令和３年度　修士論文

包除積分を利用したディープラーニングネットワークの構築

令和３年２月

九州工業大学大学院情報工学府　学際情報工学専攻

システム創成情報工学分野

19677501

板橋　将之

指導教員：本田　あおい

目次

第1章 序論

1.1 研究背景と目的

近年、画像処理や自然言語処理などのデータ分析の分野において機械学習の手法の一つであるディープラーニングが使用され高い成果を上げている。ディープラーニングが用いられる理由としては複雑な特徴量を自動検出し高い精度を得ることがあげられる。しかし、このディープラーニングは複雑であるために解釈性が損なわれるという問題点があり、一般的に解釈性と精度はトレードオフの関係性にあると考えられている。解釈性が必要な例で言うと、例えば、ディープラーニングで学習した医療診断ソフトにより病気の診断をしたとしても、その原因となるものが何かわからないのでは対処のしようがないのである。そこで私の研究では説明可能でありながら高い精度も保持できるようなネットワークを考えた時に、1+1 が 2 にならない非加法的測度であるファジィ測度と多項演算を用いて定義される包除積分という積分式を用いてネットワークを構築することを提案し、実際に高い精度が得られるのか、どのような解釈が得られるのか実験を行った。

1.2 アプローチ

包除積分はファジィ測度からなる積分式であるがメビウス逆変換を行うことによって重回帰のような式を得ることができる。包除積分を利用したディープラーニングの構築を行うために、このメビウス逆変換の式を採用し、ディープラーニングネットワークの出力で得られる式が最終的に得られる包除積分式となるようにネットワークを構築した。今回のネットワーク構築では、以前まで対応できていなかったデータに対しても正しく学習できるようにネットワークの層やユニット数を増やし、データへの汎用性を向上させた。解釈性についてもゲーム理論でよく使われるシャープレイ値をさらに拡張した 2 変数以上の相互作用関係まで見ることができる手法を取り入れさらなる解釈性の向上を目指した。また、提案手法の相対的な精度や解釈を見るためにデータセットの提供やコンペティションが行われている kaggle でよく使われている手法で、解釈もできるような手法である XGBOOST と比較する。実験に使用するデータは Python の機械学習ライブラリである scikit-learn 内に含まれている「diabetes」（日本語訳：糖尿病）に関するデータを用いることで糖尿病について予測しその原因について解釈していく。

第2章　数学的定義

本論文を通して Ω＝｛1,2,…,n｝は有限集合とする。
|A|は有限集合 A の要素の個数を、P(Ω)は Ω の部分集合からなる集合、つまり Ω のべき集合を表す。

2.1　包除積分の定義

2.1.1 包除積分

定義１（ファジィ測度[1,2]）Ω 上の集合関数 μ：P(Ω)→[0,∞]は
(i)μ(Ø)=0,
(ii)A,B∈P(Ω)で A⊂B ならば μ(A)≦μ(B)
を満たすとき、Ω 上のファジィ測度という。

定義２(t-ノルム)[0,K]上の二項演算⊗: $[0,K]^2$→　[0,K]は x,y,z∈[0,K]に対して、
(i)0⊗0=0, x⊗K=x,
(ii)x≦y implies x⊗z≦y⊗z,
(iii)x⊗y=y⊗x
(iv)(x⊗y)⊗z=x⊗(y⊗z)
を満たすとき t-ノルムという。

通常、t-ノルムは[0,1]上に定義されるのが一般的であるが本質的な違いはない、実際、[0,1]上の t-ノルム⊗は

$$x\otimes^{G}y:=\left(\frac{x}{K}\otimes\frac{y}{K}\right)K$$

とすることで簡単に[0,K]上の t-ノルムになり、このとき K が単位元となる。次に示す論理積、代数積、Dubois-Prade の t-ノルム、Dombi の t-ノルムの他にも多くの t-ノルムが提案されている。

論理積：　$x\wedge y:=\min(x,y)$,

代数積：$x\otimes^{P}y:=\frac{xy}{K}$,

Dubois-Prade：$x\otimes_{\lambda}^{DP}y:=\frac{xy}{\max(x,y,\lambda)},0\leqq\lambda\leqq K$,

Dombi: $x\otimes_{\lambda}^{Db}y:=\frac{1}{1+\left(\left(\frac{1}{x}-1\right)^{\lambda}+\left(\frac{1}{y}-1\right)^{\lambda}\right)^{\left(\frac{1}{\lambda}\right)}},\lambda\in(0,\infty)$,

性質(iv)により t-ノルムは自然に多項演算に拡張できる。例えば Dubois-Prade の t-ノルムは

$$\otimes^{DP}_{i\in A} x_i=\begin{cases} K, & A=\emptyset, \\ \bigwedge_{i\in A} x_i & x_i>\lambda\ ,i\in A, \\ \prod_{i:x_i<\lambda} \frac{x_i}{\lambda^{|\{i|x_i<\lambda\}|-1}}, & \text{otherwise.} \end{cases},$$

この掛け算型の演算、t-ノルムを用いて包除積分を定義する。被積分関数 f は Ω 上に定義された[0,K]に値をとる非負有界値関数である。今、Ω は n 個の要素から成る有限集合なので、Ω 上の関数は n 次元の数ベクトルで表せる。これを$f=(x_1, x_2,.. x_n)\in[0,K]^n$と書くことにする。

定義３（包除積分[4]）μ をファジィ測度、⊗を[0,K]上の t-ノルムとする。Ω 上の非負有界関数$f=(x_1, x_2,.. x_n) \in[0,K]^n$の μ と⊗による包除積分は次で定義される。

$$\otimes\int f\,d\mu := \sum_{A\in P(\Omega)\setminus\{\emptyset\}} M^{\otimes}(f|A)\,\mu(A),$$

ただし、

$$M^{\otimes}(f|A) := \sum_{B\in P(\Omega),B\supset A} (-1)^{|B\setminus A|} \otimes_{i\in B} x_i,$$

定義式に現れる足したり引いたりの項がたくさん出てくる$M^{\otimes}(f|A)$の計算は包除原理に基づくもので、これが包除積分の名前の由来である。Ω={1,2,3}のときの包除積分を書き下すと

$$\begin{aligned}\otimes\int f\,d\mu :=\ & (x_1 - x_1\otimes x_2 - x_1\otimes x_3 + x_1\otimes x_2\otimes x_3)\,\mu(\{1\}) \\ & +(x_2 - x_1\otimes x_2 - x_2\otimes x_3 + x_1\otimes x_2\otimes x_3)\,\mu(\{2\}) \\ & +(x_3 - x_1\otimes x_3 - x_2\otimes x_3 + x_1\otimes x_2\otimes x_3)\,\mu(\{3\}) \\ & +(x_1\otimes x_2 - x_1\otimes x_2\otimes x_3)\,\mu(\{1,2\}) \\ & +(x_1\otimes x_3 - x_1\otimes x_2\otimes x_3)\,\mu(\{1,3\}) \\ & +(x_2\otimes x_3 - x_1\otimes x_2\otimes x_3)\,\mu(\{2,3\}) \\ & +(x_1\otimes x_2\otimes x_3)\mu(\{1,2,3\})\end{aligned}$$

メビウスの反転公式を用いると包除積分は次のように別表現できる。

$$\otimes\int f\,d\mu = \sum_{A\in P(\Omega)\setminus\{\emptyset\}} \left(\otimes_{i\in A} x_i\right) m^{\mu}(A),$$

ただしm^{μ}は μ のメビウス変換：

$$m^{\mu}(A) := \sum_{B\subset A} (-1)^{|B-A|}\,\mu(B).$$

この別表現を３点集合 Ω={1,2,3}の場合で書き下すと

$$\begin{aligned}\otimes\int f\,d\mu :=\ & x_1 m^{\mu}(\{1\}) + x_2 m^{\mu}(\{2\}) + x_3 m^{\mu}(\{3\}) \\ & +(x_1\otimes x_2)\ m^{\mu}(\{1,2\}) + (x_1\otimes x_3)\ m^{\mu}(\{1,3\})\end{aligned}$$

$+(x_2 \otimes x_3)\ m^{\mu}(\{2,3\})$

$+(x_1 \otimes x_2 \otimes x_3)\ m^{\mu}(\{1,2,3\})$.

定義式に比べてすっきりとして見えるのは足したり引いたりの部分がメビウス変換に含まれているためであり、メビウス変換に含まれているためであり、メビウス変換も書き下してしまうと定義式と同程度に煩雑な式となる。こちらを定義として採用しなかったのは非加法的測度 μ がそのままの形で表れているほうが、より自然であると考えたためである。

2.1.2 単調性判別

集合関数 μ のメビウス変換m^{μ}が次の条件を満たすとき、μ は単調性を満たす。

任意の $A \in 2^{\Omega}$ に対して

$m^{\mu}(A) \geqq -\sum_{B \subset \neq A, B \ni i} m^{\mu}(B)$

が任意の$i \in A$に対して成り立つ。

2.1.3 シャープレイ値

包除積分で求められるファジィ測度は$2^n - 1$となり、Ωの要素数が少ない場合はファジィ測度からの解釈は比較的容易だが、Ωの要素数が多い場合に簡単にファジィ測度から各要因の貢献度などの比較を行うことは難しい。シャープレイ値はゲーム理論で使われている手法で、協力ゲームにおいてプレイヤーに貢献度に応じた報酬を分配する手法の一つである。このシャープレイ値をファジィ測度で計算することができ、各要因ごとの比較も可能となる。各要素のシャープレイ値は次式で計算できる。

$$\phi_i(\mu) = \sum_{A \in \Omega \setminus \{i\}} \frac{|A|!\,(n - |A| - 1)!}{n!} (\mu(A \cup \{i\}) - \mu(A))$$

$\Phi(\mu) = (\phi_1(\mu), \phi_2(\mu), \ldots, \phi_n(\mu))$の$\phi_i(\mu)$が要素$i$ の貢献度である。また、シャープレイ値は次のような性質がある。

1．個人合理性：$\phi_i(\mu) \geq \mu(\{i\})$

2．全体合理性：$\sum_{i \in \Omega} \phi_i(\mu) = \mu(\Omega)$

3．対称性：要素i, jが$\mu(S \cup \{i\}) \quad = \mu(S \cup \{j\}) \quad S: \Omega$ のi, jを含まないすべての部分集合の場合、$\phi_i(\mu) = \phi_j(\mu)$

4．加法性：2 つの特性関数vとwによって作られた協力ゲームの和$v + w$において、

$\phi_i(v + w) = \phi_i(v) + \phi_i(w)$

がΩのすべての要素に対してそれぞれ成り立つ。

5．ナルプレイヤーに関する性質：ナルプレイヤーとは、iがΩのiを含まないすべての部分集合Sについて、$\mu(S \cup \{i\}) = \mu(S)$

を満たすことを言い、このナルプレイヤーに対して報酬を与えない。

実験ではこのシャープレイ値を用いて各要素の比較を行っていく。

2.1.4 拡張型シャープレイ値

シャープレイ値は各要素の貢献度を示す値だが要素間の関係性を表すことはできていない。そこで、相互作用指標を計算できるようにしたのが次式である。

$$I(T) = \sum_{S \supseteq T} \frac{1}{|S|-|T|+1} m^{\mu}(S), \quad T \subseteq \Omega$$

$|T|=1$ の時$I(T)$はシャープレイ値となり、$|T| > 1$ の時、$I(T)$は相互作用指標となる。

2.2 ディープラーニング

2.2.1 パーセプトロン（単層ニューラルネットワーク）

ディープラーニングとは機械学習の一種で、生物のニューロンの働きをモデルとしたニューラルネットワークをコンピュータで処理したものである。ディープラーニングではニューロンの一つをユニット（もしくはノード）といい、ニューロンの部分をユニット、軸索の部分をエッジで表す。このユニットをいくつも繋げ入力層、隠れ層、出力層と複数の層にしていったものをディープラーニングという。

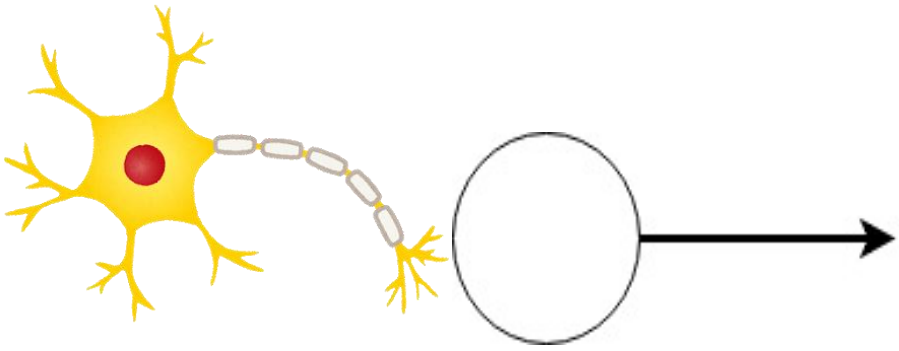


図 2.2.1　ニューロンとユニット

データを受け取る入力層と予測結果を出力する出力層の 2 層で構成されているものをパーセプトロンという。図 2.21 はパーセプトロンを示したものである。

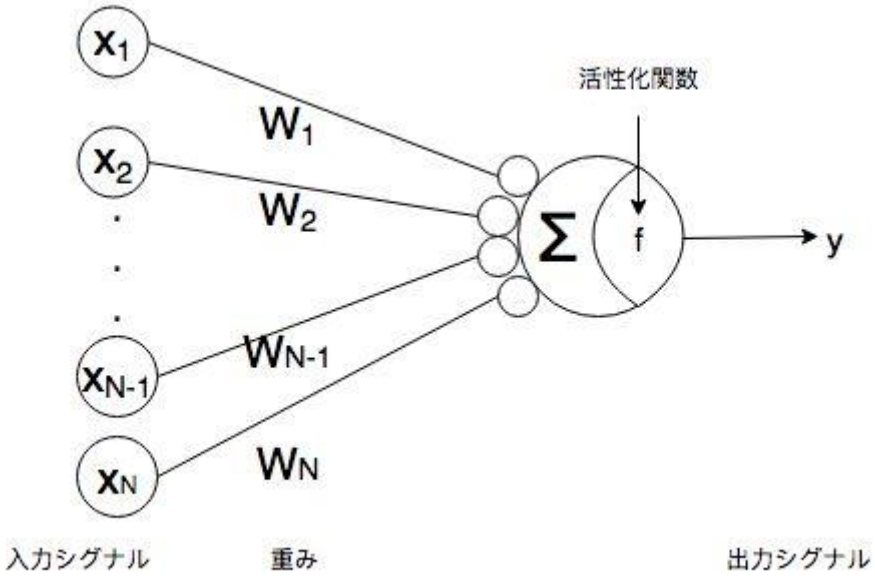


図 2.2.2　パーセプトロン

入力層と出力層の間のエッジには重みが与えられ、この重みを変えていくことにより出力で予測行う。図 2 のパーセプトロンの出力は $y=f(\sum_{i=1}^{N} x_i W_i)$となる。

このパーセプトロンを何層にもいくつも繋げたものを多層パーセプトロンという。ディープラーニングは多層パーセプトロンの重みをある手法で更新していくことで学習を行うものである。

2.2.2 活性化関数

活性化関数はニューロンの動きを模倣したもので、ある閾値を超えると発火して次のニューロンに電気信号を渡す働きを再現している。シグモイド関数、ReLU 関数、ステップ関数の他にも多数の活性化関数がある。

x は変数とする。

シグモイド関数：$\frac{1}{1+e^{-x}}$

ReLU 関数：$\max(0,x)$

ステップ関数：$\begin{cases} 1 & x>0 \\ 0 & x\leq 0 \end{cases}$

2.2.3 損失関数

損失関数は教師データと予測データの誤差を最小にするため重みやバイアスの値を最適化する仕組みを担うものである。この損失関数はディープラーニングで出力した予測値と実際の値との誤差全体を表す。今回実験に使用した損失関数は平均二乗誤差である。

平均二乗誤差：$E=\frac{1}{N}\sum_{i=1}^{N}(t_i-y_i)^2$

E：誤差関数、N：データ数、t：教師データ、y：予測データ

2.2.4 勾配降下法

勾配降下法は誤差関数を最小にするような重み、バイアスを求める手法である。
勾配降下法はいくつも提案されており、最も一般的な確率的勾配降下法の形を示すと、

確率的勾配降下法：$w^{t+1}=w^{t}-\mu\frac{\partial E(w^{t})}{\partial w^{t}}$

w：重み、t：更新タイム、E：誤差関数、μ：学習係数
の形で表す。今回の実験では Adam と呼ばれる手法を用いている。

Adam

$$m_{t+1}=\beta_1 m_t+(1-\beta_1)\nabla E(w^{t})$$
$$v_{t+1}=\beta_2 v_t+(1-\beta_2)\nabla E(w^{t})^2$$
$$\hat{m}=\frac{m_{t+1}}{1-\beta_1^{t}}$$
$$\hat{v}=\frac{v_{t+1}}{1-\beta_2^{t}}$$
$$w^{t+1}=w^{t}-\alpha\frac{\hat{m}}{\sqrt{\hat{v}}+\varepsilon}$$

ただしm_0=0、v_0=0、α=0.001、 β_1=0.9、β_2=0.999、 $\varepsilon=10^{-8}$とする。
w：重み、t：更新タイム、α：学習率、m と vは初期値 0 の変数である。

2.2.5 バッチ処理

バッチ処理は更新に使用するデータ数を変えていく処理である。データ全体を使ってパラメータを更新することをバッチ学習という。データ一つを使ってパラメータを更新することをオンライン学習という。今回使用するのはこのバッチ学習とオンライン学習の中間であるミニバッチという手法である。ミニバッチはデータ全体をいくつかに等分したものでそれぞれ更新する手法である。Dをランダムに M 等分して、各D_iで学習する場合、

ミニバッチ：$D=\bigcup_{i=1}^{M} D_i$

となる。

2.2.6 スパースモデリング

スパースモデリングはデータが不足している場合でも、少量のデータから解析を可能とする手法である。大量のデータを使わないため解析時間を短縮でき、データの構造をわかりやすく表現することができる。

2.3 重回帰分析

重回帰分析とは一つの目的変数と多数の説明変数の間の関係を予測する分析である。各説明変数の偏回帰係数a_iと切片a_0を求めることによって目的変数 y を求める。下に重回帰式を示す。

重回帰式：$y=a_1x_1+a_2x_2+\ldots+a_nx_n+a_0$

2.4XGBOOST

2.5 その他機械学習

2.6 評価方法

回帰問題では損失関数や決定係数、赤池情報量などの評価方法があるがこの論文では評価方法として以下の 3 つを採用し、実験ではこの値を見ることによってモデルの性能を評価していく。

1：平均絶対誤差（Mean Absolute Error　:MAE）

MAE は教師データと予測データの絶対値を平均したもので、小さいほど誤差が少なく良い予測モデルが得られていることがわかる。計算式は以下のようになる。

平均絶対誤差：$E=\frac{1}{N}\sum_{i=1}^{N}|t_i-y_i|$

E：誤差関数、N：データ数、t：教師データ、y：予測データ

2：平均 2 乗誤差（Mean Squared Error　:MSE）

MSE は 2.2.3 で示したような式で、MAE に比べ大きな誤差がある場合に大きな値になり易い。実験の損失関数ではこれを用いて学習を行うが MAE と同じく誤差が小さいほど良い予測モデルが得られていることがわかる。

３：決定係数（R^2）

回帰モデルの評価に良く用いられる指標で、良い予測モデルの場合 1 に近い値をとる。計算式はいくつか存在するが、今回の実験では scikit-learn の sklearn.metrics.r2_score 関数で使われている計算式を用いる。計算式は以下のようになる。

決定係数：$R^2(t,y)=1-\frac{\sum_{i=1}^{N}(t_i-y_i)^2}{\sum_{i=1}^{N}(t_i-\bar{t})^2}$

N：データ数、t：教師データ、y：予測データ、$\bar{t}$：教師データの平均値

2.7　k-分割交差検証

k-分割交差検証とはモデルの汎化性能を測る手法である。

第3章　提案手法

3.1 メビウス型包除積分モデル１

包除積分はメビウス変換で重回帰形に式変形をすることができるが、そのメビウス変換の値を求めることが課題となっている。この研究ではニューラルネットワークによる重みの計算でそのメビウス変換の値を求め、包除積分を得ることでモデルの精度比較、解釈を行っていくことを目的としている。実際に作成するモデルはデータの入力層、データの前処理を行う層、包除積分の式を表す層、そして出力層の大きく分けて 4 層からなるネットワークモデルとなる。図６？は入力を 3 変数としたときのネットワークモデルである。図６？では入力層の各ユニットの説明変数x_1～x_3それぞれが前処理層のそれぞれのユニットにエッジを持ち、前処理層では活性化関数として sigmoid 関数を用いることで、各説明変数を 0～1 の間の値にすることができる。この前処理によって各説明変数を 0～1 の間にすることで t-ノルムの定義である x,y,z∈[0,K]を x,y,z∈[0,1]にし、代数積はただの積の多項演算とすることができる。包除積分を表す層では、前処理層によって 0～1 になった値を受け取り、t-ノルムの演算を行った後、活性化関数を恒等関数として出力し、その出力と重みの積を出力層に渡す。この時の重み$W_{\{1\}}$~$W_{\{1,2,3\}}$は多項演算との積、つまりメビウス変換であるmμ({1})～mμ({1,2,3})に相当するため、この重みを求めることで包除積分を得ることができる。出力層では回帰問題や分類問題によって活性化関数を変えることができ、この y の値でデータに対する予測を行う。このモデルの特徴としては、説明変数を前処理によって 0~1 の sigmoid 関数の波形として得るため、目的変数と説明変数の相関関係からある程度初期値を予測することが可能であり、適切な初期値を与えることで学習の収束にかかる時間を早めることができる。実験ではそれぞれの各変数のユニットから前処理層につながっているエッジの重みをa_m(m=1,2,…説明変数の数),バイアスをb_mとして、各説明変数データの最大値、最小値を max($\boldsymbol{x}_m$)、min($\boldsymbol{x}_m$)として、目的変数に対する説明変数$\boldsymbol{x}_m$の相関が正ならば次の式で初期値を計算する。

$$a_m = \frac{6}{\max(\boldsymbol{x}_m) - \min(\boldsymbol{x}_m)}$$

$$b_m = \frac{-6(\max(\boldsymbol{x}_m) + \min(\boldsymbol{x}_m))}{2(\max(\boldsymbol{x}_m) - \min(\boldsymbol{x}_m))}$$

