

確率論理プログラミング言語PRISMとその 深層化T-PRISMの紹介

小島 諒介
(京都大学 医学研究科)



自己紹介

名前：小島 諒介

略歴：

2014年 東京工業大学 情報理工学研究科
修士課程修了

2017年 東京工業大学 博士（工学）

2017年～ 京都大学 医学研究科
人間健康科学系専攻
ビッグデータ医科学分野
奥野研究室
特定助教

興味：

人工知能の手法開発と実世界応用（特に時系列データ分析）

最近新たに取り組んでいる分野：医療データ解析/創薬

目次

- 背景・問題共有
 - リアルワールドデータ
 - リアルワールドのモデリング・アルゴリズム
- 確率論理プログラミング：P R I S M
- 深層確率論理プログラミング：T - P R I S M

背景：医療 × AI

理研・JST/神戸リサコン

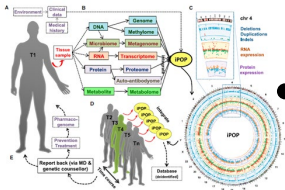
個人の健康を可視化・モデル化・予測できるシミュレータの開発

ヘルスケア・未病

健康状態の可視化・モデル化・将来予想
予防医学の実現



ライフビッグデータ



京大・JST/弘前COI

2,000項目×1,000人×14年の
健診ビッグデータ解析
新たな疾患概念構築により、
予防医学の可能性を切り拓く

文科省/ポスト「京」重点課題1

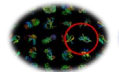
次世代スパコン ポスト「京」による
次世代の計算創薬の開拓

創薬

予防・先制医療

WG1. 未病・先制医療

- 健康診断データによる発症予測
- マイクロバイーム・オミクスデータ解析
- デジタルヘルス



ターゲット探索

リード探索

リード最適化

バイオアッセイ

前臨床試験

臨床試験

承認

薬物治療

WG3. 創薬テーマ創出

- 有望提携先や研究テーマの自動探索
- 標的分子探索
- ドラッグリポジショニング

WG4. 分子シミュレーション

- タンパク質立体構造・機能予測
- AIによるドッキング計算高度化
- 分子動力学計算によるAI活用
- AIを用いた高精度分子力場



WG6. トランスレーショナルリサーチ

- 非臨床データからのヒトADMET予測
- 疾患メカニズム解明・ブリッジング予測

WG2. 臨床・診断

- がんゲノム医療におけるAI活用
- シミュレーションによる細胞分離
- AIによる病理画像処理
- AIによる電子カルデ処理

WG7. バイオロジクス・製剤・ロボティクス

- バイオロジクス関連AI
- 結晶形・製剤関連AI
- 調剤ロボティクス

WG8. 治験・市販後・メディカルアフェアーズ

- AIによる治験の効率化
- 有害事象の情報基盤
- 製品Q&Aシステム
- アウトカムリサーチ・医療技術評価

バイオメディカル・基礎から臨床への開発プロセス

診断・治療

WG9. 知識ベース・NLP

- 知識ベース

知識ベース / 計算機サーバー

WG10. AI基盤

- ライフサイエンスのためのAI基盤

ライフインテリジェンスコンソーシアム (LINC)

