

制限付きベイジアンネットワーク BESOM の正則化の一手法

A Method of Regularization for the BESOM Bayesian Network

一杉裕志^{1*}

¹ 産業技術総合研究所

¹ National Institute of Advanced Industrial Science and Technology(AIST)

Abstract: We describe a method of regularization for the restricted Bayesian network BESOM, which has network structure like deep learning. Two types of priors, win-rate penalty and lateral-inhibition penalty are introduced to avoid overfitting and local minimum problems.

1 はじめに

計算論的神経科学における注目すべき進展として、「大脳皮質がベイジアンネットワーク [8] である」という仮説の登場がある [11]¹。大脳皮質は、脳の中で知能にもっとも関係する重要な器官である。大脳皮質はベイジアンネットワークと機能と構造の面で多くの類似性を持ち、実際に様々な神経科学的現象がベイジアンネットワークを用いたモデルで再現されている (例えば [3][4][6][13][14])。

一方で、機械学習の分野では Deep Learning と呼ばれる深い層構造を持つニューラルネットワークが、高い認識性能を発揮することで注目を集めている。この技術は、脳の視覚野の腹側経路の構造を模した神経回路モデルである、ネオコグニトロン [1] を源流とする。

最新の計算論的神経科学の知見を Deep Learning 技術に取り込み、Deep Learning と同じ層構造を持ったベイジアンネットワークを用いれば、より本物の脳に近づき、より高機能・高性能な機械学習技術になる可能性がある。

そのような目的から、筆者らは BESOM と呼ぶ機械学習アルゴリズム [4][5][7] を開発中である。BESOM は条件付確率表に制限を入れた層構造をしたベイジアンネットワークである。BESOM は機械学習アルゴリズムとして未完成だが、工学的有用性と、大脳皮質の計算論的モデルとしての神経科学的妥当性の両方を目指している。

単に脳との類似性という理由だけでなく、技術的な観点からも、以下の点で、ベイジアンネットワークを用いた Deep Learning は有望だと思われる。

- ベイジアンネットワークは複数の事象の間の因果関係を確率により表現する知識表現の技術である。ベイジアンネットワークは信号源と観測データの間の因果関

係を比較的少ないメモリで簡潔に表現できる場合があり、その場合は少ない計算量で様々な推論を行うことができる。

- 推論動作は、ネットワーク全体の情報を使って行われる。入力からのボトムアップの情報だけでなく、文脈からのトップダウンの予測の情報も用いたロバストな認識が行える [3]。したがって、feedforward 型ニューラルネットワークよりもはるかに高機能である。
- 階層的な生成モデルを素直に表現できるため、学習対象の事前知識が作り込みやすい。事前知識をネットワーク構造や、パラメタの事前分布の形で作り込むことで、学習の性能を上がられる可能性がある。これらの事前知識は、学習対象に対する工学的な観点からの分析だけでなく、神経科学的知見 (領野間の接続構造や各領野が表現する情報の特徴) から得られる可能性がある。

以上の利点があるにも関わらず、ベイジアンネットワークを用いた大規模 Deep Learning があまり使われていない理由として、計算量の問題と、過適合・局所解の問題の2つがあるのではないかとと思われる。

計算量の問題については、制限された条件付確率表を前提として最適化した確率伝搬アルゴリズムを用いることで解決しつつある [16]。

もう1つの過適合・局所解の問題については、大規模ベイジアンネットワークの表現力の高さゆえに起きると考えられる。この問題はパラメタに事前分布を与えることで緩和できる。