目的変数に対する説明変数$\boldsymbol{x}_m$の相関が負ならば、

$$a_m = \frac{-6}{\max(\boldsymbol{x}_m) - \min(\boldsymbol{x}_m)}$$

$$b_m = \frac{6(\max(\boldsymbol{x}_m) + \min(\boldsymbol{x}_m))}{2(\max(\boldsymbol{x}_m) - \min(\boldsymbol{x}_m))}$$

とただ符号を逆転した初期値を与える。このモデルの欠点としては、説明変数が出力に対して単調でない場合、例えば説明変数に対して 2 次関数の凸となるような sigmoid 関数では

表現できない波形が最適解だった場合に高い精度を出すことができないというような点がある。そのため、データとしては目的変数に対して相関の強いものを説明変数とすることが好まれる。

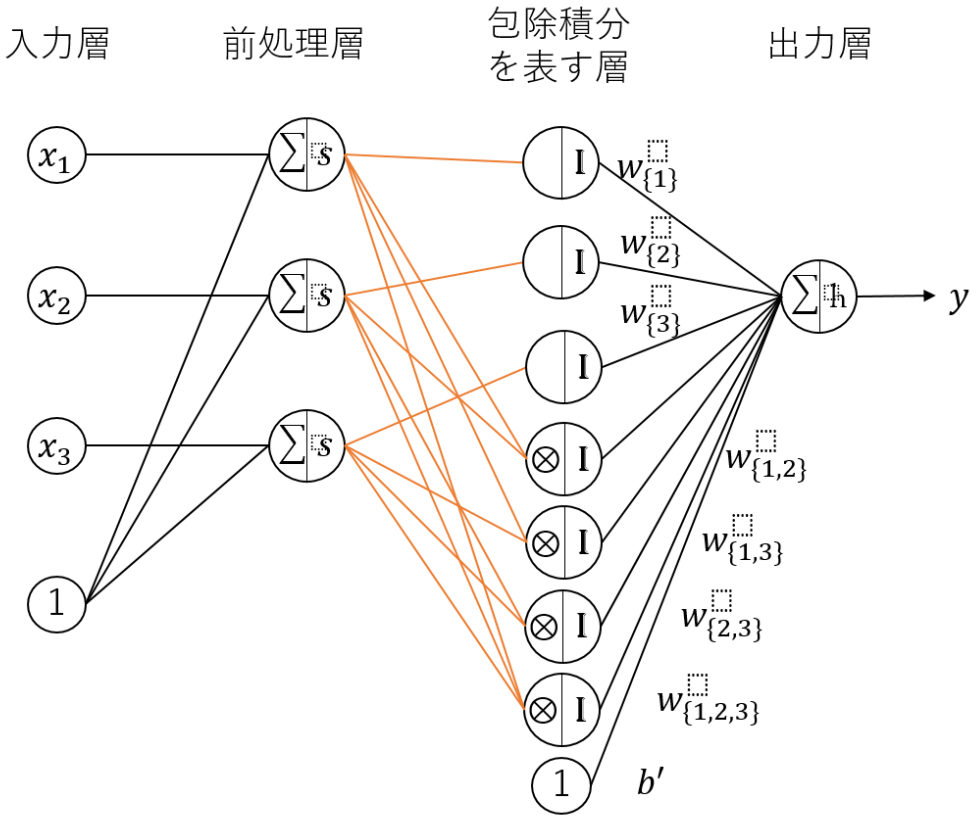


図６？

3.2 メビウス型包除積分モデル２

3.1 で述べた欠点を補うため入力層から前処理層の間に複数のユニットを用意し、各説明変数の特徴を学習するようにした。図７？は 3 入力としたときのメビウス型包除積分モデル２である。適当なユニット数とは図８？のようなネットワークを略したもので、各説明変数のデータによってどのような前処理を行えばよいかは異なるため初期値は予測が難しい。ユニット数は少なすぎると最適解を得られず、多すぎると学習に時間がかかるため適当なユニットは手動で調整する必要がある。このモデルの特徴は用意したユニットに対し適切な初期値を与えることができないため、3.1 のモデルより学習に時間がかかる。ただし、2 次関数の凸となるような波形であっても表現することができる。

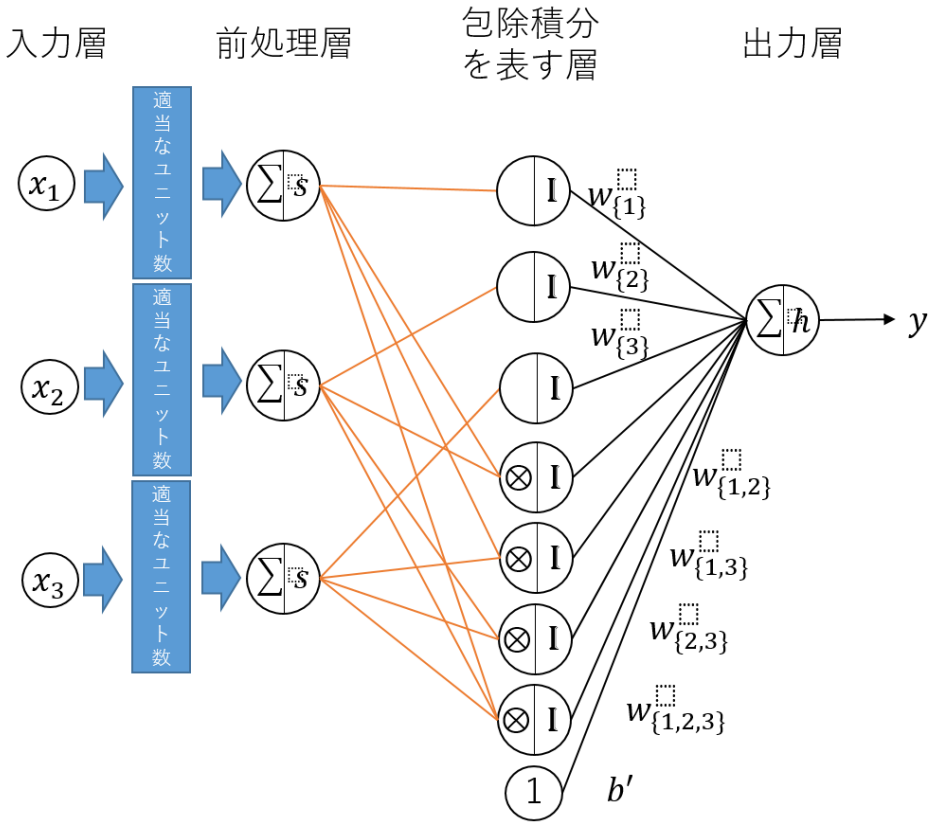

入力層
前処理層
包除積分
を表す層
出力層
適当なユニット数
適当なユニット数
適当なユニット数
x_1
x_2
x_3
s
s
s
I
I
I
I
I
I
I
$w_{\{1\}}$
$w_{\{2\}}$
$w_{\{3\}}$
$w_{\{1,2\}}$
$w_{\{1,3\}}$
$w_{\{2,3\}}$
$w_{\{1,2,3\}}$
1
b'
h
y

図７？

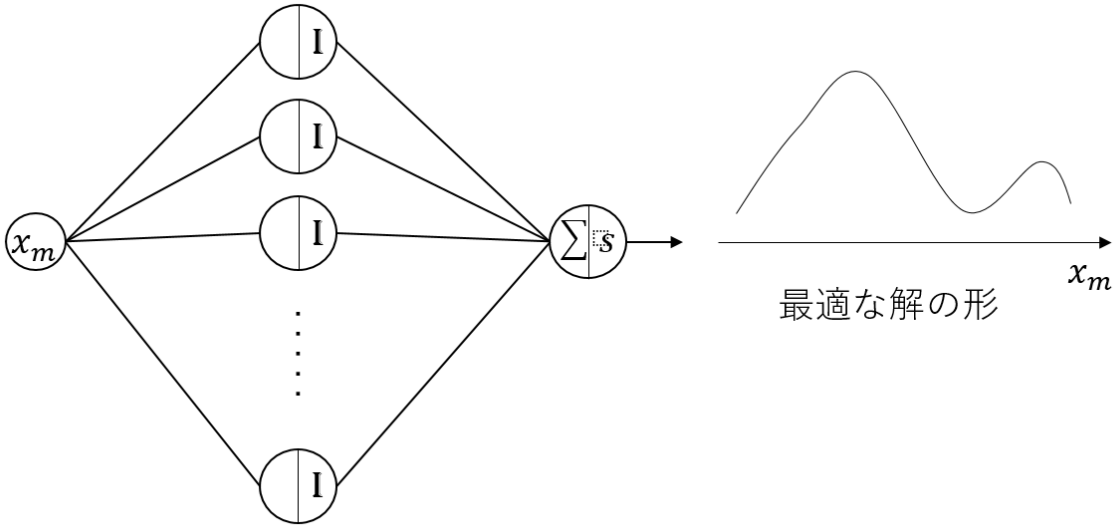

I
I
I
I
x_m
s
x_m
最適な解の形

図８？

第4章　実験

実験では提案手法の表現力と汎化性能を調べるため回帰問題を解くデータセットと分類問題を解くデータセットを用いて各機械学習手法を用いて比較を行っていく。まず表現力を調べるために各データセットの全データを学習に用いた時の精度を調べ、次に各データセットを 5 分割交差検証で汎化性能を調べる。

4.1 回帰問題に対する表現力の比較

実験 4.1 では回帰問題に対する表現力を測るためテストデータと学習データに分けず全てのデータを学習用データとし、評価方法として全てのデータに対する平均絶対誤差（Mean Absolute Error ： MAE)、平均二乗誤差（Mean Squared Error ： MSE)、決定係数（R2）の 3 つの指標で評価を行う。回帰問題として使用するデータとして Python の機械学習ライブラリである scikit-learn で提供されている「diabesets」（糖尿病）に関するデータを使用する。データラベルは表？？のようになっており、説明変数を年齢、性別、BMI 値、平均血圧、総コレステロール、悪玉、善玉、血清に関する指標の計 10 として目的変数である y の 1 年後の疾患進行度を予測するような回帰問題を解くデータとなっている。また、scikit-learn のライブラリから読み込む場合、すでに標準化されているため、https://www4.stat.ncsu.edu/~boos/var.select/diabetes.tab.txt よりオリジナルデータを読み込み表？？のようにデータの詳細を見ていく。データ数は 442 で各変数の最大値、最小値が異なり、目的変数を 0〜1 の値、説明変数も 0〜1 の値にする必要があるため正規化を行う。説明変数においては前処理層があるため正規化の必要はないが今回はデータの正規化を行っていく。正規化の式は以下のようにする。

$$x_{new} = \frac{x_{old} - \min(\boldsymbol{x})}{\max(\boldsymbol{x}) - \min(\boldsymbol{x})}$$

この時x_{new}が正規化後の説明変数の値で、x_{old}が正規化前の値、$\max(\boldsymbol{x})$、$\min(\boldsymbol{x})$はそれぞれ各説明変数データの最大値と最小値を示す。このようにすることですべての変数に対して 0〜1 の値にすることができる。データの相関関係は図？？で表され、図？？から ldl と tc、tch と ldl に高い正の相関があることが分かる。また、tch と hdl にも強い負の相関がみられた。これらの ldl, tc, tch, hdl は多重共線性となる可能性があるが今回はこれらの変数をすべて用いて実験を行う。

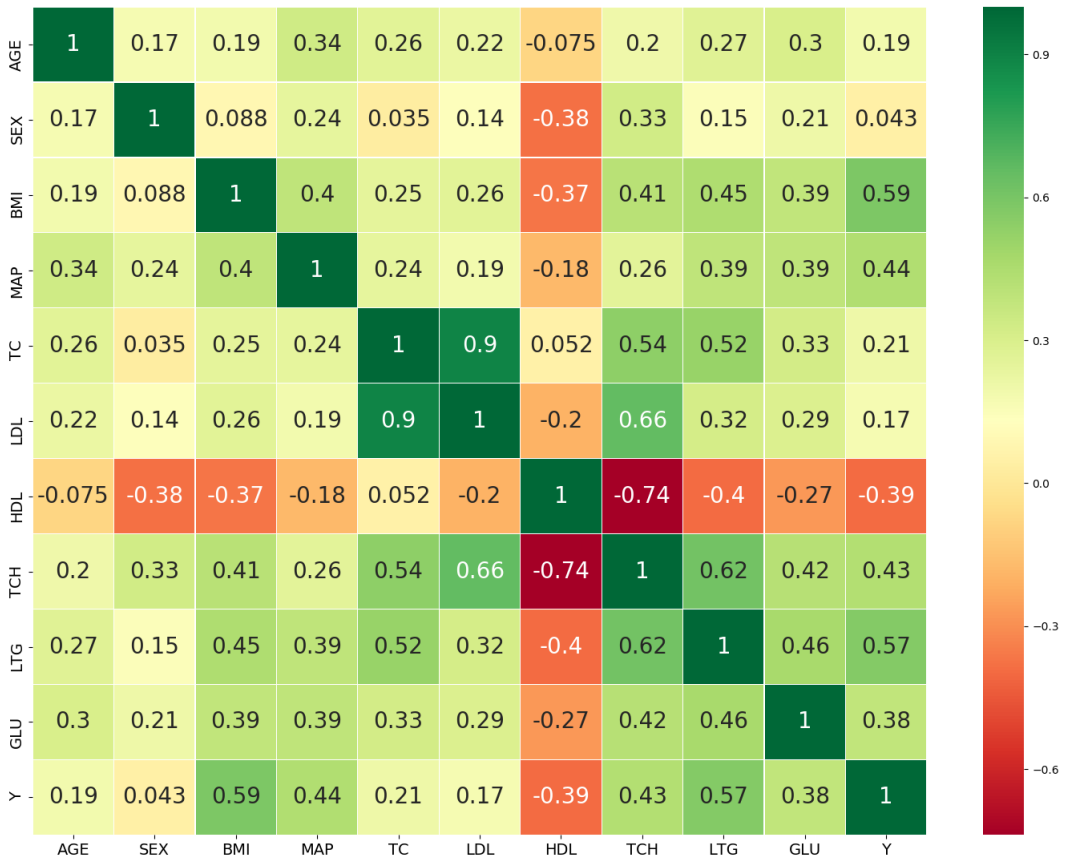
表？？：diabetes データの各変数の説明

age	年齢
sex	性別
bmi	BMI 値
map	平均血圧
tc	総コレステロール
ldl	悪玉

hdl	善玉
tch	血清に関する指標
ltg	
glu	
y	1 年後の疾患進行度

表？？：diabetes データの詳細

	count	mean	std	min	25%	50%	75%	max
AGE	442.0	48.5	13.1	19.0	38.3	50.0	59.0	79.0
SEX	442.0	1.5	0.5	1.0	1.0	1.0	2.0	2.0
BMI	442.0	26.4	4.4	18.0	23.2	25.7	29.3	42.2
MAP	442.0	94.6	13.8	62.0	84.0	93.0	105.0	133.0
TC	442.0	189.1	34.6	97.0	164.3	186.0	209.8	301.0
LDL	442.0	115.4	30.4	41.6	96.1	113.0	134.5	242.4
HDL	442.0	49.8	12.9	22.0	40.3	48.0	57.8	99.0
TCH	442.0	4.1	1.3	2.0	3.0	4.0	5.0	9.1
LTG	442.0	4.6	0.5	3.3	4.3	4.6	5.0	6.1
GLU	442.0	91.3	11.5	58.0	83.3	91.0	98.0	124.0
Y	442.0	152.1	77.1	25.0	87.0	140.5	211.5	346.0



図？？:diabetes データの変数間の相関関係図

4.1.1　重回帰式による全 diabetes データの分析

重回帰式は目的変数一つを複数の説明変数で予測するもので以下の式で表される。

$$f(x_i) = w^T x_i + b$$

$f(x_i)$:重回帰式　x_i:説明変数　w　:重みベクトル　b：バイアス

この時重みは次に示す最小 2 乗誤差を取るような重みとなる。

$$E = \sum_{i=1}^{n} (y_i - f(x_i))^2$$

E:二乗誤差関数　y_i:目的変数　n：サンプル数

表？？は重回帰式による分析結果である

。

表？？：重回帰式による diabetes データの分析結果

評価指標	値
MAE	0.1348
MSE	0.0278
R2	0.5177

考察：

重回帰式では決定係数 0.5177 となることがわかった。

4.1.2 Support Vector Machine による全 diabetes データの分析

Support Vector Machine とは教師あり学習でよく用いられるパターン認識モデルの一種で回帰分析にも用いることができ、Support Vector Regression (SVR) として scikit-learn でその学習用ライブラリが提供されている。単純な SVR モデル式は以下のようになる。

$$f(x_i) = w^T x_i + b$$

$f(x_i)$: SVR モデル　x_i:説明変数 w :重みベクトル b：バイアス

この重みとバイアスを求める式が以下の式である。

$$(w, b) = arg \min_{w,b} \frac{1}{2}\|w\|^2 + C\sum_i max[|t_i - (w^T x_i + b)| - \varepsilon, 0]$$

x_i:説明変数 w :重みベクトル b：バイアス　t:教師データ

この式の C と ε が SVR のハイパーパラメータとなり、C は値が大きいほど過学習が起きやすく、 ε は教師データと予測データの誤差の絶対値が ε 以下であるデータを除くことで頑強なモデルにしようというパラメータである。3 つ目のハイパーパラメータはカーネル関数を選択するようなハイパーパラメータで、今回は線形カーネル、多項カーネル、RBF カーネルとカーネル関数を変え、C=1、 ε =0.1 として実験を行った。表？？は SVR による分析結果である。

表？？：SVM による diabetes データの分析結果

SVMによる分析結果（C=1, ε =0.1）			
カーネル関数 / 評価指標	線形カーネル	多項カーネル	RBFカーネル
MAE	0.1348	0.1190	0.1113
MSE	0.0280	0.0227	0.0202
R2	0.5127	0.6054	0.6486

考察：

表？？より SVM では MAE、MSE、R2 のいずれの評価指標でも RBF カーネルでよい結果となった。

4.1.3 回帰木による全 diabetes データの分析

回帰木は決定木の一種で、木構造を用いることで回帰問題を解くことができる機械学習手法である。決定木では木構造が深くなればなるほど学習データに対して過学習を起こしてしまうため通常は木の深さを指定し学習を行う。表？？は木の最大の深さを変え実験を行った際の評価指標の値を示している。

表？？：回帰木による diabetes データの分析結果

回帰木							
木の深さ / 評価指標	3	4	5	7	10	15	20
MAE	0.1377	0.1251	0.1115	0.0698	0.0206	2.27E-05	0
MSE	0.0287	0.0244	0.0196	0.0104	0.002	0.0009	0
R2	0.5007	0.57561	0.6595	0.82	0.9645	0.9996	1

考察：

表？？から木の深さを深くするにつれて精度が高くなっていることがわかり、深さ 20 では誤差 0 となるほどの高い精度を出している。

4.1.4 ランダムフォレストによる全 diabetes データの分析

ランダムフォレストとは学習データの一部を重複ありで取り出し、それぞれの異なるデータで学習した複数の異なる決定木から多数決を取るような機械学習手法である。この多数決を取ることで一つの決定木では陥りやすかった過学習を防ぐことができる。表？？は決定木の数を 100 とし、木の最大の深さを指定した際の評価指標の値である。

表？？：ランダムフォレストによる diabetes データの分析結果

ランダムフォレスト(決定木の数：100)							
木の深さ / 評価指標	3	4	5	7	10	15	20
MAE	0.1298	0.1187	0.0167	0.0831	0.0607	0.0542	0.0553
MSE	0.0251	0.0208	0.1063	0.0101	0.0056	0.0045	0.0047
R2	0.5639	0.6387	0.7103	0.825	0.9034	0.9189	0.918

