3P1-3in2

確率論理推論の推論結果を端的に説明するルール集合の特定

佐々木 健太郎 Kentarou Sasaki　アンドラーデ ダニエル Daniel Andrade　渡邉 陽太郎 Yotaro Watanabe　定政 邦彦 Kunihiko Sadamasa

日本電気株式会社 情報・ナレッジ研究所
NEC Knowledge Discovery Research Laboratories

Probabilistic Logical Inference (PLI) Systems, like Markov Logic Networks (MLN), enable logical reasoning under uncertainty.However, in contrast to pure logical reasoning systems (like Prolog which provide a proof path), explaining the inference result to the user is difficult. In PLI all logical rules might be related to the outcome, although only few can be considered to be relevant enough. In this work, in the first part, we try to formalize which rules should be considered as relevant, and define criteria for selecting appropriate rule subset. Secondly, we propose a greedy algorithm which approximately solves the problem defined in the first part. Furthermore, we illustrate our proposed method using an example.Finally, we discuss open problems and future applications.

1. はじめに

確率論理推論手法は, ルールを論理式で明示的に記述でき, 確率を用いることで不確実性を表現できる利点がある. 代表的なものとしては,Markov Logic Network (MLN [6]) や Probabilistic Soft Logic (PSL [1, 4]) があり, ルール集合 L と観測 $\mathcal{O}$ のもとでクエリ Q の成立確率 $P(Q|\mathcal{O}, L)$ を求める推論が可能である.

このとき, 推論の根拠を端的に把握するにはどうすればよいだろうか. MLN を用いる場合の例あげる. あなたが落語世界の桶屋だとして, いまの社会状況 (観測) で桶が売れる確率を決める要因 (ルール) を端的に把握したいとしよう. 予めルールの候補は知っているものとする (表 1.).

F_1	1.2	風が吹く → 埃が舞う
F_2	0.8	埃が舞う → ねずみの増加
F_3	1	ねずみの増加 → 桶が売れる
F_4	1.5	ねずみの増加 → 食中毒の増加
F_5	1.1	衛生環境悪化 → ねずみが増加

表 1: ルール

観測:　風が吹く, 衛生環境悪化, 食中毒の増加
クエリ:　桶が売れる

既知の社会状況を観測,「桶が売れる」というイベントをクエリとして表現すれば, 現状での桶の売れる確率は求められる. この根拠を端的に把握するには, どのルールが推論結果を導くかが分かれば良い. Prolog のように, 確率のない推論では, 使用したルールとそうでないルールが明確に分かれる. 一方で確率論理推論では, クエリや観測にあたる確率変数と連結なグラフ上のルール全てが推論に関わる. そのうえ, 各ルールの結果への影響の仕方は複雑な数式を介しているため各ルールがどれほど確率に影響したかは非自明である. そこで, 結果を導出するのに重要な役割を果たしたルールを何らかの基準で絞り込み, 根拠とする方法が考えられる. このとき推論結果の導出にどのルールと観測が重要なのかを選別する基準が必要である.

本稿ではこの基準を定義し, 実際にルールを絞り込むアルゴリズムも提案する. まず確率論理推論手法について軽く述べる. 次に根拠となるルール集合の満たすべき要件を考察し, 解法を定式化する. その後ルール選択基準を定義し, それを用いてルール部分集合を選択する具体的なアルゴリズムを示し, 上記の例に対する出力結果を考察する. 最後に提案法の応用例と将来の課題を述べる.

連絡先: k-sasaki@ez.jp.nec.com

2. 確率論理推論手法

確率論理推論手法とは, 論理式をもとにモデルを作り, 確率的な推論を行う手法である. 代表的なものとして, Markov Logic Network (MLN [6]) , Probabilistic Soft Logic (PSL [1, 4]) Bayesian Logic Program (BLP [3]) などがある. MLN と PSL は共に, 論理式をテンプレートとして Markov 確率場を定義し, その上で推論を行う手法である. BLP は, 論理式をテンプレートとして,Bayesian Network を定義し, その上で推論を行う手法である. その他, 近年提案様々な手法が提案されている.Problog[2], ProPPR[8, 9] などがその一例である. Problog は Prolog をベースにに確率を導入した手法である. ProPPR も Prolog をベースに確率を導入するが, 推論にページランクアルゴリズムを用いている.

