

基于警示传播与 DPLL 算法的启发式极性决策算法

秦永彬 许道云 王晓峰

(贵州大学计算机科学与信息学院 贵阳 550025)

摘要 警示传播(WP)算法是信息传播算法的重要基础,WP 算法的本质是因子图上警示信息的迭代过程,在算法收敛时得到一组稳定的警示信息,并利用局部腔域得到公式变元的部分赋值。分析了警示传播算法的基本原理,给出了算法的改进。RB 实例集上的实验证明,改进后的算法比原算法具有迭代次数和运行时间,提高了收敛速度。然而,在 RB 模型产生的大部分实例集上,警示传播算法不收敛,因而不能有效求解公式。警示传播算法与 DPLL 算法的组合使用使回溯计算次数大大降低,从而有效地弥补了 WP 算法的不足。通过在 RB 实例集上的测试实验表明,该方法是有用的。

关键词 信息传递,警示传播算法,收敛性,DPLL 算法

中图分类号 TP301 **文献标识码** A

Heuristic Polarity Decision Making Algorithm Based on Warning Propagation and DPLL Algorithm

QIN Yong-bin XU Dao-yun WANG Xiao-feng

(College of Computer Science and Information, Guizhou University, Guiyang 550025, China)

Abstract The warning propagation(WP) algorithm is an important foundation of message propagation algorithm, the essence of the WP algorithm is the iteration process of warning message on the factor graph. When the algorithm is convergent, it can get a set of stable warning message and get some partial assignment of formula variables by local cavity domain. The analysis of basic principle of the WP algorithm was presented, and the improvement of the algorithm was given. The experiment on the RB sets shows that the improved algorithm has fewer iteration times, less execute time and faster convergence speed than the original WP algorithm. However, in the most of the RB sets, the WP algorithm is not convergent, and then it can not solve the formula effectively. The combination of the WP and DPLL algorithm can reduce the times of the backdating calculation, and then make up the shortage of the WP algorithm. The result of the experiment on the RB sets shows that the method is effective.

Keywords Message passing, Warning propagation(WP) algorithm, Convergence, DPLL algorithm

1 引言

逻辑公式的可满足问题(SAT 问题)是理论计算机科学的著名问题,在过去的几十年中,世界各国的许多学者都对此做了大量的研究,取得了一些卓有成效的成果,提出的许多高效、实用的算法^[1-6]。其中最为经典的就是 Davis 等人提出的 DPLL 算法^[7,8]以及欧洲学者 M. Mezard, G. Parisi 等人^[9,10]提出的一些新的信息传播算法,如警示传播(Warning Propagation, WP)算法、置信传播(Belief Propagation, BP)算法和调查传播(Survey Propagation, SP)算法等,并由此衍生出了很多改进算法和相关启发式算法。

WP 算法是一种基于统计物理学的信息传播算法^[9],警示传播算法是信息传播算法的重要基础,它通过在因子图上进行某种警示信息的迭代和信息传递,以高概率来确定部分变量的取值,对原公式进行相应的化简。

通过对 WP 算法数学原理和收敛性的分析,我们给出了 WP 算法在初始步的改进方法,并在 RB 实例集上进行了验证。通过实验我们发现,WP 算法在 RB 模型产生的大部分实例集上都不收敛,公式不能得到有效的求解。我们将 WP 算法与 DPLL 算法相结合,提出了基于 WP 与 DPLL 算法的启发式决策算法。

2 基础知识

定义 1 $F = \{a_1, \dots, a_m\}$ 为一个 k-CNF 公式,含有 n 个变量 x_1, \dots, x_n 。公式 F 可以用一个二分图 $G = (X \cup A, E)$ 表示,称为因子图(Factor Graph)或变量-子句图,其中,变量结点集 $X = \{1, \dots, n\}$,子句结点 $A = \{a_1, \dots, a_m\}$ 。G 中的边分为两类:实边和虚边。有时用 a, b, c, d, \dots 表示子句结点,用 i, j, k, \dots 表示变量结点。

实边: $(a_i, j) \in E \Leftrightarrow$ 子句 a_i 含正文字 x_j ;

到稿日期:2010-01-15 返修日期:2010-04-09 本文受国家自然科学基金(60863005, 61011130038),贵州省省长基金(200404),贵州大学自然科学基金青年基金(2009021)资助。

