

2019年度

修士論文

N次元高分子鎖の統計学による時系列解析

Time series analysis in terms of the statistics of
n-dimensional polymer chains

指導教員

古沢 浩 教授

高知工科大学大学院 工学研究科

基盤工学専攻 環境数理コース

学籍番号 1225084

川下 慈順

Yoshiyuki Kawashita

はじめに

本論文は、2018年4月から2020年3月まで、高知工科大学大学院工学研究科基盤工学専攻環境数理コース古沢研究室において著者が行った「N次元高分子鎖の統計学による時系列解析」についての研究をまとめたものである。

目次

| | |
|-----------------------------|----|
| はじめに | 1 |
| 第 1 章 緒論 | 5 |
| 1.1 研究背景 | 5 |
| 1.2 研究目的 | 7 |
| 第 2 章 解析データ及び方法 | 8 |
| 2.1 解析データ | 8 |
| 2.2 解析方法 | 10 |
| 第 3 章 結果 | 16 |
| 3.1 2次元における標準化データの定性的特徴 | 16 |
| 3.2 3次元における高分子的特徴（幾何的観点） | 18 |
| 3.3 3次元における高分子的特徴（定量的観点） | 20 |
| 3.4 慣性半径（3次元）の相対的順位と市場組み合わせ | 22 |
| 第 4 章 比較 | 24 |
| 4.1 地方度による統計量の評価（3次元） | 24 |
| 4.2 地方度と県間距離による評価（3次元） | 26 |
| 4.3 4次元への拡張 | 28 |
| 第 5 章 結論 | 30 |
| 参考文献 | 31 |
| 付録 | 32 |
| 付録 1 Python によるスクレイピングコード | 32 |
| 付録 2 R による統計量算出コード | 34 |
| 謝辞 | 45 |

目次

| | | |
|------|---|----|
| 1.1 | 1次元における n 種の時系列データのイメージ図 | 5 |
| 1.2 | 高次元超立方体のイメージ図 | 5 |
| 1.3 | 主成分分析のイメージ図 | 6 |
| 1.4 | 等方的データのイメージ図 | 6 |
| 1.5 | 平均曲率と脈波伝搬の関係 | 7 |
| 2.1 | 対象市場のある都道府県 (赤色) | 8 |
| 2.2 | 拡張した標準化のイメージ図 | 9 |
| 2.3 | 幾何的マッピングのイメージ図 | 10 |
| 2.4 | 幾何的マッピングのデータ例 | 10 |
| 2.5 | バネビーズモデル | 11 |
| 2.6 | 曲率の幾何学的イメージ図 | 12 |
| 2.7 | 高分子のランダムコイルと慣性半径 | 13 |
| 2.8 | 慣性半径のイメージ図 | 14 |
| 2.9 | 経路長 S のイメージ図 | 14 |
| 2.10 | 高分子のスケーリング則 | 15 |
| 3.1 | 2市場の散布図の例 (沖縄、高松) | 16 |
| 3.2 | 2市場の散布図の例 (大阪、東京) | 16 |
| 3.3 | 2市場の全組み合わせにおける相関係数 | 17 |
| 3.4 | 任意の3市場の野菜の標準化価格データの3次元グラフ | 18 |
| 3.5 | 時間差 τ と $\sqrt{\langle R ^2 \rangle}$ の両常用対数プロット | 19 |
| 3.6 | ねぎの3市場の全組み合わせ ${}_{14}C_3 = 364$ 通りにおける $\frac{\text{慣性半径}}{\text{ボンド長平均}}$ | 20 |
| 3.7 | 全品目で慣性半径が最大値・最小値をとるときの市場 | 22 |
| 4.1 | 地方度導出に伴う L のイメージ図 | 24 |
| 4.2 | 3次元地方度と慣性半径の散布図 | 25 |
| 4.3 | 3次元地方度と曲率平均の散布図 | 25 |
| 4.4 | 県間距離導出に伴う D のイメージ図 | 26 |
| 4.5 | キャベツにおける4次元地方度と慣性半径の散布図 | 28 |

表目次

| | | |
|-----|---|----|
| 2.1 | 高分子と時系列データの対応関係 | 11 |
| 3.1 | $\frac{\text{慣性半径}}{\text{ボンド長平均}}$ の各品目に対する平均・分散 | 21 |
| 3.2 | 慣性半径が上位 3 位 (降べきの順) と下位 3 位 (昇べきの順) のときの市場組み合わせ | 23 |
| 4.1 | 各品目に対する統計量と 3 次元地方度の相関係数 | 25 |
| 4.2 | 重回帰分析の結果 | 27 |
| 4.3 | 4 次元における慣性半径 (上位・下位) と市場組み合わせ | 28 |
| 4.4 | 各品目に対する慣性半径と 4 次元地方度の相関係数 | 29 |

第1章

緒論

1.1 研究背景

1.1.1 多変量統計解析の問題点1

1次元で n 種の時系列データを扱う場合を考える。図 1.1 は、1次元における n 種の時系列データのイメージ図である。ここでの次元は時刻 t のパラメータ空間 $D_n = (d_1, d_2, d_3, \dots, d_n)$ を表す。高次元では、次元の呪いが発生する。これは、リチャード・ベルマンが使ったもので空間の次元が増えるのに対応して、問題の算法が指数関数的に増加するというものである。また、高次元空間における性質として超立方体の中心から頂点までの距離は、次元が増えるほど長くなり、データは表面的なものだけになる（図 1.2）。次元の呪いの回避方法として、特徴の選択や次元の削減がある。

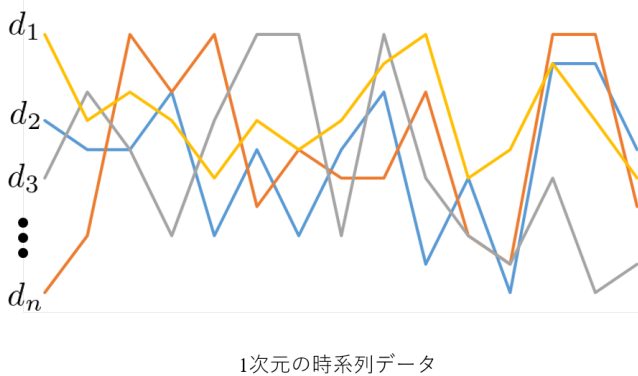


図 1.1 1次元における n 種の時系列データのイメージ図

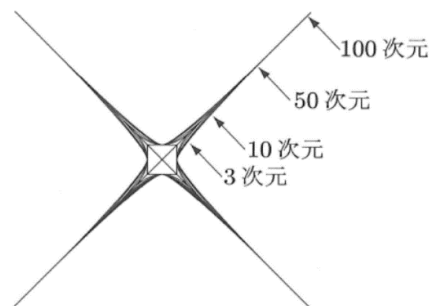


図 1.2 高次元超立方体のイメージ図

1.1.2 多変量統計解析の問題点 2

多変量解析手法の代表例として、主成分分析、因子分析、重回帰分析が挙げられる。ここでは、主成分分析を例として問題点を述べる。主成分分析は、多変量のデータに対し、異方性がある新変数を設定し、データを解析する分析方法である。図 1.3 は、主成分分析のイメージ図である [1]。最も異方性がある方向を軸として解釈する。

多変量解析の課題点として、等方的データの場合には散布図上に特徴のある方向がないため有効な分析にならないことが挙げられる (図 1.4)。

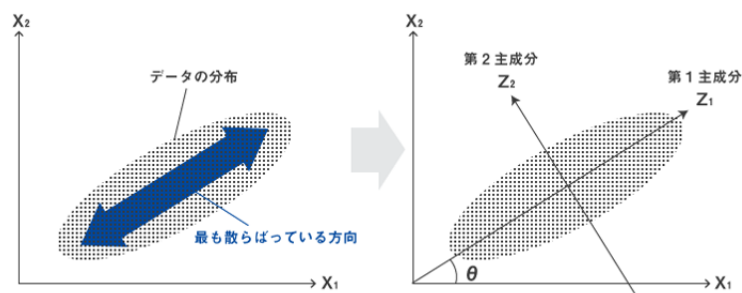


図 1.3 主成分分析のイメージ図



図 1.4 等方的データのイメージ図

1.1.3 多変量解析への幾何学の適用

多変量解析の新しい試みとして、チェコのグループがウェアラブルセンサーによって得られる生物医学的な多変量時系列データの解析に曲率を用いて幾何的性質を抽出した例が挙げられる [2]。具体的には、 n 個のセンサーを体の様々な部位に取り付け、 n 次元の脈波が各センサーに到達する時間と平均曲率の極大値に対応関係があることを示した。図 1.5 は、平均曲率と脈波伝搬の関係である。これにより、視覚では読み取りづらい多次元量の特性を曲率という統計量を用いて把握することを可能とした。

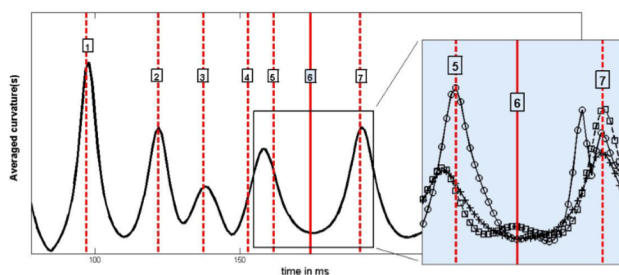


図 1.5 平均曲率と脈波伝搬の関係

1.2 研究目的

前節の多変量の現状を踏まえ、本研究では、多変量時系列データ解析を高次元空間上で行い、幾何的な統計量を算出することにより相関が弱く、等方的な時系列データの特徴量を抽出することを大目的とした。

