Vol.2013-MPS-95 No.2
2013/9/26

順序回帰のための 高速な疎Bayes学習アルゴリズム

長島 主尚[1,a)]　井上 真郷[1,b)]

概要：順序回帰問題とは，多クラス分類問題のうち，クラス間に順序関係があるものである．順序回帰問題への疎なアプローチの一つとして，automatic relevance determination (ARD) 事前分布を用いたものが Chang らにより提案されている．これは Tipping により 2001 年に提案された，回帰問題と分類問題への疎なアプローチである relevance vector machine (RVM) と類似のものである．RVM は 2003 年に Tipping らによってより高速なアルゴリズムが提案され，欠点の一つであった計算量の問題を大きく改善した．本研究では，まず既存研究に比べて解析の容易なモデルを定義する．さらに，RVM に倣い，順序回帰問題の高速な疎学習アルゴリズムを提案する．また，既存モデルと同等以上の性能を保ちつつ，計算量が劇的に改善することを数値実験で示す．

キーワード：順序回帰，Bayes 推定，automatic relevance determination 事前分布，疎学習，Taylor 近似

Fast Sparse Bayesian Learning Algorithm for Ordinal Regression

Nagashima Kazuhisa[1,a)]　Inoue Masato[1,b)]

Abstract: Ordinal regression is a multiclass classification problem in which classes have an order relation. One of the sparse approaches for ordinal regression has been proposed by Chang et al., which utilizes automatic relevance determination (ARD) prior. This idea is similar to the one of the relevance vector machines (RVMs) for regression and classification problems by Tipping, 2001. A fast algorithm for solving RVMs has also been proposed by Tipping et al. in 2003. This algorithm greatly improves the computational complexity of RVMs. In this manuscript, we introduce a new model for ordinal regression that is easy to handle and propose a fast sparse learning method according to that of RVMs. We then illustrate that the proposed method runs remarkably faster than existing methods with equivalent or better precision rates by numerical experiments.

Keywords: ordinal regression, Bayesian estimation, automatic relevance determination prior, sparse learning, Taylor approximation

1. はじめに

近年，頑健な推定が可能であったり，アルゴリズムが高速になるといった理由から，推定問題に対する疎なアプローチが注目されている．本研究では代表的な疎モデルの一つである，relevance vector machine (RVM) に注目した．これは当初 2001 年に Tipping が提案した回帰問題や分類問題への疎なアプローチであるが，2003 年には疎性を利用することで高速なアルゴリズムが導出できることが明らかになった．本研究のテーマである順序回帰問題に対しても，RVM と同様の疎なアプローチは提案されているが，疎性を利用した高速なアルゴリズムは存在しない．そこで本研究では，より解析が容易なモデルと共に適切な近似を用いることで，RVM と同様に高速なアルゴリズムが導出できることを示す．

[1] 早稲田大学大学院 先進理工学研究科 電気・情報生命専攻
[a)] kazuhisa@suou.waseda.jp
[b)] masato.inoue@eb.waseda.ac.jp

2. 順序回帰問題

2.1 表記

本稿では，確率関数・確率密度関数の条件付確率の条件側の変数について，これが確率変数であれば，| を，確率変数でない場合は ; を用いて表す．例えば a は確率変数，b は確率変数でない変数の場合，

$$p(x \mid a; b)$$

などと表記する．また，確率変数でない変数，特に後述の入力や基底関数の出力は，表記を省略する（一般に確率変数の方は，省略すると意味が変わるため省略できない）．

確率変数 $\mathbf{x} \in \mathbb{R}^N$ が平均 $\boldsymbol{\mu}$, 分散共分散行列 $\boldsymbol{\Sigma}$ の多変量正規分布に従う時，この確率密度関数を次のように表記する．

$$\mathcal{N}(\mathbf{x}; \boldsymbol{\mu}, \boldsymbol{\Sigma}) \equiv \frac{1}{\sqrt{|2\pi\boldsymbol{\Sigma}|}} \exp\left(-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})\right)$$

但し，$\mathbb{R}$ は実数体を，$|\bullet|$ は行列式を，$\bullet^\top$ は転置を表す．本稿ではベクトルは断りが無い場合は縦ベクトルを表すものとする．また，ロジスティックシグモイド関数，指示関数を次のように表記する．

$$\sigma(x) \equiv \frac{1}{1+\mathrm{e}^{-x}}$$

$$1_\bullet \equiv \begin{cases} 1 & (\text{if } \bullet \text{ is true}) \\ 0 & (\text{otherwise}) \end{cases}$$

また，σ'，σ'' をそれぞれ σ の一階微分，二階微分とする．また，正の実数の集合を $\mathbb{R}^+$ と表記する．$\mathbf{0}_{i,j}$ は $(i \times j)$ の零行列，$\mathbf{I}_i$ は $(i \times i)$ の単位行列とする．diag は与えられたベクトルを対角要素とする対角行列を返す．

2.2 順序回帰問題

クラス判別問題とは，入力 x に対してその出力であるクラスラベル $t \in \{0, 1, \cdots, K\}$ を推定する問題である．通常は教師あり学習の枠組みで，入力と正しいクラスラベルの組 N 個 $\{x_n, t_n\}_{n=1}^N$ が事前に与えられており，これを手掛かりに推定する．x_n，t_n はそれぞれ n 番目の入出力を表す．また，クラスラベルを推定しなければならない入出力を便宜上 $N+1$ 番目とし，x_{N+1}，t_{N+1} と表記する．また，$\mathbf{t} \equiv [t_1, ..., t_N]^\top$ とする．

