Construct The Structure of Stochastic Multilayer Perceptron Using The Model Search Method

Shiro IKEDA

The Institute of Physical and Chemical Research (RIKEN)

Abstract

The author gives an algorithm to search the structure of a stochastic models with hidden variable. The author have shown the algorithm to find the hidden structure of the Hidden Markov Model and in this article, the algorithm is applied for one of the other stochastic models which have hidden probabilistic variables.

I. Introduction

The Neural Networks have great ability to express various types of functions. The Neural Network has this ability because it has a lot of parameters and also because it can have many kinds of structures which is determined by the number of the cells and the connections between them. Conversely speaking, if we want to use the Neural Network as a powerful tool, it is important to set these parameters and also determine the structure correctly.

To set the values of the parameters, we have some algorithms. Though, we do not have any successful algorithm to choose the structures of the Networks except trial and error.

The problem to determine the structures of the model is equivalent to so-called "model selection" in statistics. In model selection, we usually prepare some candidates, and use a measure to define how "good" each model is by some information criteria. AIC (Akaike's information criterion) is one of such measures. AIC can be used for the models whose parameters are estimated by the maximum likelihood method.

It seems easy to carry out the model selection for selecting the structure of the Neural Networks by model selection. Though there is a practical problem. To have a good model, we have to prepare a lot of candidates with different structures. If we try to estimate parameters of many possible candidates, it takes enormous time. Therefore, we have to prepare models of different structures in an effective manner. The author proposes an algorithm in which the model of simple structure is modified successively to more complicated ones by adding some new parameters to the model. Thus, the machine "searches" the optimal structure of the model.

In this article, the author gives the outline of the algorithm and simple experiment which is applied for probabilistic Neural Networks, stochastic multilayer Perceptron which was proposed by Amari[1][2]. In the stochastic multilayer Perceptron, each cell behaves in stochastic manner and the cells in hidden layer are the hidden probabilistic variables. For these probabilistic model, we can use EM (Expectation Maximization) algorithm[3] to estimate the parameters. And in this article, the EM algorithm acts the important role for searching the model structure. The outline of the algorithm is as follows.

1. Give the simple structure.
2. Estimate parameters using EM algorithm.
3. Give more connections which would increase its likelihood function best.
4. Measure the model with AIC. If further operations would be useless, exit, otherwise go to 2.

In the algorithm, the number of the cells is fixed, and those cells are not fully connected each other. By increasing the number of the connections between the cells, this algorithm can generate an Neural Network which works better.

In the algorithm, the key point is how to select the parameter to add the network. If it is chosen at random, this algorithm is just a random search. The parameter should be the one which makes the model better. By using EM algorithm, we can select the parameter which is closest to the gradient decent. This means that by EM algorithm, we can select the parameter which will be most effective to add. The author shows this result through information geometrical analysis [2].

II. Stochastic Multilayer Perceptron

Stochastic Multilayer Perceptron (Fig. 1) was introduced by Amari[1][2]. In this article, we consider only the network with one hidden-layer Perceptron with a single output unit. Let $\boldsymbol{x}= (x_i)$, $i = 1, \cdots, n$ is input vector and $\boldsymbol{z}= (z_j)$, $j = 1, \cdots, m$ be the outputs of m hidden units. Each x_i z_i takes binary values 0 and 1. When $\boldsymbol{x}$ is the input, the probability of $z_i = 0, 1$ is,

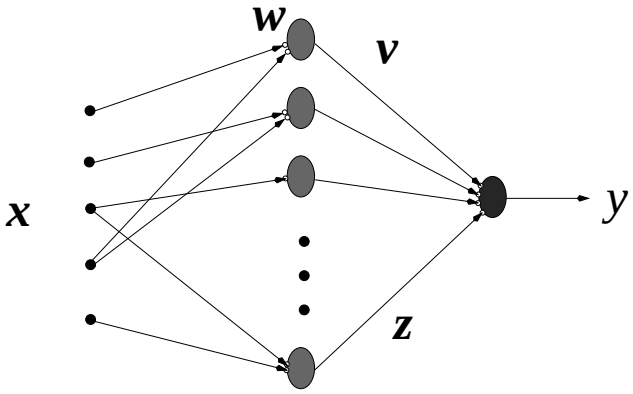

Figure 1. Stochastic Multilayer Perceptron

$$p(z_i|\boldsymbol{x}) = \frac{\exp(z_i \boldsymbol{w}_i \cdot \boldsymbol{x})}{1 + \exp(\boldsymbol{w}_i \cdot \boldsymbol{x})}. \tag{1}$$