多変量時系列データの例としては、複数箇所での流量・降水量などのフィールドデータや複数品目・銘柄に関する価格などの流通データなどが挙げられる。本研究では、複数箇所の市場価格として野菜を用いて新たな手法開発を行った。研究の詳細を以下に述べる。まず、野菜の価格データを高分子鎖アナロジーを用いて解析した。次に、1日スケールの短期的変化、数日スケールの中期的変化、1か月以上のスケールの長期的変化の特徴を踏まえた幾何的な特徴量をそれぞれ作成し評価した。最後にこれらの統計量を次元を拡張して評価した。

第2章

解析データ及び方法

2.1 解析データ

2.1.1 野菜価格のデータ

本研究では、e-stat 政府統計の統合窓口 [3] から得た青果物卸売市場の野菜価格の日時データを使用した。対象品目はキャベツ、だいこん、トマト、なす、ねぎの5品目とし、対象市場は、沖縄、福岡、北九州、広島、高松、神戸、京都、大阪、金沢、名古屋、東京、神奈川、仙台、札幌の14市場とした。対象市場がある都道府県を赤く塗った日本地図を以下に示す(図2.1)。

また、本研究に使用したデータの期間は、2017年5月から2019年5月の2年間を採用した。

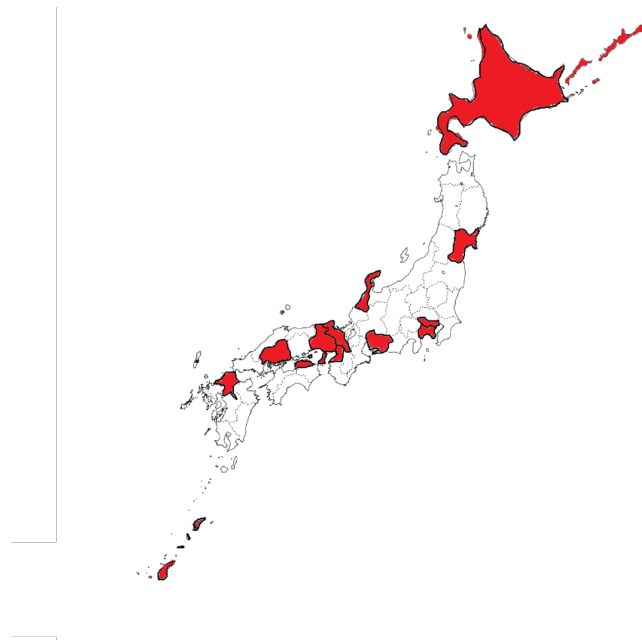


図 2.1 対象市場のある都道府県 (赤色)

2.1.2 拡張した標準化の定義

本研究で用いる拡張した標準化を以下のように定める。まず、野菜価格データに対して各市場ごとの合計で割り、次に日ごとについて以下の式

$$Z_{ij} = \frac{F_{ij} - F(x)}{S(x)}$$

で標準化した。ここで、 $F(x)$ は日ごとの 14 市場価格の平均値、 $S(x)$ は日ごとの 14 市場価格の標準偏差を表す。これにより、各市場が 14 市場の平均から相対的にどのくらい変化しているか、すなわち 14 市場における相対的ポジションの日変動を意味するデータに変換したことになる。

図 2.2 は、拡張した標準化のイメージ図である。以下、拡張した標準化を単に標準化と記す。

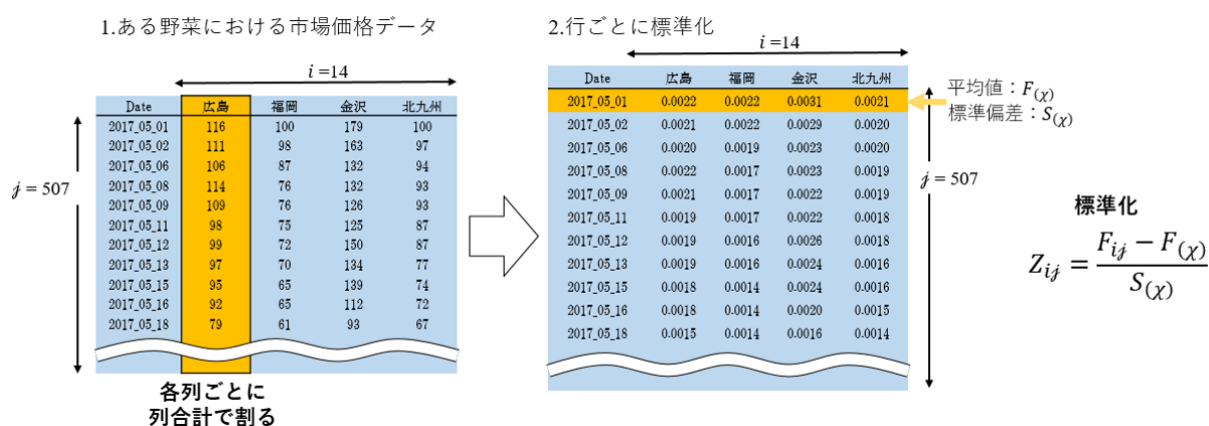


図 2.2 拡張した標準化のイメージ図

2.1.3 解析装置

解析装置は、以下の 2 点を利用した。

- Python
- R

なお、Python は主としてスクレイピング [4] によるデータ取得を目的として、R はデータの解析を目的として利用した。それぞれのコードは付録に記した。

2.2 解析方法

2.2.1 多変量時系列データの幾何的マッピング

特定時間における複数時系列データの任意の組み合わせを位置座標とし、多変量時系列データの幾何的マッピングを行った。図 2.3 は 3 次元における幾何的マッピングのイメージ図である。左の図は、任意の 3 市場に対して野菜価格データをマッピングしたイメージ図であり、点の座標は各市場の野菜価格を表している。右の図は、前節で述べた標準化を行った野菜価格データをマッピングしたイメージ図である。点の座標は各市場の標準化した野菜価格を表している。点と点を結んだ線は、時間の変化を表す。

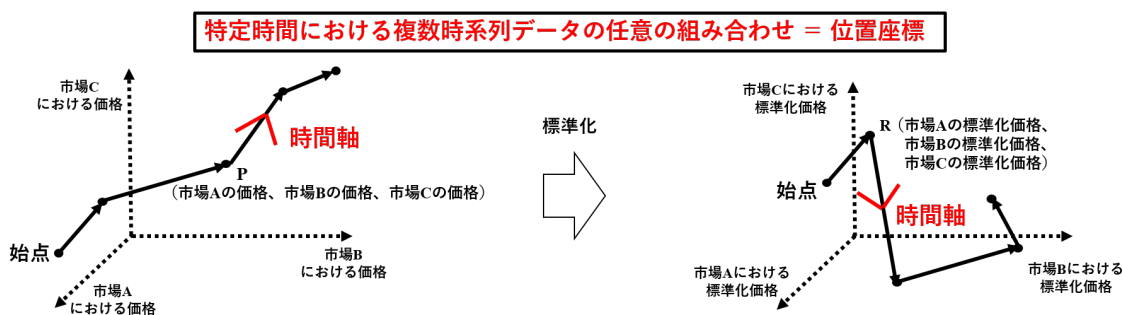


図 2.3 幾何的マッピングのイメージ図

図 2.4 は 3 次元における幾何的マッピングのデータ例である。左の図が野菜価格データのマッピング図の例であり、点と点を結んだ線がある方向に伸びたような幾何学的なふるまいをしている。一方、右の図は標準化した野菜価格データのマッピング図の例であり、線の軌跡がよりランダムな幾何学的なふるまいをしているように見える。

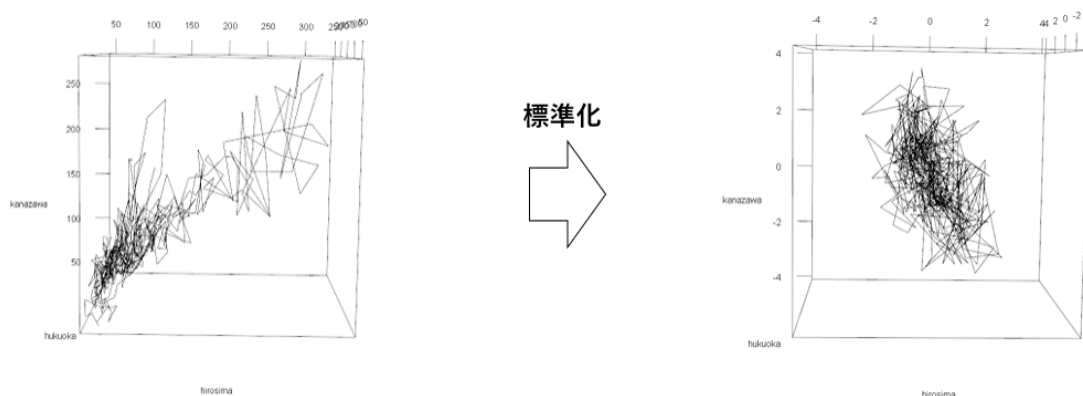


図 2.4 幾何的マッピングのデータ例

2.2.2 高分子アナロジー

前節の標準化した野菜価格データのマッピング図において、点間の線の長さは時間ごとに変動している。そこで、高分子鎖の主モデルの1つであるバネビーズモデル（図 2.5）にアナロジーした。なお、バネビーズモデルとは玉がバネでつながったモデルであり、ボンドの長さが一定でなく可変なモデルである。

