

放電加工機の異常検知(その2)

1. はじめに

前回、近傍法による異常部位検出 と 特異スペクトル変異法 で、異常検知を試みました。

今回は、(その2)として、自己回帰(AR)モデル と Change Finder で、異常検知を試みます。

自己回帰(AR)モデルによる異常検知は、

「入門 機械学習による異常検知—Rによる実践ガイド—」井出剛 著 コロナ社 #1

で、前回の手法と並べて紹介されています。また、Rのプログラム構文は この本を、参照してます。

Change Finder は

「データマイニングによる異常検知」山西健司 共立出版

に 紹介されており、参考にしました。(この本は、#1 に 紹介されており、知りました)

Rのプログラム構文は、以下を、参照しています。

ビューコミュニケーションズ「異常値検知のための基本的モデルの考察」

結果を述べますが、(3.3 項 参照)

自己回帰(AR)モデルは、#1 に “突発的に大きな値をとるデータにはうまくいかない” と記載されていますが、やはり、放電加工機の異常検知には不向きようです。

Change Finder は、放電加工機が異常と判定したタイミングを、全て、検出していました。

エラーメッセージの重みを少し変えているのですが、不思議な気もします。(私が、Change Finder のアルゴリズムを十分に理解していないのかも知れませんが)

異常検知について思うのですが、不具合が発生したら、膨大な損失を受ける通信キャリア とか エネルギー関連の事業者とかは、相当の対応をしてるはずですが。ただ、開示すると、攻撃者に手の内を見せることになるのか、WEB 検索しても、なかなか分かりません。技術部においては、基本の(古典的)手法を、学びつつ、進めるしかないようです。

2. 時系列データの異常検知

異常検知の方法として、今回、

- ・自己回帰(AR)モデルによる異常検知
- ・Change Finder による異常検知 の 二つで試みてみました。

2.1 考え方

2.1.1 自己回帰(AR)モデルによる異常検知

t 時刻の予測値を、過去の観測値から予測し、t 時刻の実測値と予測値を比較して、異常度を算出し、異常を検知する方法です。

予測値を $\xi(t)$ として、今より前の 3 個の値の 1 次結合と考えます。 (少ない個数は説明のため)

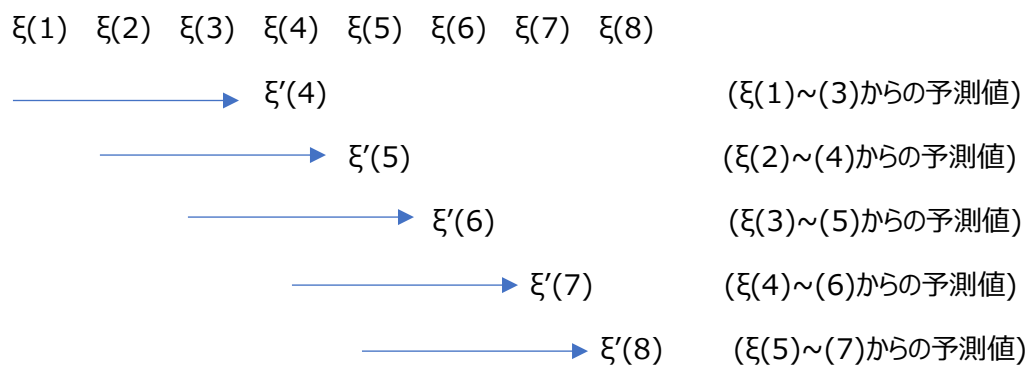
$$\xi(t) = a_1\xi(t-3) + a_2\xi(t-2) + a_3\xi(t-1) \text{ になります。}$$

この個数を次数(order)、 a_1 から a_3 の値を係数と呼びます。(合わせてパラメータと呼びます)

最適な次数、係数値、また、平均値、分散値は R の ar 関数により、求めることができます。

従って、訓練用データで各値を算出しておき、検証するデータに適用し、異常を検知します。

例ですが、検証するデータで、総数が 8 個とあったとします。(少ない個数は説明のため)



$\xi(4)$ と $\xi'(4)$ 、 $\xi(5)$ と $\xi'(5)$ 、 $\xi(6)$ と $\xi'(6)$ 、 $\xi(7)$ と $\xi'(7)$ 、 $\xi(8)$ と $\xi'(8)$ の比較により、異常度を算出します。

以下は ご参考です。

AR(Autoregressive)モデルは MA(Moving Average)モデルとともに時系列モデルの基礎で、

ついで、AR と MA を組み合わせた ARMA モデル、

更に ARIMA モデル (Autoregressive Integrated Moving Average) があるようです。

2.1.2 Change Finder による異常検知

Change Finder は、ネットワークのシステム障害検出等に活用されてるようで、AR モデルがベースです。ただ、AR モデルは、時系列が定常であるという仮定があるので、非定常の場合にも適用できるよう工夫されてます。AR モデルでのパラメータに加えて、忘却パラメータが追加され、過去のデータの重みを減らすことで、非定常なデータに対応します。

