需要予測のための統計ﾓﾃﾞﾙの研究

異常値検知のための基本的ﾓﾃﾞﾙの考察

平成 26 年 3 月

東京大学大学院
情報理工学系研究科　教授　竹村 彰通
博士課程　小川 光紀
修士課程　笹井 健行

特定非営利活動法人
ﾋﾞｭｰ･ｺﾐｭﾆｹｰｼｮﾝｽﾞ　副理事長　小松 秀樹
主任研究員　池谷 貞彦
研究員　河内 敦雄

目 次

第１章　はじめに

1.1　本共同研究の目的

企業にとって，販売に関する時系列データの統計的な解析によって在庫量を動的に適切に調整することは重要である．時系列解析の手法としては ARIMA モデル (自己回帰和分移動平均モデル) が標準的であり，ARIMA モデルの活用によって在庫管理の合理化が期待される．しかしながら実際の販売データには異常値 (はずれ値) や構造変化が含まれる場合があり，そのような場合にはモデルの変更を含めた検討が必要となり，ARIMA モデルの機械的なあてはめだけでは不十分となる．

予測残差の中であるしきい値以上の値が現れた場合には，異常値であるかの検討が必要となる．あい続く複数の予測残差に，分散の増加など何等かのそれまでと異なる傾向が現れた場合には，構造変化 (モデル自体の変化) を検討する必要がある．

異常値や構造変化がおこる原因は，天候，ニュースとなるような事件，一般的な経済状況の変化など，さまざまな外的なものも考えられるため，注目している時系列データの分析だけにはとどまらないこともある．また，注目している時系列データ解析の範囲に限ってみても，しきい値の具体的な設定のための確率計算が複雑であり，簡便な方法がない現状である．本研究では，異常値検出や構造変化検出の実際上の重要性に鑑みて，これらの手法のサーベイをおこなうとともに，実際に売り上げデータへの適用例を示す．

1.2　報告書の構成

本報告書の２章では時系列解析，特に最も簡便なモデルである自己回帰モデル (AR モデル) に関して必要となる事項をまとめている．異常値検出のためには複雑な計算が必要とされるため，想定するモデルとしてまず AR モデルを用いることが考えられる．３章では単一の時系列の時間的な相関を表す自己相関関数，および複数の時系列の間の相関を表す相互相関係数について説明する．これらの相関係数に顕著なパターンが見られれば，モデルを用いることなく異常値の検討が可能となる可能性がある．しかしながら，異常値の検出はやさしい問題ではなく，モデルに基づく方法が必要となる．４章では異常値の検出および構造変化の検出に関して，統計的検討に基づくよりも

計算量の面で有利な方法である Change Finder とよばれる手法について説明している．この方法は AR モデルの仮定のもとで，異常値検出の方法と平滑化を組み合わせることによって，異常値検出と同時に構造変化の検出もおこなうことができ，すぐれた方法である．最後に 5 章では Change Finder を実際の複数の売り上げデータに応用した分析例を示し，この方法の有用性を検証する．

第2章　時系列解析の基本

本章では，まず時系列分析の基本的な枠組みを確認する．次に，今回の分析に用いるアルゴリズムにおいて重要な役割を果たす AR モデルを解説する [1].

2.1　時系列解析の基本

2.1.1　確率過程

考える時点の集合を T とする．時点 $t \in T$ においてある事象が観測される確率は確率密度関数 x_t に従うとする．さらに，確率変数 $p(x_t)$ の集合を $\{p(x_t)\}_{t \in T}$ と表す．このような時刻 t に関する確率変数の系列を確率過程と呼ぶ．本報告書で用いる確率過程は，$T = \{1, 2, ...\}$ である離散的確率過程と呼ばれるものである．

2.1.2　定常過程

離散的確率過程 $\{x_t\}$ が次の二つの性質を持つとき，$\{x_t\}$ は弱定常過程であるという．

- $\mathrm{E}(x_t) = \mu$（平均が時間によらず一定）
- $\mathrm{Cov}(x_t, x_{t+1}) = \gamma(|h|)$（異なる時点間の共分散が時間差のみに依存する）

定常過程 $\{x_t\}$ の分散は $\mathrm{V}(y_t) = \gamma(0)$ となる．つまり，時間に依存せずに一定である．自己共分散を分散で割った量

$$\rho(h) = \frac{\gamma(h)}{\gamma(0)} = \frac{\gamma(t)}{\mathrm{V}(y_t)} = \rho(-h)$$

を時差 h の自己相関といい，$\rho(h), h = 0, 1, 2, ...$ の全体をコレログラムという．$\rho(h)$ の絶対値は 1 以下であり，1 に近いほど相関が強いことを表す.

2.1.3　短期記憶性と長期記憶性

次のような時差の自己共分散の絶対総和

$$S = \sum_{h=-\infty}^{\infty} |\mathrm{Cov}\,(x_t, x_{t+h})| = \sum_{h=-\infty}^{\infty} |\gamma\,(h)|$$

について，S が有限の値に収束するときは時系列が短期記憶的，そうでないならば長期記憶的であるという．

2.2　自己回帰 (AR) モデル

短期記憶的なモデルの代表的なものとして，自己回帰 (AR) モデルが挙げられる．時系列 $\{x_t\}_{t\in T}$ が

$$x_t = \phi_1 x_{t-1} + \cdots + \phi_p x_{t-p} + \varepsilon_t,\ \{\varepsilon_t\} \sim \text{i.i.d.}\mathcal{N}\left(0, \sigma^2\right) \tag{2.1}$$

によって表されるとき，$\{x_t\}$ は次数 p の AR(p) モデルに従うという．AR(p) モデルの定常性について，次の定理が有名である：

> AR(p) モデルにおいて，特性方程式
>
> $$\phi\,(x) = 1 - \phi_1 x - \phi_2 x^2 - \cdots - \phi_p x^p = 0$$
>
> の根の絶対値がすべて 1 よりも大きいとき，定常過程である．

定常性 AR(p) モデルが与えられた場合に，その自己相関を求める方法を述べる．式 (2.1) の両辺に $y_{t-j}\,(j>0)$ をかけて期待値を取り

$$\gamma\,(j) = \phi_1\gamma\,(j-1) + \phi_2\gamma\,(j-2) + \cdots + \phi_p\gamma\,(j-p) \tag{2.2}$$

を得る．この両辺を $\gamma\,(0)$ で割ることで得られる p 階の差分方程式

$$\rho\,(j) = \phi_1\rho\,(j-1) + \phi_2\rho\,(j-2) + \cdots + \phi_p\rho\,(j-p),\ (j>0) \tag{2.3}$$

を考える．特性方程式の根によって場合分けを行う．ここでは，特性方程式の根 $\lambda_1, \lambda_2, \cdots, \lambda_p$ がすべて異なる場合のみを取り上げて考える．特性方程式の一般解は，$c_1, c_2, \cdots, c_p$ を定数として

$$\rho\,(j) = \sum_{i=1}^{p} \frac{c_i}{\lambda_i^j}$$

と与えられる．定数 $c_1, c_2, ..., c_p$ を求めるに先立ち，式 (2.3) において $j = 1, 2, ..., p-1$ とし，$\rho\,(1), \rho\,(2), ..., \rho\,(p-1)$ に関する次の連立方程式を解く：

$$\begin{aligned}
\rho(1) &= \phi_1 + \phi_2\rho(1) + \cdots + \phi_p\rho(p-1) \\
\rho(2) &= \phi_1\rho(1) + \phi_2 + \cdots + \phi_p\rho(p-2) \\
&\vdots \\
\rho(p-1) &= \phi_1\rho(p-2) + \phi_2\rho(p-3) + \cdots + \phi_p\rho(1).
\end{aligned}$$

この上で，$c_1, c_2, .., c_p$ を未知数に持つ，次の連立方程式を解けばよい：

$$\begin{aligned}
1 &= c_1 + c_2 + \cdots + c_p \\
\rho(1) &= \frac{c_1}{\lambda_1} + \frac{c_2}{\lambda_2} + \cdots + \frac{c_p}{\lambda_p} \\
&\vdots \\
\rho(p-1) &= \frac{c_1}{\lambda_1^{p-1}} + \frac{c_2}{\lambda_2^{p-1}} + \cdots + \frac{c_p}{\lambda_p^{p-1}}.
\end{aligned}$$

逆に，AR(p) モデルの自己相関が与えられた場合に，モデルの係数 $\phi_1, \phi_2, ..., \phi_p$ を求める場合を考える．このために，式 (2.3) において $j = 1, 2, ..., p$ の場合を考えて，次の p 本の方程式を得る：

$$\begin{aligned}
\rho(1) &= \phi_1 + \phi_2\rho(1) + \cdots + \phi_p\rho(p-1) \\
\rho(2) &= \phi_1\rho(1) + \phi_2 + \cdots + \phi_p\rho(p-2) \\
&\vdots \\
\rho(p) &= \phi_1\rho(p-1) + \phi_2\rho(p-2) + \cdots + \phi_p.
\end{aligned}$$

これらを，行列やベクトルで表現すると次を得る：

$$\begin{pmatrix}
1 & \rho(1) & \cdots & \rho(p-1) \\
\rho(1) & 1 & \cdots & \rho(p-2) \\
\vdots & \vdots & \ddots & \vdots \\
\rho(p-1) & \rho(p-2) & \cdots & 1
\end{pmatrix}
\begin{pmatrix}
\phi_1 \\ \vdots \\ \vdots \\ \phi_p
\end{pmatrix}
=
\begin{pmatrix}
\rho(1) \\ \vdots \\ \vdots \\ \rho(p)
\end{pmatrix}.$$