順序回帰問題とは，クラスラベルが**順序尺度**となっているようなクラス判別問題である．即ち，クラスラベル間に，整数や実数のような大小関係 class 0 < class 1 < ⋯ < class K が存在する．また，任意の二つのクラスラベルについて，大小関係が不明であったり，等しい関係にあったりしてはならない．一方，**間隔尺度**ではないため，class 0 と class 1 間の距離・類似度は class 1 と class 2 間のそれとは異なっていても構わない．

2.3 基底関数

本稿では，入力の特徴を実数として出力する M 個の基底関数 $\phi_m(x) \in \mathbb{R}$ $(m = 1, 2, ..., M)$ を用いる．入力 x は全て基底関数を通して使われ，基底関数なしに直接使われることはない．従って，適切な基底関数が用意できれば，入力 x が整数であるか実数であるか，またはスカラーであるかベクトルであるか，などには制約が無い．また，便宜上次の二つのベクトルをよく用いる．

$$\boldsymbol{\phi}_n \equiv [\phi_1(x_n), ..., \phi_M(x_n)]^\top \in \mathbb{R}^M \tag{1}$$

3. 既存手法

3.1 尤度関数

既存モデル [3] の概要を説明する．これは基底関数の線形和に対して正規ノイズが加わったものが，K 個の閾値 $\mathbf{b} \equiv [b_1, b_2, ..., b_K]^\top \in \mathbb{R}^K$（但し $b_1 < b_2 < ... < b_K$）を用いて，クラス判別されるものである．本稿では，正規ノイズの分散を 1 とする．

$$p\left(t_n \mid \epsilon_n, \mathbf{w}; \mathbf{b}\right) \equiv 1_{b_{t_n} \le \boldsymbol{\phi}_n^\top \mathbf{w} + \epsilon_n < b_{t_n+1}} \tag{2}$$

$$p(\epsilon_n) \equiv \mathcal{N}(\epsilon_n; 0, 1) \tag{3}$$

ここで，$\mathbf{w} \equiv [w_1, ..., w_M]^\top \in \mathbb{R}^M$ は線形和の重みベクトルである．また，便宜上 $b_0 \equiv -\infty$，$b_{K+1} \equiv +\infty$ を導入した．ノイズを周辺化除去すると，尤度関数

$$\begin{aligned} &p(t_n \mid \mathbf{w}; \mathbf{b}) \\ &= \int_{-\infty}^{+\infty} p(t_n \mid \epsilon_n, \mathbf{w}; \mathbf{b})\ p(\epsilon_n) \mathrm{d}\epsilon_n \\ &= \mathrm{cum}(b_{t_n+1} - \boldsymbol{\phi}_n^\top \mathbf{w}) - \mathrm{cum}(b_{t_n} - \boldsymbol{\phi}_n^\top \mathbf{w}) \end{aligned} \tag{4}$$

を得る．但し，cum は標準正規分布の累積分布関数である．このモデルは，$K = 1$(クラス数 2) の時にプロビット回帰と同等になる．

また異なるモデルとして，ロジスティックシグモイド関数を組み合わせたモデルも存在する [4]．これは，入力 x_n に対応するクラスラベル t_n の累積確率関数を

$$p(t_n \le k \mid \mathbf{w}; \mathbf{b}) \equiv \sigma(b_{k+1} - \boldsymbol{\phi}_n^\top \mathbf{w}) \tag{5}$$

と定義するものである．すると，尤度関数

$$p(t_n \mid \mathbf{w}; \mathbf{b}) = \sigma(b_{t_n+1} - \boldsymbol{\phi}_n^\top \mathbf{w}) - \sigma(b_{t_n} - \boldsymbol{\phi}_n^\top \mathbf{w}) \tag{6}$$

を得る．これは (4) と同じ形になっており，先のモデルでノイズの分布をロジスティックシグモイド関数を累積密度関数に持つような分布（平均 0，分散 $\frac{\pi^2}{3}$ のロジスティック分布）としたものと同等である．

$$p(\epsilon_n) \equiv \frac{\mathrm{e}^{-\epsilon_n}}{(1+\mathrm{e}^{-\epsilon_n})^2} \tag{7}$$

3.2 事前分布

尤度 (4) や (6) において，重み $\mathbf{w}$ の事前分布を，精度パラメータ $\boldsymbol{\alpha} \equiv [\alpha_1, ..., \alpha_M]^\top \in (\mathbb{R}^+)^M$ を用いて

$$p(\mathbf{w}; \boldsymbol{\alpha}) \equiv \prod_{m=1}^{M} \mathcal{N}(w_m; 0, \alpha_m^{-1}) \tag{8}$$

と定義する．これは automatic relevance determination (ARD) 事前分布と呼ばれ，Tipping が relevance vector machine (RVM) で用いたものと同じである [6].

重みが与えられた下では，各入出力標本は独立であると定義する．すると，全ての確率変数についての同時分布は次式で与えられる．

$$p(t_{N+1}, \mathbf{t}, \mathbf{w}; \mathbf{b}, \boldsymbol{\alpha}) = \left(\prod_{n=1}^{N+1} p\left(t_n \mid \mathbf{w}; \mathbf{b}\right) \right) p(\mathbf{w}; \boldsymbol{\alpha}) \tag{9}$$

同時分布が分かれば，事後分布，予測分布などの全ての周辺分布，条件付分布を導出することができる．

3.3 基準・推定量

ここからは，モデル (6) についての例を示す．(4) に対しては，σ を cum に置き換えて考えれば良い．Bayes 推定の枠組みでは，推定すべきクラスラベル t_{N+1} の分布として最良のものは予測分布である．

$$p(t_{N+1} \mid \mathbf{t}; \mathbf{b}^\star, \boldsymbol{\alpha}^\star) \tag{10}$$