秦永彬(1980—),男,博士生,讲师,CCF 会员,主要研究方向为可计算分析, E-mail: cse. ybqin@gzu. edu. cn; 许道云(1959—),男,博士,教授,博士生导师,CCF 高级会员,主要研究方向为可计算分析; 王晓峰(1980—),男,硕士生,主要研究方向为可计算分析。

虚边: $(a_i, j) \in E \Leftrightarrow$ 子句 a_i 含负文字 $\neg x_j$ 。

给定公式

$$F = \{a_1, a_2, a_3, a_4\} = \{(x_1 \vee \neg x_2 \vee \neg x_3), (\neg x_1 \vee x_2 \vee x_4), (\neg x_2 \vee x_3 \vee x_5), (\neg x_2 \vee x_4 \vee x_5)\}$$

根据公式可知变量结点集为 $X = \{1, 2, 3, 4, 5\}$, 子句集合为 $A = \{a_1, a_2, a_3, a_4\}$, 公式的因子图如图 1 所示。

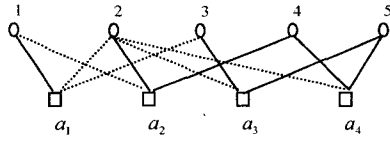


图 1 变量-子句图

定义 2 给出 $V(a), C(i), I_a^i, C_a^+(i), C_a^-(i)$ 等记号的定义。

$V(a)$: 出现在子句 a 中的变量集合, $V(a) := V_+(a) \cup V_-(a)$, 其中,

$V_+(a)$: 出现在子句 a 中的正文字对应的变量标识集合;

$V_-(a)$: 出现在子句 a 中的负文字对应的变量标识集合。

$$V(a) \setminus i := V(a) - \{i\}.$$

$C(i)$: 含变量 x_i 的子句集合, $C(i) := C_+(i) \cup C_-(i)$, 其中,

$C_+(i)$: 变量 x_i 正出现的子句集合;

$C_-(i)$: 变量 x_i 负出现的子句集合。

$$C(i) \setminus a := C(i) - \{a\}.$$

I_a^i : 出现子句 a 中的变量 x_i 的符号识别参数。

$$I_a^i = \begin{cases} -1, & x_i \in a \\ 1, & \neg x_i \in a \end{cases}$$

$C_a^+(i)$: 除 a 以外, 含有变量 x_i 且变量 x_i 的出现与其在子句 a 中的出现一致的子句集。

$C_a^-(i)$: 除 a 以外, 含有变量 x_i 且变量 x_i 的出现与其在子句 a 中的出现不一致的子句集。

$C_a^+(i), C_a^-(i)$ 形式定义为

$$C_a^+(i) = \begin{cases} C_-(i) \setminus a, & I_a^i = 1 \\ C_+(i) \setminus a, & I_a^i = -1 \end{cases}$$

$$C_a^-(i) = \begin{cases} C_+(i) \setminus a, & I_a^i = 1 \\ C_-(i) \setminus a, & I_a^i = -1 \end{cases}$$

显然, $C(i) \setminus a = C_a^+(i) \cup C_a^-(i)$ 。

根据定义 2, 对于给定的公式

$$F = \{(x_1 \vee \neg x_2 \vee \neg x_3), (\neg x_1 \vee x_2 \vee x_4), (\neg x_2 \vee x_3 \vee x_5), (\neg x_2 \vee x_4 \vee x_5)\} = \{a, b, c, d\}$$

有: $I_a = (-1, 1, 1), I_b = -1, I_c = 1, I_d = 1$

$$C_a(2) := \{c, d\}, C_a^-(2) := \{b\}$$

3 WP 算法原理分析及改进

WP 算法的基本思想在于: 在计算过程中, 从子句结点 a 向变元结点 i 传递一个布尔值 (0 或 1), 记为 $u_{a \rightarrow i}$ 。其中, $(a, i) \in E$ 。更新 $u_{a \rightarrow i}$ 的迭代函数定义为

$$u_{a \rightarrow i}(t+1) = \prod_{j \in V(a) \setminus i} \theta(-I_a^j (\sum_{b \in C(j) \setminus a} I_b u_{b \rightarrow j}(t)))$$

$$\text{简记为: } u_{a \rightarrow i} = \prod_{j \in V(a) \setminus i} \theta(-I_a^j (\sum_{b \in C(j) \setminus a} I_b u_{b \rightarrow j}))。$$

