

Tomo-e Gozen 広域サーベイで 検出された人工天体の ライトカーブのクラスター分析

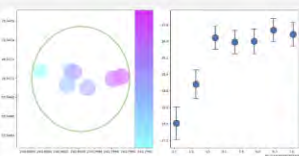
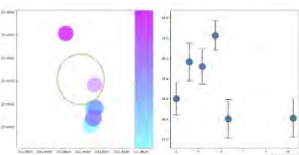
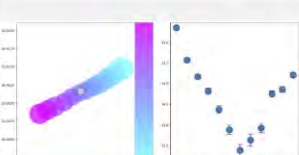
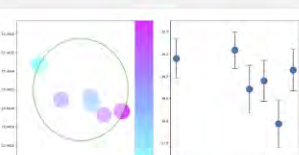
東京大学理学部天文学科 4年生 大木亮吾
酒向重行、瀧田怜（東京大学）

Tomo-e Gozen全天サーベイと人工天体

- Tomo-e Gozen は毎日広域サーベイデータを取得、1日あたり数千件の高速移動天体を検出
そのうち地球接近小惑星(NEO)などの天文学的天体は数例のみ
ほとんどはスペースデブリなどの**人工天体**
- 観測される未知の人工天体について位置、速度を用いて軌道を決定し、カタログ化
 - NEO探査の効率化
近宇宙の軌道資源の安全化
- 軌道決定のために、異なる時刻に観測された人工天体データを**マッチング**したい

Tomo-e Gozen全天サーベイによる取得データ

- Tomoe-Gozen により検出された高速移動天体データは以下のようになっている。

NO.	OBSTIME OBJECT FITS	MATCHING	IMAGE	LIGHT CURVE	SCORE	SN	RA, DEC	V_MEAN ["/sec]	V_PA [deg]	V_RA ["/sec]	V_DEC ["/sec]
13	2024-04-28T19:08:02.658 24Dcqdea_16053+19547 TMQ4202404280119730044.fits NEO(radius:30.0arcmin, num:1) Tracklet Search Circle fit	All Satellite NEO NEOCP			0.45 FN Add ml flag	7.06	16h03m12.0111s +19d32m49.3701s	1.09	266.21	-1.09	-0.07
11	2024-04-28T19:08:55.643 24Dcqded_16124+25891 TMQ1202404280119730325.fits Tracklet Search Circle fit	All Satellite NEO NEOCP			0.67 FN Add ml flag	6.31	16h07m27.4109s +25d53m30.3159s	1.66	16.19	0.46	1.6
9	2024-04-28T19:09:22.716 24Dcqdee_16254+16811 TMQ1202404280119730423.fits SAT(radius:30.0arcmin, num:1) Tracklet Search Circle fit	All Satellite NEO NEOCP			0.98 FN Add ml flag	16.71	16h15m17.1125s +16d48m41.6628s	8.91	117.14	7.93	-4.07
10	2024-04-28T19:09:22.005 24Dcqdee_16328+15983 TMQ1202404280119730412.fits Tracklet Search Circle fit	All Satellite NEO NEOCP			0.23 FN Add ml flag	5.79	16h19m44.3318s +15d58m59.2363s	0.96	241.9	-0.85	-0.45