これをユール・ウォーカーの方程式という．

第3章　データの相関からのアプローチ

与えられた時系列データの構造を調べるためには，時系列データの相関を見ることが有効であると考えられる．この相関は，相互相関関数といった形で考えることができる．与えられたデータが大規模であるとき（例えば金融に関するデータ），リカレンスプロット [2] のような視覚的にも明確な手法を用いることができる．今回扱う購買データは大規模なデータではないものの，データに強い相関が見られる場合にはこの相関を用いたアプローチが考えられる．

3.1　単一のデータ系列における相関

ここでは，ある単一のデータ系列が与えられた場合にその相関を見ることを考える．例えば，ある食品が月曜日に特売されるのであれば，周期的に良く売れる日があるはずである．データの相関をみることでこういった情報を見出すことができる．

これは，次のように定式化する．対象の時系列を x_t，$(t = 1, 2, ...)$ とし，平均を μ_x とする．さらに位相遅れを τ とする．例えば，$\tau = 7$ とすれば，データにおける一週間の周期を見ることとなる．このとき，自己相関関数は

$$\gamma_{xx}(\tau) = \mathrm{E}\left[(x_t - \mu_x)(x_{t+\tau} - \mu_x)\right]$$

と定義される．

3.2　複数のデータ系列における相関

ここでは，複数のデータ系列にまたがる相関関係を見る．例えばある食品 A と別の食品 B が同時に良く売れる場合がある可能性，また，A が良く売れた次の日に B が良く売れる可能性があるのであれば，こういった情報を見出すことができる．

これは，次のように定式化する．対象の時系列を x_t, y_t $(t = 1, 2, ...)$ とし，それぞれの平均を μ_x, μ_y とする．さらに位相遅れを τ とする．例えば，$x_t =$ 野菜 1，$y_t =$ 野菜 2，$\tau = 7$ とすれば，ある日の野菜 1 の売り上げとその日

から一週間後の野菜 2 の売り上げの関係を見ることとなる．このとき，相互相関関数は

$$\gamma_{xy}(\tau) = \mathrm{E}\left[(x_t - \mu_x)(y_{t+\tau} - \mu_y)\right]$$

と定義される．

3.3　データに見られる相関

ここでは，データの相関を見るという比較的素朴な方法を考えた．しかし，今回のデータには実務として耐えうる強い相関を見出すことができなかった．従って，今回の分析ではよりモデルに基づくものをとることを考えた．

第4章　異常値，変化点検出ツール：Change Finder

購買データに関する異常値，変化点の検出を行う．このために [3, 4] で紹介されている，異常値と変化点を検出するためのツールである，Change Finder を紹介する．それに先立ち，ここでは変化点の検出，異常値の検出それぞれについて，解説を行う．異常値の検出は，ある時点において異常な数値が（時系列の振る舞いの変化ではなく）偶発的に起きた事象によって得られた場合に，これを変化点の検出としてではなく異常値として判別することである．食品の購買データを例に取ると，特売日に多く売れた場合，その特売日は異常値と捉えられるべきである．変化点の検出は，時系列の振る舞いの急激な変化を検出するものである．第二章において，時系列モデルの一つである AR モデルを解説した．たとえば，ある時系列が AR モデルでよく記述できるとする．しかし，ある時点を境に，時系列を記述する AR モデルのパラメータが急激に変化するとき，この点を変化点と呼ぶ．食品の購買データを例に取ると，ある日を境に急激に寒くなった場合，鍋物のような暖かい料理によく用いられる食材がその日を境に売れ行きがよくなる現象が観測される，と予想できる．この場合は，その急激に気温が下がった日を変化点と呼ぶことができる．本章の議論は，主に [3, 4] に従う．

4.1　定式化

ここでは変化点検出を定式化する．時系列データを $x_1x_2...$ とおく．これらのデータは，確率過程 p に従うものとする．記号を $x^{t-1} = x_1x_2...x_{t-1}$ とし，条件付確率を $p\left(x_t|x^{t-1}\right)$ と書く．確率過程 p が例えば確率過程 $p^{(1)}$, $p^{(2)}$ に，次のように分割できるとする：

$$
\begin{aligned}
p\left(x_t|x^{t-1}\right) &= p^{(1)}\left(x_t|x^{t-1}\right), \qquad t < a, \\
p\left(x_t|x^{t-1}\right) &= p^{(2)}\left(x_t|x^{t-1}\right), \qquad t \geq a.
\end{aligned}
$$

これらが “異なる” 場合 $t = a$ が変化点である．これらが異なるかどうかを判別するために，次のカルバック-ライブラー情報量を考える：

$$D\left(p^{(2)}||p^{(1)}\right) := \lim_{n\to\infty}\frac{1}{n}E_{p^{(2)}}\left[\log\frac{p^{(2)}\left(x^n\right)}{p^{(1)}\left(x^n\right)}\right],$$

この指標は非負の値をとり，$p^{(1)} = p^{(2)}$ の場合に $D = 0$ となる．カルバック-ライブラー情報量は，一般に計算が困難であるものの，容易に計算できる場合もある．カルバック-ライブラー情報量を計算し設けた基準と照らし合わせることで，変化点や異常値を検出することができる．例えば，以下のような場合である．

1. jumping mean

$p^{(1)}, p^{(2)}$ がそれぞれ正規分布 (μ_1, σ)，(μ_2, σ) に独立に従うとする．いま $D\left(p^{(1)}||p^{(2)}\right) = \left(\mu_1 - \mu_2\right)^2 / \left(2\sigma^2\right)$ である．

2. jumping variance

$p^{(1)}, p^{(2)}$ がそれぞれ正規分布 (μ, σ_1)，(μ, σ_2) に独立に従うとする．いま $D\left(p^{(1)}||p^{(2)}\right) = \frac{1}{2}\left(\frac{\sigma_2^2}{\sigma_1^2} - 1 - \log\frac{\sigma_2^2}{\sigma_1^2}\right)^2$ である．

4.2　統計的検定に基づく変化点検出

変化点検出において最も基本的な方法は，統計的検定に基づくものである．ここではその概要を述べる．これは，データ列に対して様々な時系列モデルを当てはめ，ある点の前後において別々の時系列モデルを当てはめた場合のほうが，一つのモデルを当てはめた場合と比較して誤差が有意に小さくなるか否かを検定によって考えるものである．有意に少なくなる場合，これらの差が最大となる点を変化点とみなす．

これをより厳密に定式化する．データ系列を $\mathbf{x}_1^n = \mathbf{x}_1, ..., \mathbf{x}_n$ とし，変化点の候補を t とする．時系列データ $\mathbf{x}_i, (i = 1, \cdots, n)$ の次元は d とする．この候補の前後の時系列データをそれぞれ $\mathbf{x}^t = \mathbf{x}_1, ..., \mathbf{x}_t$，$\mathbf{x}_t^n = \mathbf{x}_{t+1}, ..., \mathbf{x}_n$ とする．例として，各時刻 t に対して $\mathbf{x}_{t-k}^{t-1} = \mathbf{x}_{t-k}, ..., \mathbf{x}_{t-1}$ が与えられたときの $\mathbf{x}_t$ の条件付き確率密度関数が次のように与えられるモデルを考える．

$$p\left(\mathbf{x}_t|\mathbf{x}_{t-k}^{t-1}\right) = \frac{1}{(2\pi)^{d/2}\left|\Sigma\right|^{1/2}}\exp\left[-\frac{\left(\mathbf{x}_t - \mathbf{w}_t\right)^T\Sigma^{-1}\left(\mathbf{x}_t - \mathbf{w}_t\right)}{2}\right]$$

ここで，Σ は分散共分散行列である．さらに，例えばデータが AR モデルに従うとするとするとき，k を与えられた正整数，$\alpha_1, \alpha_2, ..., \alpha_k, \mu$ を実数値パラメータとして

$$\mathbf{w}_t = \sum_{i=1}^{k}\alpha_i\left(\mathbf{x}_{t-i} - \mu\right) + \mu$$

である．パラメータはデータから最尤推定するとし，その推定されたパラメータを用いて AR モデルによって求めた $\mathbf{w}_t$ の予測値を $\hat{\mathbf{w}}_t$ とするとき，当てはめ誤差を二乗誤差関数を用いて以下のようにする：

$$I\left(\mathbf{x}_1^n\right) = \sum_{t=1}^{n} \left\|\mathbf{x}_t - \hat{\mathbf{w}}_t\right\|^2.$$

さらに，時刻 t の前の当てはめ誤差 $I\left(\mathbf{x}_1^t\right)$ を $Error1$ とし，時刻 t より後の当てはめ誤差 $I\left(\mathbf{x}_{t+1}^n\right)$ を $Error2$ とするとき，これらの和が時刻 $t = t^*$ において最小値をとるとする．しきい値 $\delta > 0$ を定め

$$\frac{I\left(\mathbf{x}_1^n\right) - \left(I\left(\mathbf{x}_1^{t^*}\right) + I\left(\mathbf{x}_{t^*+1}^n\right)\right)}{n} > \delta$$

となるとき，t^* を変化点とする．といったものである．

この方法の一番の問題点は，候補点すべてに対して統計的検定を行うために計算量が大きく，動的な計算が難しい点にある．今回の研究対象である購買データは，データの量が大きくはないためこの点が問題になることはない．しかし，今回はより新しいアプローチを行いたいため，Change Finder と呼ばれる計算量の観点から非常に優れるアルゴリズムを用いる．

4.3　Change Finder の概要

Change Finder は，時系列データから外れ値や変化点を検知するツールである．Change Finder の特徴は二段階学習を行う点にある．ここでは大枠のみを述べ，次節で具体的に確率モデルやスコアを与えた場合を考える．