考察：

表？？の結果から木の深さを深くして行くほど精度が上がっているが、木の深さ 15～木の深さ 20 にかけて誤差が増えている。

4.1.5 XGBOOST による全 diabetes データの分析

XGBOOST はデータ分析を行う際によく使われる手法であり、多数の決定木を使うという点でランダムフォレストと似ているが、XGBOOST では一つの決定木で得られた誤差を小さくするような決定木をラウンド毎に追加していくという手法を取っている。この実験ではその誤差を最小化させる損失関数を線形回帰関数（ハイパラメータの引数としては reg:linear）として実験を行った。表？？は木の最大の深さを指定した際の評価指標の値である。

表？？：XGBOOST による diabetes データの分析結果

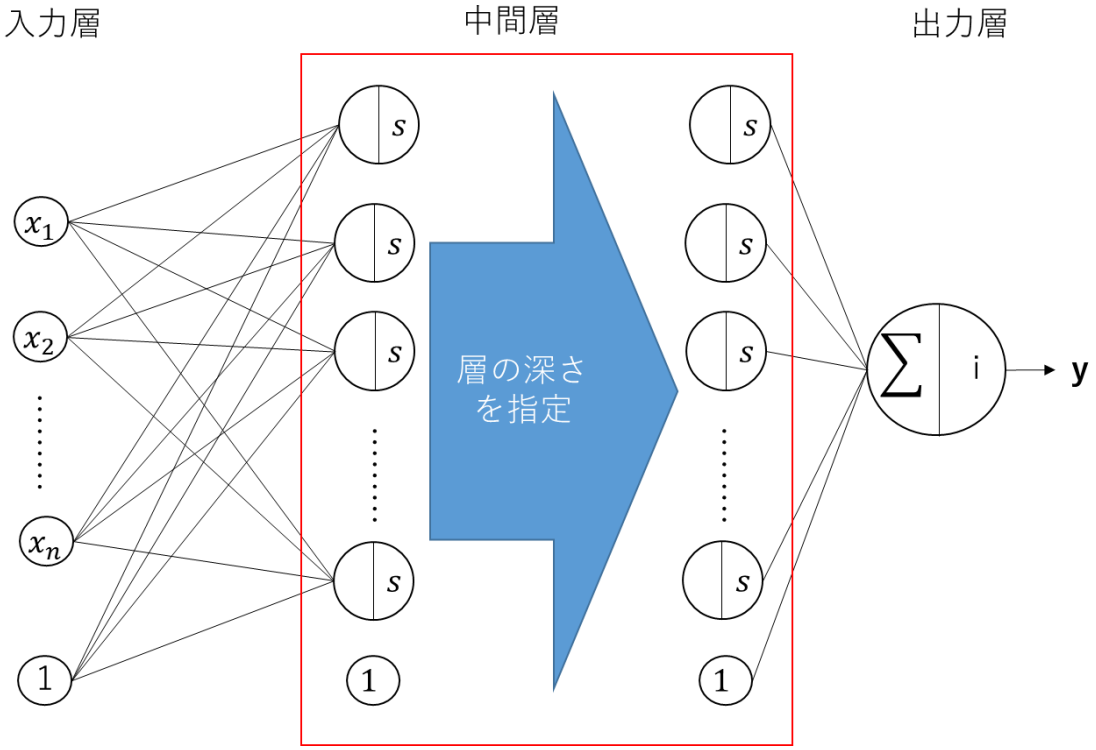
XGBOOST(決定木の最大数：100)							
木の深さ / 評価指標	1	2	3	4	5	10	20
MAE	0.1183	0.0856	0.0461	0.0169	0.0039	0.0007	0.0004
MSE	0.0217	0.0119	0.0037	0.0005	3.16E-05	9.76E-07	3.27E-07
R2	0.623	0.7935	0.9351	0.99	0.9995	0.9999	1

考察：

木の深さが深くなるにつれて精度が上がっており、木の深さが 20 の場合決定係数がほぼ 1 となった。

4.1.6 NN による全 diabetes データの分析

NN モデルは層の深さやユニット数、活性化関数や誤差関数などを変えられるため、今回の実験では次の図のように活性化関数をシグモイド関数とし、一層ごとに 1000 ユニットからなる全結合層の深さを 1～3 まで変えて実験を行った。



x :説明変数、n:説明変数の数 s:シグモイド関数、i:恒等関数、y:予測値

図？？：NN モデルのネットワーク

学習条件は表？？のようにし、この NN モデルの中間層の層の数を変えながら実験を行った結果が表？？である。

表？？：diabetes データにおける NN モデルの学習条件

学習回数	バッチサイズ	更新式	誤差関数
10000回	75	adam	平均二乗誤差

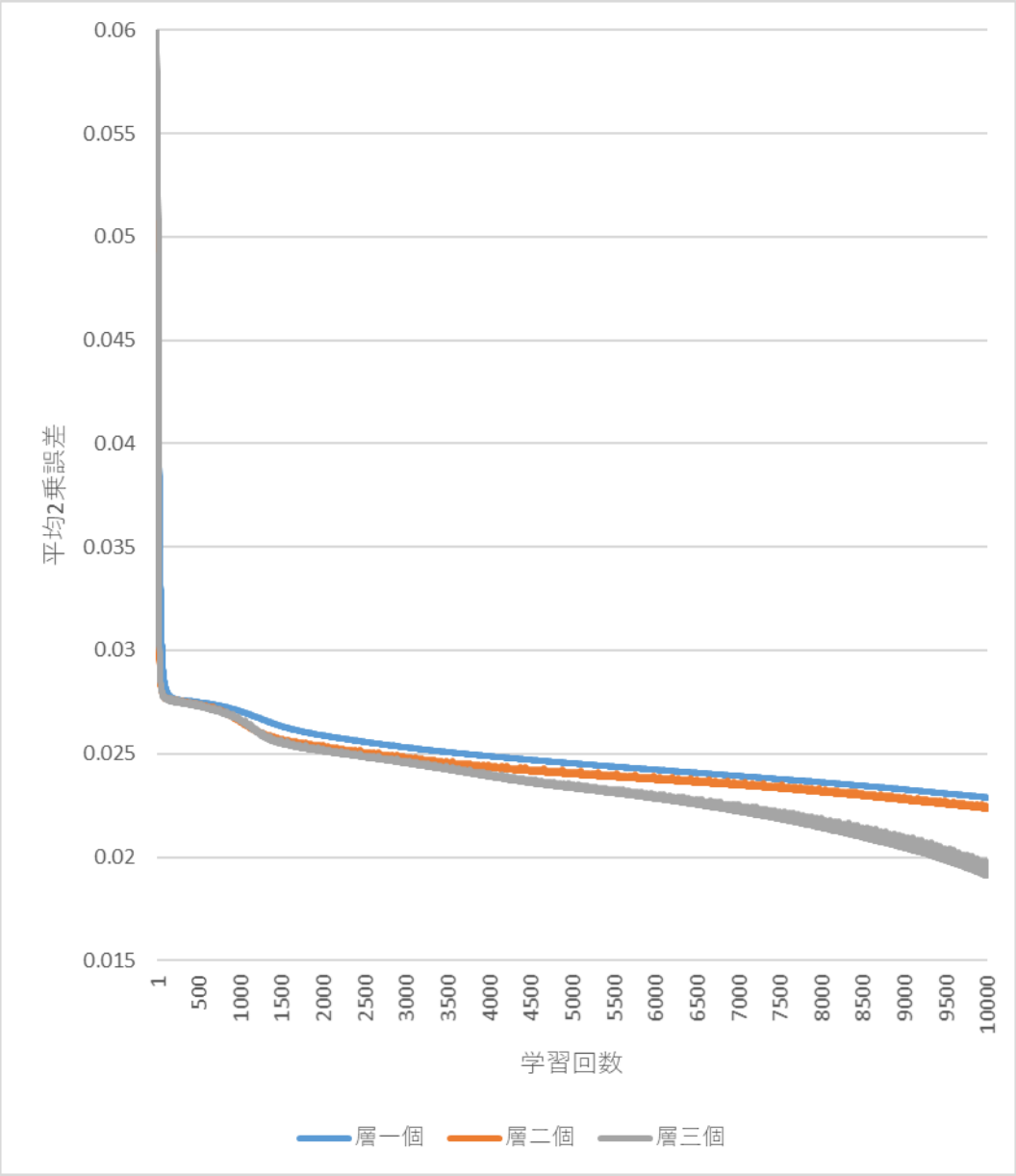
表？？：NN による diabetes データの分析結果

評価指標	層一個	層二個	層三個
MAE	0.1199	0.1184	0.1101
MSE	0.0229	0.0223	0.0192
R2	0.6024	0.6118	0.6672

表？？と図？？は NN モデルの層の数を変えて実験を行った際の学習における平均二乗誤差の学習曲線を図示し、具体的な数値を示したものである。

表？？：NN による diabetes データの学習過程の平均二乗誤差

学習回数	層一個	層二個	層三個
1	0.0593	0.0569	0.0601
500	0.0275	0.0274	0.0273
1000	0.0271	0.0266	0.0266
1500	0.0263	0.0257	0.0256
2000	0.0259	0.0253	0.0251
2500	0.0256	0.0250	0.0249
3000	0.0253	0.0248	0.0245
3500	0.0251	0.0245	0.0242
4000	0.0249	0.0243	0.0239
4500	0.0247	0.0242	0.0236
5000	0.0245	0.0240	0.0233
5500	0.0244	0.0239	0.0232
6000	0.0242	0.0238	0.0229
6500	0.0241	0.0237	0.0226
7000	0.0239	0.0235	0.0222
7500	0.0238	0.0234	0.0222
8000	0.0236	0.0232	0.0217
8500	0.0234	0.0230	0.0211
9000	0.0233	0.0227	0.0204
9500	0.0231	0.0225	0.0199
10000	0.0229	0.0223	0.0192



図？？：NN による diabetes データの学習過程の平均二乗誤差

考察：

表？？の結果から層が増えるほど平均 2 乗誤差、平均絶対誤差、決定係数はよくなり精度が高くなっていることがわかる。

4.1.7 メビウス型包除積分モデル 1 による全 diabetes データの分析

提案手法の一つであるメビウス型包除積分モデル 1 は前処理にシグモイド関数を用いて適切な初期値を与えることでランダムに初期値を与えた場合より比較的早く学習を行うことができるのが特徴である。表？？は学習条件を示している。

表？？：diabetes データにおけるメビウス型包除積分モデル 1 の学習条件

学習回数	バッチサイズ	更新式	誤差関数	t-norm
10000回	75	adam	平均二乗誤差	代数積

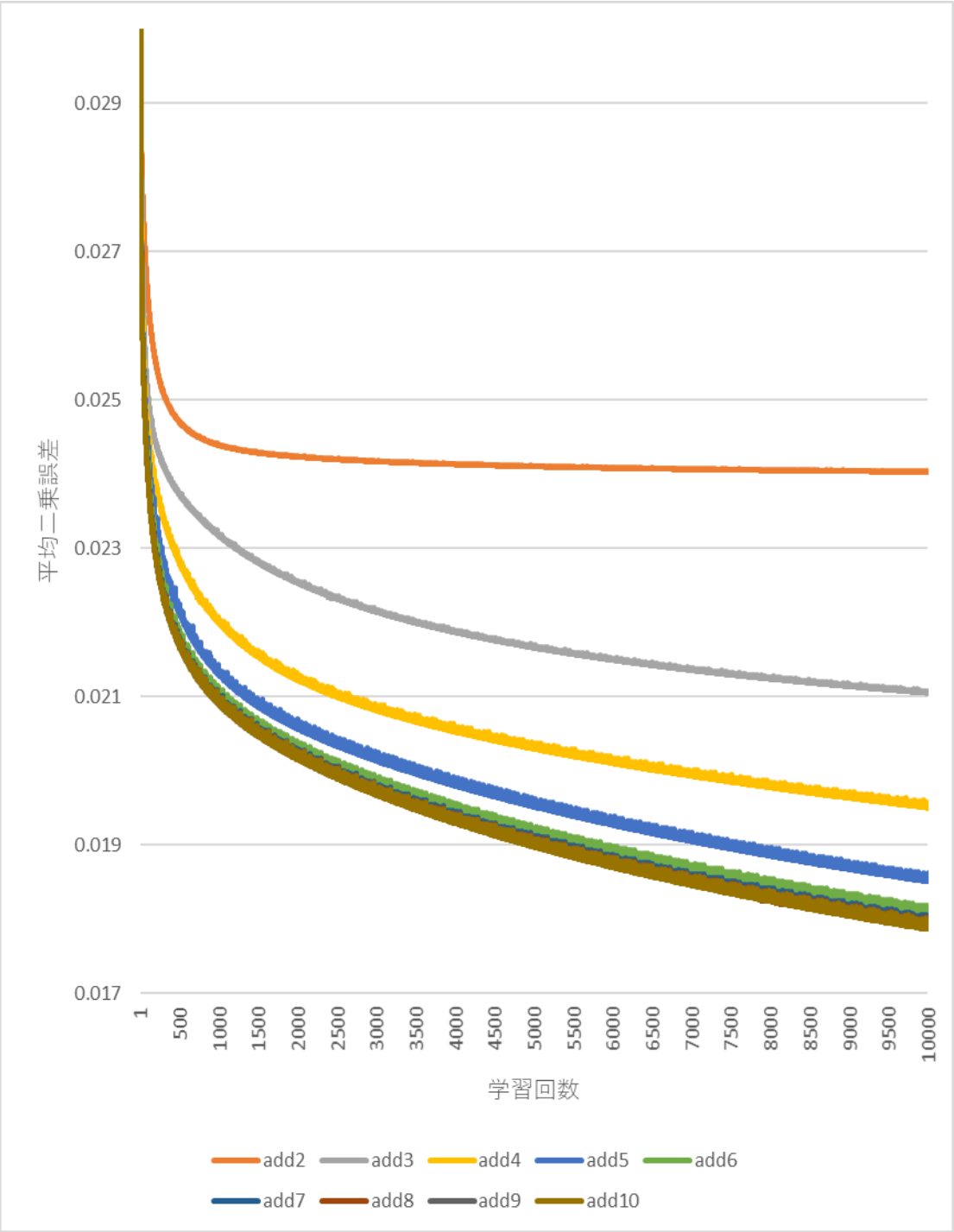
また、提案手法では説明変数が 10 個の時は包除積分を表す層に存在するユニット数は2^{10} ＝1024となる。これはすべての代数積を含むユニット数であるためこの代数積のユニットを減らすことによってよりスパースなモデルにすることができる。具体的には全 10 加法（$x_1 \otimes x_2 \sim x_1 \otimes x_2 \otimes x_3 \otimes x_4 \otimes x_5 \otimes x_6 \otimes x_7 \otimes x_8 \otimes x_9 \otimes x_{10}$まで）のところを 2 加法（$x_1 \otimes x_2 \sim x_5 \otimes x_6$まで）から 9 加法（$x_1 \otimes x_2 \sim x_2 \otimes x_3 \otimes x_4 \otimes x_5 \otimes x_6 \otimes x_7 \otimes x_8 \otimes x_9 \otimes x_{10}$まで）に制限を加えることでユニット数を減らす。表？？は 2～全加法までに加法性を制限した場合の各学習回数に応じた平均 2 乗誤差を示しており、表？？は学習終了時に全データを入力した際の評価指標の値を示している。図？？は 2 加法から全加法までの学習曲線を図示したものである。

表？？：メビウス型包除積分モデル 1 による diabetes データの分析結果

制限 / 評価指標	add2	add3	add4	add5	add6	add7	add8	add9	add10
MAE	0.1229	0.1142	0.1089	0.1051	0.1035	0.1028	0.1027	0.1026	0.1026
MSE	0.0240	0.0211	0.0196	0.0185	0.0181	0.0179	0.0179	0.0179	0.0179
R2	0.5824	0.6340	0.6600	0.6781	0.6860	0.6885	0.6892	0.6894	0.6894

表？？：メビウス型包除積分モデル 1 による diabetes データの学習過程の平均二乗誤差

制限 / 学習回数	add2	add3	add4	add5	add6	add7	add8	add9	add10
1	0.0407	0.0388	0.0376	0.0378	0.0382	0.0384	0.0385	0.0385	0.0385
500	0.0247	0.0237	0.0228	0.0222	0.0219	0.0218	0.0218	0.0218	0.0218
1000	0.0244	0.0232	0.0220	0.0213	0.0211	0.0210	0.0210	0.0210	0.0210
1500	0.0243	0.0228	0.0215	0.0209	0.0206	0.0205	0.0205	0.0205	0.0205
2000	0.0242	0.0225	0.0212	0.0206	0.0203	0.0202	0.0202	0.0202	0.0202
2500	0.0242	0.0223	0.0210	0.0203	0.0200	0.0199	0.0199	0.0199	0.0199
3000	0.0242	0.0221	0.0208	0.0202	0.0198	0.0197	0.0197	0.0197	0.0197
3500	0.0241	0.0220	0.0207	0.0200	0.0196	0.0195	0.0195	0.0195	0.0195
4000	0.0241	0.0219	0.0205	0.0198	0.0195	0.0194	0.0193	0.0193	0.0193
4500	0.0241	0.0218	0.0204	0.0197	0.0193	0.0192	0.0192	0.0191	0.0191
5000	0.0241	0.0217	0.0203	0.0196	0.0192	0.0190	0.0190	0.0190	0.0190
5500	0.0241	0.0216	0.0202	0.0194	0.0190	0.0189	0.0189	0.0188	0.0188
6000	0.0241	0.0215	0.0201	0.0194	0.0190	0.0189	0.0188	0.0188	0.0188
6500	0.0241	0.0214	0.0200	0.0192	0.0188	0.0186	0.0186	0.0186	0.0186
7000	0.0241	0.0214	0.0199	0.0191	0.0186	0.0185	0.0185	0.0185	0.0185
7500	0.0241	0.0213	0.0199	0.0190	0.0186	0.0184	0.0184	0.0184	0.0184
8000	0.0241	0.0213	0.0198	0.0189	0.0185	0.0183	0.0183	0.0183	0.0183
8500	0.0240	0.0212	0.0197	0.0188	0.0183	0.0182	0.0182	0.0182	0.0182
9000	0.0240	0.0212	0.0197	0.0187	0.0183	0.0181	0.0181	0.0181	0.0181
9500	0.0240	0.0211	0.0196	0.0186	0.0182	0.0180	0.0180	0.0180	0.0180
10000	0.0240	0.0211	0.0196	0.0185	0.0181	0.0179	0.0179	0.0179	0.0179



図？？メビウス型包除積分モデル 1 による diabetes データの学習過程の平均二乗誤差

考察

メビウス型包除積分モデルの加法性を制限することによって表？？のように 2 加法～全加法にかけて平均 2 乗誤差は減少しており、精度が高くなっていることがわかる。また、9 加法に制限した場合は制限を付けない場合とほぼ同様の結果となった。

4.1.8 メビウス型包除積分モデル 2 による全 diabetes データの分析

次にメビウス型包除積分モデル２で同様の実験を行うが、加法性の制限をせず全加法のみで実験を行う。学習条件は？？とし、前処理に使用するユニット数を 1000 ユニット、層は 1 層だけ用意し学習を行う。図？？はその学習の様子を示し、表？？は実際の数値を示している。

表？？：diabetes データにおけるメビウス型包除積分モデル 2 の学習条件

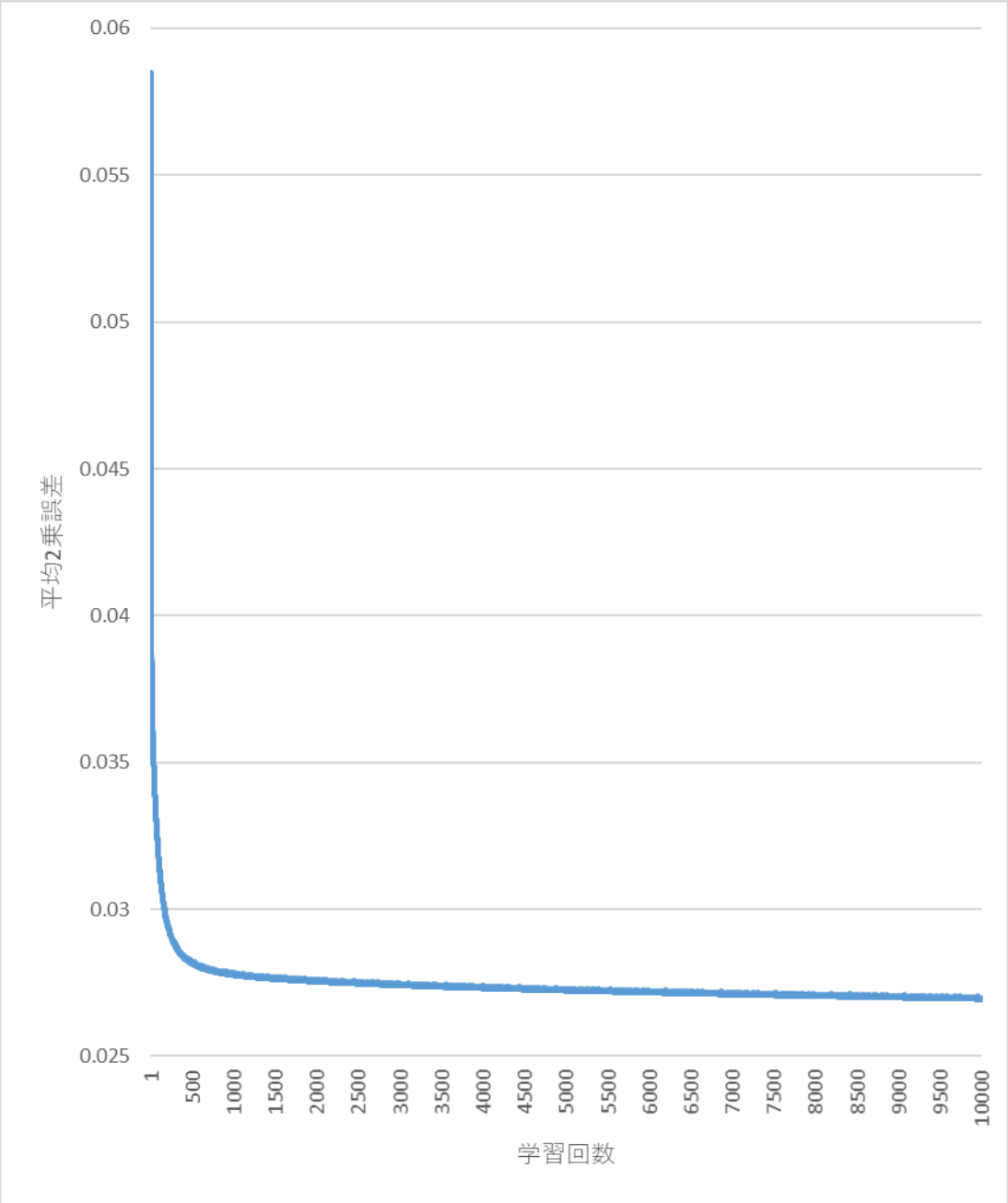
学習回数	バッチサイズ	更新式	誤差関数	t-norm	適当なユニット数	層の深さ
10000回	75	adam	平均二乗誤差	代数積	1000	1