110の団体 (アカデミア・製薬・IT) からなる
産官学コンソーシアムでAI創薬のプラットフォームを開発

NEDO/次世代人工知能・ ロボット中核技術開発

新薬開発を効率化・加速する
製剤処方設計AIの開発

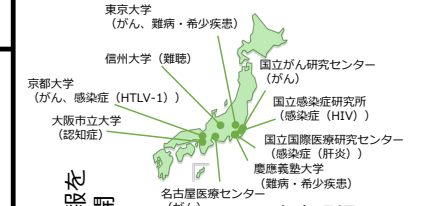
AMED/ 創薬インフォマティクス基盤構築

医薬品・ADMETの統合型データベース
を構築し、創薬支援に活用

理研/データプラットフォーム事業

AI創薬のためのデータ解析基盤の
構築

医療



11拠点を臨床ゲノム情報を
疾患横断的に収集し公開

臨床現場への
フィードバック



AIによる解釈

疾患ゲノムデータベース MGenD

AMED/臨床ゲノム

国内唯一の疾患横断的
臨床ゲノムデータベースを構築
ゲノム医療実装の基盤構築

AI関連プロジェクトにおける課題の分類

ヘルスケア・未病

創薬

医療

医療用 AI 開発プロジェクト

メンバー

医師・看護師・医療従事者・ケミスト
ケモインフォ・バイオインフォ
データサイエンティスト **etc**

データ

電子カルテ・公開データベース・
収集データ・実験データ

モデル・アルゴリズム

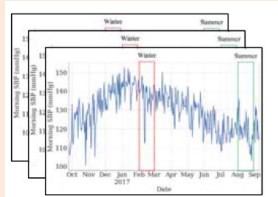
深層学習・統計モデル・その他機械学習・数理モデル

- ・ 人に関する課題
- ・ データに関する課題
- ・ モデル・アルゴリズムに関する課題

医療リアルワールドデータ

さまざまな形式（関係データ/時系列データ）の
不確実なデータ（欠損や外れ値などを含む）が存在している

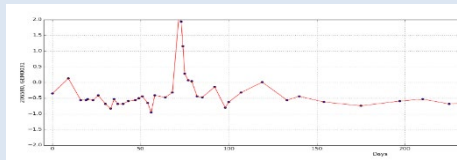
収集データ



- ・実験データ
 - ・研究計画に基づいた収集データ
- 欠損：少
データ量：少

電子カルテなどの 記録データ

- ・記録データ
- 欠損：多
データ量：多

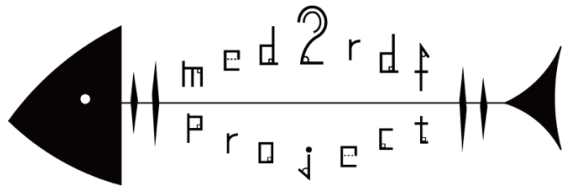


公共データ

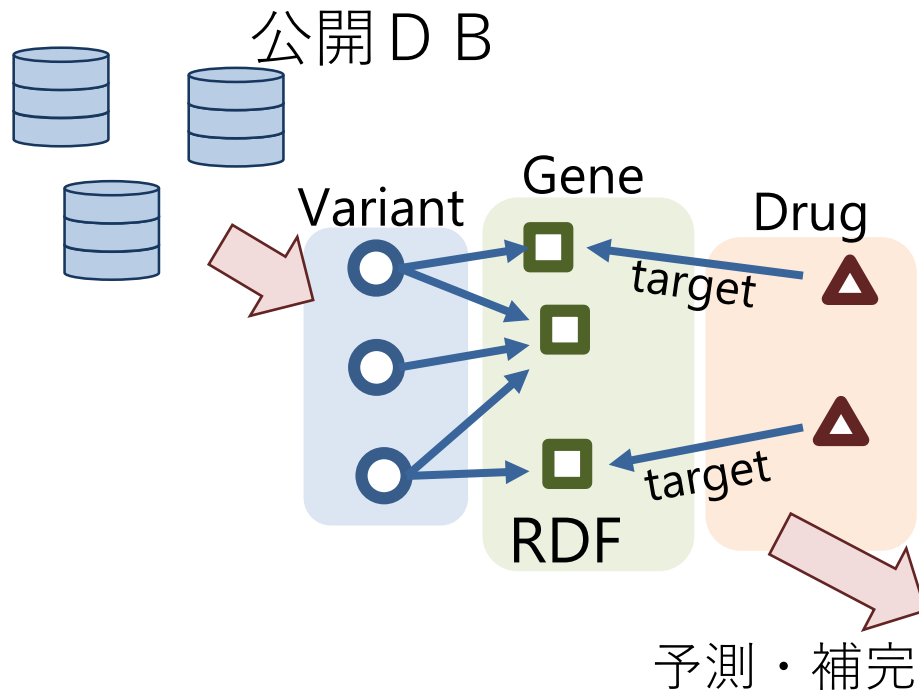


- ・薬剤情報
 - ・ゲノム関連情報
- データベース間の揺らぎ
アノテーション：少

公共データ活用の例：Med2RDF



ゲノム関連知識グラフを公開DBをから構築



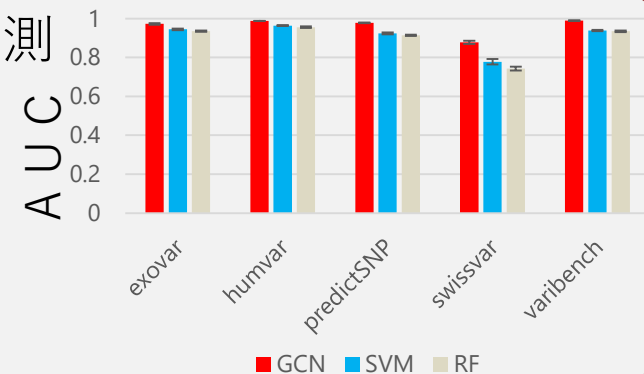
関係データを利用することで、
性能を上げることができる

<http://med2rdf.org/>

M. Kamada, T. Katayama, S. Kawashima, R. Kojima, M. Nakatsui, Y. Okuno:
Construction of Knowledge-base for Clinical Interpretation of Genomic Variants. Semantic Web
Applications and Tools for the Life Sciences (SWAT4LS), 2017.

Category	DB	Information
Pathogenicity of variant	ClinVar	Pathogenicity of variant
	COSMIC	Impact of variants in human cancer
	CIViC	Variants with published clinical evidence for cancer
Frequency in population	dbSNP	Genetic variation with MAF
	dbVar	Large scale genomic variation
	ExAC	Allel frequency in epoch population
Effect of variant on protein function	dbNSFP	Effect scores prediction by 21 bioinformatics tools
Drug interaction	DGIdb	Gene-drug interaction
Molecular interaction	HiNT	Protein- protein interaction from interactome resources
	INstruct	3D-protein interaction with structure

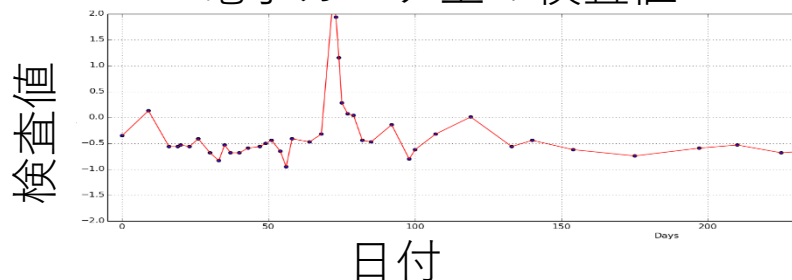
病原性予測



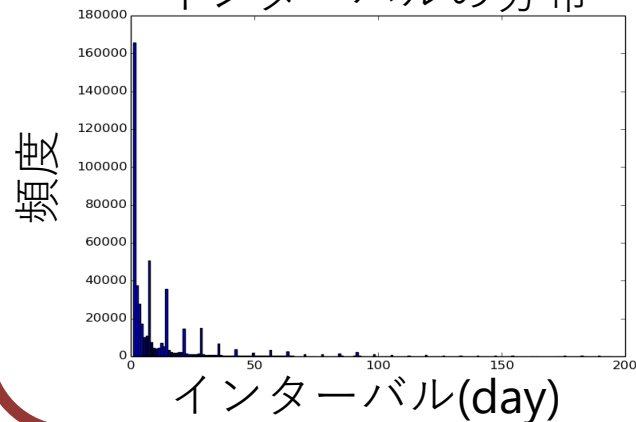
電子カルテ分析の例：HMMを用いた検査頻度の分析

対象データ：京大病院データ
がん患者：5264名

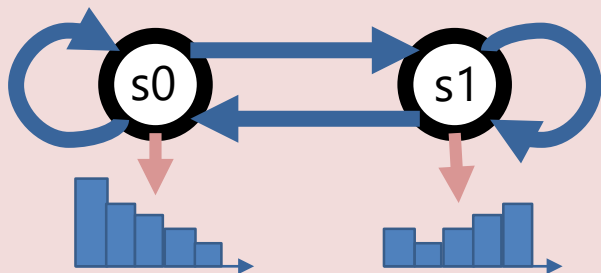
電子カルテ上の検査値



インターバルの分布



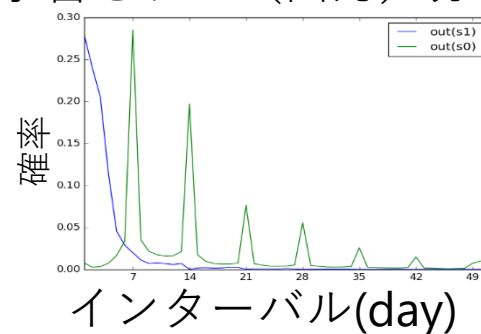
インターバルをHidden Markov Model (HMM)でモデル化



ランダムフォレストで検査値から
HMMの状態を予測：正答率 74.6 ± 0.2



学習された（出力）分布



状態の切り替わりを予測できる
→より少ない検査頻度で済ませられる可能性がある

収集データの例：血圧の変動予測

データ：

通院中の患者を対象に423人分の
毎日朝晩の家庭血圧を収集

血圧指標と管理基準

平均的な血圧 に関する指標

血圧の変動性 に関する指標

種別	基準値
家庭血圧 (週間平均で評価)	SBP: 135 / DBP:85 mmHg以上
診察室血圧 [1]	SBP: 140 / DBP:90 mmHg以上
診察室訪問間血圧差 (Visit-to-Visit) [2]	基準は未決定
日間変動性 (Day-by-Day) [3]	基準は未決定
季節変動 (Seasonal)	基準は未決定

血圧平均値: 5.11-6.65 mmHgで予測可能

血圧変動性: 0.67-0.70

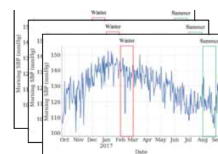
* 電子血圧計の精度基準は聴診に対して誤差平均5mmHg以内

H. Koshimizu, R. Kojima, K. Kario, Y. Okuno:
Prediction of blood pressure variability using deep neural networks.
In *International Journal of Medical Informatics*, Vol. 136 pp. 104067, 2020.

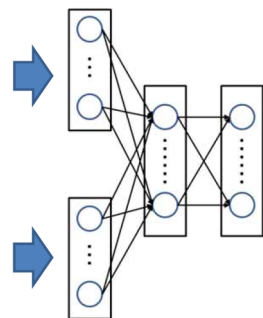
診察室情報

- 既往症
- 体重/身長
- 年齢
- 性別
- ...

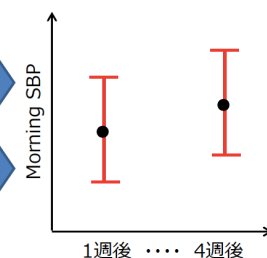
56日間の血圧



深層学習モデル



血圧の平均
+ 変動性の予測



予測対象	値	RMSE (mmHg)		SR	
		提案手法	ベンチマーク	提案手法	ベンチマーク
1週間後	平均	5.11	15.89	0.69	0.38
	最良	5.04	13.56	0.70	0.43
2週間後	平均	5.91	16.67	0.68	0.37
	最良	5.82	13.52	0.69	0.43
3週間後	平均	6.34	16.43	0.67	0.37
	最良	6.26	14.07	0.68	0.42
4週間後	平均	6.65	16.51	0.67	0.37
	最良	6.60	14.27	0.67	0.42

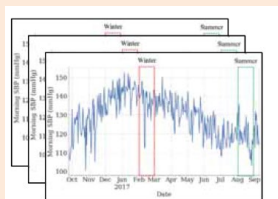
血圧の平均的な値とその変動性（標準偏差）を予測可能性を示した

[1] S. Umemura et al., Chap2, The Japanese Society of Hypertension Guidelines for the Management of Hypertension, HYPERTENS RES, 42, pp.1248–1281, 2019.
[2] Rothwell PM et al., Limitations of the usual blood-pressure hypothesis and importance of variability, instability, and episodic hypertension. Lancet.2010;375:938–48
[3] Kuriya M et al., Day-by-day variability of blood pressure and heart rate at home as a novel predictor of prognosis: the Ohasama study. Hypertension, 52(6):1045-50, 2008

