A Computational Model of Motor Areas Based on Bayesian Networks and Most Probable Explanations

Yuuji Ichisugi

National Institute of Advanced Industrial Science and Technology(AIST)
Tsukuba Central 2,Tsukuba,Ibaraki 305-8568, Japan

Abstract. We describe a computational model of motor areas of the cerebral cortex. The model combines Bayesian networks, competitive learning and reinforcement learning. We found that decision-making using MPE (Most Probable Explanation) approximates the ideal decision-making in this model, which suggests that MPE calculation is a promising model of not only sensory-cortex recognition, already addressed by previous works, but also motor-cortex decision-making.

1 Introduction

Motor areas and prefrontal areas are important areas of the brain that are related to motor control and decision-making. Elucidating the basic mechanism of these areas is a necessary and important step for understanding the mechanism of the whole brain, because these areas are connected to other important organization, such as the sensory-cortex, the basal ganglia and the cerebellum. All cortical areas, including the motor and prefrontal areas, may be thought to function via the same basic mechanism, because they have essentially the same anatomical structure, that is, a six-layered and columnar structure.

In the case of the visual cortex, some computational neuroscientists have begun to understand that the basic mechanism is a *Bayesian network* [1]. Bayesian networks are a technology for knowledge representation that can efficiently express the causal relationships among many random variables. Computational models based on Bayesian networks can not only exhibit robust pattern recognition[7][15] but also elegantly explain various electrophysiological phenomena[5][6][12][13][16][17][18], psychophysical phenomena[12] and anatomical structures[7][8][9][14].

Assuming that the cerebral cortex is a kind of Bayesian network, there would be more than one candidate for a computational model of recognition mechanism. A promising candidate is the calculation of *MPE (Most Probable Explanation)*[1]. A circuit that has a layered and columnar structure like the cerebral cortex can efficiently execute an approximate calculation of MPE[11][14]. The amount of computation is linear with respect to the number of nodes if edges are sparse[14].

However, no previous work has addressed the relation between MPE calculation and motor-cortex decision-making.

Yuuji Ichisugi,
A Computational Model of Motor Areas Based on Bayesian Networks and Most Probable Explanations,
In Proc. of The International Conference on Artificial Neural Networks (ICANN 2012), Sep 2012. (to appear)
The original publication will be available at www.springerlink.com
http://www.springer.com/computer

In this paper, with theoretical study and computer experiments, we show that decision-making using MPE approximates the ideal decision-making if hidden nodes in a Bayesian network execute competitive learning. This result is new indirect evidence supporting the hypothesis that "MPE calculation is the basic mechanism of the cerebral cortex including sensory areas and motor areas."

2 Bayesian networks and MPE

A Bayesian network is a model of knowledge representation that expresses causal relationships between random variables using a directed acyclic graph. Random variables are expressed as *nodes*, and relationships between random variables are expressed as *edges*. Each node has a table of conditional probability, which denotes the degree to which nodes are related to the set of its parent nodes.

In a Bayesian network, an MPE is the set of values of nodes that most likely explains given observed data. Let $\mathbf{i}$ be a set of values of observed random variables and $\mathbf{h}$ be a set of values of hidden variables (unobserved random variables). MPE $\hat{\mathbf{h}}$ is defined by the following equations:

$$\hat{\mathbf{h}} = \operatorname*{argmax}_{\mathbf{h}} P(\mathbf{h}|\mathbf{i}) = \operatorname*{argmax}_{\mathbf{h}} P(\mathbf{h}, \mathbf{i}), \quad P(\mathbf{h}, \mathbf{i}) = \prod_{x \in \mathbf{h} \cup \mathbf{i}} P(x|parents(x)) \tag{1}$$

where $parents(x)$ denotes the set of values of parent nodes of node X.

3 Model of motor areas using a Bayesian network

We use an algorithm of reinforcement learning that is similar to the "sarsa"[2] algorithm. An agent learns an *action-value function* with trial and error. In our model, a probabilistic model of the action-value function is represented by a Bayesian network [1].