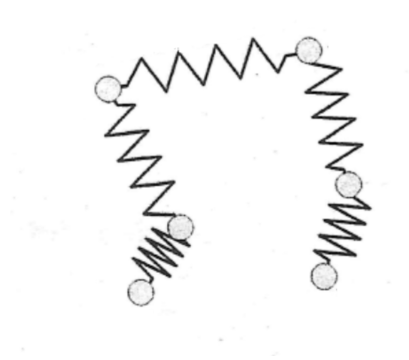


図 2.5 バネビーズモデル

高分子と時系列データの対応関係を表 2.1 に示す。高分子における位置座標、総モノマー数、任意のモノマー間のモノマー数をそれぞれ時系列データの特定時間の複数の組み合わせ、全期間の総日数、時間差 τ に対応させる。次節以降では、時系列データを高分子学的に解析するための統計量について述べていく。

表 2.1 高分子と時系列データの対応関係

| 高分子 | 時系列データ |
|----------------|---------------|
| 位置座標 | 特定時間の複数の組み合わせ |
| 総モノマー数 | 全期間の総日数 |
| 任意のモノマー間のモノマー数 | 時間差 τ |

2.2.3 ボンド長平均の定義

短期的な幾何的特徴を抽出する統計量として、ボンド長平均を次式のように定義した。

$$\langle \vec{R}_n \rangle = \frac{1}{n} \sum_{k=1}^n |\vec{R}_k| \quad (\vec{R}_n = R_{n+1} - R_n) \quad (2.1)$$

ここで、 R_n は n 番目の点の座標を表す。ボンド長 \vec{R}_n は本研究において、任意の市場における日付 n と日付 $n+1$ に対する標準化した野菜価格データの座標の点間の長さである。これは日ごとの変動の激しさと変動の増加・減少（前日比）を反映した量であると解釈できる。高分子的観点においてはモノマー間の結合の大きさとみなせる。

また、ボンド長平均 $\langle \vec{R}_n \rangle$ は、任意の市場の組み合わせにおける $n-1$ 個のボンド長の平均値であり、各組み合わせに対して1つの値が算出できる。

2.2.4 曲率平均の定義

中期的な幾何的特徴を抽出する統計量として、曲率平均を用いた。曲率を次式のように定義した。本研究では、日時データを扱っているため、微分幾何ではなく離散幾何の曲率 [5] を採用した。

$$\kappa_n = 2 \tan \frac{\phi_n}{2} \quad (2.2)$$

ここで、 $\phi_n = \cos^{-1} \frac{\vec{R}_{n-1} \cdot \vec{R}_n}{|\vec{R}_{n-1}| |\vec{R}_n|}$ とする。本研究における曲率 κ_n は、日付 $n-1, n, n+1$ に対する標準化した野菜価格データの座標の3日にわたる2本のボンドの曲がり具合を表す。また、 ϕ_n は、2本のベクトルの外角を表す。曲率の幾何学的イメージ図（図 2.6）を以下に示す。曲率 κ_n の平均値を曲率平均とする。

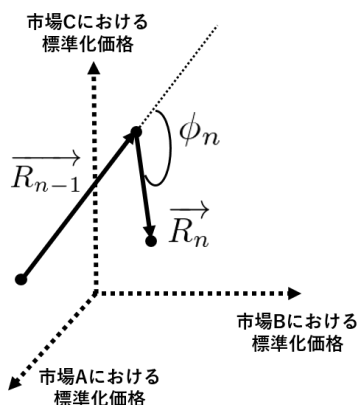


図 2.6 曲率の幾何学的イメージ図

2.2.5 慣性半径の定義

高分子学では、高分子鎖の拡がりを表す統計量として分子鎖の重心からの拡がりを表す慣性半径が用いられる。図 2.7 は、高分子のランダムコイルの図であり、点線は慣性半径を表している。なお、ランダムコイルとはポリマーを構成するモノマーが隣接したモノマーと結合しながらランダムに配向したものである。



図 2.7 高分子のランダムコイルと慣性半径

そこで、本研究の時系列データに対し長期的な幾何的特徴を抽出する統計量として慣性半径 [6] を以下のように定義した。

$$R_g \equiv \frac{1}{n} \sum_{i=1}^n R_i \quad (2.3)$$

$$S_i \equiv R_i - R_g \quad (2.4)$$

$$\sqrt{\langle s^2 \rangle} = \sqrt{\frac{1}{n} \sum_{i=1}^n S_i^2} \quad (2.5)$$

ここで、(2.3) 式は、各座標の加算平均に相当し、 R_g は中心座標すなわち重心を表している。また、 S_i は重心 R_g から日付 i の座標へのベクトルとし (2.4 式)、慣性半径は S_i の 2 乗平均平方根として導出した (2.5 式)。これは、任意の市場組み合わせに対する時系列データの全期間におけるふるまいを反映した量であると解釈できる。図 2.8 は、本研究における座標空間上での慣性半径のイメージ図である。

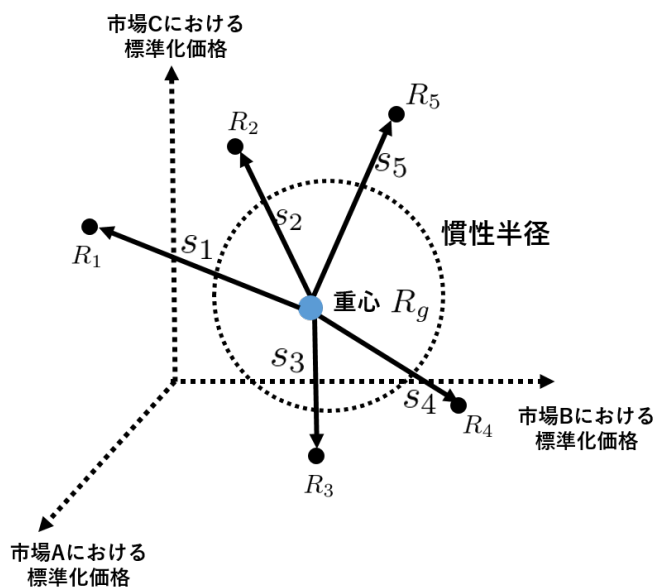


図 2.8 慣性半径のイメージ図

2.2.6 2 乗平均末端分子間距離の平方根の定義

経路長 S を任意の位置座標間の距離と定義すると、経路長 S の具体例として高分子では任意のモノマー間のモノマー数、時系列データでは時間差 τ とみなすことができる (2.2.2 節)。経路長 S のイメージ図を以下に示す (図 2.9)。

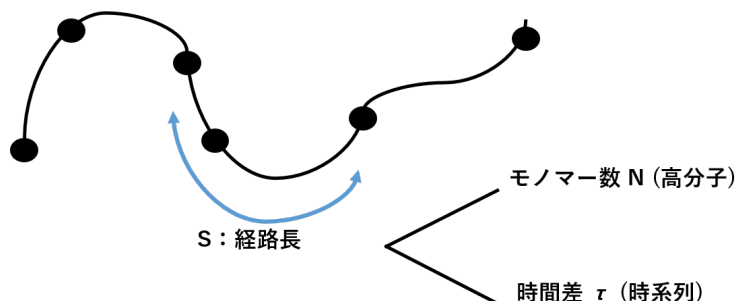


図 2.9 経路長 S のイメージ図

一般に、高分子には以下の関係があることが知られている。

$$\sqrt{\langle \|R\|^2 \rangle} \sim N^\alpha \quad (2.6)$$

ここで、 $\sqrt{\langle \|R\|^2 \rangle}$ は 2 乗平均末端分子間距離の平方根、すなわち経路長 S 、 N はモノマー数、 α はべき乗値を表す。べき乗値 α によって鎖の拡がり方が異なり、 $\alpha = 0.6$ のとき排除体積鎖、 $\alpha = 0.5$ のとき理想鎖、 $\alpha = \frac{1}{3}$ のとき凝集鎖と呼ばれており、それぞれ鎖が膨潤、 R の確率分布がガウス分布に従う、鎖が凝集するという特性をもっている。これらの性質を高分子のスケーリング則 (図 2.10) という。

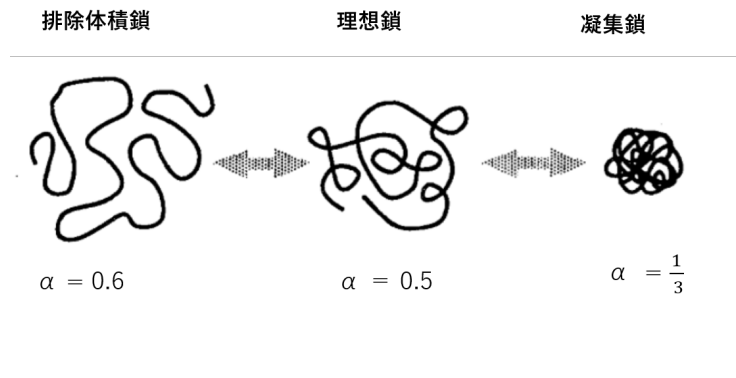


図 2.10 高分子のスケーリング則

そこで、本研究の時系列データに適応した 2 乗平均末端分子間距離の平方根を以下のように定義した。

$$\sqrt{\langle \|R\|^2 \rangle} = \sqrt{\langle \|r_\tau\|^2 \rangle} \sim \tau^\alpha \quad (2.7)$$