本稿では勝率ペナルティと側抑制ペナルティと呼ぶ2種類の事前分布と、それを用いた時の近似的な学習規則について述べる²。この2つの機構は同時に適用可能

*連絡先： 産業技術総合研究所
茨城県つくば市梅園 1-1-1 中央第2
E-mail: y-ichisugi@aist.go.jp

¹参考: 「脳とベイジアンネットワーク」
<https://staff.aist.go.jp/y-ichisugi/besom/j-index.html>

²なお本稿の内容は草稿として公開済みの文章 [15] の一部をもとにしている。

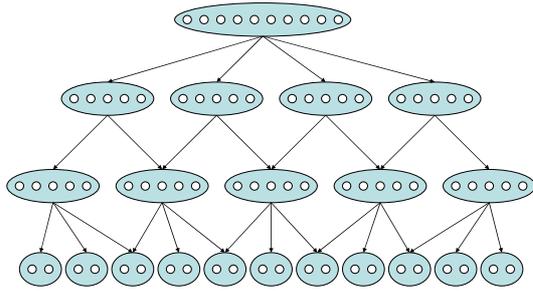


図 1: BESOM ネットワークの例。楕円はノード（確率変数）、その中の白丸はユニット（確率変数がとり得る値）。

である。勝率ペナルティはノード内の各値が均等に用いられるように、側抑制ペナルティは同じ層内のノード同士が独立になるように、認識結果にバイアスを与える。

2 BESOM のアーキテクチャ

BESOM は Deep Learning と同じ、深い層構造をした特殊なベイジアンネットである（図 1）。多くの Deep Learning のアーキテクチャと同様、同一の層内には結合を持たず、層間は局所受容野を持つように結合する。

BESOM では変数をノード、変数がとり得る値をユニットと呼ぶ。一般にノードは多値変数である。

白黒画像を学習させる場合、入力のパixel値を最下層の 2 値変数の観測値として与える。

教師なし学習器として用いる場合は、最下端の変数以外は、すべて隠れ変数となる。この場合は、最上位層に特徴抽出された値が表現される。

教師あり学習器として用いる場合は、教師信号の与え方はいろいろあり得る。例えば最上位層の 1 つのノード（教師ノード）の観測値として与える方法がある。この場合は、認識時には教師ノードも隠れ変数にして最下層に値を入力し、教師ノードの値の推論結果（事後確率最大の値）を認識結果とする。

BESOM のもう 1 つの大きな特徴は、条件付確率表の制限である。この制限については、5 章で述べる。

3 学習の目的関数

パラメタ θ のもとでの隠れ変数の値の組 \mathbf{h} と入力変数の値の組 \mathbf{i} との間の同時確率のモデルを $P(\mathbf{h}, \mathbf{i}|\theta)$ とする。また、時刻 t における入力変数の値の組を $\mathbf{i}(t)$ とする。各時刻の入力は i.i.d. (独立同分布) に従うと仮

定すると、 θ のもとで入力データの列 $\mathbf{i}(1), \mathbf{i}(2), \dots, \mathbf{i}(t)$ が生じる確率は以下ようになる。

$$\begin{aligned} & \prod_{i=1}^t P(\mathbf{i}(i)|\theta) \\ &= \prod_{i=1}^t \sum_{\mathbf{h}} P(\mathbf{h}, \mathbf{i}(i)|\theta) \end{aligned} \quad (1)$$

学習の目的は、以下のようにパラメタを MAP 推定すること、すなわちパラメタ θ の事後確率を最大にすることである。

$$\theta^* = \operatorname{argmax}_{\theta} \left[\prod_{i=1}^t \sum_{\mathbf{h}} P(\mathbf{h}, \mathbf{i}(i)|\theta) \right] P(\theta) \quad (2)$$

