質的変数で条件付けられた数法則の発見法

Discovery of a Set of Nominally Conditioned Polynomials

中野 良平* 斉藤 和巳
Ryohei NAKANO Kazumi SAITO

NTT コミュニケーション科学基礎研究所
NTT Communication Science Laboratories

Abstract : This paper shows that a connectionist law discovery method called RF5X can discover a law in the form of a set of nominally conditioned polynomials, from data containing both nominal and numeric values. RF5X learns a compound of nominally conditioned polynomials by using single neural networks, and selects the best one among candidate networks, and decomposes the selected network into a set of rules. Here a rule means a nominally conditioned polynomial. Experiments showed that RF5X works well in discovering such a law even from data containing irrelevant variables and a small amount of noise.

1 はじめに

科学的発見を支援する研究において、データから数法則を発見する課題は中心的である。法則発見システムを使えば、数値データからケプラーの第３法則 $T = kr^{3/2}$ などが発見できる。先駆的研究である BACON システム [3] の後、様々な法則発見法 [5, 9] が提案されたが、探索において組合せ爆発が生じる、あらかじめ適切な関数の定義が必要、データノイズに弱いなどの欠点があった。ニューラルネット（３層パーセプトロン）を用いた法則発見アプローチは、こうした問題点の解決に有望であり、筆者たちはすでに、２次の高速学習アルゴリズム BPQ[8] を用いたニューラルネット法則発見法 RF5 を提案した [7]。そして、RF5 法が大規模問題にも適用可能であるとともに、比例尺度としての変数変換に対して不変の法則を発見できることを示した [4]。

さて、RF5 法は全データに対して単一の多項式法則をあてはめるが、本来のデータ分布が部分空間毎に別々の多項式に従うような場合には、単一の多項式をあてはめようとしてもうまく行かない。たとえば、クーロンの法則 $F = 4\pi\epsilon q_1 q_2 / r^2$ は、r だけ離れた電荷 q_1 と q_2 の間に働く力 F の大きさを規定する。この法則は媒体の誘電率 ϵ を比例定数とし、たとえば、水の中では $F = 8897.352\ q_1 q_2 / r^2$、空気中では $F = 111.280\ q_1 q_2 / r^2$ のようになる。このような場合には、部分空間毎に別々の法則を用意すべきである。ここでは、部分空間が質的変数（名義尺度または順序尺度で計量された変数）で分割される場合を想定し、RF5 法を拡張した RF5X 法を提案する。

*連絡先：NTT コミュニケーション科学基礎研究所
619-0237 京都府相楽郡精華町光台 2-4
Tel: 0774-93-5151, Fax: 0774-93-5155
Email: nakano@cslab.kecl.ntt.co.jp

2 RF5X 法

2.1 基本的枠組み

各サンプルが変数群 $(q_1, \cdots, q_{K_1}, x_1, \cdots, x_{K_2}, y)$ で表わされるとする。q_k は質的変数、x_k は量的変数、y は基準変数とする。また、L_k を質的変数 q_k がとり得るカテゴリーの数とする。処理の便宜上、q_k を以下のような２進のダミー変数 q_{kl} で表す。

$$q_{kl} = \begin{cases} 1 & \text{第 } l \text{ カテゴリーをとるとき} \\ 0 & \text{それ以外のとき} \end{cases} \quad (1)$$

ここでは、真のデータモデルが以下のようなルールから成ると仮定する（$r = 1, \cdots, R$）。各ルールの条件部は質的変数の論理積とする。ただし、$Q^{(r)}$ は条件部に現れるダミー変数の集合とする。

$$if \bigwedge_{q_{kl} \in Q^{(r)}} q_{kl} = 1 \quad then \quad y = f(\boldsymbol{x}; \boldsymbol{W}^{(r)}) \quad (2)$$