Figure 1 is a Bayesian network that hypothetically represents the anatomical structure of motor areas and related organizations. This structure is expressed by the following equation:

$$P(q, s, a, v) = P(s|q)P(a|q)P(v|q)P(q) \tag{2}$$

Nodes S, A and V are random variables that represent *a state*, *an action* and *a value of the state-action pair*, respectively. Node Q is a hidden variable that learns about the relationships among S, A and V. The learning is achieved by a kind of competitive learning algorithm .

We do not distinguish reward r ($0 <= r <= 1$) with probability 1 from reward 1 with probability r, because both of them have the same expectation. We model the variable V as a binary variable whose value is 0 or 1. In this case,

[1] Although the essential part of the reinforcement learning algorithm in this paper is the same as the work presented by Hosoya[10], our algorithm is based on formalization that is more detailed.

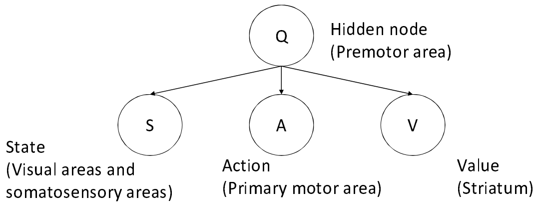

Fig. 1. A Bayesian network that hypothetically represents the anatomical structure of motor areas and related organizations.

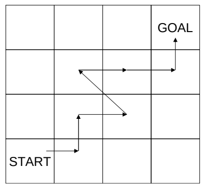

Fig. 2. Reaching task (a maze task with no walls).

when the state is s and the action is a, an expectation of accumulated reward is expressed as $P(V=1|s,a)$.

Given the model of action value function $P(q,s,a,v)$ and a current state s, an action a^{Ideal} that maximizes expectation of accumulated reward, in other words, an action based on *the ideal decision-making*, is as follows:

$$a^{Ideal} = \underset{a}{\operatorname{argmax}} P(V=1|s,a) = \underset{a}{\operatorname{argmax}} \frac{\sum_q P(q,s,a,V=1)}{\sum_q \sum_v P(q,s,a,v)} \tag{3}$$

From the point of view of computation amount, naive calculation of this value is not efficient, because it contains marginalization of the variable Q. It is no problem if the number of hidden nodes is small; however, the calculation becomes intractable if the number is as large as the number of macro-columns in the premotor area. Therefore, we do not think that the brain actually performs such a calculation.

On the other hand, assuming that "recognition by the visual cortex is an MPE calculation[11][14]," it must be more natural to assume that "the motor area also uses MPE calculation for decision-making." Based on this assumption, the action a^{MPE} selected by *MPE decision-making* is defined as follows:

$$(q^{MPE}, a^{MPE}) = \underset{(q,a)}{\operatorname{argmax}} P(q,s,a,V=1) \tag{4}$$

In general, a^{Ideal} and a^{MPE} may be different values; however, surprisingly, we found these two values are approximately coincident if the action-value function is learned using competitive learning. The reason for this is explained as follows.

When competitive learning is ideally completed, each state-action pair (s,a) will completely correspond to a value of node Q, q_{sa}. Therefore, the following approximations will hold:

$$P(s|q_{sa}) \approx 1, \quad P(a|q_{sa}) \approx 1, \quad P(V=1|q_{sa}) \approx P(V=1|s,a) \tag{5}$$

In addition, assuming that the value of $P(q)$ is not dependent on q, the following approximation will hold:

$$a^{MPE} \tag{6}$$

$$\approx \underset{a}{\operatorname{argmax}} P(q_{sa}, s, a, V=1) = \underset{a}{\operatorname{argmax}} P(s|q_{sa})P(a|q_{sa})P(V=1|q_{sa})P(q_{sa})$$
$$\approx \underset{a}{\operatorname{argmax}} P(V=1|q_{sa}) \approx \underset{a}{\operatorname{argmax}} P(V=1|s,a) = a^{Ideal}$$

The validity of this theoretical study is confirmed by computer experiments described in the next section.

4 Computer Experiments

4.1 The task

To evaluate the performance of the proposed model as a reinforcement-learning algorithm, we use a reaching task (Fig.2). An agent walks into a two-dimensional discrete space that has 4x4 positions. The agent starts from the start point at the lower left corner and goes toward the goal point at the upper right corner. It moves in one of eight directions by steps. If it reaches the goal, it gets a reward of 1. Otherwise, at each step, it gets a minus reward worth a small value. We set the reward discount factor γ=1.