パラメタ θ を推定するアルゴリズムとしては、オンライン EM アルゴリズムもしくはその近似アルゴリズムを用いる。

近似の 1 つの方法として、MPE (*Most Probable Explanation*) を用いる方法を以下に述べる。

学習アルゴリズムは、認識ステップと学習ステップの動作を表す 2 つの数式で書ける。

認識ステップでは、まず現在のパラメタ $\theta(t)$ に基づいて、与えられた入力 $\mathbf{i}(t)$ に対する隠れ変数の値の組の最大事後確率推定値 $\hat{\mathbf{h}}(t)$ (すなわち MPE) を次のように求める。

$$\begin{aligned} \hat{\mathbf{h}}(t) &= \operatorname{argmax}_{\mathbf{h}} P(\mathbf{h}|\mathbf{i}(t), \theta(t)) \\ &= \operatorname{argmax}_{\mathbf{h}} P(\mathbf{h}, \mathbf{i}(t)|\theta(t))/P(\mathbf{i}(t)) \\ &= \operatorname{argmax}_{\mathbf{h}} P(\mathbf{h}, \mathbf{i}(t)|\theta(t)) \end{aligned} \quad (3)$$

次に学習ステップでは、式 (2) における隠れ変数の周辺化を推定値 $\hat{\mathbf{h}}(i)$ を用いることで近似してパラメタ推定し、結果を $\theta(t+1)$ とする³。

$$\theta(t+1) = \operatorname{argmax}_{\theta} \left[\prod_{i=1}^t P(\hat{\mathbf{h}}(i), \mathbf{i}(i)|\theta) \right] P(\theta) \quad (4)$$

4 パラメタ θ の事前分布

パラメタ θ の事前分布は、下記のように 2 つの因子から構成される。

$$P(\theta) = P^{WinRate}(\theta) P^{Lateral}(\theta) \quad (5)$$

このあと、勝率ペナルティ $P^{WinRate}(\theta)$ と側抑制ペナルティ $P^{Lateral}(\theta)$ について、それぞれ 6 章と 7 章で述べる。

³似た近似は Olshausen らの sparse-coding [9] でも行っている ([10] の p.3315 参照)。なお、我々はオンライン EM アルゴリズムも実装している。本稿では述べないが、概要については [16] を参照されたい。

5 条件付確率表のモデル

BESOM の重要な特徴として、条件付確率表の制限がある。(ただし本論文の主題である勝率ペナルティおよび側抑制ペナルティの機構は、この制約とは無関係に動作すると思われる。)

ベイジアンネットワークにおいて各ノードの条件付確率表を表現するためには、一般に親ノードの数 m に対して $O(2^m)$ 個のパラメタが必要になる。これは計算量・メモリ量の爆発や、過適合・局所解の原因になる。

そこで、条件付確率をより少ないパラメタで表現できるように、以下の形に制限する。

$$P(x|u_1, \dots, u_m) = \frac{1}{m} \sum_{k=1}^m w(x, u_k) \quad (6)$$

$w(x, u_k)$ のもっとも簡単なものとして、下記のものを現在用いている。

$$w(x, u_k) = P(x|u_k) \quad (7)$$

条件付確率 $P(x|u_k)$ は1つのパラメタ w_{xu_k} で表現される。

このような制約を入れると確率伝搬アルゴリズムが非常に少ない計算量で近似実行でき、しかもそのときのノード間の情報の流れが脳皮質の特徴的な解剖学的構造とよく一致することが示されている [4][6]。

一方で、このように条件付確率表に強い制限を入れることは、モデルとしての表現力を下げることになる。この問題についての考察は9章で述べる。

6 勝率ペナルティ

6.1 目的

BESOM ネットワークのパラメタを素朴な方法で学習すると、多値ノード内の多数のユニットのうちごく一部しか学習が進まず、ほとんどのユニットは初期値のまま変化しないという現象が起きる。これは学習が局所解に陥っているものと思われる。

この問題を緩和するためには、パラメタの事前分布を適切に設定し、学習結果において各ノード内のユニットが均等に使われるベイジアンネットワークになるようなバイアスを与えればよい。

具体的には、各ユニットが目標とする勝率の分布と、実際の分布とのKL情報量に基づいて、差が大きいときにペナルティを与えるものとする。ここで、勝率とは、ユニットがノードの推定値になる頻度である。BESOM においてユニットとは確率変数がとり得る値であり、1