実行部はパラメータ $\boldsymbol{W}^{(r)}$ を持つ多項式とする。

$$\begin{aligned} f(\boldsymbol{x}; \boldsymbol{W}^{(r)}) &= w_0^{(r)} + \sum_{j=1}^{J^{(r)}} w_j^{(r)} \prod_{k=1}^{K_2} x_k^{w_{jk}^{(r)}} \\ &= w_0^{(r)} + \sum_{j=1}^{J^{(r)}} w_j^{(r)} \exp\left(\sum_{k=1}^{K_2} w_{jk}^{(r)} \ln x_k \right) \\ &= \sum_{j=0}^{J^{(r)}} w_j^{(r)} \exp\left(\sum_{k=1}^{K_2} w_{jk}^{(r)} \ln x_k \right) \end{aligned} \quad (3)$$

2.2 条件部の数値表現

条件部を数値表現するために、以下のような関数 g を導入する。

$$g(\boldsymbol{q};\boldsymbol{V}^{(r)}) \;=\; \exp\left(\sum_{k=1}^{K_1}\sum_{l=1}^{L_k} v_{kl}^{(r)} q_{kl}\right) \qquad (4)$$

ここで、$\boldsymbol{V}^{(r)}$ は上式右辺に現れるパラメータから成るベクトルとする。いま、このパラメータ値を以下のように設定する。

$$v_{kl}^{(r)} \;=\; \begin{cases} 0 & if\ q_{kl} \in Q^{(r)} \\ -\beta & if\ q_{kl} \notin Q^{(r)} \exists q_{kl'} \in Q^{(r)}, l' \neq l \\ 0 & if\ \forall q_{kl'} \notin Q^{(r)} \end{cases}$$

β を大きな正数とすれば、条件部が成立するとき、$g(\boldsymbol{q};\boldsymbol{V}^{(r)}) \;=\; 1$ となり、成立しないときには、$\exp(-\beta) \approx 0$ が効いてきて、$g(\boldsymbol{q};\boldsymbol{V}^{(r)}) \approx 0$ となるので、以下は式 (2) を十分な精度で近似する。

$$\begin{aligned} & g(\boldsymbol{q};\boldsymbol{V}^{(r)})\ f(\boldsymbol{x};\boldsymbol{W}^{(r)}) \\ &= \sum_{j=0}^{J^{(r)}} w_j^{(r)} \exp\left(\sum_{k=1}^{K_1}\sum_{l=1}^{L_k} v_{kl}^{(r)} q_{kl} + \sum_{k=1}^{K_2} w_{jk}^{(r)} \ln x_k\right) \\ &\approx \begin{cases} f(\boldsymbol{x};\boldsymbol{W}^{(r)}) & if \bigwedge_{q_{kl}\in Q^{(r)}} q_{kl} = 1 \\ 0 & \text{それ以外のとき} \end{cases} \end{aligned} \qquad (5)$$

したがって、それらを足し合わせた下式は真のデータモデルを十分な精度で近似する。ただし、$\boldsymbol{\Psi}$ はパラメータ群 $\boldsymbol{V}^{(r)}, \boldsymbol{W}^{(r)}, r = 1, \cdots, R$ から成る。

$$F(\boldsymbol{q},\boldsymbol{x};\boldsymbol{\Psi}) \;=\; \sum_{r=1}^{R} g(\boldsymbol{q};\boldsymbol{V}^{(r)})\ f(\boldsymbol{x};\boldsymbol{W}^{(r)}) \qquad (6)$$

2.3 ニューラルネットによる学習

利用できるデータを $\{(\boldsymbol{q}^\mu, \boldsymbol{x}^\mu, y^\mu), \mu = 1, \cdots, N\}$ とする。このとき、以下の計算式は J を適切に用意すれば、式 (6) を完全に表現することができる。

$$\begin{aligned} & y(\boldsymbol{q}^\mu, \boldsymbol{x}^\mu; \boldsymbol{\Theta}) = w_0 \\ & + \sum_{j=1}^{J} w_j \exp\left(\sum_{k=1}^{K_1}\sum_{l=1}^{L_k} v_{jkl} q_{kl}^\mu + \sum_{k=1}^{K_2} w_{jk} \ln x_k^\mu\right) \end{aligned} \qquad (7)$$