但し，分布を持たないパラメータは周辺化で消去することができないため，第二種最尤推定により求める．

$$\{\mathbf{b}^\star, \boldsymbol{\alpha}^\star\} \equiv \underset{\mathbf{b}, \boldsymbol{\alpha}}{\operatorname{argmax}}\, p(\mathbf{t}; \mathbf{b}, \boldsymbol{\alpha}) \tag{11}$$

本モデルについては，通常いくつかの $\alpha_m^\star$ が無限大に発散する．このような場合，対応する重み w_m の事前分布が Dirac のデルタ関数になることに相当し，$w_m = 0$ を意味する．すると，対応する基底関数 $\phi_m(\cdot)$ は存在しないことと同等になり，結果的に疎な解が得られる．

一方，最も適切な重み $\mathbf{w}$ を推定（点推定）したいこともある．この場合，最適性の基準は唯一ではないが，一般によく用いられるものの一つに最大事後確率（maximum a posteriori: MAP）推定が挙げられる．

$$\mathbf{w}^\star \equiv \underset{\mathbf{w}}{\operatorname{argmax}}\, p(\mathbf{w} \mid \mathbf{t}; \mathbf{b}^\star, \boldsymbol{\alpha}^\star) \tag{12}$$

また，これを経由して t_{N+1} の分布を推定する方法もある．

$$p(t_{N+1} \mid \mathbf{w}^\star) \tag{13}$$

この分布は予測分布に比べると不正確であるが，予測分布より簡単な計算で求まることが多く，多用される．

t_{N+1} を点推定したい場合は，上記の予測分布や $\mathbf{w}^\star$ を介した分布を MAP 推定する方法などがある．

$$t_{N+1}^\star \equiv \underset{t_{N+1}}{\operatorname{argmax}}\, p(t_{N+1} \mid \mathbf{t}; \mathbf{b}^\star, \boldsymbol{\alpha}^\star) \tag{14}$$

$$t_{N+1}^\star \equiv \underset{t_{N+1}}{\operatorname{argmax}}\, p(t_{N+1} \mid \mathbf{w}^\star) \tag{15}$$

これの最適性も $t_{N+1}^\star$ を何に使うかによるため唯一でない．しかし，一般に点推定したいものが離散確率変数の場合は，確率質量を最大にするという意味で，MAP 推定量は良い推定量である．

3.4 最適化

ここでは，$\mathbf{w}^\star$ を経由する方法に焦点を当て，近似的に $\mathbf{b}^\star$，$\boldsymbol{\alpha}^\star$，$\mathbf{w}^\star$ を求める．本アルゴリズムは，適当な初期値 $\mathbf{b}^{(0)}$，$\boldsymbol{\alpha}^{(0)}$ を設定した後，以下の $\mathbf{w}^{(i)}$ と $\{\mathbf{b}^{(i)}, \boldsymbol{\alpha}^{(i)}\}$ の交互最適化を $i = 0, 1, ...$ と繰り返すことにより，これらが真の最適解 $\mathbf{b}^\star$，$\boldsymbol{\alpha}^\star$，$\mathbf{w}^\star$ へと収束することを期待するものである．

まず，$\mathbf{b}$，$\boldsymbol{\alpha}$ の値を固定したうえで，$\mathbf{w}$ の値を最適化する．

$$\mathbf{w}^{(i+1)} \equiv \underset{\mathbf{w}}{\operatorname{argmax}}\, p(\mathbf{w} \mid \mathbf{t}; \mathbf{b}^{(i)}, \boldsymbol{\alpha}^{(i)}) \tag{16}$$

この最適化は解析的には困難であるため，実際には数値的な最適化，例えば Newton-Raphson 法を用いる．具体的には，初期値を $\mathbf{w}^{(i,0)} \equiv \mathbf{w}^{(i)}$ として，次の更新式を $j = 0, 1, ...$ と繰り返せばよい．

$$\mathbf{w}^{(i,j+1)} \equiv \mathbf{w} - \mathbf{H}(\mathbf{w}; \mathbf{b}, \boldsymbol{\alpha})^{-1} \mathbf{h}(\mathbf{w}; \mathbf{b}, \boldsymbol{\alpha}) \Big|_{\bullet}$$
$$\bullet : \mathbf{w} = \mathbf{w}^{(i,j)}, \mathbf{b} = \mathbf{b}^{(i)}, \boldsymbol{\alpha} = \boldsymbol{\alpha}^{(i)} \tag{17}$$

$$h(\mathbf{w}; \mathbf{b}, \boldsymbol{\alpha}) \equiv -\ln p(\mathbf{t}, \mathbf{w}; \mathbf{b}, \boldsymbol{\alpha}) \tag{18}$$

$$\mathbf{h}(\mathbf{w}; \mathbf{b}, \boldsymbol{\alpha}) \equiv \frac{\partial h(\mathbf{w}; \mathbf{b}, \boldsymbol{\alpha})}{\partial \mathbf{w}} = \sum_{n=1}^{N} \frac{s_1}{s_0} \boldsymbol{\phi}_n + \mathbf{A}\mathbf{w} \tag{19}$$

$$\mathbf{H}(\mathbf{w}; \mathbf{b}, \boldsymbol{\alpha}) \equiv \frac{\partial^2 h(\mathbf{w}; \mathbf{b}, \boldsymbol{\alpha})}{\partial \mathbf{w} \partial \mathbf{w}^\top} \tag{20}$$

$$= \sum_{n=1}^{N} \left(\left[\frac{s_1}{s_0} \right]^2 - \frac{s_2}{s_0} \right) \boldsymbol{\phi}_n \boldsymbol{\phi}_n^\top + \mathbf{A} \tag{21}$$

$$s_0 \equiv \sigma(b_{t_n+1} - \boldsymbol{\phi}_n^\top \mathbf{w}) - \sigma(b_{t_n} - \boldsymbol{\phi}_n^\top \mathbf{w}) \tag{22}$$

$$s_1 \equiv \sigma'(b_{t_n+1} - \boldsymbol{\phi}_n^\top \mathbf{w}) - \sigma'(b_{t_n} - \boldsymbol{\phi}_n^\top \mathbf{w}) \tag{23}$$

$$s_2 \equiv \sigma''(b_{t_n+1} - \boldsymbol{\phi}_n^\top \mathbf{w}) - \sigma''(b_{t_n} - \boldsymbol{\phi}_n^\top \mathbf{w}) \tag{24}$$

