通过贝叶斯方法研究了计算机辅助作曲中的音高的表示和推理,讨论了利用贝叶斯网通过计算机来表达音乐的不确定性,建立了一个关于音高的贝叶斯网推理的

(下转第 28 页)

表2 关系标记“首先/然后/最后”的规则

ID	keyword	prior/constraint	constraints	result
124	首先/然后/最后	1	0(首先,然后)=1,0(然后,最后)=0	0(首先)=true,0(然后)=true,0(最后)=false
125	首先/然后/最后	2	0(首先,然后)=1,0(然后,最后)=1	0(首先)=true,0(然后)=true,0(最后)=false
126	首先/然后/最后	2	0(首先,然后)=1,0(然后,最后)=1,0(最后)=1	0(首先)=true,0(然后)=true,0(最后)=false
127	首先/然后/最后	2	0(首先,然后)=1,0(然后,最后)=1,0(最后)=1	0(首先)=true,0(然后)=true,0(最后)=false
128	首先/然后/最后	2	0(首先,然后)=1,0(然后,最后)=1,0(最后)=1	0(首先)=true,0(然后)=true,0(最后)=false
129	首先/然后/最后	2	0(首先,然后)=1,0(然后,最后)=1	0(首先)=true,0(然后)=true,0(最后)=true

所以可知 ID=78、ID=126 和 ID=127 的规则满足匹配，又可知，根据规则解析器调用 4 张规则表的顺序，ID=126 和 ID=127 的规则优先采用，ID=78 的规则就丢弃；ID=126 的规则结果“首先”和“然后”充当关系词，“最后”判定为不充当关系词，而 ID=127 的规则结果“首先”和“然后”充当关系词，“最后”判定为充当关系词，所以根据正覆盖策略，当一个关系标记已标记为 false，再判定标记为 true 时，要对先前的赋值标记进行重新赋值，最终的结果选择搭配关系为“首先/然后/最后”，复句中的分词“首先”、“然后”和“最后”都充当关系词。

结束语 本文设计了适用于复句关系词自动标识的规则引擎的框架结构，并提出 3 大实现策略，以解决关系标记搭配的匹配问题、规则冲突的消解问题，以及最优结果的选取问题。

本文对提出的算法，用大量复句实例进行了测试。实验结果表明，本文提出的策略优化了关系标记搭配的匹配的推理过程，大大提高了系统的可维护性，提高了系统的效率，使汉语复句标识变得方便快捷。同时，由于解决了关系词匹配过程的问题，关系词识别的正确率高达 90.5%。

参考文献

[1] 邢福义. 汉语复句研究[M]. 北京: 商务印书馆, 2001

(上接第 24 页)

智能 MIDI 音乐系统。实验表明，我们将贝叶斯网与计算机自动作曲结合，更能反映音乐不确定性这一特性，并能生成符合音乐乐理规则的音乐。

从贝叶斯网与 MIDI 音乐库中定义并提取更多独立的证据，使音乐推理更加地灵活、丰富，是我们将要进行研究的工作。

参考文献

[1] 冯寅, 周昌乐. 算法作曲的研究进展[J]. 软件学报, 2006, 17(2): 209-215

[2] Tromp M, Bod R, Honingh A. Learning Symbolic Music through Hidden Markov Model Induction[J]. 2012

[3] Fernández J D, Vico F. AI methods in algorithmic composition: a comprehensive survey[J]. Journal of Artificial Intelligence Research, 2013, 48(1): 513-82

[4] Bell C. Algorithmic music composition using dynamic Markov chains and genetic algorithms[J]. Journal of Computing Sciences in Colleges, 2011, 27(2): 99-107

[5] Salas H A G, Gelbukh A, Calvo H. Music composition based on linguistic approach[M] // Advances in Artificial Intelligence. Springer, 2010: 117-128

[6] Boutsinas C, Ha B. Automatic interactive music improvisation based on data mining[J]. International Journal on Artificial Intelligence Tools, 2012, 21(4): 1-24