医療リアルワールドデータ

さまざまな形式（関係データ/時系列データ）の
不確実なデータ（欠損や外れ値などを含む）が存在している

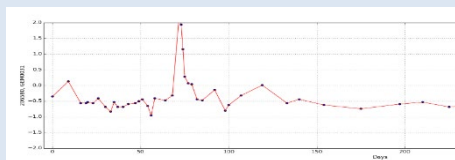
収集データ



- ・実験データ
 - ・研究計画に基づいた収集データ
- 欠損：少
データ量：少

電子カルテなどの 記録データ

- ・記録データ
- 欠損：多
データ量：多



公共データ



- ・薬剤情報
 - ・ゲノム関連情報
- データベース間の揺らぎ
アノテーション：少

これらの不確実なデータや関係を適切に処理する必要がある

AI関連プロジェクトにおける課題の分類

ヘルスケア・未病

創薬

医療

医療用 AI 開発プロジェクト

メンバー

医師・看護師・医療従事者・ケミスト
ケモインフォ・バイオインフォ
データサイエンティスト etc

データ

電子カルテ・公開データベース・
収集データ・実験データ

モデル・アルゴリズム

深層学習・統計モデル・その他機械学習・数理モデル

- 人に関する課題
- データに関する課題
- モデル・アルゴリズムに関する課題

実現場におけるモデル・アルゴリズムの課題

実現場では

- ・ 既存の機械学習モデルがそのまま使えないことも多い
- ・ モデルを一度作って終わりではない

モデルやアルゴリズムのプロトタイピングの必要性

例

「データ収集まだこれからだけど、モデル開発してよ」
「少ないデータでうまくいきそうだったら本番データを送るよ」

モデルやアルゴリズムのメンテナンス性の問題

例

「〇〇も考慮しないとイケないんだよねー」
「分析でうまくいったのでオンライン化してよ」
「あっちのプロジェクトでうまくいったなら、こっちでも使えるでしょ」

現状の課題まとめ

モデリング：実データの不確実さに合わせた柔軟なモデル構築が必要

- ・ 確率的なふるまいを扱いたい
- ・ 通常のライブラリは既存のモデルからの変更が容易でないことが多い

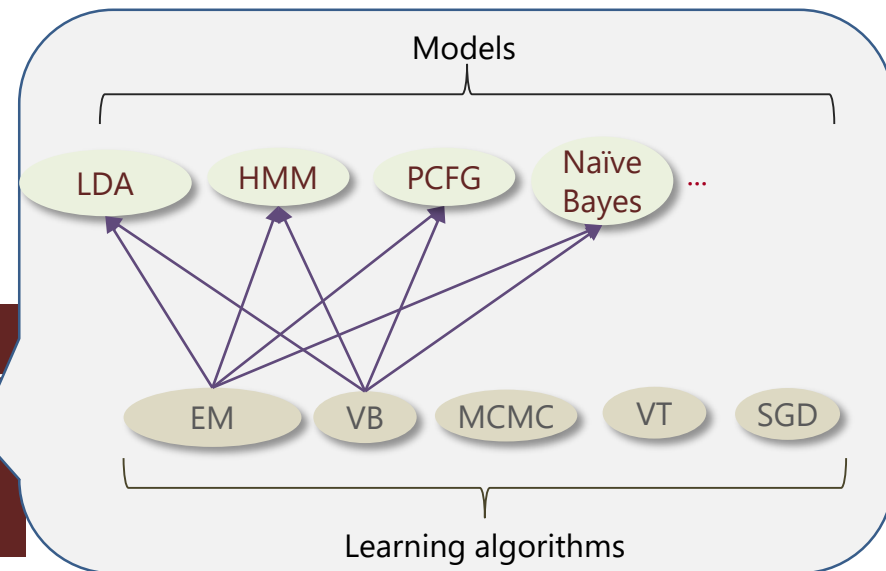
実装コスト：プロトタイピングが必要

モデルや学習法ごとに得手不得手があり、試行錯誤に労力がかかる

再実装コスト：メンテナンス性

少し変更を加えたモデルであっても
学習の再実装が必要

プロジェクトごとにモデルを1から
実装&学習アルゴリズムの変更の
労力が膨大



確率プログラミングによるアプローチ

確率プログラミング言語を用いることで

モデリング

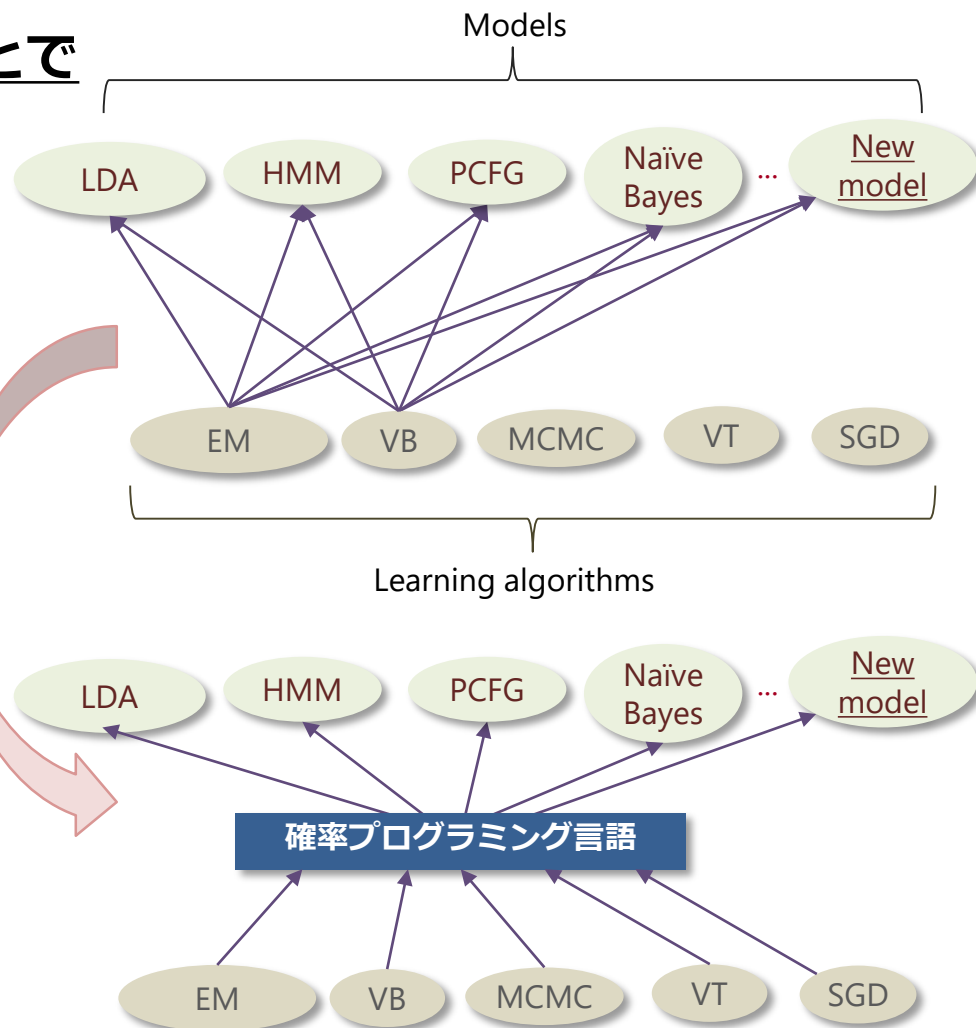
- ・ 確率モデルを使うことで実データのノイズなどに対応可能
- ・ 二つのモデルの組み合わせやモデルの改変が容易

実装コスト

プロトタイピングにおいて様々なモデルや学習アルゴリズムを試すことが可能

再実装コスト

- ・ モデルの改変が容易
- ・ 学習アルゴリズムの切り替えも容易



確率プログラミング言語を用いることでアプリケーションにおいてより重要なモデリングに注力できる

確率プログラミング言語たち

確率処理（確率計算、確率学習、ベイズ推論、サンプリングなど）を持った汎用プログラミング言語

論理型： PRISM[Sato+95], ProbLog[DeRaedt+07], PITA[Riguzzi+11], ICL[Poole+93], ProPPR[Wang+13]