Step 1：**第一段学習**

与えられた時系列データを $\mathbf{x}_1, \mathbf{x}_2, ...$ とする．これらのデータを用いて，時系列データの確率モデルに学習させる．ここで得られた確率密度関数の列を $\{p_t\left(\mathbf{x}\right) : t = 1, 2, ...\}$ とする．ここで $p_{t-1}\left(\mathbf{x}\right)$ は $\mathbf{x}^{t-1} = \mathbf{x}_1, ..., \mathbf{x}_{t-1}$ から学習された確率密度関数である．この確率密度関数列から，各時点 t のデータ $\mathbf{x}_t$ の外れ値スコアを定める．これらのスコアを $Score\left(\mathbf{x}_t\right)$，$t = 1, 2, ...$ とする．

Step 2：**平滑化**

幅 $W > 0$ の窓を設定し，窓内のデータに関して Step1 で求めた外れ値スコアの平均を計算する（平滑化）．さらに，この窓を移動することで平均移動スコアによる時系列 $\{\mathbf{y}_t : t = 1, 2, ...\}$ を新たに計算する．まとめると，本 Step ではスコア系列 $\{Score\left(\mathbf{x}_i\right) : i = t - W + 1, ..., t\}$ に対して W-平均スコア系列 y_t をスコア移動平均として次のように定義する：

$$y_t = \frac{1}{W} \sum_{t=t-W+1}^{t} Score\left(\mathbf{x}_i\right).$$

Step 3：**第二段階学習**

ここでは，Step1 と似た作業を再び行う．Step2 で得た新たな時系列データを y_t，$t = 1, 2, ...$ とし，これらのデータを用いて再び時系列データの確率モデルに学習させる．ここで得られた確率密度関数の列を $\{q_t\left(y_t\right) : t = 1, 2, ...\}$ とする．この確率密度関数列から，各時点 t のデータ y_t の外れ値スコアを定める．これらのスコアを $Score\left(y_t\right)$，$t = 1, 2, ...$ とする．ここで幅 $W' > 0$ の窓を設定し，時刻 t における W'-平均スコアを

$$Score\left(t\right) = \frac{1}{W'} \sum_{i=t-TW+1}^{t} Score\left(y_i\right)$$

とする．

冒頭で，Change Finder の特徴が二段階学習にあることを述べた．これは，第一段階学習では時系列の外れ値検知のみを行うが，Step2 の平滑化によってノイズによる外れ値を取り除き，Step3 の第二段階目の学習で本質的な変動を捉え，変化点検知を行うことにある．ところで，Step2 で用いた T の値によって Change Finder の性能に変化が出る．W の値が小さい時は，外れ値・変化点検知共に敏感に行えるがそれらの見分けが難しい．W の値が大きいときは，変化点検知に遅れが出てしまうものの外れ値の影響が取り除かれ変化点のみが検知できるようになる．従って，必要に応じて W の値を調整する必要がある．

4.4　Change Finder の例

ここでは，Change Finder において時系列モデルとして AR モデルを用いた場合に具体的にどのように計算を行えば良いかを述べる．この場合は，特に計算量を軽減する方法が提案されている．また，具体的にどういったスコア関数を用いれば良いかも述べる．

Step 1：**第一段階学習**

時系列データ $\mathbf{x}_t$，$t = 1, 2, ...$ これを，後に詳しく紹介する SDAR アルゴリズムを用いて学習し，得た確率密度関数の列を $\{p_t\left(\mathbf{x}\right) : t = 1, 2, ...\}$ とする．SDAR アルゴリズムは，一般的な変化点検知の手法である統計的検定によるものに比べて，計算量が非常に軽減されるものであり，次節で詳細に述べる．

各時点 t のデータ $\mathbf{x}_t$ の外れ値スコアを評価するスコア関数についてであるが，例えば対数損失

$$Score\left(\mathbf{x}_t\right) = -\log p_{t-1}\left(\mathbf{x}_t\right)$$

または，ヘリンジャー距離

$$Score\left(\mathbf{x}_t\right) = d\left(p_{t-1}, p_t\right) = \sum\left(\sqrt{p\left(\mathbf{x}_{t-1}\right)} - \sqrt{p\left(\mathbf{x}_t\right)}\right)^2$$

を用いて計算する．

d 次元の AR モデルは次のように記述できる：

$$\mathbf{z}_t = \sum_{i=1}^{k} A_i \mathbf{z}_{t-i} + \varepsilon$$

いま A_i は係数行列，ε は平均 0，分散共分散行列が Σ であるノイズである．ここで，観測された時系列 $\{\mathbf{x}_t : t = 1, 2, ...\}$ が

$$\mathbf{x}_t = \mathbf{z}_t + \mu$$

を満たすと仮定する．さらに $\mathbf{x}_{t-k}^{t-1} = \mathbf{x}_{t-k}...\mathbf{x}_{t-1}$ とて，$\mathbf{x}_{t-k}^{t-1}$ の条件の下での $\mathbf{x}_t$ の条件付確率密度は

$$p\left(\mathbf{x}_t|\mathbf{x}_{t-k}^{t-1} : \theta\right) = \frac{1}{(2\pi)^{k/2}\left|\Sigma\right|^{1/2}} \exp\left(-\frac{1}{2}\left(\mathbf{x}_t - \mathbf{w}\right)^T \Sigma^{-1}\left(\mathbf{x}_t - \mathbf{w}\right)\right),$$

$$\mathbf{w} = \sum_{i=1}^{k} A_i\left(\mathbf{x}_{t-i} - \mu\right) + \mu$$

である．パラメータの設定を考える．$\theta = (A_1, ..., A_k, \mu, \Sigma)$ と置き，x_t の元で推定した θ を θ_t として，$p_t = p\left(\cdot|\cdot, \theta_t\right)$ と置く．θ の推定は次を最大化することで行う (SDAR アルゴリズム)：

$$\sum_{i=1}^{k}(1-r)^{t-i} \log p\left(\mathbf{x}_i|\mathbf{x}^{i-1}, \theta\right).$$

Step 2：**第二段階学習**

Step1 で得た新しい時系列データ $\{y_t : t = 1, 2, ...\}$ に対し，AR モデルを用いてモデル化し，再び SDAR アルゴリズムによって学習を行う．ここで得られた確率モデルの時系列を $\{q_t : t = 1, 2, ...\}$ と置く．再び $W' > 0$ を設定し，時刻 t における W'-平均スコアを次の対数損失

$$Score\,(t) = \frac{1}{W'} \sum_{i=t-W'+1}^{t} \left(-\log q_{i-1}\,(y_i)\right)$$

または Step1 と同様にヘリンジャー距離

$$Score\,(t) = \frac{1}{W'} \sum_{i=t-W'+1}^{t} d\,(q_{i-1}, q_i)$$

を用いて定める．$Score\,(t)$ の値が大きいほど時刻 t が変化点である可能性が高いとみなすことができる．

4.5　SDAR アルゴリズム

ここでは，AR モデルのオンライン忘却学習アルゴリズムである，SDAR アルゴリズム (Sequencially Discounting AR model learning) について解説をする．総数 n の時系列データ $\mathbf{x}_1, ..., \mathbf{x}_n$ を考える．$\mathbf{x}_t = \mathbf{z}_t + \mu$ と変換をすれば，$\mathbf{z}_1, ..., \mathbf{z}_n$ に関する尤度は次のようになる：

$$\prod_{t=1}^{k} p\,(\mathbf{z}_t|\theta) \cdot \prod_{t=k+1}^{n} p\left(\mathbf{z}_t|\mathbf{z}_{t-k}^{t-1} : \theta\right).$$

対数尤度は

$$\sum_{t=1}^{k} \log p\,(\mathbf{z}_t|\theta) + \sum_{t=k+1}^{n} \log p\left(\mathbf{z}_t|\mathbf{z}_{t-k}^{t-1} : \theta\right).$$

である．$n \gg k$ の条件の下，上の式は第二項が支配的である．そこで，AR モデルの確率密度関数を考え，以下のように近似する：

$$-(n-k)\log\left((2\pi)^{1/2}\,|\Sigma|^{1/2}\right) - \frac{1}{2}\sum_{t=k+1}^{n}\left(\mathbf{z}_t - \sum_{i=1}^{k} w_i \mathbf{z}_{t-i}\right)^T \Sigma^{-1}\left(\mathbf{z}_t - \sum_{i=1}^{k} w_i \mathbf{z}_{t-i}\right).$$

この式を w について偏微分すれば，対数尤度を最大にする $w_i\,(i = 1, ..., k)$ は次を満たす（ユール - ウォーカーの方程式）：

$$\sum_{i=1}^{k} w_i C_{j-i} = C_j \;\; (j = 1, ..., k).$$

いま，C_j は次で与えられる自己共分散である：

$$C_j = \frac{1}{n-k}\sum_{t=k+1}^{n} \mathbf{z}_t \mathbf{z}_{t-j}^T$$
$$= \frac{1}{n-k}\sum_{t=k+1}^{n} \left(\mathbf{x}_t - \mu\right)\left(\mathbf{x}_{t-j} - \mu\right)^T .$$

また，すべての正の整数 s に対して次が満たされる：

$$C_s = C_{-s}^T.$$

μ と Σ の最尤推定量は以下のように求める：

$$\hat{\mu} = \frac{1}{n-k}\sum_{t=k+1}^{n} \mathbf{x}_t,$$

C_j の μ に $\hat{\mu}$ を代入したときのユール - ウォーカー方程式の解を $\hat{w}_1, ..., \hat{w}_k$ とすると，

$$\hat{\Sigma} = \frac{1}{n-k}\sum_{t=k+1}^{n} \left(\mathbf{z}_t - \sum_{i=1}^{k} \hat{w}_i \mathbf{z}_{t-i}\right)\left(\mathbf{z}_t - \sum_{i=1}^{k} \hat{w}_i \mathbf{z}_{t-i}\right)^T$$
$$= \frac{1}{n-k}\sum_{t=k+1}^{n} \left(\mathbf{x}_t - \hat{\mu} - \sum_{i=1}^{k} \hat{w}_i \left(\mathbf{x}_{t-i} - \mu\right)\right)\left(\mathbf{x}_t - \hat{\mu} - \sum_{i=1}^{k} \hat{w}_i \left(\mathbf{x}_{t-i} - \mu\right)\right)^T$$

