定理間の類似度の定式化について (Work in Progress)

中正　和久

山口大学大学院 創成科学研究科 工学系学域 知能情報工学分野

発表次第

1. モチベーション
2. 先行研究
3. 定理検索の課題
4. 定理類似度の定義
5. 定理類似度の応用
6. 今後の展望

賢い定理検索システムを作りたい

- 巨大化するライブラリ
- パターンマッチの限界

巨大化するライブラリ

- **Mizar Mathematical Library**
  - 8852 symbols, 12,114 definitions, 59,076 theorems, 3 million LOC
- **Further Lessons learned from Flyspeck Project (Thomas C. Hales)**
  1. Follow good software development practices for a medium sized project, including version control, a stable module system, careful control of mutable state, a manager (Mark Adams).
  2. Structure the project around a clean mathematical text.
  3. **Search for theorem names is a major difficulty. There are over 15,000 theorems in HOL Light + FLYSPECK and many dozens of specialized tactics. Better tools are needed.**
  4. Keep each lemma short and crisp: productivity plummets once lemmas reach a certain level of complexity.

G.Bancerek et.al: The Role of the Mizar Mathematical Library for Interactive Proof Development in Mizar. J Autom Reason. 61(1): 9–32. 2018

Thomas C. Hales: Lessons learned from the Flyspeck Project http://www-sop.inria.fr/manifestations/MapSpringSchool/contents/ThomasHales.pdf

パターンマッチの限界

- **論理式の同値変形はパターンマッチと相性が悪い**

1. $\forall x, y(A[x] \to B[y])$
2. $\forall x, y(\neg B[x] \to \neg A[y])$
3. $\exists x. A[x] \to \forall y. B[y]$

1〜3の全てを列挙するパターンマッチを構成するのは困難

- **特徴量に基づいた類似度判定アルゴリズムが必須**
  - そもそも「論理式同士が類似している」という判断基準は？

先行研究リソース

	数式検索	定理検索	自動定理証明
データベース	NTCIR-12 MathIR	-	TPTP (CADE) HolStep (Google)
アルゴリズム	パターンマッチ 特徴量抽出	パターンマッチ 機械学習(LSI)	パターンマッチ 特徴量抽出 機械学習
システム	Whelp Wolfram alpha	(Mizarに限定すると) findvoc MML query MML reference	Vampire E prover Otter SPASS 他多数

定理検索はデータベースが存在せず，アルゴリズム研究もあまり進んでいない

数式検索

- **データベース**
  - NTCIR-12 MathIR
    - https://ntcir-math.nii.ac.jp/
    - データセットは人力でタグ付け？
    - 数式であれば人間の直感で十分良い精度が出そう
- **アルゴリズム**
  - 構文木の部分パス集合(Path Indexing)を特徴量とする方法 (Yokoi, Aizawa: DML 2009:27-35)
  - シンボルの位置・構文木内での深さを特徴量とする方法(A. Asperti et al.: TYPES 2004: 17-32)
  - 構文木での2つのシンボル間の相対位置を特徴量とする方法 (Pattaniyil, Zanibbi: The Tangent Math Search Engine at NTCIR 2014)
- **システム**
  - Wolfram Alpha
  - Whelp

定理検索

- **データベース**
  - 存在しない
- **アルゴリズム**
  - パターンマッチング (正規表現 + α)
  - LSI (単語の出現回数を特異値分解して特徴量を得る方法) (Cairns: MKM 2004: 58-72)
- **システム**
  - findvoc : コマンドベース
  - MML query : パターンマッチング， Web型，包括的検索が可能
  - MML reference : シンボル検索，APIリファレンスのようなシステム

自動定理証明

- **データベース**
  - TPTP (CADE Competitionで利用)
  - HolStep (Google w/ C. Kalyszyk)
- **アルゴリズム**
  - Term Indexing
    - パターンマッチング(Path Indexing, Discrimination Treeなど)
  - Premise Selection (J.Urban, C.Kalyszyk, etc)
    - 機械学習(Naive Bayes, n-gram, k-NN, ランダムフォレスト, ページランク, DNN)
  - Strategy Selection (J.Urban, etc)
    - 節の特徴量抽出
- **システム**
  - Vampire, E, Otter, SPASS, ...
  - http://www.tptp.org/CASC/27/WWWFiles/DivisionSummary1.html

