AIによる数法則発見の時系列データへの拡張と金融データへの応用

Modeling and Visualization of Social Reality
Using Latent Profile Analysis and Number Law Discovery Methods
for Evidence-Based Policy Making

蒲田 涼馬 (Ryoma Gamada)
u455007@st.pu-toyama.ac.jp

富山県立大学大学院 工学研究科 電子・情報工学専攻
情報基盤工学講座

N212, 09:30-10:00 Tuesday, February 13, 2024.

本研究の背景 1

情報技術の発達により，社会における様々なデータを観測・収集することが可能に

→ 経済分析においても将来予測などの研究が急速に発展.
しかし要因分析に関する研究はそれほど進んでいるとは言えない.

経済に影響を与える要因を分析する研究

因果探索による要因分析.
シンボリック回帰を用いた要因分析.

本研究

シンボリック回帰を用いて分析を行う.

本研究の背景 2

なぜシンボリック回帰？

経済分野では，原因と結果の間に成り立つ関係性が重要

→ 複数の要因が複雑に影響しあうため，因果探索では具体的にどのように絡み合って影響を与えるかを詳細に分析できない．

アプローチ

公開されている金融データ，経済データ，市場間データを用いて分析を行い，データ間の関係性を数理モデルによって表す．

数理モデルの例

$$(\text{データ A}) = 2.0 \cdot (\text{データ B}) + 1.0 \cdot (\text{データ C}) - 1.0 \cdot (\text{データ D})$$

経済データと市場データ

公開されているデータ

XMMT5 や Investing.com，日本銀行時系列サイトで様々なデータが公開されている．

Table 1: 公開されている様々な経済データ

データ項目	
為替レート	金利
コモディティ価格	エネルギー価格
マネーストック	ボラティリティ指数
出来高	スプレッド
株価指数	ニュース

目的

時系列経済データを用いて，時系列を考慮した数法則の発見を行いデータ間の関係性をモデル化する手法を提案する．

手法の概要

使用する手法の概要

機械学習を用いたシンボリック回帰手法である「End to end symbolic regression with Transformers」を拡張させ，時系列を考慮した分析を行う．可読性と得られる情報量を重視し，人間が式を見ることである程度その式が何を表しているのかをわかるレベルのものを生成させる．

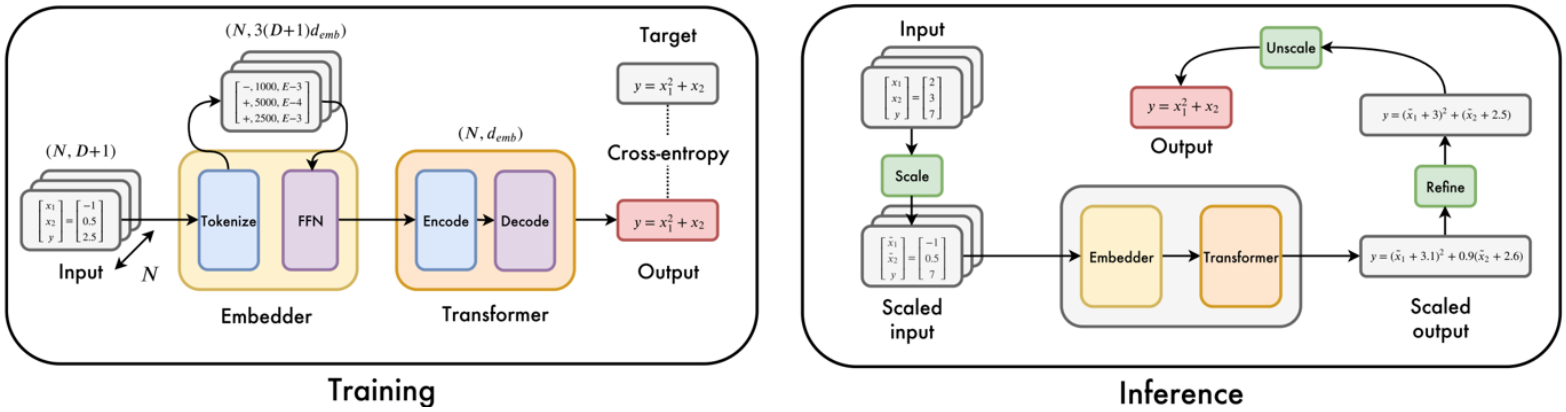
End to end Symbolic Regression with Transformers の概要

データから数式を自動発見する深層学習アプローチ

従来のシンボリック回帰の課題: 計算コストが非常に高い.
Transformer アプローチの着想:
数式は，演算子，定数，変数といった要素が並んだシーケンスとして表現する.
例) $y = x + 2 \cdot sin(z) \rightarrow$ + x * 2 sin z Transformer はシーケンスデータの複雑なパターン学習と高速なシーケンス生成能力を持つため，数式発見に応用できる.

