SFI
VERTYS Inc.

ANOMALY DETECTION & CLUSTERING GUIDE

# SFI
# 異常予知＆クラスタリング
# アルゴリズム解説資料

## ― 異常検知5種 ＋ クラスタリング11種 ―

～ 現場で使える16のモデル、その本質と選び方 ～

製造現場の生産技術・製造技術・製品開発担当者向け

16

ALGORITHMS

PUBLISHED BY

Vertys Inc. | SFI Product Team

Manufacturing Data Analysis Platform

PieceMaker

01 INTRODUCTION

# 本資料の目的

**アルゴリズム名を「覚える」ための資料ではありません。**
**現場の異常予知課題に、どのアルゴリズムを当てるかが判断できるようになるための資料です。**

製造現場では、「**この設備の挙動はいつもと違う**」「**こんな波形は初めて見る**」といった、**ラベルがない**状態で異常を察知したい場面が多々あります。SFIには16種類のアルゴリズム（純粋な異常検知5種＋クラスタリング11種）が搭載されており、それぞれに得意不得意があります。「とりあえずアイソレーションフォレストを使う」という考え方は、本質的ではありません。

本資料は、統計や機械学習の専門家ではない、**生産技術・製造技術・製品開発の担当者**の皆様が、ご自身の異常予知課題に対して「どのアルゴリズムが筋がよいか」を素早く見極められるよう、たとえ話と図解を中心に整理したものです。

## 本資料が目指す3つの状態

### 01

**2系統の使い分けが分かる**

「異常検知5種」（外れ値特定）と「クラスタリング11種」（パターン発見）の使い分けが言語化できる。

### 02

**「正常モード」を定義できる**

異常予知は「正常を定義し、そこから外れたものを検出する」が本質。クラスタリング＋異常検知の組合せでこれを実現する。

### 03

**「教師なし」の限界を理解する**

ラベルがない以上、「正解」は存在しない。現場知見とのすり合わせが必須であることを腹落ちさせる。

02 TWO APPROACHES

# 異常予知の2つのアプローチ

「外れ値を直接見つける」と「正常パターンを見つけて外れを判定する」

APPROACH 1

## 異常検知系（5種）

Anomaly Detection

「**外れ値を直接見つける**」専用アルゴリズム。
正常データを学習し、新サンプルが正常領域から外れているかを**異常スコア**で出力。

**含まれるモデル：**
◆ アイソレーションフォレスト
◆ 局所外れ値因子 (LOF)
◆ ワンクラスSVM
◆ MT法（マハラノビス・タグチ）
◆ SGDワンクラスSVM

APPROACH 2

## クラスタリング系（11種）

Clustering

「**パターンを発見**」する教師なし学習。
クラスタに属さない点を異常候補として扱う。

**含まれるモデル：**
◆ 重心系（k-means / ミニ / 二分 / BIRCH）
◆ 確率系（GMM / ベイズGMM）
◆ 密度系（DBSCAN / HDBSCAN / OPTICS）
◆ 階層・グラフ系（スペクトラル / 階層凝集）

### 2系統の使い分け

| こんな時は | 推奨アプローチ |
|---|---|
| 外れ値を直接見つけたい | ▶ **異常検知系**（アイソレーションフォレスト等） |
| 運転モードを分類したい | ▶ **クラスタリング系**（k-means / GMM等） |
| 未知の異常パターンを発見 | ▶ **密度系クラスタリング**（DBSCAN / HDBSCAN） |
| 大規模IoTデータの監視 | ▶ **アイソレーションフォレスト** / **LightGBM系** |
| 日本の品質工学に整合 | ▶ **MT法** |

**本資料の重要な構成**

Part 1（P.07-11）で **異常検知系5種**を、Part 2（P.12-22）で **クラスタリング系11種**を解説します。
比較表・早見表では両系統を横断的に整理。