(SDAR アルゴリズムと呼んでます 9 ページ目に簡単な説明を追加しました)

手順は 第一段階学習(異常値検知) → 平滑化 → 第二段階学習(変化点検知)

と進みます。2 段階の学習の間で、平滑化することで、異常値の影響を緩和しようとしています。

第一段階学習(異常値検知): 入力された時系列データから AR モデルを定義し、SDAR アルゴリズムでパラメータの学習を行い、時系列データの 異常値スコアを算出します。

平滑化 : 第一段階学習により求めた異常値スコアに対し、一定時間 T (平滑パラメータ) ごとに移動平均による平滑化処理を行います。

第二段階学習(変化点検知): 異常値スコアを平滑化したものに対し、再度 SDAR アルゴリズムにより算出したスコアを変化点スコアとする。これにより、異常値の影響を低減します。

(以上を数式で解説できれば良いのですが、私には荷が重いです。)

パラメータについて、繰り返します。AR モデルによる異常検知は、関数により算出しましたが、Change Finder は、入力が必要です。

r : 忘却パラメータ (今回の例では、第一段階 0.02 第二段階 0.002 を入力してます)

確率密度関数を算出する際、どの程度過去の影響をコントロールするかを示しています。

この値を小さくすると過去の影響が大きくなり、変化点のバラツキが大きくなります。

$order$: AR モデルの次数 (今回の例では、第一段階 3 第二段階 2 を入力してます)

どこまで過去の値をモデルに組み込むかを設定します。

Smooth (窓) : 平滑化の範囲 (今回の例では、第一段階 3 第二段階 5 を入力してます)

ここを長くすれば長くするほど異常値ではなく「変化」が捉えられますが、大きくしすぎるとそもそも変化そのものが捉えづらくなります。

3.2 前処理されたデータ (=元データ) の内容

(技術報告書 “放電加工機の異常検知”と同じです 除く 下の3行部)

今回は 4月1日から6月20日の分について、データを作成しました。6317個あります。

実データから下記の元データを作成するのに、手動では面倒ですので、マクロを活用してます。

Point とは メッセージに対応する点数で、障害の程度で付与してます。

エラーメッセージ 3~4 ハルトメッセージ 2~2.5 コメントメッセージ 0.5~1.5 としました。

同じメッセージ内でも、数値を加減してます。

以下の元データを作成します。(エクセルからCSV(コンマ区切り)に変換します)

when	point
2022/04/01 08:28:01	0.5
↓	
2022/06/20 08:35:57	1
2022/06/20 08:36:10	1