The output y is also take 0 and 1 with,

$$p(y|\boldsymbol{z}, \boldsymbol{x}) = \frac{\exp(y\boldsymbol{v} \cdot \boldsymbol{z})}{1 + \exp(\boldsymbol{v} \cdot \boldsymbol{z})}. \tag{2}$$

From (1) and (2),we can have,

$$\begin{aligned} p(y, \boldsymbol{z}|\boldsymbol{x}) &= p(y, \boldsymbol{z}|\boldsymbol{x}) \prod_i p(z_i|\boldsymbol{x}) \\ &= \frac{\exp(y\boldsymbol{v} \cdot \boldsymbol{z})}{1 + \exp(\boldsymbol{v} \cdot \boldsymbol{z})} \prod_i \frac{\exp(z_i \boldsymbol{w}_i \cdot \boldsymbol{x})}{1 + \exp(\boldsymbol{w}_i \cdot \boldsymbol{x})}, \end{aligned} \tag{3}$$

and

$$p(y|\boldsymbol{x}) = \sum_{\boldsymbol{z}} p(y, \boldsymbol{z}|\boldsymbol{x}). \tag{4}$$

You can see from (3), that $p(y, \boldsymbol{z}|\boldsymbol{x})$ is an exponential family which is easily understood from,

$$\begin{aligned} l(y, \boldsymbol{z}|\boldsymbol{x}) &= \log p(y, \boldsymbol{z}|\boldsymbol{x}) \\ &= y\boldsymbol{v} \cdot \boldsymbol{z} - \log(1 + \exp(\boldsymbol{v} \cdot \boldsymbol{z})) \\ &\quad + \sum_i \left(z_i \boldsymbol{w}_i \cdot \boldsymbol{x} - (1 + \exp(\boldsymbol{w}_i \cdot \boldsymbol{x}))\right). \end{aligned} \tag{5}$$

If we only can have some data for training which is a set of $(y_i, \boldsymbol{x}_i)$, we cannot see the probability of $\boldsymbol{z}$. This means $\boldsymbol{z}$ are hidden probabilistic variables. In this article, the author suppose that the true distribution is known. And we have to estimate the parameter and the structure of the stochastic multilayer Perceptron. For parameter estimation, EM algorithm is a good algorithm. In the next section, the author gives the outline of the EM algorithm and the model search algorithm.

III. EM algorithm and Model search algorithm

A. EM algorithm

Now, the probability of the teacher, $q(y, \boldsymbol{x})$ is available. And we have to evaluate the parameter of the model. Or in general cases, $q(y, \boldsymbol{x})$ is not available but we have some samples for estimating the parameter. Then we can have the empirical distribution $\hat{q}(y, \boldsymbol{x})$. In both cases, it is difficult to estimate the parameters by Maximum Likelihood Estimate method, direct expression of MLE is to solve

$$\hat{\theta} = \operatorname*{argmax}_{\theta} \sum_i \log p(y^i|\theta). \tag{6}$$

It is difficult to have the answer because the model has some hidden probabilistic variables z.

The EM algorithm [3] can be used for this situation. The EM algorithm consists of two steps, E-step (expectation) and M-step (maximization). By iterating these two steps, θ is modified successively and we can obtain the MLE as the converged point[3]. The geometrical understanding of EM algorithm was given by Amari[2](Fig. 2).