但し，目的関数を同等な式 (18) に置き換えた．先に定義した二つのノイズモデル（標準正規分布とロジスティック分布）については，Hesse 行列 $\mathbf{H}(\mathbf{w}; \mathbf{b}, \boldsymbol{\alpha})$ は $\mathbf{w}$ の全域に亘って正定値であるため（但し，$[\boldsymbol{\phi}_1, ..., \boldsymbol{\phi}_N]$ のランクが M 以上という条件あり），比較的少ない繰り返し数により精度よく $\mathbf{w}^{(i+1)}$ を求めることができる．

次に，$p(\mathbf{t}; \mathbf{b}, \boldsymbol{\alpha})$ を Laplace 法により近似した後，$\mathbf{b}$，$\boldsymbol{\alpha}$ を最適化する．Laplace 法とは，一般に関数 $f(\mathbf{w})$，$\mathbf{w} \in \mathbb{R}^M$ に関する次の積分を最小点周りの 2 次の Taylor 近似より解くものである（ここで用いた変数名，関数名は他と関係ない）．

$$\int_{\mathbb{R}^M} \mathrm{e}^{-f(\mathbf{w})}\mathrm{d}\mathbf{w} \simeq \int_{\mathbb{R}^M} \mathrm{e}^{-f(\boldsymbol{\mu})-\frac{1}{2}[\mathbf{w}-\boldsymbol{\mu}]^\top \mathbf{F}[\mathbf{w}-\boldsymbol{\mu}]}\mathrm{d}\mathbf{w} = \frac{\mathrm{e}^{-f(\boldsymbol{\mu})}}{\sqrt{\left|\frac{1}{2\pi}\mathbf{F}\right|}}$$

$$\boldsymbol{\mu} \equiv \underset{\mathbf{w}}{\mathrm{argmin}} f(\mathbf{w}), \quad \mathbf{F} \equiv \left.\frac{\partial f(\mathbf{w})}{\partial \mathbf{w}\partial \mathbf{w}^\top}\right|_{\mathbf{w}=\boldsymbol{\mu}}$$

具体的には次のように近似することで $\mathbf{b}$，$\boldsymbol{\alpha}$ を最適化する．

$$p(\mathbf{t};\mathbf{b},\boldsymbol{\alpha}) = \int_{\mathbb{R}^M} p(\mathbf{t},\mathbf{w};\mathbf{b},\boldsymbol{\alpha})\mathrm{d}\mathbf{w} \simeq s(\mathbf{b},\boldsymbol{\alpha}) \quad (25)$$

$$s(\mathbf{b},\boldsymbol{\alpha}) \equiv \frac{p(\mathbf{t},\mathbf{w}^{(i+1)};\mathbf{b},\boldsymbol{\alpha})}{\sqrt{\left|\frac{1}{2\pi}\mathbf{H}(\mathbf{w}^{(i+1)};\mathbf{b},\boldsymbol{\alpha})\right|}} \quad (26)$$

ここで用いた近似は，標準的な Laplace 法ではない．例えば，Taylor 近似する際の基点は最大点とするのが本来の方法であるが，近似的に $\mathbf{w}^{(i+1)}$ で代用している．また，これにより Taylor 近似による 1 次の項が一般に 0 でなくなるが，この項も無視している．ここで $\mathbf{b}$ と $\boldsymbol{\alpha}$ 最適化順などについては提案された論文では触れられていないため，下記は本稿の比較実験で行った方法である．

$$\boldsymbol{\alpha}^{(i+1)} \equiv \underset{\boldsymbol{\alpha}}{\mathrm{argmax}}\, s(\mathbf{b}^{(i)},\boldsymbol{\alpha}) \quad (27)$$

$$\mathbf{b}^{(i+1)} \equiv \underset{\mathbf{b}}{\mathrm{argmax}}\, s(\mathbf{b},\boldsymbol{\alpha}^{(i+1)}) \quad (28)$$

ここでの最適化も解析的には解けないため，$\boldsymbol{\alpha}$ については自己無撞着方程式による更新で近似する（自己無撞着方程式の作り方は一意ではない）．

$$\alpha_m^{(i+1)} \equiv \frac{1-\alpha_m^{(i)}[\mathbf{H}(\mathbf{w}^{(i+1)},\mathbf{b}^{(i)},\boldsymbol{\alpha}^{(i)})^{-1}]_{m,m}}{w_m^2} \quad (29)$$

$\mathbf{b}$ については，勾配が求まるため勾配法を用いる．初期値を $\mathbf{b}^{(i,0)} \equiv \mathbf{b}^{(i)}$ として，次の更新式を $j = 0, 1, ...$ と繰り返せばよい．

$$b_k^{(i,j+1)} \equiv b_k - \eta \sum_{n=1}^{N} \left.\frac{(1_{t_n+1=k} - 1_{t_n=k})\sigma'(b_k - \boldsymbol{\phi}_n^\top \mathbf{w})}{\sigma(b_{t_n+1} - \boldsymbol{\phi}_n^\top \mathbf{w}) - \sigma(b_{t_n} - \boldsymbol{\phi}_n^\top \mathbf{w})}\right|_{\bullet}$$
$$\bullet : \mathbf{w} = \mathbf{w}^{(t+1)}, \mathbf{b} = \mathbf{b}^{(i,j)} \quad (30)$$

