人　工　知　能 103－6
（1996．1．17）

最尤法を用いたニューラルネットワークからの命題獲得

森田 千絵　月本 洋
(株) 東芝 研究開発センター

筆者らは以前ニューラルネットワークから命題を抽出する方法を提示した。その方法はニューラルネットワークの各素子が多重線形関数 (定義域が連続の時は近似的に) であることに着目して、それを (連続) ブール関数で近似することである。その近似の際にはユークリッド距離を用いていたが、本稿では最尤法を用いる近似方法を提示し、良好な結果が得られることを示す。

Finding Propositions from Neural Networks using Maximum Likelihood Method

Chie Morita　Hiroshi Tsukimoto
Research & Development Center,
Toshiba Corporation

Authors presented an algorithm for finding propositions from neural networks. The outline is as follows. When the domain is discrete, the functions which neural networks learn are multi-linear functions. The space of multi-linear functions is an Euclidean space and includes Boolean functions. When the domain is continuous, the above fact approximately holds. So, multi-linear functions are approximated by (continuous) Boolean functions. This paper describes the algorithm using maximum likelihood method and shows that experimental results are good.

1 はじめに

筆者は、以前ニューラルネットワークから命題を抽出する方法を提示した。それは、ニューラルネットワークが学習する関数は定義域が離散の場合には多重線形関数になる (連続値の場合には近似的に) ことに着目し、それをブール関数で近似する、というものであった。近似の際、ユークリッド距離を適用して関数間の距離を測定していたが、本稿では KL 情報量によって距離を測定し、良好な結果が得られることを示す。

以下、第 2 章でニューラルネットワークからの学習方法について述べ、第 3 章で KL 情報量を用いたアルゴリズムについて説明する。そして、第 4 章で実験結果について述べ、まとめとする。

2 ニューラルネットワークからのブール関数獲得

本章では、ブール関数の獲得方法について簡単に述べる。詳細は、文献 [1] を参照。

2.1 多重線形関数空間

多重線形関数空間について文献 [1] に述べられているので、ここでは簡単に説明する。

n 変数の多重線形関数 $f(x_1,\cdots,x_n)$ とは、

$$f(x_1,\cdots,x_n)=\sum p_i x_1^{e_1}\cdots x_n^{e_n}$$

のことである。ここで、p_i は実数、x_i は変数、e_i は 1 か 0 である。例えば、2 変数の多重線形関数 $f(x,y)$ は $f(x,y)=pxy+qx+ry+s$ である。このような関数の集合を多重線形関数空間と呼ぶ。

今、変数の定義域を $\{0,1\}$ とし、内積を以下のように定義する。

$$<f,g>=\sum_{\{0,1\}^n} fg$$

この時、多重線形関数空間はブール関数のブール代数の原子で張られるユークリッド空間である。

定義域が $[0,1]$ の場合は、内積 $<f,g>$ を次の様に定義すると多重線形関数空間はユークリッド空間になる。証明は [2] を参照。

$$<f,g>=2^n\int_0^1 \tau(fg)dx$$

ただし $\tau(x^n)=x$ である。

以降では、多重線形関数を論理関数とも言う。多重線形関数空間はユークリッド空間であるから、この論理関数はベクトルで表現される。これを論理ベクトルと呼ぶ。

2.2 ニューラルネットワークのブール関数近似

文献 [3] にある通り、ニューラルネットワークが学習できる関数は、定義域が離散の場合は多重線形関数である。従って、ニューラルネットワークは直観主義論理の命題とみなせる。定義域が連続の場合には近似的に考えることができる。すなわち、

$$x^n=\begin{cases} x(n\le a) \\ 0(n>a)\end{cases}$$

とする。但し a はある自然数である。

今、学習された多重線形関数を最も近いブール関数で近似することを考える。多重線形関数の論理ベクトルを $\boldsymbol{f}=(f_i)$、ブール関数の論理ベクトルを $\boldsymbol{g}=(g_i)$ $(g_i=0,1)$ とする。これは、n 変数の場合はそれぞれ 2^n 次元ベクトルである。このとき、

$$g_i=\begin{cases} 1 & f_i\ge 0.5 \text{ の時} \\ 0 & f_i<0.5 \text{ の時}\end{cases}$$

によって、g_i を求めればよい。それは以下のような理由による。多重線形関数に最も近いブール関数ということは、$\sum(f_i - g_i)^2$ を最小にするような g_i を求めれば良い。それは、各項は独立に最小化するのと同値なので、各 i に対して $|f_i - g_i|$ をそれぞれ最小にすれば良いことになる。また、g_i はブール関数の論理ベクトルの成分なので、$g_i = 0$ または 1 である。従って、0.5 をしきい値として近似することにより、多重線形関数をブール関数で近似することができる。

