THE OTHER VARIANT BOLTZMANN MACHINE

Cheng-Yuan Liou
Associate Professor, IEEE & INNS member
Department of Computer Science and Information Engineering
National Taiwan University

Shiao-Lin Lin
MD, Department of Neurology
National Taiwan University Hospital

Correspondence Address:
Department of Computer Science and Information Engineering
National Taiwan University, Taipei
Taiwan, 10764, Republic of China
Telephone: 8862-3622444
FAX: 8862-3637204

Key phrases: Neural Network, Boltzmann Machine, Cross-Entropy, Self-Learning, Asymmetric Divergence

ABSTRACT

In this work we present a new learning theory to study the learning behavior of a neural network which is formed by interconnecting neurons. This learning theory is a new approach for the Boltzmann machine. The central idea presented in this work based on minimizing one of the two cross-entropy-like criteria. The twin criteria are the cross-entropy and the reversed cross-entropy, the latter is used in deriving the Boltzmann machine by Ackley et al. The results derived by our new approach are closely related to those by Ackley et al. with several significant modifications in the algorithm. A detailed discussions of the new algorithm is presented. It is shown that the new algorithm is a probability weighted version of the algorithm by Ackley et al.

I. Introduction

There are a large number of applications in which it is necessary to forming the internal representation of an environmental data set giving random samples of the set. The simplest approach is to induce some linear relations from the samples. Unfortunately, forming the linear relations cannot be applied to most cases. Severe nonlinear relations or high-order constraints exist in most data sets. There exist several promising methods in the area of neural networks to solve this problem such as the backpropagation models [5] and self-organization feature maps [3].

Ackley et al. [1] devised the Boltzmann machine. They build a Hopfield network and the network operates with the principle of thermal dynamics. The network can from the internal representation of an environmental data set by using the random samples from the set. They obtain an algorithm by minimizing a 'reversed cross-entropy' (RCE) [4] objective function to achieve the internal representations of the environmental data set. We will call their algorithm the 'Reversed Cross-Entropy' (RCE) algorithm. This object function has a powerful twin, the 'cross-entropy (CE)' [6]. These two object functions between two probability functions $\{P'(V_\alpha)$ and $P(V_\alpha)\}$ are given as follows

$$\text{RCE}: \ G_{RCE}(P, P') = \sum_\alpha P(V_\alpha) \ln \frac{P(V_\alpha)}{P'(V_\alpha)} ,$$

and

$$\text{CE}: \ G_{CE}(P', P) = \sum_\alpha P'(V_\alpha) \ln \frac{P'(V_\alpha)}{P(V_\alpha)} .$$

The roles of the two probability functions are reversed in the above formulas. Both CE and RCE are used as the principles for probabilistic inferences.

In the next section we will use the CE as the objective function and derive a new algorithm. We will call this new algorithm the CE algorithm. The CE algorithm is then compared with the RCE algorithm. The same notations and procedures as in [1] are adopted to facilitate the reading. A detailed algorithm of the new approach and some discussions are presented in the last section.

II. Derivation of the Cross-Entropy Algorithm

In this section we will briefly review the derivations by Ackley et al. and derive the CE algorithm. The same notations and the model of the network as in [1] are also adopted in this work.

II. 1. Review the Network Model

The configuration of the network is that it has an arbitrary number N of neurons and weights $\{\omega_{ij} = \omega_{ji}, \omega_{ii} = 0; 1 \leqslant i, j \leqslant N\}$ where ω_{ij} is the strength of the interconnection between the ith neuron and the jth neuron. All neurons are bistate neurons with state $\delta_i = 0$ or $\delta_i = 1$. The transition rule for the state changes of each neuron is that the state δ_k of the kth neuron, where $1 \leqslant k \leqslant N$, is set to $\delta_k = 1$ with the probability

$$p_k = \frac{1}{(1 + e^{-\Delta E_k/T})}, \quad \text{with } \Delta E_k = \sum_i \omega_{ki}\delta_i$$

where ΔE_k is the incoming excitation energy and T is the pseudo-temperature. The transitions of states of the neurons is asynchronous.