其中, 截尾函数 $\theta(x)$ 定义如下

$$\theta(x) = \begin{cases} 1, & x > 0 \\ 0, & x \leq 0 \end{cases}$$

说明: $u_{a \rightarrow i} = 1$ 表示 a 的满足性完全依赖变量 x_i ; $u_{a \rightarrow i} = 0$ 表示 a 的满足性并非完全依赖变量 x_i , 即其它变量的取值有可能使 a 可满足。

WP 算法是一个逐步迭代算法, 算法有可能收敛, 也有可能不收敛, 通过设置一个最大迭代步 t_{\max} 强制算法终止。

步骤 1 $t=0$: 对因子图 $G=(V, E)$ 中的每一条边 $(a, i) \in E$, 分别以 0.5 的概率随机地从 $\{0, 1\}$ 中取一个值赋给 $u_{a \rightarrow i}(0)$ 。

步骤 2 $t > 0$: 在因子图 $G=(V, E)$ 中, 对每条边 $(a, i) \in E$, 分别更新 $u_{a \rightarrow i}(t-1)$ 到 $u_{a \rightarrow i}(t)$ (由一个 WP 更新子算法完成)。

如果存在某个 $t_0 < t_{\max}$, 使得对所有的边 $(a, i) \in E$, 都有 $u_{a \rightarrow i}(t_0) = u_{a \rightarrow i}(t_0 - 1)$, 表示算法收敛, 则令 $u_{a \rightarrow i}^* = u_{a \rightarrow i}(t_0)$; 否则, 输出算法不收敛。

如果 WP 算法收敛, 通过迭代, 可以得到不动点 $u^* = \{u_{a \rightarrow i}^*; (a, i) \in E\}$ 。利用 $u^* = \{u_{a \rightarrow i}^*; (a, i) \in E\}$ 启发式定义每一变量的取值, 并对公式进行简化, 如此重复, 寻找到公式的可满足指派集合。

在 WP 算法中, 警示信息 $u_{a \rightarrow i}$ 和信息传递可解释为:

$u_{a \rightarrow i} = 1$: 变量 i 从子句 a 得到这样的信息: 为使子句 a 可满足, 它应该接受 x_i 的取值。

对每一变量 i , 可以计算一个局部腔域 H_i 和一个冲突标识 $c_i \in \{0, 1\}$:

$$H_i = -\sum_{b \in C(i)} I_b u_{b \rightarrow i}^* = \sum_{b \in C_+(i)} u_{b \rightarrow i}^* - \sum_{b \in C_-(i)} u_{b \rightarrow i}^*$$

$$c_i = \begin{cases} 1, & \text{if } (\sum_{b \in C_+(i)} u_{b \rightarrow i}^*) (\sum_{b \in C_-(i)} u_{b \rightarrow i}^*) > 0 \\ 0, & \text{w} \end{cases}$$

当 $c_i = 1$ 时, $(\sum_{b \in C_+(i)} u_{b \rightarrow i}^*) (\sum_{b \in C_-(i)} u_{b \rightarrow i}^*) > 0$ 表示: 变量 x_i 正、负出现的两组子句中, 对 x_i 的取值至少出现一对冲突要求。即, 存在 $b \in C_+(i), u_{b \rightarrow i}^* = 1; b' \in C_-(i), u_{b' \rightarrow i}^* = 1$ 。如: b 告诉 $i: x_i = 1; b'$ 告诉 $i: x_i = 0$ 。

当 $c_i = 0$ 时, $(\sum_{b \in C_+(i)} u_{b \rightarrow i}^*) (\sum_{b \in C_-(i)} u_{b \rightarrow i}^*) = 0$, 通过 $H_i = -\sum_{b \in C(i)} I_b u_{b \rightarrow i}^* = \sum_{b \in C_+(i)} u_{b \rightarrow i}^* - \sum_{b \in C_-(i)} u_{b \rightarrow i}^*$ 的符号, 定义变量 x_i 的值为

$$x_i = \begin{cases} 1, & H_i > 0 \\ \{0, 1\}, & H_i = 0 \\ 0, & H_i < 0 \end{cases}$$

即

$$x_i = \begin{cases} 1, & \sum_{b \in C_+(i)} u_{b \rightarrow i}^* > 0, \sum_{b \in C_-(i)} u_{b \rightarrow i}^* = 0 \\ \{0, 1\}, & \sum_{b \in C_+(i)} u_{b \rightarrow i}^* = \sum_{b \in C_-(i)} u_{b \rightarrow i}^* = 0 \\ 0, & \sum_{b \in C_+(i)} u_{b \rightarrow i}^* = 0, \sum_{b \in C_-(i)} u_{b \rightarrow i}^* > 0 \end{cases}$$