これらの手法は, MAP 推論や事後確率を求める推論など, 一般に複数の推論を行える. 本稿で対象とする推論は, ルール集合 L と観測 $\mathcal{O}$ のもとでクエリ Q の成立確率 $P(Q|(O), L)$ を求める推論である. 本稿で対象とする確率論理推論は, この推論を行う任意の確率論理推論である. 動作例には MLN を採用するが, 提案法は MLN に限らず適用できる.

3. 定式化

より少ないルール部分集合 $L' \subseteq L$ であって推論結果をおよそ再現可能なものを特定するのが課題である. ルールの選択基準を定義する前に,L' の満たすべき必要条件を考察する. 推論結果を再現できるには, 次の条件は必要である.

条件 1. $|P(Q|\mathcal{O}, L) - P(Q|\mathcal{O}, L')|$ を評価できる

ところで MLN や PSL では, ルールには重みが付いており, ルールの重要度を表すとしばしば説明される. そこでこの重みをルールの選択基準に採用してみよう.

手法 1. 重みの絶対値の大きいルールを重要なルールと考え, 重みの絶対値の小さいルールから順にルール集合 L からルールを減らしていく方法.

例えば MLN の場合, この手法 1 ではクエリの事後確率に対する条件 1 は満たされない. 表 1. のルールの一部を用いて例をあげる. ルール集合を $L = \{F_3, F_4, F_5\}$ とし (表 3.) , 観測とクエリは以下のようにする.

F_3	1	ねずみの増加 → 桶が売れる
F_4	1.5	ねずみの増加 → 食中毒の増加
F_5	1.1	衛生環境悪化 → ねずみが増加

表 2: ルール 2

観測: 衛生環境の悪化, 食中毒の増加
クエリ: 桶が売れる
このとき, 次式が成り立ち, 重さの大きいルールを残して推論した場合のほうが, 確率の差異が大きい事がわかる. *1 *2

$$P(Q|\mathcal{O}, L) = P(Q|\mathcal{O}, \{F_3, F_4\})$$
$$P(Q|\mathcal{O}, L) \neq P(Q|\mathcal{O}, \{F_3, F_5\})$$

こうなる原因は, F_5 の真理値に対応する確率変数と, 他のルールの真理値に対応する確率変数とが観測のもとで条件付き独立になるためである. 従って, ルール F_5 は, あってもなくても, 事後確率に影響しないといえる.

本稿では, 条件 1 そのものをヒントにルール選択基準を定義する. さらに推論の根拠として用いるため, なるべくコンパクトなルール集合を得たい. そこで確率の差異を小さく保ったまま, 可能な限り小さい部分集合を求める最適化問題として定式化する.

$$\begin{aligned} &\text{minimize} \quad Card(L') \\ &\text{subject to} \quad |P(Q|\mathcal{O}, L) - P(Q|\mathcal{O}, L')| \leq \varepsilon \end{aligned}$$

ただし,$Card((L')$ は, 集合 L' の要素数を表し, ε はユーザ定義のパラメータである.

確率の差の値をまったく変えずに, もしくは, 少しの誤差を許して効率的な推論を行う研究は, 推論高速化の文脈で多くの先行研究がある. その中には, 部分グラフに制限して推論する手法もある ([7]). 本稿の提案法も部分グラフをとる手法といえるが, 高速化でなく根拠の把握が目的であるため, ルール単位でグラフを切り出し, ルール数が最小になるようにする.

4. 提案手法

提案法は, ルールの重要度を, 推論結果に対する影響度でもって定義して, これをルール選択基準に用いる. 初めに記号の定義を行う.

4.1 記号の定義

$L = \{F_1, \ldots. F_{N_L}\}$ をルール, すなわち論理式の集合とする. MLN や PSL で現れるような重み付きルールも, 本稿では単にルールと呼ぶ. 各ルール F_i はグラウンドされている, すなわち, 全ての変数には定数が代入されてるものとする, 観測とは, いくつかの原始命題への真理値の指定であり,$\mathcal{O}$ で表す. 本稿の例では, 簡単のため, 観測は特定の原始命題に真の真理値を割り当てるものとし, $\mathcal{O}$ を単に原始命題の集合とみなす. クエリは, それが真となる確率を知りたい原始命題からなる集合であり,Q で表す. ルール集合 L' を用いた場合の, 観測 $\mathcal{O}$ のもとでの, クエリ Q の事後確率を $P(Q|\mathcal{O}, L^{'})$ で表す.