つのノード(確率変数)の推定値に対応するユニットを勝者ユニットと呼んでいる。

このようなペナルティによって、学習が進むにつれ各ユニットの勝率が目標とする値に近づくと期待できる。他のユニットよりも勝率が大きいユニットにペナルティを与えることから、この機構を **勝率ペナルティ** と呼んでいる。

6.2 複雑な事前分布の問題点とその解決策

隠れ変数を含むベイジアンネットワークのパラメタの最尤解はEMアルゴリズムで推定でき、それは確率伝搬アルゴリズムのメッセージの収束結果を用いて効率的に実行できる [12]。しかし、パラメタが複雑な事前分布を持つ場合は、パラメタのMAP解を求めるEMアルゴリズムが容易に実行できるとは限らない。

もし、パラメタの事前分布を持つベイジアンネットワークが、それと等価な事前分布を持たないベイジアンネットワークに変換できれば、変換後のベイジアンネットワーク上でのEMアルゴリズムの実行は容易になる。

本論文で述べる事前分布の定義は、幸いなことに、近似的にほぼ等価な事前分布を持たないベイジアンネットワークに変換できる。変換後のベイジアンネットワークにおいては、認識結果にバイアスを与える役割を持つ **制約ノード** が新たに加わる。

6.3 事前分布の定義と、それと等価なベイジアンネットワークの導出

勝率ペナルティ $P^{WinRate}(\theta)$ を以下のように定義する。

$$P^{WinRate}(\theta) = \prod_{X \in \mathbf{X}} e^{-C^{WinRate} D_{KL}(Q(X)||P(X;\theta))} \quad (8)$$

ただし \mathbf{X} はすべてのノードの集合、 $C^{WinRate}$ は勝率ペナルティの強さを決める定数である。

$Q(X)$ は、ノード X の勝率の目標となる分布であり、ネットワーク設計者が分布の形を決める。例えばノード内のユニットの勝率を一様にした場合、ノード X のユニット数を s 個とすれば、すべての i ($i = 1, 2, \dots, s$) に対して

$$Q(X = x_i) = 1/s \quad (9)$$

とすればよい。

$D_{KL}(Q(X)||P(X;\theta))$ は分布 P と Q の間のKL情報量で、以下の式で定義される。

$$D_{KL}(Q(X)||P(X;\theta)) = \sum_x Q(x) \log \frac{Q(x)}{P(x;\theta)} \quad (10)$$

ここで

$$R(x; \theta) = s \frac{Q(x)}{P(x; \theta)} \log \frac{Q(x)}{P(x; \theta)} \quad (11)$$

と定義する。ただし s はノード X のユニット数である。 $x(i)$ を θ のもとでの時刻 i における変数 X の MPE の値の十分によい近似値とすれば⁴、

$$D_{KL}(Q(X)||P(X; \theta)) \approx \sum_{i=1}^t \frac{1}{t} R(x(i); \theta) \quad (12)$$

である。

すると $P^{WinRate}(\theta)$ は次のように書き直せる。

$$\begin{aligned} P^{WinRate}(\theta) &= \prod_{X \in \mathbf{X}} e^{-C^{WinRate} D_{KL}(Q(X)||P(X; \theta))} \\ &\approx \prod_{X \in \mathbf{X}} e^{-C^{WinRate} \sum_{i=1}^t (1/t) R(x(i); \theta)} \\ &= \prod_{X \in \mathbf{X}} \prod_{i=1}^t e^{-(1/t) C^{WinRate} R(x(i); \theta)} \\ &= \prod_{i=1}^t \prod_{X \in \mathbf{X}} e^{-(1/t) C^{WinRate} R(x(i); \theta)} \quad (13) \end{aligned}$$