03 HOW TO READ

# 本資料の読み方

まず早見表、次に詳細。逆引きで使うのがおすすめです。

## RECOMMENDED FLOW — 推奨の読み方

STEP 01
**早見表で当たりをつける**

→

STEP 02
**候補2〜3個の詳細ページを読む**

→

STEP 03
**SFIで全部試して比較**

## 本資料の構成

| ページ | セクション | 内容 |
|---|---|---|
| P.02-04 | **はじめに** | 目的・2系統の使い分け・読み方 |
| P.05-06 | **全体マップ** | 16アルゴリズムを5グループに分けて俯瞰 |
| P.07-11 | **Part 1: 異常検知系** | 5アルゴリズム（iForest / LOF / OCSVM / MT / SGD-OCSVM） |
| P.12-22 | **Part 2: クラスタリング系** | 11アルゴリズム（重心・確率・密度・階層/グラフ） |
| P.23-24 | **比較表** | 16モデル × 7観点の◯△×評価 |
| P.25-27 | **早見表（逆引き）** | 現場シチュエーションから推奨アルゴリズム |
| P.28-29 | **SFI活用の流れ** | データ準備から現場展開までの実務フロー |
| P.30-31 | **よくある誤解と注意点** | 「ノイズ＝ゴミ」批判ほか |
| P.32 | **まとめ** | アルゴリズムは道具、選び方は現場知見が決める |

**アルゴリズム詳細ページの色分け凡例**

異常検知系（5種） 重心ベース系 確率モデル系 密度ベース系 階層・グラフ系

04 ALGORITHM MAP

# 16アルゴリズム全体マップ

2系統5グループで俯瞰してみましょう

SFI
**異常予知**
**16種**

PART 1

## 異常検知系

Anomaly Detection / 5種

「外れ値を直接見つける」専用設計。
正常データから境界・密度・距離を学習し、異常スコアを出力する。

- ◆ アイソレーションフォレスト
- ◆ 局所外れ値因子(LOF)
- ◆ ワンクラスSVM
- ◆ MT法（マハラノビス）
- ◆ SGDワンクラスSVM

PART 2 **クラスタリング系 / 11種** Clustering（4グループ）

### GROUP A
**重心ベース系**
Centroid / 4種

- ◆ k-means
- ◆ ミニバッチk-means
- ◆ 二分k-means
- ◆ BIRCH

### GROUP B
**確率モデル系**
Probabilistic / 2種

- ◆ ガウス混合 モデル(GMM)
- ◆ ベイズガウス 混合モデル

### GROUP C
**密度ベース系**
Density / 3種

- ◆ DBSCAN
- ◆ HDBSCAN
- ◆ OPTICS

### GROUP D
**階層・グラフ系**
Hierarchical / 2種

- ◆ スペクトラル クラスタリング
- ◆ 階層凝集 クラスタリング

▸ **POINT** 異常予知ならまず**アイソレーションフォレスト**（Part 1）と**HDBSCAN**（Part 2 / GROUP C）の併用が定石。