$H_i > 0$ 时, $\sum_{b \in C_+(i)} u_{b \rightarrow i}^* > 0, \sum_{b \in C_-(i)} u_{b \rightarrow i}^* = 0$, 如果 WP 算法收敛, 可以通过高概率确定部分变量的取值。必有子句 $b \in C_+(i)$ 的满足性完全依赖于所含的正文字 x_i 的取值, 而且对所有的子句 $b' \in V_-(i)$, 其满足性并不完全依赖于所含的负文字 $\neg x_i$ 。于是, 取 $x_i = 1$ 。

$H_i = 0$ 时, $\sum_{b \in C_+(i)} u_{b \rightarrow i}^* = \sum_{b \in C_-(i)} u_{b \rightarrow i}^* = 0$, 称变量 x_i 取值倾向不明确 (称 x_i 为不固定变元)。

$H_i < 0$ 时, $\sum_{b \in C_+(i)} u_{b \rightarrow i}^* = 0, \sum_{b \in C_-(i)} u_{b \rightarrow i}^* > 0$, 必有子句 $b \in C_-(i)$ 的满足性完全依赖于所含的负文字 $\neg x_i$ 的取值, 而且对所有的子句 $b' \in C_+(i)$, 其满足性并不完全依赖于所含的正文字 x_i 。于是, 取 $x_i = 0$ 。

因此,由 WP 算法生成的收敛的 $u^* = \{u_{a \rightarrow i}^*; (a, i) \in E\}$ 作为基础,构造一个用于寻找 CNF 公式可满足指派集的算法 Warning Inspired Decimation(WID)^[13]。在 WID 算法中,含有一个计算冲突标识的过程,如果发现冲突,则算法终止,输出公式不可满足。否则,利用 $u^* = \{u_{a \rightarrow i}^*; (a, i) \in E\}$ 诱导变元赋值以高概率满足输入公式^[10]。

为使 WP 算法快速收敛,对 WP 算法中 $t=0$ 步随机赋值作如下修改:

1)对 $(a, i) \in E$,如果 a 是一个叶结点。此时, a 为单位子句,对应 $V(a) \setminus i = \phi$,取 $u_{a \rightarrow i} = 1$;对于 $b \in C(i) \setminus a$,如果 $I_a I_b > 0$,则取 $u_{b \rightarrow i} = 1$,并对所有的 $j \in V(b) \setminus i$,取 $u_{b \rightarrow j} = 0$,原因是 b 的满足性已由 x_i 的固定取值完成,其它变元 x_j 对 b 的满足性贡献不大。如果 $I_a I_b < 0$,则取 $u_{b \rightarrow i} = 0$ 。原因是这样的 x_i 的取值已固定,对 b 的满足性无贡献。

2)剩余边再分别以 0.5 的概率作随机取值。

通过上述改进,算法的迭代次数和运行时间都有了一定的减少,收敛速度得到了提高。

实验环境是 CPU 频率为 P4 2.93GHz,内存为 2GBytes,操作系统为 Windows XP,编程环境为 VC6.0 的一台计算机。采用 RB 模型实例产生器^[11]。实验结果如表 1、表 2 所列。

表 1 WP 算法(初始随机赋值)在 RB 集上的效果

RB	Var	Clause	Time(s)	Con_times	De_v_nums
RB15	220	4360	11.98	11	189
RB16	220	4540	67.77	0	0
RB17	220	4900	17.44	8	162
RB18	220	5080	110.23	0	0
RB19	220	5260	54.98	6	153
RB20	220	5440	171.74	0	0
RB21	220	5620	47.36	7	94
RB22	220	5980	334.27	0	0
RB23	220	6160	388.91	0	0
RB24	220	6340	424.50	0	0
RB25	220	6520	482.61	0	0

表 2 改进 WP 算法(初始限定赋值)在 RB 集上的效果

RB	Var	Clause	Time(s)	Con_times	De_v_nums
RB15	220	4360	11.59	13	200
RB16	220	4540	70.31	0	0
RB17	220	4900	16.16	9	149
RB18	220	5080	109.66	0	0
RB19	220	5260	51.61	7	170
RB20	220	5440	148.33	0	0
RB21	220	5620	43.02	8	114
RB22	220	5980	325.38	0	0
RB23	220	6160	384.89	0	0
RB24	220	6340	409.81	0	0
RB25	220	6520	472.28	0	0