The moving direction of the agent is not deterministic because of *action noise*. The agent can move in its desired direction with only probability 0.8. With probability 0.1, it moves in the direction rotated 45 degrees to the left. With probability 0.1, the direction is rotated 45 degrees to the right. This stochastic action noise has a role of exploration of the optimal path.

Before the reinforcement learning phase, we introduced a pre-training phase, because the agent has a very small chance of reaching the goal if it starts learning from a randomly initialized action-value function. In that phase, the agent first goes toward the lower right corner and then toward the goal at the upper right corner. The agent repeats the moving sequence 1000 times and learns it.

4.2 The algorithm

The model of the action-value function is represented with some parameters as follows:

$$P(q,s,a,v) = P(s|q)P(a|q)P(v|q)P(q) \quad (7)$$
$$P(s|q) = w_{qs}, \quad P(a|q) = w_{qa}, \quad P(V=1|q) = w_{qv}, \quad P(q) = constant$$

Let s^Q, s^S and s^A be the number of states of nodes Q, S and A, respectively. The number of parameters w_{qs}, w_{qa} and w_{qv} are $s^Q s^S$, $s^Q s^A$ and s^Q, respectively. (We do not count $P(V=0|q) = 1 - w_{qv}$ as parameters.)

At each step in each episode, the current state s^* is given. Then, action a^* is determined based on the current action-value function as follows:

$$a_0 = decision(s^*), \quad a^* = noise(a_0) \quad (8)$$

where $noise(a_0)$ is an action actually performed, determined by a desired action a_0 and action noise.

In the case of ideal decision-making, the decision-making function $decision(s)$ is defined as follows:

$$decision^{Ideal}(s) = a^{Ideal} = \operatorname*{argmax}_a P(V = 1|s, a) \tag{9}$$

In the case of MPE decision-making, it is defined as follows:

$$\begin{aligned} decision^{MPE}(s) &= a^{MPE} \\ where\ (q^{MPE}, a^{MPE}) &= \operatorname*{argmax}_{(q,a)} P(q, s, a, V = 1) \end{aligned} \tag{10}$$

After performing the action a^* and obtaining a reward r, the parameters of the action-value function are updated as follows. First, a value q^*, which is *the winner* of a competition at node Q, is determined as follows:

$$q^* = \operatorname*{argmax}_q \sum_v P(q, s^*, a^*, v)A(q) \tag{11}$$

where $A(q)$ is as described below.

Then, the parameters related to q^* are updated as follows:

$$\begin{aligned} w_{q^*s} &\leftarrow w_{q^*s} + \alpha(\delta_{ss^*} - w_{q^*s}) \\ w_{q^*a} &\leftarrow w_{q^*a} + \alpha(\delta_{aa^*} - w_{q^*a}) \\ w_{q^*v} &\leftarrow w_{q^*v} + \alpha(r + w_{q'v} - w_{q^*v}) \end{aligned} \tag{12}$$

where α is the learning rate, q' is q^* at the next step, Kronecker delta $\delta_{xx} = 1$ and $\delta_{xy} = 0\,(x \neq y)$.

The win-rate penalty $A(q)$ is introduced to avoid the undesirable situation in which some values never become the winner. $A(q)$ is defined as follows:

$$A(q) = (r_q)^{-C} \tag{13}$$

where C is a coefficient to control the strength of the effect of the win-rate penalty. The win-rate r_q of a value q is updated as follows:

$$r_q \leftarrow r_q + \alpha(\delta_{qq^*} - r_q) \tag{14}$$

We set the parameters as follows: $s^Q = 64$, $s^S = 4 \times 4, s^A = 8$, $\alpha = 0.001$ and $C = 0.2$.

4.3 Evaluation methods

To evaluate the performance of reinforcement learning quantitatively, we plot graphs of transition of the mean steps elapsed between the start and the goal.