元データから、異常度を評価します。

計算は R 言語 で行いました。R 言語とは、

「オープンソース・フリーソフトウェアの統計解析向けのプログラミング言語」とのことです。

Python の機械学習もありますが、

参考書が R による実践ガイド だったこともあり、R で検証してます。

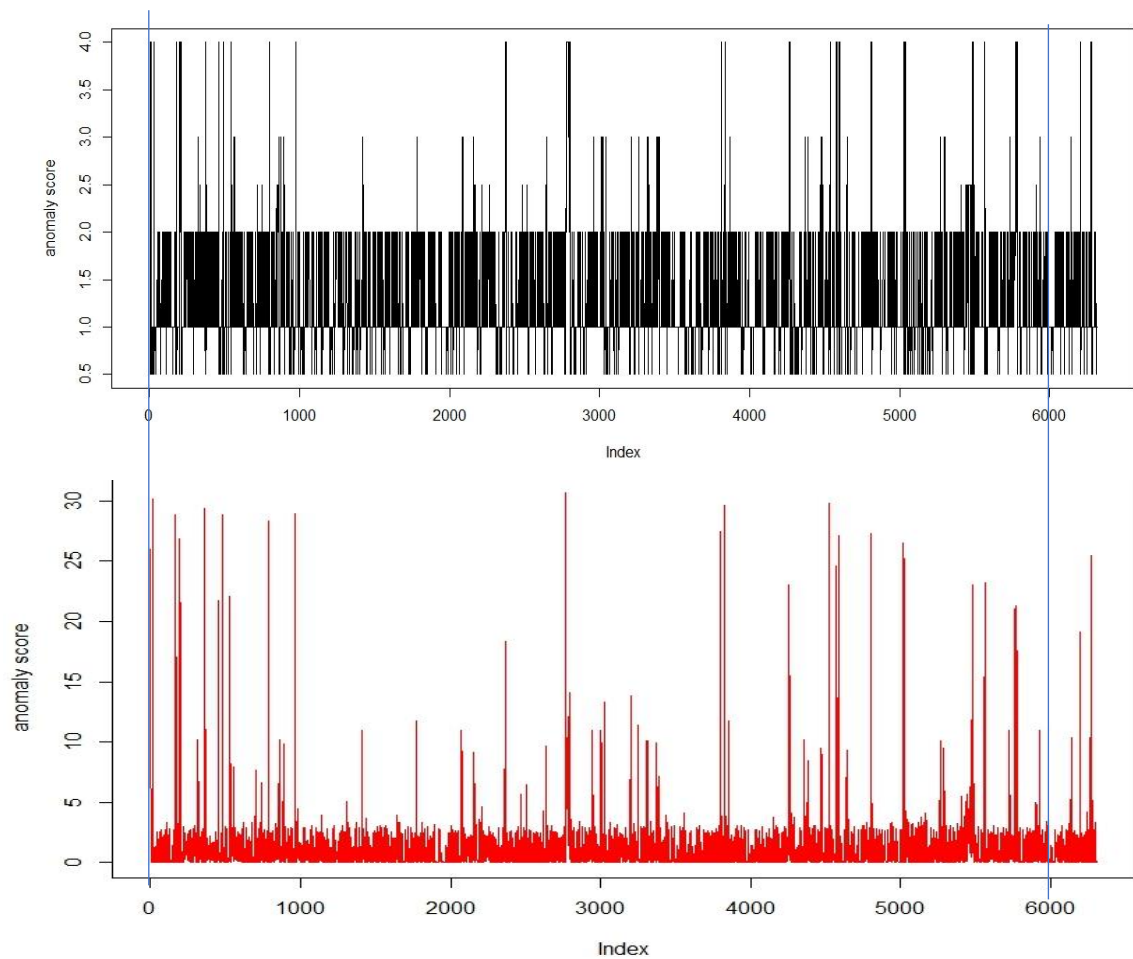
元データを Rstudio (R の開発環境) に取込んで、

- ・ 自己回帰(AR)モデルによる異常検知 と ・Change Finder による異常検知の 2 種について、計算して、プロットしました。

3.3 結果のプロット

3.3.1 自己回帰(AR)モデルによる異常検知

元データは黒のグラフです。6317 個のデータに対して、0.5 から 4.0 のポイントが付与されています。

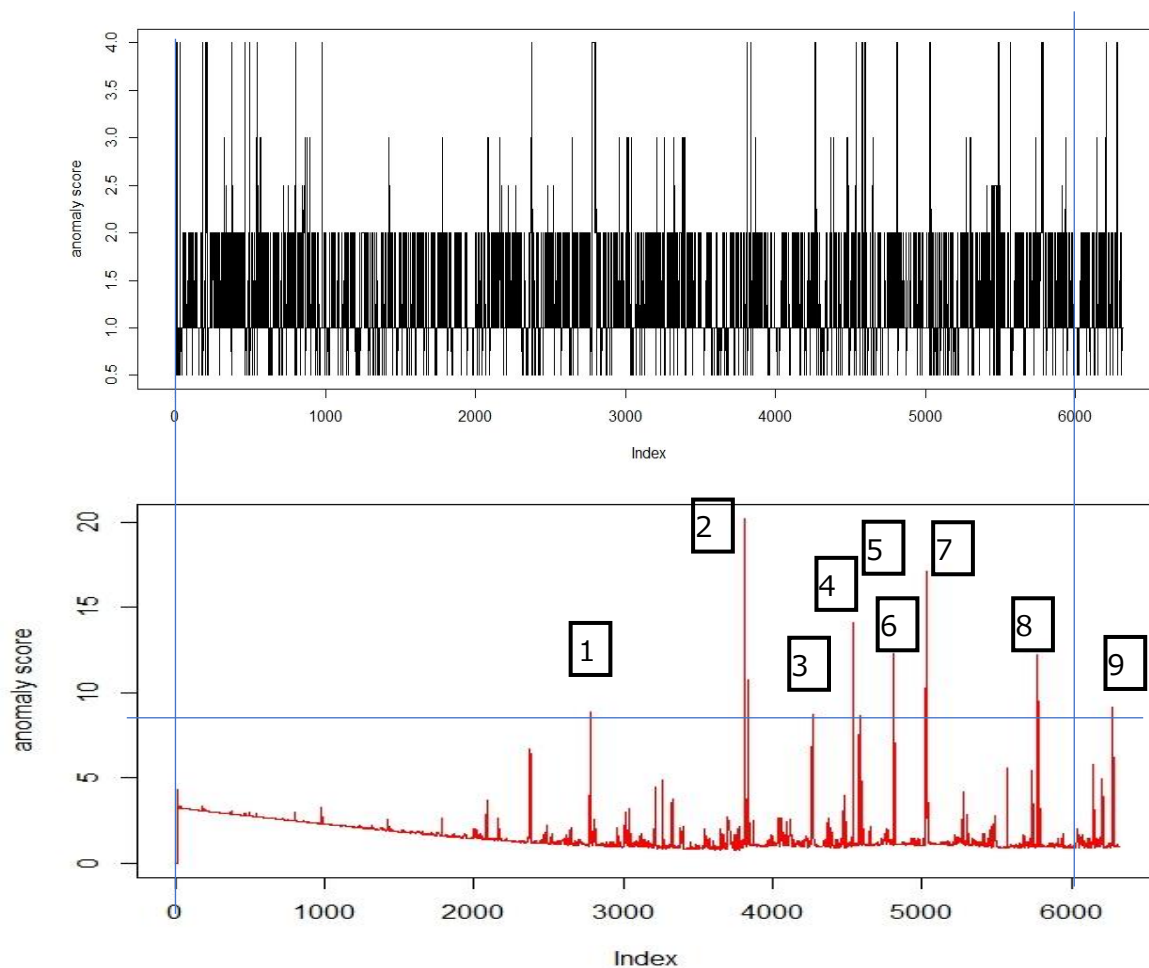


異常度については、検出するとは言えないようです。

階段状に変化するデータを、自己回帰でモデル化するのは無理があるようです。

3.3.2 Change Finder による異常検知

黒の元データは 3.3.1 と同じです。異常度のグラフは赤のグラフです



異常度 8 以上に注目して、メッセージを確認しました。

- ① は [H00207: () テンション制御装置が異常です。ENT スイッチを押して下さい。]が多発
- ② は [E00335: (X)モータ過負荷保護が働きました。]が発生
- ③ は [E00245: () 機械座標が確定していないため実行できません。]
- ④ は [E00175: (G00) パワーを入れてください。]が発生
- ⑤ ⑥ ⑦ は [E00175: (G01) パワーを入れてください。]が発生
- ⑧ は [H00207: () テンション制御装置が異常です。ENT スイッチを押して下さい。]が多発
- ⑨ は [E00350: (V) フィードバック速度異常です。]が発生

3.3.3 プロットへの評価

自己回帰(AR)モデルによる異常度の結果は、
元データと同じようなプロットで、異常検知には、不適當のようです。

「入門 機械学習による異常検知—R による実践ガイド—」にも以下の記述がありました。
自己回帰モデルは、
“心電図データのように、突発的に大きな値をとるデータにはうまくいかないことが知られています。
そのようなデータについては、適切な前処理をして変化を「なます」ことが通例です。”

Change Finder による異常度の結果については、