しかし、この方法では各項について計算することが必要であり、計算量が指数オーダーになる。そこで、多項式オーダーでブール関数を生成するアルゴリズムを開発した。これは、ブール関数の形を DNF 式とし、低次の項から生成していき、途中で打ち切る。打ち切らずに最高次まで求めれば各項について求めた場合と同等のブール関数が得られるが、指数オーダーになる。途中で打ち切った場合には、多少誤差が発生するが、ほとんどの情報は低次の項に存在し、ある次数までで良い近似が得られることがわかっている [4]。

$S(\cdot)$ を出力関数として (今はシグモイド関数とする)、ニューラルネットワークの素子を

$$S(p_1x_1 + \cdots + p_nx_n + p_{n+1})$$

とした時、近似後のブール関数に、

$$x_{i_1} \cdots x_{i_k} \overline{x}_{i_{k+1}} \cdots \overline{x}_{i_l}$$

という項が存在する条件は以下の通りである。

$$S(\sum_{i_1}^{i_k} p_{i_j} + p_{n+1} + \sum_{1 \le j \le n, j \ne i_1, \cdots, i_l, p_j \le 0} p_j) \ge 0.5$$

この判定条件を用いて低次から項を生成していき、生成された項を論理和で接続することによりブール関数が得られる。

ところで、この手法はユークリッド距離を仮定している。すなわち、しきい値に 0.5 を用いているのは、論理関数間にユークリッド距離が入っていると仮定し、成分ごとに 1 か 0 の近い方へ近似するためである。しかし、その他の距離も考えることが可能であり、本稿では KL 情報量を用いて近似する方法を提案する。

3 KL 情報量によるしきい値の決定

3.1 KL 情報量

KL 情報量 (Kullback-Leibler 情報量) は、与えられたデータから真の確率分布を推定する際に、モデルと真の分布の距離を測る規準の 1 つとして知られている。離散分布の場合、真の分布を $\boldsymbol{p} = \{p_1, p_2, \cdots, p_m\}$、モデルを $\boldsymbol{q} = \{q_1, q_2, \cdots, q_m\}$ とすると、KL 情報量 $I(\boldsymbol{p}; \boldsymbol{q})$ は、

$$I(\boldsymbol{p}; \boldsymbol{q}) = E[\log \frac{\boldsymbol{p}}{\boldsymbol{q}}] = \sum_{i=1}^{m} p_i \log \frac{p_i}{q_i} = \sum_{i=1}^{m} p_i \log p_i - \sum_{i=1}^{m} p_i \log q_i$$

である。この $I(\boldsymbol{p}; q)$ の値が小さく 0 に近いほど、モデル $\boldsymbol{q}$ は真の分布に近いとみなされる。右辺の第 1 項は、真の分布 $\boldsymbol{p}$ に依存する定数なので、KL 情報量の大小の比較のためには、第 2 項が推定できればよい。この $\sum_{i=1}^{m} p_i \log q_i$ は、確率変数 $\log \boldsymbol{q}$ の期待値であり、これは平均対数尤度と呼ばれる。実際には、真の確率 $\boldsymbol{p}$ も観測された度数 $\boldsymbol{N} = (N_1, \cdots, N_m)$ によって推定されるが、この対数尤度をの大きなモデルが良いモデルということになる。いくつかモデルがある場合には、対数尤度が最大となるモデルを選択する。この推定法は最大尤度法、略して最尤法と呼ばれている。

ところで、論理ベクトルは確率と対応づけることができる。それにより KL 情報量を適用して、学習された多重線形関数の論理ベクトルを、ブール関数の論理ベクトルで近似することが実現される。そこでまず、次節では、論理と確率との対応について述べる。

3.2 確率との対応

確率と論理を以下のようにして対応づける。例えば、$X \vee \bar{X}$ という命題を論理ベクトルで表すと (1,1) である。この命題はトートロジであり、恒等的に正しく、具体的情報を含んでいない。一方、このように情報を含んでいない場合の確率分布は (1/2, 1/2) である。ここで、無差別原理を適用している。これを拡張すると、n 変数のトートロジの場合には、

$$(1.\cdots,1) \leftrightarrow (\frac{1}{2^n},\cdots,\frac{1}{2^n})$$

という対応が存在することになる。

また、1 変数の場合に X という命題は、論理ベクトルでは $(1,0)$ と表され、確率分布では事象 X が起こることが既知なので $(1,0)$ というベクトルが対応する。さらに、X、Y の 2 変数の場合の X という命題のベクトル表示は $(1,1,0,0)$ であり、確率分布は $X \wedge Y$ か $X \wedge \bar{Y}$ なので $(1/2,1/2,0,0)$ となる。これを拡張すると、m 個の成分が 1 であるような論理ベクトルとそれに対応する確率ベクトルとの対応は

$$(\overbrace{1,\cdots,1}^{m},0,\cdots,0) \leftrightarrow (\overbrace{\frac{1}{m},\cdots,\frac{1}{m}}^{m},0,\cdots,0)$$