表 1、表 2 中, var 表示变元个数, clause 表示子句个数, Time 表示 WP 算法的运行时间, Con_times 表示 WP 算法在运行过程中公式的收敛次数, De_v_nums 表示 WP 算法得到的部分赋值的变元个数。

比较表 1 和表 2 可以发现,改进后的 WP 算法的运行时间明显小于原 WP 算法的运行时间,而且部分实例集使得算法的收敛次数 Con_times 发生了改变,部分实例集下被赋值的变元数增加,从而达到更大程度地简化公式的目的。引起算法运行时间变短的主要原因是算法迭代次数减少,也就是说,为了得到稳定的警示信息,算法可能要迭代多次,而通过对警示信息初始化值的设置,大大减少了算法迭代的次

数,算法以较少时间趋于收敛,因而降低了算法的运行时间,算法效率明显提高。

如果 WP 算法不收敛,不能得到一组稳定的警示信息,进而无法确定变元的赋值。基于此,我们对 WP 算法进行改进,提出了基于 WP 与 DPLL 算法的启发式极性决策算法,弥补了 WP 算法的不足。

4 启发式极性决策算法

针对 WP 算法在某些情况下出现的不收敛性,Parisi G 等人提出一种回溯策略^[12],其基本思想是:估计当前部分赋值下解空间的大小 N ,并给每个变量 i 赋值后解空间的大小 N_i 占此空间的比例 $p_i = N_i/N$,选择变量 i ,使得 $p_{\max} = \max\{p_i; \forall i \in V\}$,记剩下的解空间大小为 N' ,计算回溯每个变量 k 后解空间的大小 N_k ,两者的比例 $p_k = N'/N_k$, $p_{\min} = \min\{p_k; \forall k \in V\}$,如果 $p_{\max} > p_{\min}$,那么给变量 i 赋值,并取消变量 k 的赋值,这一估量在每一步都进行,从而可能纠正由前面计算偏差导致的错误赋值。实验表明,这种策略效果固然很好,但回溯方法复杂,计算量较大。李韶华等人^[13]提出了一种简单的办法,即在算法不收敛时,撤销最后一次收敛时的 m 个部分变量赋值,把剩余的子公式交给 Zchaff 解决。实验表明,这种策略尽管得到了很好的效果,但也存在不足,即在相变区域算法基本不收敛,得到的子公式往往是不可满足的,导致算法错误判断。另外,当 WP 算法在原公式不收敛时,算法退化为单一的 Zchaff 算法。

基于上述思想,我们提出两种相应的改进方法:

方法 1 WP 算法的不收敛是发生在某个简化后的子公式上。假设图 G_1, G_2, \dots, G_k 是调用 WP 算法时,生成的一系列子公式对应的因子图,图 $G_i (1 \leq i \leq k)$ 是由图 G_{i-1} 通过消解 m_i 个变元后生成的子图,这 m_i 个变元为 $x_{i1}, x_{i2}, \dots, x_{im_i}$ 。假设在第 k 个子图 G_k 上运行 WP 算法时,导致算法不收敛。这时我们撤销这 m_k 个变元 $x_{k1}, x_{k2}, \dots, x_{km_k}$ 的取值,保留已固定的变元 $x_{11}, x_{12}, \dots, x_{1m_1}, x_{21}, x_{22}, \dots, x_{2m_2}, \dots, x_{(k-1)1}, x_{(k-1)2}, \dots, x_{(k-1)m_{(k-1)}}$ 的取值,没有被固定的变元随机取值。我们得到一组近似赋值,把这组赋值作为 DPLL 变元极性决策。值得注意的是,我们是在原公式上运行 DPLL,杜绝了上述错误判断的可能性。

方法 2 WP 算法的不收敛是发生在原公式上。我们定义两次迭代的距离如下

$$num = \sum_a \sum_{i \in V(a)} (u_{a \rightarrow i}(t-1) \oplus u_{a \rightarrow i}(t))$$

根据收敛性原理,通过算法迭代,警示信息会向一组稳定的状态逼近。因此,在这里,我们用某次迭代的一组警示信息 $u_{a \rightarrow i}(t)$,使得 num 较小。 $u_{a \rightarrow i}(t)$ 可能更接近算法收敛时的稳定警示信息状态。以此警示信息为依据,利用局部域 H_i ,得到部分变元赋值,剩余变元随机选取赋值。把得到的这组近似赋值作为 DPLL 变元极性决策,启发 DPLL 算法的搜索。