Tomoe NEO Viewer の画面キャプチャ

- 検出された天体の座標や速度、ライトカーブデータが格納されている

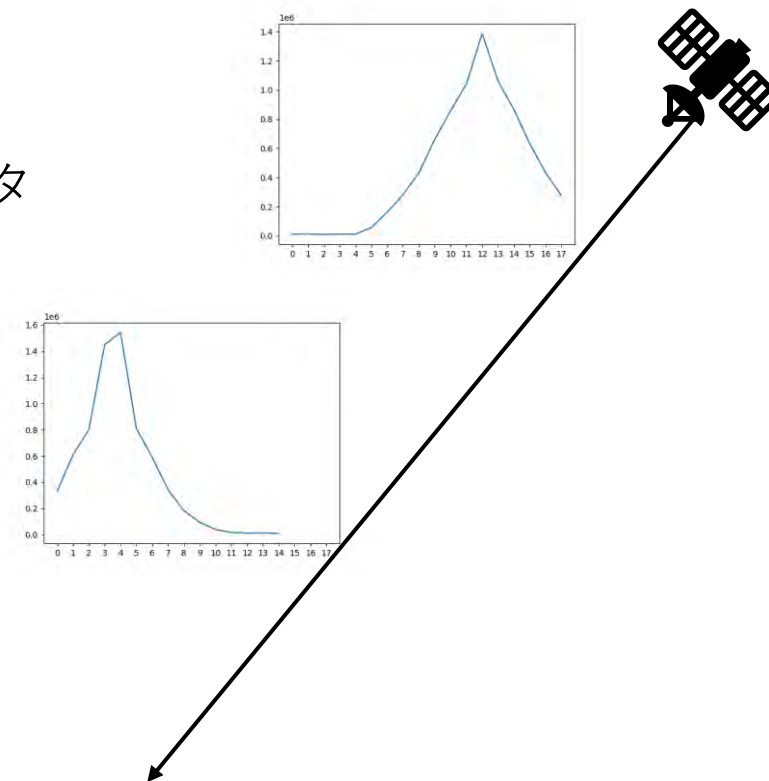
現状の課題

<現状>

異なる時刻に観測されたデータを位置、速度や軌道要素データでのマッチング

<課題>

位置、速度データだけではマッチングが難しいケースも多い
ライトカーブデータの利用による改善



<本研究の目的>

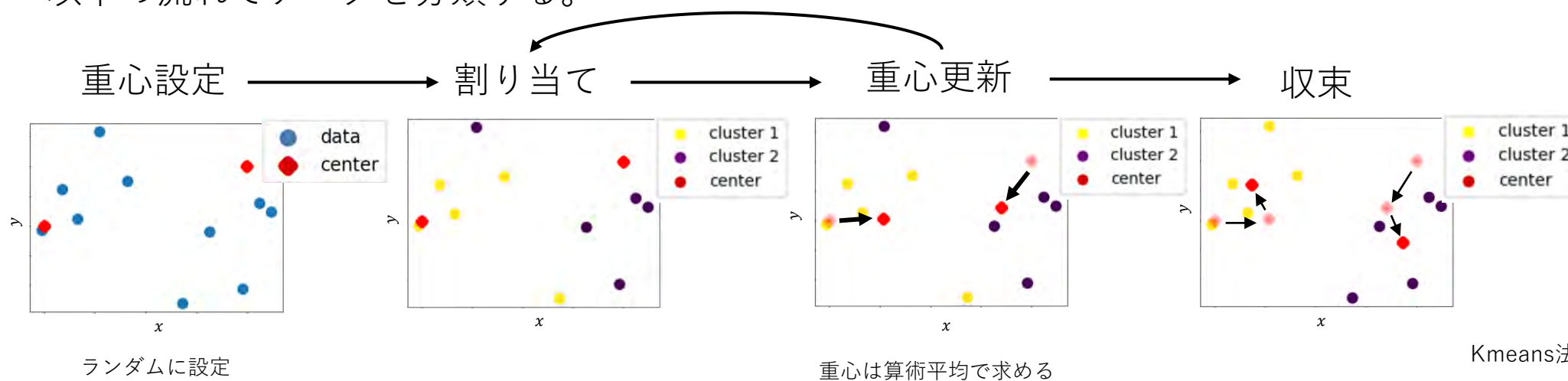
時系列クラスタリングの手法を適用して

ライトカーブデータも考慮した人工天体マッチング方法の確立を目指す

クラスタリング手法

- KMeans法

以下の流れでデータを分類する。



Kmeans法のイメージ図

クラスタの数は事前に指定する

クラスタリング手法

- 時系列データを用いたKMeans法

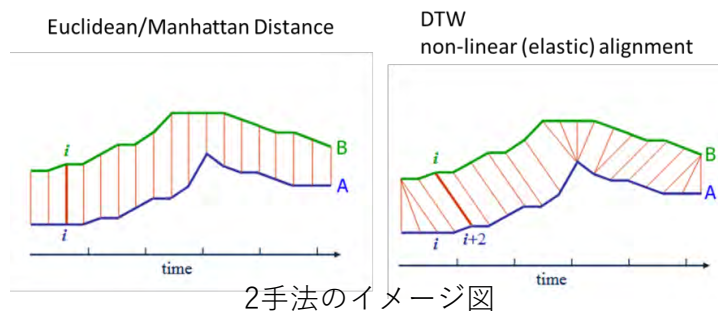
時系列データを用いる場合でも、基本は一般のKmeans法と同じアルゴリズム
ただ、時系列データ間の**距離指標**をどう定義するかが重要
主に使われるのは以下の二手法

Euclidean

各時刻の差の絶対値の合計値

長所：単純、計算が早い

短所：同じ形状でも位相がずれると遠いと認識



2手法のイメージ図

DTW (Dynamic Time Warping)

2つの時系列の誤差の絶対値を総当たりで計算、距離が最小となるパスを探索

長所：**位相がずれたデータ**も近いと認識できる

短所：計算コストが大きい、位相のずれが大きいと認識できない

[DTW\(Dynamic Time Warping\)動的時間伸縮法 – S-Analysis \(data-analysis-stats.jp\)](http://data-analysis-stats.jp/)