このとき、学習ステップにおけるパラメタ θ を MAP 推定する式 (4) は次のように書き直せる。(ここでは説明を簡単にするために $P(\theta) = P^{WinRate}(\theta)$ とする。)

$$\begin{aligned} \theta(t+1) &= \operatorname{argmax}_{\theta} \left[\prod_{i=1}^t P(\hat{\mathbf{h}}(i), \mathbf{i}(i)|\theta) \right] P^{WinRate}(\theta) \\ &= \operatorname{argmax}_{\theta} \left[\prod_{i=1}^t \prod_{X \in \mathbf{X}} P(x(i)|\text{parents}(x(i)); \theta) \right] \\ &\quad \left[\prod_{i=1}^t \prod_{X \in \mathbf{X}} e^{-(1/t) C^{WinRate} R(x(i); \theta)} \right] \\ &= \operatorname{argmax}_{\theta} \prod_{i=1}^t \prod_{X \in \mathbf{X}} \left[P(x(i)|\text{parents}(x(i)); \theta) \right. \\ &\quad \left. e^{-(1/t) C^{WinRate} R(x(i); \theta)} \right] \quad (14) \end{aligned}$$

ただし、 $\text{parents}(x(i))$ は時刻 i におけるノード X の親ノードの値の組である。

上記の式は、すべてのノード X の下に 2 値の制約ノード R_X があり、条件付き確率が

$$P(R_X = 1|X = x; \theta) = e^{-(1/t) C^{WinRate} R(x; \theta)} \quad (15)$$

⁴学習がある程度収束に近づいた時にはじめてこの仮定が成り立つ点には注意が必要である。7 章の側抑制ペナルティについても同様である。

で定義されているようなベイジアンネットワークにおけるパラメタ θ の最尤推定と同じ形をしている。(ノード R_X には常に観測値 1 が与えられるものとする。)したがって、EM アルゴリズムを用いたパラメタ推定が容易に実行できるようになる。

正則化パラメタ $C^{WinRate}$ に $1/t$ がかかっているの
で、時刻とともにペナルティの影響は小さくなっていく。

$R(x; \theta)$ を計算する際に必要となる勝率 $P(x; \theta)$ は素朴なオンライン学習によって推定したものを使う。

6.4 勝率ペナルティの強さの近似式

$f(x) = x \log(x)$ のとき $f'(x) = \log(x) + 1$, $f(1) = 0$, $f'(1) = 1$ 、また、 P と Q がほぼ一致するならば $Q(x)/P(x; \theta) \approx 1$ なので、下記の近似式が成り立つ。

$$\begin{aligned} R(x; \theta) &= s(Q(x)/P(x; \theta)) \log Q(x)/P(x; \theta) \\ &\approx s(Q(x)/P(x; \theta) - 1) \quad (16) \end{aligned}$$

この式を用いることで計算時間を多少減らすことができる。また、生物にとってもこのような近似式の方が神経回路による実現が容易かもしれない。

7 側抑制ペナルティ

7.1 目的

BESOM ネットワークのパラメタを素朴な方法で学習すると、同じ子ノードからの入力を受け取る同一層内のノード同士が似た情報を表現するようになるという現象が起きがちである。この現象も局所解または過適合であろうと思われる。

BESOM の各隠れ層は Deep Learning と同様、特徴抽出器として働くことが期待されるが、同じ特徴が重複して表現されていると、上位層の認識にとって好ましくなく、問題である。

この問題を緩和するため、前章と同様に、ペナルティを事前分布として定義し、それと等価な構造を持つベイジアンネットを導くアプローチをとる。

事前分布は、2つのノードが似た値を表現するときペナルティを与えることで、ノード同士が独立な情報を表現するようにバイアスをかける。

この機構は脳皮質内における側抑制の機構と似た役割を持つと思われることから、**側抑制ペナルティ**と呼んでいる。

7.2 事前分布の定義と、それと等価なベイジアンネットの導出

側抑制に相当するペナルティ $P^{Lateral}(\theta)$ を以下のように定義する。

$$P^{Lateral}(\theta) = \prod_{(U,V) \in L} e^{-C^{Lateral} I(U,V;\theta)} \quad (17)$$