$\boldsymbol{\Theta}$ はパラメータベクトルで、$w_j, j \;=\; 0, \cdots, J$、$v_{jkl}, j \;=\; 1, \cdots, J, k \;=\; 1, \cdots, K_1, l \;=\; 1, \cdots, L_k$、および $w_{jk}, j = 1, \cdots, J, k = 1, \cdots, K_2$ から成る。また、その次数を M とする。式 (7) は、J 個の隠れユニットを持つ 3 層ニューラルネットのフィードフォーワード計算式と解釈できる。

ニューラルネット学習では、誤差関数 $E(\boldsymbol{\Theta})$ に重みのペナルティ項 $\boldsymbol{\Omega}(\boldsymbol{\Theta})$ を付加すると汎化性能が著しく向上することが知られている [1]。さらに、ペナルティ項付加には不要な重みをゼロにする効果も期待でき、法則発見の観点からも都合が良い。筆者たちの実験によれば、2 乗ペナルティと 2 次学習アルゴリズムの組合せが汎化性能の向上効果が著しい [6]。そこで、以下のような目的関数が考えられる。ただし、λ はペナルティ係数、$\theta_m \in \boldsymbol{\Theta}$ とする。

$$\begin{aligned} E(\boldsymbol{\Theta}) + \lambda\boldsymbol{\Omega}(\boldsymbol{\Theta}) \;=\; & \frac{1}{2}\sum_{\mu=1}^{N} \left(y^\mu - y(\boldsymbol{q}^\mu, \boldsymbol{x}^\mu; \boldsymbol{\Theta})\right)^2 \\ & + \lambda\left(\frac{1}{2}\sum_{m=1}^{M} \theta_m^2\right) \end{aligned} \qquad (8)$$

しかし、このようにペナルティ係数が単一の 2 乗ペナルティは、変数変換に不変でない重みを求めるという欠点がある [1]。実際、$\widetilde{x}_k = a_k x_k$、$\widetilde{y} = cy + d$ のような変数変換に不変な重みを求めるには、重み毎に異なるペナルティ係数を用意する必要があるが、これは係数の数が多すぎて学習が容易でない。そこで、ここでは、入力層から隠れ層への重みだけにペナルティを課すことにする。こうすれば、上記のような変数変換にも不変な重みを単一のペナルティ係数で求めることができる。すなわち、式 (7) に従う法則の発見問題は、以下の目的関数を最小にするパラメータを求める問題として定式化できる。

$$\begin{aligned} J(\boldsymbol{\Theta}) \;=\; & E(\boldsymbol{\Theta}) \\ & + \lambda\left(\frac{1}{2}\sum_{j=1}^{J}\sum_{k=1}^{K1}\sum_{l=1}^{L_k} v_{jkl}^2 + \frac{1}{2}\sum_{j=1}^{J}\sum_{k=1}^{K2} w_{jk}^2\right) \end{aligned} \qquad (9)$$

2.4 モデル選択の基準

いま、われわれはデータを与えられているが、式 (7) の最適隠れユニット数 J^* も式 (9) の最適ペナルティ係数 λ^* も知らない。データを説明する最適なモデルとモデルパラメータを求めるためには、データから J^* や λ^* を選択する基準が必要である。RF5X ではこの基準として交差検証法 cross-validation を用いる。交差検証法はニューラルネットなどの学習機械の汎化性能を評価するのにしばしば利用される [1]。ここで、汎化とは、学習に用いないデータに対する性能を指す。交差検証法では、与えられたデータ D をランダムに S 個のセグメントに分割する $(G_s,\ s = 1, \cdots, S)$。そして、$S-1$ 個のセグメントを学習に使用し、残りの 1 個を（汎化の）テストに使用する。セグメント数とサンプル数が等しい場合は、特に、一つ抜き法 leave-one-out と呼ばれ