ここで、 r_τ は、経路長 S を時間差 τ として結んだベクトルを表す。これにより、時系列データの幾何学的なふるまいを高分子の鎖とみなして、時間差 τ のべき乗則で評価できる。

第3章

結果

3.1 2次元における標準化データの定性的特徴

以下の図は2市場における標準化した野菜価格の散布図の例である。図3.1は、沖縄と高松、図3.2は、大阪と東京の市場を示している。なお、初期日（2017年5月1日）のデータを原点とした。標準化により14市場の平均と各市場の相対的な日ごとの価格の差異を確認できる。図3.1は、横軸と縦軸の幅が約8であり14市場平均とのずれが大きく、図3.2は、横軸と縦軸の幅が約4で14市場平均とのずれが小さい。よって、14市場平均から地方の市場はずれが大きく、都市の市場ではずれが小さくなっていることが分かる。これは、都市の市場の野菜価格が14市場平均に近いことを示しており、都市市場が野菜の価格変動の基準となっていると考えられる。

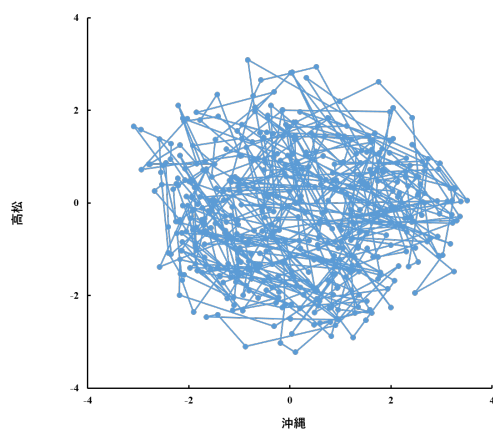


図 3.1 2市場の散布図の例（沖縄、高松）

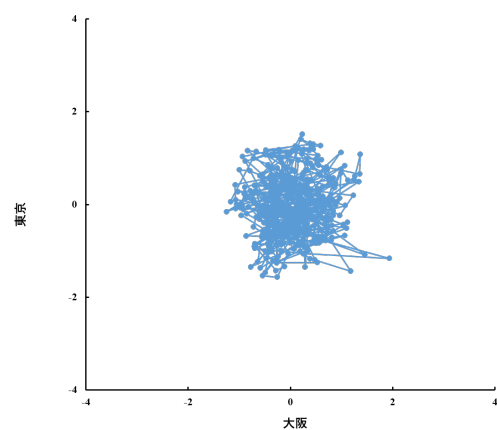


図 3.2 2市場の散布図の例（大阪、東京）

図 3.3 は、2 市場の全組み合わせにおける相関係数のグラフである。なお、対象品目はキャベツ、だいこん、トマト、なす、ねぎの 5 品目である。横軸は、 ${}_{14}C_2 = 91$ 通りの 2 市場の全組み合わせ、縦軸は、各組み合わせにおける相関係数を示している。このグラフから、ほとんど相関なし・低い相関とされる $-0.4 \sim 0.4$ の相関係数をとる組み合わせが多いことが分かる。これは標準化処理したデータは図 3.1、3.2 のように幾何的に等方なものが多いことを示している。

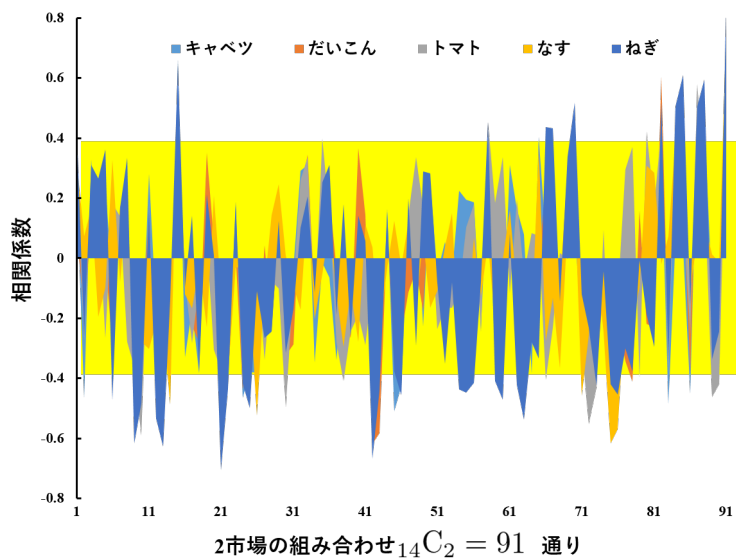


図 3.3 2 市場の全組み合わせにおける相関係数

3.2 3次元における高分子の特徴（幾何的観点）

図 3.4 は、任意の 3 市場の野菜の標準化価格データの 3 次元グラフである。左の図は全品目で慣性半径が最大である 3 市場のグラフである。品目はトマト、市場は沖縄、札幌、高松である。点と点を結んだ線の軌跡が、ひろがっており等方的にみえる。このとき、2 乗平均末端分子間距離の平方根を時間差 τ のべき乗に近似したときのべき乗値は $\alpha = 0.32$ であった。

真ん中の図は、異方的にみえる場合の 3 次元図である。線の軌跡が扁平にみえる。このとき、べき乗値は $\alpha = 0.25$ であった。

右の図は全品目で慣性半径が最小である 3 市場のグラフである。品目は、トマト、市場は名古屋、大阪、横浜である。線の軌跡が等方的で凝集しているようにみえる。このとき、べき乗値は $\alpha = 0.13$ であった。

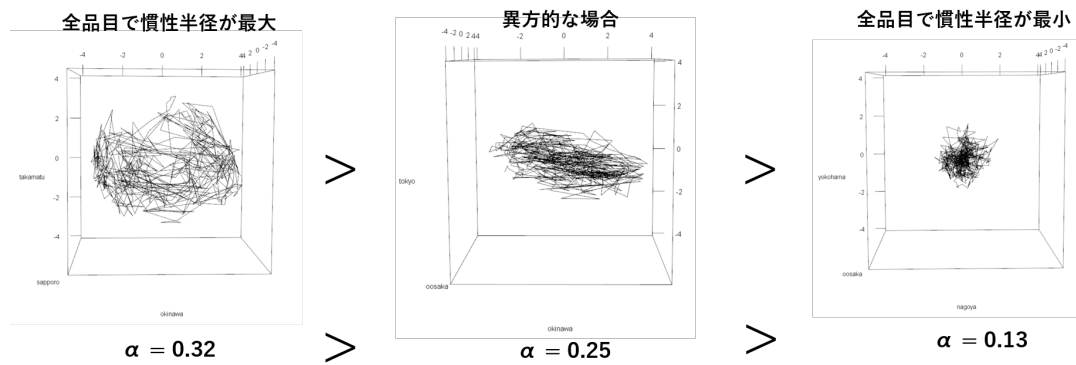


図 3.4 任意の 3 市場の野菜の標準化価格データの 3 次元グラフ

図 3.5 は、全品目で慣性半径が最大するとき（図 3.4 の左）における時間差 τ と $\sqrt{\langle \|R\|^2 \rangle}$ の両常用対数プロットである。傾き、すなわちべき乗値 α は 0.32 である。

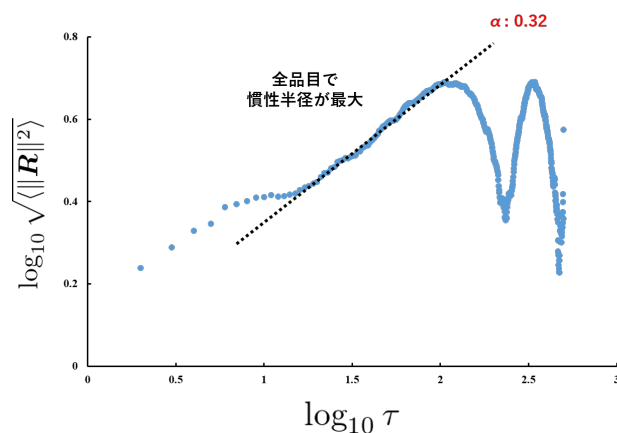


図 3.5 時間差 τ と $\sqrt{\langle \|R\|^2 \rangle}$ の両常用対数プロット

慣性半径が最大・最小、異方的な場合の全てにおいて $\alpha < 0.3$ であった。慣性半径が最大するとき、べき乗値 α が 0.32 であり、凝集鎖のスケーリング則 ($\alpha = \frac{1}{3}$) と一致している。よって、高分子的な凝集鎖とみなせる。また、慣性半径が最大、異方的、最小の順にべき乗値 α が減少している。これは時系列データに高分子のスケーリング則を適用でき、べき乗値 α から幾何学的な線の軌跡を推定できることを示唆している。本研究で用いた野菜価格の時系列データは、市場と関連が深くランダム的ではないので、凝集鎖になったと考えられる。