ただし $C^{Lateral}$ は側抑制ペナルティの強さを決める定数、 L は側抑制を行うノードのペアの集合とする。通常、同じ層に属し、同じ子ノードを共有するノードどうしがお互いに側抑制するように、側抑制のネットワークを構築しておく。

$I(U, V; \theta)$ はノード U と V の間の相互情報量で、以下の式で定義される。

$$I(U, V; \theta) = \sum_u \sum_v P(u, v; \theta) \log \frac{P(u, v; \theta)}{P(u; \theta)P(v; \theta)} \quad (18)$$

ここで

$$\begin{aligned} R(u, v) &= s^2 \frac{P(u, v)}{P(u)P(v)} \log \frac{P(u, v)}{P(u)P(v)} \\ &= s^2 (P(u|v)/P(u)) \log P(u|v)/P(u) \end{aligned} \quad (19)$$

と定義する。 $(\theta$ は省略した。) $u(i), v(i)$ を θ のもとの時刻 i における変数 U, V の MPE の値の十分によい近似値とすれば、

$$I(U, V; \theta) \approx \sum_{i=1}^t \frac{1}{t} R(u(i), v(i); \theta) \quad (20)$$

である。

あとは6章と同様の議論により、側抑制するノードのペア $(U, V) \in L$ ごとに2値の共通子ノード R_{UV} があってその条件付き確率が

$$P(R_{UV} = 1|u, v; \theta) = e^{-(1/t)C^{Lateral} R(u, v; \theta)} \quad (21)$$

であるとみなすことができる。したがって前章と同様に、EMアルゴリズムによってパラメタの最尤値が容易に推定できるようになる。

なお、 $R(u, v; \theta)$ の値の計算に必要な $P(u|v; \theta)$ 、 $P(u; \theta)$ の値は、6章と同様、素朴なオンライン学習により推定したものを使う。

	勝率ペナルティあり	勝率ペナルティなし
側抑制ペナルティあり	84.7%	84.0%
側抑制ペナルティなし	79.9%	61.3%

図 2: MNIST 手書き数字データベースでの認識率の評価。

7.3 側抑制の強さの近似式

U と V がほぼ独立ならば $P(u|v)/P(u) \approx 1$ なので、下記の近似式が成り立つ。

$$\begin{aligned} R(u, v) &= s^2 (P(u|v)/P(u)) \log P(u|v)/P(u) \\ &\approx s^2 (P(u|v)/P(u) - 1) \end{aligned} \quad (22)$$

8 評価

本手法の効果を MNIST 手書き数字データベース⁵での認識率で評価した。

実験には4層 BESOM ネットワークを使用した。最下層にはサイズ 28 x 28 の入力画像のピクセル値を二値化して与える2ユニットのノード 28 x 28 個、最上位層に教師信号を与える10ユニットのノード1個を配置した。隠れ層は2層あり、入力層の1つ上の層は20ユニットのノードを5 x 5 個、さらにその上の層は30ユニットのノードを3 x 3 個配置した。

認識率は、訓練データ6万個のうちのランダムな1万個を学習させた後、テストデータ1万個のうちランダムな1000個を認識させて評価した。

学習は、ここではオンラインEMアルゴリズムの非常に粗い近似を用いている。まず loopy な確率伝搬アルゴリズム (ただし条件付確率表の制限を前提に計算量を落としたもの [16]) でノードごとの周辺事後確率を計算し、そのノードごとの最大値を真の値と見なし、パラメタをオンラインで更新している。

図2は結果をまとめたものである。勝率ペナルティと側抑制ペナルティは、いずれもそれがない場合と比べて認識率が上がっていることが確かめられた。

9 おわりに

現在、BESOM は機械学習アルゴリズムとしてなんとか動作するようになったが、まだ実用上十分な認識

⁵ 「MNIST handwritten digit database」
<http://yann.lecun.com/exdb/mnist/>

精度は出ていない。現状の BESOM の MNIST 手書き数字の認識率は、単純な 3 層パーセプトロン等と比べてもよくない。

最大の理由は、5 章で述べた条件付確率表のモデルの制約が強すぎる点にあると考えている。現在の条件付確率表のモデルではノード数やユニット数を十分に増やしても、任意の生成モデルが十分な精度で近似できないと思われる。この問題を解決する新たな条件付確率表のモデルと、それに合わせた近似確率伝搬アルゴリズムを開発する必要がある、現在取り組んでいるところである。