将上述算法称为启发式极性决策(Heuristic Polarity Decision making, HPD)算法,算法的执行过程描述如下:读入 CNF 文件,调用 WP 算法得到警示信息。

1)若算法收敛,则调用 WID 算法固定部分变量的值,化简公式,然后判定是否可以确定公式可满足性;如果不能确定公式的可满足性,则继续调用 WP 算法;如果可以确定公式的可满足性,则输出结果,算法终止;

2)若算法不收敛,则计算局部警示信息 $u = \{u_1, u_2, \dots, u_n\}$,利用局部警示信息可以得到部分变量的赋值,其余变元赋值随机选取。把这组近似赋值作为 DPLL 的极性决策变量赋值,调用 DPLL 算法。

5 实验结果分析

实验环境是 CPU 频率为 P4 2.93GHz,内存为 2GBytes,操作系统为 Windows XP,编程环境为 VC6.0 的一台计算机。

在 RB 模型实例产生器中,取 $n=20, a=0.8, r=3$,根据文献[11]的结论,得到相变点 $p_c=0.23$ 。在相变点附近区域,实例的求解难度较大,在相变点的两侧区域,实例的求解难度相对较小。如表 3、表 4 是 HPD 算法在 RB 模型产生的实例集上的测试结果。

表 3 RB 产生的强迫性可满足实例集

P	Time(s)	Con_times	De_v_nums
0.15	0.00	10	191
0.16	0.01	6	148
0.17	0.02	8	180
0.18	0.98	0	0
0.19	0.05	13	166
0.20	1.51	0	0
0.21	1.88	1	74
0.22	2.92	1	68
0.23	3.48	0	0
0.24	0.95	4	135
0.25	3.37	0	0
0.26	2.80	0	0
0.27	1.95	3	123
0.28	1.21	0	0
0.29	0.76	6	147
0.30	0.57	0	0
0.31	0.53	8	119
0.32	0.92	0	0
0.33	0.14	13	174
0.34	0.18	8	163

表 4 RB 产生的非强迫性可满足实例集

P	Time(s)	Con_times	De_v_nums
0.15	0.01	11	189
0.16	0.64	0	0
0.17	0.02	8	162
0.18	0.98	0	0
0.19	0.05	6	153
0.20	1.34	0	0
0.21	0.07	7	94
0.22	2.90	0	0
0.23	3.46	0	0
0.24	3.85	0	0
0.25	3.36	0	0
0.26	0.27	8	152
0.27	2.30	0	0
0.28	1.62	0	0
0.29	0.46	2	130
0.30	1.20	0	0
0.31	0.89	0	0
0.32	0.73	2	115
0.33	0.09	2	99
0.34	0.27	0	0

表 3、表 4 中 P 是 RB 模型中的紧致度变量,Time 表示 HPD 算法的运行时间,Con_times 表示 WP 算法在运行过程中子公式的收敛次数,De_v_nums 表示 WP 算法得到的部分被赋值的变元个数。

通过上述实验数据不难发现,由 RB 模型使用直接编码后,产生的强迫性可满足实例集(如表 3 所列),在这些实例集上 WP 算法不收敛,分析其原因发现:在相变点两侧区域,WP 算法在原公式上基本收敛,而随着 WP 算法的进行,由于对公式的化简,导致在化简后的子公式上不收敛,而在相变点附近区域,WP 算法在原公式上不收敛。因此,WP 算法无法判定这些公式的可满足性,但 HPD 算法有效地解决了这些实例,其可以在 Time 时间内判定公式的可满足性。对于非强迫性实例集(如表 4 所列),该算法也有较好的效率。本算法在强迫性实例集和非强迫性实例集上的运行时间接近,即对于 HPD 算法来说,求解强迫性实例集和非强迫性实例集的难度相同。

当 WP 算法在原公式上不收敛时,WP 算法不能提供更有效的全局警示信息,此时就可以利用第 4 节中所说的方法得到近似解,引导 DPLL 算法的搜索,实验表明这种策略在大部分 WP 算法不收敛的公式实例上都有较好的表现。而在个别实例上,由于 WP 算法不能得到一组更有效的全局警示信息,因而 WP 算法对 DPLL 算法的变量极性部分的贡献不明显,影响了算法的搜索时间。

根据 RB 模型的原理,随着 P 的增大,实例的子句数增加,得到的因子图比较复杂,WP 算法本身的运行时间也相应增加,因而影响了 HPD 算法的求解时间,另外,步长也影响了算法的运行时间。