表？？：メビウス型包除積分モデル 2 による diabetes データの分析結果

評価指標	値
MAE	0.1333
MSE	0.0269
R2	0.5317

表？？：メビウス型包除積分モデル 2 による diabetes データの学習過程の平均二乗誤差

学習回数	平均2乗誤差
1	0.0585
500	0.0282
1000	0.0278
1500	0.0277
2000	0.0275
2500	0.0275
3000	0.0274
3500	0.0274
4000	0.0273
4500	0.0273
5000	0.0272
5500	0.0272
6000	0.0272
6500	0.0272
7000	0.0271
7500	0.0271
8000	0.0271
8500	0.0270
9000	0.0270
9500	0.0270
10000	0.0269



図？？：メビウス型包除積分モデル 2 による diabetes データの学習過程の平均二乗誤差

考察：

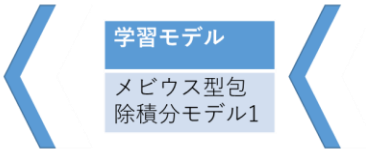
学習回数 500 までは誤差を大きく下げているがそれ以降はほとんど下がっておらずゆっくりと学習が行われている。

4.1.9 実験 4.1 全体の考察

実験 4.1 全体としては回帰木、ランダムフォレスト、XGBOOST の精度が高く木の深さを深くするほど高い精度が得られることがわかり、同様に NN モデルでも層の深さが精度に大き

く影響していることがわかった。提案手法については、メビウス型包除積分モデル 1 に比べメビウス型包除積分モデル 2 の精度が低くなっていることがわかった。これは前処理層による学習がうまくいっておらず適切な波形を得ることができていないためと思われる。提案手法の表現力について決定係数の値を表現力として見るとメビウス型包除積分モデル 1（全加法）の大小関係は次の図のように表すことができる。

学習モデル	学習条件
重回帰式	最小二乗誤差
SVM	RBFカーネル関数
回帰木	木の深さ5
ランダムフォレスト	木の最大の深さ4
XGBOOST	木の最大の深さ1
NNモデル	3層に指定した場合



学習モデル	学習条件
回帰木	木の深さ6
ランダムフォレスト	木の最大の深さ5
XGBOOST	木の最大の深さ2

図？？：決定係数を基準とした提案手法の diabetes データにおける大小関係

メビウス型包除積分モデル 1 の表現力は学習回数によってはここから上下すると思われるが回帰問題においておおよそこの図程度の表現力を有していると考えられる。

4.2 回帰問題に対する汎化性能の比較

実験 4.2 では実験 4.1 で測れなかったモデルの汎化性能について K 分割交差検証によって比較を行っていく。実験に使用するデータは実験 4.1 同様に diabesets データとし、5 分割交差検証で汎化性能の比較を行う。提案手法と比較を行うため実験 4.1 で用いた機械学習手法でも 5 分割交差検証を行う。モデルにより過学習が起きているかどうかを見るため、5 分割交差検証で分割した学習データとテストデータのそれぞれのデータで評価指標の値を計算し、5 つそれぞれの学習データとテストデータの評価指標の値の平均を取る。評価指標は実験 4.1 同様に MAE、MSE、R2 とする。

4.2.1 重回帰式による diabetes データに対する汎化性能

重回帰式の汎化性能を評価指標により評価していく。表？？は重回帰式による 5 分割交差検証の結果をまとめたものである。

表？？：重回帰式の diabetest データに対する 5 分割交差検証で得られた分析結果

学習データに対する評価指標の平均値

評価指標	値
MAE	0.1341
MSE	0.0275
R2	0.5212

テストデータに対する評価指標の平均値

評価指標	値
MAE	0.1401
MSE	0.0298
R2	0.4728

考察：

表より学習データとテストデータでは精度が異なることがわかる。

4.2.2 SVM による diabetes データに対する汎化性能

SVM の汎化性能を評価指標により評価していく。実験条件は C=1、ε=0.1 としカーネル関数を変えて評価指標の値を計算する。表？？は SVM による 5 分割交差検証の結果をまとめたものである。

表？？：SVM の diabetest データに対する 5 分割交差検証で得られた分析結果

学習データに対する評価指標の平均値

SVMによる分析結果（C=1, ε=0.1）			
カーネル関数 / 評価指標	線形カーネル	多項カーネル	RBFカーネル
MAE	0.1345	0.1166	0.1101
MSE	0.0279	0.0212	0.0194
R2	0.5142	0.6315	0.6627

テストデータに対する評価指標の平均値

SVMによる分析結果（C=1, ε=0.1）			
カーネル関数 / 評価指標	線形カーネル	多項カーネル	RBFカーネル
MAE	0.1384	0.1584	0.1433
MSE	0.0292	0.0416	0.0333
R2	0.4899	0.2731	0.4194

考察：

表より学習データに対する精度は RBF カーネル関数を使った場合が最もよくどの指標においてもよい結果となり実験 4.1 で得られた結果と同様の傾向となっている。しかし、テストデータに対しては線形カーネル関数を用いた場合の精度が最もよくどの指標でもよい値を示している。また、学習データとテストデータの評価指標で差分を取った時に線形カーネル関数が最も小さくなり 3 つの関数の中でも最も汎化性能が高いことがわかる。逆に多項カーネル関数は差分が大きく汎化性能が低いことがわかる。

4.2.3 回帰木による diabetes データに対する汎化性能

回帰木は木の深さを 1, 2, 3, 4, 5, 10, 20 と変えて汎化性能を評価する。表？？は回帰木による汎化性能の結果をまとめたものである。

表？？：回帰木の diabetest データに対する 5 分割交差検証で得られた分析結果

学習データに対する評価指標の平均値

回帰木							
木の深さ / 評価指標	1	2	3	4	5	10	20
MAE	0.1665	0.1442	0.1347	0.1225	0.1077	0.0203	0
MSE	0.0405	0.0320	0.0278	0.0235	0.0190	0.0018	0
R2	0.2964	0.4424	0.5164	0.5909	0.6696	0.9686	1

テストデータに対する評価指標の平均値

回帰木							
木の深さ / 評価指標	1	2	3	4	5	10	20
MAE	0.1726	0.1559	0.1558	0.1580	0.1637	0.1941	0.1969
MSE	0.0442	0.0374	0.0363	0.0416	0.0437	0.0591	0.0655
R2	0.2239	0.3233	0.3630	0.2717	0.2347	-0.0529	-0.1757

考察：

回帰木では学習データに対しては木の深さを深くするにつれ精度を上げることができているがテストデータに対しては木の深さ 3 以降は深くなるにつれ精度が低下しており過学習が発生している。

4.2.4　ランダムフォレストによる diabetes データに対する汎化性能

ランダムフォレストの学習条件は決定木の数を 100 とし、木の深さを 1, 2, 3, 4, 5, 10, 20 と変えて汎化性能を評価する。表？？はランダムフォレストによる 5 分割交差検証の結果をまとめたものである。

表？？：ランダムフォレストの diabetest データに対する 5 分割交差検証で得られた分析結果

学習データに対する評価指標の平均値

ランダムフォレスト（決定木の数：100）							
木の深さ / 評価指標	1	2	3	4	5	10	20
MAE	0.1564	0.1389	0.1266	0.1148	0.1019	0.0592	0.0539
MSE	0.0349	0.0284	0.0237	0.0194	0.0153	0.0053	0.0045
R2	0.3925	0.5063	0.5873	0.6619	0.7344	0.9085	0.9222

テストデータに対する評価指標の平均値

ランダムフォレスト（決定木の数：100）							
木の深さ / 評価指標	1	2	3	4	5	10	20
MAE	0.1629	0.1506	0.1470	0.1459	0.1465	0.1476	0.1484
MSE	0.0378	0.0337	0.0325	0.0317	0.0324	0.0337	0.0343
R2	0.3363	0.4043	0.4311	0.4401	0.4185	0.3987	0.3952

考察：

表？？より木の深さが 4 より深くなるとテストデータに対する評価指標の精度が下がり木の深さ 5 以上は過学習が発生していることがわかる。よって汎化性能の高さは木の深さが 4 の時最も高くなると考えられる。

4.2.5　XGBOOST による diabetes データに対する汎化性能

XGBOOST の学習条件は決定木の最大数を 100 とし、木の最大の深さを 1,2,3,4,5,10,20 と変えて汎化性能を評価する。ただし、XGBOOST の学習では過学習を抑えるためのハイパーパラメータである early_stopping_rounds という値を 10 に設定し学習を行った。表？？は XGBOOST による 5 分割交差検証の結果をまとめたものである。

表？？：XGBOOST による diabetest データに対する 5 分割交差検証で得られた分析結果

学習データに対する評価指標の平均値

XGBOOST（決定木の最大数：100）							
木の深さ / 評価指標	1	2	3	4	5	10	20
MAE	0.1234	0.1080	0.0906	0.0689	0.0513	0.0266	0.0253
MSE	0.0233	0.0181	0.0129	0.0078	0.0044	0.0012	0.0011
R2	0.5952	0.6853	0.7758	0.8656	0.9228	0.9790	0.9805

テストデータに対する評価指標の平均値

XGBOOST（決定木の最大数：100）							
木の深さ / 評価指標	1	2	3	4	5	10	20
MAE	0.1434	0.1422	0.1442	0.1521	0.1528	0.1596	0.1572
MSE	0.0310	0.0309	0.0320	0.0359	0.0362	0.0398	0.0390
R2	0.4586	0.4452	0.4410	0.3707	0.3502	0.2892	0.3074

考察：

XGBOOST は木の深さが 1 より深くすることでテストデータに対する評価指標の精度が下がっており木の深さを 2 以上で過学習が発生していることがわかる。よって汎化性能の高さは木の深さが 1 の時最も高くなると考えられる。

4.2.6　NN による diabetes データに対する汎化性能

NN の学習に使用するモデルは実験 4.1.5 同様に図？？のように活性化関数をシグモイド関数とし、一層ごとに 1000 ユニットからなる全結合層の深さを 1～3 まで変えて実験を行った。学習条件は実験 4.1.5 と同様表？？とする。ただし、過学習を防止するため学習毎にテストデータに対する平均 2 乗誤差が学習以前の平均 2 乗誤差の最小値より大きい場合をカウントし、50 カウントその状態が続いた場合学習を止める。表？？は NN による汎化性能の結果をまとめたものである。

表？？：NN の diabetest データに対する 5 分割交差検証で得られた分析結果

学習データに対する評価指標の平均値

評価指標	層一個	層二個	層三個
MAE	0.1325	0.1307	0.1317
MSE	0.0272	0.0266	0.0269
R2	0.5273	0.5368	0.5326

テストデータに対する評価指標の平均値

評価指標	層一個	層二個	層三個
MAE	0.1360	0.1358	0.1365
MSE	0.0286	0.0283	0.0287
R2	0.4959	0.4992	0.4967

考察：

過学習を抑えたため学習データに対する評価指標は層を変えたとしても同程度の結果を得ることができた。また、テストデータに対する評価指標は学習データに対する評価指標と同様に中間層を 2 層にした場合に最もよい結果となった。

4.2.7　メビウス型包除積分モデル 1 の汎化性能

提案手法では実験 4.1.6 同様に学習条件として表？？を用い、過学習を防ぐため 4.2.5 の NN で行った学習毎にテストデータに対する平均 2 乗誤差が学習以前の平均 2 乗誤差の最小値より大きい場合をカウントし、50 カウントその状態が続いた場合学習を止める手法をとる。使用する提案手法は全データを用いた学習で比較的精度が良かったメビウス型包除積分モデル 1 とし学習を行う。表？？は加法性に制限を付けた時の評価指標の値の平均の値である。

表？？：メビウス型包除積分モデル 1 の diabetest データに対する 5 分割交差検証で得ら

れた分析結果

学習データに対する評価指標の平均値

制限 / 評価指標	add2	add3	add4	add5	add6	add7	add8	add9	add10
MAE	0.1316	0.1261	0.1260	0.1233	0.1241	0.1251	0.1247	0.1234	0.1235
MSE	0.0265	0.0246	0.0247	0.0239	0.0241	0.0244	0.0243	0.0239	0.0239
R2	0.5392	0.5721	0.5718	0.5839	0.5816	0.5763	0.5774	0.5840	0.5840

テストデータに対する評価指標の平均値

制限 / 評価指標	add2	add3	add4	add5	add6	add7	add8	add9	add10
MAE	0.1361	0.1356	0.1356	0.1369	0.1360	0.1387	0.1383	0.1376	0.1380
MSE	0.0285	0.0287	0.0286	0.0290	0.0290	0.0296	0.0296	0.0291	0.0294
R2	0.5008	0.5018	0.4968	0.4926	0.4931	0.4861	0.4819	0.4858	0.4816

考察：

学習データでは 2 加法まで制限した場合に精度が最も低く、9～10 に制限した場合に精度が高くなっているが、テストデータでは 2～4 加法まで制限した場合に高い精度となっており 5～10 加法まで制限した場合には過学習の傾向がみられた。

4.2.8　実験 4.2 全体の考察

各学習モデルの中でもテストデータに対する決定係数の値が最も高くなったモデルで順位付けを行うと表？？のようになった。

表？？：決定係数による汎化性能比較

rank	モデル	決定係数
1	メビウス型包除積分モデル 1:制限 3	0.5018
2	NN:層数 2	0.4992
3	SVM:線形カーネル	0.4899
4	重回帰式：最小二乗誤差	0.4728
5	XGBOOST：木の最大の深さ 1	0.4586
6	ランダムフォレスト：木の最大の深さ 4	0.4401
7	回帰木：木の深さ	0.363

表？？から決定木を用いた手法である回帰木、ランダムフォレスト、XGBOOST は決定係数が低くなる傾向がみられた。決定木を用いた手法の中では XGBOOST が最も高く回帰木は最も低くなるという結果となった。提案手法であるメビウス型包除積分モデル 1 は決定係数における汎化性能が最も高いモデルとなり、全結合からなる NN より高い汎化性能が得られた。重回帰式や SVM は決定木を用いた手法より高い汎化性能を持ち NN の手法よりは低い値となっている。このことから決定木は回帰分析に対する汎化性能が低く重回帰式や SVM、NN など

は回帰分析に対する汎化性能が高いことがわかり、その中でもメビウス型包除積分モデル 1 は最も汎化性能に優れていると考えられる。

4.3　回帰問題における XGBOOST と提案手法の解釈比較

実験 4.3 では XGBOOST の学習モデルから得られる gain 値と shap 値を比較対象とし、提案手法の学習モデルからシャープレイ値＋拡張したシャープレイ値を算出し比較を行っていく。また、全データを学習データとして使ったときの重要度と 5 分割交差検証で得られた重要度がどのように異なるのかを調べるためその二パターンで分けて実験を行った。

4.3.1　diabetes データの全データを学習データとして得られた XGBOOST の解釈

次の表？？は全データを学習データとして用いた重回帰式の係数の値（重み）を示している。

表？？：

説明変数	係数の値
AGE	-0.0068
SEX	-0.0712
BMI	0.4224
MAP	0.2470
TC	-0.6927
LDL	0.4669
HDL	0.0892
TCH	0.1443
LTG	0.6078
GLU	0.0576

考察：

係数の値の大小を重回帰式に対する貢献度だと考えると LTG は正の値で最大となり重回帰式の中でも最も正に貢献していると考えられる。逆に TC は負の値で、係数の中でも最小となっているため重回帰式の中でも最も負に貢献していると考えられる。

4.3.2　diabetes データの全データを学習データとして得られた XGBOOST の解釈

次の表？？と表？？は全データを学習データとして用いた XGBOOST の学習モデルの gain 値と shap 値を示している。

表？？：diabetes データの全データを学習データとして得られる gain 値

XGBOOST（決定木の最大数：100）							
木の深さ / 説明変数	1	2	3	4	5	10	20
AGE	0.058	0.044	0.025	0.012	0.011	0.007	0.007
SEX	0.063	0.059	0.031	0.027	0.027	0.007	0.006
BMI	0.549	0.214	0.104	0.055	0.045	0.046	0.036
MAP	0.266	0.087	0.048	0.023	0.021	0.013	0.012
TC	0.068	0.027	0.025	0.016	0.013	0.010	0.009
LDL	0.034	0.033	0.028	0.016	0.012	0.008	0.007
HDL	0.175	0.047	0.030	0.011	0.013	0.010	0.009
TCH	0.025	0.108	0.036	0.017	0.014	0.007	0.007
LTG	1.020	0.442	0.189	0.094	0.101	0.074	0.066
GLU	0.069	0.053	0.026	0.017	0.016	0.011	0.009

表？？：diabetes データの全データを学習データとして得られる shap 値

XGBOOST（決定木の最大数：100）							
木の深さ / 説明変数	1	2	3	4	5	10	20
AGE	0.013	0.019	0.022	0.019	0.020	0.018	0.019
SEX	0.023	0.028	0.026	0.023	0.023	0.015	0.014
BMI	0.069	0.066	0.067	0.071	0.065	0.080	0.077
MAP	0.031	0.031	0.036	0.033	0.031	0.029	0.029
TC	0.003	0.014	0.018	0.018	0.019	0.013	0.014
LDL	0.013	0.016	0.021	0.018	0.016	0.012	0.012
HDL	0.024	0.022	0.021	0.019	0.020	0.021	0.017
TCH	0.007	0.010	0.012	0.011	0.009	0.007	0.009
LTG	0.075	0.090	0.095	0.095	0.106	0.091	0.094
GLU	0.020	0.026	0.024	0.027	0.023	0.022	0.020

考察:

表？？より gain 値では木の深さによって全体的な値が小さくなっていく傾向がみられ、LTG、BMI、MAP の値は他の説明変数の gain 値より比較的高くなり XGBOOST の学習モデルにおいて重要視されていることがわかる。shap 値では木の深さによって値が小さくなる傾向は見られず、gain 値と同様に LTG、BMI、MAP の値が他の説明変数の gain 値より比較的高くなっていることがわかる。よって XGBOOST の学習モデルから LTG、BMI、MAP が重要な変数であると考えられる。

4.3.3　diabetes データの全データを学習データとして得られた提案手法の解釈

次に提案手法の中でも精度の良いメビウス型包除積分モデル 1 で学習を行った学習モデルからシャープレイ値を求める。表？？は得られたモデルのシャープレイ値を示しており、次の表？？では加法性を制限した中で計算できる拡張したシャープレイ値による大小関係をランキングとしており、表内の[ ]の中の数字は表？？の説明変数を数字で置き換えたものである。

表？？：説明変数の割り当て

1	2	3	4	5	6	7	8	9	10
AGE	SEX	BMI	MAP	TC	LDL	HDL	TCH	LTG	GLU

表？？: diabetes データの全データを学習データとして得られるシャープレイ値

説明変数＼制限	add2	add3	add4	add5	add6	add7	add8	add9	add10
AGE	0.046	0.007	-0.057	-0.132	-0.171	-0.147	-0.068	0.005	0.028
SEX	-0.129	-0.092	-0.114	-0.213	-0.347	-0.441	-0.468	-0.438	-0.416
BMI	0.369	0.169	0.044	-0.032	-0.024	0.065	0.173	0.247	0.270
MAP	0.333	0.183	-0.161	-0.476	-0.635	-0.649	-0.563	-0.474	-0.446
TC	-0.512	-0.779	-0.703	-0.687	-0.846	-1.037	-1.107	-1.063	-1.029
LDL	0.262	0.236	-0.127	-0.507	-0.697	-0.720	-0.675	-0.621	-0.599
HDL	0.076	0.123	0.021	-0.465	-1.274	-1.952	-2.173	-2.105	-2.057
TCH	-0.045	-0.347	-0.601	-0.716	-0.680	-0.556	-0.446	-0.383	-0.362
LTG	0.397	0.663	0.802	0.814	0.926	1.115	1.269	1.360	1.390
GLU	0.332	0.354	0.286	0.195	0.065	0.046	0.143	0.235	0.266