05 GROUP CHARACTERISTICS

# 2系統5グループの特徴

それぞれの「向き」と「不向き」を整理します

PART 1

## 異常検知系

5種

**「外れ値を直接見つける」専用設計。**
**正常データから境界・密度・距離を学習し、異常スコアを出力。**

▸ 得意
外れ値の連続スコア化、
閾値で警報レベル分け、ラベルなし運用

▸ 苦手
「パターン分類」（運転モード抽出）、
現場説明（要追加解析）

GROUP A

## 重心ベース系

クラスタ / 4種

**「クラスタの重心」を基準に各点を振り分ける。高速で大規模データに強い。**

▸ 得意
大規模データ、球形クラスタ、計算速度

▸ 苦手
複雑な形、ノイズ、クラスタ数不明

GROUP B

## 確率モデル系

クラスタ / 2種

**ガウス分布で「所属確率」を計算。柔らかな境界と異常スコア化が可能。**

▸ 得意
確率的解釈、楕円形クラスタ、異常スコア

▸ 苦手
分布仮定の崩壊、高次元データ

GROUP C

## 密度ベース系

クラスタ / 3種

**「点が密集した領域」をクラスタとし、まばらな点を異常候補として分離。**
**異常予知の本命。**

▸ 得意
ノイズ・外れ値検出、複雑な形、
クラスタ数不要

▸ 苦手
高次元データ、大規模データ

GROUP D

## 階層/グラフ系

クラスタ / 2種

**木構造・グラフ理論で複雑形状や階層構造を捉える。樹形図の可視化が強み。**

▸ 得意
複雑形状、階層分類、樹形図可視化

▸ 苦手
大規模データ、ハイパラ調整

**POINT** Part 1（異常検知系）とPart 2 GROUP C（密度ベース系）が、異常予知の2本柱。

PART 1 ・ 異常予知系

01

# アイソレーションフォレスト

Isolation Forest（木で外れ値を切り分ける）

ひと言で言うと

**「外れ値ほど少ない切り分けで孤立する」性質を利用した、製造業の異常検知でデファクトとされる定番モデル。**

## 仕組みのイメージ

**たとえ話：**
1000人の社員の中から「特定の1人」を当てるとき、
普通の社員は何度も質問しないと絞れません。
しかし**「変わり者」**は数回の質問ですぐ特定できます。

**ポイント：**
ランダムな質問（特徴量・閾値）を繰り返して
データを切り分けていき、**「少ない切り分けで孤立した点」を異常**と判定。
ラベルなしで動き、高次元・大規模データに強い。

## 現場ユースケース

### ■ 設備の異常運転検知（本命用途）

正常運転データだけで学習し、
新規データの異常スコアを算出。
閾値を超えたら警報を出す。

### ■ 多次元の品質データの外れ値検出

検査値が10〜20次元ある製品で、
複数項目の組合せで異常なロットを発見。

## ▸ 長所

- **大規模・高次元データに強い**
- 連続値の異常スコアが得られる
- パラメータ調整が少なくて済む
- 製造業の異常検知でデファクト

## ▸ 短所・注意点

- 結果がランダム性で微妙にブレる
- 異常が密集している場合は弱い
- カテゴリ変数はそのまま使えない
- 「なぜ異常か」の説明は要追加解析（SHAP等）

## こんな時に選ぶ

◆ **異常予知が主目的**で、まず最初に試すべき第一選択肢
◆ 特徴量数が10次元以上ある（k-NNやLOFが破綻する領域）
◆ 大規模データ（数十万行〜）で高速に異常スコアを出したいとき

▸ **異常予知での立ち位置**「迷ったらアイソレーションフォレスト」と言われる定番。
LOF・ワンクラスSVMと横並びで比較するのが定石。

PART 1 ・ 異常予知系

02

# 局所外れ値因子（LOF）

Local Outlier Factor（周囲より密度が低い点を見つける）

ひと言で言うと

**「周囲の点と比べて、自分の周りだけ密度が低い」点を異常と判定する。**
**局所的な異常検出の代表モデル。**

### 仕組みのイメージ

**たとえ話：**
満員電車の中で「1人だけ離れて立っている」乗客は、
たとえ全体の中央付近にいても**「周囲が空いている」**
から目立ちます。
逆に車両端でも周囲が同じくらいの密度なら普通。

**ポイント：**
自分の近傍k点との**「局所密度の比」**でスコア化。
「全体としての異常」より**「局所的な異常」**を捉えるのが得意。

### 現場ユースケース

**■ 運転モードごとの局所異常検知**

設備が複数の運転モードを持つとき、
「低負荷モード内では普通の値だが、
高負荷モードの中では異常」を発見できる。

**■ 品種別の品質異常検出**

品種ごとに正常値の分布が違う場合に、
品種ごとの「ローカル異常」を判定。

### ▸ 長所

- **局所的な異常**を捉えられる
- 密度がモードごとに違うデータに強い
- 連続値の異常スコアが得られる
- 分布の仮定が不要

### ▸ 短所・注意点

- **大規模データに弱い**（距離計算が重い）
- 近傍点数kの選び方で結果が変わる
- 高次元データで精度が落ちる（次元の呪い）
- スケーリングの前処理が必須

### こんな時に選ぶ

◆ **複数の運転モード・品種が混在**し、モードごとの異常を検出したいとき
◆ アイソレーションフォレストで「モード横断的な異常」を取りこぼすとき
◆ データ数が中規模（数千〜数万）以内