实验表明,HPD 算法是有效的,即使是最坏的情况下(得到的这组被赋值的决策变量没有出现在 VSIDS 的自由变量中),HPD 算法就会退化为 Zchaff 算法,然而这里的改进效果也是很明显的,因为 Zchaff 算法本身具有很高的效率,同时引入了随机部分的变量赋值决策。

结束语 本文分析了 WP 算法的基本原理,给出了算法的局部改进策略,并在 RB 实例集上进行了实验验证,结果表明在某些 RB 实例集上,改进后的 WP 算法比原算法具有更好的效果。此外,针对 WP 算法不收敛的情况下不能有效求解公式的特点,将 WP 算法与 DPLL 算法组合使用,从而减少了回溯计算的次数,有效地弥补了 WP 算法的不足,通过 RB 实例集上的验证表明,HPD 算法具有较好的求解效果。

参考文献

- [1] 黄文奇,金人超.求解 SAT 问题的拟物拟人算法——Solar[J].中国科学(E 辑),1997,27(2):179-186
- [2] 李未,黄文奇.一种求解合取范式可满足性问题的数学物理方法[J].中国科学(A 辑),1994,24(11):1208-1217
- [3] 梁东敏,吴晔,马绍汉.一种求解结构 SAT 问题的高效局部搜索算法[J].计算机学报,1998,21(增刊):92-97
- [4] 刘涛,李国杰.求解 SAT 问题的分级重排搜索算法[J].软件学报,1996,7(4):201-210
- [5] Gu J, Purdom P W, Franco J, et al. Algorithms for satisfiability (SAT) problem: A survey[J]. Discrete Mathematics and Theoretical Computer Science: Satisfiability (SAT) Problem, 1997, 35:19-152
- [6] Selman B, Kautz H, Cohen B. Local search strategies for satisfiability testing[J]. Discrete Mathematics and Theoretical Computer Science, 1996, 26:521-531

将情感改变为相反极性的情感,从而一定程度上提高情感识别的准确率。

5 实验流程和结果

本文采用的实验语料来自 DUTIR 的情感语料库^[15],选择其中的儿童文学期刊,共计 26757 个句子,近 80 万字。其中训练语料 17457 句,测试预料 9300 句。依据上述语料,实验分为 3 种类型,采用单句评估和多句联合评估的方法^[14]对比实验结果。方法一是目前比较常用的关键字匹配,计算各类情感词汇的词频。方法二选择一定的特征,利用条件随机域识别句子的情感^[14]。方法三是在方法一的基础上引入情感极性的处理,方法四在方法一的基础上加入标记语特征,方法五在方法一的基础上加入恐惧图式,方法六是在方法一的基础上加入愤怒表现图式,方法七是在方法一的基础上加入所有图式,即上面的情感认知模型。实验结果如表 2 所列。

表 2 7 种方法的实验结果

准确率	方法一	方法二	方法三	方法四
微平均	0.4130	0.4314	0.4253	0.4406
喜	0.5056	0.3534	0.5023	0.4993
怒	0.1108	0.4744	0.1307	0.1477
哀	0.1686	0.1202	0.1655	0.1696
惧	0.1625	0.0092	0.1625	0.1612
恶	0.2409	0.4838	0.3156	0.4542
惊	0.1022	0.0895	0.1022	0.1118
平静	0.6284	0.6872	0.6303	0.6085
准确率	方法五	方法六	方法七	
微平均	0.4202	0.4345	0.4462	
喜	0.4916	0.4944	0.4858	
怒	0.1307	0.7244	0.8949	
哀	0.1562	0.1654	0.1665	
惧	0.2840	0.1572	0.2761	
恶	0.2692	0.2611	0.2941	
惊	0.0990	0.1022	0.1150	
平静	0.6189	0.6237	0.5997	

从实验结果可以看出,采用文本情感认知模型的方法七实验效果较好,分别加入情感极性的调整、恐惧图式、标记语特征和愤怒表现图式,对识别的效果都有一定幅度的提高。其中加入愤怒表现图式,怒的准确率提高了近 70%,但是因为愤怒类型的语句总数较少,在测试语料中所占的比例也较少,所以对整体情感识别的准确率提高幅度较小。方法二采用条件随机域的方法标注结果,其准确率与方法一比较提高了近 2%,从情感连续性的角度考虑,条件随机域的方法比方法一高 5%。加入恐惧图式后,恐惧类比方法一提高了 12%,尽管恐惧类的语句较少,总的准确率也提高近 1%,说明情感图式的加入对情感起着积极的修正作用。