Transformer によるシンボリック回帰 (前準備)

Transformer によるシンボリック回帰



Embedder

入力データをトークン化する.
埋め込みルックアップテーブルを使ってトークンをベクトルに変換する (512 次元).
ベクトルを FFN に入力し，短いベクトルに圧縮する.
この処理をすべてのデータ点に対して行う.
これを Transformer 本体に渡す.

Transformer のメカニズム

Transformer はエンコーダーとデコーダーから成り，Attention メカニズムが核を担っている．

- **Attention メカニズム**
  シーケンス内の各要素が，ほかのどの要素に注意を向けるべきかを動的に判断し，その重要度について重みづけを行う．

$$Scores = QK^T \tag{1}$$

$$ScaledScore = \frac{QK^T}{\sqrt{d_k}} \tag{2}$$

$$AttentionWeight = softmax(\frac{QK^T}{\sqrt{d_k}}) \tag{3}$$

$$Attention = softmax(\frac{QK^T}{\sqrt{d_k}})V \tag{4}$$

ここで Q はクエリ行列，K はキー行列，V は値，d_k はキーの次元を意味する．

- **Feed-Forward Network**
  ベクトルを圧縮する際やベクトルを拡張する際に使われる.
  Transformer モデルでは両方の使われ方がされている.

$$FFN(x) = max(0, xW_1 + b_1)W_2 + b_2 \tag{5}$$

ここで x は前のステップからの入力，W_1, W_2, b_1, b_2 はモデルが学習する重みとバイアスを意味する.

エンコーダー

入力されたデータ点を読み込み，それらがどんな関数の特徴を持っているかを分析する.

Transformer によるシンボリック回帰

エンコーダー

Embedder によってベクトル化された N 個のデータ点を受け取る.
関連性の計算 (self-attention) を行い，各データ点がほかのすべてのデータ点とどの程度関連しているかを計算する.
FFN によって関連性を加味した情報をさらに深く処理する.
関連性の計算と FNN による処理を 4 層繰り返す.
入力データ全体の特徴を要約した情報をデコーダーに渡す.

エンコーダーの self-attention

入力された全データ点の，相互の関連性の強さを計算する.
自分自身を含むすべての入力データ点を参照する.
これによって各データ点のベクトルにデータセット全体における他の点との関係性を埋め込む.

デコーダー

エンコーダーからの情報と数式の一部を受け取る (初回は開始トークンのみを受け取る).
自己参照 (初回は開始トークンのみ，2 回目以降はトークン列を受け取る)
エンコーダーからの要約情報に注目し，作る関数を理解する.
上記 2 角情報をもとに，数式の次に来るべきトークンを予測する.
自己参照から予測までの流れを終了トークンが出力されるまで 1 トークンずつ繰り返す.
完成した数式を出力する.

デコーダーの自己参照

$$Attention(Q, K, V) = softmax(\frac{QK^T}{\sqrt{d_k}} + M)V \tag{6}$$

注目しているトークンとほかの全トークンとの関連度を計算する.
マスクを利用して未来のトークンに関する部分を $-\infty$，それ以外の部分を 0 にする.
これに softmax 関数を適用することでモデルが過去と現在のトークンのみを参照して次のトークンを予測するようにする.