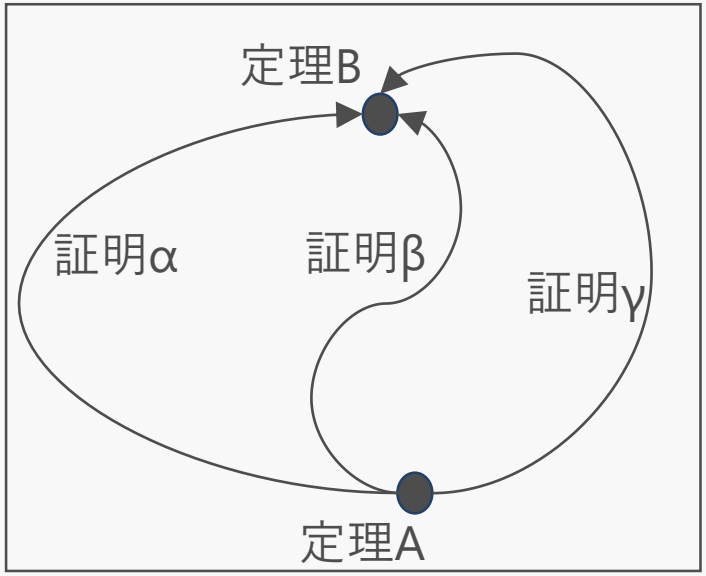
そもそも「定理の類似性」が定義されていない？

- **データベースがないと始まらない**
  - 検索アルゴリズムに機械学習を使うにしろ，検索アルゴリズムの性能評価をするにしろ，データベースが必須
- **データベースはどうやって作る？**
  - 通常，検索向けのデータベースは人力で作ることが多い
  - 定理の構文解析木は簡単な同値変形でも構造が大きく変化するので，人間の感覚によって類似性を判断するのは危うい
  - 定理のように論理的な対象を，音声認識や画像認識（＝人の脳による認識が絶対的な判断基準）人力で分類する必要はないのでは？
    - とはいえ，数学は論理のみで構成されているわけではないので，自然言語処理の手法や人力支援も有効に働くと思います
- **万人が納得する「定理の類似性」の定義とは？**
  - 論理に深く根ざしたものであるべき
  - 「定理」の論理性を担保するのは「証明」

定理の類似度が満たしてほしい性質

- **自然であること**
  - 万人が納得できる基準であること，理にかなっていること
- **計算容易性**
  - データベースを構築できる程度には計算が容易であること
  - 検索等で使う時には高速化された近似アルゴリズムで十分
- **有用性**
  - 検索・分類・クラスタリング・自動定理証明などに応用したい
- **距離公理**
  - $d(A, A) = 0$
  - $d(A, B) \geqq 0$
  - $d(A, B) = d(B, A)$
  - $d(A, C) + d(C, B) \geqq d(A, B)$
  - 対称性がない場合は擬距離

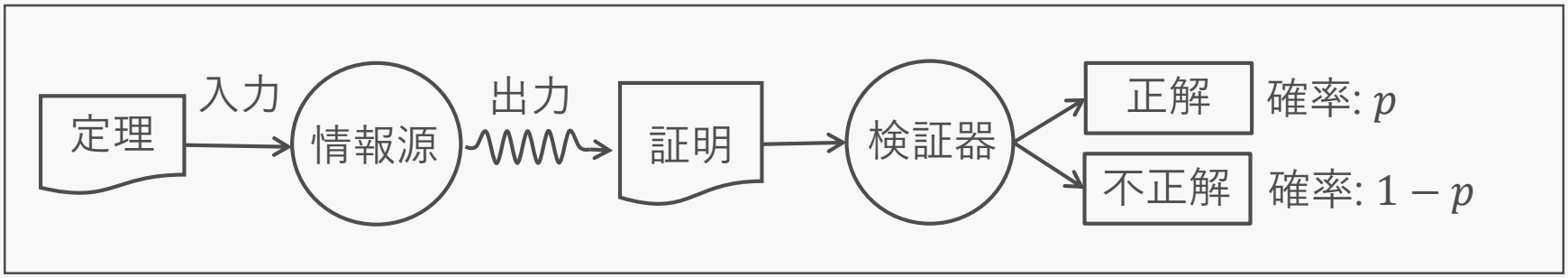
アイデア1：定理を点，証明を経路とする空間



- **定理A→定理Bは必ず証明を持つ**
  - 定理Bは単独でも証明できるので，定理A→定理Bもまた定理
- **証明は無数に存在する**
- **証明は非対称である**
  - 定理B→定理Aは別証明