4.2 推論におけるルールの重要度

4.2.1 ルールの重要度

ルール集合 L, ルール $F \in L$ が与えられたとき. $P(Q|\mathcal{O}, L)$ と $P(Q|\mathcal{O}, L \setminus \{F\})$ の差を $D_L(F, \mathcal{O}, Q)$ で表し, その絶対値を, ルール F の重要度と定義する.

$$\begin{aligned} D_L(F, \mathcal{O}, Q) &:= P(Q|\mathcal{O}, L) - P(Q|\mathcal{O}, L \setminus \{F\}) \\ F\text{ の重要度} &:= |D_L(F, \mathcal{O}, Q)| \end{aligned}$$

この値は, そのルールを取り除いた際の推論結果の変化する度合いであるため, 推論結果に対するルールの影響度と解釈できる.

4.2.2 ルール部分集合の重要度

ルール部分集合 $L" \subseteq L$ の, 推論結果に対する重要度を定義する. $L' = L \setminus L"$ とし, $P(Q|\mathcal{O}, L)$ と, $P(Q|\mathcal{O}, L')$ の差を $D_L(L", \mathcal{O}, Q)$ で表し, その絶対値を $L"$ の重要度と定める.

$$\begin{aligned} D_L(L", \mathcal{O}, Q) &:= P(Q|\mathcal{O}, L) - P(Q|\mathcal{O}, L') \\ \text{L"の重要度} &:= |D_L(L", \mathcal{O}, Q)| \end{aligned}$$

$|D_L(L", \mathcal{O}, Q)|$ の値が小さい場合には, L' に属するルールのみで推論すると L の場合と近い確率値を得られることが分かる. この意味で, このとき $L"$ に属するルールは取り除いても推論結果が大きく変化しないルールであり, 推論結果に対する影響度が小さいと言える.

4.2.3 ルールの重要度とルール集合の重要度の関係

$L" \subseteq L$ に対し,$L" = \{F_i\}_{i=1,\ldots,n}$ と添数付ける. また,$\{L_i\}_{i=1,\ldots,n}$ を次のように定義する.

$$L_0 := L, \;\; L_i := L_{i-1} \setminus \{F_i\}$$

このとき,

$$D_L(L", \mathcal{O}, Q) = \sum_{i=1}^{n} D_{L_{i-1}}(F_i, \mathcal{O}, Q)$$

が成立し, 添数 i の付け方によらない.

4.3 ルールを絞り込む問題の定式化

以上の記号のもとで, 冒頭の最適化問題は次のように表される.

$$\begin{aligned} &\text{minimize} \quad Card(L') \\ &\text{subject to} \quad |D_L(L \setminus L', \mathcal{O}, Q)| \leq \varepsilon \end{aligned}$$

この定式化は, 推論に対する影響度が閾値 ε 以下となるルール部分集合 $L"$ であってなるべく大きいものを特定し, 残るルール集合 $L' = L \setminus L"$ を求める. ゆえに L' のみを用いて再推論した際の, 確率の差異は ε 以内に保証される.

4.4 アルゴリズム

この最適化問題の真の解を求めるには $Card(2^L)$ 回の推論が必要となり, 計算量に困難がある. また「$D_L(F, \mathcal{O}, Q)$ の値が正で絶対値の大きい F と, 負で絶対値が大きい F を削除したことで, 偶然確率の変化量の総量 $D_L(L \setminus L', \mathcal{O}, Q)$ が小さくなる」解 L' は避けたい. 今回は貪欲法によって近似的に解き, この両問題に同時に対処する. ルールの集合から, ルールを一

*1 ここではクエリを残すために, クエリを含むルールは必ず残すようにしている.

*2 実際に Alchemy-2 ([5]) を用いて計算すると, $P(Q|\mathcal{O}, L) = 0.625987, P(Q|\mathcal{O}, \{F_3, F_4\}) = 0.625987, P(Q|\mathcal{O}, \{F_3, F_5\}) = 0.593991$, となる.

つずつ削っていき, 制約を満たす限り最小のルール数になったところで削除を停止する方法である.