る。この手順を S 回繰り返し、最終的には以下の平均2乗誤差 MSE_{CV} を求める。

$$MSE_{CV} = \frac{1}{N}\sum_{s=1}^{S}\sum_{\mu\in G_s}(y^\mu - y(\boldsymbol{q}^\mu, \boldsymbol{x}^\mu; \widehat{\boldsymbol{\Theta}}_s))^2.$$

2.5 ルール分解アルゴリズム

上記基準を用いて、一つのニューラルネットを最良の法則候補として選択できたとする。それを元のルール集合に分解する必要がある。重み $\boldsymbol{W}$ は多項式の形成に使えばよいが、重み $\boldsymbol{V}$ の利用の仕方はそれほど単純ではなく、以下に示す手順でルール分解に使う。なお、微小な変動を無視するパラメータ ϵ を導入し、無用なルールの発生を防ぐ。

step 1: 無変動重みのゼロシフト
以下の手順を各隠れユニット j について繰り返す。質的変数 k の重みが $\max_l(v_{jkl}) - \min_l(v_{jkl}) \leq \epsilon$ のように無変動とみなせるときには、定常分を重み w_j に掛けてからゼロにシフトする ($v_{jkl} = 0$)。

$$w'_j = w_j \exp\left(\sum_k \frac{1}{L_k}\sum_l v_{jkl}\right). \quad (10)$$

step 2: 基本ルールの抽出
以下の手順を各隠れユニット j について繰り返す。変動する重みを持つ質的変数群については、各ダミー変数の全ての組合せをつくり、u で採番する。u 番目の組合せに対して、ダミー変数の集合 $Q_{j:u}$ が定まるので、それに対応する重みの寄与分を計算し、重み w'_j に掛ける。

$$w_{j:u} = w'_j \exp\left(\sum_{(k,l):q_{kl}\in Q_{j:u}} v_{jkl}\right) \quad (11)$$

かくして、質的変数の論理積を条件部に持つ以下のような基本ルールが抽出できる。なお、$|w_{j:u}| < \epsilon$ である重みに対しては、$t_{j:u} = 0$ とする。

$$if \bigwedge_{q_{kl}\in Q_{j:u}} q_{kl} = 1 \quad (12)$$
$$then\ t_{j:u} = w_{j:u}\exp\left(\sum_{k=1}^{K_2} w_{jk}\ln x_k\right)$$

step 3: 基本ルールの合成
隠れユニット j、j' で求めた基本ルールを合成する。合成は全ての組合せについて行ない、条件部は論理積として合成し、実行部は単なる和をとる。

$$if \left(\bigwedge_{q_{kl}\in Q_{j:u}} q_{kl} = 1\right) \bigwedge \left(\bigwedge_{q_{k'l'}\in Q_{j':u'}} q_{k'l'} = 1\right)$$
$$then\ t_{j'':u''} = t_{j:u} + t_{j':u'} \quad (13)$$

なお、条件部が常に「偽」となる組合せは破棄する。上記の合成を隠れユニットが無くなるまで繰り返して、共通の定数項 w_0 を各実行部に可算すれば、式 (2) のようなルールに分解できる。

3 数値実験

3.1 人工データ

質的変数 q_2 で条件付けられた多項式群を考える。

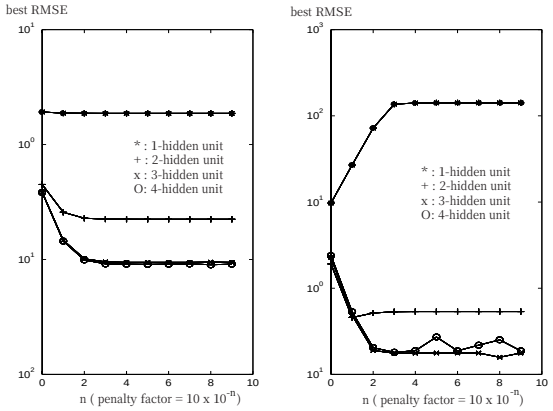
$$\begin{cases} if \quad q_{21} = 1 \quad then \quad y = 2 + 3x_1^{-1}x_2^3 \\ if \quad q_{22} = 1 \quad then \quad y = 3 + 4x_3x_4^{1/2}x_5^{-1/3} \\ if \quad q_{23} = 1 \quad then \quad y = 3 + 4x_3x_4^{1/2}x_5^{-1/3} \end{cases} \quad (14)$$