结束语 本文将认知语言学和心理学的知识引入到情感识别中,尝试从一个新的角度来解决现有的问题。以拉扎勒斯的认知-评价理论和认知语境为理论基础,提出了一个新的

文本情感认知模型。通过实验证明,该模型在文本情感识别中有一定的作用。加入各个图式对相应类型的情感识别提高较大,但是也会对其他类型的情感产生一些影响,总体来说都有些提高。

文中提出的情感图式还较少,实现方式主要还是以关键字匹配和词语搭配为主要方法,利用语用推理的知识较少,从而使模型的灵活性较差。对情绪心理学和认知语用学方面知识了解得不够透彻,使知识的语用停留在表层水平。今后将继续深入研究认知语言学和心理学的相关知识,以便在更广阔的范围内为文本情感识别提供理论支撑;加强图式的推理功能,加入更多的百科知识,以增强情感图式的灵活性。

参考文献

- [1] Picard R W. Affective Computing[M]. Cambridge, MA: MIT Press, 1997
- [2] 孟昭兰. 情绪心理学[M]. 北京:北京大学出版社, 2005(3)
- [3] 梁建宁. 心理学导论[M]. 上海:上海教育出版社, 2006(10)
- [4] Matsumoto K, Ren Fuji, Kuproiwa S, et al. Emotion Estimation Algorithm Based on Interpersonal Emotion Included in Emotional Dialogue Sentence[A]//Mexican International Conference on Artificial Intelligence[C]. 2007; 1035-1045
- [5] 熊学亮. 认知语用学概论[M]. 上海:上海外语教育出版社, 1999
- [6] Dan S, Deirdre W. Relevance; Communication and Cognition [M]. Foreign Language Teaching and Research Press, 2001
- [7] 熊学亮. 认知语用学[M]. 上海:上海外语教育出版社, 2001
- [8] 何自然, 冉永平. 关联理论——认知语用学基础[A]//语用与认知[C]. 北京:外语教学与研究出版社
- [9] 何自然. 语用学与英语学习[M]. 上海:上海外语教育出版社, 1998
- [10] 熊学亮. 语用学和认知语境[J]. 外语学刊, 1996, 3:12-18
- [11] Su Wen-poh. Personality and Emotion-Based High-Level Control of Affective Story Characters[J]. IEEE transactions on visualization and computer graphics, 2007, 13(2): 281-293
- [12] De Raad B. The Big Five Personality Factors, The Psycholexical Approach to Personality[M]. Hogrefe & Huber Publisher, 2000
- [13] Jiang Peilin, Xiang Hua, Ren Fuji, et al. The Framework of Mental State Transition Analysis[A]//Mexican International Conference on Artificial Intelligence[C]. Mexico, 2007; 1046-1055
- [14] 徐琳宏, 林鸿飞. 基于语义特征和本体的语篇情感计算[J]. 计算机研究与发展, 2007, 44(S2): 356-360
- [15] 徐琳宏, 林鸿飞. 情感语料库的构建[J]. 中文信息学报, 2008, 22(1): 116-122
- [16] 徐琳宏, 林鸿飞. 基于语义理解的文本倾向性识别机制[J]. 中文信息学报, 2007, 21(1): 96-100

(上接第 181 页)

- [7] Davis M, Putnam H. A computing procedure for quantification theory[J]. JACM, 1960, 7(3): 201-215
- [8] Davis M, Logemann G, Loveland D. A machine program for theorem proving[J]. Communications of the ACM, 1962, 5(7): 394-397
- [9] Mézard M, Parisi G, Zecchina R. Analytic and algorithmic solution of random satisfiability problems[J]. Science, 2002, 297: 812-815
- [10] Maneva E, Mossel E, Wainwright M. A New Look at Survey

Propagation and its Generalizations[J]. Journal of the ACM, 2007, 54(4): 1089-1098

- [11] Xu K, Hemery F, Boussemart F, et al. Random constraint satisfaction: Easy generation of hard(satisfiable) instances[J]. Artificial Intelligence, 2007, 171: 514-534
- [12] Parisi G A. Backtracking survey propagation algorithm for K-satisfiability[OL]. <http://arxiv.org/abs/cond-mat/0308510> v1, Aug. 2003
- [13] 李韶华, 张健. Survey propagation: 一种求解 SAT 的高效算法[J]. 计算机科学, 2005, 32(1): 132-137