▸ **立ち位置** アイソレーションフォレストと相補的。「全体異常 vs 局所異常」の使い分けが本質的なポイント。

PART 1 ・ 異常予知系

03

# ワンクラスSVM

One-Class SVM（正常データの境界を学習する）

ひと言で言うと

**「正常データを囲む境界線」を引き、その外に出たサンプルを異常と判定する。**
**非線形境界も扱える定番モデル。**

## 仕組みのイメージ

**たとえ話：**
正常運転データの散布図に、それを**「ぴったり囲む輪ゴム」**を描くイメージ。
新しいサンプルが輪ゴムの中なら正常、
外に出たら異常と判定します。

**ポイント：**
「カーネル」と呼ばれる仕掛けで、
輪ゴムを**複雑な形に変形**できます。
直線では囲めない複雑な正常領域にも対応できるのが強み。

## 現場ユースケース

### ■ 正常モードが複雑な形をするとき

設備の正常運転領域が「曲がった範囲」や
「島状に分かれた範囲」のとき、
カーネルSVMで自然に学習できる。

### ■ 振動波形の異常検知

正常波形の特徴量を学習し、
未知の振動パターンを境界外として検出。

## ▸ 長所

- **非線形な正常境界**を学習できる
- 正常データだけで学習可能（教師なし）
- 高次元データでも比較的安定
- 理論的基盤がしっかりしている

## ▸ 短所・注意点

- **大規模データに弱い**（計算量が重い）
- カーネル・パラメータの選択が結果に影響大
- 異常スコアの絶対値の解釈が難しい
- スケーリングの前処理が必須

## こんな時に選ぶ

◆ **正常領域が複雑な形**をしていそうなとき
◆ アイソレーションフォレストで「形を捉えきれない」と感じたとき
◆ データ数が中規模（数千〜数万）以内

▸ **立ち位置** かつての異常検知の主役。今はアイソレーションフォレストに譲ったが、「非線形境界」を扱う場面では依然強力。

PART 1 ・ 異常予知系

04

# MT法（マハラノビス・タグチ法）

Mahalanobis-Taguchi（データの中心から遠い点を見つける）

ひと言で言うと

**日本のタグチメソッドが起源。**
**「正常データの中心からの距離」で異常を判定する、製造業の品質工学の伝統技。**

## 仕組みのイメージ

**たとえ話：**
正常データの「中心点」と「ばらつきの形（楕円）」を計算し、新しいサンプルとの**「マハラノビス距離」**を測る。
普通の直線距離ではなく、ばらつきを考慮した距離です。

**ポイント：**
距離の指標を**「MD値」**と呼び、日本の品質工学・QC会議で広く使われています。
理論がシンプルで現場説明がしやすいのが特徴。

## 現場ユースケース

### ■ 品質工学に基づく異常検知

タグチメソッドを導入している現場で、
SN比やMD値を用いて多変量の異常判定を行う。

### ■ 監査・規格対応

古くから日本の製造業で使われ、
品質会議・社内規格との親和性が高い。
説明資料に最適。

## ▸ 長所

- **日本の品質工学との親和性**が高い
- 理論がシンプル、説明性が高い
- 計算が高速
- 変数間の相関を考慮した距離

## ▸ 短所・注意点

- **正規分布の仮定**が前提（崩れると精度低下）
- 非線形・複雑な正常領域には弱い
- 外れ値の影響を受けやすい
- 高次元データでは共分散行列が不安定

## こんな時に選ぶ

◆ **タグチメソッド・SN比管理**を導入している現場
◆ 品質会議・社内規格との整合性が必要なとき
◆ 正常データの分布が正規分布に近いと分かっているとき

▸ **立ち位置** 日本の製造業に深く根付いた古典手法。アイソレーションフォレストとの「精度 vs 説明性」比較で価値を発揮します。

PART 1 ・ 異常予知系

05

# SGDワンクラスSVM

SGD One-Class SVM（確率的勾配降下で線形の正常境界を学習）

ひと言で言うと

**ワンクラスSVMの「大規模データ対応・高速化版」。線形境界に限定する代わりに、桁違いに速く動く。**

## 仕組みのイメージ

**たとえ話：**
普通のワンクラスSVMが「全データを一度に見て境界を引く」のに対し、SGD版は**「データを1点ずつ見て、少しずつ境界を動かす」**方式で大規模データに対応します。

**ポイント：**
非線形カーネルは使えませんが、
**数百万行のデータでも実用時間で動作。**
リアルタイム監視やストリーミングデータに適合します。

## 現場ユースケース

### ■ 大規模IoTデータの異常検知

数秒間隔で記録される設備センサデータ（数百万行）から、線形境界で異常を高速検出。

### ■ ストリーミング異常検知

逐次データが流れてくる環境で、
モデルを少しずつ更新しながら異常を検知。

## ▸ 長所

- **大規模データで高速**
- 逐次更新（オンライン学習）が可能
- メモリ消費が少ない
- 線形なので結果の解釈がしやすい

## ▸ 短所・注意点

- **非線形境界は扱えない**（線形限定）
- 学習率の調整が必要
- 結果が学習順序にやや依存
- 複雑な正常領域には不向き

## こんな時に選ぶ

◆**数百万行クラスの大規模データ**でワンクラスSVMが動かないとき
◆リアルタイム監視・ストリーミング処理が必要なとき
◆正常領域がシンプル（線形で囲める）と判断できるとき