$\eta > 0$ は勾配法のステップ幅で，例えば 10^{-4} などとする．

4. 提案手法

4.1 尤度関数

既存モデルに比べて比較的解析が容易な尤度を定義する．

$$p(t_n \mid \mathbf{w},\mathbf{b}) \equiv \frac{1}{Z_n}\left[\prod_{k=1}^{t_n} y_{n,k}\right]\prod_{k=t_n+1}^{K}(1-y_{n,k}) \quad (31)$$

$$Z_n \equiv \sum_{t_n=0}^{K}\left[\prod_{k=1}^{t_n} y_{n,k}\right]\prod_{k=t_n+1}^{K}(1-y_{n,k}) \quad (32)$$

$$y_{n,k} \equiv \sigma(\boldsymbol{\phi}_n^\top \mathbf{w} + b_k) \quad (33)$$

ここで Z_n は正規化定数を表す．Z_n は重み $\mathbf{w} \equiv [w_1, ..., w_M]^\top \in \mathbb{R}^M$ やバイアス $\mathbf{b} \equiv [b_1, ..., b_K]^\top \in \mathbb{R}^K$ に依存するが，バイアスの値 b_k が互いに 3 程度離れていればほぼ 1 となるため，以後常に 1 と近似する．

4.2 事前分布

ここでは，[7] に倣い，ARD 事前分布 (8) を導入する．重み $\mathbf{w}$ に対する精度パラメータ $\boldsymbol{\alpha} \equiv [\alpha_1, ..., \alpha_M]^\top \in (\mathbb{R}^+)^M$ のみならず，バイアス $\mathbf{b}$ についても精度パラメータ $\boldsymbol{\beta} \equiv [\beta_1, ..., \beta_K]^\top \in (\mathbb{R}^+)^K$ を導入する．

$$\boldsymbol{\omega} \equiv [\mathbf{w}^\top, \mathbf{b}^\top]^\top \equiv [\omega_1, ..., \omega_{M+K}]^\top \quad (34)$$

$$\boldsymbol{\chi} \equiv [\boldsymbol{\alpha}^\top, \boldsymbol{\beta}^\top]^\top \equiv [\chi_1, ..., \chi_{M+K}]^\top \quad (35)$$

$$p(\boldsymbol{\omega};\boldsymbol{\chi}) \equiv \prod_{m=1}^{M+K} \mathcal{N}(\omega_m; 0, \chi_m^{-1}) \quad (36)$$

また，必要に応じて $\mathbf{C} \equiv \mathrm{diag}(\boldsymbol{\chi})$ を用いる．

重みが与えられたもとでは，各入出力標本は独立であるとモデル化する．すると，全ての確率変数についての同時分布は

$$p(t_{N+1}, \mathbf{t}, \boldsymbol{\omega};\boldsymbol{\chi}) = \left(\prod_{n=1}^{N+1} p(t_n \mid \boldsymbol{\omega})\right) p(\boldsymbol{\omega};\boldsymbol{\chi}) \quad (37)$$

で与えられる．同時分布が分かれば，事後分布，予測分布などの全ての周辺分布，条件付分布を導出することができる．

4.3 基準・推定量

ここでは，$\boldsymbol{\omega}$ の MAP 推定を介した二段階最適化を行う．具体的には，分布を持たないパラメータは第二種最尤推定し，この値を用いて $\boldsymbol{\omega}$ の MAP 推定を行い，この結果を用いて t_{N+1} の MAP 推定を行う．

$$\boldsymbol{\chi}^\star \equiv \underset{\boldsymbol{\chi}}{\mathrm{argmax}}\, p(\mathbf{t};\boldsymbol{\chi}) \quad (38)$$

$$\boldsymbol{\omega}^\star \equiv \underset{\boldsymbol{\omega}}{\mathrm{argmax}}\, p(\boldsymbol{\omega} \mid \mathbf{t};\boldsymbol{\chi}^\star) \quad (39)$$

$$t_{N+1}^\star \equiv \underset{t_{N+1}}{\mathrm{argmax}}\, p(t_{N+1} \mid \boldsymbol{\omega}^\star) \quad (40)$$

4.4 最適化

本アルゴリズムは，適当な初期値 $\boldsymbol{\chi}^{(0)}$ を設定した後，以下の $\boldsymbol{\omega}^{(i)}$ と $\boldsymbol{\chi}^{(i)}$ の交互最適化を $i = 0, 1, ...$ と繰り返すことにより，これらが真の最適解 $\boldsymbol{\chi}^\star$，$\boldsymbol{\omega}^\star$ へと収束すること

を期待するものである．

まず，$\boldsymbol{\chi}$ の値を固定したうえで，$\boldsymbol{\omega}$ の値を最適化する．

$$\boldsymbol{\omega}^{(i+1)} \equiv \underset{\boldsymbol{\omega}}{\operatorname{argmax}}\, p(\boldsymbol{\omega} \mid \mathbf{t}; \boldsymbol{\chi}^{(i)}) \tag{41}$$

