

変分ベイズ名付けゲームに基づく 多エージェントによる記号創発の評価

福岡慶太¹ 長野匡隼¹ 中村友昭¹ 谷口彰² 谷口忠大^{3,2}

¹ 電気通信大学 ² 立命館大学 ³ 京都大学

k_fukuoka@radish.ee.uec.ac.jp

概要

記号創発現象をモデル化するために、複数のエージェントが同じ対象を観測し、それを表現する共有されたサインを推論する、集合的予測符号化仮説が提唱されている。その推論方法として、我々は変分ベイズ法に基づく手法を提案し、複数のエージェント間で共有されたサインが形成可能であることを示している。変分ベイズ法は一般的にサンプリングを用いた手法に比べて学習効率が高いことが知られている。本稿では、1 エージェントが得る観測データのみでは正確な名付けが困難な設定や、エージェント数・データ数を増加させた設定を用いて、サンプリングを用いた従来手法との比較実験を行った。実験の結果、従来手法に比べて、より難しい課題に対しても学習が可能であり、大規模な設定でも収束性が高いことが確認された。

1 はじめに

人間は他者や環境との相互作用を通じて、集団内で意味を共有し、意図を伝達できる記号を創り出すことが可能である。この過程を創発コミュニケーションと呼び、そのメカニズムを構成論的に解明する研究が進められている。

谷口は、言語が集合的な予測符号化によって形成されるという集合的予測符号化仮説を提唱している [1]。集合的予測符号化では、図 1 のように複数の独立したエージェントが同一の対象を観測し、それを表現する共有サイン z を推論する問題として記号創発現象をモデル化している。共有されたサインの具体的な推論法として、言語ゲームの一種である名付けゲームを活用した Metropolis Hastings Naming Game (MHNG) [2, 3] および変分ベイズ名付けゲーム (VBNG) [4] が提案されている。

MHNG は MH 法に基づく推論法であり、2 エー

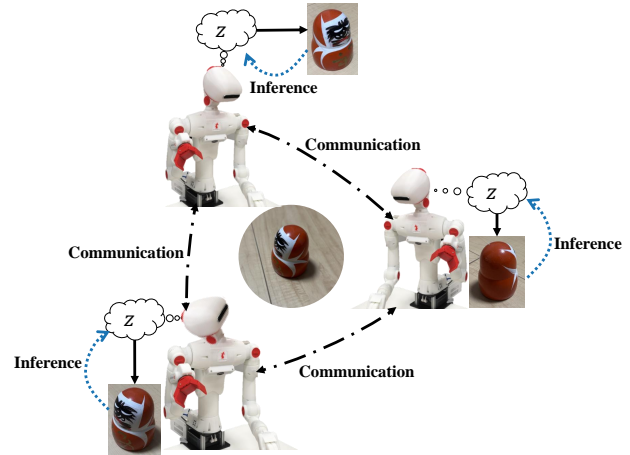


図 1 提案手法の概要: コミュニケーションすることで、他者の観測情報を間接的に使用し共有される記号の創発をしている。

ジェント間で観測対象を表現するサインの提案と受理・棄却を繰り返すことで推論する。MHNG による 3 体以上のエージェント間での記号の創発では、2 エージェント間のコミュニケーションを再帰的に繰り返す必要があった [5]。一方で我々は、変分ベイズ法に基づく推論法である VBNG により、シンプルな形式で 3 エージェント以上への拡張が可能であることを示している [4]。また、一般に MH 法に比べて変分ベイズ法の方が学習効率が良いことが知られている。複数のエージェントでのコミュニケーションが重要になる設定や、エージェント数やデータ数が多い設定においては顕著な違いが生じると考えられる。そこで本稿では、VBNG と MHNG の学習性能・効率性の比較を行う。

関連研究として、深層強化学習を用いた創発コミュニケーションの研究が行われている [6, 7]。これらの研究では、送り手から受け手への一方向のコミュニケーションを用いて、適切な対象を選択する Referential Game タスクを強化学習により学習している。一方、本稿では、双方向コミュニケーションを利用した教師なし表現学習によって、エージェ