クロスエントロピーによる数式の評価

$$H(p,q) = -\sum_i p(x_i) \log q(x_i) \tag{7}$$

ここで $p(x_i)$ は正解の確率分布，$q(x_i)$ はモデルが予測した確率分布を意味する．
これを計算することで生成した数式の評価を行い，これを高めていくように学習していってモデルを作る．

クロスエントロピーによる数式の評価

$$H(p,q) = -\sum_i p(x_i) \log q(x_i) \tag{8}$$

ここで $p(x_i)$ は正解の確率分布，$q(x_i)$ はモデルが予測した確率分布を意味する．
これを計算することで生成した数式の評価を行い，これを高めていくように学習していってモデルを作る．

今回やったこと

今回やったこと

時系列クラスタリング手法の検討および実装.
時系列クラスタリング手法の実験.
時系列クラスタリングで求めたレジームごとにシンボリック回帰を動かす.

モデルベースの時系列クラスタリング

各区間からその付きの特徴を表すベクトルを線形回帰モデルを用いて抽出し，それを k-means を用いてクラスタリングする．
t における目的変数を y_t，説明変数を D 個のベクトル
$X_t = [x_{t,1}, x_{t,2}, \cdots x_{t,D}]$ とし，以下の式で表される．

$$y_t = \beta_{m,0} + \beta_{m,1} x_t + \beta_{m,2} x_t, 2 + \cdots \beta_{m,D} x_{t,D} + \epsilon_t \tag{9}$$

ここで $t \in m$ であり，

数値実験の概要 1

使用したデータ

項目：10 種類の経済データ
対象年：2015 年 1 月 1 日から 2024 年 12 月 31 日までの 10 年間
目的変数：USDJPY の t のときの値
説明変数：Table2 の t-1 のときの値

Table 2: 数値実験に用いたデータ

データ項目	
SP500 価格	日経平均株価
日米の金利差 10 年	日本 10 年国債
米国 10 年国債	日米の金利差 2 年
オイル価格	金価格
USDJPY	VIX 指数

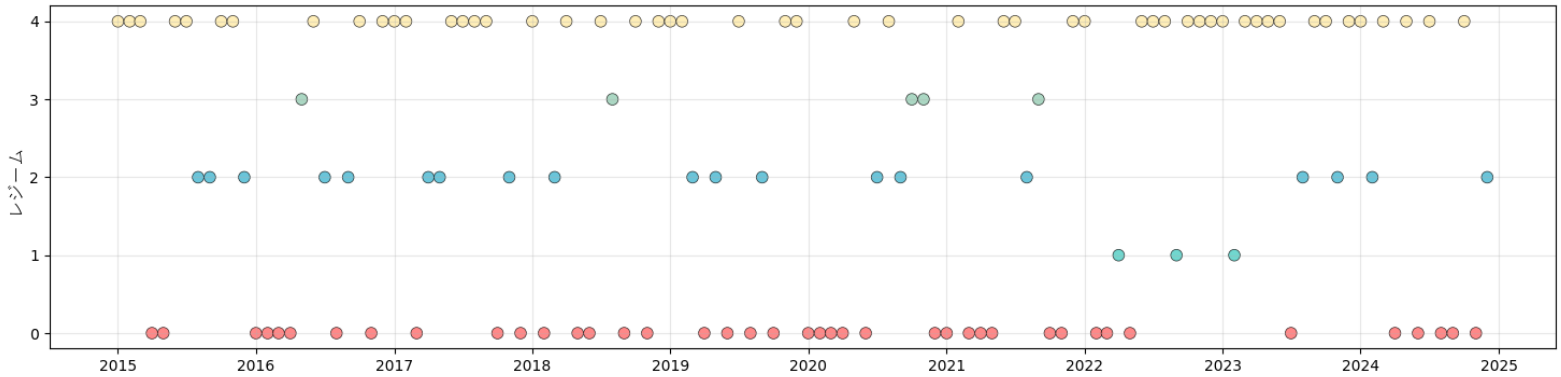
実験の結果 1

時系列クラスタリングの結果

時系列クラスタリングの結果を以下に示す.
今回は月ごとにデータを区分し，クラスター数 5 個にわけることとした.