Defining the total energy E_g of the network as,

$$E_g = -\sum_{i<j} \omega_{ij}\delta_i^g \delta_j^g$$

where δ_i^g denotes the ith neuron in a global states g of the network. When the network reach "thermal equilibrium" the relative probability of two global states g1 and g2 of the network will follow the Boltzmann distribution:

$$\frac{P'_{g1}}{P'_{g2}} = e^{-(E_{g1} - E_{g2})/T}.$$

The neurons are further divided into two parts, one is called visible neurons and the other is called hidden neurons. Accordingly, we divide the neuron's index set into two subsets, $\{1, 2, \ldots, N\} = \{\alpha \mid \alpha \in \text{visible neurons}, 1 \leqslant \alpha \leqslant N_v\} \cup \{\beta \mid \beta \in \text{hidden neurons}, N_v + 1 \leqslant \beta \leqslant N\}$, where $N - N_v = N_h$, N_v is the total number of visible neurons, N_h is the total number of hidden neurons. To facilitate the analyses, let V_α denotes the vector states of the all N_v visible neurons, H_β denotes the vector states of the all N_h hidden neurons, and $V_\alpha \wedge H_\beta$ denotes the global states of the all N neurons in the network. The main idea of the Boltzmann machine is to devise an algorithm which takes full advantage of the mechanism of the network. In their work the internal representation of the network is achieved by minimizing the distance between the network's own probability function $P'(V_\alpha)$ over the visible neurons and the environmental probability function $P(V_\alpha)$ over the visible neurons. The distance employed by Ackley et al. is the RCE. In the following subsection we will review their derivations briefly to facilitate some useful derivations.

II. 2. Review the RCE Approach

The distance they employed is the RCE,

$$G_{RCE} = \sum_\alpha P(V_\alpha) \ln \frac{P(V_\alpha)}{P'(V_\alpha)} \tag{1}$$

where $P(V_\alpha)$ is the probability of the vector state of the visible neurons when their states are determined by the environment, and $P'(V_\alpha)$ is the corresponding probability when the network is running freely with no environmental input. We want to adjust the ω_{ij} to reduce the distance G_{RCE}. To achieve the minimum distance we need to follow the negative direction of the gradient $\frac{\partial G_{RCE}}{\partial \omega_{ij}}$ and find a practical method to obtain an estimated gradient $\widehat{\frac{\partial G_{RCE}}{\partial \omega_{ij}}}$. We now briefly review the derivations of $\frac{\partial G_{RCE}}{\partial \omega_{ij}}$ in order to facilitate some useful derivations. According to Boltzmann distribution:

$$P'(V_\alpha) = \sum_\beta P'(V_\alpha \wedge H_\beta) = \frac{\Sigma_\beta e^{-E_{\alpha\beta}/T}}{\Sigma_{\alpha\beta} e^{-E_{\alpha\beta}/T}};$$

$$E_{\alpha\beta} = -\sum_{i<j} \omega_{ij}\delta_i^{\alpha\beta}\delta_j^{\alpha\beta} \tag{2}$$

where $E_{\alpha\beta}$ is the pseudo-energy of the system in a global state $V_\alpha \wedge H_\beta$. The following two formulas given in [1] are useful for obtaining the gradient of the distance (1).

$$\frac{\partial e^{-E_{\alpha\beta}/T}}{\partial \omega_{ij}} = \frac{1}{T} \delta_i^{\alpha\beta} \delta_j^{\alpha\beta} e^{-E_{\alpha\beta}/T}$$