To evaluate the validity of MPE decision-making, we plot the relation between the number of occurrence of a state-action pair (s, a) and the value v_1/v_2, defined as follows:

$$\begin{aligned} v_1 &= P(V = 1|s, a) \\ v_2 &= P(q^*, s, a, V = 1) \quad where \quad q^* = \operatorname*{argmax}_q P(q, s, a, V = 1) \end{aligned} \tag{15}$$

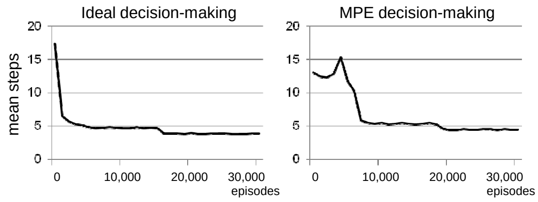

Fig. 3. Examples of the transition of the mean number of steps in the case of ideal decision-making (left) and MPE decision-making (right). The horizontal axis denotes the number of learned episodes.

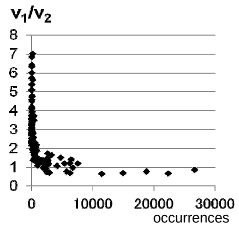

Fig. 4. A scatterplot of v_1/v_2 in the case of MPE decision-making. The horizontal axis denotes the number of occurrences of state-action pairs.

The number of points in the scatterplot is the number of all combinations of s and a, $(4 \times 4) \times 8 = 128$. The values v_1 and v_2 are the scores to determine ideal and MPE decision-making, respectively. Therefore, if the value v_1/v_2 is close to 1, MPE decision-making will be close to ideal decision-making.

4.4 Results

Figure 3 (left) is an example of the transition of the mean number of steps in the case of ideal decision-making. The mean number of steps quickly approach the optimal value, which is about 3 steps. This means that the proposed reinforcement learning algorithm combined with a Bayesian network and competitive learning works correctly. In the case of ideal decision-making, the performance is stable even if the value of s^Q (the number of states of the hidden node Q) and/or C (the strength of the win-rate penalty) are/is changed.

Figure 3 (right) is the mean number of steps in the case of MPE decision-making. The mean number of steps gradually become a small value. This means reinforcement learning using MPE also works correctly.

Figure 4 shows a scatterplot of v_1/v_2 in the case of MPE decision-making, after 30,000 episodes are learned. We see that v_1/v_2 is close to 1 if a corresponding (s, a) occurs a large number of times. This means MPE approximates ideal decision-making in many cases. On the other hand, if (s, a) occurs infrequently, v_1/v_2 is far from 1. We can expect that some treatment for these situations will improve the performance of MPE decision-making in the future.

5 An interpretation of the cortico-basal ganglia loop

Assuming that the cortico-basal ganglia loop executes the algorithm described in this paper, the meaning of the output of each organization can be interpreted as portrayed in Fig.5. We assume that MPE is calculated only by using the circuit within the cerebral cortex[11][14].

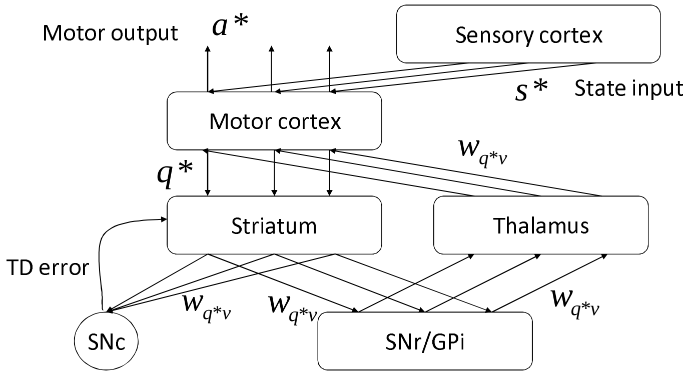

Fig. 5. A hypothetical interpretation of the cortico-basal ganglia loop based on the model presented in this study. Indirect paths in the basal ganglia are omitted, as in a previous work[4]. SNc: Substantia nigra pars compacta, SNr: Substantia nigra pars reticulata, GPi: Internal segment of globus pallidus.

The cerebral cortex sends q^* to the striatum: active columns send 1, and inactive columns send 0. The striatum learns the value w_{q^*v} using q^* sent from the cerebral cortex and TD-error sent from SNc. The striatum sends the current value of w_{q^*v} to the cerebral cortex via SNr/GPi and the thalamus. Active columns in the cerebral cortex copy the value w_{q^*v} sent from the thalamus. The copied value is used for MPE calculation at the next step of reinforcement learning.