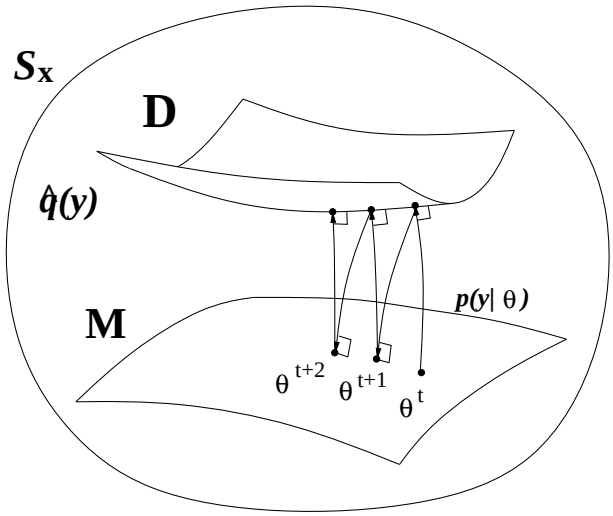

Figure 2. Geometrical view of the EM algorithm

- E-step Calculate $E_{\theta^t}[l(y,\boldsymbol{z}|\boldsymbol{x},\theta)]_{q(x,y)}$

$$Q(\theta,\theta^t) = \int l(y,\boldsymbol{z}|\boldsymbol{x},\theta)p(\boldsymbol{z}|y,\boldsymbol{x},\theta^t)q(\boldsymbol{x},y)d\mu_y d\mu_z d\mu_x \tag{7}$$

- M-step Calculate the θ which maximize $Q(\theta,\theta^t)$, and let the parameters θ^{t+1}.

$$\frac{\partial Q(\theta,\theta^t)}{\partial\theta}\Big|_{\theta^{t+1}} = 0 \tag{8}$$

Because $p(y,\boldsymbol{z}|\boldsymbol{x},\theta)$ is an exponential family. If θ^{t+1} is close to θ^t,

$$\frac{\partial Q(\theta,\theta^t)}{\partial\theta}\Big|_{\theta^t} + \frac{\partial^2 Q(\theta,\theta^t)}{\partial\theta^2}\Big|_{\theta^t}(\theta^{t+1}-\theta^t) \simeq 0. \tag{9}$$

It can be easily shown that,

$$\frac{\partial}{\partial\theta}\int q(\boldsymbol{x},y)l(y|\boldsymbol{x},\theta^t)d\mu_y d\mu_z d\mu_x = \frac{\partial Q(\theta,\theta^t)}{\partial\theta}\Big|_{\theta^t}.$$

$$\frac{\partial}{\partial\theta}KL(q(y,\boldsymbol{x}),q(\boldsymbol{x})p(y|\boldsymbol{x},\theta)) = \frac{\partial Q(\theta,\theta^t)}{\partial\theta}\Big|_{\theta^t}. \tag{10}$$

$KL(,)$ is Kullback-Leibler divergence. Also it can be easily shown that,

$$\frac{\partial^2 Q(\theta,\theta^t)}{\partial\theta^2}\Big|_{\theta^t} = -G(\theta). \tag{11}$$

Here, $G(\theta^t)$ is the Fisher information matrix where,

$$G(\theta) = -\int q(\boldsymbol{x})p(\boldsymbol{z},y|\boldsymbol{x},\theta)\frac{\partial^2 l(z,y|\boldsymbol{x},\theta)}{\partial\theta^2}d\mu_x d\mu_y d\mu_z.$$

Using these facts, the first term of (9) is equal to (10), and the second term of (9) is equal to (11). (9) can be rewritten as (13)(This result is also shown in [4]).

$$\frac{\partial L(y_1^N|\theta^t)}{\partial\theta} - G(\theta^t)(\theta^{t+1}-\theta^t) \simeq 0 \tag{12}$$

$$(\theta^{t+1}-\theta^t) \simeq G^{-1}(\theta^t)\frac{\partial L(y_1^N|\theta^t)}{\partial\theta}. \tag{13}$$

B. Model Search Algorithm

In this article, the author gives an algorithm to find the best model by modifying the structure of the model little by little. In the algorithm, the author proposes a way to modify the model by adding new parameters to the model.

If new parameters are added at random, then the algorithm is just a random search algorithm. In this article, the author shows the way to evaluate how good the parameter will make the model without estimating them. With the evaluation, we can decide which parameter should be added to the model.