となる．

SDAR アルゴリズムは上の計算を逐次的に実行するが，その際にパラメータや統計量を，現在の値と新しい値の $(1-r) : r$ の比の重みつき平均の形で更新する．$0 < r < 1$ は忘却パラメータであり，r が小さいほど SDAR アルゴリズムは過去のデータの影響を大きく受けることとなる．AR モデルは時系列の定常性を仮定するモデルであるが，変化点検出にあたっては非定常性を仮定しなくてはならない．SDAR アルゴリズムは，忘却効果を取り入れることで AR モデルを主体的に用いつつも非定常なモデルの学習を実現している．ここで，SDAR アルゴリズム視覚的にまとめておく．

SDAR アルゴリズム

忘却パラメータ:$0 < r < 1$

$\hat{\mu}, C_j, \hat{w}_j\,(j = 1, 2, ...)\,, \hat{\Sigma}$ を設定

$t\,(= 1, 2, ...)$ に対して $\mathbf{x}_t$ を読み込み，次を計算する：

$$\hat{\mu} := (1 - r)\,\hat{\mu} + r\mathbf{x}_t,$$
$$C_j := (1 - r)\,C_j + r\,(\mathbf{x}_t - \hat{\mu})\,(\mathbf{x}_{t-j} - \hat{\mu})^T\,.$$

さらに，次のユール - ウォーカー方程式を解く

$$\sum_{i=1}^{k} w_i C_{i-i} = C_j \;\; (j = 1, ..., k)\,.$$

上の式の解を $\hat{w}_1, ..., \hat{w}_k$ とおき，以下を計算する.

$$\hat{\mathbf{x}}_t := \sum_{i=1}^{k} \hat{w}_i\,(\mathbf{x}_{t-i} - \hat{\mu}) + \hat{\mu},$$
$$\hat{\Sigma} := (1 - r)\,\hat{\Sigma} + r\,(\mathbf{x}_t - \hat{\mathbf{x}}_t)\,(\mathbf{x}_t - \hat{\mathbf{x}}_t)^T$$

第5章　分析結果，考察

ここでは，Change Finder を用いた分析の結果を示す．ここで扱うデータは，スーパーの販売のデータに関するものである．野菜五種類，肉六種類のデータがあり，11 種類のデータは，同一チェーン店の，異なる店舗による販売データである．それぞれ，Change Finder による異常値検知または変化点検知を行った．

5.1　Change Finder の第一段階学習による異常値検知

以下では，実データを用いた分析を行う．SDAR モデルの自由度と忘却パラメータに関しては，AR モデルの次数を 2，忘却パラメータを 0.02 とした．

また，各々の分析において $Score\left[x\left(t\right)\right]$ の平均を $\mathrm{E}\left[Score\left[x\left(t\right)\right]\right]$，標準偏差を $\sigma\left[Score\left[x\left(t\right)\right]\right]$ として

$$\mathrm{E}\left[Score\left[x\left(t\right)\right]\right]+4\sigma\left[Score\left[x\left(t\right)\right]\right]$$

よりも大きな $Score$ を示した時点でのデータを異常値とした．

また，比較のために，$Score\left[x\left(t\right)\right]$ ではなく $x\left(t\right)$ をそのまま用いて

$$\mathrm{E}\left[x\left(t\right)\right]-2\sigma\left[x\left(t\right)\right]\sim\mathrm{E}\left[x\left(t\right)\right]+2\sigma\left[x\left(t\right)\right]$$

の範囲に入らない時点のデータを異常値とした場合との比較も行った．各グラフの先頭数日間は，SDAR モデルの学習に用いられるため，参考とならない．

5.1.1　野菜 1 の分析

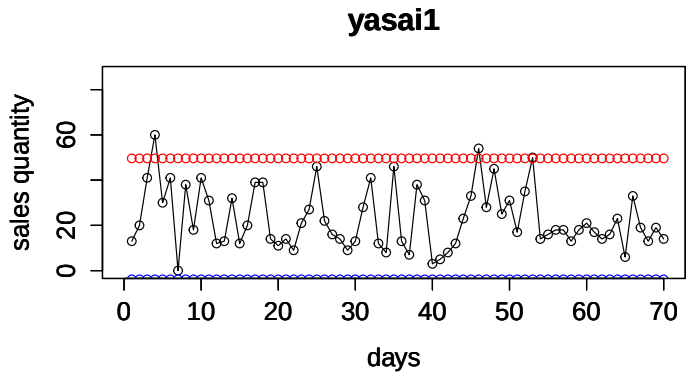

図 5.1.1: 野菜 1 の販売量．横軸が経過日数，縦軸が販売実績．赤い線と青い線は異常値のしきい値を表す．

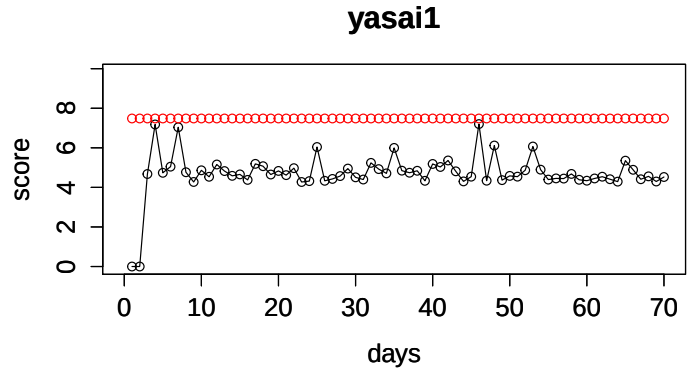

図 5.1.2: 野菜 1 の分析．横軸が経過日数，縦軸がスコア．赤い線は異常値のしきい値を表す．

野菜 1 の販売データは全体を通じて変動が大きく，明らかに異常値と考えられるデータはない．しかし，第 46 日目の売り上げは，両手法において共に異常値と認識された．

5.1.2　野菜 2 の分析

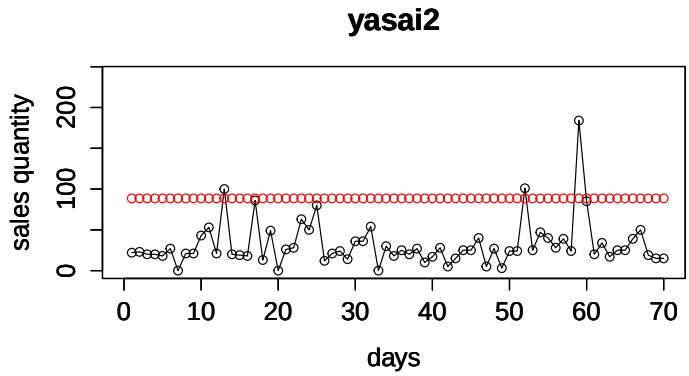

図 5.1.3: 野菜 2 の販売量．横軸が経過日数，縦軸が販売実績．赤い線は異常値のしきい値を表す．青い線のしきいは負であり，図には表れない．

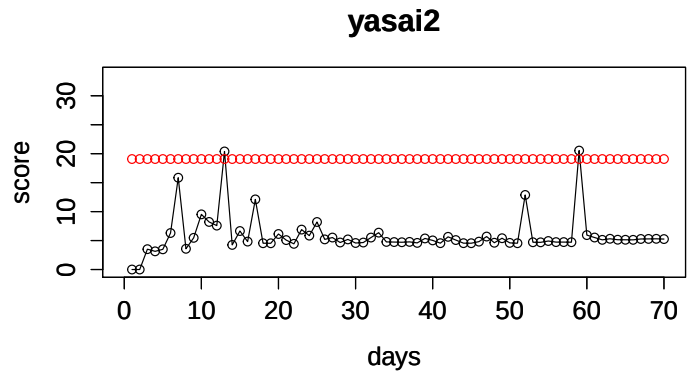

図 5.1.4: 野菜 2 の分析．横軸が経過日数，縦軸がスコア．赤い線は異常値のしきい値を表す．

野菜 2 の販売データでは，第 59 日目の売り上げが突出している．販売データから直接判別する方法では，第 13, 17, 25, 52, 29 日目の売り上げが異常値として判断された．第一段階学習スコアから判別する手法では，第 13, 59 日目の売り上げが異常値として認識された．

5.1.3　野菜 3 の分析

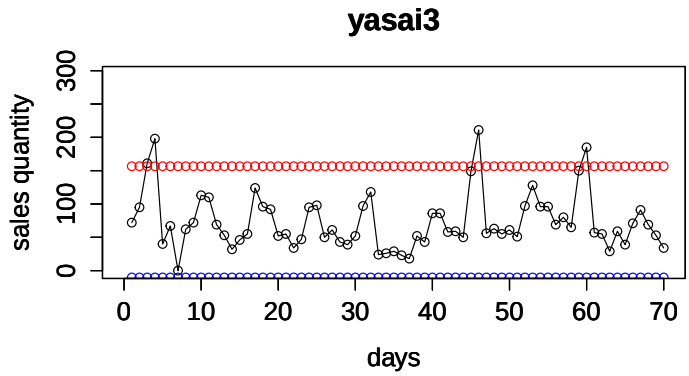

図 5.1.5: 野菜 3 の販売量．横軸が経過日数，縦軸が販売実績．赤い線と青い線は異常値のしきい値を表す．

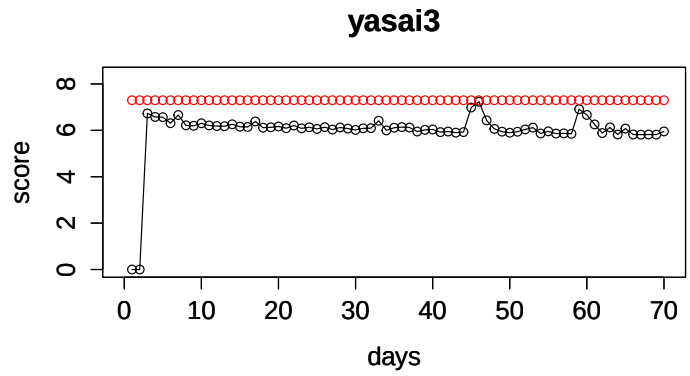

図 5.1.6: 野菜 3 の分析．横軸が経過日数，縦軸がスコア．赤い線は異常値のしきい値を表す．

野菜 3 のデータでは，販売データから直接判別する方法では，第 45, 46, 59, 60 日目が異常値として判別された．第一段階学習スコアを用いる方法では，顕著な変化ではないものの，第 45, 46, 59, 60 日目にスコアが大きくなる様子が観察された．