Differentiating (2) yields

$$\frac{\partial P'(V_\alpha)}{\partial \omega_{ij}} = \frac{\frac{1}{T} \Sigma_\beta e^{-E_{\alpha\beta}/T} \delta_i^{\alpha\beta} \delta_j^{\alpha\beta}}{\Sigma_{\alpha\beta} e^{-E_{\alpha\beta}/T}} - \frac{\Sigma_\beta e^{-E_{\alpha\beta}/T} \frac{1}{T} \Sigma_{\lambda\mu} e^{-E_{\lambda\mu}/T} \delta_i^{\lambda\mu} \delta_j^{\lambda\mu}}{(\Sigma_{\lambda\mu} e^{-E_{\lambda\mu}/T})^2}$$

$$= \frac{1}{T} [\sum_\beta P'(V_\alpha \wedge H_\beta) \delta_i^{\alpha\beta} \delta_j^{\alpha\beta} - P'(V_\alpha) \sum_{\lambda\mu} P'(V_\lambda \wedge H_\mu) \delta_i^{\lambda\mu} \delta_j^{\lambda\mu}] . \quad (3)$$

By using the above two formulas we get the gradient of (1)

$$\frac{\partial G_{RCE}}{\partial \omega_{ij}} = -\sum_\alpha \frac{P(V_\alpha)}{P'(V_\alpha)} \frac{\partial P'(V_\alpha)}{\partial \omega_{ij}}$$

$$= -\frac{1}{T} \sum_\alpha \frac{P(V_\alpha)}{P'(V_\alpha)} [\sum_\beta P'(V_\alpha \wedge H_\beta) \delta_i^{\alpha\beta} \delta_j^{\alpha\beta} - P'(V_\alpha) \sum_{\lambda\mu} P'(V_\lambda \wedge H_\mu) \delta_i^{\lambda\mu} \delta_j^{\lambda\mu}] . \quad (4)$$

From the definition of conditional probability

$$P(V_\alpha \wedge H_\beta) = P(H_\beta/V_\alpha) P(V_\alpha),$$
$$P'(V_\alpha \wedge H_\beta) = P'(H_\beta/V_\alpha) P'(V_\alpha) \quad (5)$$

and the fact

$$P'(H_\beta/V_\alpha) = P(H_\beta/V_\alpha) \quad (6)$$

We get the equality

$$P'(V_\alpha \wedge H_\beta) \frac{P(V_\alpha)}{P'(V_\alpha)} = P(V_\alpha \wedge H_\beta) .$$

Substituting the above equality in (4) and using the fact $\Sigma_\alpha P(V_\alpha) = 1$, we get the following gradient equation

$$\frac{\partial G_{RCE}}{\partial \omega_{ij}} = -\frac{1}{T} [p_{ij} - p'_{ij}] \quad (7)$$

where $p_{ij} = \Sigma_{\alpha\beta} P(V_\alpha \wedge H_\beta) \delta_i^{\alpha\beta} \delta_j^{\alpha\beta}$ and $p'_{ij} = \Sigma_{\lambda\mu} P'(V_\lambda \wedge H_\mu) \delta_i^{\lambda\mu} \delta_j^{\lambda\mu}$. The p_{ij} is the averge probability of two units both being the on state when the environment is clamping the states of the visible neurons, and p'_{ij} is the corresponding probability when the environmental input is not present and the network is running freely.

To minimize G_{RCE}, it is therefore sufficient to observe (estimated) p_{ij} and p'_{ij} during the network is in thermal equilibrium, and to change each weight by an amount proportional to the difference between this two quantities:

$$\Delta \omega_{ij} = \epsilon (p_{ij} - p'_{ij}) . \quad (8)$$

Note that the relative-frequencies [2], $\hat{p}_{ij}$ and $\hat{p}'_{ij}$, of the event $\{ \delta_i = 1 \text{ and } \delta_j = 1 \}$ at equilibrium is good estimators of p_{ij} and p'_{ij}.