貪欲法のアルゴリズムが,n 回のルール削除ののちに終了したとして, ルール集合が次のように減っていったとする.

$$L = L_0 \supseteq L_1, \ldots, \supseteq L_n = L'$$

最初のルール集合が L で, 最終的なルール集合が L' である. また, 各段階で削除されるルールを $F_1, F_2, \ldots, F_n$ とする. このとき, 制約条件に現れる $D_L(L \setminus L', \mathcal{O}, Q)$ の式は,

$$D_L(L \setminus L', \mathcal{O}, Q) \quad = \quad \sum_{i=1}^{n} D_{L_{i-1}}(F_i, \mathcal{O}, Q)$$

となる. ゆえに制約は次のように変形される.

$$|\sum_{i=1}^{n} D_{L_{i-1}}(F_i, \mathcal{O}, Q)| \leq \varepsilon$$

以上のもとで最適化問題を近似的に解くアルゴリズムを Algorithm1 に示す. SD はその段階までの, その段階までの各ステップの $D_{L_{j-1}}(F_j, \mathcal{O}, Q)$ の総和 (sum of differences) であり, その段階での $D_L(L \setminus L_i, \mathcal{O}, Q)$ の値である.

```
Algorithm 1
  input Q, O, L, ε
  L” ← φ
  SD ← 0
  while min_{j:F_j∈L\L”} |D_{L'}(L” ∪ {F_j}, O, Q) + SD| ≤ ε do
    j = argmin_{j:F_j∈L\L”} |D_{L'}(L” ∪ {F_j}, O, ) + SD|
    SD = D_{L'}(L” ∪ {F_j}, O, Q) + SD
    L” = L” ∪ {F_j}
    if L \ L” = φ then
      break
    end if
  end while
  return L' = L \ L”
```

アルゴリズムに関して, いくつか注釈を述べる. まず, 本アルゴリズムは前述のとおり近似解法である. つぎに, $L \setminus L'$ に属する任意のルール F_i に対して, $|D_{L_{i-1}}(F_i, \mathcal{O}, Q)| \leq 2\varepsilon$ が保証される. これはアルゴリズムの 4 行目の不等式から分かる. ゆえに「$D_L(F, \mathcal{O}, Q)$ の値が正で絶対値の大きい F と, 負で絶対値が大きい F を削除したことで, 偶然確率の変化量の総量が小さくなる」ということは起こらない. また, ある整数 ε' を指定して, 各 F_i に対して $|D_{L_{i-1}}(F_i, \mathcal{O}, Q)| \leq \varepsilon'$ としたい場合, つまり個別のルールに対する変化量をもっと小さく留めたい場合はこれを条件に追加すればよい. すなわち,while 条件部の min を検索する対象のルールとして, 上記条件を満たすものに限るのである. 最後に, 実際の利用では削除対象のルールに制約を設たほうが良い場合がある.「クエリを含むルールは削除候補にしない」や「観測を含むルールは少なくともひとつは残す」などの制約である. 本稿の例でも, クエリを含むルールは削除しない制約をおいている.

4.5 提案法の変形

4.5.1 ルール部分集合の重要度のバリエーション

ルール部分集合の重要度 $D_L(L”, \mathcal{O}, Q)$ を, L を用いた推論に対する各ルール $F \in L$ の重要度 $D_L(F, \mathcal{O}, Q)$ の総和とする方法も考えられる.

$$D_L(L”, \mathcal{O}, Q) := \sum_{F \in L”} D_L(F, \mathcal{O}, Q)$$

こうすると, アルゴリズムの while ループは一回で済むという利点がある. 一方で, $|D_L(L”, \mathcal{O}, Q)| \neq |P(Q|\mathcal{O}, L) - P(Q|\mathcal{O}, L')|$ となり, このままでは確率の差異の保証は困難である. そのため本稿では先に定義した重要度を採用した.

4.5.2 もう一つの定式化

推論の根拠を端的に把握したい状況には, ルールを何件程度に絞り込みたいか予め分かる場合がある. C 件程度のルールなら目視可能だから,C 件以内に絞りたいという状況である.