関数型： Anglican[Wang+13], BLOG[Milch+05], Church[Goodman+08]

手続き型： Stan[Bingham+17], PyMC3[Salvatier+16], Dyna[Eisner+04], Factorie[McCallum+08], Infer.NET[Minka+18], Gen [Cusumano-Towner+19]

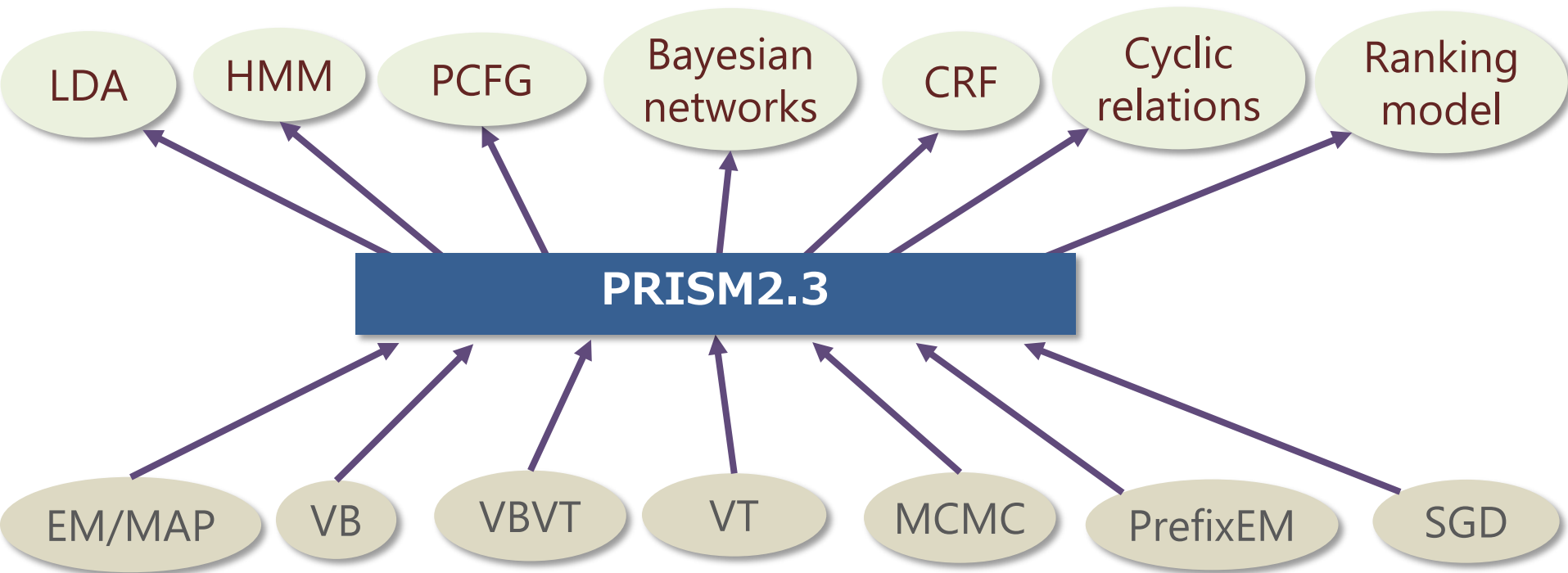
オブジェクト指向 : Figaro[Pfeffer+09], Venture[Mansinghka+14]

深層学習型： Edward[Tran+16], Pyro [Bingham+18], Pixyz [Suzuki+]

多くはほぼ単一の学習アルゴリズムしかサポートしておらず実際にモデルを組んで学習アルゴリズムと相性が悪かった場合の試行錯誤が困難

PRISM2.3

released on August 5, 2017 : <http://rjida.meijo-u.ac.jp/prism/>



記号統計モデリング向けの分布意味論に基づくProlog拡張[Sato+ 95]

- 述語論理でモデルを定義し, 命題論理で確率計算と学習を行う
- 効率的な探索(Tabling [Zhou+ 08])と探索によって得られたデータ構造(説明グラフ)上での学習アルゴリズム

PRISMのしくみ

[論理] **Proof tree**の集合 (の圧縮表現)

↑
↓
分布意味論 + 実用上は問題にならない程度の仮定

[確率] 確率の積和式の集合
(積和計算のグラフ)

探索により説明グラフを自動生成(Prolog)

PRISMコード

```
bloodtype(P) :-
```

```
  g(X,Y),
```

```
  ( X=Y -> P=X
```

```
  ; X=0 -> P=Y
```

```
  ; Y=0 -> P=X
```

```
  ; P=ab).
```

```
g(X,Y) :-
```

```
  msw(father_gene,X),
```

```
  msw(mother_gene,Y).
```

PRISMでは全ての学習アルゴリズムが
説明グラフを介して実装されている(C/C++)

確率計算

→説明グラフ上のDP

学習 (outside確率計算・勾配計算)

→説明グラフの上の形式微分

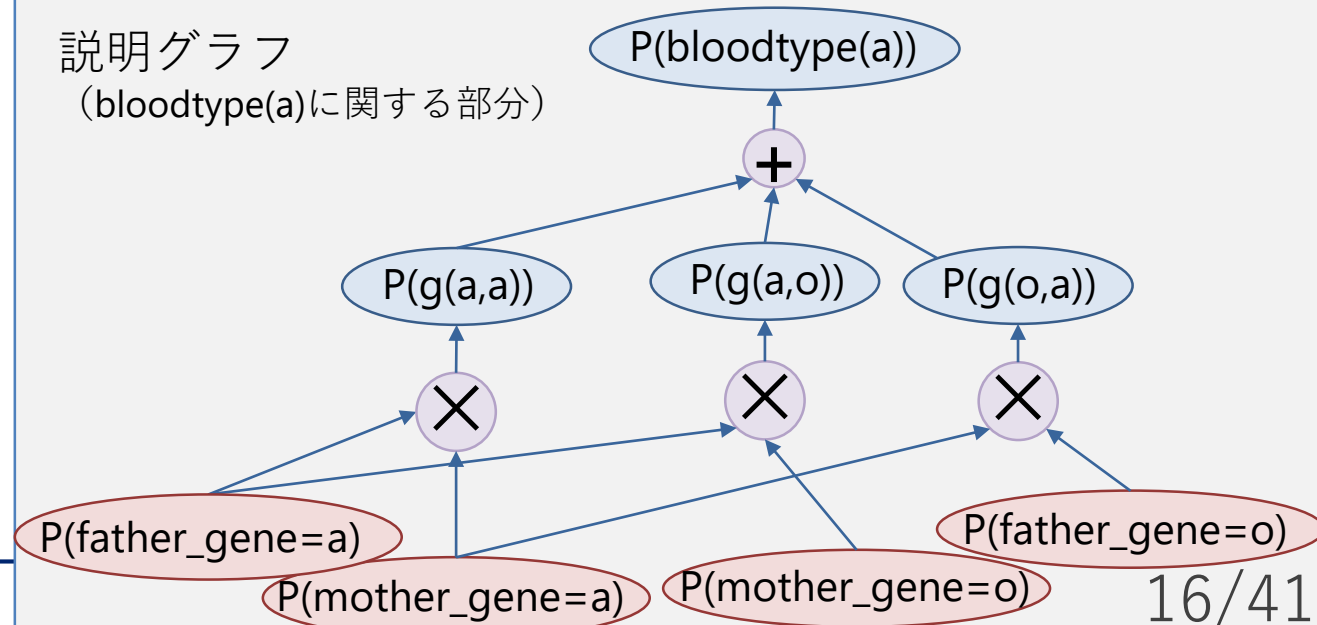
minibatch法

→部分説明グラフ

Prefix確率計算

→循環説明グラフ
(線形・非線形)

説明グラフ
(bloodtype(a)に関する部分)