3.3 3次元における高分子の特徴（定量的観点）

前節では3次元における高分子の特徴を幾何的観点から述べた。本節では定量的観点から分析する。高分子では理想鎖（ガウス鎖）の慣性半径が

$$\sqrt{\langle s^2 \rangle_o} = \sqrt{\frac{1}{6}na^2} \quad (3.1)$$

となることが知られている。ここで、 a はボンド長、 n はモノマー数を表す。本研究での n は全期間の総日数を表す。(式 3.1) を式変形すると

$$\frac{\sqrt{\langle s^2 \rangle_o}}{a} = \sqrt{\frac{n}{6}} \quad (3.2)$$

となる。

図 3.6 の青線は、ねぎの 3 市場の全組み合わせ ${}_{14}C_3 = 364$ 通りの各組合せにおける $\frac{\text{慣性半径}}{\text{ボンド長平均}}$ の値である。ある値を基準とした横ばいのグラフとなっている。また、赤線はガウス鎖の場合の $\frac{\text{慣性半径}}{\text{ボンド長平均}}$ の値すなわち $\sqrt{\frac{n}{6}}$ である。青線は、赤線よりかなり小さい値をとっている。このため、定量的観点からも高分子的に凝集している状態だと分かる。

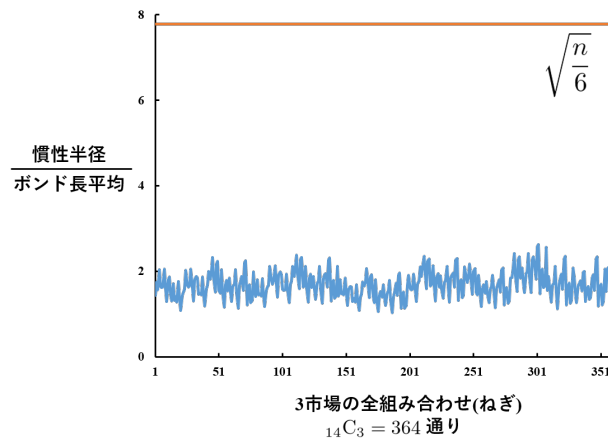


図 3.6 ねぎの 3 市場の全組み合わせ ${}_{14}C_3 = 364$ 通りにおける $\frac{\text{慣性半径}}{\text{ボンド長平均}}$

$\frac{\text{慣性半径}}{\text{ボンド長平均}}$ の各品目に対する平均・分散の表を以下に示す (表 3.1)。どの品目においてもすこのばらつきがあり、組み合わせによって誤差があるものの各品目において平均値を基準に値をとっていることが分かる。よって、慣性半径で各組合せの特徴を類別可能ではないかと考えた。また、全品目に対して、 $\frac{\text{慣性半径}}{\text{ボンド長平均}} < \sqrt{\frac{n}{6}}$ となっており、全ての品目において高分子的な凝集鎖であるとみなせる。

表 3.1 $\frac{\text{慣性半径}}{\text{ボンド長平均}}$ の各品目に対する平均・分散

| | キャベツ | だいこん | トマト | なす | ねぎ |
|----|-------|-------|-------|-------|-------|
| 平均 | 1.59 | 1.60 | 1.70 | 1.63 | 1.70 |
| 分散 | 0.033 | 0.077 | 0.080 | 0.075 | 0.100 |

3.4 慣性半径 (3 次元) の相対的順位と市場組み合わせ

前節の結果から慣性半径に着目し、慣性半径の相対的順位と市場の組み合わせを確認した。図 3.7 は、全品目で慣性半径が最大値・最小値をとるときの市場がある県を色付けした日本地図である。赤色で塗られた都道府県は、全品目で慣性半径が最大のときの市場組み合わせを示している。品目はトマトで市場組み合わせは沖縄、札幌、高松の 3 市場となった。一方、青色で塗られた都道府県は、全品目で慣性半径が最小のときの市場組み合わせを示している。品目はトマトで市場組み合わせは名古屋、大阪、横浜の 3 市場となった。

慣性半径が最大のときの市場組み合わせは、全て地方市場であり最小のときは全て都市市場となっている。ただし、日本の大都市である東京・大阪を基準とし、そこからの地理的距離が近い県にある市場を都市市場、遠い県にある市場を地方市場と定めた。



図 3.7 全品目で慣性半径が最大値・最小値をとるときの市場

慣性半径が上位3位（降べきの順）と下位3位（昇べきの順）のときの市場組み合わせを以下に示す（表3.2）。全品目に対して、慣性半径が上位のとき地方市場の組み合わせが多く、慣性半径が下位のとき都市市場の組み合わせが多い傾向にあると分かる。

表3.2 慣性半径が上位3位（降べきの順）と下位3位（昇べきの順）のときの市場組み合わせ

| | 順位 | キャベツ | だいこん | トマト | なす | ねぎ |
|--------------|----|-----------|-----------|-----------|-----------|-----------|
| 慣性半径 大きい順 | 1 | 沖縄_札幌_高松 | 沖縄_札幌_高松 | 沖縄_札幌_高松 | 沖縄_札幌_高松 | 京都_沖縄_札幌 |
| | 2 | 金沢_沖縄_札幌 | 北九州_沖縄_札幌 | 北九州_沖縄_札幌 | 北九州_沖縄_高松 | 北九州_沖縄_札幌 |
| | 3 | 北九州_沖縄_札幌 | 北九州_札幌_高松 | 金沢_沖縄_札幌 | 金沢_沖縄_高松 | 沖縄_札幌_仙台 |
| 慣性半径 小さい順 | 1 | 神戸_大阪_東京 | 名古屋_大阪_東京 | 名古屋_大阪_横浜 | 京都_東京_横浜 | 名古屋_大阪_東京 |
| | 2 | 神戸_名古屋_大阪 | 名古屋_東京_横浜 | 名古屋_大阪_東京 | 京都_大阪_東京 | 金沢_名古屋_大阪 |
| | 3 | 神戸_大阪_横浜 | 名古屋_大阪_横浜 | 京都_名古屋_大阪 | 大阪_東京_横浜 | 広島_名古屋_大阪 |

以上より、慣性半径の大小で任意の市場組み合わせを地方性という特徴で類別可能であると考えることができる。

第4章

比較

4.1 地方度による統計量の評価（3次元）

前章では、慣性半径により地方性の検知が可能であると示唆していることを述べた。本節ではそれを定量的に評価するため、3次元の場合の地方度を

$$Loc = \frac{1}{3}(\min(L1, L2) + \min(L3, L4) + \min(L5, L6)) \quad (4.1)$$

と定義して検討した。ここで、 L は任意の組み合わせ 3 市場の各市場と東京・大阪への地理的距離を表す。また、地方度 Loc （式 4.1）は、任意の組み合わせ 3 市場の各市場と東京・大阪への距離が小さい方の平均値を表す。

図 4.1 は、 L のイメージ図である。任意の組み合わせ 3 市場をそれぞれ市場 A、市場 B、市場 C とし、各市場から東京・大阪への距離をそれぞれ ($L1$ 、 $L2$)、($L3$ 、 $L4$)、($L5$ 、 $L6$) としている。

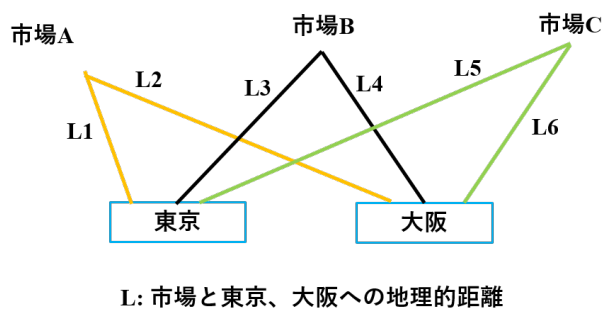


図 4.1 地方度導出に伴う L のイメージ図

図 4.2 は、キャベツにおける 3 次元地方度と慣性半径の散布図である。横軸は 3 次元地方度、縦軸は慣性半径を表す。相関係数は 0.83 となり高い相関を示した。図 4.3 は、キャベツにおける 3 次元地方度と曲率平均の散布図である。横軸は 3 次元地方度、縦軸は曲率平均を表す。相関係数は -0.20 となり相関なしであった。よって、慣性半径は地方性を反映していると考えられる。

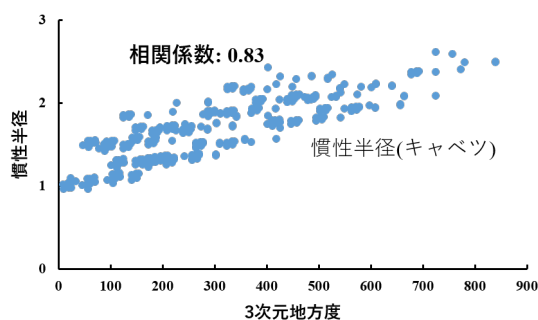


図 4.2 3 次元地方度と慣性半径の散布図

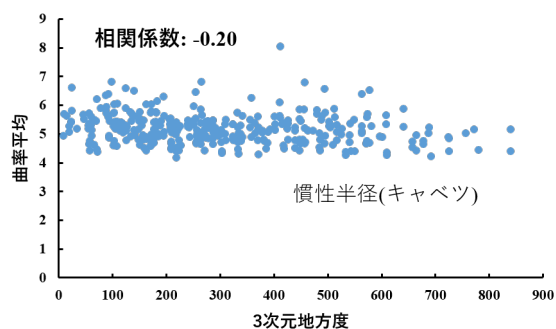


図 4.3 3 次元地方度と曲率平均の散布図

各品目に対する統計量と 3 次元地方度の相関係数の表を以下に示す (表 4.1)。各品目に対して、ボンド長平均・慣性半径・曲率平均の 3 種統計量と地方度の相関係数を導出している。ボンド長と慣性半径は地方度との相関係数が大きく、曲率平均は地方度と相関なしであった。また、各品目において全て慣性半径の方がボンド長平均よりも相関係数が大きい。よって、ボンド長平均と慣性半径は地方度を類別できており、曲率平均は地方度を類別できていないことが明らかとなった。また、慣性半径がより地方度を類別していることから慣性半径が地方度類別に最も適していると分かる。