となる。

3.3 アルゴリズム

KL 情報量を用いた近似方法を説明する。多重線形関数の論理ベクトル $\boldsymbol{f}=(f_i)$ に対応する確率分布を $\boldsymbol{p}=(p_i)$、ブール関数の論理ベクトル $\boldsymbol{g}=(g_i)$ に対応する確率分布を $\boldsymbol{q}=(q_i)$ と書くことにする。$\boldsymbol{p}$ に近い $\boldsymbol{q}$ を求め、それに対応するブール関数を求めることが目的である。そこで、$\boldsymbol{q}$ に関する $\boldsymbol{p}$ の KL 情報量を最小にすることを考えよう。すなわち、

$$\sum p_i \log p_i - \sum p_i \log q_i \rightarrow \min \tag{1}$$

となるような (q_i) を求める。(1) 式において、第一項は定数になるので、

$$\begin{aligned} &-\sum p_i \log q_i \\ &= -(p_1 \log q_1 + p_2 \log q_2 + \cdots + p_{2^n} \log q_{2^n}) \rightarrow \min \end{aligned} \tag{2}$$

と同値になる。ところで、q_i はブール関数の論理ベクトルなので $q_i=0$ または $1/m$ であった。今、$p_1 \geq p_2 \geq \cdots \geq p_{2^n}$ となるように (p_i) を並べ替えると、p_i と q_i は近いことが予想されるので、p_i を大きい順に並べた場合、対応する q_i も大きい順に並ぶ。つまり、$q_1=\cdots=q_m=\frac{1}{m}$、$q_{m+1}=\cdots=q_{2^n}=0$ となる。これにより、

$$\begin{aligned} (2) &= -(p_1+\cdots+p_m)\log\tfrac{1}{m} - (p_{m+1}+\cdots+p_{2^n})\log\epsilon \\ &= (p_1+\cdots+p_m)\log m + (p_{m+1}+\cdots+p_{2^n})\log\tfrac{1}{\epsilon} \end{aligned} \tag{3}$$

となる。但し、$\log 0 = \log\epsilon(\epsilon>0)$ とおいた。

すなわち、これを満たすような m を求めることが、KL 情報量を最小にすることになる。このことは、前節で述べた対応によって論理ベクトル側に置き換えると、ブール関数の論理ベクトルの成分のうち 1 である個数を定めていることになる。これは、多重線形関数の論理ベクトル $\boldsymbol{f}$ の成分に対してしきい値を求めていることに他ならない。

さらに、$p_i=\frac{f_i}{\sum f_i}$ で置き換えると

$$\begin{aligned} (3) &= \tfrac{1}{\sum f_i}(f_1+\cdots+f_m)\log m + \tfrac{1}{\sum f_i}(f_{m+1}+\cdots+f_{2^n})\log\tfrac{1}{\epsilon} \\ &= \tfrac{1}{\sum f_i}((f_1+\cdots+f_m)\log m + (f_{m+1}+\cdots+f_{2^n})\log\tfrac{1}{\epsilon}) \rightarrow \min \\ &\Leftrightarrow (f_1+\cdots+f_m)\log m + (f_{m+1}+\cdots+f_{2^n})\log\tfrac{1}{\epsilon} \rightarrow \min \end{aligned} \tag{4}$$

となる。ところで、今$\sum q_i = 1$である。このうち、m個が$1/m$で$2^n - m$個が0だが、今0の代わりに$\epsilon(> 0)$を用いているので、最初のm個の値をxとすると、$mx + (2^n - m)\epsilon = 1$より、

$$x = \frac{1 - (2^n - m)\epsilon}{m}$$

となる。よって、

$$(4) = (f_1 + \cdots + f_m) \log \frac{1 - (2^n - m)\epsilon}{m} + (f_{m+1} + \cdots + f_{2^n}) \log \frac{1}{\epsilon} \rightarrow \min \tag{5}$$