[7] Khan A H. Artificial intelligence approaches to music composition[D]. Northern Kentucky University, 2013

[2] 王跃龙, 姬东鸿. 汉语树库综述[J]. 当代语言学, 2009, 12(1): 47-55

[3] 姚双云. 复句关系标记的搭配研究[M]. 武汉: 华中师范大学出版社, 2008

[4] 李文翔, 晏蒲柳, 张滨. 基于语料库的关联词识别方法[J]. 计算机工程与应用, 2004(7)

[5] 刘云. 复句关系词语离析度考察[J]. 语言教学与研究, 2008(6)

[6] 吴锋文. 基于关系标记的汉语复句分类研究[J]. 汉语学报, 2011(3): 63-73

[7] 李艳翠, 孙静, 周国栋, 等. 基于清华汉语树库的复句关系词识别与分类研究[J]. 北京大学学报, 2013(12)

[8] 胡金柱, 舒江波, 等. 面向中文信息处理的复句关系词提取算法[J]. 计算机工程与科学, 2009(10)

[9] 胡金柱, 陈江曼, 杨进才. 基于规则的连用关系标记自动标识研究[J]. 计算机科学, 2012(5): 190-194

[10] 胡金柱, 雷利利, 杨进才. 多重复句关系标记搭配的求解模型研究[J]. 计算机工程与科学, 2011(11): 177-182

[11] 姚双云, 胡金柱, 等. 关联词搭配的自动发现[J]. 计算机应用研究, 2011(12)

[12] 胡金柱, 舒江波, 等. 基于 VML 的复句关系层次树的可视化研究[J]. 计算机应用研究, 2010(1)

[13] 姚双云, 胡金柱. 篇章连贯语义关系的自动标注方法[J]. 计算机工程, 2012(7)

[14] 邓沌华, 胡金柱, 等. 面向现代汉语复句信息工程的语料仓库构建研究[J]. 信息系统工程, 2013(9)

[15] 宋震, 郭福顺, 李莲治. IMPR: 一种优于 RETE 算法的多模式/多对象匹配算法[J]. 小型微型计算机系统, 2002(2)

[8] Edwards M. Algorithmic composition: computational thinking in music[J]. Communications of the ACM, 2011, 54(7): 58-67

[9] Donnelly P, Sheppard J. Evolving four-part harmony using genetic algorithms[M] // Applications of Evolutionary Computation. Springer, 2011: 273-82

[10] Morone A, Manzolli J, Von Zuben F, et al. Evolutionary Computation applied to Algorithmic Composition[J]. NICS Reports, 2013(3): 49-53

[11] Matic D. A genetic algorithm for composing music[J]. Yugoslav Journal of Operations Research, 2013, 20(1): 2334-6043

[12] Kirke A, Miranda E R. A survey of computer systems for expressive music performance [J]. ACM Computing Surveys (CSUR), 2009, 42(1): 3

[13] 刘惟一, 李维华, 岳昆. 智能数据分析[M]. 北京: 科学出版社, 2007

[14] Raphael C, et al. A Bayesian Network for Real-Time Musical Accompaniment[M] // Advances in Neural Information Processing Systems. 2002

[15] Oyen D, Lane T, et al. Leveraging Domain Knowledge in Multi-task Bayesian Network Structure Learning[C] // AAAI, 2012

[16] 李金鑫. 计算机自动混音系统设计——面向自动作曲系统 MIDI 表达的实现[J]. 电脑编程技巧与维护, 2010(14): 104-1055

[17] 曹西征, 毛文涛, 乔锐, 等. 基于音高旋律元的柔和乐曲的自动作曲算法[J]. 自动化学报, 2012, 38(10): 1627-1638

[18] Povel D-J. Melody generator: A device for algorithmic music construction[J]. Journal of Software Engineering and Applications, 2010, 3(7): 683