PRISMの概要

PRISMコード

```
bloodtype(P) :-  
    g(X,Y),  
    ( X=Y -> P=X  
    ; X=0 -> P=Y  
    ; Y=0 -> P=X  
    ; P=ab).  
g(X,Y) :-  
    msw(father_gene,X),  
    msw(mother_gene,Y).
```

[論理] 一階述語論理での記述

[確率] 無限個の対象を扱う確率モデル (HMM, PCFG, etc) も有限の確率モデル (LDA, BN, etc) も一貫して記述できる

B-Prolog の tabled search エンジン

中間データ構造

説明グラフ

[論理] 命題論理レベルでの記述

[確率] 有限個の確率の積和方程式 (巨大な線形or多項式方程式)

計算エンジン

- 説明グラフ上での確率の積和方程式を解く

学習エンジン

- 説明グラフ上の方程式の下でパラメータに関して目的関数 (尤度・変分下限 etc) の最大化 (最小化) を行う

Distribution (PRISM) semanticsの概要

mswを使って確率方程式 = 同時確率分布を表現

$$\text{msw}(x, v) \leftarrow P(X = v)$$

基底アトム 確率値

論理積に対応した計算

$$\text{msw}(x, v) \wedge \text{msw}(y, u)$$



$$P(X = v) \cdot P(Y = u)$$

論理和に対応した計算

$$\text{msw}(x, v) \vee \text{msw}(y, u)$$



(注) PRISMでは
排他性を仮定

$$P(X = v) + P(Y = v)$$

PRISMにおける確率モデルの例

LDA

(Latent Dirichlet Allocation)

「文書にトピックがあり、トピックに依存して単語の分布が決まる」

* 事前分布などはデフォルトで用意されるので書く必要はない

```
values(word(_), [1-100]).  
values(topic, [1-10]).  
doc([W|R]):-  
    msw(topic,Z),  
    msw(word(Z),W),  
    doc(R).
```

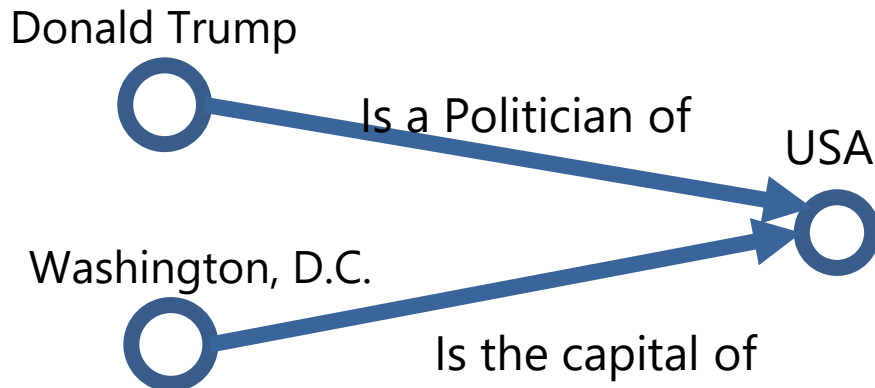
Markov chain

「一つ前の状態に依存して、次の状態が確率的に決まる」

```
values(tr(_), [1-10]).  
  
mc(_, []).  
mc(S, [T|R]):-  
    msw(tr(S),T),  
    mc(T,R).
```

例 2 : 知識グラフと埋め込み学習

知識グラフ：主語S、目的語O、関係Rの三つ組みをグラフで表現したもの



- 知識グラフは正例のみ含む
- 疑似負例を生成[Socher+,13]
- PRISM2.3 の rank learning を用いて学習
 - rank learning では正例(s,o,true) と負例(s',o',false)から成る順位ペア ((s,o,true) > (s',o',false))について $P(\text{rel}(s, o, \text{true})) > P(\text{rel}(s', o', \text{false}))$ となるようにパラメータを学習
 - 正例・負例を正しく判別できるかを確認
 - 対数尤度の差のヒンジロス: $\text{Loss}(g_+, g_-) = \max(0, l(g_-) - l(g_+) + c)$ 、学習法はSGD
- モデル：DistMultをベースにしたPRISM-DistMult
- データ：FB15Kの一部
(/award/award_nominee/award_nominations-
/award/award_nomination/award_nominee)

双線形モデル

RESCAL [Nickel,13],..., **DistMult** [Yang+,15], [Trouillon+,16]

DistMult : $\text{rel}(s, o, \text{true})$ となるスコア : $e_s^T W e_o$ (W :diagonal)

結果

PRISM-DistMult(実体ベクトル= 確率分布) :

PRISMコード

```
values(e(_), [1-50]).  
values(w(_,_), [true, false]).  
rel(S,O,R):- msw(e(S), I), msw(w(I, I), R), msw(e(O), I).
```



	正答率
PRISM-DistMult	89.1%
ProPPR-DistMult	67.4%

例 2：最尤推定 vs 順位学習

パラメータ学習には順位学習がよいのか？

- 最尤推定には正例があれば良い
- 順位学習では正例(**s, rel, o**) と疑似負例から成る順位ペア

	rel0	rel1	rel2	rel3	rel4	rel5	rel6	rel7	rel8	rel9
Rank Learning(%)	87.4	79.8	74.1	72.7	62.0	62.7	54.8	55.0	69.4	69.5
MLE (%)	85.9	60.4	68.7	72.6	61.8	56.1	52.0	50.0	66.5	69.7
T-test	=	>	=	=	=	>	=	>	=	=

この場合には順位学習が最尤推定より悪くないパラメータを推定する

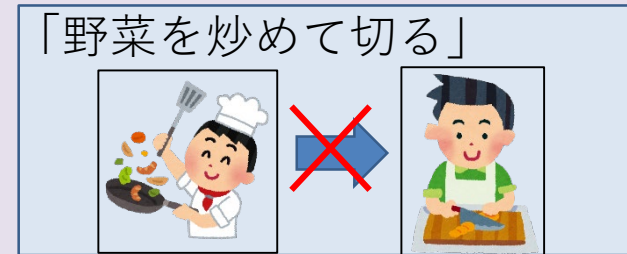
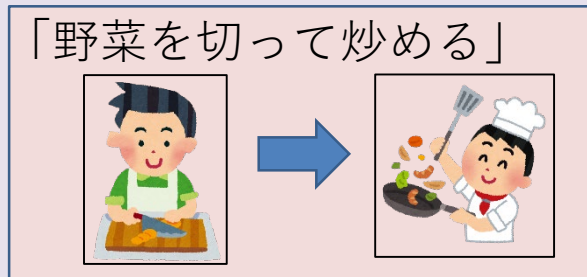
PRISM まとめ

- 現実の複雑な問題に対して効率よくプロジェクトを進めるためにはモデリング言語が必須である
 - 確率プログラミング言語を使ってモデリングと学習・計算を分離することにより確率モデルの開発・維持の手間を減らす
- **PRISM2.3**は論理ベースの確率プログラミング言語として多くの学習機能を提供している
 - これらの学習機能を切り替えて容易に比較できる
- モデリング例を幾つか紹介した