この二つのうちだと**DTW**の方が
位相のずれも考慮できる点で有用

クラスタリング手法

- KShape法

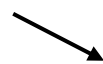
基本的なアルゴリズムはKmeans法と同じだが、**距離指標と重心の求め方が異なる**

距離指標

SBD (Shape Based Distance) を用いる

$$SBD(\vec{x}, \vec{y}) = 1 - \max_w \left(\frac{CC_w(\vec{x}, \vec{y})}{\sqrt{R_0(\vec{x}, \vec{x})R_0(\vec{y}, \vec{y})}} \right)$$

データを相対的にずらして**正規化相互相関関数**を計算し、その最大値を1から引く

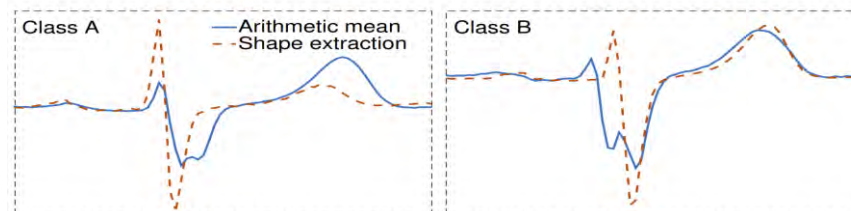


ベクトル間の類似度をはかる関数、ずらして重なった部分の内積をとるイメージ

重心計算

算術平均ではなく、**距離(SBD)の二乗和を最小にする**という条件で重心を求める

→ 形状の特徴を効果的に反映できる



算術平均の方法と二乗和を利用する方法の比較

Paparrizos, John & Gravano, Luis. (2016). k-Shape: Efficient and Accurate Clustering of Time Series. ACM SIGMOD Record. 45. 69-76. 10.1145/2949741.2949758.

評価データ

- 2023年4月中の合計8日分のNEO検出パイプラインの結果
その中のライトカーブデータのうち18フレームが揃っている

(Tomo-e Gozen All-Sky Surveyは各視野18フレームを取得している。そのうち、画像の端にある、暗いなどによるデータの欠損がないもの。)