となる。ここで、f_iはニューラルネットワークの素子から得られる多重線形関数の係数なので、

$$f_i = S(\sum a_i x_i + b) \ (= S_i \text{とおく})$$

である。Sはシグモイド関数である。したがって、

$$(5) = (S_1 + \cdots + S_m) \log \frac{1 - (2^n - m)\epsilon}{m} + (S_{m+1} + \cdots + S_{2^n}) \log \frac{1}{\epsilon} \rightarrow \min \tag{6}$$

である。さて、このf_iは2^n個存在するので、これらの組合せを全て求めて最小値を求めるのは莫大な計算量になる。そこで、我々は、$a_i (i = 1, \cdots, n)$を以下のように絶対値の順に並べ換え、

$$|a_1| \geq |a_2| \geq \cdots \geq |a_n|$$

上位20個の値のみを用いて(6)式が最小となる値を探す。つまり、S_iの大小に影響を与えているのは、a_iのうち絶対値の大きいものであることによって近似を行うのである。それにより、(5)式が最小になるようなmを求め、ブール関数gが得られる。

ただし、現在のこのアルゴリズムでは、いくつかの点で真の最尤法にはなっていない。1つには、最小値を求める際に全解探索を行っていないこと、また、$\log 0$を$\epsilon(> 0)$によって$\log \epsilon$で置き換えていることがある。これらは今後の課題である。

4 実験

実験について述べる。実験手順は次の通り。

1. ニューラルネットワークの学習

   データを与え、ニューラルネットワークの学習をおこなう。

2. ブール関数生成

   前章で説明した方法により、学習された多重線形関数をブール関数で近似し、命題を求める。

3. 評価

   命題によって各事例のクラスを予測し、正解率を求めることによって、得られた命題がデータをどの程度表しているかを評価する。

実験を行ったデータは、UCIのMachine Leraning Databasesから入手した*voting-records*である。データの意味、属性等については、文献[3]を参照。

4.1 *voting-records*

入力は16個であり、出力は1個である。クラスは「民主党」と「共和党」の2つである。学習方法はバックプロパゲーションで反復は2乗誤差0.01で停止している。学習には232個の事例を用い、予測も同じデータで行なっている。学習後のニューラルネットワークの予測正解率は100%である。

中間素子数を0、2、3、4個、重み係数の初期値を3通りで実験を行なった。その結果、得られた命題の正解率は以下の通りである。

中間素子数0

	正解率	しきい値
初期値1	0.970	1.43e-13
初期値2	0.970	7.97e-14
初期値3	0.970	7.34e-14

中間素子数2

	正解率	しきい値
初期値1	0.957	9.80e-01
初期値2	0.944	9.77e-01
初期値3	0.966	9.78e-01

中間素子数3

	正解率	しきい値
初期値1	0.935	2.62e-03
初期値2	0.953	7.22e-05
初期値3	0.944	4.42e-03

中間素子数4

	正解率	しきい値
初期値1	0.944	1.03e-03
初期値2	0.944	2.97e-03
初期値3	0.940	7.42e-03

KL情報量によって求めたしきい値は全体に小さいものとなった。正解率は良好である。
次に、民主党に関する命題を示す。共和党はその否定である。

```
(physician-fee-freeze:no)
∨(adoption-of-the-budget-resolution:yes)(synfuels-corporation-cutback:no)
```

これは、初期値2、中間素子数3個の場合の命題である。他の初期値の場合、中間素子数が異なる場合、で得られた命題もだいたい似たものとなった。しかし、やや複雑であった。命題を得た後の枝刈りは今後の課題である。

5 おわりに

本稿では、ニューラルネットワークからの命題抽出アルゴリズムにおいて、しきい値の決定に最尤法を用いる手法を提案した。これにより安定した精度の命題を獲得できることが実験によって確認された。最尤法によって、データの構造に依存せず、他のデータでも良いしきい値を求めることができると考えられる。さらに正解率を安定させるために、厳密に最尤法を行うこと、ニューラルネットワークの学習パラメータの調整等が今後の課題である。また、理解し易い命題を得るために、獲得した命題の枝刈りを行うことも検討していきたい。

参考文献

[1] 月本 洋：パターン処理の近似としての記号処理, 電子情報通信学会論文誌, Vol.J78-D-II No.2, 1995.

[2] 月本 洋: 命題論理の幾何的モデル, 情報処理学会論文誌, Vol.31, pp.783-791, 1990.

[3] 月本 洋、森田 千絵：ニューラルネットワークからの命題抽出, 情報処理学会研究報告 95-AI-103-5, 1996.

[4] Linial, N., Mansour, Y. and Nisan, N. : Constant depth circuits, Fourier Transform, and Learnability, *Proceedings of the 29th Annual Symposium on the Foundations of Computer Science*, pp.574-579, 1989.