表 4.1 各品目に対する統計量と 3 次元地方度の相関係数

| | キャベツ | だいこん | トマト | なす | ねぎ |
|--------|-------|------|------|------|-------|
| ボンド長平均 | 0.76 | 0.53 | 0.79 | 0.80 | 0.48 |
| 慣性半径 | 0.83 | 0.81 | 0.91 | 0.84 | 0.69 |
| 曲率平均 | -0.20 | 0.13 | 0.24 | 0.01 | -0.20 |

4.2 地方度と県間距離による評価（3次元）

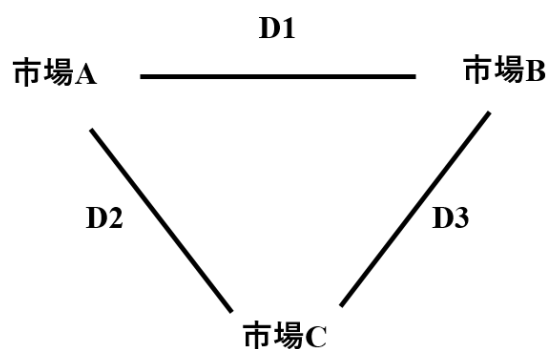
前節では、各統計量を地方度によって定量的に評価した。本節では、他にも市場と関係があると考えられる県間距離を定義し、各統計量を評価した。また、県間距離および地方度と統計量を重回帰分析によっても評価した。

3次元の場合の県間距離を

$$d = \frac{1}{3}(D1 + D2 + D3) \quad (4.2)$$

と定義した。ここで、 D は任意の組み合わせ 3 市場の各市場がある県間の地理的距離を表す。また、県間距離 d (式 4.2) は、任意の組み合わせ 3 市場がある各県間の距離平均に相当する。

図 4.4 は、 D のイメージ図である。任意の組み合わせ 3 市場をそれぞれ市場 A、市場 B、市場 C とし、(市場 A、市場 B)、(市場 A、市場 C)、(市場 B、市場 C) がある県間の地理的距離をそれぞれ $D1$ 、 $D2$ 、 $D3$ とした。



D: 市場がある県間の地理的距離

図 4.4 県間距離導出に伴う D のイメージ図

3次元地方度、3次元県間距離、統計量を標準化し、各統計量を目的変数として重回帰分析を行った。

定義式は以下のようなになる。

$$y = ax_1 + bx_2 + c \quad (4.3)$$

ここで、 y は各統計量（ボンド長平均、慣性半径、曲率平均）、 x_1 は県間距離 x_2 は地方度を表す。

重回帰分析の結果を以下に示す（表 4.2）。各品目における 3 種の統計量について 3 県間距離係数、3次元地方度係数、3 県間距離の p 値、3次元地方度の p 値、重決定係数 R^2 の値を示している。

前述の重回帰分析により、係数を比べることで係数の大小から各変数の重要性を判断することができる。全品目についてボンド長平均・慣性半径は県間距離よりも地方度の係数が大きく、本

研究で用いたデータは地方度に連動していることが明らかとなった。また、推定された回帰式の当てはまりの良さを表す重決定係数 R^2 はねぎ以外の全品目でボンド長平均・慣性半径が大きい傾向にあったため、この2つの統計量は、地方度と県間距離で説明できる傾向にあることを示唆している。

表 4.2 重回帰分析の結果

| 品目 統計量 | キャベツ | | | だいこん | | | トマト | | | なす | | | ねぎ | | |
|-----------|------------|----------|----------|------------|----------|----------|------------|----------|----------|------------|----------|----------|------------|----------|----------|
| | ボンド長 平均 | 慣性 半径 | 曲率 平均 | ボンド長 平均 | 慣性 半径 | 曲率 平均 | ボンド長 平均 | 慣性 半径 | 曲率 平均 | ボンド長 平均 | 慣性 半径 | 曲率 平均 | ボンド長 平均 | 慣性 半径 | 曲率 平均 |
| 3県間距離係数 | -0.33 | 0.094 | 0.07 | -0.55 | 0.039 | 0.35 | -0.35 | 0.11 | 0.32 | 0.04 | 0.12 | 0.09 | -0.29 | 0.26 | 0.05 |
| 地方度係数 | 1.05 | 0.75 | -0.26 | 1.01 | 0.78 | -0.18 | 1.09 | 0.81 | -0.05 | 0.76 | 0.73 | -0.07 | 0.73 | 0.47 | -0.25 |
| 3県間距離p値 | 1.5E-06 | 1.2E-01 | 5.1E-01 | 1.0E-09 | 5.4E-01 | 9.5E-04 | 1.5E-07 | 1.2E-02 | 2.2E-03 | 5.4E-01 | 4.0E-02 | 3.9E-01 | 2.2E-03 | 1.1E-03 | 6.1E-01 |
| 地方度p値 | 1.4E-41 | 2.3E-29 | 1.5E-02 | 1.5E-26 | 3.9E-29 | 8.4E-02 | 8.0E-48 | 3.8E-53 | 6.6E-01 | 7.3E-27 | 4.2E-30 | 5.1E-01 | 1.1E-13 | 4.9E-09 | 1.9E-02 |
| 重決定 R2 | 0.61 | 0.69 | 0.04 | 0.35 | 0.66 | 0.05 | 0.65 | 0.83 | 0.08 | 0.64 | 0.71 | 0.00 | 0.25 | 0.49 | 0.04 |

4.3 4次元への拡張

これまで3次元の場合を述べてきたが、4次元へ拡張して評価を行う。

4次元における慣性半径(上位・下位)と市場組み合わせの表を以下に示す(表4.3)。3次元の場合と同様に4次元においても慣性半径の上位は地方市場、下位は都市市場の組が多い傾向にあった。

表 4.3 4次元における慣性半径(上位・下位)と市場組み合わせ

| 順位 | キャベツ | だいこん | トマト | なす | ねぎ |
|----------|----------------|--------------|--------------|--------------|--------------|
| 慣性半径大きい順 | 1 金沢_沖縄_札幌_高松 | 北九州_沖縄_札幌_高松 | 北九州_沖縄_札幌_高松 | 北九州_沖縄_札幌_高松 | 北九州_京都_沖縄_札幌 |
| | 2 北九州_沖縄_札幌_高松 | 福岡_沖縄_札幌_高松 | 金沢_沖縄_札幌_高松 | 金沢_沖縄_札幌_高松 | 京都_沖縄_札幌_仙台 |
| | 3 福岡_沖縄_札幌_高松 | 沖縄_札幌_仙台_高松 | 金沢_北九州_沖縄_札幌 | 沖縄_札幌_仙台_高松 | 福岡_京都_沖縄_札幌 |
| 慣性半径小さい順 | 1 神戸_名古屋_大阪_東京 | 名古屋_大阪_東京_横浜 | 名古屋_大阪_東京_横浜 | 北九州_沖縄_札幌_高松 | 金沢_名古屋_大阪_東京 |
| | 2 神戸_大阪_東京_横浜 | 神戸_名古屋_大阪_東京 | 京都_名古屋_大阪_横浜 | 金沢_沖縄_札幌_高松 | 広島_名古屋_大阪_東京 |
| | 3 神戸_京都_大阪_東京 | 神戸_名古屋_東京_横浜 | 神戸_名古屋_大阪_横浜 | 沖縄_札幌_仙台_高松 | 広島_金沢_名古屋_大阪 |

図 4.5 は、キャベツにおける4次元地方度と慣性半径の散布図である。横軸は4次元地方度、縦軸は慣性半径を表す。相関係数は0.83となり高い相関を示した。よって、4次元においても慣性半径は地方性を示していることが分かる。

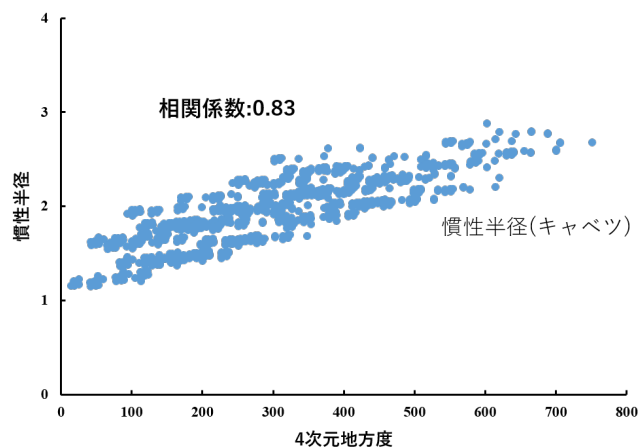


図 4.5 キャベツにおける4次元地方度と慣性半径の散布図

各品目に対する慣性半径と4次元地方度の相関係数の表を以下に示す(表4.4)。全ての品目において約0.7以上の相関係数をとっている。よって4次元においても慣性半径は地方度と相関係数が高いことが明らかとなった。

表4.4 各品目に対する慣性半径と4次元地方度の相関係数

| | キャベツ | だいこん | トマト | なす | ねぎ |
|------|------|------|------|------|------|
| 相関係数 | 0.83 | 0.81 | 0.91 | 0.84 | 0.69 |