PRISMによる確率プログラミングと 深層学習を組み合わせる

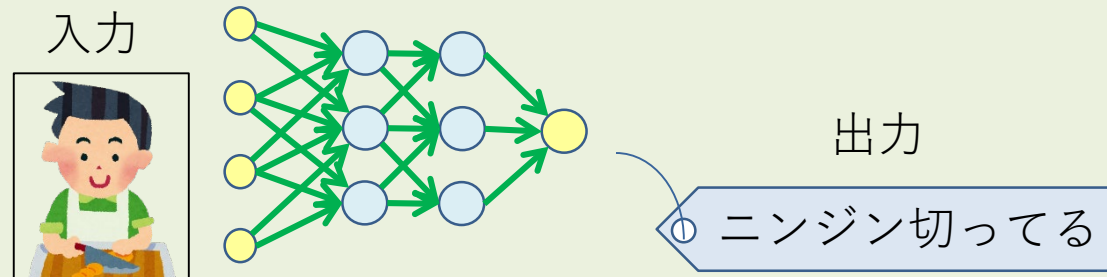
• PRISMによる確率プログラミング

- 明確にデータの生成過程（モデル）をプログラミング言語を用いて柔軟に設計できる．規則・法則があるものに適している．



• 深層学習

- 人間が明確なモデルが与えられないような状況で学習するような問題に適している．入力・出力のペアからモデルを学習するような状況．



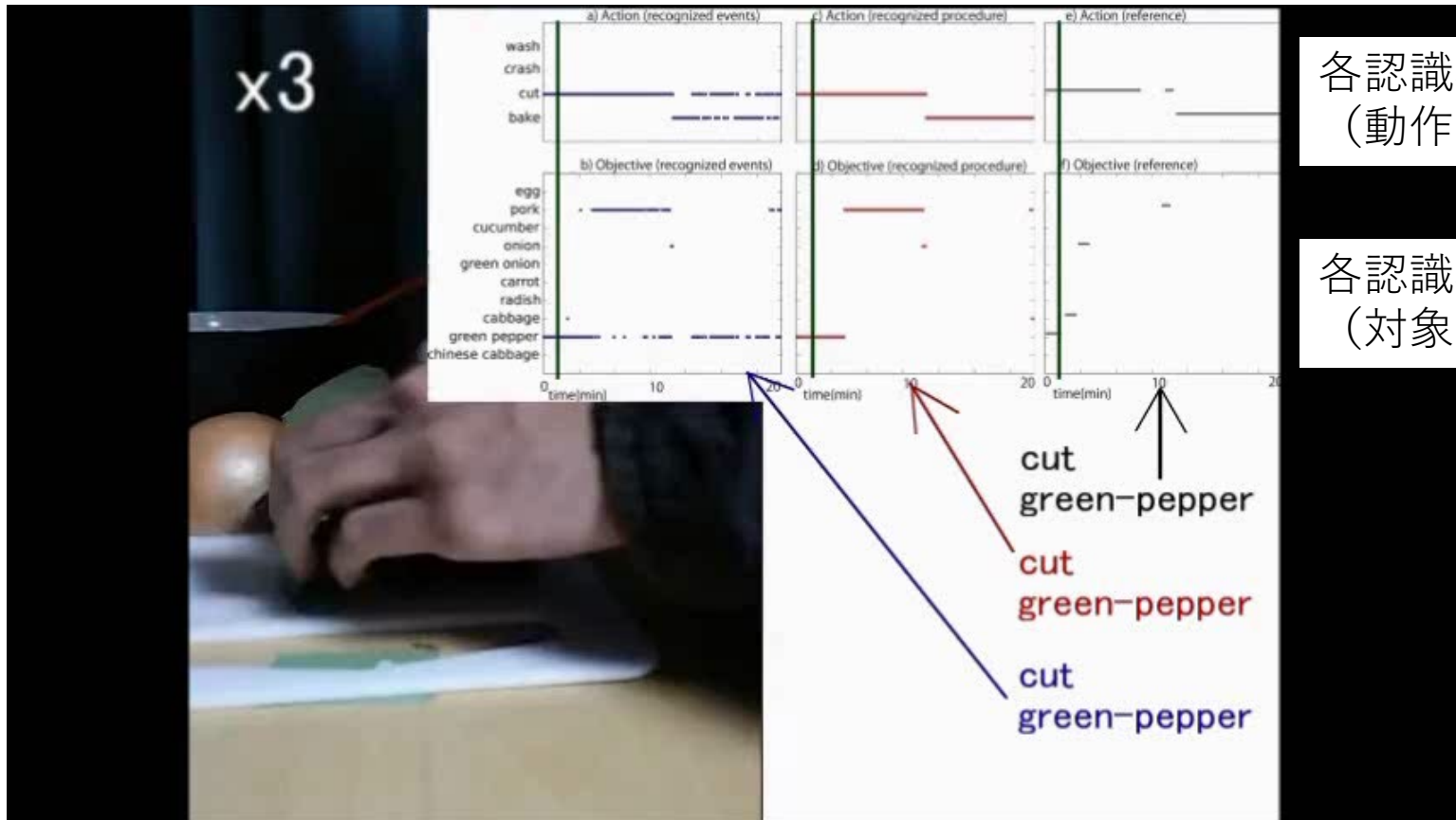
得意分野が異なる → 組み合わせることでお互いの欠点を補える可能性がある

PRISMによる確率プログラミングと 深層学習を組み合わせた例 (1/2)

深層学習 (画像 + 音)
+PRISM (レシピ)

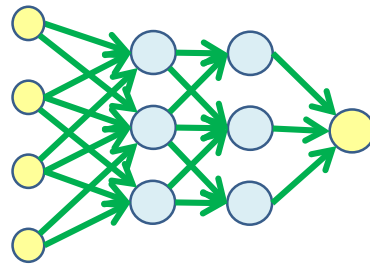
深層学習
(画像 + 音)

答え



PRISMによる確率プログラミングと 深層学習を組み合わせた例 (2/2)

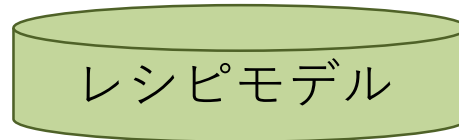
深層学習ツール (Theano)



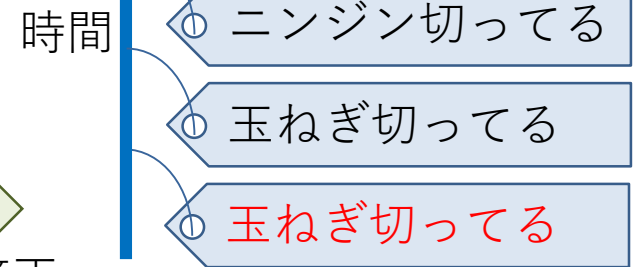
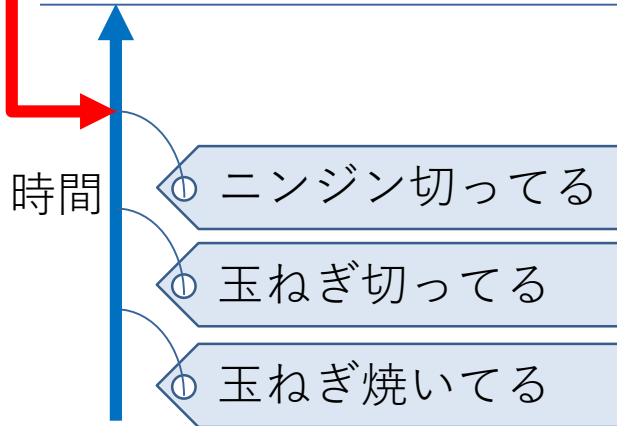
出力

① ニンジン切ってる

確率プログラミング (PRISM)



認識ミスをレシピから修正



問題点：

深層学習と確率プログラミングのシステムが分離している

- システム全体の最適化が難しい
- デバッグ・チューニングが困難

統合したツールが欲しい！！

Tensorized-PRISM(T-PRISM)