II. 3. CE Approach

The only difference between the CE approach and the RCE approach is that we employ CE for the definition of distance. By definition, CE is

$$G_{CE} = \sum_\alpha P'(V_\alpha) \ln \frac{P'(V_\alpha)}{P(V_\alpha)} . \quad (9)$$

We now derive the gradient $\frac{\partial G_{CE}}{\partial \omega_{ij}}$ as follows.

$$\frac{\partial G_{CE}}{\partial \omega_{ij}} = \sum_\alpha \frac{\partial P'(V_\alpha)}{\partial \omega_{ij}} \ln \frac{P'(V_\alpha)}{P(V_\alpha)} + \sum_\alpha P'(V_\alpha) \frac{P(V_\alpha)}{P'(V_\alpha)} \frac{1}{P(V_\alpha)} \frac{\partial P'(V_\alpha)}{\partial \omega_{ij}}$$

$$= \sum_\alpha \frac{\partial P'(V_\alpha)}{\partial \omega_{ij}} \ln \frac{P'(V_\alpha)}{P(V_\alpha)} + \sum_\alpha \frac{\partial P'(V_\alpha)}{\partial \omega_{ij}}$$

: obtained by simplifying the above formula

$$= \sum_\alpha [\ln \frac{P'(V_\alpha)}{P(V_\alpha)}] \frac{1}{T} [\sum_\beta P'(V_\alpha \wedge H_\beta) \delta_i^{\alpha\beta} \delta_j^{\alpha\beta} - P'(V_\alpha) p'_{ij}]$$

$$+ \sum_\alpha \frac{1}{T} [\sum_\beta P'(V_\alpha \wedge H_\beta) \delta_i^{\alpha\beta} \delta_j^{\alpha\beta} - P'(V_\alpha) p'_{ij}]$$

: obtained by substituting (3) in

$$= \sum_\alpha [\ln \frac{P'(V_\alpha)}{P(V_\alpha)}] \frac{1}{T} [\sum_\beta P'(V_\alpha \wedge H_\beta) \delta_i^{\alpha\beta} \delta_j^{\alpha\beta} - P'(V_\alpha) p'_{ij}]$$

: the second term in the above formula equals to zero

$$= \sum_\alpha [\, P'(V_\alpha) \ln \frac{P'(V_\alpha)}{P(V_\alpha)} \,] \frac{1}{T} [\, \sum_\beta \frac{P'(V_\alpha \wedge H_\beta)}{P'(V_\alpha)} \delta_i^{\alpha\beta} \delta_j^{\alpha\beta} - p'_{ij}]$$

: factorizing $P'(V_\alpha)$

$$= \sum_\alpha [\, P'(V_\alpha) \ln \frac{P'(V_\alpha)}{P(V_\alpha)} \,] \frac{1}{T} [\, \sum_\beta P'(H_\beta \mid V_\alpha) \delta_i^{\alpha\beta} \delta_j^{\alpha\beta} - p'_{ij}]$$

: using definition of conditional probability function

$$= \sum_\alpha [\, P'(V_\alpha) \ln \frac{P'(V_\alpha)}{P(V_\alpha)} \,] \frac{1}{T} [\, \sum_\beta P(H_\beta \mid V_\alpha) \delta_i^{\alpha\beta} \delta_j^{\alpha\beta} - p'_{ij}]$$

: using (6) $P'(H_\beta \mid V_\alpha) = P(H_\beta \mid V_\alpha)$

So,

$$\frac{\partial G_{CE}}{\partial \omega_{ij}} = \frac{1}{T} \sum_\alpha [\, P'(V_\alpha) \ln \frac{P'(V_\alpha)}{P(V_\alpha)} \,] \; [\, p_{ij/V_\alpha}(\mid V_\alpha) - p'_{ij}] \quad (10)$$

where

$$p_{ij/V_\alpha}(\mid V_\alpha) = \sum_\beta P(H_\beta \mid V_\alpha) \delta_i^{\alpha\beta} \delta_j^{\alpha\beta}$$

is the average conditional probability of two neurons both being in the on state when an environmental state vector V_α is clamping the states of the visible neurons. Let

$$W(V_\alpha) = P'(V_\alpha) \ln \frac{P'(V_\alpha)}{P(V_\alpha)}$$

The gradient formula can be further simplified as

$$\frac{\partial G_{CE}}{\partial \omega_{ij}} = \sum_\alpha W(V_\alpha) \frac{1}{T} [\, p_{ij/V_\alpha}(\mid V_\alpha) - p'_{ij}] \,.$$