質的変数を q_1, q_2, q_3、量的変数を $x_1, \cdots, x_9$ とし、カテゴリ数 L_k は 2,3,4 とする。$q_1, q_3, x_6, \cdots, x_9$ は法則に無関係な変数である。各変数値はランダムに生成するが、量的変数値は区間 $[0, 1]$ 内の値とし、基準変数値には小さな正規乱数を付加する。サンプル数を 400 とする ($N = 400$)。

学習では、重み v_{jkl}、w_{jk} の初期値は、平均 0、標準偏差 1 の正規分布に従うように、また、重み w_j の初期値は 0、定数項 w_0 の初期値は基準変数値の平均値とした。学習は、勾配が十分小さくなったとき ($\|\nabla J(\boldsymbol{\Theta})\|^2/M < 10^{-8}$)、または処理時間が 100 秒を越えたとき終了させた。ペナルティ係数 λ は、10×10^0 から 10×10^{-9} まで、10^{-1} 刻みとした。また、隠れユニット数は、1 から 4 まで変化させた ($J = 1, 2, 3, 4$)。同じ J と λ に対して、試行を 10 回繰り返した。

学習結果を図 1に示す。学習データに対する誤差 RMSE は $J = 4$ のとき最小になるが、学習データとは別に生成したテストデータ (10,000 サンプル) に対しては、$J = 3$ のとき最小になる。これより、$J^* = 3$ と推測できる。$J = 3$ のとき、RF5X が発見した法則例を示す。

$$\begin{aligned} y = {} & -7.71 + 9.16\exp(0.06q_{11} + 0.06q_{12} \\ & -0.03q_{21} + 0.07q_{22} + 0.07q_{23} + 0.03q_{31} \\ & +0.03q_{32} + 0.03q_{33} + 0.03q_{34}) + 0.046\exp \\ & (-0.01q_{11} + 0.01q_{12} + 4.17q_{21} - 1.73q_{22} \\ & -2.44q_{23} + 0.01q_{33} - 0.01q_{34})x_1^{-1.00}x_2^{2.98} \\ & +0.90\exp(-3.00q_{21} + 1.50q_{22} + 1.49q_{23} \\ & -0.01q_{31})x_3^{1.00}x_4^{0.49}x_5^{-0.33} \end{aligned} \quad (15)$$



(a) RMSE for training data (b) RMSE for test data

図 1: 人工データに対する RF5X の性能

これに、2.5 節で示したルール分解アルゴリズムを適用することにより、以下の法則が発見できた。

$$
\begin{aligned}
&if\ q_{21}=1 \quad then\ y = 2.01 + 2.98 x_1^{-1.00} x_2^{2.98} \\
&if\ q_{22}=1 \quad then\ y = 3.04 + 4.03 x_3^{1.00} x_4^{0.49} x_5^{-0.33} \\
&if\ q_{23}=1 \quad then\ y = 3.04 + 3.99 x_3^{1.00} x_4^{0.49} x_5^{-0.33}
\end{aligned}
$$

既存法 ABACUS のように事前に特定の関数形を用意する必要もなく、元の法則とほぼ同じものが復元できた。また、条件付き多項式群が 1 枚のニューラルネットで学習できる点も RF5X の特長である。

3.2 自動車データ

自動車データ†は 1985 年に米国に輸入された自動車やトラックの諸元と価格のデータである。11 の質的説明変数、14 の量的説明変数、および基準変数 (価格) がある。欠測値のあるサンプルを除いて、159 サンプルが利用できる ($N = 159$)。従来分析 [2] では、線形回帰と $k-NN$ 法を適用して、それぞれ、MDE=14.2 % と 11.8 % であった。ただし、MDE(mean deviation error) は $\frac{100}{N}\sum_\mu |y^\mu - \widehat{y}^\mu|/y^\mu$ で定義される。全データが学習に使用され、テストにも使用された。