かつ、**振幅 $>20 \times$ 測光誤差(カウント値)** となっているもの

- 最終的な学習データは**1124個**

解析手法

- Pythonライブラリの**tslearn**を利用

DTW

TimeSeriesKMeans

- `n_clusters = 30`
- `metric = 'dtw'`
- `max_iter = 100`
- `random_state = 0`

実行時間 1min

SBD

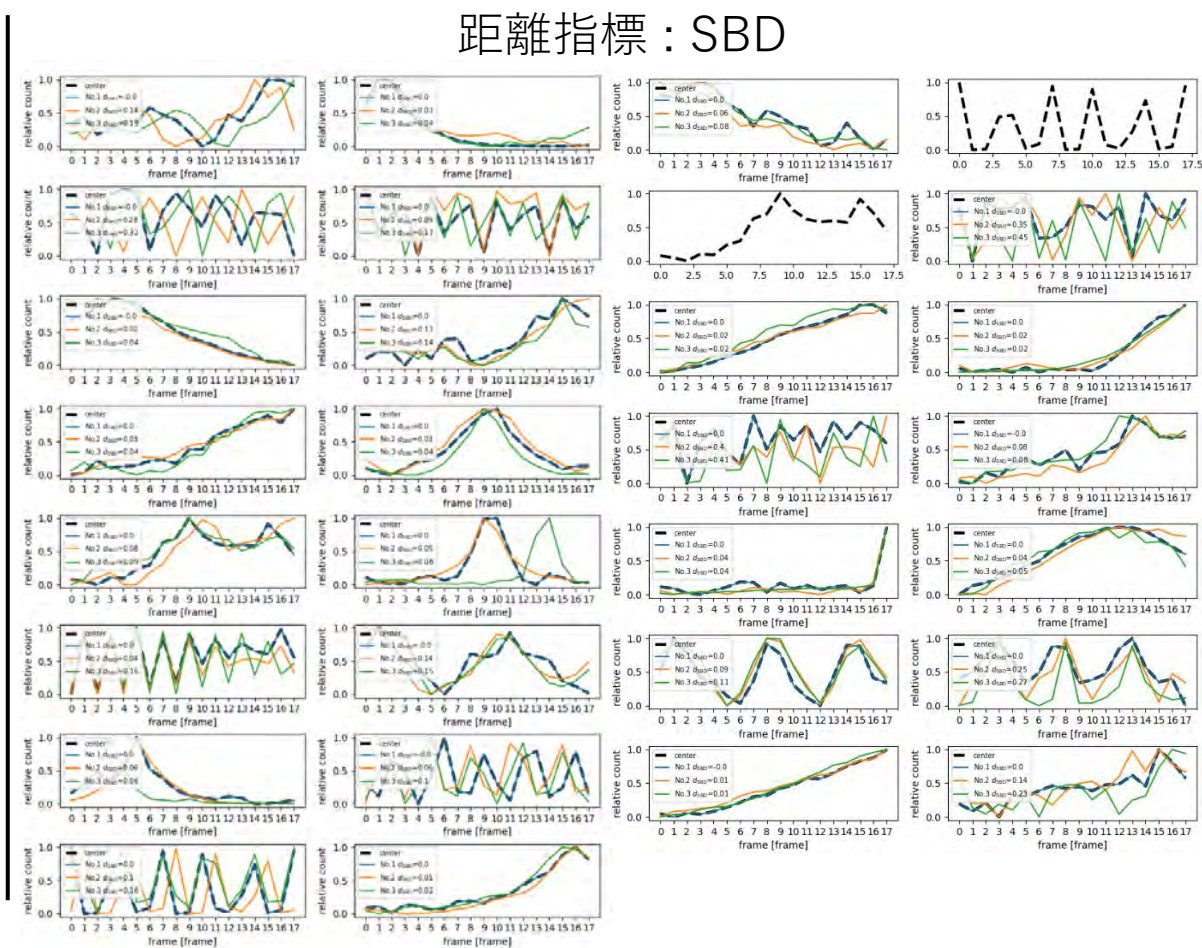
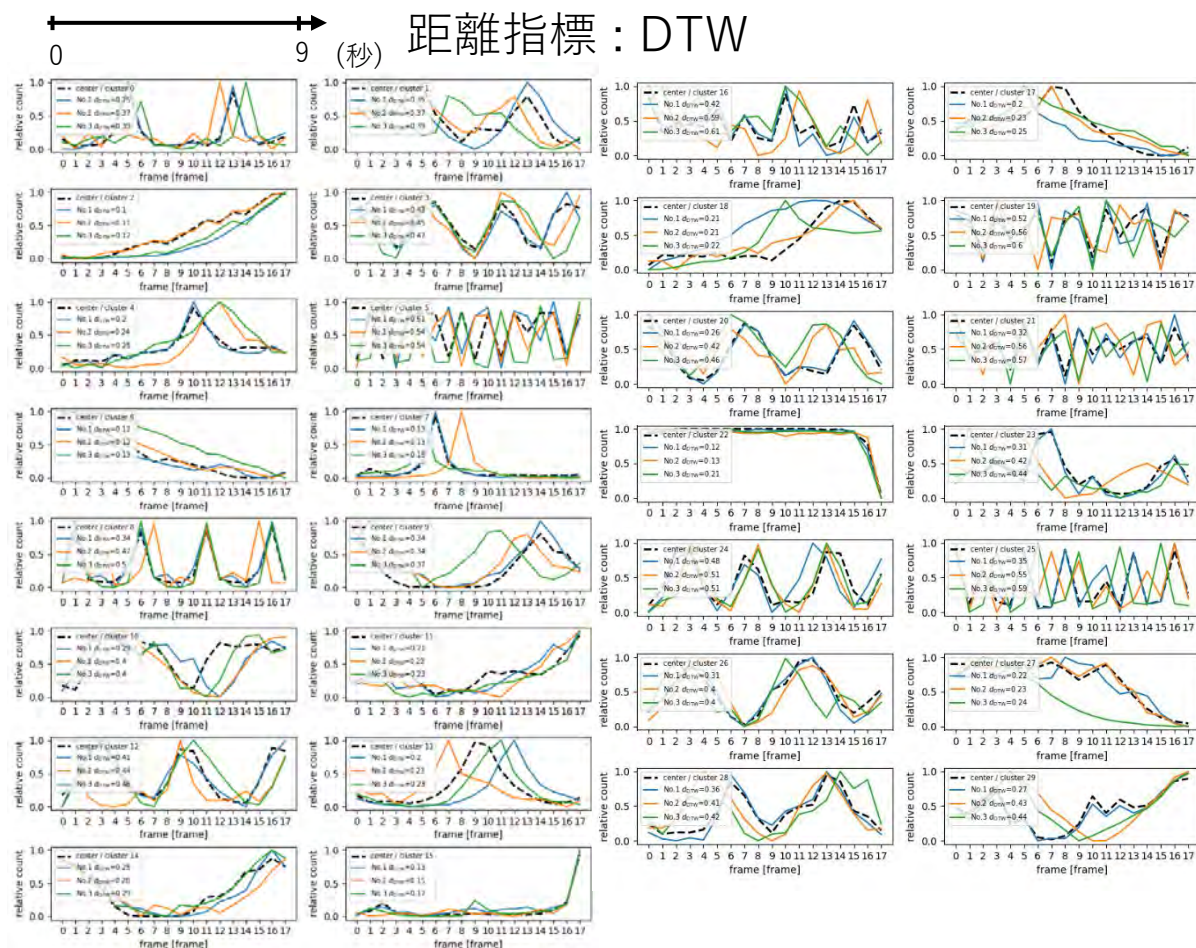
KShape