モデル

*Deep
neural
network*

VAE

LDA

Bayesian
networks

HMM

Cyclic
relations

CRF

Ranking
model

PCFG

モデルを書くだけで
様々な学習・推論が
様々な実行環境で
そのまま実行可能

深層確率論理プログラミング言語: Tensorized-PRISM

学習・推論

VI

EM

VBVT

VB

VT

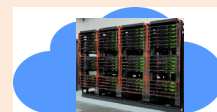
MCMC

PrefixEM

SGD

実行環境

クラウド: Docker
クラスタ: MPI



汎用エッジデバイス

・ Raspberry-pi, Jetson

PC

・ Linux, Mac, Windows

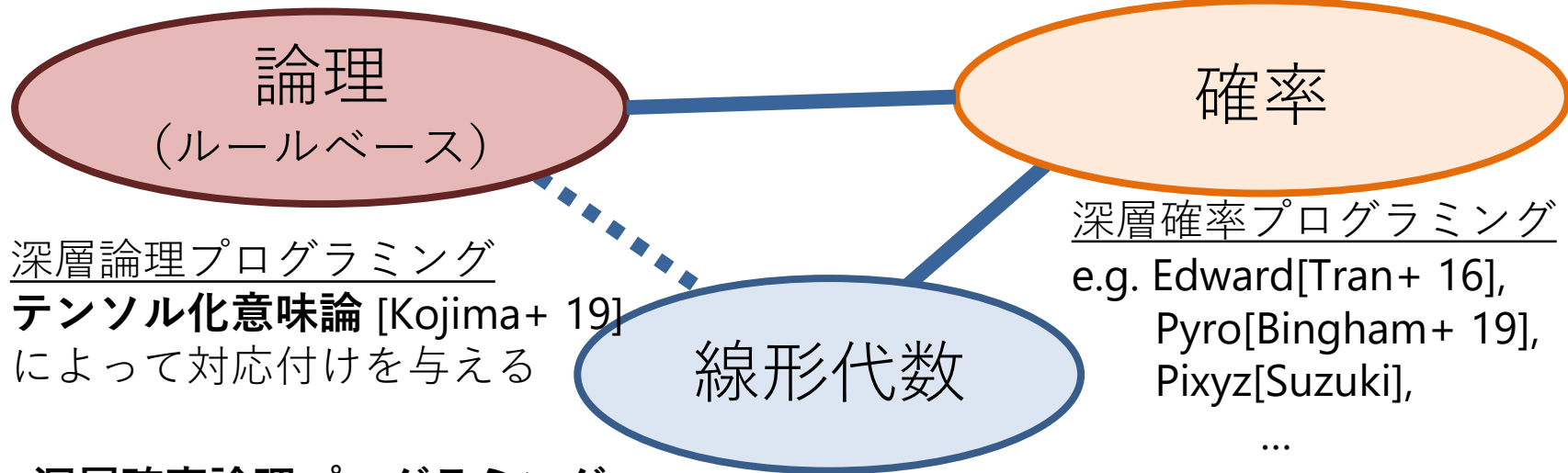
Tensorized-PRISMのコンセプト

深層学習から確率モデリングまでの幅広いモデリングが可能で、
(論理) シンボルと結び付けた解釈可能なモデリング言語

確率論理プログラミング

e.g. PRISM [Sato+ 97], ProbLog[De Raedt+ 07],...

分布意味論[Sato+ 95]によって対応付けされている



深層論理プログラミング

テンソル化意味論 [Kojima+ 19]

によって対応付けを与える

深層確率プログラミング

e.g. Edward[Tran+ 16],

Pyro[Bingham+ 19],

Pixyz[Suzuki],

...

深層確率論理プログラミング



- ・ルールベースによる説明可能性
- ・線形代数のスケーラビリティ
- ・確率による不確実さの取り扱い

関連研究・関連システム

	ニューラルネットワークを設計可能	論理推論が可能	確率モデリングが可能	論理式の構成要素が分散表現をもつ	
NTP [Rocktaschel 17]				○	Datalog
dILP [Evans 18]				○	Datalog
LTN [Serafini 16]	○			○	一階述語論理 (subset)
TensorLog [Cohen 17]		○			Datalog
DeepProbLog [Manhaeve 18]	○	○	○		Prolog
T-PRISM [Kojima 19]	○	○	○	○	Prolog

T-PRISMで書くのに適したモデル

- ルール＋確率のモデル

PRISM

T-PRISM

- 従来PRISMモデルで高次元や大量データを扱いたい場合

- ルール＋ニューラルネットが混じるモデル

世の中の多くのシステムでは両方を扱う必要がある

- 例1：ルール＋画像処理（自動運転＝交通ルール＋物体認識）
- 例2：制約＋信号処理（センサの予測＝物理制約＋予測）

T-PRISM

- （深層確率）時系列モデル（無限の対象を含む背景知識）

- 確率モデル＋ニューラルネットワーク

HMM-DNN, PCFG-DNN, etc.

- （制約付きの）潜在空間モデルやRNN

T-PRISM

特にルールや時系列の絡むものに強い

T-PRISMの概要

T-PRISMコード

```
:-set_index_range(i,20).  
rel(S,R,O):-  
    tensor(v(S),[i]),  
    tensor(v(O),[i]),  
    tensor(r(R),[i]).
```

[論理] 一階述語論理での記述

[tensor] 無限個の対象を扱うモデルも有限のモデルも一貫して記述できる

B-Prolog の tabled search エンジン

説明グラフ : atomに対してtensorを割り当てる

[論理] 命題論理レベルでの記述

[tensor] 有限個の**tensor**の積和方程式
(巨大な線形/非線形方程式)

PRISM: [確率]
有限個の確率の積和方程式

計算エンジン

- 説明グラフ上での確率の積和方程式を解く

学習エンジン

- 説明グラフ上の方程式の下でパラメータに関して目的関数（尤度・変分下限 etc）の最大化（最小化）を行う

Distribution (PRISM) semantics vs Tensorized (T-PRISM) semantics

PRISM:
Distribution semantics

mswを使って確率方程式
= 同時分布を表現

$\text{msw}(x, v)$



$P(X = v)$

確率値

T-PRISM:
Tensorized semantics

tensorを使ってテンソル
方程式を表現

$\text{tensor}(x, i)$



x_i



インデックスシンボル : i

ベクトル : x

Distribution (PRISM) semantics vs Tensorized (T-PRISM) semantics

論理和に対応した計算

PRISM

$$\text{msw}(x, v) \vee \text{msw}(y, u)$$



$$P(X = v) + P(Y = v)$$

確率値

T-PRISM

$$\text{tensor}(x, i) \vee \text{tensor}(y, i)$$



x_i



y_i

ベクトル： $\mathbf{x} + \mathbf{y}$

インデックスシンボル： i

∨の両側のインデックスシンボルが
等しいことを仮定

Distribution (PRISM) semantics vs Tensorized (T-PRISM) semantics

論理積に対応した計算

PRISM

$$\text{msw}(x, v) \wedge \text{msw}(y, u)$$



$$P(X = v) \cdot P(Y = u)$$

確率値

T-PRISM

$$\text{tensor}(x, i) \wedge \text{tensor}(y, i)$$



x_i



y_i

スカラー： $\mathbf{x}^T \cdot \mathbf{y}$

インデックスシンボル：empty

$$\text{tensor}(x, i) \wedge \text{tensor}(y, j)$$



x_i



y_j

行列： $\mathbf{x} \cdot \mathbf{y}^T$

インデックスシンボル： i, j

説明グラフ with Einstein's notation

T-PRISMにおける 論理積 = アインシュタインの縮約記法

同じ項で添字が二つ以上ある場合、
その添字について和を取る

例： $\text{tensor}(x, [i, j, k]) \wedge \text{tensor}(y, [j, k]) \wedge \text{tensor}(z, [j, m])$

$$x_{ijk} y_{jk} z_{jm}$$

行列： $\sum_{jk} x_{ijk} y_{jk} z_{jm}$

インデックスシンボル： i, m

サイズはあらかじめ与える
 $\text{tensor_atom}(x, [10, 10, 10]).$
 $\text{tensor_atom}(y, [10, 10]).$
 $\text{tensor_atom}(z, [10, 10]).$

* アインシュタインの縮約記法はANDの交換則とも相性が良い

非線形対応とニューラルネットワーク

Operator atom を導入し、
論理積を関数適用として解釈する

`operator(sigmoid) $\wedge$
tensor(x, [i, j, k])`

$\uparrow$
 $\sigma(x_{ijk})$

* 一般にANDの交換則
は満たさない

`output(y, x) $\Leftrightarrow$ tensor(w0, [i, j]) $\wedge$ layer1(y, x).
layer1(y, x) $\Leftrightarrow$ operator(sigmoid)
 $\wedge$ tensor(w1, [j, k]) $\wedge$ layer2(y, x).
layer2(y, x) $\Leftrightarrow$ tensor(input(x), [k]).`

構築されるテンソル方程式

$$Output = W_0 \cdot L_1$$

$$L_1 = \sigma(W_1 \cdot L_2)$$

$$L_2 = input$$

```
tensor_atom(w0, [10, 256]).  
tensor_atom(w1, [256, 784]).  
tensor_atom(input, [784]).
```

```
output(Y, X) :-  
    tensor(w0, [i, j]),  
    layer1(Y, X).
```

```
layer1(Y, X) :-  
    operator(sigmoid),  
    tensor(w1, [j, k]),  
    layer2(Y, X).
```