5.1.4　野菜 4 の分析

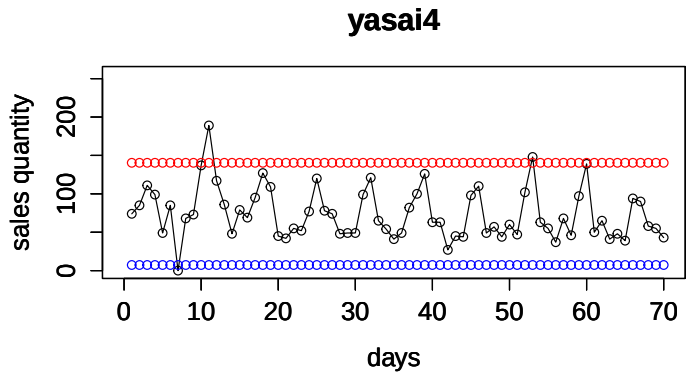

図 5.1.7: 野菜 4 の販売量．横軸が経過日数，縦軸が販売実績．赤い線と青い線は異常値のしきい値を表す．

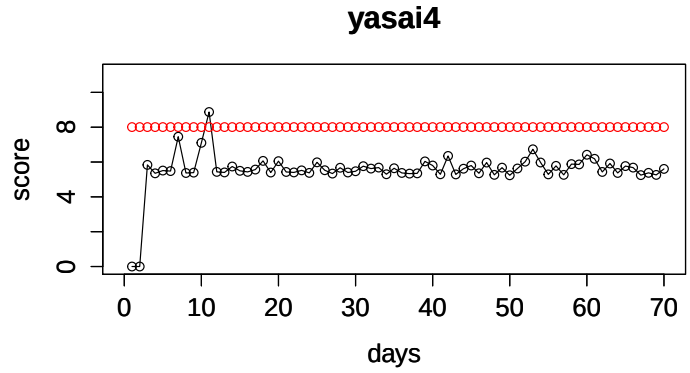

図 5.1.8: 野菜 4 の分析．横軸が経過日数，縦軸がスコア．赤い線は異常値のしきい値を表す．

野菜 4 の分析においては，第 11 日目の販売データが，両手法において共に異常値と判別された．

5.1.5　野菜 5 の分析

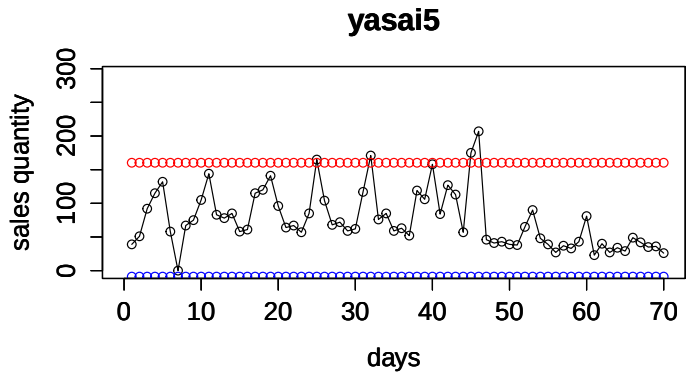

図 5.1.9: 野菜 5 の販売量．横軸が経過日数，縦軸が販売実績．赤い線と青い線は異常値のしきい値を表す．

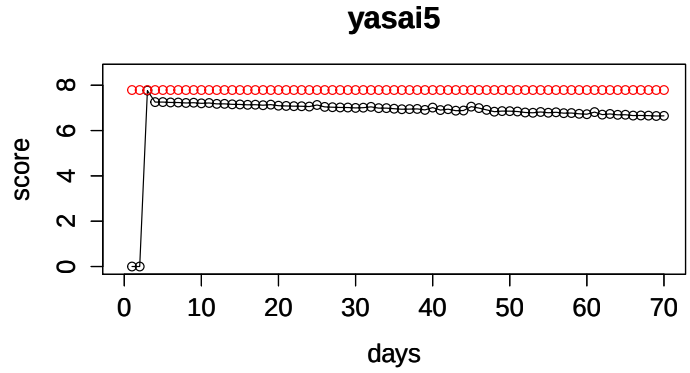

図 5.1.10: 野菜 5 の分析．横軸が経過日数，縦軸がスコア．赤い線は異常値のしきい値を表す．

野菜 5 の分析では，販売データから直接判別する手法では異常値と判断されるデータが多く存在したものの，第一段階学習スコアを用いて判別する手法では異常値とされるデータが存在しなかった．

5.1.6　肉 1 の分析

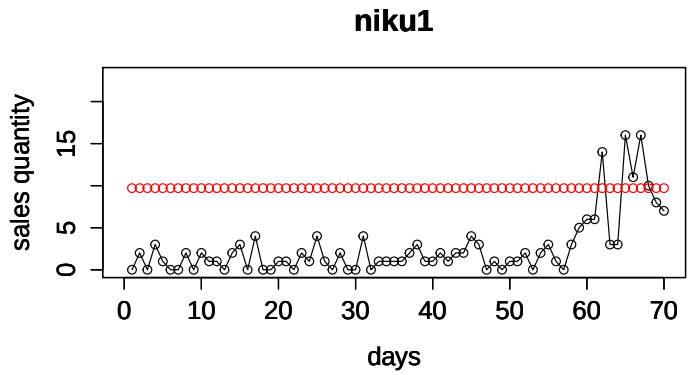

図 5.1.11: 肉 1 の販売量．横軸が経過日数，縦軸が販売実績．赤い線は異常値のしきい値を表す．青い線のしきいは負であり，図には表れない．この分析では，$T = 2$ の時はユール・ウォーカー方程式が不安定なため，AR モデルの次数を 3 としている．

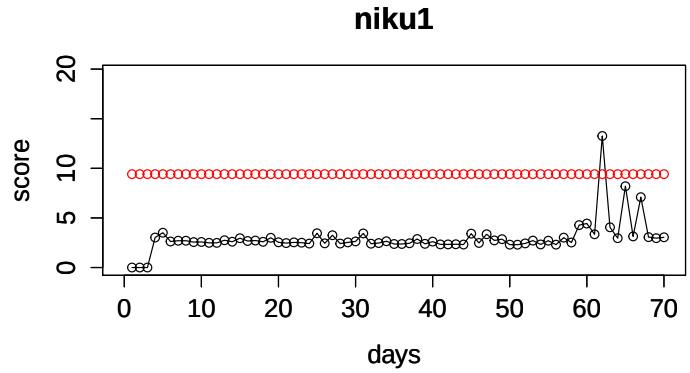

図 5.1.12: 肉 1 の分析．横軸が経過日数，縦軸がスコア．赤い線は異常値のしきい値を表す．

肉 1 の分析においては，販売データから直接判別する方法，第一段階学習スコアから判別する方法ともに第 62 日目の売り上げが異常値と判断された．売り上げデータに見られる 63 日以降の大きな値は，第一段階学習スコアとしては異常値として判断されなかった．

5.1.7　肉 2 の分析

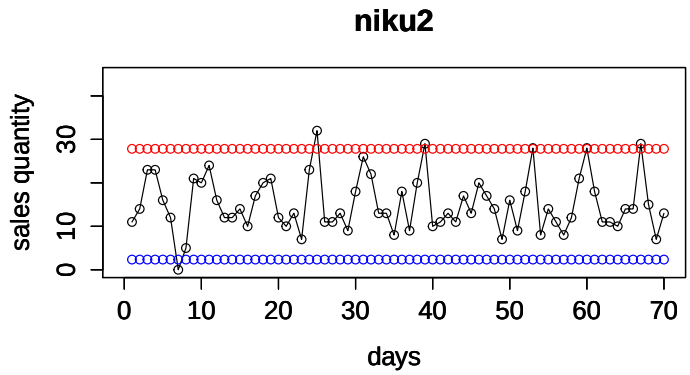

図 5.1.13: 肉 2 の販売量．横軸が経過日数，縦軸が販売実績．赤い線と青い線は異常値のしきい値を表す．

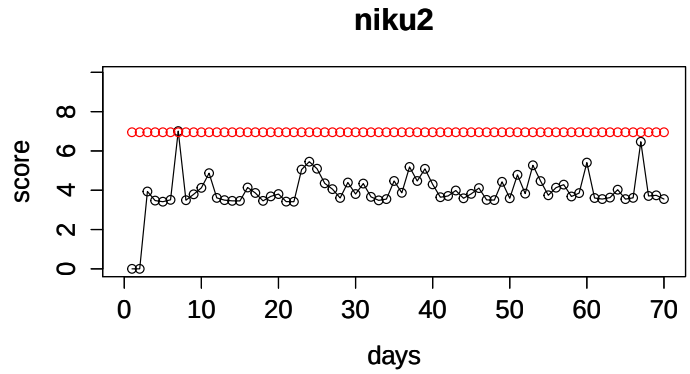

図 5.1.14: 肉 2 の分析．横軸が経過日数，縦軸がスコア．赤い線は異常値のしきい値を表す．

肉 2 の販売データは全体を通じて変動が大きい．販売データから直接判別する方法では多くの日の売り上げが異常値と判断された．　第一段階学習スコアを用いる方法では，その中で第 7, 67 日目のデータが異常値と判別された．