高い異常度スコアで、エラーメッセージ E00175 の「パワーをいれてください」を検出しています。

((4)、(5)、(6)、(7) の項目です)

これは、何かしらの異常があり、電源を自動 OFF した後に、出されるメッセージです。

異常度の検出としては、適切なもので、先回の報告書で説明した異常検知では、検出できてない
項目です。また、異常度スコアを下げると、検出内容は、先回のものと同じ内容となっています。

3.3.4 今後の取組

継続して、データ採取して、どれくらい役に立つのか検証していきます。

SDAR アルゴリズムの紹介

まずは AR モデルについての説明です。

時系列変数 Z_t は、 d 次元ベクトルとして、 k 次のARモデルとして以下で表します。

$$z_t = \sum_{i=1}^k w_i z_{t-i} + \varepsilon$$

w_i は d 次のパラメータ行列で、自己回帰係数です。

k は 自己回帰次数 です。 ε は平均 0、正規分布する白色雑音 です。

実際に観測される時系列を

$$x_t = z_t + \mu \quad \text{とします。} (\mu \text{は平均値})$$

X_t の確率密度関数は、以下で表します。

$$P(x_t | x_{t-k}^{t-1} : \theta) = 1 / (2\pi)^{d/2} |\Sigma|^{1/2} * \exp(-1/2(x_t - w)^t \Sigma^{-1} (x_t - w))$$

$$w = \sum_{i=1}^k w_i (x_{t-i} - \mu) + \mu \quad \text{です。}$$

この式より、 x_t を計算するのですが、 w_i (自己回帰係数) と Σ (分散共分散関数) が分かってないので、算出できません。この算出は、最尤法を用います。(この紹介は別途にします)

ここから SDAR アルゴリズム 忘却パラメータの説明です。

忘却パラメータは r とし、 $0 < r < 1$ です。

μ (平均値) と C_j (自己共分散関数) について、現在の値と新しい値を

$(1-r) : r$ の比で 重み付けをして更新し、上記の計算を行います。

$$\mu = (1-r) \mu + r X_t$$

$$C_j = (1-r) C_j + r(x_t - \mu)(x_t - j - \mu)^t$$

r が小さいほど、過去のデータの影響が大きくなります。

忘却効果で、非定常なモデルの学習をしていると言われています。