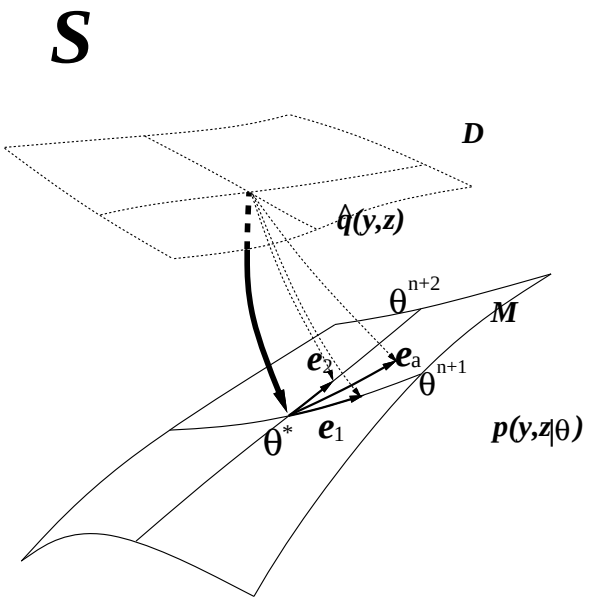

Figure 3. Geometrical view of the algorithm

Figure 3 describes the algorithm schematically. The S is the space of probabilistic models that each point of S is a probabilistic distribution. The M is a submanifold of the model $q(\boldsymbol{x})p(y,z|\boldsymbol{x},\theta)$ which is embedded in S, where θ is the coordinate. Observed data can only give us the empirical distribution of the visible data. Therefore the probability distribution of the hidden variable z can be assigned arbitrarily, and this ambiguity consists submanifold D can be defined in S.

Now, the model $p(y,z|\boldsymbol{x},\theta)$ has n parameters and θ^* is the Maximum Likelihood Estimator. Suppose the case we can add only one of $\theta_{n+1}\cdots\theta_{n+k}$ in the next step. In this figure only two dimensions for θ_{n+1} and θ_{n+2} are shown, but actually, the model consists a manifold of $n+k$ dimensions [5].

With θ^*, $\partial_i KL(q,p) = \partial KL(q(y,\boldsymbol{x}),q(\boldsymbol{x})p(y,\boldsymbol{x}))/\partial\theta_i = 0$, $i=1,\cdots,n$. The candidate parameters are $\theta_{n+1}\cdots\theta_{n+k}$. We want to select the parameter which makes the model better among them. Here "better" means making $KL(q(y,\boldsymbol{x}),q(\boldsymbol{x})p(y|\boldsymbol{x},\theta))$ smaller. This is equal to make the likelihood function larger. For new parameters $\theta_{n+1}\cdots\theta_{n+k}$, usually $\partial_i KL(q,p)$ are not 0. Therefore it seems that we should select the parameter which makes $\partial_i KL(q,p)$ largest. But if we compare only the values of $\partial_i KL(q,p)$, it is not enough. We should discuss them regarding the Fisher metric of the $n+k$ dimensional manifold.

Let $\Theta = (\theta_1,\cdots,\theta_n,\theta_{n+1},\cdots,\theta_{n+k})$, and $\Theta_{(i)} = (\theta_1,\cdots,\theta_n,\theta_{n+i})$. By projecting $\partial L(y_1^N|\theta)/\partial\theta_i$ to the tangent space of the Model $p(y|\Theta)$, we can find the direction of the steepest gradient. In Figure 3, it

is shown as e_a. If we add only θ_{n+1} to the model, the steepest gradient is e_1 on the tangent space of $p(y|\Theta_{(1)})$, because the projection and the tangent space are different. Also, if θ_{n+2} is added, e_2. If we have to select one of θ_{n+1} and θ_{n+k} then it is natural to select the one that is closer to e_a

To determine which to select, if we can calculate the inner product of e_a and, e_i, then, we can compare the *cosine* between e_a and e_i and decide which is better.