To minimize G_{CE}, it is therefore sufficient to observe $p_{ij}(\mid V_\alpha)$, p'_{ij}, and $W(V_\alpha)$ when the network is in thermal equilibrium, and to change each weight by an amount proportional to the weighted difference between these two probabilities:

$$\Delta\omega_{ij} = -\,\epsilon \sum_\alpha W(V_\alpha) [\, p_{ij/V_\alpha}(\mid V_\alpha) - p'_{ij}] \,. \quad (11)$$

To compare the two gradients we revise the formula $\frac{\partial G_{RCE}}{\partial \omega_{ij}}$ by substituting the identity (5, 6) in (4) and get

$$\frac{\partial G_{RCE}}{\partial \omega_{ij}} = \frac{1}{T} \sum_\alpha \{ -P(V_\alpha) \} \; [\, p_{ij/V_\alpha}(\mid V_\alpha) - p'_{ij}] \,. \quad (12)$$

The two formulas, (10) and (12), show that the CE algorithm is a fine cross-entropy weighted version of the RCE algorithm. When $W(V_\alpha) = -P(V_\alpha)$ the CE algorithm is identical to the RCE algorithm.

Comparing the equation (10) and (12) we see that the CE algorithm matches the detailed shape of the two probability functions. Since p_{ij} and p'_{ij} are also the second order statistics (cross-covariance functions), the RCE algorithm matches only the two probability's second order statistics but not the detailed shape.

We will further discuss the detailed CE algorithm in the following section.

III. Comparison and Discussion

In comparing the two algorithms first we note that the two updating formulas (7) and (10). The formula (7) is biophysically meaningful and is still within the scope of the assumptions proposed by Hebb [3]. The formula (10) has less physical meaning. It is hard to devise a plausible biophysical mechanism to implement the formula (10), even through the (10) is proved to be a powerful algorithm in learning.

People might wonder that if we can switch the meanings of the two probabilities in formulas (1) and (9) and say that (1) is the CE and (9) is the RCE. The answer is not easy. This is because their are four invariance and consistency axioms behind the cross-entropy (9). See [6] for details.

We now present and discuss the detailed CE algorithm for the variant Boltzmann machine. The following algorithm for each learning cycle is a modified version of those described in [1].

CE Algorithm for each Learning Cycle: Let n denotes the nth learning cycle.

Step 1 Estimation of $p^n_{ij/V_\alpha}(\mid V_\alpha)$: Each environment vector in turn is clamped over the visible neurons. For each environment vector, the network is allowed to reach equilibrium twice. Statistic about how often pairs of neurons are both on together are collected at equilibrium for <u>every different</u> environmental clamped vector V_α. The same noisy clamping technique as in [1] is also applied to prevent the weights from growing too large.

Step 2 Estimation of p'^{n}_{ij}: The network is completely unclamped and allowed to reach equilibrium at low temperature (by annealing). Statistics about how often pairs of neurons are both on together are then collected for as many annealings as are used to estimate $p_{ij/V_\alpha}(|V_\alpha)$.

Step 3 Estimation of $P^n(V_\alpha)$: The statistics about how often the occurrences (relative-frequency) for every different environmental vector, V_α, are collected as an estimation of the $P(V_\alpha)$.