6 Conclusion and future work

With theoretical study and computer experiments, we have shown that decision-making using MPE can approximate ideal decision-making. The main contributions of this paper to computational neuroscience are summarized as follows:

1. We proposed a model of motor areas based on a Bayesian network with more detailed formalization than the previous study[10].
2. We showed indirect evidence that MPE calculation is a unified model of both sensory-cortex recognition and motor-cortex decision-making.
3. We provided a new interpretation of the cortico-basal ganglia loop.

In the future, we will extend this model to a multiple-hidden node and multiple-layered model that is more close to the actual brain and that can realize more complex behavior of motor-areas and high-level decision-making of prefrontal areas. We also aim to show the practical usefulness of efficient MPE decision-making in some nontrivial application.

References

1. J. Pearl, Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference, Morgan Kaufmann, 1988.
2. Richard S.Sutton and Andrew G.Barto, Reinforcement Learning: An Introduction, The MIT Press, 1998.
3. G. E. Alexander et al., PARALLEL ORGANIZATION OF FUNCTIONALLY SEGREGATED CIRCUITS LINKING BASAL GANGLIA AND CORTEX, Annual Review of Neuroscience 9: 357-381 1986.
4. K. Doya, Complementary roles of basal ganglia and cerebellum in learning and motor control, Current Opinion in Neurobiology 10 (6): 732-739 Dec 2000.
5. Lee, T.S., Mumford, D. , Hierarchical Bayesian inference in the visual cortex. Journal of Optical Society of America, A. 20(7): 1434-1448, 2003.
6. R. Rao, Bayesian inference and attention in the visual cortex. Neuroreport 16(16), 1843-1848, 2005.
7. 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.
8. 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 (IJCNN2007), pp.1065–1070, Aug 2007.
9. Florian Roehrbein, Julian Eggert and Edgar Koerner, Bayesian Columnar Networks for Grounded Cognitive Systems, In Proc. of the 30th Annual Conference of the Cognitive Science Society, pp.1423–1428, 2008.
10. Haruo Hosoya, A motor learning neural model based on Bayesian network and reinforcement learning. In Proceedings of International Joint Conference on Neural Networks, pages 1251-1258, 2009.
11. Shai Litvak, Shimon Ullman, Cortical Circuitry Implementing Graphical Models, Neural Computation 21, 3010.3056, 2009.
12. Chikkerur, S., T. Serre, C. Tan and T. Poggio, What and where: A Bayesian inference theory of attention. Vision Research. 55(22), pp. 2233–2247, 2010.
13. Yuuji Ichisugi, Haruo Hosoya, Computational Model of the Cerebral Cortex that Performs Sparse Coding Using a Bayesian Network and Self-Organizing Maps, In Proc. of 17th International Conference on Neural Information Processing (ICONIP 2010), Part I, LNCS 6443, pp.33–40, Nov 2010.
14. Yuuji Ichisugi, Recognition Model of Cerebral Cortex based on Approximate Belief Revision Algorithm, In Proc. of The 2011 International Joint Conference on Neural Networks (IJCNN 2011), pp.386–391, 2011.
15. Hiroaki Hasegawa, Masafumi Hagiwara, Visual Shape Recognition Neural Network Using BESOM Model. In Proc. of Artificial Neural Networks (ICANN 2010), Lecture Notes in Computer Science Volume 6354, pp.102–105, 2010.
16. Haruo Hosoya, Bayesian Interpretation of Border-Ownership Signals in Early Visual Cortex, In Proc. of International Conference on Neural Information (ICONIP 2010), Part I, LNCS 6443, pp.1–8, 2010.
17. Timm Lochmann, Sophie Deneve, Neural processing as causal inference, Current Opinion in Neurobiology, Volume 21, Issue 5, pp. 774–781, 2011.
18. Haruo Hosoya, Multinomial Bayesian Learning for Modeling Classical and Nonclassical Receptive Field Properties, Neural Computation,Vol. 24, No. 8, pp. 2119–2150, 2012.