It can be easily shown that, we can have the criterion for selecting the new parameter which is corresponding to the cosine between e_a and e_i. That is,

$$C_i = \frac{(\mathcal{L}^a)^t G^{-1}(\Theta)\mathcal{L}^i}{\sqrt{(\mathcal{L}^a)^t G^{-1}(\Theta)\mathcal{L}^a}\sqrt{(\mathcal{L}^i)^t G^{-1}(\Theta)\mathcal{L}^i}}. \quad (14)$$

Here, $\mathcal{L}^a = (\partial_1 KL, \cdots, \partial_n KL, \partial_{n+1} KL, \cdots, \partial_{n+k} KL)^t$ and $\mathcal{L}^i = (\partial_1 KL, \cdots, \partial_n KL, 0, \cdots, 0, \partial_{n+i} KL, 0, \cdots, 0)^t$. In the criterion, $\sqrt{(\mathcal{L}^a)^t G^{-1}(\Theta)\mathcal{L}^a}$ is negligible because it is common to all the models. And you can see, from equation (13), $G^{-1}(\Theta)\mathcal{L}^a$ is approximately equal to one step of EM algorithm. Therefore the criterion (14) can be rewritten as

$$C_i = \frac{(EM(\Theta) - \Theta^*)^t \mathcal{L}^i}{\sqrt{(EM(\Theta_{(i)}) - \Theta^*_{(i)})^t \mathcal{L}^i}}, \quad (15)$$

where, $EM(\theta)$ is the parameter which we can have after only one EM step. If you want to estimate the parameter, you have to iterate EM step for many times, but C_i can be calculated without iteration.

IV. Experiments

I made a simple experiment. Figure 4 is the target model which gives the $q(y|\boldsymbol{x})$ and Figure 5 is the probabilistic model. the dimensions of input vector $\boldsymbol{x}$ and $\boldsymbol{z}$ are 4 for both. the probability of $q(\boldsymbol{x})$ is uniform. So, this time, $q(\boldsymbol{x}) = 1/16$ for each. First, we estimate the parameter of Figure 5 with EM algorithm. After the parameters are estimated, I select the parameter to add the model. If the parameter is the same place as the target model, then the algorithm is thought to be effective.

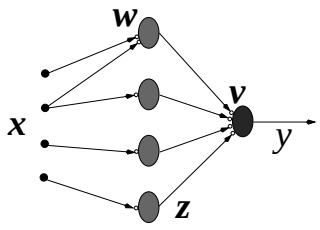

Figure 4. Target

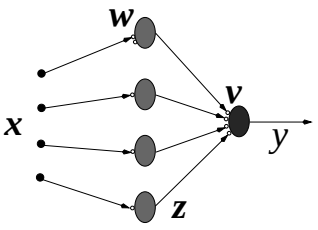

Figure 5. Model

100 target models were made at random. And each time the parameter to be added is selected according to the algorithm. Candidate parameters are 12 at each time. If you select the parameter at random, then the correct rate would be 1/12. But the result shows the rate is double. The results are shown in Table 1. The result shows that the algorithm is effective for finding the better parameter.

Random	1/12(=0.083)
Algorithm	17/100(=0.17)

Table 1. Result

V. Conclusion

By the algorithm, it seems that we can select the better parameters. Still it is not 100% correct. This algorithm tries to find the best parameter according to the first order approximation. This means that in the algorithm, the model is approximated by the tangent space and the one candidate is selected. But the manifold is usually not flat. The approximation does not work well when the converged point of the MLE is far from the current point.

Even selected one parameter is OK, there still remained some problem. If we continue this way of selecting the parameter, what the consequent models would be? The author is working for theoretical understanding of the algorithm and the study will help to understand this.

There are still some problems for this algorithm. In this case, this algorithm cannot be used for adding the new cell. Adding the new cell is corresponding to adding a new hidden variable and this is a difficult problem. The author solved this problem according to each model, but has not found global solution for this problem.

References

[1] Shun-ichi Amari, "Dualistic geometry of the manifold of higher-order neurons," *Neural Networks*, vol. 4, no. 4, pp. 443–451, 1991.

[2] Shun-ichi Amari, "The EM Algorithm and Information Geometry in Neural Network Learning," *Neural Computation*, vol. 7, no. 1, pp. 13–18, 1995.

[3] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *J. R. Statistical Society, Series B*, vol. 39, pp. 1–38, 1977.

[4] D.M Titterington, "Recutsive parameter estimation using incomplete data," *J. R. Statistical Society, Series B*, vol. 46, no. 2, pp. 257–267, 1984.

[5] Shun-ichi Amari, *Differential-Geometrical Methods in Statistics*, vol. 28 of *Lecture Notes in Statistics*, Springer-Verlag, Berlin, 1985.