Step 4 Estimation of $P'^n(V_\alpha)$: The statistics about how often the occurrences for every different state V_α of the visible neurons are collected at equilibrium for as many annealings as are used to estimate p'^{n}_{ij}.

Step 5 Updating the weight $\Delta\omega^n_{ij}$: The estimated values $\{ p^n_{ij}(|V_\alpha)$, p'^n_{ij}, $P^n(V_\alpha)$, $P'^n(V_\alpha) \}$ are then substituted in (11) to obtain the weight changes $\Delta\omega^n_{ij}$.

$$\Delta\omega^n_{ij} = -\epsilon \sum_\alpha W^n(V_\alpha) \; [\, p^n_{ij/V_\alpha}(|V_\alpha) - p'^n_{ij} \,] \, . \qquad (13)$$

Renew the weights $\omega^{n+1}_{ij} = \omega^n_{ij} + \Delta\omega^n_{ij}$. Check if the network is converge yet using the formula max $\{ |\Delta\omega^n_{ij}|, 1 \leqslant i < j \leqslant N \} < \delta$, where δ is a predetermined threshold. If it is not converge, then renew the weights $\omega^{n+1}_{ij} = \omega^n_{ij} + \Delta\omega^n_{ij}$, start the next cycle, and repeat the five steps.

Note the same annealing technique as in [1] can be used in this CE algorithm.

We now discuss the above algorithm. Since in most cases the number of visible neurons N_v is large and the total visible states of the network is 2^{N_v}. All the 2^{N_v} states have none-zero probability at any none-zero temperature. And in many cases the total number N_e of the environmental state vectors is much smaller than 2^{N_v}. So, the probability $P(V_\alpha)$ is zero for many system's states V_α. In both RCE and CE distances this zero probability, $P(V_\alpha) = 0$, will cause the distance infinitive large. And the updated weights using the estimated gradients tend to mainly match the two probabilities $P'(V_\alpha)$ and $P(V_\alpha)$ at those states V_α which have zero or small probability $P(V_\alpha) = 0$. This is because this kind matching will of course reduces a large distance. This phenomenon also affects some of the weights grows very large { if the weight $-\infty < \omega_{ij} < \infty$ can have infinitive range } to prevent those network's states from happening { or make those states have very high energy }. Note that this point is according to Boltzmann distribution, a high energy state has small probability. In order to remedy this problem Ackley at el. suggest using noisy environmental state vectors to smooth the destructive zero-probability of some environmental states $P(V_\alpha) = 0$.

For this problem we devise a strategy to keep the weight ω_{ij} from growing too large. Since infinitive range of the weight $-\infty < \omega_{ij} < \infty$ causes the network system has a very wide range of linear flexibility. This wide flexibility survives those system's states V_α with non-zero probabilities when the same environmental states V_α make no appearance, that is $P(V_\alpha) = 0$ for some of the network's own states V_α. For this problem, the strategy is to limit the infinitive range of the linear weights by using weights with limits and exhaust the flexibility of the network system as much as possible by the environmental states with non-zero probability, that is by the states with $P(V_\alpha) \neq 0$.

We describe the method as in the following context. We obtain a new weight ω_{ij} by applying a sigmod function to the updated weight, $\omega^{\dagger n}_{ij} = \omega^n_{ij} + \Delta\omega^n_{ij}$, and obtaining the new weight $\omega^n_{ij} = \sigma[\, \omega^{\dagger n}_{ij} \,]$, where $\sigma[x]$ is a sigmoid function and n denotes the number of the learning cycle. The simplest sigmoid function which may be used is the linear threshhold logic function,

$$\sigma[x] = \begin{cases} x, & \text{if } |x| \leqslant c_1 ; c_1 > 0 ; \\ c_1 & \text{if } x > c_1 ; \\ -c_1 & \text{if } x < c_1 ; \end{cases}$$

where c_1 is a fixed constant. With properly adjusted c_1 the flexibility of the whole network could be exhausted by the finite number 'N_e' environmental states. This strategy indirectly keeps the probabilities of those states V_α, which has a $P(V_\alpha) = 0$, as close to zero as possible.