表？？: diabetes データの全データを学習データとして得られる拡張したシャープレイ値のランキング

rank＼制限	add2	add3	add4	add5	add6	add7	add8	add9	add10
1	[8,10]	[3,6,9]	[3,6,9]	[3,6,9]	[3,6,9]	[1,3,6,9]	[1,3,6,9]	[1,3,6,9]	[1,3,6,9]
2	[2,5]	[3,5,9]	[6,9]	[3,9,10]	[3,9,10]	[3,6,9]	[3,9,10]	[3,9,10]	[3,9,10]
3	[6,9]	[8,10]	[6,9,10]	[6,9,10]	[1,3,6,9]	[3,9,10]	[3,6,9]	[3,6,9]	[3,6,9]
4	[3,4]	[3,7,8]	[1,3,6]	[6,9]	[6,9,10]	[3,6,9,10]	[3,6,9,10]	[1,2,3,9,10]	[1,2,3,9,10]
5	[7,8]	[6,9]	[4,5,7]	[1,3,6,9]	[3,6,9,10]	[6,9,10]	[1,2,3,9,10]	[3,6,9,10]	[3,6,9,10]
6	[9]	[1,3,6]	[1,5,6]	[4,5,7]	[6,9]	[3,5,9,10]	[3,5,9,10]	[3,5,9,10]	[3,5,9,10]
7	[3]	[4,6,8]	[3,9,10]	[9,10]	[4,9,10]	[4,9,10]	[4,9,10]	[4,9,10]	[4,9,10]
8	[4]	[3,9]	[3,5,9]	[3,6,9,10]	[9,10]	[9,10]	[6,9,10]	[6,9,10]	[6,9,10]
9	[10]	[7,8]	[9,10]	[4,9,10]	[4,5,7]	[1,2,3,9,10]	[1,2,3,6,9]	[1,2,3,6,9]	[1,2,3,6,9]
10	[1,9]	[8,9,10]	[3,9]	[5,8,10]	[3,5,9,10]	[6,9]	[9,10]	[9,10]	[3,5,8,9,10]

また、シャープレイ値と拡張したシャープレイ値は負の値を取ってしまいランク付けを行った場合負の値を低く評価してしまうため、この値の絶対値を取ることで負に貢献したと

とらえることができる。表？？はシャープレイ値の絶対値を重要度として示しており、表？？は拡張したシャープレイ値の絶対値のランキングを示している。

表？？: diabetes データの全データを学習データとして得られる重要度

制限 / 説明変数	add2	add3	add4	add5	add6	add7	add8	add9	add10
AGE	0.046	0.007	0.057	0.132	0.171	0.147	0.068	0.005	0.028
SEX	0.129	0.092	0.114	0.213	0.347	0.441	0.468	0.438	0.416
BMI	0.369	0.169	0.044	0.032	0.024	0.065	0.173	0.247	0.270
MAP	0.333	0.183	0.161	0.476	0.635	0.649	0.563	0.474	0.446
TC	0.512	0.779	0.703	0.687	0.846	1.037	1.107	1.063	1.029
LDL	0.262	0.236	0.127	0.507	0.697	0.720	0.675	0.621	0.599
HDL	0.076	0.123	0.021	0.465	1.274	1.952	2.173	2.105	2.057
TCH	0.045	0.347	0.601	0.716	0.680	0.556	0.446	0.383	0.362
LTG	0.397	0.663	0.802	0.814	0.926	1.115	1.269	1.360	1.390
GLU	0.332	0.354	0.286	0.195	0.065	0.046	0.143	0.235	0.266

表？？: diabetes データの全データを学習データとして得られる拡張した重要度のランキング

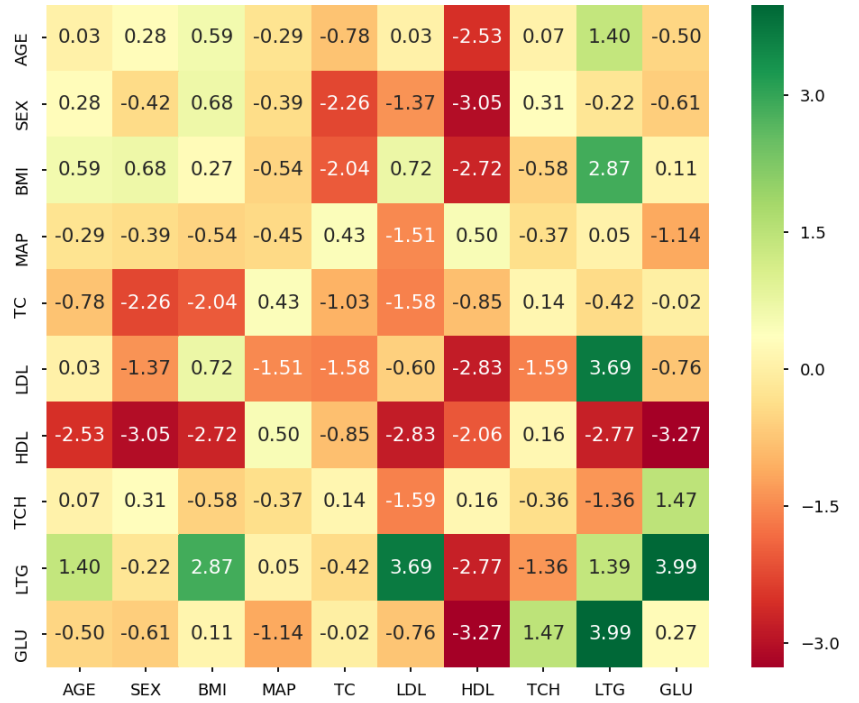
制限 / rank	add2	add3	add4	add5	add6	add7	add8	add9	add10
1	[8,10]	[3,5,8]	[3,6,9]	[3,6,9]	[3,6,9]	[1,3,6,9]	[1,3,6,9]	[1,3,6,9]	[1,3,6,9]
2	[2,6]	[3,4,8]	[3,5,8]	[3,9,10]	[3,9,10]	[3,6,9]	[3,9,10]	[3,9,10]	[3,9,10]
3	[2,5]	[3,6,8]	[6,9]	[3,7,9]	[1,3,6,9]	[3,9,10]	[3,6,9]	[3,6,9]	[3,6,9]
4	[6,9]	[3,6,9]	[3,4,8]	[3,5,6,8]	[2,5,6,8]	[2,5,6]	[2,5,6]	[2,5,6]	[2,5,6]
5	[5]	[3,5]	[3,7,9]	[6,9,10]	[2,5,6]	[2,5,6,8]	[2,5,6,8]	[2,5,6,8]	[2,5,6,8]
6	[3,4]	[5,8]	[3,6,8]	[6,9]	[3,5,6,7]	[3,5,6,7]	[2,5,6,10]	[2,5,6,10]	[1,2,3,9,10]
7	[3,5]	[3,5,9]	[3,5]	[1,3,6,9]	[6,9,10]	[2,5,6,10]	[3,5,6,7]	[3,5,6,7]	[2,5,6,10]
8	[7,10]	[3,8]	[6,9,10]	[3,6,8]	[3,6,9,10]	[3,6,9,10]	[3,6,9,10]	[1,2,3,9,10]	[3,6,9,10]
9	[7,8]	[8,10]	[3,5,6]	[4,5,7]	[3,7,9]	[6,9,10]	[1,2,3,9,10	[3,6,9,10]	[3,5,6,7]
10	[5,8]	[3,7,8]	[1,3,6]	[2,5,6,8]	[6,9]	[3,5,9,10]	[3,5,9,10]	[3,5,9,10]	[3,5,9,10]

考察：

拡張したシャープレイ値は説明変数の数だけ指数関数的に増加するため、2 変数以下の拡張したシャープレイ値を示した図（2 加法まで制限した場合と全加法の場合のみ）と表？？と表？？から考察を行っていく。

add2

add10



図？？：diabetes データの全データを学習データとして得られるモデルの拡張したシャープレイ値

図？？は 2 変数間の拡張したシャープレイ値を示しており AGE,AGE などの対角線上はシャ

ープレイ値の値を取るような図となっている。シャープレイ値と拡張したシャープレイ値は表より制限を 2 加法～全加法にかけて条件を緩くすることでその最大最小値の幅が大きくなる傾向がみられた。表？？からシャープレイ値の特徴的な変数としては正に最も貢献しているLTG、負に最も貢献しているHDL、重要度の最も低いAGEがあげられる。図？？の全加法からその変数の 2 変数間の関係を順番に見ていくと LTG は GLU,LDL などの変数と正の関係になるがHDLと負の関係を持つことがわかる。HDLはMAPやTCHを除く変数と負の関係を多く持っていた。AGEは重要度が低いもののHDLと高い負の関係であり、LTGとも高い正の関係であることがわかった。

4.3.4　diabetesデータを5分割交差検証して得られた重回帰の解釈

全データを学習して得られた解釈と 5 分割交差検証で汎化性能を高めることで得られる解釈が異なるかどうかを見るためまず重回帰式で 5 分割交差検証を行った際の係数の値を求める。ただし、5分割交差検証で得られるモデルは5つなので5つのモデルから得られた係数の値を平均したものを表？？に示している。

表？？：diabetesデータを5分割交差検証して得られる重回帰式の係数の値

説明変数	係数の値
AGE	-0.0066
SEX	-0.0709
BMI	0.4207
MAP	0.2473
TC	-0.6957
LDL	0.4683
HDL	0.0904
TCH	0.1449
LTG	0.6085
GLU	0.0562

考察：

表？？より 5 つの重回帰式はLTGの係数を正の最大とし、TCを負の最小とする傾向がみられた。

4.3.5　diabetesデータを5分割交差検証して得られたXGBOOSTの解釈

XGBOOSTで5分割交差検証を行った際のgain値とshap値を計算する。ただし、5分割交差検証で得られるモデルは5つなので5つのモデルから得られたgain値とshap値の平均したものを表？？と表？？に示している。

表？？：diabetes データを 5 分割交差検証して得られる gain 値

XGBOOST（決定木の最大数：100）							
木の深さ / 説明変数	1	2	3	4	5	10	20
AGE	0.054	0.078	0.060	0.040	0.032	0.012	0.011
SEX	0.077	0.095	0.071	0.047	0.041	0.015	0.019
BMI	1.051	0.674	0.450	0.275	0.190	0.070	0.068
MAP	0.338	0.242	0.140	0.076	0.054	0.026	0.023
TC	0.044	0.068	0.068	0.041	0.035	0.013	0.013
LDL	0.057	0.069	0.072	0.054	0.048	0.017	0.014
HDL	0.184	0.152	0.113	0.052	0.035	0.013	0.012
TCH	0.014	0.076	0.090	0.072	0.053	0.023	0.016
LTG	1.237	0.859	0.435	0.376	0.202	0.100	0.091
GLU	0.148	0.143	0.097	0.060	0.045	0.022	0.017

表？？：diabetes データを 5 分割交差検証して得られる shap 値

XGBOOST（決定木の最大数：100）							
木の深さ / 説明変数	1	2	3	4	5	10	20
AGE	0.008	0.012	0.014	0.015	0.015	0.016	0.016
SEX	0.016	0.021	0.018	0.017	0.019	0.015	0.015
BMI	0.070	0.069	0.075	0.073	0.073	0.072	0.074
MAP	0.028	0.033	0.029	0.030	0.031	0.030	0.030
TC	0.006	0.008	0.009	0.011	0.011	0.011	0.012
LDL	0.005	0.007	0.014	0.011	0.012	0.011	0.010
HDL	0.020	0.021	0.020	0.021	0.017	0.017	0.018
TCH	0.001	0.004	0.006	0.009	0.008	0.009	0.008
LTG	0.077	0.081	0.085	0.083	0.086	0.091	0.091
GLU	0.016	0.019	0.018	0.020	0.019	0.021	0.019

考察：

表？？より gain 値では木の深さによって全体的な値が小さくなっていく傾向がみられる。LTG、BMI、MAP の値は他の説明変数の gain 値や shap 値より比較的高くなり XGBOOST の学習モデルにおいて重要視されていることがわかる。よって XGBOOST の学習モデルから LTG、BMI、MAP が重要な変数であると考えられる。

4.3.4　5 分割交差検証で得られた提案手法の解釈

メビウス型包除積分モデル 1 で同様に 5 分割交差検証を行った際に得られた 5 つの学習モデルからシャープレイ値を求め平均をとったものを表？？に示す。拡張したシャープレイ値も平均をとりランキングとして表？？に示す。

表？？：diabetes データを 5 分割交差検証して得られるシャープレイ値

制限 / 説明変数	add2	add3	add4	add5	add6	add7	add8	add9	add10
AGE	0.042	0.048	0.054	0.024	-0.032	-0.134	-0.149	-0.222	-0.204
SEX	-0.047	0.006	0.067	0.111	0.079	0.053	0.039	0.001	-0.009
BMI	0.340	0.305	0.228	0.160	0.058	-0.044	-0.062	-0.151	-0.128
MAP	0.221	0.257	0.263	0.272	0.193	0.091	0.078	0.030	0.019
TC	-0.186	-0.259	-0.439	-0.477	-0.620	-0.781	-0.718	-0.873	-0.827
LDL	0.050	0.104	0.230	0.152	0.111	0.021	-0.012	-0.021	0.029
HDL	0.092	0.083	0.059	0.020	-0.068	-0.163	-0.184	-0.231	-0.229
TCH	0.091	0.043	-0.028	-0.084	-0.167	-0.250	-0.256	-0.328	-0.323
LTG	0.358	0.418	0.498	0.471	0.476	0.383	0.309	0.356	0.334
GLU	0.083	0.131	0.184	0.199	0.176	0.121	0.088	0.096	0.041

表？？： diabetes データを 5 分割交差検証して得られる拡張したシャープレイ値のランキング

制限 / rank	add2	add3	add4	add5	add6	add7	add8	add9	add10
1	[9]	[9]	[9]	[9]	[9]	[9]	[9]	[6,9,10]	[9]
2	[3]	[3]	[6,9]	[6,9]	[6,9,10]	[9,10]	[6,9,10]	[9]	[6,9,10]
3	[4]	[4]	[4]	[9,10]	[9,10]	[6,9,10]	[9,10]	[9,10]	[6,9]
4	[3,4]	[3,4]	[1,9]	[1,2]	[6,9]	[6,9]	[1,2]	[6,9]	[9,10]
5	[7]	[6,9]	[6]	[4]	[1,2]	[2,9,10]	[6,9]	[2,9,10]	[2,6,9,10]
6	[8]	[1,9]	[3]	[6,9,10]	[6,8,9,10]	[1,2]	[1,2,9]	[6,8,9,10]	[2,9,10]
7	[1,9]	[10]	[9,10]	[1,9]	[1,2,9]	[3,6,9,10]	[2,9,10]	[6,7,9,10]	[2,6,9]
8	[8,10]	[1,2]	[3,4]	[1,2,9]	[2,9,10]	[1,2,3]	[1,2,3]	[2,6,9,10]	[6,8,9,10]
9	[10]	[9,10]	[10]	[1,2,3]	[1,9]	[2,6,9,10]	[2,6,9,10]	[3,6,9,10]	[6,7,9,10]
10	[2,3]	[8,10]	[6,9,10]	[10]	[1,2,3]	[6,7,9,10]	[2,4]	[1,2]	[1,2,9]

5 分割交差検証で得られた 5 つの学習モデルそれぞれのシャープレイ値の絶対値を平均したものを重要度として表？？に示す。同様に拡張したシャープレイ値の絶対値を平均し重要度としてランキングにしたものを表？？に示す。

表？？：diabetes データを 5 分割交差検証して得られる重要度

説明変数＼制限	add2	add3	add4	add5	add6	add7	add8	add9	add10
AGE	0.042	0.051	0.062	0.037	0.055	0.138	0.161	0.222	0.214
SEX	0.047	0.021	0.067	0.111	0.104	0.140	0.106	0.064	0.135
BMI	0.340	0.305	0.228	0.160	0.078	0.118	0.200	0.151	0.165
MAP	0.221	0.257	0.263	0.272	0.193	0.151	0.106	0.115	0.111
TC	0.186	0.259	0.439	0.477	0.620	0.781	0.718	0.873	0.827
LDL	0.100	0.104	0.230	0.152	0.125	0.101	0.131	0.066	0.114
HDL	0.092	0.097	0.068	0.107	0.113	0.256	0.233	0.231	0.238
TCH	0.091	0.047	0.037	0.084	0.167	0.250	0.256	0.328	0.323
LTG	0.358	0.418	0.498	0.471	0.476	0.383	0.309	0.356	0.334
GLU	0.083	0.131	0.184	0.199	0.176	0.121	0.092	0.105	0.071

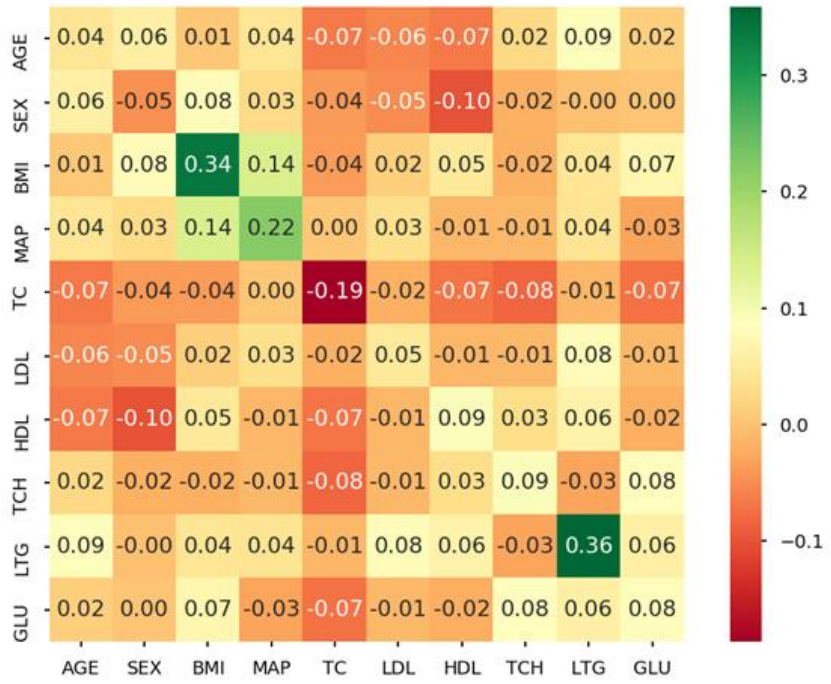
表？？： diabetes データを 5 分割交差検証して得られる拡張した重要度のランキング

rank＼制限	add2	add3	add4	add5	add6	add7	add8	add9	add10
1	[9]	[9]	[9]	[5]	[3,5,8]	[3,5,8]	[3,5,8]	[3,5,8]	[3,5,8]
2	[3]	[3]	[5]	[9]	[5]	[5]	[5]	[3,5]	[5]
3	[4]	[5]	[5,8]	[3,5,8]	[3,5]	[3,5]	[5,8]	[5]	[3,5]
4	[5]	[4]	[6,9]	[5,8]	[5,8]	[5,8]	[3,5]	[5,8]	[5,8]
5	[3,4]	[3,4]	[3,5,8]	[3,5]	[3,8]	[3,5,6,8]	[3,8]	[3,8]	[3,8]
6	[2,7]	[5,8]	[4]	[3,8]	[9]	[3,8]	[3,5,6,8]	[3,5,6,8]	[5,7]
7	[6]	[2,7]	[3,8]	[5,7]	[3,5,6,8]	[3,5,6]	[5,7]	[3,5,6]	[3,5,7]
8	[7]	[6,9]	[5,7]	[6,9]	[3,6,8]	[3,5,7]	[3,5,7]	[3,5,7]	[3,5,6,8]
9	[8]	[1,9]	[3,5]	[9,10]	[3,5,6]	[5,6,8]	[3,4,5,8]	[1,3,5,8]	[3,4,5,8]
10	[1,9]	[1,5]	[1,2]	[1,2]	[5,7]	[5,7]	[3,5,6]	[3,4,5,8]	[1,3,5,8]