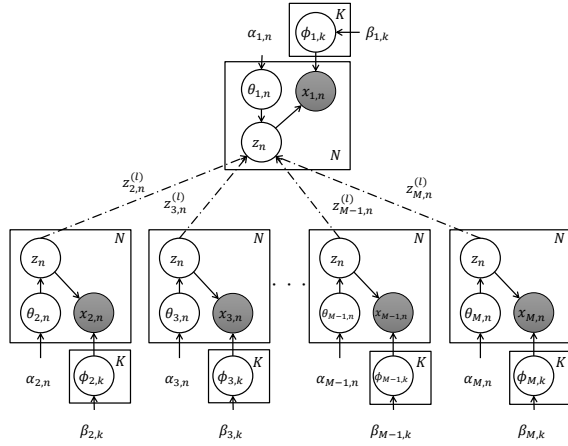


図2 VBNGのグラフィカルモデル（白色ノードは未観測、灰色ノードは可観測を表す。一点鎖線矢印はコミュニケーションによる推論を表す）

表1 各変数の定義

変数	定義
$z_{m,n}$	観測を表現する共有されたサイン
$x_{m,n}$	各エージェントの n 番目の観測
$\theta_{m,n}, \phi_{m,k}$	多項分布のパラメータ
$\alpha_{m,n}, \beta_{m,k}$	ディリクレ事前分布のパラメータ

ント間で共有される記号を学習することが目的である。また、コミュニケーションチャンネルを用いた深層強化学習も、エージェント間でコミュニケーションに使用する共有された記号を学習していると見ることができる [8, 9]。これらの研究では複数のエージェントをネットワークで接続し、他エージェントの内部状態から計算された誤差情報を逆伝播することで学習している。しかし、他者の内部状態を直接利用することは、独立した個体間のコミュニケーションでは不可能であると考えられる。さらに、これらの研究の目的は記号の創発現象のモデル化ではなく、特定のタスクを高精度に達成することであり、本稿の目的とは異なる。

2 変分ベイズ名付けゲーム

本稿では、図2のグラフィカルモデルでの分散的ベイズ推論を考える。各変数の定義は表1の通りである。 m はエージェントを区別するための添字であり、 k はサインのインデックスである。変分ベイズ名付けゲーム (VBNG) では、各エージェントが同一の対象を異なる視点や異なるセンサーで観察することで得られる観測 $x_{m,n}$ ($m \in \{1, 2, 3, \dots, M\}, n = 1, 2, \dots, N$) のみから、対象を表現する共有されるサイン z を推論する。

更新式は次式となる。ただし、添字 d は D 次元ベクトルの d 番目の要素であることを表している。

$$q(z_n) = \text{Cat}(z_n | \eta_n) \quad (1)$$

$$\eta_n = \eta_{A,n} \ln \theta_{n,z} q(\theta_{n,z}) \prod_{m \in \{1, 2, \dots, M\}} \eta_{m,n} \quad (2)$$

$$\eta_{m,n} = \exp \left(\sum_d x_{m,n,d} \ln \phi_{m,z_n} q(\phi_{m,z}) \right) \quad (3)$$

$$q(\theta_n) = \text{Dir}(\theta_n | \alpha'_n) \quad (4)$$

$$\alpha'_{n,z} = \eta_n + \alpha_z \quad (5)$$

$$q(\phi_{m,z}) = \text{Dir}(\phi_{m,z} | \beta'_m) \quad (6)$$

$$\beta'_{m,d} = \sum_n \eta_n x_{m,n,d} + \beta_d \quad (7)$$

式(2)において、各エージェントは直接観測することができない他エージェントの内部パラメータを利用しており、このままではパラメータを更新することができない。そこで、エージェント1を聞き手、その他のエージェントを話し手とした名付けゲームを考える。話し手のエージェント m ($m \in \{2, 3, \dots, M\}$) は $\eta_{m,n}$ をパラメータとしたカテゴリ分布からサンプリングした L 個の離散値 $z_{m,n}^{(l)}$ ($l = 1, \dots, L$) をサインとしてエージェント1に送信する。