この最適化は解析的には困難であるため，Newton-Raphson 法を用いる．具体的には，初期値を $\boldsymbol{\omega}^{(i,0)} \equiv \boldsymbol{\omega}^{(i)}$ として，次の更新式を $j = 0, 1, \ldots$ と繰り返せばよい．

$$\boldsymbol{\omega}^{(i,j+1)} \equiv \boldsymbol{\omega} - \mathbf{H}(\boldsymbol{\omega}; \boldsymbol{\chi})^{-1}\mathbf{h}(\boldsymbol{\omega}; \boldsymbol{\chi})\big|_{\bullet}$$
$$\bullet : \boldsymbol{\omega} = \boldsymbol{\omega}^{(i,j)}, \boldsymbol{\chi} = \boldsymbol{\chi}^{(i)} \tag{42}$$

$$h(\boldsymbol{\omega}; \boldsymbol{\chi}) \equiv -\ln p(\mathbf{t}, \boldsymbol{\omega}; \boldsymbol{\chi}) \tag{43}$$

$$\mathbf{h}(\boldsymbol{\omega}; \boldsymbol{\chi}) \equiv \frac{\partial h(\boldsymbol{\omega}; \boldsymbol{\chi})}{\partial \boldsymbol{\omega}} = \boldsymbol{\Psi}^\top \mathbf{d} + \mathbf{C}\boldsymbol{\omega} \tag{44}$$

$$\mathbf{H}(\boldsymbol{\omega}; \boldsymbol{\chi}) \equiv \frac{\partial^2 h(\boldsymbol{\omega}; \boldsymbol{\chi})}{\partial \boldsymbol{\omega} \partial \boldsymbol{\omega}^\top} = \boldsymbol{\Psi}^\top \mathbf{D} \boldsymbol{\Psi} + \mathbf{C} \tag{45}$$

$$\mathbf{d} \equiv \begin{pmatrix} \left[\sum_{k=1}^{K} y_{n,k} - t_n\right]_{n=1}^{N} \\ \left[\sum_{n=1}^{N} (y_{n,k} - 1_{k \le t_n})\right]_{k=1}^{K} \end{pmatrix} \tag{46}$$

$$\boldsymbol{\Psi} \equiv \begin{pmatrix} \boldsymbol{\Phi} & \mathbf{0}_{N,K} \\ \mathbf{0}_{K,M} & \mathbf{I}_K \end{pmatrix}, \quad \boldsymbol{\Phi} \equiv \begin{pmatrix} \boldsymbol{\phi}_1^\top \\ \vdots \\ \boldsymbol{\phi}_N^\top \end{pmatrix} \tag{47}$$

$$\mathbf{D} \equiv \begin{pmatrix} \mathbf{D}_N & \mathbf{D}_{N,K} \\ \mathbf{D}_{N,K}^\top & \mathbf{D}_K \end{pmatrix} \tag{48}$$

$$[\mathbf{D}_{N,K}]_{n,k} \equiv y'_{n,k} \equiv \sigma'(\boldsymbol{\phi}_n^\top \mathbf{w} + b_k) \tag{49}$$

$$\mathbf{D}_N \equiv \operatorname{diag}\left(\left[\sum_{k=1}^{K} y'_{n,k}\right]_{n=1}^{N}\right) \tag{50}$$

$$\mathbf{D}_K \equiv \operatorname{diag}\left(\left[\sum_{n=1}^{N} y'_{n,k}\right]_{k=1}^{K}\right) \tag{51}$$

但し，目的関数を同等な式 (43) に置き換えた．また，ここに記載した勾配，Hesse 行列の具体的な式は，$y_{n,k}$ がロジスティックシグモイド関数の時にのみ成立する表記になっているため，これを別の関数に置き換える時は注意されたい．また，$\mathbf{d}$，$\mathbf{D}$，$y_{n,k}$，$y'_{n,k}$ は $\boldsymbol{\omega}$ に依存して変化することに注意されたい．

次に $\boldsymbol{\chi}$ を最適化する．具体的には，まず $-\ln p(\mathbf{t} \mid \boldsymbol{\omega})$ を $\boldsymbol{\omega}$ に関して $\boldsymbol{\omega}^{(t+1)}$ 周りで 2 次の Taylor 近似を行う．

$$g(\boldsymbol{\omega}) \equiv -\ln p(\mathbf{t} \mid \boldsymbol{\omega}) \tag{52}$$

$$\mathbf{g}(\boldsymbol{\omega}) \equiv \frac{\partial g(\boldsymbol{\omega})}{\partial \boldsymbol{\omega}} = \boldsymbol{\Psi}^\top \mathbf{d} \tag{53}$$

$$\mathbf{G}(\boldsymbol{\omega}) \equiv \frac{\partial^2 g(\boldsymbol{\omega})}{\partial \boldsymbol{\omega} \partial \boldsymbol{\omega}^\top} = -\boldsymbol{\Psi}^\top \mathbf{D} \boldsymbol{\Psi} \tag{54}$$

$$g(\boldsymbol{\omega}) \simeq g(\check{\boldsymbol{\omega}}) + [\boldsymbol{\omega} - \check{\boldsymbol{\omega}}]^\top \mathbf{g}(\check{\boldsymbol{\omega}}) + \frac{1}{2}[\boldsymbol{\omega} - \check{\boldsymbol{\omega}}]^\top \mathbf{G}(\check{\boldsymbol{\omega}})[\boldsymbol{\omega} - \check{\boldsymbol{\omega}}] \tag{55}$$

$$= -\ln \mathcal{N}(\check{\boldsymbol{\tau}}; \boldsymbol{\Psi}\boldsymbol{\omega}, \check{\mathbf{D}}^{-1}) + \text{const} \tag{56}$$

$$\check{\boldsymbol{\tau}} \equiv \boldsymbol{\Psi}\check{\boldsymbol{\omega}} - \check{\mathbf{D}}^{-1}\check{\mathbf{d}} \tag{57}$$