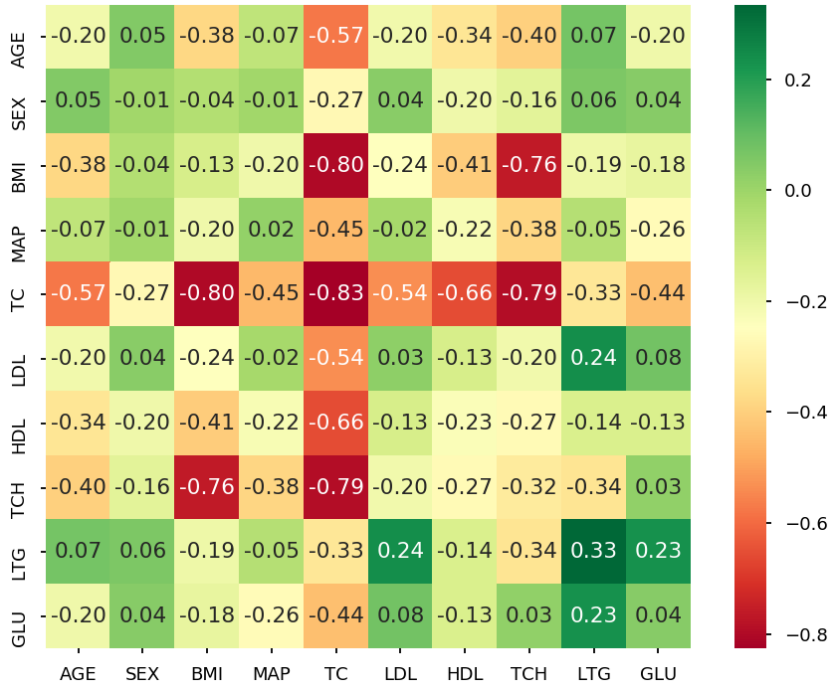
考察：

実験 4.3.3 同様に 2 変数以下の拡張したシャープレイ値を示した図？？（2 加法まで制限した場合と全加法の場合のみ）と表？？と表？？から考察を行っていく。

add2



Add10



図？？：diabetes データの全データを学習データとして得られるモデルの拡張したシャープレイ値

表？？と表？？よりシャープレイ値の正に貢献する傾向が LTG にみられ、負に貢献する傾向が TC にみられた。2 変数間の関係を見る図？？から、2 加法の場合は BMI の貢献度が高

くなっているが全加法ではそれほど大きくない値となっており、BMI や TC,TCH などには強い負の貢献となる組み合わせが存在していることがわかった。

4.3.5　実験 4.3 全体からの解釈

・重回帰についての解釈

全データの場合の係数、5 分割交差検証時の係数の平均を比べてもあまり変化がなく重回帰モデルでは LDL の係数を正に大きくとり TC を負に大きくとる傾向がみられたことから LDL（悪玉）の値を下げ、TC（総コレステロール）の値を上げることで 1 年後の疾患進行度を減少させることができると解釈ができる。

・XGBOOST についての解釈

2 つの解釈手法の全データの場合、5 分割交差検証時の平均を比べてもあまり変化がみられなかったが gain では浅い木に比べ深い木の gain 値は小さな値を示した。解釈手法によっては各変数の大小関係が異なったが BMI や LTG はどちらの指標でも比較的高い値を取った。以上より XGBOOST では BMI（BMI 値）と LTG(血清に関する指標)の二つがより 1 年後の疾患進行度に大きく影響を与えているという解釈ができる。

・提案手法についての解釈

全データの場合、5 分割交差検証時の平均を比べると、全データで学習を行ったときの重要度の値は 5 分割交差検証時と比べ大きくなる傾向がみられ、値の大小関係も異なるものが多かった。また、加法性の制限によって TC や LTG 等の変数は順位などに大きな変化はなかったが AGE、SEX、BMI などのシャープレイ値は正や負になるなどの増減をしていた。汎化性能の高い 5 分割交差検証時の拡張したシャープレイ値を見ていくと正に貢献しているものに LDL（悪玉）、LTG（血清に関する指標）、GLU（血清に関する指標）などの変数が多くみられた。負に貢献しているものには BMI（BMI 値）、TC(総コレステロール)、TCH（血清に関する指標）などの変数が多くみられた。以上より提案手法では LDL、LTG、GLU（特に LTG）の値を減らし、BMI、TC、TCH（特に TC）の値を上げることで 1 年後の疾患進行度をより減少させることができると解釈ができる。他にも

4.4　二値分類問題に対する表現力の比較

実験 4.4 では二値分類問題に対する表現力を見るためすべてのデータを学習データとして使って学習を行う。回帰分析と異なり二値分類問題となるため評価指標として精度（正答率）、適合率、再現率、f-1 値、AUC、MAE、MSE の計 7 の指標で評価を行う。使用するデータは機械学習用データセットが提供されているサイト kaggle（https://www.kaggle.com/）で提供されている Titanic データセットで、diabetes データと異なり二値分類を行うデータとなっている。このデータは映画にもなったタイタニック号と呼ばれる豪華客船が沈没した事故の乗客の生存を予測するデータとなっており、乗客の様々なデータから生存できた

かどうかを予測するデータとなっている。データの各変数名と変数の値（最初の 5 人分のデータ）は表？？のようになっており、目的変数を Survived として残りの変数を説明変数とするデータである。

表？？：Titanic データセットの先頭 5 件

PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
1	0	3	Braund, M	male	22	1	0	A/5 21171	7.25		S
2	1	1	Cumings,	female	38	1	0	PC 17599	71.2833	C85	C
3	1	3	Heikkinen	female	26	0	0	STON/O2.	7.925		S
4	1	1	Futrelle, M	female	35	1	0	113803	53.1	C123	S
5	0	3	Allen, Mr.	male	35	0	0	373450	8.05		S

表？？：各変数の欠損値の数

変数名	PassengerId	Survived	Pclass	Name	Sex	Age
欠損値の数	0	0	0	0	0	177

変数名	SibSp	Parch	Ticket	Fare	Cabin	Embarked
欠損値の数	0	0	0	0	687	2

しかし、表？？と表？？のように文字列や欠損データなどが含まれているため何らかの処理を施す必要がある。そのためまず予測に不要な PassengerId、欠損値の多い Age と Cabin、情報の読み取りづらい Ticket の 4 つを削除した。Name は文字列だが敬称を含み判別するための指標とすることができ、“title_group”という変数名でまとめなおす。表？？は敬称別に整数値を振ったものである。

表？？：敬称別数値の割り当て

敬称名	値
Mme,Ms,Lady,Sir,Mile,Aff	5
Mrs,Miss,Master	4
Major,Col,Dr	3
Mrs,Miss,Master	2
Don,Rev,Capt	1

Sex は male を 1、female を 0 とした。Embarked は “S”となるとき 1 とし、それ以外の“Q”、“C”、欠損値をすべて 0 として、“seaport”という変数名でまとめた。SibSp と Parch は値がともに 0 となる場合 1、それ以外の場合を 0 とすることで “alone”という 1 つの変数にまとめる。Pclass と Fare はそれぞれ“pclass”、“fare”という変数名に書き換える。これまでの処理をまとめて変数名を変えたものが表？？になり、この処理を行ったデータで学習を行う。またこのように処理を行うことで相関関係は図？？で表すことができる。

表？？：処理を行った Titanic データの詳細

	count	mean	std	min	25%	50%	75%	max
title_grupe	891	2.809	1.003	1	2	2	4	5
sex	891	0.648	0.478	0	0	1	1	1
alone	891	0.603	0.490	0	0	1	1	1
guest_class	891	2.309	0.836	1	2	3	3	3
fare	891	32.20	49.69	0	7.91	14.45	31	512.3
seaport	891	0.723	0.448	0	0	1	1	1
Survived	891	0.384	0.487	0	0	0	1	1

図？？：処理を行った Titanic データの相関関係

4.4.1 重回帰分類による全 Titanic データの分析

あああ

表？？：

評価指標	値
精度	0.7980
適合率	0.7368
再現率	0.7368
f-1値	0.7368
AUC	0.8651
MAE	0.2820
MSE	0.1398

4.4.2 Support Vector Machine による全 Titanic データの分析

SVM では分類問題を解くため Support Vector Classification (SVC) として scikit-learn でその学習用ライブラリが提供されている。SVC でもハイパーパラメータは C=1 とし、カーネル関数を変えて学習を行う。表？？は実験を行った際の評価指標の値を示している。

表？？：SVM による Titanic データの分析結果

SVMによる分析結果（C=1）			
カーネル関数 / 評価指標	線形カーネル	多項カーネル	RBFカーネル
精度	0.7946	0.6734	0.6824
適合率	0.7202	0.9180	0.7122
再現率	0.7602	0.1637	0.2895
f-1値	0.7397	0.2779	0.4116
AUC	0.7881	0.5773	0.6083
MAE	0.2054	0.3266	0.3176
MSE	0.2054	0.3266	0.3176

4.4.3 分類木による全 Titanic データの分析

分類木は決定木の一種で、木構造を用いることで分類問題を解くことができる機械学習手法である。表？？は木の深さを変え実験を行った際の評価指標の値を示している。

表？？：分類木による Titanic データの分析結果

分類木							
木の深さ / 評価指標	1	2	3	4	5	10	20
精度	0.7912	0.7980	0.8395	0.8462	0.8586	0.9091	0.9226
適合率	0.7131	0.9175	0.8043	0.8060	0.8600	0.9117	0.9333
再現率	0.7632	0.5205	0.7690	0.7890	0.7544	0.8450	0.8596
f-1値	0.7373	0.6642	0.7862	0.7862	0.8037	0.8771	0.8950
AUC	0.7860	0.7457	0.8262	0.8355	0.8389	0.8970	0.9107
MAE	0.2088	0.2020	0.1605	0.1538	0.1414	0.0909	0.0774
MSE	0.2088	0.2020	0.1605	0.1538	0.1414	0.0909	0.0774

考察：

表？？より木が深くなるにつれ評価指標の値はよくなっており木の深さ 20 で精度（正答率）も 92%と高い。

4.4.4 ランダムフォレストによる全 Titanic データの分析

ランダムフォレストも分類木を用いることで同様に分類を行っていくことができる。表？？は木の最大の深さを変え実験を行った際の評価指標の値を示している。

表？？：ランダムフォレストによる Titanic データの分析結果

ランダムフォレスト（決定木の数：100）							
木の深さ / 評価指標	1	2	3	4	5	10	20
精度	0.7868	0.8002	0.8328	0.8485	0.8552	0.9158	0.9226
適合率	0.7420	0.7485	0.8227	0.8581	0.8790	0.9238	0.9200
再現率	0.6813	0.7222	0.7193	0.7251	0.7222	0.8509	0.8743
f-1値	0.7104	0.7351	0.7676	0.7861	0.7929	0.8858	0.8966
AUC	0.7669	0.7855	0.8114	0.8252	0.8301	0.9036	0.9135
MAE	0.2132	0.1998	0.1672	0.1515	0.1448	0.0842	0.0774
MSE	0.2132	0.1998	0.1672	0.1515	0.1448	0.0842	0.0774

考察：

表？？より木が深くなるにつれ評価指標の値はよくなっており木の深さ 20 で精度（正答率）も 92%と高い。

4.4.5 XGBOOST による全 Titanic データの分析

XGBOOST も分類木を用いることで同様に分類を行っていくことができる。XGBOOST では一つの決定木で得られた誤差を小さくするような決定木をラウンド毎に追加していくとい

う手法を取っており、実験 4.4 からその誤差を最小化させる損失関数を 2 値分類で確率を返す関数（ハイパラメータの引数としては binary:logistic）として実験を行った。表？？は木の最大の深さを変え実験を行った際の評価指標の値を示している。

表？？：XGBOOST による Titanic データの分析結果

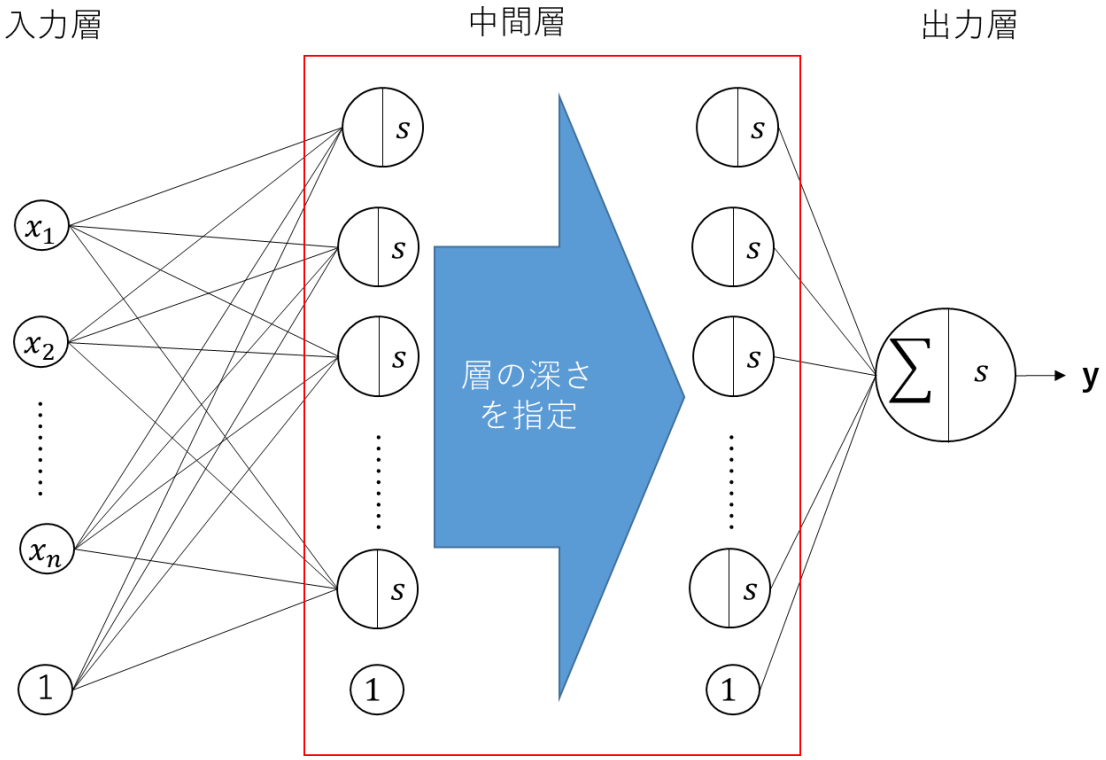
XGBOOST（決定木の最大数：100）							
木の深さ / 説明変数	1	2	3	4	5	10	20
精度	0.8204	0.8765	0.8855	0.8900	0.8945	0.9091	0.9125
適合率	0.7758	0.8766	0.8704	0.8813	0.8924	0.9117	0.9125
再現率	0.7485	0.7895	0.8246	0.8246	0.8246	0.8450	0.8538
f-1値	0.7619	0.8308	0.8468	0.8520	0.8571	0.8771	0.8822
AUC	0.8799	0.9215	0.9388	0.9474	0.9504	0.9624	0.9632
MAE	0.2720	0.2192	0.1988	0.1859	0.1781	0.1581	0.1559
MSE	0.1284	0.1001	0.0891	0.0828	0.0787	0.0690	0.0681

考察：

表？？より木が深くなるにつれ評価指標の値はよくなっており木の深さ 20 で精度（正答率）も 91%と高い。

4.4.6 NN による全 Titanic データの分析

NN モデルは図？？では出力の値が 0～1 の値を取らないため図？？のように出力を行うユニットに活性化関数としてシグモイド関数を用いて 0～1 の値を出力するようにした。この時閾値を 0.5 とし二値分類を行う。



x :説明変数、n:説明変数の数 s:シグモイド関数、i:恒等関数、y:予測値

図？？：NN モデルのネットワーク 2

前処理層は実験 4.1、4.2 で行ったように一層ごとに 1000 ユニットからなる全結合層の深さを 1～3 まで変えて実験を行った。学習条件は表？？とする。表？？は層の数を変え実験を行った際の評価指標の値を示している。

表？？：Titanic データにおける NN の学習条件

学習回数	バッチサイズ	更新式	誤差関数
10000回	200	adam	平均二乗誤差

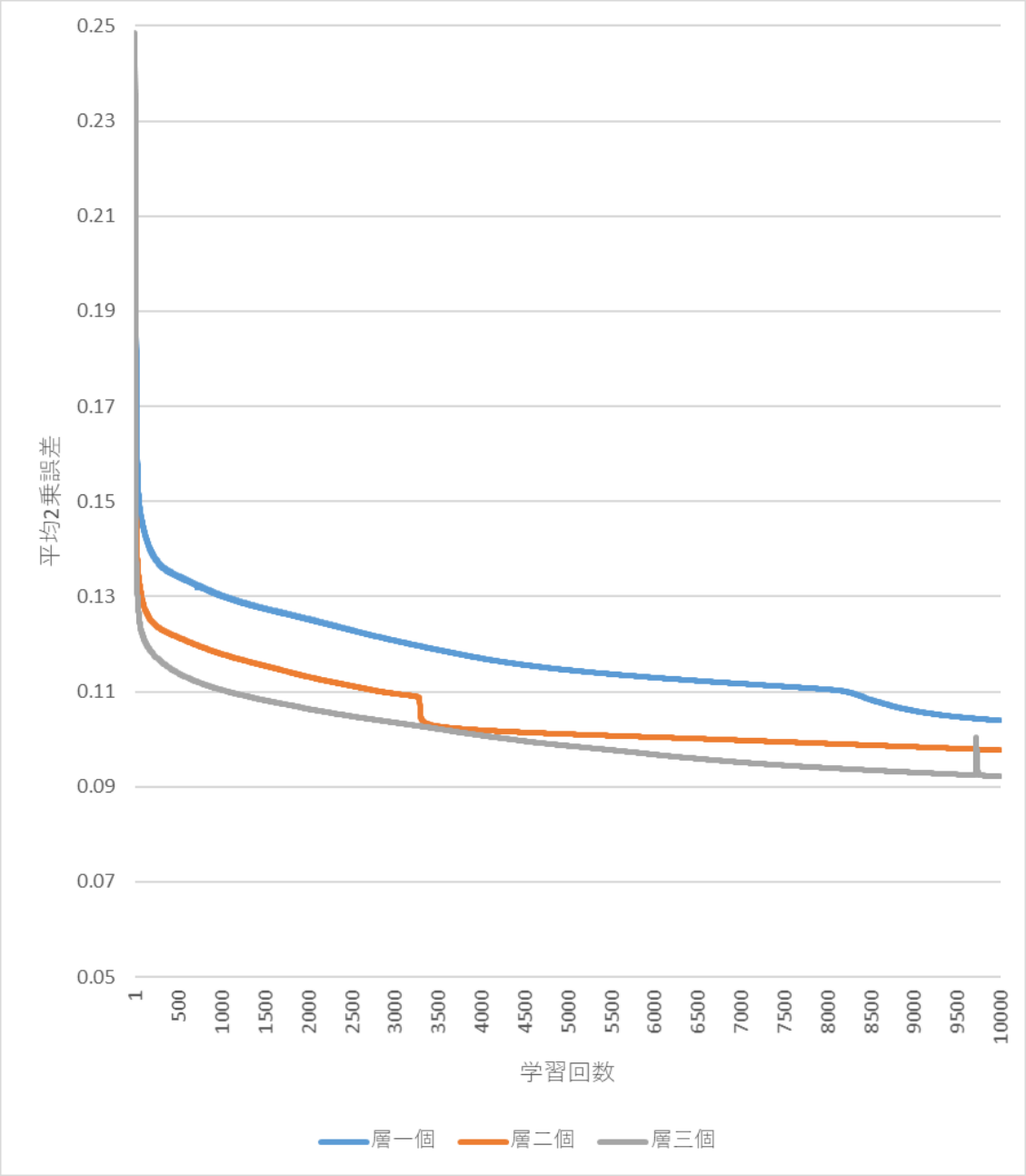
表？？：NN による Titanic データの分析結果

評価指標	層一個	層二個	層三個
精度	0.8600	0.8740	0.8730
適合率	0.8980	0.8820	0.8980
再現率	0.8120	0.8620	0.8400
f-1値	0.8530	0.8720	0.8680
AUC	0.9280	0.9301	0.9355
MAE	0.2160	0.1876	0.1763
MSE	0.1039	0.0977	0.0921

表？？と図？？はNNモデルの層の数を変えて実験を行った際の学習における平均二乗誤差の学習曲線を図示し、具体的な数値を示したものである。

表？？：NNによるTitanicデータの学習過程の平均二乗誤差

学習回数	層一個	層二個	層三個
1	0.2348	0.2446	0.2486
500	0.1341	0.1214	0.1139
1000	0.1301	0.1179	0.1104
1500	0.1274	0.1154	0.1082
2000	0.1252	0.1131	0.1064
2500	0.1229	0.1112	0.1048
3000	0.1207	0.1096	0.1035
3500	0.1188	0.1027	0.1022
4000	0.1170	0.1019	0.1008
4500	0.1156	0.1014	0.0996
5000	0.1145	0.1010	0.0986
5500	0.1137	0.1007	0.0977
6000	0.1129	0.1004	0.0968
6500	0.1123	0.1001	0.0959
7000	0.1117	0.0998	0.0951
7500	0.1111	0.0994	0.0945
8000	0.1105	0.0991	0.0939
8500	0.1083	0.0987	0.0935
9000	0.1060	0.0984	0.0930
9500	0.1047	0.0980	0.0926
10000	0.1039	0.0977	0.0921



図？？：NN による Titanic データの学習過程の平均二乗誤差

考察：

表？？より層の数が増えるほど誤差は減少し AUC は高くなっているが f-1 値や精度（正答率）は 2～3 にかけて減少している。

4.4.7　メビウス型包除積分モデル 1 による全 Titanic データの分析

メビウス型包除積分モデル 1 も図？？より出力の値が 0～1 の値を取らないため実験 4.4.4 でしたように出力を行うユニットに活性化関数としてシグモイド関数を用いて 0～1 の値を出力するようにした。この時閾値を 0.5 とし二値分類を行う。学習条件は表？？とす