- `n_clusters = 30`
- `max_iter = 150`
- `random_state = 0`

実行時間 1.5min

- 前処理として、最大値を1、最小値を0とする正規化を実施
- **TimeSeriesScalerMinMax** を使用

30個のクラスタリングの結果

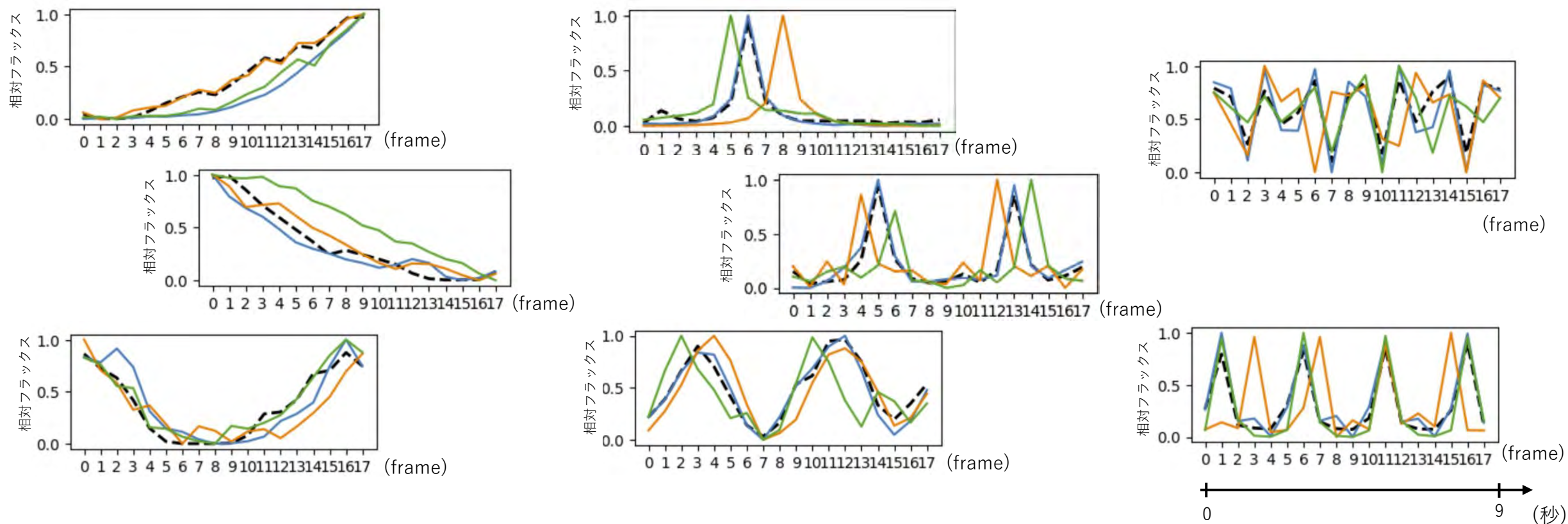


どのグラフも、黒点線がクラスタ重心、青線がクラスタ内のデータで最もクラスタ重心に近いデータ、橙線が二番目に近いデータ、緑線が三番目に近いデータ。

凡例に書かれているdは、それぞれクラスタ重心からの距離。（添え字は距離指標）横軸はフレーム数で縦軸は正規化したフラックス。

30個のクラスタリングの結果

DTW、SBD両手法ともに上昇、下降傾向やピークの有無、個数、形状を認識できた



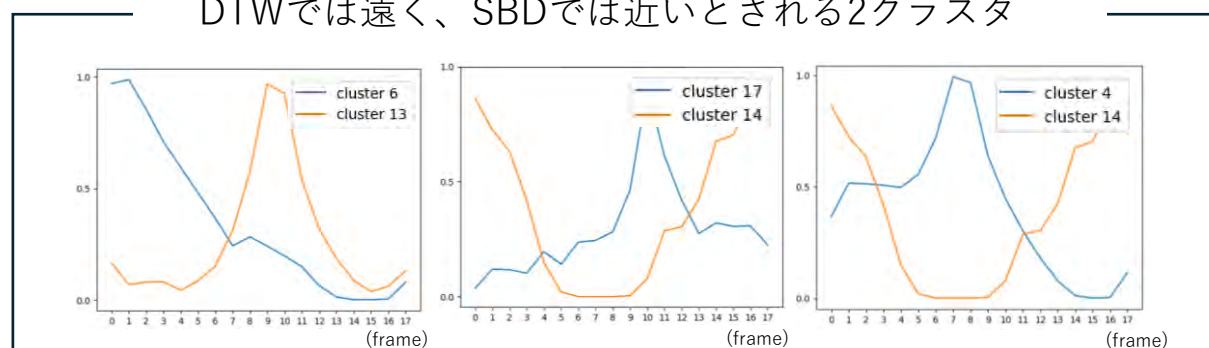
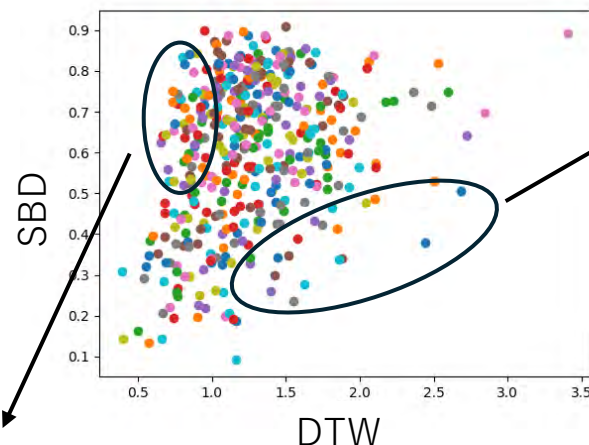
上の図は、DTWで分けた結果の中から抜粋したもの。