5.1.8　肉 3 の分析

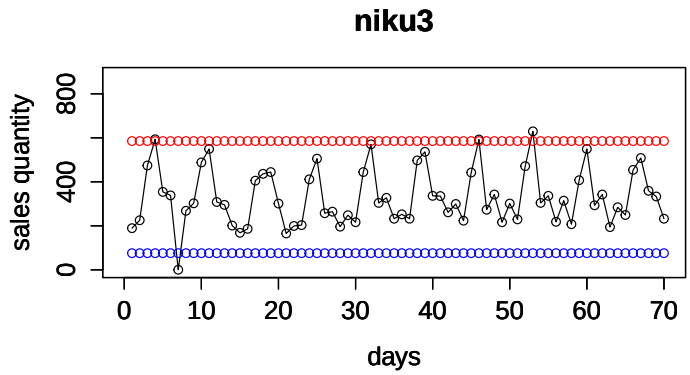

図 5.1.15: 肉 3 の販売量．横軸が経過日数，縦軸が販売実績．赤い線と青い線は異常値のしきい値を表す．

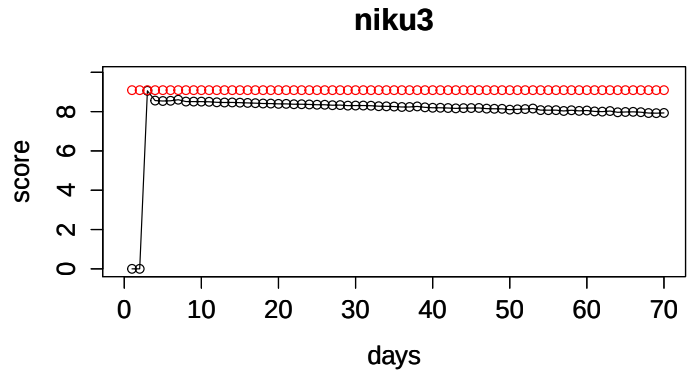

図 5.1.16: 肉 3 の分析．横軸が経過日数，縦軸がスコア．赤い線は異常値のしきい値を表す．この分析のみ AR モデルの次数を 3 とした．

肉 3 の分析においては，販売データから直接判断する方法では異常値と判別される日が多く存在したが，第一段階学習スコアを用いる方法だと異常値と判断される日は存在しなかった．

5.1.9　肉 4 の分析

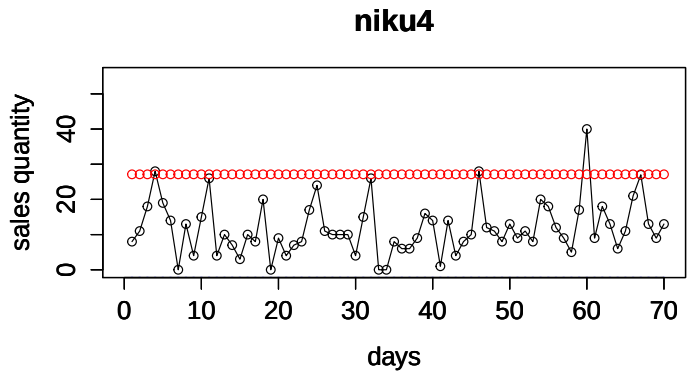

図 5.1.17: 肉 4 の販売量．横軸が経過日数，縦軸が販売実績．赤い線は異常値のしきい値を表す．青い線のしきいは負であり，図には表れない．

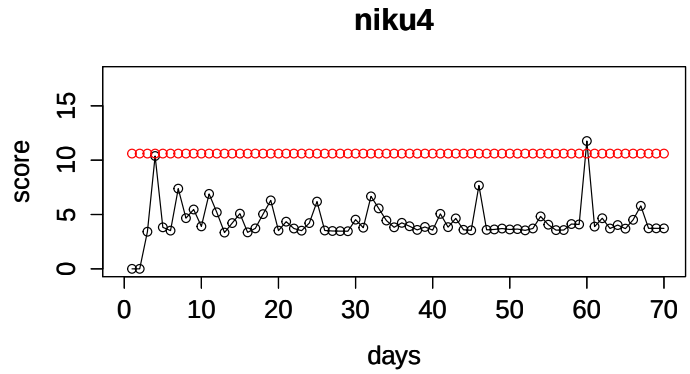

図 5.1.18: 肉 4 の分析．横軸が経過日数，縦軸がスコア．赤い線は異常値のしきい値を表す．

肉 4 の分析においては，販売データから直接判別する手法では異常値と判断される日が多いが，第一段階学習スコアを用いる方法では，60 日目が異常値として判断される．元データの 60 日目以外に見られる大きな値は，スコアの方ではあまり大きな影響がみられなかった．

5.1.10　肉 5 の分析

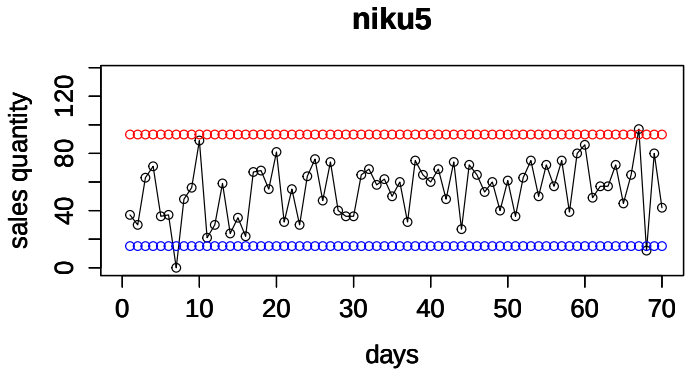

図 5.1.19: 肉 5 の販売量．横軸が経過日数，縦軸が販売実績．赤い線と青い線は異常値のしきい値を表す．

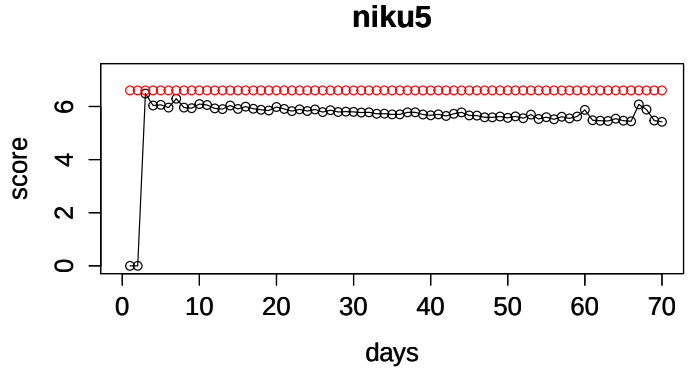

図 5.1.20: 肉 5 の分析．横軸が経過日数，縦軸がスコア．赤い線は異常値のしきい値を表す．

肉 5 の分析においては，販売データから直接判別する手法では売り上げの大小それぞれにおいて異常値と判断される日が存在したが，第一段階学習スコアを用いる方法では，異常値と判断された日は存在しなかった．

5.1.11　肉 6 の分析

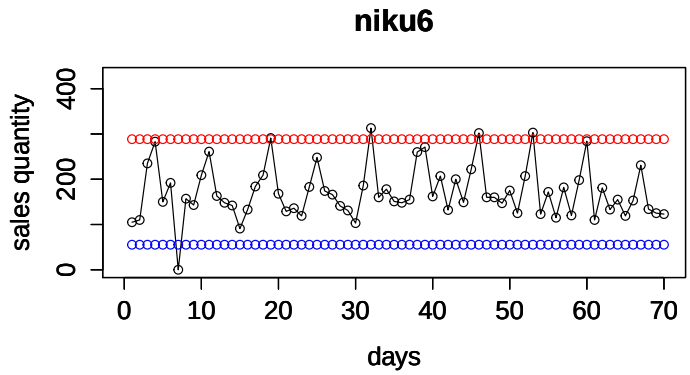

図 5.1.21: 肉 6 の販売量．横軸が経過日数，縦軸が販売実績．赤い線と青い線は異常値のしきい値を表す．

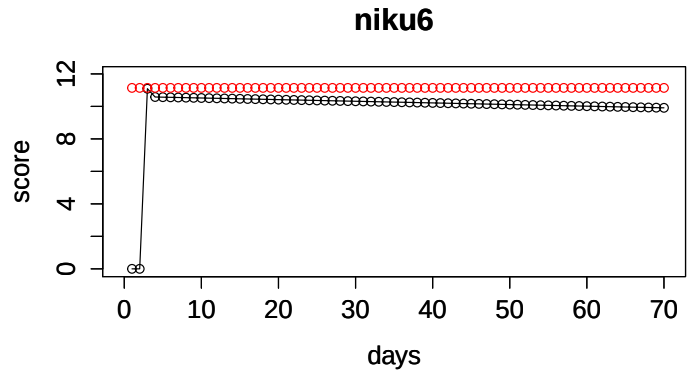

図 5.1.22: 肉 6 の分析．横軸が経過日数，縦軸がスコア．赤い線は異常値のしきい値を表す．

肉 6 の分析においては，販売データから判別する手法では多くの日が異常値と判断されたが，第一段階学習スコアを用いる方法では異常値と判断された日は存在しなかった．

5.2　Change Finder による異常値検知の考察

忘却パラメータが大きいとき，つまり，過去のデータの引きずりが小さい場合は，元データの振動がスコアに直接反映されるため，よい評価ができないと考えられる．今回の購買データの分析においては，忘却パラメータを小さく設定することが好ましいといえる．また，AR モデルの次数は，直感的