る。表？？は加法性の制限を変え実験を行った際の評価指標の値を示している。

表？？：Titanic データにおけるメビウス型包除積分モデル 1 の学習条件

学習回数	バッチサイズ	更新式	誤差関数	t-norm
10000回	200	adam	平均二乗誤差	代数積

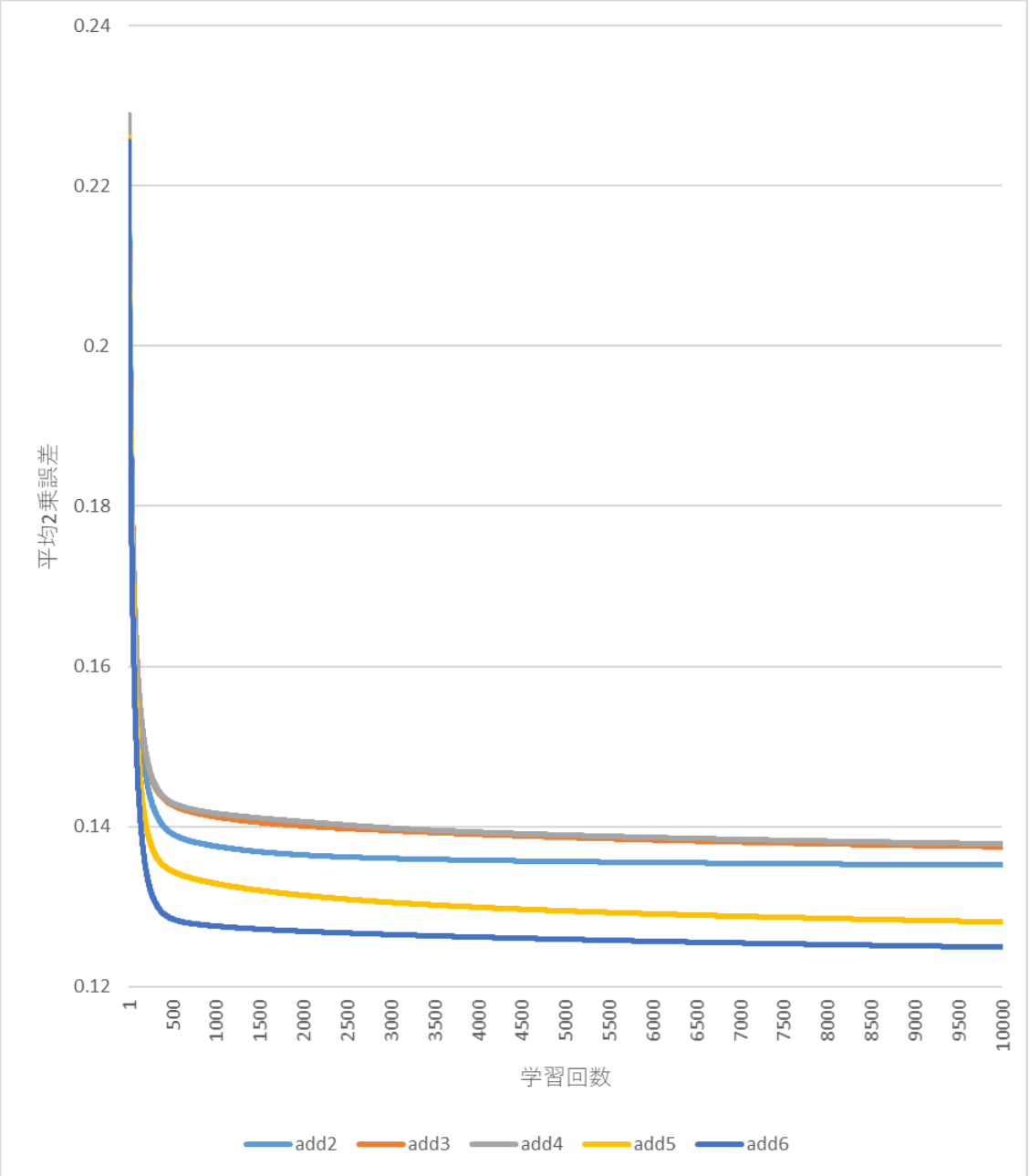
表？？：メビウス型包除積分モデル 1 による Titanic データの分析結果

制限 / 評価指標	add2	add3	add4	add5	add6
精度	0.8160	0.8000	0.8010	0.8290	0.8310
適合率	0.8540	0.8660	0.8270	0.8570	0.8700
再現率	0.7610	0.7110	0.7610	0.7900	0.7770
f-1値	0.8050	0.7810	0.7930	0.8220	0.8210
AUC	0.8754	0.8732	0.8694	0.8826	0.8912
MAE	0.2630	0.2706	0.2732	0.2534	0.2496
MSE	0.1352	0.1375	0.1378	0.1281	0.1249

表？？と図？？はメビウス型包除積分モデル 1 の加法性の制限を変えて実験を行った際の学習における平均二乗誤差の学習曲線を図示し、具体的な数値を示したものである。

表？？：メビウス型包除積分モデル 1 による Titanic データの学習過程の平均二乗誤差

制限 / 学習回数	add2	add3	add4	add5	add6
1	0.2289	0.2284	0.2289	0.2261	0.2255
500	0.1391	0.1427	0.1429	0.1344	0.1285
1000	0.1375	0.1412	0.1416	0.1329	0.1276
1500	0.1368	0.1405	0.1411	0.1320	0.1272
2000	0.1364	0.1401	0.1406	0.1314	0.1269
2500	0.1362	0.1397	0.1402	0.1309	0.1267
3000	0.1360	0.1395	0.1398	0.1305	0.1265
3500	0.1359	0.1392	0.1395	0.1302	0.1263
4000	0.1358	0.1390	0.1393	0.1299	0.1262
4500	0.1357	0.1388	0.1391	0.1297	0.1260
5000	0.1356	0.1386	0.1389	0.1295	0.1259
5500	0.1356	0.1385	0.1388	0.1293	0.1258
6000	0.1355	0.1383	0.1386	0.1291	0.1257
6500	0.1355	0.1382	0.1385	0.1289	0.1256
7000	0.1354	0.1381	0.1384	0.1288	0.1255
7500	0.1354	0.1379	0.1383	0.1287	0.1254
8000	0.1353	0.1378	0.1382	0.1285	0.1253
8500	0.1353	0.1377	0.1381	0.1284	0.1252
9000	0.1352	0.1376	0.1380	0.1283	0.1251
9500	0.1352	0.1375	0.1379	0.1282	0.1250
10000	0.1352	0.1375	0.1378	0.1281	0.1249



図？？：メビウス型包除積分モデル 1 による Titanic データの学習過程の平均二乗誤差

考察：

制限 2～6 にかけて表？？のように平均絶対誤差、平均二乗誤差は縮小しているが精度、適合率、再現率、f-1 値、AUC の値は上下している。

4.4.8　メビウス型包除積分モデル 2 による全 Titanic データの分析

メビウス型包除積分モデル 2 も図？？より出力の値が 0～1 の値を取らないため実験 4.4.4 でしたように出力を行うユニットに活性化関数としてシグモイド関数を用いて 0～1 の値を出力するようにした。この時閾値を 0.5 とし二値分類を行う。学習条件は表？？とす

る。表？？は全加法（6 加法まで）で学習を行った際の評価指標の値を示している。。

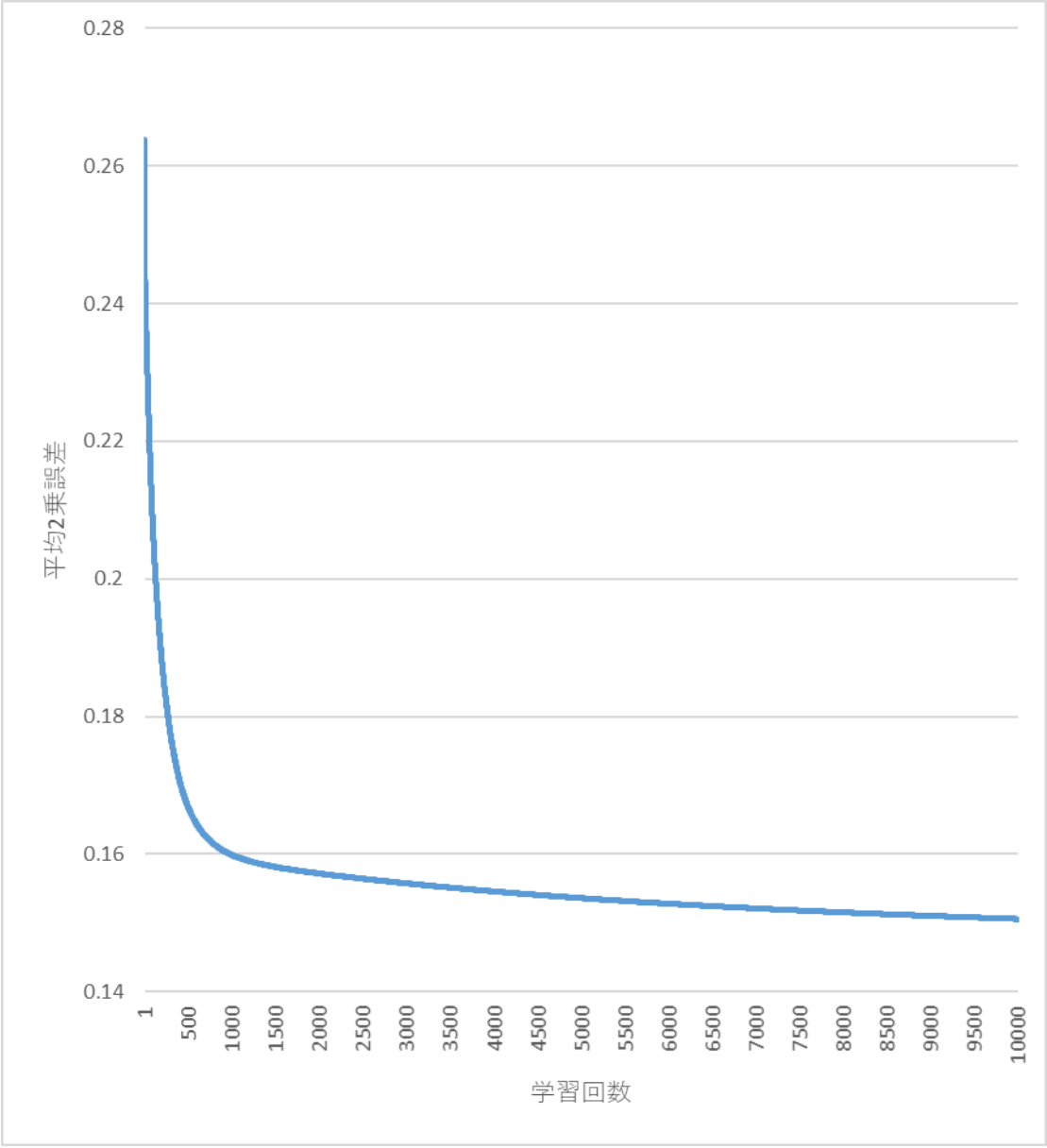
表？？：メビウス型包除積分モデル 2 による Titanic データの分析結果

評価指標	値
精度	0.7760
適合率	0.7550
再現率	0.8180
f-1値	0.7850
AUC	0.8529
MAE	0.3127
MSE	0.1506

表？？と図？？はメビウス型包除積分モデル 2 の学習における平均二乗誤差の学習曲線を図示し、具体的な数値を示したものである。

表？？：メビウス型包除積分モデル 2 による Titanic データの学習過程の平均二乗誤差

学習回数	平均2乗誤差
1	0.2639
500	0.1669
1000	0.1599
1500	0.1581
2000	0.1572
2500	0.1564
3000	0.1557
3500	0.1551
4000	0.1546
4500	0.1541
5000	0.1536
5500	0.1532
6000	0.1528
6500	0.1524
7000	0.1521
7500	0.1518
8000	0.1515
8500	0.1513
9000	0.1510
9500	0.1508
10000	0.1506



図？？：メビウス型包除積分モデル 2 による Titanic データの学習過程の平均二乗誤差

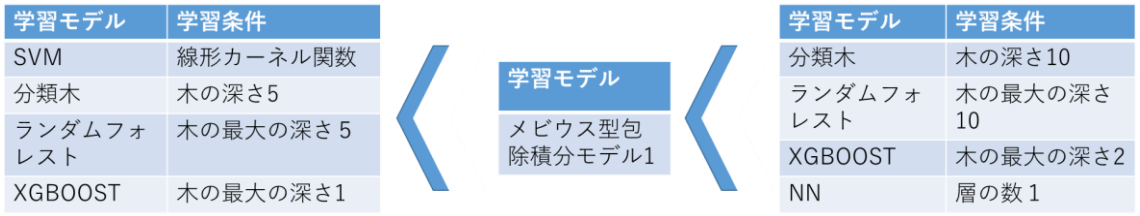
考察：

学習回数 1000 までは誤差を大きく下げているがそれ以降はゆっくりと学習が行われている。

4.4.9 実験 4.4 全体の考察

実験 4.4 全体としては分類木やランダムフォレスト、XGBOOST などの精度が高い。ACU においても XGBOOST の深さを 20 にした時のものが最も高くなっているが NN モデルの AUC も次いで高い値となった。提案手法については、メビウス型包除積分モデル 1 に比べメビウス型包除積分モデル 2 の精度が低くなっていることがわかった。提案手法の表現力について

AUC の値を表現力として見るとメビウス型包除積分モデル 1（全加法）の大小関係は次の図のように表すことができる。



図？？：AUC を基準とした提案手法の Titanic データにおける大小関係

図？？よりメビウス型包除積分モデル 1 の表現力は学習回数によってはここから上下すると思われるが分類問題においておおよそこの程度の表現力を有していると考えられる。

4.5　二値分類問題に対する汎化性能の比較

実験 4.5 では実験 4.4 で測れなかったモデルの汎化性能について K 分割交差検証によって比較を行っていく。実験に使用するデータは実験 4.4 同様に Titanic データとし、5 分割交差検証で汎化性能の比較を行う。提案手法と比較を行うため実験 4.4 で用いた機械学習手法でも 5 分割交差検証を行う。モデルにより過学習が起きているかどうかを見るため、5 分割交差検証で分割した学習データとテストデータのそれぞれのデータで評価指標の値を計算し、5 つそれぞれの学習データとテストデータの評価指標の値の平均を取る。評価指標は実験 4.4 同様に精度（正答率）、適合率、再現率、f-1 値、AUC、MAE、MSE とする。

4.5.1　重回帰による Titanic データに対する汎化性能

あああ

表？？：

学習データに対する評価指標の平均値

評価指標	値
精度	0.7796
適合率	0.7734
再現率	0.7920
f-1値	0.7824
AUC	0.7796
MAE	0.2204
MSE	0.2204

テストデータに対する評価指標の平均値

評価指標	値
精度	0.7777
適合率	0.6814
再現率	0.7881
f-1値	0.7305
AUC	0.7789
MAE	0.2223
MSE	0.2223

4.5.2　SVM による Titanic データに対する汎化性能

SVM の汎化性能を評価指標により評価していく。実験条件は C=1 としカーネル関数を変えて評価指標の値を計算する。表？？は SVM による 5 分割交差検証の結果をまとめたものである。

表？？：SVM の Titanic データに対する 5 分割交差検証で得られた分析結果

学習データに対する評価指標の平均値

SVMによる分析結果（C=1）			
カーネル関数 / 評価指標	線形カーネル	多項カーネル	RBFカーネル
精度	0.7893	0.6033	0.7248
適合率	0.8010	0.9001	0.7087
再現率	0.7698	0.2324	0.7616
f-1値	0.7850	0.3693	0.7333
AUC	0.7893	0.6033	0.7248
MAE	0.2107	0.3967	0.2752
MSE	0.2107	0.3967	0.2752

テストデータに対する評価指標の平均値

SVMによる分析結果（C=1）			
カーネル関数 / 評価指標	線形カーネル	多項カーネル	RBFカーネル
精度	0.7913	0.6768	0.7082
適合率	0.7136	0.7996	0.5945
再現率	0.7645	0.2134	0.7469
f-1値	0.7377	0.3351	0.6615
AUC	0.7865	0.5896	0.7152
MAE	0.2087	0.3232	0.2918
MSE	0.2087	0.3232	0.2918

考察：

表？？より、どの評価指標でも線形カーネルが優れており学習データとテストデータに対する値も同程度で過学習がほとんど起きていない。

4.5.3　分類木による Titanic データに対する汎化性能

分類木は木の深さを 1,2,3,4,5,10,20 と変えて汎化性能を評価する。表？？は分類木による汎化性能の結果をまとめたものである。

表？？：分類木の Titanic データに対する 5 分割交差検証で得られた分析結果

学習データに対する評価指標の平均値

回帰木							
木の深さ / 評価指標	1	2	3	4	5	10	20
精度	0.7834	0.7933	0.8210	0.8372	0.8589	0.9168	0.9307
適合率	0.7987	0.8067	0.8557	0.8583	0.8837	0.9459	0.9398
再現率	0.7584	0.7718	0.7762	0.8096	0.8270	0.8844	0.9204
f-1値	0.7778	0.7888	0.8126	0.8325	0.8540	0.9140	0.9299
AUC	0.7834	0.7933	0.8210	0.8372	0.8589	0.9168	0.9307
MAE	0.2166	0.2067	0.1790	0.1628	0.1411	0.0832	0.0693
MSE	0.2166	0.2067	0.1790	0.1628	0.1411	0.0832	0.0693

テストデータに対する評価指標の平均値

回帰木							
木の深さ / 評価指標	1	2	3	4	5	10	20
精度	0.7912	0.7912	0.8238	0.8215	0.8260	0.8261	0.8238
適合率	0.7130	0.7146	0.7796	0.7680	0.7654	0.7711	0.7714
再現率	0.7592	0.7569	0.7477	0.7768	0.7802	0.7657	0.7789
f-1値	0.7348	0.7350	0.7595	0.7676	0.7712	0.7671	0.7730
AUC	0.7849	0.7841	0.8048	0.8128	0.8161	0.8143	0.8158
MAE	0.2088	0.2088	0.1762	0.1785	0.1740	0.1739	0.1762
MSE	0.2088	0.2088	0.1762	0.1785	0.1740	0.1739	0.1762

考察：

表？？から木の深さを深くするにつれて学習データとテストデータの評価指標の値は共によくなっており、木の深さ 10 までは過学習はほとんど起こっていない。木の深さが 10～20 にかけてテストデータに対する誤差が上がっているが AUC も上がっている。

4.5.4　ランダムフォレストによる Titanic データに対する汎化性能

ランダムフォレストの学習条件は決定木の数を 100 とし、木の深さを 1,2,3,4,5,10,20 と変えて汎化性能を評価する。表？？はランダムフォレストによる 5 分割交差検証の結果をまとめたものである。

表？？：ランダムフォレストの Titanic データに対する 5 分割交差検証で得られた分析結果

学習データに対する評価指標の平均値

ランダムフォレスト（決定木の数：100）							
木の深さ / 評価指標	1	2	3	4	5	10	20
精度	0.7916	0.7849	0.7993	0.8327	0.8622	0.9257	0.9243
適合率	0.8080	0.7972	0.8219	0.8586	0.8901	0.9483	0.9405
再現率	0.7652	0.7642	0.7650	0.7982	0.8268	0.9006	0.9060
f-1値	0.7859	0.7803	0.7921	0.8261	0.8571	0.9238	0.9229
AUC	0.7916	0.7849	0.7993	0.8327	0.8622	0.9257	0.9243
MAE	0.2084	0.2151	0.2007	0.1673	0.1378	0.0743	0.0757
MSE	0.2084	0.2151	0.2007	0.1673	0.1378	0.0743	0.0757

テストデータに対する評価指標の平均値

ランダムフォレスト（決定木の数：100）							
木の深さ / 評価指標	1	2	3	4	5	10	20
精度	0.7901	0.7879	0.7935	0.8125	0.8226	0.8171	0.8159
適合率	0.7090	0.7005	0.7146	0.7579	0.7781	0.7626	0.7493
再現率	0.7564	0.7638	0.7640	0.7700	0.7551	0.7604	0.7743
f-1値	0.7303	0.7302	0.7381	0.7587	0.7641	0.7612	0.7602
AUC	0.7834	0.7816	0.7876	0.8062	0.8090	0.8056	0.8077
MAE	0.2099	0.2121	0.2065	0.1875	0.1774	0.1829	0.1841
MSE	0.2099	0.2121	0.2065	0.1875	0.1774	0.1829	0.1841

考察：

木の深さが 1～5 まではテストデータの誤差も減少しており、それ以上深くすると誤差は上昇している。よって木の深さ 5～10、10～20 にかけて過学習が発生している。

4.5.5　XGBOOST による Titanic データに対する汎化性能

XGBOOST の学習条件は決定木の最大数を 100 とし、木の最大の深さを 1,2,3,4,5,10,20 と変えて汎化性能を評価する。ただし、XGBOOST の学習では過学習を抑えるためのハイパーパラメータである early_stopping_rounds という値を 10 に設定し学習を行った。表？？は XGBOOST による 5 分割交差検証の結果をまとめたものである。