▸ **使いどころ** ワンクラスSVMで「データが大きすぎて動かない」場合の代替案。アイソレーションフォレストと役割が近い。

GROUP 01 ・ 重心ベース系

01

# k-means

k-means Clustering（重心の近さでグループ化）

ひと言で言うと

**クラスタリングの「最も基本」。**
**データを指定した個数（k個）のグループに、重心からの距離で振り分ける。**

## 仕組みのイメージ

**たとえ話：**
工場の運転データを「3つの運転モードがあるはず」と仮定して、k=3を指定。
アルゴリズムが3つの重心を見つけ、
各データ点を最も近い重心のグループに割り当てます。

**ポイント：**
「k（クラスタ数）を人間が事前に決める」
必要があるのが最大の特徴。
**単純で速いが、複雑な形のクラスタや異常検知には弱い。**

## 現場ユースケース

### ■ 運転モードの分類

設備の運転データを「立上げ／定常／立下げ」のように事前に既知の数だけグループ化したいとき。

### ■ ロット品質のグルーピング

品質データを「A級／B級／C級」のように3つに分けたい、と明確な目的があるとき。

## ▸ 長所

- **高速・シンプル**（最古典のアルゴリズム）
- 大規模データでも実用速度
- 結果が直感的に分かりやすい
- ベースラインとして必ず最初に試す

## ▸ 短所・注意点

- **クラスタ数kを事前指定**が必要
- 球形に近いクラスタしか扱えない
- 外れ値・ノイズに弱い（異常検知に不向き）
- 初期値依存で結果がブレることがある

## こんな時に選ぶ

◆ クラスタ数が**事前に分かっている**（運転モードの数など）
◆ データがおおむね球形にまとまっていそうなとき
◆ クラスタリングの**「ベースライン」**として最初に走らせるとき

▸ **異常予知での立ち位置** k-means自体は異常検出には不向き。ただし「正常モードを定義し、そこから外れた距離で異常スコア化」という使い方は可能です。

GROUP 01 ・ 重心ベース系

02

# ミニバッチ k-means

Mini-Batch k-means（小分けデータで重心を更新する大規模向け版）

ひと言で言うと

**大規模データ向けの「軽量版k-means」。**
**データを小分けにして重心を逐次更新するため、桁違いに速い。**

### 仕組みのイメージ

**たとえ話：**
普通のk-meansは「100万行を毎回ぜんぶ見て」重心を計算します。
ミニバッチ版は「ランダムに1000行だけ取り出して重心を少し動かす」を何度も繰り返します。

**ポイント：**
精度はk-meansよりやや劣りますが、
**数百万行のIoTデータ**でも実用時間で動くのが最大の強み。

### 現場ユースケース

**■ 大規模IoTデータの運転モード抽出**

数秒間隔のセンサデータが数千万行ある工場で、
設備運転モードを抽出。

**■ ストリーミングデータ対応**

データが逐次流れてくる状況で、
定期的に再学習する用途。

### ▸ 長所

- **k-meansの数十〜数百倍高速**
- メモリ消費が少ない
- ストリーミング処理に適合
- k-meansと同じ結果になりやすい

### ▸ 短所・注意点

- k-meansよりわずかに精度が劣る
- k-meansと同じ限界（球形仮定・k事前指定）
- バッチサイズの調整が必要
- 外れ値・ノイズに弱い（異常検知には不向き）