には一週間ごとのトレンドを反映する 7 がよいように思えるが，実際には 2 または 3 に設定するのがよいだろう．

例えば野菜 1 のように，大きく全体が振動しているようなデータに対しては，アルゴリズムがよく動いていないと考えられる．

5.3　Change Finder の第二段階学習による変化点検知

異常検知を行った 11 種類のデータの中で，特に変化点を持つと考えられる野菜 1，野菜 2，肉 1 のデータについて，Change Finder を適用した結果を掲載する．各グラフの先頭数日間は，SDAR モデルの学習に用いられるため，参考とならない．

5.3.1　野菜 1 の分析

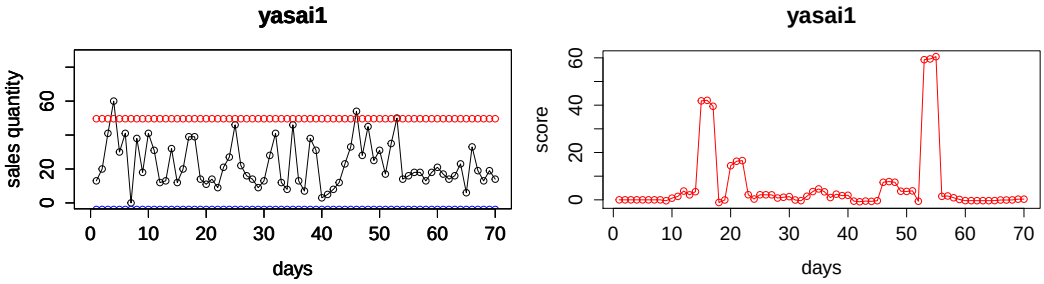

図 5.3.1: 第一段階学習において AR モデルの次数を 3，忘却パラメータを 0.02，窓幅を 4 とし，第二段階学習において AR モデルの次数を 2，忘却パラメータを 0.02 窓幅を 4 とした場合の分析．横軸が経過日数．縦軸がスコア．

野菜 1 の販売データでは，50 日目以降に，平均と分散が小さくなる，変化点と考えられる部分がある．パラメータをうまく調整することで，この変化点を検出することができる．

5.3.2　野菜 5 の分析

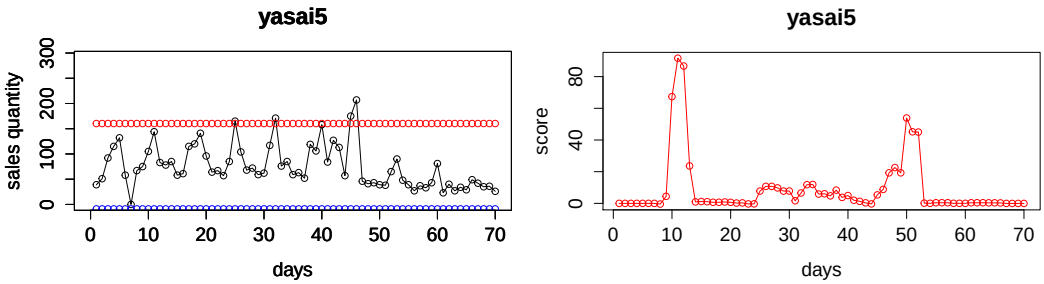

図 5.3.2: 第一段階学習において AR モデルの次数を 2，忘却パラメータを 0.02，窓幅を 3 とし，第二段階学習において AR モデルの次数を 2，忘却パラメータを 0.002 窓幅を 3 とした場合の分析．横軸が経過日数．縦軸がスコア．

野菜 5 の販売データでは，50 日目前後に平均と分散が小さくなる，変化点と考えられる部分がある．パラメータをうまく調整することで，この変化点を検出することができる．

5.3.3　肉 1 の分析

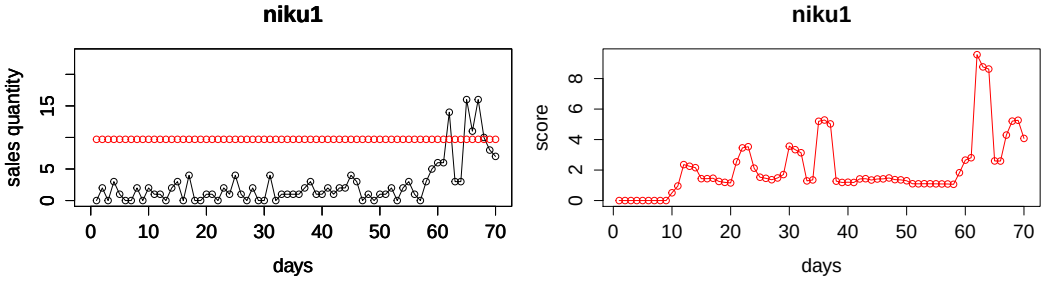

図 5.3.3: 第一段階学習において AR モデルの次数を 3，忘却パラメータを 0.02，窓幅を 4 とし，第二段階学習において AR モデルの次数を 3，忘却パラメータを 0.002 窓幅を 4 とした場合の分析．横軸が経過日数．縦軸がスコア．

肉 1 の販売データでは，60 日目以降に，平均値と分散が大きくなる変化点と考えられる部分がある．パラメータをうまく調節することで，この変化点を検出することができる．

5.4　Change Finder による変化点検知の考察

今回の分析では，販売データにおいて変動の様子が明らかに変化した時点に関して，その時点を変化点として捕らえることができた．しかしながら，現時点ではパラメータの設定は手作業で行う必要がある．この作業を自動化することができれば，より便利なものとなるだろう．

第6章　まとめとこれからの課題

今回の分析によって，本来は大規模なデータに対して用いられるアルゴリズムである Change Finder が，今回のような購買データに対しても効果的であることがわかった．特に，異常値検知に関してはある程度の知見を与えることができると思われる．一方で，変化点検知に関しては今回の分析ではよくわからない点があったので，変化点を含みそうな購買データに対しても分析を行ってみたいと思う．(例えば，消費税の増税は変化点なのか異常値なのか，など)

また，今回のデータは，異なるデータ列間には相関が見られないものであった．もし，データ列間に相関がある程度見られるならば，Change Finder を多次元に拡張することで，よりよい異常値検知，変化点検知を行うことができると考えられる．

関連図書

[1] 田中勝人，現代時系列分析，岩波書店，2006.

[2] J. P. Peckmann, S. Oliffson Kamphorst, and D. Ruelle, ”Recurrence Plots of Dynamical Systems”, *Europhysics Letters*, Vol. 4, No. 9, pp. 973-977, 1987.

[3] 山西健司，データマイニングによる異常検知，共立出版，2009.

[4] J. Takeuchi and K. Yamanishi, “A Unifying Framework for Detecting Outliers and Change Points from Time Series”, *IEEE Transaction on Knowledge and Data Engineering*, Vol. 18, No. 4, pp482-492, 2006.

[5] C. M. ビショップ（著），元田浩，栗太多喜雄，樋口知之，松本裕治，村田昇（監修），パターン認識と機械学習　上，丸善出版，2008.

付 録A　数学的な補足

A.1　カルバック-ライブラー情報量について

ここでは，学習理論においてよく用いられるカルバック-ライブラー情報量の解説を行う [5].

A.1.1　情報量の概念

離散的確率変数 x を考える．情報量 $h(x)$ は，この変数 x に関する値を得た場合に，値を得たことがどのくらい価値のあることかを測るものであってほしい．そこで，x の従う確率密度関数を $p(x)$ として，次が要請される．1. $h(x)$ は $p(x)$ の単調な関数である．2. 事象 x, y が独立（無関係）であるとき，$h(x,y) = h(x) + h(y)$ が成立する．これらと，$p(x,y) = p(x)\,p(y)$ の関係より，$h(x)$ は $p(x)$ の対数となるため，

$$h(x) = -\log p(x)$$

と定義する．低い確率で起きる事象 x に対して，大きな情報量 $h(x)$ が得られる．対数の底の選び方には自由度がある．情報理論では一般的に 2 が底として用いられ，単位はビットである．

A.1.2　エントロピーについて

上で定義した情報量を用いて，エントロピーを定義する．エントロピーは情報を分布に関して平均値をとったものであり，次のように定義する：

$$\mathrm{H}[x] = -\sum_{x} p(x) \log p(x)\,.$$

エントロピーに対する，物理的な解釈を与える．N 個の同じ物体がたくさんの箱に入れられている状況を考える．i 番目の箱には n_i 個の物体が入るものとして，物体を箱に入れる場合の数を考える．このとき，n_i 番目の箱には $n_i!$ 通りの順番付けの方法があるため，N 個の物体を箱に入れる，入れ方の総数は

$$W = \frac{N!}{\prod_i n_i!}$$

だけあり，これを多重度という．このときのエントロピーは，この多重度の対数を適当に定数倍したものと定義し，

$$\mathrm{H} = \frac{1}{N} \ln W = \frac{1}{N} \ln N! - \frac{1}{N} \sum_i \ln n_i!$$

となる．ここで，n_i/N を一定として $N \to \infty$ の極限を考える．スターリングの公式

$$\ln N! \simeq N \ln N - N$$

を用いると，$\sum_i n_i = N$ より

$$\mathrm{H} = -\lim_{N\to\infty} \sum_i \left(\frac{n_i}{N}\right) \ln \left(\frac{n_i}{N}\right) = -\sum_i p_i \ln p_i$$

ここで，$p_i = \lim_{N\to\infty} (n_i/N)$ は物体が i 番目の箱に割り当てられる確率をあらわす．物理的な解釈を述べると，箱中の物体の特定の状態をミクロ状態と呼び，n_i/N で表される物体の占有率の分布はマクロ状態と呼ぶ．また，多重度 W はマクロ状態の重みと呼ぶ．