表？？：XGBOOST の Titanic データに対する 5 分割交差検証で得られた分析結果

学習データに対する評価指標の平均値

XGBOOST（決定木の最大数：100）							
木の深さ / 説明変数	1	2	3	4	5	10	20
精度	0.7960	0.8600	0.8810	0.8850	0.8960	0.9090	0.9140
適合率	0.7860	0.8770	0.8960	0.9030	0.9050	0.9310	0.9310
再現率	0.8140	0.8380	0.8610	0.8630	0.8840	0.8840	0.8940
f-1値	0.8000	0.8570	0.8780	0.8820	0.8940	0.9060	0.9120
AUC	0.8750	0.9220	0.9490	0.9510	0.9600	0.9690	0.9720
MAE	0.3010	0.2490	0.2060	0.2030	0.1870	0.1650	0.1550
MSE	0.1430	0.1120	0.0890	0.0870	0.0790	0.0700	0.0660

テストデータに対する評価指標の平均値

XGBOOST（決定木の最大数：100）							
木の深さ / 説明変数	1	2	3	4	5	10	20
精度	0.7800	0.8260	0.8360	0.8270	0.8130	0.8240	0.8290
適合率	0.6810	0.7710	0.7780	0.7590	0.7410	0.7710	0.7900
再現率	0.8080	0.7790	0.8000	0.8010	0.7870	0.7680	0.7650
f-1値	0.7390	0.7740	0.7880	0.7790	0.7620	0.7690	0.7750
AUC	0.8580	0.8710	0.8840	0.8720	0.8780	0.8640	0.8570
MAE	0.3100	0.2740	0.2480	0.2580	0.2480	0.2360	0.2340
MSE	0.1480	0.1310	0.1260	0.1320	0.1350	0.1400	0.1400

考察：

テストデータに対する精度、f-1 値、AUC、平均二乗誤差は木の深さが 3 の時をピークとして木の深さ 4 で悪くなっていることから木の深さ 4 から過学習が発生している。だが平均絶対誤差は木を深くするにつれて学習データでもテストデータでも減少している。

4.5.6　NN による Titanic データに対する汎化性能

NN の学習に使用するモデルは実験 4.4.5 同様に図？？のように活性化関数をシグモイド関数とし、一層ごとに 1000 ユニットからなる全結合層の深さを 1～3 まで変えて実験を行った。学習条件は実験 4.4.5 と同様表？？とする。ただし、過学習を防止するため学習毎にテストデータに対する平均 2 乗誤差が学習以前の平均 2 乗誤差の最小値より大きい場合をカウントし、50 カウントその状態が続いた場合学習を止める。表？？は NN による汎化性能の結果をまとめたものである。

表？？：NN の Titanic データに対する 5 分割交差検証で得られた分析結果

学習データに対する評価指標の平均値

評価指標	層一個	層二個	層三個
精度	0.8350	0.8405	0.8318
適合率	0.8543	0.8435	0.8433
再現率	0.8078	0.8368	0.8186
f-1値	0.8302	0.8396	0.8296
AUC	0.8907	0.8949	0.8898
MAE	0.2485	0.2399	0.2471
MSE	0.1242	0.1214	0.1268

テストデータに対する評価指標の平均値

評価指標	層一個	層二個	層三個
精度	0.8092	0.8114	0.8148
適合率	0.7435	0.7424	0.7530
再現率	0.7685	0.7829	0.7761
f-1値	0.7537	0.7594	0.7624
AUC	0.8690	0.8685	0.8663
MAE	0.2616	0.2600	0.2589
MSE	0.1342	0.1361	0.1346

考察：

過学習を抑制したため層が浅くても深くても評価指標の値は同程度となった。

4.5.7　メビウス型包除積分モデル１による Titanic データに対する汎化性能

提案手法では実験 4.4.6 同様に学習条件として表？？を用い、過学習を防ぐため 4.5.5 の NN で行った学習毎にテストデータに対する平均 2 乗誤差が学習以前の平均 2 乗誤差の最小値より大きい場合をカウントし、50 カウントその状態が続いた場合学習を止める手法をとる。使用する提案手法は全データを用いた学習で比較的精度が良かったメビウス型包除積分モデル 1 とし学習を行う。表？？は加法性に制限を付けた時の評価指標の値の平均の値である。

表？？：メビウス型包除積分モデル 1 の Titanic データに対する 5 分割交差検証で得られた分析結果

学習データに対する評価指標の平均値

制限 / 評価指標	add2	add3	add4	add5	add6
精度	0.8069	0.8174	0.8118	0.8196	0.8033
適合率	0.8687	0.8796	0.8642	0.8731	0.8677
再現率	0.7234	0.7358	0.7400	0.7484	0.7168
f-1値	0.7892	0.8012	0.7972	0.8056	0.7846
AUC	0.8738	0.8796	0.8726	0.8770	0.8705
MAE	0.2703	0.2679	0.2712	0.2675	0.2774
MSE	0.1344	0.1321	0.1350	0.1327	0.1370

テストデータに対する評価指標の平均値

制限 / 評価指標	add2	add3	add4	add5	add6
精度	0.8103	0.8137	0.8182	0.8036	0.8092
適合率	0.7924	0.7870	0.7925	0.7619	0.7916
再現率	0.6889	0.7034	0.7136	0.7141	0.6903
f-1値	0.7354	0.7423	0.7505	0.7340	0.7361
AUC	0.8591	0.8524	0.8573	0.8599	0.8604
MAE	0.2787	0.2804	0.2809	0.2762	0.2841
MSE	0.1368	0.1378	0.1374	0.1367	0.1390

考察：

過学習を抑制しているため 2～6 に制限したが学習データ、テストデータともに大きく誤差による変化はない。

4.5.8　実験 4.5 全体の考察

各学習モデルの中でもテストデータに対する正答率の値が最も高くなったモデルで順位付けを行うと表？？のようになった。

表？？：精度（正答率）による汎化性能比較

rank	モデル	正答率
1	XGBOOST：木の最大の深さ3	0.8360
2	回帰木：木の深さ5	0.8261
3	ランダムフォレスト：木の最大の深さ5	0.8226
4	メビウス型包除積分モデル1：制限4	0.8182
5	NN：層数3	0.8148
6	SVM:線形カーネル	0.7913

表？？から決定木を用いた手法は正答率が高い傾向がみられる。次に検証用データに対する正答率を測り順位付けを行うと表？？のような結果が得られた。

表？？：検証データに対する正答率

rank	モデル	正答率
1	メビウス型包除積分モデル1：制限4	0.7895
2	ランダムフォレスト：木の最大の深さ5	0.7799
3	NN：層数3	0.7751
4	回帰木：木の深さ5	0.7727
5	XGBOOST：木の最大の深さ3	0.7632
6	SVM:線形カーネル	0.756

表？？と表？？からテストデータではうまく学習できているが検証用データでうまくいかないモデルもあるがメビウス型包除積分モデル 1 は検証用データに対してもうまく特徴を捉えることができていた。

4.6　二値分類問題における XGBOOST と提案手法の解釈比較

実験 4.6 では実験 4.3 と同様の操作を行い XGBOOST の学習モデルから得られる gain 値と shap 値を比較対象とし、提案手法の学習モデルからシャープレイ値＋拡張したシャープレイ値を算出し比較を行っていく。

4.6.1　Titanic データの全データを学習データとして得られた重回帰の解釈

あああ

表？？：

説明変数	係数の値
タイトルグループ	0.2090
性別	-0.1323
単独乗車	0.0461
旅客等級	-0.1547
乗船料金	0.0001
乗船港	-0.0697

考察：

4.6.2　Titanic データの全データを学習データとして得られた XGBOOST の解釈

次の表？？と表？？は全データを学習データとして用いた XGBOOST の学習モデルの gain 値と shap 値を示している。

表？？：Titanic データの全データを学習データとして得られる gain 値

XGBOOST（決定木の最大数：100）							
木の深さ / 説明変数	1	2	3	4	5	10	20
title_group	38.97	26.60	16.29	13.95	12.44	7.00	5.23
sex	40.38	6.85	1.12	0.64	0.65	0.39	0.34
alone	1.17	0.82	0.41	0.40	0.30	0.18	0.15
guest_class	10.24	10.08	6.92	5.46	4.89	2.67	2.39
fare	1.20	1.23	0.91	0.67	0.50	0.30	0.26
seaport	3.86	1.73	0.90	0.60	0.52	0.27	0.23

表？？：Titanic データの全データを学習データとして得られる shap 値

XGBOOST（決定木の最大数：100）							
木の深さ / 説明変数	1	2	3	4	5	10	20
title_group	1.160	1.362	1.538	1.594	1.608	1.662	1.650
sex	0.216	0.177	0.081	0.101	0.108	0.135	0.138
alone	0.054	0.097	0.049	0.052	0.079	0.084	0.106
guest_class	0.860	0.816	0.784	0.791	0.775	0.833	0.866
fare	0.244	0.531	0.631	0.758	0.807	0.939	0.983
seaport	0.161	0.323	0.260	0.276	0.314	0.365	0.357

考察：

木の深さによって gain 値、shap 値の値が異なるが、共通して title_group を重要視する傾向がみられる。深さ 20 の場合 gain 値では guest_class が重要視され fare は比較的低い値となっているが、shap 値では fare を重要視しており指標によって重要度が異なっている。

4.6.3 Titanic データの全データを学習データとして得られた提案手法の解釈

次に提案手法の中でも精度の良いメビウス型包除積分モデル 1 で学習を行った学習モデルからシャープレイ値を求める。表？？は得られたモデルのシャープレイ値を示しており、次の表？？では加法性を制限した中で計算できる拡張したシャープレイ値による大小関係をランキングとしており、表内の[ ]の中の数字は表？？の説明変数を数字で置き換えたものである。

表？？：説明変数の割り当て

1	2	3	4	5	6
title_group	sex	alone	guest_class	fare	seaport

表？？：Titanic データの全データを学習データとして得られるシャープレイ値

制限 / 説明変数	add2	add3	add4	add5	add6
title_group	0.14	0.90	-3.70	-9.60	6.10
sex	2.75	0.58	11.56	6.20	6.77
alone	-2.53	-3.55	-6.01	-6.05	-1.55
guest_class	7.21	13.93	22.20	26.71	24.67
fare	-4.89	-6.98	-3.41	-14.55	-2.86
seaport	3.97	4.13	8.43	2.32	9.40

表？？：Titanic データの全データを学習データとして得られる拡張したシャープレイ値のランキング

制限 / rank	add2	add3	add4	add5	add6
1	[4]	[2,4,5]	[2,4,5]	[4,5]	[4,5]
2	[5,6]	[4,5]	[4,5]	[2,4,5]	[2,4,5]
3	[4,5]	[4]	[4]	[4]	[4]
4	[1,3]	[2,4]	[2,4]	[2,4]	[1,4,5]
5	[2,4]	[1,3,4]	[1,2,4,5]	[3,4,5]	[1,2,4,5]
6	[6]	[5,6]	[3,4,5]	[2,3,4,5]	[1,4]
7	[3,4]	[3,4]	[2,3,4,5]	[1,2,4,5]	[2,4]
8	[2]	[3,4,5]	[1,4,5]	[3,4]	[3,4,5]
9	[2,6]	[1,4]	[2,5,6]	[1,4,5]	[1,3,4]
10	[4,6]	[6]	[5,6]	[1,2,4]	[3,4]

また、シャープレイ値と拡張したシャープレイ値は負の値を取ってしまいランク付けを行った場合負の値を低く評価してしまうため、この値の絶対値を取ることで負に貢献したととらえることができる。表？？はシャープレイ値の絶対値を重要度として示しており、表？？は拡張したシャープレイ値の絶対値のランキングを示している。

表？？：Titanic データの全データを学習データとして得られる重要度

制限 / 説明変数	add2	add3	add4	add5	add6
title_group	0.14	0.90	3.70	9.60	6.10
sex	2.75	0.58	11.56	6.20	6.77
alone	2.53	3.55	6.01	6.05	1.55
guest_class	7.21	13.93	22.20	26.71	24.67
fare	4.89	6.98	3.41	14.55	2.86
seaport	3.97	4.13	8.43	2.32	9.40

表？？：Titanic データの全データを学習データとして得られる拡張した重要度のランキング

制限 / rank	add2	add3	add4	add5	add6
1	[1,5]	[2,4,5]	[2,4,5]	[4,5]	[4,5]
2	[4]	[4,5]	[4,5]	[2,4,5]	[2,4,5]
3	[5,6]	[4]	[4]	[1,5]	[4]
4	[4,5]	[1,5]	[2,4]	[4]	[1,4,5]
5	[3,5]	[2,4]	[1,3,5]	[2,4]	[1,2,4,5]
6	[1,3]	[3,5]	[1,5]	[3,4,5]	[1,4]
7	[5]	[2,3,5]	[1,2,4,5]	[1,3,5]	[2,4]
8	[2,4]	[1,3,4]	[3,4,5]	[2,3,4,5]	[3,4,5]
9	[6]	[5]	[2,3,4,5]	[1,2,4,5]	[1,3,4]
10	[3,4]	[5,6]	[1,4,5]	[3,4]	[3,4]

考察：

表？？からシャープレイ値では guest_class が最も重要視されており、表？？の重要度をとっても guest_class が最も大きい値となっている。表？？と表？？から[2,4,5]→[sex,guest_class,fare]、もしくは[4,5]→[guest_class,fare]は重要視される傾向がみられる。

4.6.4　Titanic データを 5 分割交差検証して得られた重回帰の解釈

あああ

表？？：

説明変数	係数の値
タイトルグループ	1.2282
性別	-0.4121
単独乗車	0.2878
旅客等級	-0.9308
乗船料金	0.0013
乗船港	-0.4940

考察：

4.6.5　Titanic データを 5 分割交差検証して得られた XGBOOST の解釈

全データを学習して得られた解釈と 5 分割交差検証で汎化性能を高めることで得られる解釈が異なるかどうかを見るためまず XGBOOST で 5 分割交差検証を行った際の gain 値と shap 値を計算する。ただし、5 分割交差検証で得られるモデルは 5 つなので 5 つのモデルから得られた gain 値と shap 値の平均したものを表？？と表？？に示している。

表？？：Titanic データを 5 分割交差検証して得られる gain 値

XGBOOST（決定木の最大数：100）							
木の深さ / 説明変数	1	2	3	4	5	10	20
title_group	137.31	89.72	60.73	63.99	60.01	53.51	62.34
sex	62.16	41.26	2.50	5.33	2.77	2.19	3.31
alone	2.33	3.31	2.81	3.68	4.84	1.23	1.52
guest_class	47.55	25.90	23.79	22.11	17.22	11.26	10.68
fare	12.08	7.66	6.25	5.34	4.32	2.12	1.73
seaport	12.78	9.88	4.07	4.69	2.84	1.93	1.65

表？？：Titanic データを 5 分割交差検証して得られる shap 値

XGBOOST（決定木の最大数：100）							
木の深さ / 説明変数	1	2	3	4	5	10	20
title_group	1.071	1.228	1.552	1.483	1.564	1.648	1.740
sex	0.160	0.232	0.087	0.097	0.076	0.094	0.206
alone	0.034	0.059	0.065	0.086	0.080	0.093	0.098
guest_class	0.610	0.783	0.724	0.699	0.728	0.828	0.916
fare	0.181	0.418	0.634	0.591	0.672	0.795	0.791
seaport	0.135	0.195	0.226	0.231	0.256	0.260	0.290

考察：

表？？と表？？よりどちらの指標でも title_group が最も重要視される傾向がみられ、次いで guest_class が重要視されている。

4.6.6　5 分割交差検証で得られた提案手法の解釈

メビウス型包除積分モデル 1 で同様に 5 分割交差検証を行った際に得られた 5 つの学習モデルからシャープレイ値を求め平均をとったものを表？？に示す。拡張したシャープレイ値も平均をとりランキングとして表？？に示す。

表？？：Titanic データを 5 分割交差検証して得られるシャープレイ値

制限 / 説明変数	add2	add3	add4	add5	add6
title_group	4.11	3.40	5.51	3.95	6.83
sex	1.40	1.68	1.68	5.04	4.94
alone	-0.71	-2.31	-1.71	-1.90	0.60
guest_class	4.43	6.76	10.10	15.84	10.78
fare	-1.75	-4.37	-3.35	-3.82	0.68
seaport	1.89	2.05	3.85	4.84	6.00

表？？：Titanic データを 5 分割交差検証して得られる拡張したシャープレイ値のランキング

制限 / rank	add2	add3	add4	add5	add6
1	[4]	[4]	[4]	[4]	[4]
2	[1]	[2,4]	[1,4]	[4,5]	[4,5]
3	[2,4]	[4,5]	[4,5]	[2,4,5]	[1,4]
4	[1,4]	[1]	[1,3,4]	[2,4]	[2,4,5]
5	[5,6]	[2,4,5]	[1,4,5]	[1,4]	[2,4]
6	[1,3]	[1,4]	[2,4]	[1,4,5]	[5,6]
7	[6]	[3,4]	[3,4]	[3,4,5]	[1]
8	[3,4]	[1,3,4]	[1]	[1,2,4,5]	[1,4,5]
9	[2]	[1,3]	[2,4,5]	[3,4]	[6]
10	[2,6]	[6]	[1,6]	[2,3,4,5]	[1,3,4]

5 分割交差検証で得られた 5 つの学習モデルそれぞれのシャープレイ値の絶対値を平均したものを重要度として表？？に示す。同様に拡張したシャープレイ値の絶対値を平均し重要度としてランキングにしたものを表？？に示す。

表？？：Titanic データを 5 分割交差検証して得られる重要度

制限 / 説明変数	add2	add3	add4	add5	add6
title_group	3.26	3.61	5.51	7.12	6.83
sex	1.96	2.04	1.83	5.04	4.94
alone	1.70	2.31	2.19	2.04	1.04
guest_class	5.18	6.76	10.10	15.84	10.78
fare	4.45	4.37	3.95	4.09	1.56
seaport	1.53	2.05	3.85	5.01	6.00

表？？：Titanic データを 5 分割交差検証して得られる拡張した重要度のランキング

制限 / rank	add2	add3	add4	add5	add6
1	[4]	[4]	[4]	[4]	[4]
2	[5]	[3,5]	[1,4]	[4,5]	[4,5]
3	[3,5]	[2,4]	[4,5]	[2,4,5]	[1,4]
4	[2,4]	[5]	[3,5]	[2,4]	[2,4,5]
5	[1]	[4,5]	[1,3,4]	[1,4]	[2,4]
6	[1,5]	[1]	[1,4,5]	[3,4,5]	[5,6]
7	[1,3]	[1,4]	[2,4]	[1,4,5]	[1]
8	[3,4]	[2,4,5]	[3,4]	[1,2,4,5]	[1,4,5]
9	[5,6]	[2,3,5]	[1,3,5]	[3,4]	[6]
10	[2]	[1,5]	[1]	[2,3,4,5]	[1,3,4]

考察：

表？？、表？？のどちらにおいても guest_class が最も重要視されており、次いで title_group が重要視されている。

4.6.7　実験 4.6 全体の考察

XGBOOST では表？？と表？？のように shap 値によって fare を重要視する傾向がみられ、同様にメビウス型包除積分モデル 1 を 2 加法まで制限した場合にも fare を重要視する傾向がみられた。しかし、木の深さが深いまたはメビウス型包除積分モデル 1 の 5～6 加法まで制限した場合では fare の重要度は低くなっていた。また、XGBOOST、メビウス型包除積分モデル 1 のいずれでも title_group、guest_class は重要視されていることからこの 2 変数は重要であることが

第5章　結論

今回の実験では diabetes データを用いた回帰問題と Titanic データを用いた二値分類問題を解かせることで提案手法であるメビウス型包除積分モデルの表現力と汎化性能を調べ、解釈を行った。メビウス型包除積分モデル１の表現力はすべてのデータを学習用とした場合の精度から(学習回数にもよるが)決定木の深さ 5〜6 程度の表現力であると考えられる。メビウス型包除積分モデル１の汎化性能は回帰問題ではテストデータでも優れていたが、二値分類問題におけるテストデータで決定木を用いた手法と比較して低い結果となった。しかしながらメビウス型包除積分モデル１の検証データにおける正答率が他手法と比べ高いため一概に二値分類問題に対する汎化性能が低いということはないと考えらえる。メビウス型包除積分モデル 2 は回帰問題、二値分類問題のどちらでも低い精度となったためうまく学習できていないと考えられる。そのため前処理層については初期値を与えるなどの工夫が必要となってくる。解釈性については XGBOOST と比較を行い、XGBOOST でも重要視されたものはメビウス型包除積分モデル１でも重要視されていることがわかった。また、シャープレイ値を拡張することでより詳細に変数間の関係を見ることができるようになった。

第6章　謝辞

本研究を行うにあたり、多くの貴重な御意見、御指導を賜りました本田　あおい准教授に心より感謝いたします。実験を行うにあたり、御協力を賜りました佐々木君、Alex 君をはじめとする本田研究室の皆様に心より感謝いたします。最後になりましたが、これまで私を見守り、支え続けてくれた家族に心より感謝します。

第7章　参考文献

第8章　データ