最小化対象の関数と制約を入れ替えることで, 推論結果を可能な限り変化させずに, ルールを所望の件数に絞り込むことができる.

$$\begin{array}{ll} \text{minimize} & D_L(L \setminus L', \mathcal{O}, Q) \\ \text{subject to} & Card(L') \leq C \end{array}$$

この定式化を解いた後の $D_L(L \setminus L', \mathcal{O}, Q)$ の値が大きい場合には, C 個のルールでは推論をうまく説明できないと解釈することができる.

5. 提案法の動作と出力例

冒頭のルール (表 1.) および, 観測とクエリに対して, 提案法の動作例を示す. ただし, クエリを含むルールは削除しないという制約をおく. 本節を通じ, 確率論理推論手法としては MLN を考え, 確率の計算には alchemy-2 ([5]) を用いている. 元の確率値は $P(\text{風} \mid \text{衛}, \text{風}, \text{食}, L) = 0.682982$ である. ルールに現れる述語を, 語頭の一文字で略記している.

$\varepsilon = 0.05$ とした場合に提案法によって順にルールが削除されていく様子を図 5. に示す.

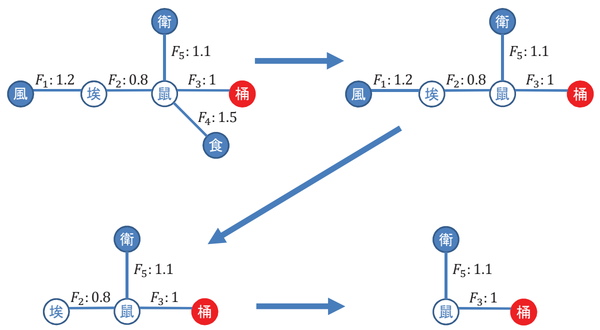

図 1: 提案法の動作

F_3	1	ねずみの増加 → 桶が売れる
F_5	1.1	衛生環境悪化 → ねずみが増加

表 3: $L_{\text{proposed}} = \{F_3, F_4\} : \varepsilon = 0.05$ としたとき提案法の選んだルール部分集合 ($\varepsilon = 0.05$)

左上のグラフがもとの L に対応するグラフであり, 右下のグラフが提案法の出力に対応する部分グラフである. 提案法の出力を $L_{proposed} = \{F_3, F_4\}$ とする (表 3). このときの確率は $P(\text{風} \mid \text{衛}, L_{\text{proposed}}) = 0.645985$ である. 確率の差の絶対値はもちろん, $\varepsilon = 0.05$ 以下になる (具体的には 0.036997).

いっぽう, 重みを基準にルールを 2 つ選択した場合, $L_{weight1} = \{F_3, F_5\}$ となり (表 4), 図 5. の左のグラフが対応する. このときの確率は $P(\text{風} \mid \text{食}, L_{\text{waight}}) = 0.593991$ であり, その変化量は 0.088990 となり, もとの確率との乖離は大きくなっている. 重み基準の場合に, ルールをもう一つ増やす

図 2: 重みを基準にする方法の出力. 左右それぞれ, ルールを 2 つおよび 3 つ残した場合のルール集合に対応するグラフ. ルール集合 $L_{weight1} = \{F_3, F_5\}, L_{weight2} = \{F_1, F_3, F_5\}$ に対応する.

F_3	1	ねずみの増加 → 桶が売れる
F_4	1.5	ねずみの増加 → 食中毒の増加

表 4: $L_{\mathrm{waight}} = \{F_3, F_4\}$: 重みの絶対値を基準に選んだルール部分集合 (上位 1 件)

と $L_{weight2} = \{F_1, F_3, F_5\}$ を得る. 対応するグラフは図 5. の右である. このときの確率は $P(風 \mid 食, L_{\mathrm{waight2}}) = 0.593991$ のままであり, 差の絶対値も変化ない. 一方, 仮に提案法で 3 つのルールを残した場合にはルール集合は $L_{propose2}$ となり, 確率の差の絶対値は 0.029997 と, やはり比較的小さい.

定式化の狙いどおり, 提案法の方が, ルール数が同じ場合には, 元の確率をよく再現するものが得られることが確認される. 言い替えると, 確率の差異が同程度の場合には, 提案法の方がコンパクトなルール集合を出力するといえる.

6. 展望

本節では, 提案法の応用の可能性と, 提案法の課題を述べる.