以上は，離散的な変数に関するエントロピーを考えたが，連続的な変数に対するエントロピーを考えることもできる．この場合は

$$\mathrm{H}[\mathrm{x}] = -\int p(\mathrm{x}) \ln p(\mathrm{x})\, \mathrm{dx}$$

となる．

A.2　カルバック-ライブラー情報量について

以上の議論から，カルバック-ライブラー情報量を考察する．ある未知の分布 $p(\mathrm{x})$ を近似的に $q(\mathrm{x})$ でモデル化したとする．つまり，$p(\mathrm{x})$ に基づいて発生する情報を近似的に $q(\mathrm{x})$ に基づくと仮定して，$q(\mathrm{x})$ を使って x を受信者に伝える状況を考える．このとき，x の値を特定するために必要な追加情報量の平均は

$$\begin{aligned}\mathrm{KL}(p||q) &= -\int p(\mathrm{x}) \ln q(\mathrm{x})\, \mathrm{dx} - \left(-\int p(\mathrm{x}) \ln p(\mathrm{x})\, \mathrm{dx}\right) \\ &= -\int p(\mathrm{x}) \ln \left\{\frac{q(\mathrm{x})}{p(\mathrm{x})}\right\} \mathrm{dx}\end{aligned}$$

となる．これを，分布 $p(x)$ と $q(x)$ の間を相対エントロピーまたはカルバック-ライブラー情報量と呼ぶ．

A.3　対数損失の意義

Change Finder において，対数損失を用いることが提案されている．対数損失の利点としては，計算量が二乗誤差の和の計算よりも少なくて済むことが挙げられる．対数損失の意味は，カルバック-ライブラー情報量の観点から説明することができる．データが未知の分布 $p(\mathrm{x})$ に従って生成されているとするとき，これをモデル化する．モデル化に当たって，パラメータ θ（変数）を持つパラメトリックな分布 $q(\mathrm{x}|\theta)$ を使って近似することを考える．θ を定める一つの方法として，$p(\mathrm{x})$ と $q(\mathrm{x}|\theta)$ の間のカルバック-ライブラー情報量を最小化することを考えることができる．$p(\mathrm{x})$ を直接知ることはできないため，$p(\mathrm{x})$ から得られた有限個の訓練点 $\mathrm{x}_n\,(n = 1, 2, ..., N)$ を用いて近似することを考える．このときのカルバック-ライブラー情報量は

$$\mathrm{KL}\,(p||q) \simeq \frac{1}{N}\sum_{n=1}^{N}\{-\ln q\,(\mathrm{x}_n|\theta) + \ln p\,(\mathrm{x}_n)\}$$

となる．つまり，カルバック-ライブラー情報量の最小化は，尤度の最大化と同等である．Change Finder で用いる対数損失は，この観点の文脈から生じたものと考えられる．

付 録B　プログラム

今回用いた，Change Finder のプログラムである．言語は R を用いた．

```
###データは.txt 形式で縦に一列に並べる.
###scorext は第一段階学習でのスコア．外れ値はここではじくことができる.
###score は第二段階学習でのスコア．変化点はこの値を見ればよい.
#####変化点検知用

###########step1
x <- read.table("yasai1.txt") ###データの読み込み
n <- nrow(x)
scorext <-matrix(0, nrow=n, ncol=1)
pt <-matrix(0, nrow=n, ncol=1)
###窓の設定
Tst1 <- 3
Tst3 <- 5
###AR モデルの自由度
kst1 <- 3
kst3 <- 2
###忘却パラメータ
rst1 <- 0.02
rst3 <- 0.002
xmean <- mean(x[,1])
Cst1 <- matrix(0, nrow=kst1+1, ncol=n)
Cst1e <- matrix(0, nrow=kst1, ncol=1)
Cmatst1 <- matrix(0, nrow=kst1, ncol=kst1)
nmatst1 <- matrix(0, nrow=kst1, ncol=kst1)
wst1 <- matrix(0, nrow=kst1, ncol=n)
xt <- matrix(0, nrow=n, ncol=1)
xtemp <- 0
#must1 <- xmean
must1 <- 0
sigmast1 <- matrix(0, nrow=n, ncol=1)
```

```
kstartst1 <- kst1+1
for (t in kstartst1 : n ) {
  must1 <- (1-rst1)*must1 + rst1*x[t,1]
  for (j in 1 : kstartst1){
    Cst1[j,t] <- (1-rst1)*Cst1[j,t-1]+rst1*(x[t,1]-must1)*(x[t-j+1,1]-must1)
  }
  Cmatst1 <- matrix(0, nrow=kst1, ncol=kst1)
  for (j in 1 : kst1){
    nmatst1 <- matrix(0, nrow=kst1, ncol=kst1)
    for (i in 1 : kst1){
      for (l in 1 : kst1){
        if (abs(l-i) == j-1){
          nmatst1 <- matrix(0, nrow=kst1, ncol=kst1)
          nmatst1[l,i] <- 1
          Cmatst1 <- Cmatst1 + Cst1[j,t]*nmatst1
        }
      }
    }
  }
  for (i in 1:kst1){
    Cst1e[i]<-Cst1[i+1,t]
  }
  wst1[,t] <- solve(Cmatst1,Cst1e)
  for (i in 1 : kst1){
    xtemp <- xtemp + wst1[i,t]*(x[t-i,1]-must1)
    xt[t] <- xtemp + must1
  }
  xtemp <- 0
  if (t == kstartst1){
    sigmast1[t] <- (x[t,1]-xt[t])*(x[t,1]-xt[t])
  }
  if (t != kstartst1){
    sigmast1[t] <- (1-rst1)*sigmast1[t-1] + rst1*(x[t,1]-xt[t])*(x[t,1]-xt[t])
  }
  pt[t] <- exp((-(xt[t]-x[t,1])^2)/(2*sigmast1[t]))*(1/(2*pi*sqrt(sigmast1[t])))
  scorext[t] <- -log(pt[t])
}
```

```
#######step2

y <- matrix(0, nrow=n, ncol=1)
ytemp <- 0
step2kst1 <- kst1+Tst1
for (t in step2kst1 : n ) {
  kstartst2 <- t-Tst1+1
  for (i in kstartst2 : t){
    ytemp <- ytemp + scorext[i]
  }
  y[t] = ytemp/Tst1
  ytemp <- 0
}


#########step3

scoreyt <-matrix(0, nrow=n, ncol=1)
qt <-matrix(0, nrow=n, ncol=1)
ytempst3 <- 0


Cst3 <- matrix(0, nrow=kst3+1, ncol=n)
Cmatst3 <- matrix(0, nrow=kst3, ncol=kst3)
Cst3e <- matrix(0, nrow=kst3, ncol=1)
nmatst3 <- matrix(0, nrow=kst3, ncol=kst3)
wst3 <- matrix(0, nrow=kst3, ncol=n)
yt <- matrix(0, nrow=n, ncol=1)
ysum <- sum(y)
#must3 <- ysum/(n-Tst1-kst1+1)
must3 <- 0
sigmast3 <- matrix(0, nrow=n, ncol=1)


kstartst3 <- kst1+Tst1+1+kst3
kstartst32 <- kst3+1
for (t in kstartst3 : n ) {
  must3 <- (1-rst3)*must3 + rst3*y[t]
```

```
  for (j in 1 : kstartst32){
    Cst3[j,t] <- (1-rst3)*Cst3[j,t-1] + rst3*(y[t,1]-must3)*(y[t-j+1,1]-must3)
  }
  Cmatst3 <- matrix(0, nrow=kst3, ncol=kst3)
  for (j in 1 : kst3){
    nmatst3 <- matrix(0, nrow=kst3, ncol=kst3)
    for (i in 1 : kst3){
      for (l in 1 : kst3){
        if (abs(l-i) == j-1){
          nmatst3 <- matrix(0, nrow=kst3, ncol=kst3)
          nmatst3[l,i] <- 1
          Cmatst3 <- Cmatst3 + Cst3[j,t]*nmatst3
        }
      }
    }
  }
  for (i in 1:kst3){
    Cst3e[i]<-Cst3[i+1,t]
  }
  wst3[,t] <- solve(Cmatst3,Cst3e)
  ytempst3 <- 0
  for (i in 1 : kst3){
    ytempst3 <- ytempst3 + wst3[i,t]*(y[t-i]-must3)
    yt[t] <- ytempst3 + must3
  }
  if (t==kstartst3){
    sigmast3[t] <- (y[t]-yt[t])*(y[t]-yt[t])
  }
  if (t!=kstartst3){
    sigmast3[t] <- (1-rst3)*sigmast3[t-1] + rst3*(y[t]-yt[t])*(y[t]-yt[t])
  }
  qt[t] <- exp(((yt[t]-y[t])^2)/(2*sigmast3[t])*(-1))*(1/(2*pi*sqrt(sigmast3[t])))
  scoreyt[t] <- -log(qt[t])
}

score <- matrix(0, nrow=n, ncol=1)
scoretemp <- 0
winsst3 <- kst1+Tst1+Tst3
for (t in winsst3 : n ) {
```

```
  kstartscore <- t - Tst3 +1
  for (i in kstartscore : t){
    scoretemp <- scoretemp + scoreyt[i]
  }
  score[t] = scoretemp/Tst3
  scoretemp <- 0
}

#plot(x[,1],xlab="days",ylab="sales quantity",type="o",main="niku6")
#par(new=T)
plot(score,xlab="days",ylab="score",col="red",main="niku1",type="o")
```

付 録C　データの出典

野菜と肉のデータは，食品スーパーのある店舗における 2013 年 10 月 17 日から 12 月 25 日の 70 日間の販売数量である．