ここで，$\check{\bullet}$ を付けた変数は，$\boldsymbol{\omega}$ に $\boldsymbol{\omega}^{(t+1)}$ の値を使用することを表す．また，const は $\boldsymbol{\omega}$，$\boldsymbol{\chi}$ に関して定数．上記近似式を用いて $p(\mathbf{t}; \boldsymbol{\chi})$ を扱い易い形に変形する．

$$\begin{aligned} p(\mathbf{t}; \boldsymbol{\chi}) &= \int_{\mathbb{R}^{M+K}} p(\mathbf{t} \mid \boldsymbol{\omega}) p(\boldsymbol{\omega}; \boldsymbol{\chi}) \mathrm{d}\boldsymbol{\omega} \\ &\simeq \int_{\mathbb{R}^{M+K}} \mathcal{N}(\check{\boldsymbol{\tau}}; \boldsymbol{\Psi}\boldsymbol{\omega}, \check{\mathbf{D}}^{-1}) \mathcal{N}(\boldsymbol{\omega}; \mathbf{0}, \mathbf{C}^{-1}) \mathrm{d}\boldsymbol{\omega} \\ &= \mathcal{N}(\check{\boldsymbol{\tau}}; \mathbf{0}, \check{\mathbf{D}}^{-1} + \boldsymbol{\Psi}\mathbf{C}^{-1}\boldsymbol{\Psi}^\top) \end{aligned} \tag{58}$$

これを最大化する $\boldsymbol{\chi}$ を求める．$\boldsymbol{\chi}$ ベクトル全体を一度に最適化するのは困難であるので，各次元ごとに最適化することにする．

$$\chi_m^{(i+1)} \equiv \underset{\chi_m}{\operatorname{argmax}}\, \mathcal{N}(\check{\boldsymbol{\tau}}; \mathbf{0}, \check{\mathbf{D}}^{-1} + \boldsymbol{\Psi}\mathbf{C}^{-1}\boldsymbol{\Psi}^\top) \tag{59}$$

これは解析的に解けて，次の更新式を得る．

$$\chi_m^{(i+1)} = \begin{cases} \frac{s_m^2}{q_m^2 - s_m} & (q_m^2 > s_m) \\ \infty & (\text{otherwise}) \end{cases} \tag{60}$$

$$s_m \equiv \frac{\boldsymbol{\psi}_m^\top \mathbf{E}^{-1} \boldsymbol{\psi}_m}{1 - \left(\chi_m^{(i)}\right)^{-1} \boldsymbol{\psi}_m^\top \mathbf{E}^{-1} \boldsymbol{\psi}_m} \tag{61}$$

$$q_m \equiv \frac{\boldsymbol{\psi}_m^\top \mathbf{E}^{-1} \boldsymbol{\tau}}{1 - \left(\chi_m^{(i)}\right)^{-1} \boldsymbol{\psi}_m^\top \mathbf{E}^{-1} \boldsymbol{\psi}_m} \tag{62}$$

$$\mathbf{E} \equiv \check{\mathbf{D}}^{-1} + \boldsymbol{\Psi}\left(\mathbf{C}^{(i)}\right)^{-1} \boldsymbol{\Psi}^\top \tag{63}$$

ここで，$\boldsymbol{\psi}_m$ は $\boldsymbol{\Psi}$ の m 番目の列ベクトルを表す．また，(61) と (62) は $\mathbf{E}$ の定義 (63) から，$\chi_m^{(i)}$ に依存するように見えるが，展開すると消去されることに注意されたい．また，$\mathbf{D}$，$\mathbf{C}$ がそれぞれ $\boldsymbol{\omega}^{(i+1)}$，$\boldsymbol{\chi}^{(i)}$ に依存するため，更新ステップ毎に $\mathbf{E}$ を計算し直す必要がある．また，近似周辺尤度 (58) は重み $\boldsymbol{\omega}$ に依存して決まるため，χ_m が変化した際は再び $\boldsymbol{\omega}$ の更新を行う必要がある．

4.5 計算量

精度 α_m が無限大のとき，対応する重み w_m は 0 となり，基底 $\phi_m(\cdot)$ はモデルに影響を与えない．従って，計算にも影響を与えないことになる．各更新ステップにおいて，重みが非零の基底番号の集合 $\mathcal{S}^{(i)}$ を定義する．

$$\mathcal{S}^{(i)} \equiv \left\{ m \in \{1, \cdots, K+M\} \mid \chi_m^{(i)} \neq \infty \right\} \tag{64}$$

同様に，$\mathcal{S}^{(i)}$ 番目の χ から成る対角行列を $\mathbf{C}_{\mathcal{S}}$，$\mathcal{S}$ 番目の基底から成る基底行列を $\boldsymbol{\Psi}_{\mathcal{S}}$，$\mathcal{S}$ 番目の重みから成るベクトルを $\boldsymbol{\omega}_{\mathcal{S}}$，$M_{\mathcal{S}} \equiv \left|\mathcal{S}^{(i)}\right|$ とする．これらは全て $\mathcal{S}^{(i)}$ に依存するため，ステップごとに異なることに注意されたい．

パラメータの更新式 (42) は，一般に $\mathcal{O}\left((M+K)^3 + (M+K)N^2\right)$ の計算量を必要とするが，疎性を利用すると $\mathcal{O}\left(\left(M_{\mathcal{S}}^{(i)}\right)^3 + M_{\mathcal{S}}^{(i)} N^2\right)$ で十分である．最終的に，(42) は，

$$\boldsymbol{\omega}_{\mathcal{S}}^{(i,j+1)} = \left(\boldsymbol{\Psi}_{\mathcal{S}}^{\top}\check{\mathbf{D}}\boldsymbol{\Psi}_{\mathcal{S}} + \mathbf{C}_{\mathcal{S}}\right)^{-1}\left(\boldsymbol{\Psi}_{\mathcal{S}}^{\top}\check{\mathbf{D}}\boldsymbol{\Psi}_{\mathcal{S}}\check{\boldsymbol{\omega}}_{\mathcal{S}} + \boldsymbol{\Psi}_{\mathcal{S}}^{\top}\check{\mathbf{d}}\right) \tag{65}$$