なお、本稿で述べたアルゴリズムの実装は、ソースコードつきで公開している⁶。

参考文献

- [1] K. Fukushima, Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biological Cybernetics*, 36(4): 93-202, 1980.
- [2] J. Pearl, *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*, Morgan Kaufmann, 1988.
- [3] George, D. Hawkins, J., A hierarchical Bayesian model of invariant pattern recognition in the visual cortex, In *proc. of IJCNN 2005*, vol. 3, pp.1812-1817, 2005.
- [4] Yuuji ICHISUGI, The cerebral cortex model that self-organizes conditional probability tables and executes belief propagation, In *Proc. of International Joint Conference on Neural Networks (IJCNN 2007)*, pp.1065-1070, Aug 2007. <http://staff.aist.go.jp/y-ichisugi/besom/20070509ijcnn-paper.pdf>
- [5] 一杉裕志、「脳の情報処理原理の解明状況」産業技術総合研究所テクニカルレポート AIST07-J00012, Mar 2008. <http://staff.aist.go.jp/y-ichisugi/besom/AIST07-J00012.pdf>
- [6] Yuuji Ichisugi: "Recognition Model of Cerebral Cortex based on Approximate Belief Revision Algorithm", To appear in *Proc. of IJCNN 2011*. <http://staff.aist.go.jp/y-ichisugi/besom/2011ijcnn.pdf>
- [7] 一杉裕志、「大脳皮質のアルゴリズム BESOM Ver.2.0」産業技術総合研究所テクニカルレポート AIST11-J00009, Sep 2011. <http://staff.aist.go.jp/y-ichisugi/besom/AIST11-J00009.pdf>
- [8] J. Pearl, *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*, Morgan Kaufmann, 1988.
- [9] Olshausen BA, Field DJ, Emergence of simple-cell receptive field properties by learning a sparse code for natural images, *NATURE* 381 (6583): 607-609 JUN 13 1996.
- [10] Olshausen BA, Field DJ, Sparse Coding with an Overcomplete Basis Set: A Strategy Employed by V1?, *Vision Res.*, Vol. 37, No. 23, pp. 3311-3325, 1997.
- [11] 一杉裕志, 解説: 大脳皮質とベイジアンネット、日本ロボット学会誌 Vol.29 No.5, pp.412-415, 2011.
- [12] Kevin B. Korb and Ann E. Nicholson, *Bayesian Artificial Intelligence*, Second Edition, CRC Press, 2011.
- [13] Haruo Hosoya, Multinomial Bayesian Learning for Modeling Classical and Nonclassical Receptive Field Properties, *Neural Computation*, Vol. 24, No. 8, pp. 2119-2150, 2012.
- [14] Salvador Dura-Bernal, Thomas Wennekers, Susan L. Denham, Top-Down Feedback in an HMAX-Like Cortical Model of Object Perception Based on Hierarchical Bayesian Networks and Belief Propagation *PLoS ONE*, Vol.7, No.11., 2012.
- [15] 一杉裕志、BESOM Ver.3.0 β版のアルゴリズム、2013. <https://staff.aist.go.jp/y-ichisugi/besom/20130717algorithm.pdf>
- [16] 一杉裕志、高橋直人、制限付きベイジアンネット BESOM における認識アルゴリズム OOBP, 2014 年度 人工知能学会全国大会, 2014. <https://kaigi.org/jsai/webprogram/2014/pdf/373.pdf>

⁶<https://staff.aist.go.jp/y-ichisugi/besom/download.html>