$$z_{m,n}^{(l)} \sim \text{Cat}(z_{m,n} | \eta_{m,n}) \quad (8)$$

エージェント1は、話し手 m から受け取った L 個のサインから、その予測分布 $p(z_{m,n} | z_{m,n}^{(1)}, \dots, z_{m,n}^{(L)})$ を計算し、エージェント m の内部パラメータ $\eta_{m,n}$ の近似値 $\eta'_{m,n}$ を計算する。

$$p(z_{m,n} | z_{m,n}^{(1)}, \dots, z_{m,n}^{(L)}) = \text{Cat}(z_{m,n} | \eta'_{m,n}) \quad (9)$$

$$\eta'_{m,n,z} \propto \sum_l \delta(z, z_{m,n}^{(l)}) + \gamma \quad (10)$$

ただし、 γ はディリクレ事前分布のパラメータである。エージェント1は、他の話し手のサインについても同様に近似値を計算して、式(2)の代わりに次式のように η_n を計算する。

$$\eta_n = \eta_{1,n} \ln \theta_{n,z} q(\theta_{n,z}) \prod_{m \in \{2, 3, \dots, M\}} \eta'_{m,n} \quad (11)$$

この η_n を用いて、式(1)~式(7)を計算することができる。このようなサイン $z_{m,n}^{(l)}$ をやり取りするコミュニケーションを、聞き手と話し手を変えながら繰り返すことで、共有された記号を推論することができる。

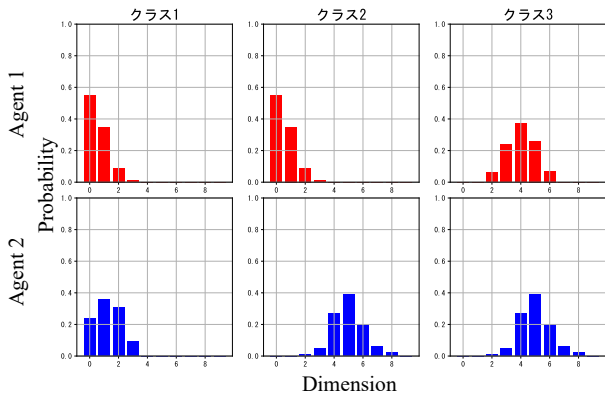


図3 人工データを生成した分布の例

3 実験

複数のエージェントが同一の物体を観測し、物体のクラス毎に共有した名前を推論することを想定した名付けタスクを行い、提案するVBNGと従来手法の性能を比較した。

3.1 実験設定

推論の設定として、パラメータの更新を50ステップ行い、初期値を変更して100回試行した。

3.1.1 データセット

各エージェントの観測として、人工的に生成した50次元のベクトルを用いた。人工データは、10クラスの観測を想定し、クラス毎にピーク位置をずらした多項分布からサンプリングして生成した。図3は、2エージェントが観測するクラス1, 2, 3の人工データを生成する多項分布の一部を示している。

エージェント1では、クラス1とクラス2の分布のピーク位置が、エージェント2では、クラス2とクラス3の分布のピーク位置が一致しており、それぞれ類似した観測ベクトルが生成される。このように、各エージェントで一部のクラスが識別困難となるよう設定した。その結果、1エージェントが得る観測データのみでは正確な名付けが困難になる。

また、各エージェントの観測ベクトルのピーク位置が一致しているクラスの数オーバーラップ数と定義した。例えば、図3のデータではそれぞれのエージェントで、観測ベクトルのピーク位置が一致しているクラスが2つとなるので、オーバーラップ数は2となる。オーバーラップ数を増加させることで、より名付けが困難なデータを生成することができる。オーバーラップ数を変えた分布から各クラスの

データを複数サンプリングすることで、複数のデータセットを生成した。オーバーラップ数ごとのピーク位置の詳細は、付録に記載する。

3.1.2 比較手法

提案モデルであるVBNGをトップラインモデル(VB)と従来手法と比較した。トップラインモデルには、式(2)を用い、全エージェントの内部状態が互いに観測可能な推論手法(VB)を用いた。従来手法としては、One-sample(OS)近似を用いたRMHNGを使用した。従来手法のアルゴリズム詳細については、[5]を参照されたい。