DTWとSBDの比較

- クラスタ分けの結果からDTWとSBDの性能を比較する

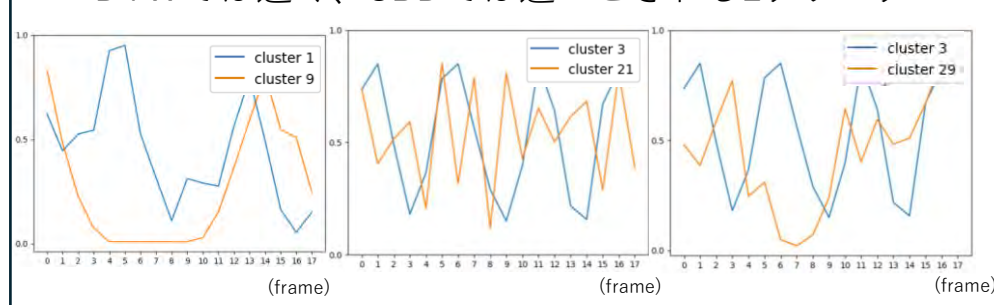
以下はDTWで分けた30クラスタから二つ選び、それらの重心間の距離をDTW、SBDで測りプロットしたもの

DTWでは遠く、SBDでは近いとされる2クラスタ



位相のずれが大きいデータ間の距離を
DTWでは遠いとしてしまっているが、SBDでは近いと判断できている

DTWでは近く、SBDでは遠いとされる2クラスタ



人間の目では似てないように見えるデータ間の距離をDTWでは近いとしてしまっているが、
SBDでは遠いと判断できている

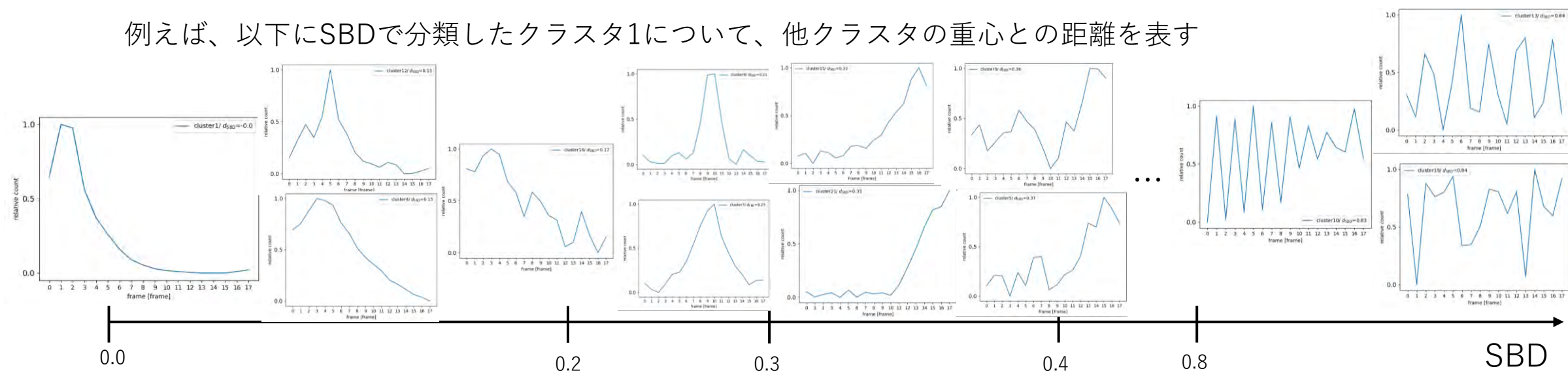
SBDの方がより形状の類似度を測れていることがわかる
ただし、さらなる定量的な比較も必要である