- **類似した定理同士であれば，それらを結ぶ証明も短いはず！**

アイデア2：経路の距離 = 証明の情報量



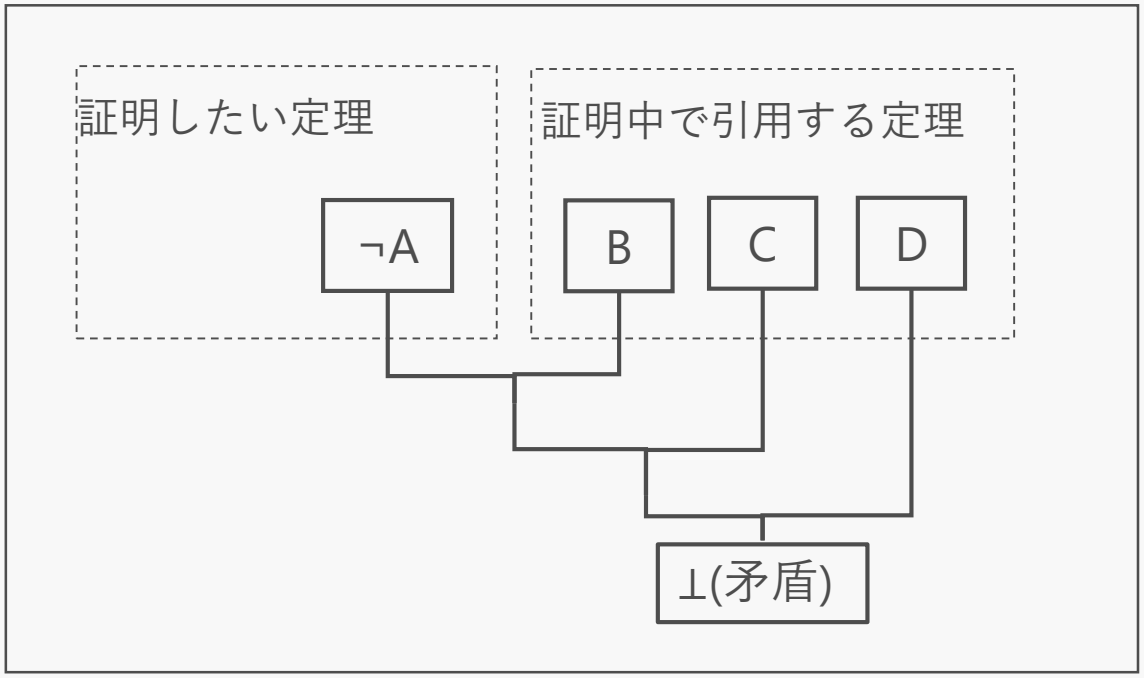
- **証明を出力する情報源モデル**
  - 定理を入力とし，証明を出力とする情報源モデル
  - 情報源モデルでは「ライブラリ=既知の定理」は固定
  - 上の場合は，定理の情報量は $-\log \boldsymbol{p}$ で定義される
- **情報源に仮定するバイアス**
  - 情報源に仮定するバイアスによって，証明の情報量が確定する
  - 少なくとも擬距離の性質を満たすようなバイアスを仮定したい

類似度を定義するフレームワークの候補

- **最短証明の情報量 ← これが最有力**
  - 最短証明は情報量最小という意味です
  - 証明の最短性を示すのは難しいが，既にある証明が最短に近いものだと仮定すればデータベースを構築できる程度には計算が容易
  - うまくバイアスを選べば，擬距離の性質を満たす
- **全証明の情報量の総和**
  - 経路積分のような考え方
  - 情報源モデルが正しい証明を出力する確率に対応する
  - 定理証明は無数にあるので計算が困難
  - 距離公理も満たさない
- **自動定理証明器(ATP)の実行時間**
  - 情報源=ATPという考え方
  - ATPの実行時間は正しい証明を発生する確率に反比例するだろうという仮定
  - 計算はATPを実行させるだけなので簡単（ランダム性があると安定しない）
  - 距離の性質は満たさない

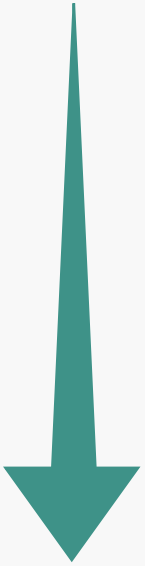
情報源バイアスの候補

- **簡単のため，情報源に一階述語論理の線形導出でUnificationするリテラルを間違えないようなバイアスを仮定する**
  - これで情報源は単にライブラリ内から定理(正確には節)を選び出す機械とみなせる



情報源バイアスの候補

1. **バイアスなし(=等確率で定理を発生)**
   - 情報量は証明の長さとほぼ比例
2. **定理毎に一定の発生確率を仮定 ← 今のところ最有力**
   - 既存の証明で引用される確率に応じて定理毎に発生確率を変更
3. **定理の発生確率にマルコフ性を仮定**
   - ある定理が別の定理のあとに引用される確率が高まる，など
4. **入力定理に応じて定理の発生確率を変動させる**