3.1.3 評価指標

正解と推論された記号の一致度を評価するため、Adjusted Rand Index (ARI)を用いた。ARIは推論された記号が正解に近いほど値が大きくなり、最大値は1である。

また、学習の収束を確認するため、各手法の目的関数の値を計算した。VBNGとRMHNGはモデルが異なるため、それぞれの目的関数も異なる。VBNGではELBOを、RMHNGでは対数尤度を使用した。各学習ステップにおける100回の試行の平均値を計算した。また、収束の効率性を評価するためELBOおよび対数尤度の計算と同時に、各学習ステップの終了時間を記録した。

3.1.4 計算機仕様

実験は、MacStudio(2023)：Apple M2 MAXチップ(CPU 12コア, GPU 30コア), RAM 32GBで行った。

3.2 実験結果

3.2.1 創発した記号の定量評価

表2は、オーバーラップ数を変化させた場合の、最後の学習ステップでのARIの平均値±標準偏差を示している。オーバーラップ数が増加するにつれて、すべての手法でARIが減少する傾向が見られた。特に、オーバーラップ数が7以上になるとRMHNGではARIの値が0となり、観測を表現する記号を推論することができていない。一方、VBNGではトップラインモデルに比べて、平均値の差が0.10と生じているものの、ARIは0.6以上となった。すなわち、オーバーラップ数が多い、より難しいデータセットにおいても観測を表現するサインを推

表 2 オーバーラップ数が増えたときの ARI の結果

Method	Number of Overlapped Data						
	2	3	4	5	6	7	8
VB(topline)	0.82 ±0.07	0.80 ±0.07	0.79 ±0.07	0.79 ±0.07	0.78 ±0.09	0.77 ±0.08	0.73 ±0.09
VBNG	0.8 ±0.08	0.78 ±0.08	0.78 ±0.07	0.74 ±0.08	0.72 ±0.07	0.68 ±0.08	0.63 ±0.07
RMHNG [5]	0.64 ±0.11	0.58 ±0.11	0.43 ±0.16	0.19 ±0.17	0.01 ±0.04	0.00 ±0.00	0.00 ±0.00

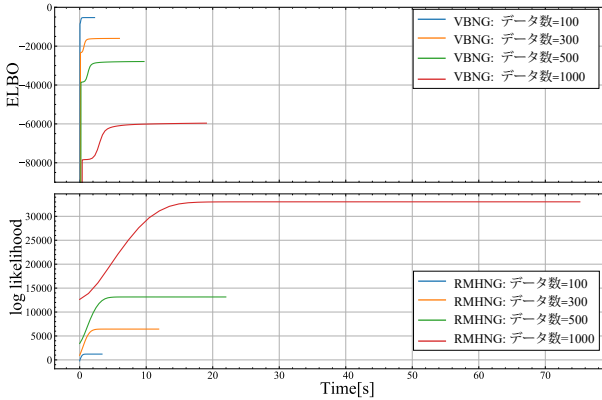


図 4 学習データ数を変化させたときの目的関数値の推移論できていることが分かる。

3.2.2 計算時間

図 4 は、学習データ数を変化させた場合の、計算時間に対する目的関数値の推移を示している。ただし、エージェント数は 10、オーバーラップ数は 2 に固定した。学習データ数を増やすと、各目的関数値が収束するまでの時間が長くなる傾向が見られた。学習データ数が 100 の場合、VBNG と RMHNG の収束点には大きな差は見られなかった。一方、学習データ数が 1000 の場合では、VBNG の方が早く収束した。

図 5 は、エージェントの数を変化させた場合の、計算時間に対する目的関数値の推移を示している。ただし、学習データ数は 300、オーバーラップ数は 2 に固定した。エージェント数が少ない場合では、VBNG と RMHNG の収束点に大きな差は見られなかった。しかし、エージェント数を 9 や 10 に増やすと、VBNG の方が RMHNG よりも早く収束することが確認された。