RF5X の実験条件は前実験と同じとした。ただし、$J = 1, 2, 3$ とした。また、各変数は次のように変数変換したが、発見する法則はこれに不変であることが示せる。$\widetilde{y} = (y - mean(y))/std(y)$、$\widetilde{x}_k = x_k/max(x_k)$。

RF5X を 3 つのケース、すなわち、1) 量的変数だけのデータ、2) 量的変数＋1 質的変数のデータ、3) 全変数のデータ、に適用した。その結果、ケース 3 では過学習が生じて、汎化性能がケース 2 よりも劣化すること、および、ケース 2 の汎化性能はケース 1 を上回ることがわかった。ケース 2 で使用した質的変数は製造メーカ q_2 である。ケース 2 の中では、隠れユニット数 $J = 1$ の汎化性能が最良であった。$J = 1$ のとき、RF5X が発見した法則を示す。これをルール群に分解するのは簡単である。この法則の MDE は 8.4 % であった。なお、第 k 番目の製造メーカ (q_{2k}) を単に q_k と表わした。

$$
\begin{aligned}
\widetilde{y} = &-1.17 + 2.61 \exp(0.67 q_1 + 1.05 q_2 \\
&-0.12 q_3 - 0.08 q_4 + 0.28 q_5 - 0.03 q_6 + 0.50 q_7 \\
&+0.31 q_8 - 0.16 q_9 - 0.00 q_{10} + 0.19 q_{11} \\
&-0.08 q_{12} + 0.49 q_{13} + 0.62 q_{14} + 0.05 q_{15} \\
&+0.11 q_{16} + 0.39 q_{17} + 0.36 q_{18})\ \widetilde{x}_1^{-0.15}\ \widetilde{x}_2^{2.40} \\
&\widetilde{x}_3^{-2.31}\ \widetilde{x}_4^{-1.38}\ \widetilde{x}_5^{-4.10}\ \widetilde{x}_6^{3.24}\ \widetilde{x}_7^{-0.26}\ \widetilde{x}_8^{0.18}\ \widetilde{x}_9^{-0.26} \\
&\widetilde{x}_{10}^{0.22}\ \widetilde{x}_{11}^{0.34}\ \widetilde{x}_{12}^{0.23}\ \widetilde{x}_{13}^{-0.42}\ \widetilde{x}_{14}^{0.05}
\end{aligned}
\tag{16}
$$

†UCI 機械学習データベースを利用した

参考文献

[1] C.M. Bishop. *Neural networks for pattern recognition.* Clarendon Press, Oxford, 1995.

[2] D. Kibler, D.W. Aha, and M.K. Albert. Instance-based prediction of real-valued attribures. *Computational Intelligence*, 5:51–57, 1989.

[3] P. Langley, H.A. Simon, G. Bradshaw, and J. Zytkow. *Scientific discovery: computational explorations of the creative process.* MIT Press, 1987.

[4] R. Nakano and K. Saito. Computational characteristics of law discovery using neural networks. In *Proc. 1st Int. Conference on Discovery Science, LNAI 1532*, pages 342–351, 1998.

[5] B. Nordhausen and P. Langley. A robust approach to numeric discovery. In *Proc. 7th Int. Conf. on Machine Learning*, pages 411–418, 1990.

[6] K. Saito and R. Nakano. Second-order learning algorithm with squared penalty term. In *Advances in Neural Information Processing Systems 9*, pages 627–633, 1996.

[7] K. Saito and R. Nakano. Law discovery using neural networks. In *Proc. 15th International Joint Conference on Artificial Intelligence*, pages 1078–1083, 1997.

[8] K. Saito and R. Nakano. Partial BFGS update and efficient step-length calculation for three-layer neural networks. *Neural Computation*, 9(1):239–257, 1997.

[9] C. Schaffer. Bivariate scientific function finding in a sampled, real-data testbed. *Machine Learning*, 12(1/2/3):167–183, 1993.