6.1 応用

提案法によって, 確率論理推論手法の内部で各ルールがどれほど結果に影響するかを把握することができる. 具体的には以下のような応用シーンが考えられる. ひとつはルールないしはルールの重みのデバック用途である. すなわち, ルールの重要度を算出することによって, 予め人が重要と考えていたルールが「そのルールの有無によって確率に大きな影響を与える」かどうかを確かめられる. もうひとつは推論に重要な役割を果たすルールを発見する用途である. すなわち, ルールの重要度を算出することによって, 事前には重要と認識してなかったルールが「ルールの有無によって確率に大きな影響を与える」場合に気づくことができる.

6.2 課題

次に, 提案法の課題を述べる. まず, 提案法は何度も推論を行うため, 大規模なルール集合上の推論に対して提案法を適用する際には, 計算量の問題が生じる. またハイパーパラメータ ε 及び C の決め方が非自明である. 望ましくは, 使用ルールや応用シーンに合わせて適当な値を定める指針が必要である. 提案法の動作の分析も課題である. 観測のもとで確率に影響のないルールが提案法により削除されることは既に述べた. より一般にルールが削除される順序の特徴付けを見出せれば, ルールのより効率的な絞り込みにつながる可能性がある. 最後に, 提案法の出力を推論根拠として人に見せる用途に用いるためには, 提案法の重要度と人が考える根拠らしさの対応を分析することも課題となる.

F_1	1.2	風が吹く → 埃が舞う
F_3	1	ねずみの増加 → 桶が売れる
F_4	1.5	ねずみの増加 → 食中毒の増加

表 5: L_{waight2} : 重みの絶対値を基準に選んだルール部分集合 (上位 2 件)

7. おわりに

本論文では, 推論結果の誤差範囲を保証しつつルール部分集合を選択する方法を提案した. これにより, 純粋な論理推論と比較して動作の解釈が困難な確率論理推論の内部で, 各ルールがどれほど結果に影響するかを把握できるようになる.

一方で, それはあくまで「推論エンジン手法内での各ルールの重要度」を表し, 人間の感覚に対する説明性の高さと一致するか否か, 双方の乖離はどの程度なのかは非自明である.

確率論理推論を人間が使いこなすには, 提案法や発展手法の示す根拠と, 人間に対する納得性の関係を分析する必要がある.

参考文献

[1] Stephen H. Bach, Matthias Broecheler, Bert Huang, and Lise Getoor. Hinge-loss markov random fields and probabilistic soft logic. arXiv:1505.04406 [cs.LG], 2015.

[2] Luc De Raedt, Angelika Kimmig, and Hannu Toivonen. Problog: A probabilistic prolog and its application in link discovery. In *IJCAI*, volume 7, pages 2462–2467, 2007.

[3] Kristian Kersting and Luc De Raedt. 1 bayesian logic programming: Theory and tool. *Statistical Relational Learning*, page 291, 2007.

[4] Angelika Kimmig, Stephen H. Bach, Matthias Broecheler, Bert Huang, and Lise Getoor. A short introduction to probabilistic soft logic. In *NIPS Workshop on Probabilistic Programming: Foundations and Applications*, 2012.

[5] Stanley Kok, Marc Sumner, Matthew Richardson, Parag Singla, Hoifung Poon, Daniel Lowd, Jue Wang, and Pedro Domingos. The alchemy system for statistical relational {AI}. 2009.

[6] Matthew Richardson and Pedro Domingos. Markov logic networks. *Machine learning*, 62(1-2):107–136, 2006.

[7] Jude Shavlik and Sriraam Natarajan. Speeding up inference in markov logic networks by preprocessing to reduce the size of the resulting grounded network. In *Proceedings of the 21st International Jont Conference on Artifical Intelligence*, IJCAI'09, pages 1951–1956, San Francisco, CA, USA, 2009. Morgan Kaufmann Publishers Inc.

[8] William Yang Wang, Kathryn Mazaitis, and William W Cohen. Programming with personalized pagerank: a locally groundable first-order probabilistic logic. In *Proceedings of the 22nd ACM international conference on Conference on information & knowledge management*, pages 2129–2138. ACM, 2013.

[9] William Yang Wang, Kathryn Mazaitis, and William W Cohen. Proppr: Efficient first-order probabilistic logic programming for structure discovery, parameter learning, and scalable inference. In *Proceedings of the AAAI 2014 Workshop on Statistical Relational AI (StarAI 2014)*, 2014.