4 考察

オーバーラップ数が増加した場合、特定のクラスを区別できるエージェントが限られる。RMHNG は 2 エージェント間でのコミュニケーションを繰り返す必要があるため、観測を識別することができない

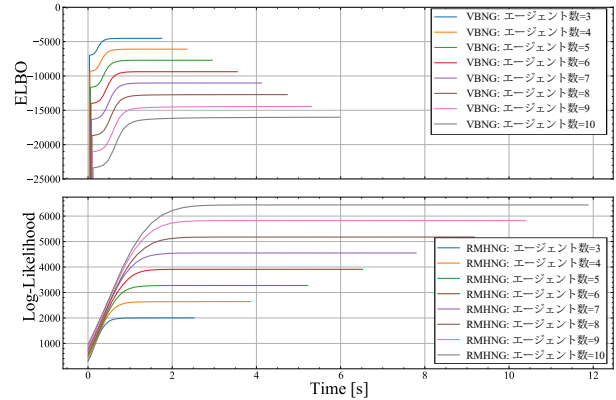


図 5 エージェント数を変化させたときの目的関数値の推移

エージェント同士でのコミュニケーションが繰り返されることで、識別するための情報が十分に共有されない可能性があると考えられる。一方、VBNG は全エージェント間で同時にコミュニケーションを行うため、各エージェントの内部状態を統合的に活用できるモデルである。

5 まとめ

本稿では、変分ベイズ法に基づく分散的ベイズ推論手法である名付けゲーム VBNG が、従来手法に比べて、より難易度の高い問題においても、観測を表現するサインの推論ができることを示した。また、エージェント数やデータ数が大規模な条件下では、従来手法よりも高い収束性を持つことが確認された。

今後は、実画像データを用いた実験を通じて、提案手法の有効性をさらに検証する予定である。

謝辞

本研究は、JST ムーンショット型研究開発事業 JPMJMS2011 の支援を受けたものである。

参考文献

- [1] Tadahiro Taniguchi. Collective predictive coding hypothesis: Symbol emergence as decentralized bayesian inference. **Frontiers in Robotics and AI**, Vol. 11, , 2024.
- [2] Yoshinobu Hagiwara, Hiroyoshi Kobayashi, Akira Taniguchi, and Tadahiro Taniguchi. Symbol emergence as an interpersonal multimodal categorization. **Frontiers in Robotics and AI**, Vol. 6, , 2019.
- [3] Tadahiro Taniguchi, Yuto Yoshida, Yuta Matsui, Nguyen Le Hoang, Akira Taniguchi, and Yoshinobu Hagiwara. Emergent communication through metropolis-hastings naming game with deep generative models. **Advanced Robotics**, Vol. 37, No. 19, pp. 1266–1282, 2023.
- [4] 中村友昭, 長野匡隼, 谷口彰, 谷口忠大. 変分ベイズ法に基づく分散的ベイズ推論. 日本ロボット学会学術講演会, No. 1L3-03, 2024.
- [5] Jun Inukai, Tadahiro Taniguchi, Akira Taniguchi, and Yoshinobu Hagiwara. Recursive metropolis-hastings naming game: symbol emergence in a multi-agent system based on probabilistic generative models. **Frontiers in Artificial Intelligence**, Vol. 6, , 2023.
- [6] Katrina Evtimova, Andrew Drozdov, Douwe Kiela, and Kyunghyun Cho. Emergent communication in a multimodal, multi-step referential game. In **International Conference on Learning Representations**, 2018.
- [7] Rahma Chaabouni, Florian Strub, Florent Altché, Eugene Tarassov, Corentin Tallec, Elnaz Davoodi, Kory Wallace Mathewson, Olivier Tieleman, Angeliki Lazaridou, and Bilal Piot. Emergent communication at scale. In **International Conference on Learning Representations**, 2022.
- [8] Sainbayar Sukhbaatar, Arthur Szlam, and Rob Fergus. Learning multiagent communication with backpropagation. In **Advances in Neural Information Processing Systems**, Vol. 29, 2016.
- [9] Jakob N. Foerster, Yannis M. Assael, Nando de Freitas, and Shimon Whiteson. Learning to communicate with deep multi-agent reinforcement learning. In **Advances in Neural Information Processing Systems**, Vol. 29, 2016.