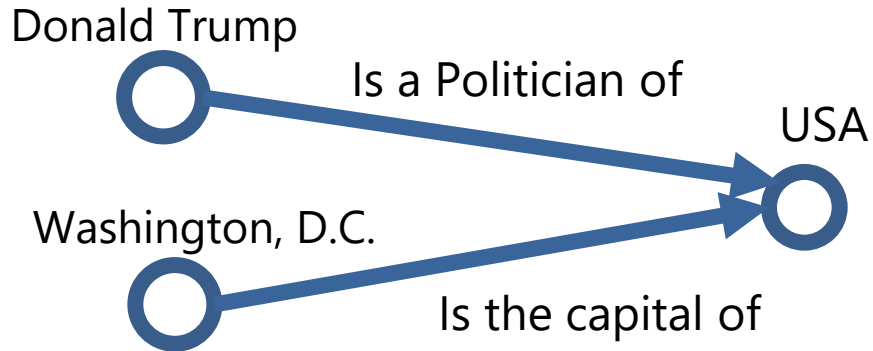
```
layer2(Y, X) :-  
    tensor(input(X), [k]).
```

T-PRISMを用いたプログラム例

- 知識グラフ応用
- **Transitive closure** の計算
- **PRISM**の再現

例 1 : 知識グラフと埋め込みの学習

知識グラフ : 主語S、目的語O、関係Rの三つ組みをグラフで表現したもの



問題 : S,Rが与えられたときのOを予測する

手法 : DistMultスコア : $e_S^T W_R e_O$ (W_R : diagonal)

T-PRISM-DistMultコード

```
tensor_atom(v(_), [20]).  
tensor_atom(r(_), [20]).  
rel(S,R,O):-  
    tensor(v(S), [i]),  
    tensor(v(O), [i]),  
    tensor(r(R), [i]).
```

従来のPRISM

データ : FB15Kの一部で学習

	正答率
PRISM-DistMult	89.1%
ProPPR-DistMult	67.4%

“Learning to rank in PRISM”. R. Kojima, T. Sato: In *International Journal of Approximate Reasoning*, Vol. 93 pp. 561 – 577, 2018.

データ : フルデータセットで実験

4行程度のT-PRISMプログラムで従来法を再現

	WN18	FB15K
MRR	60.6%	54.1%
H10	86.0%	75.8%

Method	Filtered					
	WN18			FB15k		
	MR	H10	MRR	MR	H10	MRR
DistMult (orig) (Yang et al., 2015)	-	94.2	0.83	-	57.7	0.35
DistMult (Toutanova and Chen, 2015)	-	-	-	-	79.7	0.555
DistMult (Trouillon et al., 2017)	-	93.6	0.822	-	82.4	0.654

例 2 : Transitive closure

$r1(X,Y)$ を与えられたときの
推移閉包を計算する問題

$r2(X,Y) \leftarrow r1(X,Y)$

$r2(X,Z) \leftarrow r1(X,Y) \wedge r2(Y,Z)$

```
tensor_atom(rel1,[100000, 100000]).
```

```
rel2'(U,V):-tensor(rel1,[U,V]).
```

```
rel2'(U,V):-tensor(rel1,[U,W]),rel2(W,V).
```

```
rel2(U,V):-operator(min1),rel2'(U,V).
```

再帰プログラムかつ、Cyclicな説明グラフ
を含むプログラム

従来

ノード数N=10,000のランダム
グラフで実験

Sato, Taisuke, Katsumi Inoue, and Chiaki Sakama.
"Abducting Relations in Continuous Spaces." IJCAI. 2018.

	N	#iterations	time(sec)
	100	3	0.06
	1,000	13	0.34
	10,000	5	23.1
	100,000	3	8097.62

例 3 : PRISM in T-PRISM

PRISMのmswをtensorを使って再現可能

- 確率分布の制約をsoftmaxで表現
- 確率選択をonehot encodingで表現

つまり、HMMやPCFGなどのモデルも
記述可能

```
simulated_msw(Sw, val) :-  
    get_values(Sw, values),  
    nth0(Index, values, val),  
    prob_tensor(Sw, [i]),  
    tensor(onehot(Index), [i]).
```

	Neural network	Simple model + large-scale data (e.g. DistMult)	Recursive/Cyclic program (e.g. transitive closure, Markov Chain)	Infinite model (e.g. PRISM model, PCFG)
PRISM			○	○
T-PRISM	○	○	○	○

オープンソースとして公開中

PRISM/T-PRISM

<https://github.com/prismplp/prism>

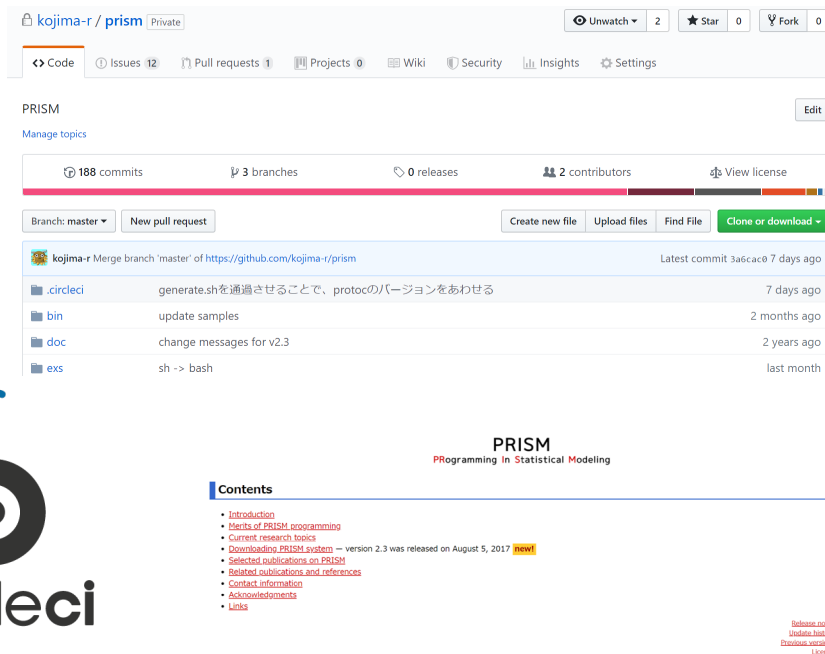
Githubで公開・開発

- Docker対応
- 自動ビルド/自動テスト
- 各種サンプルプログラム：

T-PRISM関連の例では、ニューラルネットワーク、知識グラフのモデル、Markov chainの確率計算、推移閉包の高速計算など

今後さらに強化

- アプリケーション用の実装の追加



PRISM webサイト

<https://rjida.meijo-u.ac.jp/prism/>

開発研究課題

アプリケーション・手法

- ベクトル→論理
 - 一階述語論理やアプリケーションに適した埋め込み方法
(tensorized semanticsの改良)
- 論理←ベクトル
 - 論理式で与えられる問題をどれだけ解けるか (SAT, Boolean関数の合成, etc...)

アルゴリズム

Prolog探索以外の説明グラフ構築法

- 全解探索でないサンプリングによる証明エンジン
- 一階述語論理以外からの説明グラフ構築

効率的な計算方法

説明グラフ→計算グラフコンパイラ

(説明グラフ中の同一計算を見つけてミニバッチ化することで高速化可能)

実行環境ごとの最適化

- RISC-V等への対応

アプリケーション

T-PRISMコード

B-Prolog エンジン

説明グラフ

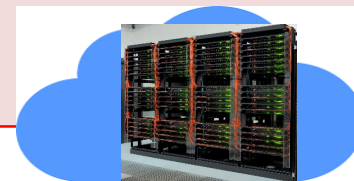
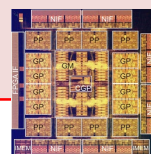
計算エンジン・学習エンジン:

PRISM部分: C/C++

T-PRISM部分:

Python/TensorFlow

実行環境



さいごに

深層確率プログラミングには、
より洗練されたアルゴリズムが必要

- グラフ
- データ構造・探索
- ベクトル埋め込み

<https://github.com/prismp1p/prism>

Pull Request Welcome