時系列クラスタリングの結果

結果は以下のようになり，クラスター 0 とクラスター 2，クラスター 4 に偏っており，クラスター 1, 3 がかなり少ない.
この結果についての考察はまだできていないので，各区間ごとにどのような状況，形状をとっているのかはまだ不明



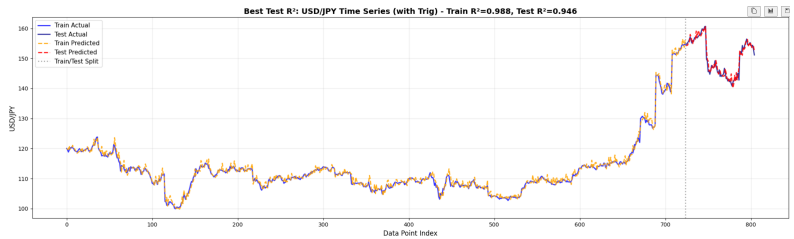
実験の結果 2

時系列シンボリック回帰の結果

次にクラスタごとに分けてシンボリック回帰を行った結果を以下に示す.
今回はデータが少ないクラスタとデータが多いクラスタをひとつづつ示す.
まずはクラスタ 0 の結果を示す.

決定係数

- test データについての決定係数 : 0.9456
- test データについての RMSE: 1.3777



$$0.999\text{‘}USD/JPY_{lag1norm filtered}\text{’} - 0.069 \arctan(0.894 - 0.695 \tan(7.528\text{‘}US_{10Yyieldnorm filtered}\text{’} - (6.184\text{‘}USD/JPY_{lag1norm filtered}\text{’} + 8.842)(1.285\text{‘}JP_{10Yyieldnorm filtered}\text{’} - 0.66\text{‘金利差}_{JP-US10Ynorm filtered}\text{’} - 97.932) + 1.605)) + 0.051$$

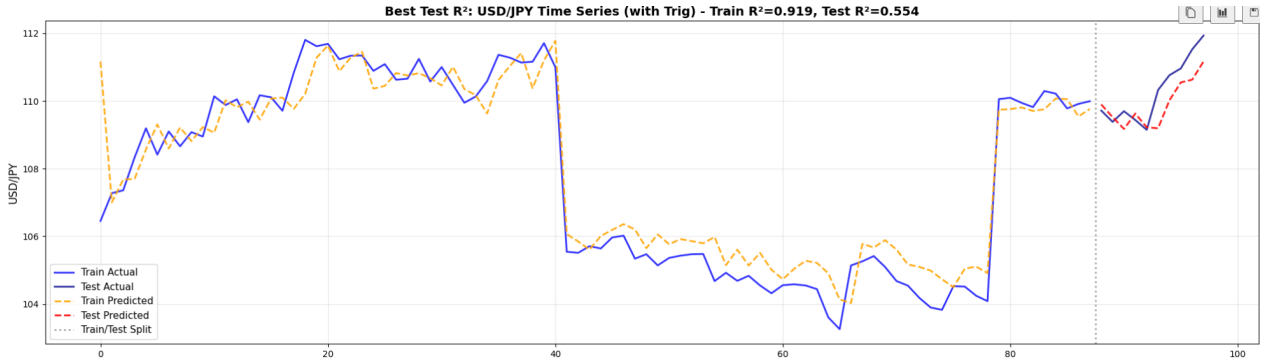
実験の結果 3

時系列シンボリック回帰の結果

次にクラスタ 2 の結果を示す.
データがかなり少ないということもあって，かなり精度が悪い.
数式は空きの関係で省略するが明らかに冗長で可読性に優れていない.

決定係数

- test データについての決定係数 : 0.5151
- test データについての RMSE: 0.6401



考察

考察

データ数が少ないクラスタに関しては当然ながらかなり精度が低くなってしまう.
三角関数の出現頻度も特に下がらず，ただそれぞれのクラスタで説明変数の登場頻度が異なる可能性がある.
またデータ数が多いクラスタでは可読性がそれなりに向上していた.
$\rightarrow$ データ数が少ないクラスタはどれにも属さないものが適当に放り込まれている可能性がある.

おわりに

まとめ

今回は時系列クラスタリング手法について色々調査し，今回の実験に適用できそうなものを適用してみた．

- 時系列クラスタリングではクラスタごとにデータ数の偏りがある．
- 変数の絞り込み結果もクラスタによって異なる．
  → 非定常性をある程度考慮できている？
- データ数が多い部分ではある程度の可読性の向上が見られた．

今後の展望

- ほかの時系列手法についても様々調査し，実装する．
- 実装ができたら今回と同じ条件で実験を行いどの手法を使うのかを決める．
- 可読性の向上についての手法を調査し，実装する．