クラスタ重心間の距離

似ているクラスタの統合が必要。またその作業を人間の目でなく機械内で完結させるのが目標

→ 統合候補を挙げるため、クラスタ重心について似ているものの間の距離が近くなっている必要がある

例えば、以下にSBDで分類したクラスタ1について、他クラスタの重心との距離を表す



目で見て似ているものの距離は近く、全く似ていないものの距離は遠い

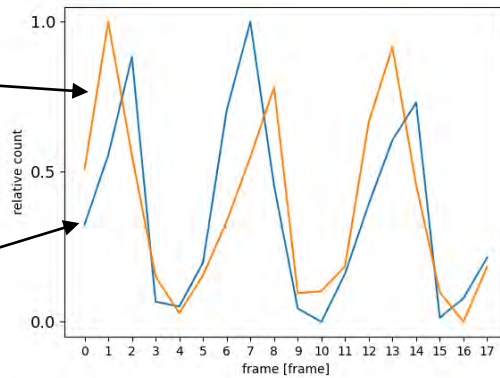
他のクラスタについても同様の図を描くと、目を見た類似性と距離はある程度一致していることが分かった

テストデータで予測

- 最新の2024年4月28日のデータをテストデータとしてクラスタを予測
- カタログ上で同一天体とされている、異なる時刻に観測された2データが同一クラスタに入ることを数例確認

カタログ上のID

BLOCK DM R/B

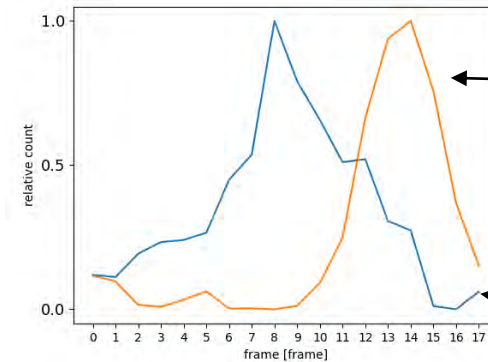


共に同じクラスタに属する

NEO Viewer で見れる GIF 画像

カタログ上のID

NILESAT 102



共に同じクラスタに属する

汎化性能評価をさらにする必要がある

課題

- 18フレーム揃ったデータのみ使用したため、サンプル数が少ない
→ うまく欠損を埋めて活用、使うサンプル数の増加
- サンプル数が増えた場合の計算速度（今回のデータ数では1分程度で完了）
- クラスタ数がどれぐらいがベストなのか？
- クラスタごとの信頼度の定量評価
あるクラスタに入ったときにどのぐらいの確率で正しくマッチングできるかを表した指標
- DTWとSBDの比較の定量評価
- 汎化性能の評価

まとめ

- Tomo-e Gozenによる人工天体データを用いたマッチングにおいて、精度向上が課題であった
- 課題解決のため、**時系列クラスタリング**の手法を用いたライトカーブデータの分類を提案した
- 距離指標としてDTWとSBDを用いた手法でクラスタリング解析を実施
- 結果、上昇下降傾向や、ピークの有無でのクラスタ分けに成功した
- **SBD**の方がより形状の特徴を捉えた分類ができるとわかった
- 今後は、サンプル数の増加、距離指標や汎化性能の定量評価をする予定