以上より、慣性半径が地方性を敏感に検知できることが示唆された。また、慣性半径という統計量は、中心からの距離であるので多次元の場合でも幾何学的イメージがしやすいと考えられる。

第 5 章

結論

本研究では、多変量時系列データ解析を高次元空間上で行い、幾何的な統計量を算出することにより相関が弱く、等方的な時系列データの特徴量を抽出することを大目的とした。

特定時間における複数時系列データの任意の組み合わせを位置座標とすることで多変量時系列データの幾何的特徴を抽出できた。

本研究において、バネビーズモデルへの高分子アナロジーにより幾何的な新たな統計量を作成・評価し得られた結果は以下の 3 点である。

- 短期的変化の特徴を踏まえたボンド長平均： 地方性を検知するも、慣性半径に劣る。
- 中期的変化の特徴を踏まえた曲率平均： 地方性を検知できない。
- 長期的変化の特徴を踏まえた慣性半径： 地方性を敏感に検知できる。

等方性が高いデータに対して、ボンド長平均、慣性半径という統計量で地方性を類別でき、慣性半径が最も高い相関性を示したという結果から幾何学的特徴を抽出する方法としてボンド長平均、慣性半径は有効であると考えられる。

本手法は、異方性のあるなしに関わらず、高次元への拡張が容易な解析手法であり、本手法を用いることで様々な時系列データに対して幾何的特徴量を検出でき、慣性半径などの統計量で類別も可能と期待される。また、100 以上の時系列データによる同時解析など多変数データを扱う場合には、より高次元における幾何的特徴を検出することでデータの特徴を捉えることが容易になると考えられる。

参考文献

- [1] 主成分分析
https://www.macromill.com/service/data_analysis/principal-component-analysis.html
(最終検索日：2020年3月12日)
- [2] Kriz, Jan, and Petr Seba. "Force plate monitoring of human hemodynamics." *Nonlinear biomedical physics* 2.1 (2008): 1.
- [3] e-Stat 政府の統合窓口 青果物卸売市場調査
<https://www.e-stat.go.jp/stat-search/files?page=1&toukei=00500226&tstat=000001015623>
(最終検索日：2020年3月12日)
- [4] Web スクレイピング
<https://www.sejuku.net/blog/51241>
(最終検索日：2020年3月12日)
- [5] 46-inoguchi.pdf
- [6] bunshi04_2.pdf

付録

ここでは、使用したプログラミングコードを提示する。

付録 1 Python によるスクレイピングコード

e-Stat 政府の統合窓口 青果物卸売市場調査から野菜価格データの取得をするためのスクレイピングコードを以下に示す。

スクレイピングコード

```
1 from time import sleep
2 from selenium import webdriver
3 from selenium.webdriver.support.select import Select
4
5 #Chrome driver path
6 chromedriver = "C:/Users/furusawa_lab/Desktop/chromedriver"
7
8 #####
9
10 #URLselect
11 url="https://www.e-stat.go.jp/stat-search/files?page=1&layout=datalist&toukei
    =00500226&tstat=000001015623&cycle=7&tclass1=000001020455&tclass2
    =000001028870&cycle_facet=tclass1%3Acycle&second2=1"
12
13 #save dir
14 save_dir="C:/Users/furusawa_lab/Desktop/DL_estart/H19"
15
16 #start,end
17 start=10
18 end=205
19
20 #####
21
22 #download dir
23 chromeOptions = webdriver.ChromeOptions()
24 prefs = {"download.default_directory" :save_dir }
```

```
22 chromeOptions.add_experimental_option("prefs",prefs)
23 driver = webdriver.Chrome(executable_path=chromedriver, chrome_options=
    chromeOptions)
24 driver.get(url)
25 sleep(5)
26
27 a=driver.find_elements_by_class_name("stat-dl_text")
28
29 for i in range(start,end,2):
30     try:a[i].click()
31     except:
32         print("None")
33         continue
34     print(i)
35
36     sleep(5)
37
38 print("finish")
39 driver.quit()
```

付録 2 R による統計量算出コード

付録 2-1 3 市場抽出及び重心算出

3 市場の全組み合わせを抽出し、かつ重心を算出した R コードを以下に示す。

3 市場抽出及び重心算出コード

```
1 clall()
2 source("myfunction.R")
3
4 yasai_list=c("yasai_list")
5
6 dir_ori="dir_path"
7 setwd(dir_ori)
8 col_name_list=read.csv("col_name_list.csv",as.is=T,header=F )
9 comb_list=as.data.frame(combn(col_name_list[1,],m=3))
10
11
12 for(o in 1:length(yasai_list)){
13
14   yasai=yasai_list[o]
15   setwd(paste0(dir_ori,"/",yasai))
16
17   dir.create("new_dir")
18   setwd( paste0(dir_ori,"/",yasai,"/new_dir") )
19   dir.create("price")
20
21   print(yasai)
22
23   setwd( paste0(dir_ori,"/",yasai,"/data_file"))
24   dir=dir()
25   for(s in 1:length(dir)){
26
27     setwd( paste0(dir_ori,"/",yasai,"/data_file"))
28     print(dir[s])
29     x0=read.csv(dir[s],as.is=T)
30     x=x0
31     colnames(x)=colnames(x0)
32
33
34   for(comb_num in 1:ncol(comb_list)){
```

```

35
36     kotei_x=comb_list[1,comb_num][[1]]
37     kotei_y=comb_list[2,comb_num][[1]]
38     kotei_z=comb_list[3,comb_num][[1]]
39
40     kotei_x_num=which(colnames(x)==kotei_x)
41     kotei_y_num=which(colnames(x)==kotei_y)
42     kotei_z_num=which(colnames(x)==kotei_z)
43
44     result=NA
45     file_name=NA
46
47     result=x[,c(1,kotei_x_num,kotei_y_num,kotei_z_num)]
48     zyusin=as.data.frame(matrix(NA,nrow(result),3))
49     colnames(zyusin)=paste0("zyusin_",colnames(result)[2:4])
50     zyusin[1,1]=mean(result[,2],na.rm=T)
51     zyusin[1,2]=mean(result[,3],na.rm=T)
52     zyusin[1,3]=mean(result[,4],na.rm=T)
53     result=cbind(result,zyusin)
54
55
56     setwd(save_dir)
57     file_name=paste0( gsub(".csv","",dir[s]),"_",colnames(result)[2],"_",
58         colnames(result)[3],"_",colnames(result)[4],".csv" )
59     write.csv(result,file_name,row.names=F)
60     print(paste0(comb_num," : ",file_name))
61 }
62 }
63 }

```

付録 2-2 ボンド長平均算出

3次元におけるボンド長平均を算出した R コードを以下に示す。

ボンド長平均算出コード

```
1 clall()
2 source("myfunction.R")
3
4 dir_ori="dir_path"
5 setwd(dir_ori)
6 dir.create("new_dir")
7
8 yasai_list=c("yasai_list")
9 setwd( dir_path2 )
10 dir=dir()
11
12
13 for(dir_num in 1:length(dir)){
14
15   result=NA
16   setwd (dir_ori)
17   col_name_list=read.csv("col_name_list.csv",as.is=T,header=F )
18   comb_list=as.data.frame(combn(col_name_list[1,],m=3))
19   result=as.data.frame(matrix(NA,length(comb_list),length(yasai_list)+1))
20   colnames(result)=c("market_name",yasai_list)
21
22   for(i in 1:nrow(result)){
23     result[i,1]=paste0(combn_list[i][1,1][[1]], "_", comb_list[i][2,1][[1]], "_",
24       comb_list[i][3,1][[1]])
25   }
26   for(o in 1:length(yasai_list)){
27
28     setwd( dir_path3)
29     date_dir=dir()
30     print( getwd() )
31
32     for(i in 1:length(date_dir)){
33
34       setwd( dir_path3))
35       x=NA
36       bond=NA
```

```

37     x=read.csv(date_dir[i],as.is=T)
38     if( sum(colnames(x)ボンド長
39         =="" ) >0 ){x=x[,-which(colnames(x)ボンド長=="")]
40
41     bond=as.data.frame(matrix(NA,nrow(x),4))
42     for(s in 2:nrow(x)){
43         bond[s,1]=x[s,2]-x[(s-1),2]
44         bond[s,2]=x[s,3]-x[(s-1),3]
45         bond[s,3]=x[s,4]-x[(s-1),4]
46         bond[s,4]=sqrt( sum( bond[s,1:3]**2 ) )
47     }
48
49     result[ which( result[,1]==paste0(colnames(x)[2],"_",colnames(
50         x)[3],"_",colnames(x)[4]) ) , which( colnames(result)==
51         strsplit(date_dir[i],"-")[[1]][1] ) ] = mean(bond[,4],na.
52         rm=T)
53     if(i%%10==0){print(paste0(i," : ",date_dir[i]))}
54
55     x_col_name=colnames(x)
56     x=cbind(x,as.data.frame(bond[,4]))
57     colnames(x)=c(x_col_nameボンド長,"")
58
59     write.csv(x,date_dir[i],row.names=F)
60 }
61
62 result[,1]=paste0(paste(strsplit(date_dir[1],"-")[[1]][2:4],collapse="")
63     ,"_",result[,1])
64
65 setwd( paste0(dir_oriボンド長平均,"/") )
66 file_name=paste0( paste(strsplit(date_dir[o],"-")[[1]][2:4],collapse="") ,"
67     ボンド長平均次元
68     __3.csv" )
69 #write.csv(result,file_name,row.names=F)
70
71 print(file_name)
72 }