According to the above strategy the Step 5 should be replaced by the following Step 5*.

Step 5* Updating the weight $\Delta\omega_{ij}^{n}$: The estimated values $\{ p_{ij}^{n}(\mid V_\alpha), p'^{n}_{ij}, P^{n}(V_\alpha), P'^{n}(V_\alpha) \}$ are then substituted in the following three formulas to obtain the weight changes $\Delta\omega_{ij}^{n}$,

$$\begin{cases} \Delta\omega_{ij}^{n} = -\epsilon \, \Sigma_{\alpha\in\Psi} \, W^{n}(V_\alpha) \, [\, p_{ij/V_\alpha}^{n}(\mid V_\alpha) - p'^{n}_{ij} \,] \,. \\ \omega_{ij}^{\dagger n} = \omega_{ij}^{n} + \Delta\omega_{ij}^{n} \\ \omega_{ij}^{n} = \sigma\,[\,\omega_{ij}^{\dagger n}\,] \end{cases} \qquad (14)$$

where

$$\Psi = \{ P^{n}(V_\alpha) \neq 0 \text{ and } P'^{n}(V_\alpha) \neq 0 \}$$

is the summation domain in which all V_α with $P^{n}(V_\alpha) > 0$ and $P'^{n}(V_\alpha) > 0$. The following sigmoid function is used.

$$\sigma\,[x] = \begin{cases} x, \text{ if } |x| \leqslant c_1;\ c_1 > 0; \\ c_1 \ \text{ if } x > c_1; \\ -c \text{ if } x < c_1; \end{cases}$$

Check if the network is converge yet; max $\{ |\Delta\omega_{ij}^{n}|,\ 1 \leqslant i < j < N \} < \delta$, where δ is a predetermined threshold. If it is not converge, then renew the weights $\omega_{ij}^{n+1} = \omega_{ij}^{n} + \Delta\omega_{ij}^{n}$, start the next cycle, and repeat the five steps. Note that the environmental zero-probability states V_α should not be included in the summation in the updating equation (14).

Further comparing the two updating formulas, (8) and (11), we see that the updating formula (11) is contributed by the components of different states V_α. In order to perfectly match the states which has high environmental probabilities, it can be achieved purposefully by summing those significant components of states V_α which has high environmental probabilities. So, the summation domain can be further modified as

$$\Psi = \{ P^{n}(V_\alpha) > \eta_1^{n} \text{ and } P'^{n}(V_\alpha) > \eta_2^{n} \}$$

where η_1^{n} and η_2^{n} are the selected positive constants for the nth cycle. The RCE algorithm can have this kind ability only through the revised formula (12).

References

[1] David H. Ackley, Geoffrey E. Hinton, and Terrence J. Sejnowski, "A Learning Algorithm for Boltzmann Machines," *Cognitive Science*, vol. 9, pp. 147-169, 1985.

[2] Alvin W. Drake, Fundamentals of Applied Probability Theory. McGraw-Hill, Inc. 1967.

[3] Teuvo Kohonen, Self-Organization and Associative Memory. Second Edition, Berlin Heidelberg: Springer-Verlag, 1988.

[4] Solomon Kullback, Information Theory and Statistics. New York: John Wiley & Sons, 1959.

[5] Terrence J. Sejnowski and Charles R. Rosenberg, "NET talk: a parallel network that learns to read aloud," The Johns Hopkins University Electrical Engineering and Computer Science Technical Report JHU/EECS-86/01, 32pp.

[6] John E. Shore and Rodney W. Johnson, "Axiomatic Derivation of the principle of Maximum Entropy and the Principle of Minimum Cross-Entropy", IEEE Trans. Info. Theory, vol. IT-26, no. 1, pp. 26-37, January 1980.