となる．ここで，$\check{\bullet}$ は j に応じて $\boldsymbol{\omega}^{(i,j)}$ と共に変化し，それ以外は更新ステップ i に応じて変化することに注意されたい．

ここまでに，各更新ステップにおいて，$M_{\mathcal{S}}$ に応じて計算量が変化することを示した．これは，既存手法 [4] においても同様である．提案手法の既存手法に対する大きな違いは，基底の追加が可能なことである．具体的には，既存手法では一度発散した精度パラメータは有限に戻ることは無いが，提案手法では起こり得る．これにより，少ない数の基底のみ含む状態を初期値として，適宜追加していくアルゴリズムを用いることが可能になる．多くの場合，全ての基底を含む状態を初期値とする場合に比べて，少ない計算時間で済む．

5. 実験

ここでは，提案手法の精度と計算時間について，既存手法と比較して評価する．比較手法として，モデル (4) に対して等分散の重みを事前分布とした場合 (GPOR)[3]，ARD 事前分布を用いた既存モデル (ORSB)，クラス間の順序関係を考慮せず，クラス毎に異なった重みを定義し，ARD 事前分布を利用して提案手法と類似の解き方をしたモデル (SoftMax) との比較を行った．比較は，10-fold cross validation を行い，予測分布で最も確率の高かったクラスが真のクラスと一致した割合を調べた．

5.1 人工データ

人工データの生成方法を説明する．まず，N 個の M 次元ベクトルを標準正規分布に従い生成し入力データとする．そのうち $M_{\mathcal{S}}$ 個の次元について重みを 0 から 1 の一様分布で生成する．入力データの重み付き和を各入力のスコアとし，スコアの最小値から最大値までを $K+1$ 等分し，データが何番目に入ったかを所属クラスとした．

図 1 に人工データについての結果を示す．汎化誤差（左図），学習時間（右図）共に比較手法よりも良い結果となった．

5.2 ベンチマーク

順序回帰へのベンチマークとして，[3] で使用されているデータを用いた．手法の比較であるため，単純に線形の基底関数を用いることとし，そのために入力が実数であるデータのみ比較を行った．また，データによっては比較手法の計算が終了しなかった．ここでは，一定時間内に比較手法のうち一つ以上が計算終了したデータについてのみ記載する．各データの汎化誤差を表 1 に示す．誤差の標準偏差を含めると，提案手法が安定して既存手法を上回るとは限らないが，大きく劣るケースは見られない．

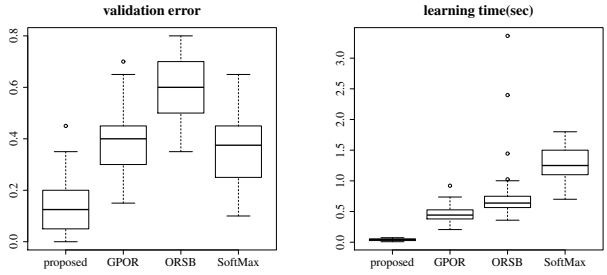

図 1 $N=100$，$M=10$，$M_{\mathcal{S}}=3$，$K=3$ のデータを 50 個生成し，それぞれ 10-fold クロスバリデーションを行った結果．汎化誤差 (左図) と学習時間 (右図)．

	Diabetes	pyrimidines	traiazines	wisconsin
proposed	0.49±0.10	0.51±0.09	0.56±0.04	0.69±0.05
GPOR	0.86±0.19	0.60±0.23	0.81±0.26	0.68±0.05
ORSB	0.57±0.24	0.81±0.17	0.70±0.04	0.88±0.07
SoftMax	0.49±0.04	-	-	-

表 1 ベンチマークの汎化誤差

この中で，ORSB は全ての基底が除外される場合があった．また，他のデータでは GPOR と SoftMax の計算が時間内に終了しないことが多かった．

6. 議論

ほとんどの実験で，提案手法の実行時間は既存手法を大きく下回る結果になった．この結果は特にデータが多い場合に顕著に現れ，他の手法が数時間掛かるデータに対して数秒程度で収束した．精度面でも比較手法に勝るか同等程度で，実用に際しても十分に有用な手法であることが解った．

参考文献

[1] Peter McCullagh, Regression Models for Ordinal Data, Journal of the Royal Statistical Society. Series B (Methodological), pp 109-142, 1980.
[2] Cande V Ananth and David G Kleinbaum, Regression models for ordinal responses: a review of methods and applications, International journal of epidemiology, 1997.
[3] Wei Chu and Zoubin Ghahramani, Gaussian Processes for Ordinal Regression, Technical Report, UCL, UK, 2004.
[4] Xiao Chang, Qinghua Zheng and Peng Lin, Ordinal Regression with Sparse Bayesian, Emerging Intelligent Computing Technology and Applications. With Aspects of Artificial Intelligence, Lecture Notes in Computer Science Volume 5755, 2009, pp 591-599
[5] C.M.Bishop, Pattern Recognition and Machine Learning, Springer, 2006.
[6] Michael E Tipping,Sparse bayesian learning and the relevance vector machine, The Journal of Machine Learning Research archive Volume 1, 2001
[7] Michael E. Tipping and Anita C. Faul, Fast marginal likelihood maximisation for sparse Bayesian models, Microsoft Research,Cambridge,U,K.,2003