A 付録

A.1 人工データセットの詳細

表 3 は、図 3 のデータ作成時に設定した、各クラスの観測のピーク位置を表している。例えば、Agent 1 のクラス 5 の観測データは、50 次元中の 12 次元目がピーク位置となるデータとなっていることを意味している。

表 4, 表 5, 表 6 は、それぞれオーバーラップ数を 4, 6, 8 とした場合のピーク位置を示している。これらの表では、識別が困難なクラスを色分けして強調している。

表 3 オーバーラップ数 2 の場合の各クラスの観測データのピーク位置

	Class Index									
	1	2	3	4	5	6	7	8	9	10
Agent 1	0	0	4	8	12	16	20	24	28	32
Agent 2	1	5	5	9	13	17	21	25	29	33
Agent 3	2	6	10	10	14	18	22	26	30	34
Agent 4	3	7	11	15	15	19	23	27	31	35
Agent 5	4	8	12	16	20	20	24	28	32	36
Agent 6	5	9	13	17	21	25	25	29	33	37
Agent 7	6	10	14	18	22	26	30	30	34	38
Agent 8	7	11	15	19	23	27	31	35	35	39
Agent 9	8	12	16	20	24	28	32	36	40	40
Agent 10	45	13	17	21	25	29	33	37	41	45

表 4 オーバーラップ数 4 の場合の各クラスの観測データのピーク位置

	Class Index									
	1	2	3	4	5	6	7	8	9	10
Agent 1	0	0	0	0	12	16	20	24	28	32
Agent 2	1	5	5	5	5	17	21	25	29	33
Agent 3	2	6	10	10	10	10	22	26	30	34
Agent 4	3	7	11	15	15	15	15	27	31	35
Agent 5	4	8	12	16	20	20	20	20	32	36
Agent 6	5	9	13	17	21	25	25	25	25	37
Agent 7	6	10	14	18	22	26	30	30	30	30
Agent 8	35	11	15	19	23	27	31	35	35	35
Agent 9	40	40	16	20	24	28	32	36	40	40
Agent 10	45	45	45	21	25	29	33	37	41	45

表 5 オーバーラップ数 6 の場合の各クラスの観測データのピーク位置

	Class Index									
	1	2	3	4	5	6	7	8	9	10
Agent 1	0	0	0	0	0	0	20	24	28	32
Agent 2	1	5	5	5	5	5	5	25	29	33
Agent 3	2	6	10	10	10	10	10	10	30	34
Agent 4	3	7	11	15	15	15	15	15	15	35
Agent 5	4	8	12	16	20	20	20	20	20	20
Agent 6	25	9	13	17	21	25	25	25	25	25
Agent 7	30	30	14	18	22	26	30	30	30	30
Agent 8	35	35	35	19	23	27	31	35	35	35
Agent 9	40	40	40	40	24	28	32	36	40	40
Agent 10	45	45	45	45	45	29	33	37	41	45

表 6 オーバーラップ数 8 の場合の各クラスの観測データのピーク位置

	Class Index									
	1	2	3	4	5	6	7	8	9	10
Agent 1	0	0	0	0	0	0	0	0	28	32
Agent 2	1	5	5	5	5	5	5	5	5	33
Agent 3	2	6	10	10	10	10	10	10	10	10
Agent 4	15	7	11	15	15	15	15	15	15	15
Agent 5	20	20	12	16	20	20	20	20	20	20
Agent 6	25	25	25	17	21	25	25	25	25	25
Agent 7	30	30	30	30	22	26	30	30	30	30
Agent 8	35	35	35	35	35	27	31	35	35	35
Agent 9	40	40	40	40	40	40	32	36	40	40
Agent 10	45	45	45	45	45	45	45	37	41	45