```

付録 2-3 曲率平均算出

3次元における曲率平均を算出した R コードを以下に示す。

曲率平均算出コード

```
1 clall()
2 source("myfunction.R")
3
4
5 dir_ori="dir_path"
6 setwd(dir_ori)
7 dir.create曲率("")
8
9 yasai_list=c("yasai_list")
10 setwd( dir_path2 )
11 dir=dir()
12
13
14 for(dir_num in 1:1){
15
16   result=NA
17   setwd (dir_ori)
18   col_name_list=read.csv("col_name_list.csv",as.is=T,header=F )
19   comb_list=as.data.frame(combn(col_name_list[1,],m=3))
20   result=as.data.frame(matrix(NA,length(comb_list),length(yasai_list)+1))
21   colnames(result)=c市場名("",yasai_list)
22
23   for(i in 1:nrow(result)){
24     result[i,1]=paste0(comb_list[i][1,1][[1]],"_",comb_list[i][2,1][[1]],"_",
25       comb_list[i][3,1][[1]])
26
27   for(o in 1:length(yasai_list)){
28
29     setwd(dir_path2)
30     date_dir=dir()
31     print( getwd() )
32
33     for(i in 1:length(date_dir)){
34
35       setwd( dir_path2)
36       x=NA
```

```

37     cos_list=NA
38     kyokuritu_list=NA
39
40     x=read.csv(date_dir[i],as.is=T)
41     if( sum(colnames(x)曲率=="") >0 ){x=x[,-which(colnames(x)曲率=="")]}
42
43     cos_list=mydf(nrow(x),2)
44     cos_list[,1]=x[,1]
45     colnames(cos_list)=c("Date","cos")
46
47     kyokuritu_list=mydf(nrow(x),2)
48     kyokuritu_list[,1]=x[,1]
49     colnames(kyokuritu_list)=c("Date曲率","")
50
51     cos_list_col_num=2:4
52
53     for(j in 2:(nrow(x)-1)){
54
55         p1=as.numeric(NULL)
56         p2=as.numeric(NULL)
57         dot=NA; cos=NA; phi=NA; kyokuritu=NA
58
59         p1=x[j+1,cos_list_col_num]-x[j,cos_list_col_num]
60         p2=x[j,cos_list_col_num]-x[j-1,cos_list_col_num]
61
62
63         dot=data.matrix(p1)%*%t(data.matrix(p2))
64         cos=dot/ ( sqrt( sum(p1**2) )*sqrt( sum(p2**2) ) )
65
66         cos_list[j,2]=cos
67
68         phi=acos(cos)
69         kyokuritu=2*tan(phi/2)
70         kyokuritu_list[j,2]=kyokuritu
71         #print(j)
72     }
73
74
75     result[ which( result[,1]==paste0(colnames(x)[2],"_",colnames(x)[3],"_"
76         ,colnames(x)[4]) ) , which( colnames(result)==strsplit(date_dir[i]
77         ],"_")[[1]][1] ) ] = mean(kyokuritu_list[,2],na.rm=T) 曲率の平均値
78     #
79     if(i%%10==0){print(paste0(i," : ",date_dir[i]))}
80

```



```

78
79     x_col_name=colnames(x)
80     x=cbind(x,kyokuritu_list[,2])
81     colnames(x)=c(x_col_name曲率,"")
82
83     write.csv(x,date_dir[i],row.names=F)
84 }
85
86
87
88 }
89
90 result[,1]=paste0(paste(strsplit(date_dir[1],"-")[[1]][2:4],collapse="")
91     ,"_",result[,1])
92
93 setwd( paste0(dir_ori曲率,"/"))
94 file_name=paste0( paste(strsplit(date_dir[o],"-")[[1]][2:4],collapse=""),"
95     曲率平均次元
96     __3.csv")
97 #write.csv(result,file_name,row.names=T)
98
99 print(file_name)
100 }

```

付録 2-4 慣性半径算出

3次元における慣性半径を算出した R コードを以下に示す。

慣性半径算出コード

```
1 clall()
2 source("myfunction.R")
3 dir_ori="dir_path"
4 setwd(dir_ori)
5 dir.create慣性半径("")
6
7 yasai_list=c("yasai_list")
8 setwd( dir_path2)
9 dir=dir()
10 for(dir_num in 1:1){
11
12     result=NA
13     setwd (dir_ori)
14     col_name_list=read.csv("col_name_list.csv",as.is=T,header=F )
15     comb_list=as.data.frame(combn(col_name_list[1,],m=3))
16     result=as.data.frame(matrix(NA,length(comb_list),length(yasai_list)
17         +1))
18     colnames(result)=c市場名 ("",yasai_list)
19
20     for(i in 1:nrow(result)){
21         result[i,1]=paste0(comb_list[i][1,1][[1]], "_", comb_list[i]
22             ][2,1][[1]], "_", comb_list[i][3,1][[1]])
23     }
24
25     for(o in 1:length(yasai_list)){
26
27         setwd( dir_path2)
28         date_dir=dir()
29         print( getwd() )
30
31         for(i in 1:length(date_dir)){
32             x=NA
33             s=NA
34             s_2=NA
35             x=read.csv(date_dir[i],as.is=T)
36             s=as.data.frame(matrix(NA,nrow(x),3)) #s: 慣性半径
```

```

36         for(w in 1:nrow(x)){
37             s[w,1]=x[w,2]-x[1,5]
38             s[w,2]=x[w,3]-x[1,6]
39             s[w,3]=x[w,4]-x[1,7]
40
41         }
42
43         s_2=as.data.frame(matrix(NA,nrow(s),1))
44         for(w in 1:nrow(s)){
45             s_2[w,1]=sum(s[w,1:3]**2)
46         }
47
48         result[ which( result[,1]==paste0(colnames(x)[2],"_"
49             ",colnames(x)[3],"_" ,colnames(x)[4]) ) , which
50             (
51                 colnames(result)==strsplit(date_dir[i],"_")
52                 [[1]][1] ) ] = mean(s_2[,1],na.rm=T)
53         if(i%%10==0){print(paste0(i," : ",date_dir[i]))}
54     }
55 }
56
57 result[,1]=paste0(paste(strsplit(date_dir[1],"_")[[1]][2:4],
58     collapse=""),"_",result[,1])
59
60 setwd( paste0(dir_ori慣性半径,"/") )
61 file_name=paste0( paste(strsplit(date_dir[o],"_")[[1]][2:4],
62     collapse=""),"慣性半径次元
63     __3.csv")
64 #write.csv(result,file_name,row.names=F)
65
66 print(file_name)
67 }

```

付録 2-5 2乗平均末端分子間距離

3次元における2乗平均末端分子間距離を算出したRコードを以下に示す。

2乗平均末端分子間距離算出コード

```
1 clall()
2 source("myfunction.R")
3
4
5 dir_ori="dir_path"
6 setwd(dir_ori)
7
8   setwd(dir_ori)
9   #dir.create乗平均平方根("2次元_3")
10  #setwd( dir_path2)
11  #dir.create(dir[dir_num])
12
13
14  setwd( paste0(dir_ori,"/data_file") )
15  data_dir=dir()
16  print( getwd() )
17
18
19  for(i in 1:length(data_dir)){
20
21    setwd( paste0(dir_ori,"/data_file") )
22    x=NA
23    result=NA
24    x=read.csv(data_dir[i],as.is=T)
25
26    result=mydf(nrow(x)-1 , 4) ;result[,1]=1:nrow(result) ;colnames(result
      )=c("tau乗","2meansqrt","log_tau","乗
      log_2meansqrt")
27
28    for(tau in 2:( nrow(x)-1 ) ){
29
30      list=1:(nrow(x)-tau)
31      r=0
32      r=mydf(length(list),2);colnames(r)=c("term","乗 R2")
33
34      for(j in list ){
35        r[j,2]=sum( ( x[(j+tau),2:4]-x[j,2:4] )**2 )
```

```

36     r[j,1]=paste0(j,"-",j+tau)
37     #print(paste0(j,"-",j+tau))
38 }
39 #print(paste0("tau:",tau))
40 #print( paste0平均(":",round(result[tau,2],2)) )
41 #Sys.sleep(0.3)
42 if(tau%%100==0){print( paste0("tau: ",tau) )}
43 }
44
45 result[,3]=log10(result[,1])  常用対数を用いる#
46 result[,4]=log10(result[,2])
47 result=na.omit(result)
48
49 setwd( paste0(dir_ori乗平均平方根,"/2次元_3_data") )
50 file_name=gsub(".csv","乗平均平方根次元_2_3.csv",data_dir[i])
51 print(file_name)
52 write.csv(result,file_name,row.names=F)
53 }

```

謝辞

本論文を結ぶにあたり、多くの方にお世話になりました。研究を進めるうえで、指導教員の古沢浩教授（高知工科大学大学院基盤工学研究科）からは、研究内容のみならず、研究の進め方から論文の書き方に至るまで、数限りない助言を頂きました。心より感謝申し上げます。

本研究を遂行する上で、プログラミングの方法、データ解析方法、ネットワークの分析方法、疑問点等に関して、適切で有意義な助言とアドバイスを与えて頂いたり、研究室での学生生活においてお世話になりました古沢研究室の皆様感謝いたします。

はなはだ簡単ではありますが、以上をもって謝辞とさせていただきます。