バイアスが強い

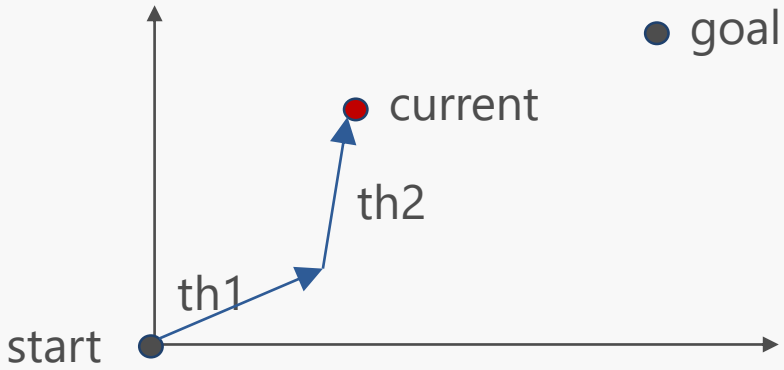
- **各候補の性質**
  - 1,2であれば擬距離の公理を満たす．(3,4は無理っぽい)
  - 1よりも2のほうが精密なバイアスなので性質が良いのでは？

定理の検索・分類

- **定理検索**
  - 機械学習，性能評価向けのデータセットの構築
  - 検索中に最短証明を探索するのは負荷が高すぎるので，証明情報量を近似する軽量アルゴリズムの開発が必須
- **定理分類・クラスタリング**
  - ライブラリの自動分類，ドキュメントの自動編纂などに利用する
  - 距離公理を満たせば既存の手法で十分な精度が出るはず
    - dが擬距離ならば，D(A,B) = (d(A,B) + d(B,A)) / 2 は距離となる

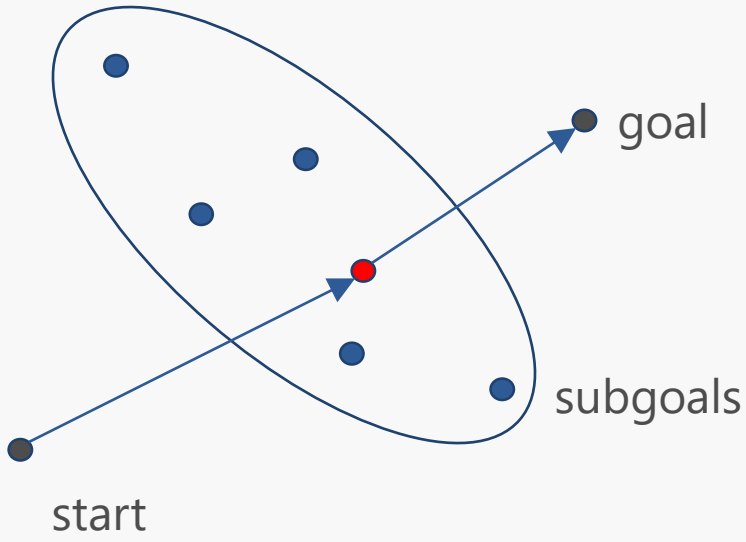
自動定理証明

- **評価関数**
  - 証明途中で定理が証明完了に近づいているかを距離推定で評価できるようになる→ **Strategy Selectionにおける疎な報酬の問題からの脱却**
- **線形近似**
  - ひとたび定理間に距離が導入されれば，SVDやWord2Vec, BERTなどによって定理空間を次元削減して定理を線形近似できる
  - 線形近似すれば引き算ができるようになるため，定理引用によってゴールにどのくらい近づくか計算可能となる
  - 証明から前後関係が失われるため，どの程度正確性が保てるかは要実験



自動定理証明

- **Sub-goal生成**
  - どのsub-goalが有力であるかを，距離推定によって評価する
    - d(start, sub-goal) + d(sub-goal, goal)が最小



定理ライブラリの品質測定

- **大きさ**
  - 定理の最小証明の情報量の総和
- **網羅性**
  - ライブラリAとライブラリBの比較
  - ライブラリAの網羅性 = ライブラリAをベースにライブラリBの定理を全て証明するための情報量の総和
- **直交性**
  - 「定理A →定理B」と「定理B」の証明の情報量が同じなら直交関係
  - これを全ての定理に対して検証する

理論面での研究

- **情報源モデルの精密化**
  - もっと良いバイアスを提案できないだろうか？
- **評価式の精密化**
  - $d(A_1 \wedge A_2, B) \leqq d(A_1, B) + d(A_2, B)$
  - $d(A_1 \vee A_2, B) \leqq \min(d(A_1, B), d(A_2, B))$

  のようなもの.
- **論理式への拡張**
  - 編集距離のようなものに拡張する？
- **導出原理以外への拡張**
  - 自然演繹，シーケンス計算，ラムダ計算
  - 高階述語論理(tactics)

定理検索システムへの適用

- データベースの構築
- 近似アルゴリズムの開発

おわり

- ご清聴ありがとうございました