### こんな時に選ぶ

◆ **数百万〜数千万行のIoTデータ**を扱うとき
◆ k-meansで「学習が終わらない」と感じたとき
◆ 定期的な再学習が必要なストリーミング用途

▸ **使いどころ** k-meansで速度に困ったらまずこれ。精度の差はわずかなので、実用上はミニバッチ版で十分なケースが多い。

GROUP 01 ・ 重心ベース系

03

# 二分 k-means

Bisecting k-means（再帰的に2分割していくk-means）

ひと言で言うと

**全データを2つに分け、また2つに分け…を繰り返すk-means。**
**階層的でツリー状の分類が得られる。**

### 仕組みのイメージ

**たとえ話：**
「とりあえず全データを2つのグループに分ける」
→「大きい方をまた2つに分ける」
→「さらに大きい方を…」と再帰的に分割していく。

**ポイント：**
結果が**ツリー（階層）構造**になるため、
「大分類→中分類→小分類」と粒度を変えて見ることができる。普通のk-meansより精度が高くなりやすい。

### 現場ユースケース

**■ 階層的な運転モード分類**

「立上げ／定常／立下げ」→「定常を低負荷・中負荷・高負荷」のように、段階的に細分化したいとき。

**■ k-meansの精度改善版**

普通のk-meansで結果が不安定なとき、
二分版に切り替えるだけで精度向上するケースが多い。

### ▸ 長所

- k-meansより**精度が高い**傾向
- 初期値依存が少ない（安定した結果）
- **階層構造**が得られる
- 大規模データでも比較的高速

### ▸ 短所・注意点

- k-meansよりは遅い
- 球形仮定は変わらず（複雑な形に弱い）
- kは事前指定が必要
- 外れ値・ノイズに弱い

### こんな時に選ぶ

◆ **「粗→細」と階層的に分類**したいとき
◆k-meansの結果が実行ごとに変わって困っているとき
◆k-meansよりも安定した結果が欲しいとき

▸ **立ち位置** k-meansの上位互換と見てよい。速度を多少犠牲にして精度・安定性を取りたい場面の標準選択肢。

GROUP 01 ・ 重心ベース系

04

# BIRCH

Balanced Iterative Reducing Clustering using Hierarchies（圧縮要約型）

ひと言で言うと

**大規模データを「要約」してからクラスタリングする。一度通すだけで動作する省メモリ設計。**

## 仕組みのイメージ

**たとえ話：**
100万行のデータを直接クラスタリングする代わりに、まず「**CFツリー**」という圧縮要約を作ります（似た点をまとめて「平均と分散」だけ保存）。要約データに対してクラスタリングするので桁違いに省メモリ。

**ポイント：**
「全データを1回スキャンするだけ」で動作。ストリーミングデータにも適合します。

## 現場ユースケース

### ■ 超大規模データの初期分析

数億行のIoTデータから、まずは「ざっくりどんなパターンがあるか」を把握する。

### ■ メモリ制約のあるエッジ環境

工場のエッジサーバなど、メモリが限られる環境でクラスタリングを動かしたいとき。

## ▸ 長所

- **省メモリ・高速**（要約してから処理）
- 1回のスキャンで動作
- ストリーミングデータに適合
- 他クラスタリングの前処理にも使える

## ▸ 短所・注意点

- 要約による情報損失（精度はやや低下）
- 高次元データには弱い
- 球形仮定（複雑な形には不向き）
- パラメータ（閾値）の調整が必要

## こんな時に選ぶ

◆ **データが大きすぎて他のアルゴリズムが動かない**とき
◆ メモリ制約のあるエッジ環境で動作させたいとき
◆ 大規模データの**初期スクリーニング**として

▸ **使いどころ** 「データが大きすぎる」が出発点の選択肢。精度が必要なら、BIRCHで圧縮→他アルゴリズムで再クラスタリング、という2段構えも有効です。

GROUP 02 ・ 確率モデル系

05

# ガウス混合モデル（GMM）

Gaussian Mixture Model（ガウス分布の混合で確率的にグループ化）

ひと言で言うと

**複数のガウス分布（正規分布）の重ね合わせでデータを表現。**
**「所属確率」と「異常スコア」が同時に得られる。**

### 仕組みのイメージ

**たとえ話：**
運転データの散布図を見て、「楕円形の山が3つ重なっている」と仮定。
各データ点に「山Aに所属する確率70%、山Bに30%」のように**確率値で割り振り**します。

**ポイント：**
k-meansが「最も近い1つ」に強制するのに対し、GMMは「どの山にもあまり所属しない＝**確率が低い＝異常候補**」という解釈が自然にできます。

### 現場ユースケース

**■ 異常スコアの算出**

正常運転データでGMMを学習させ、新サンプルの「所属確率の低さ」を異常スコアとして使う。

**■ 楕円形クラスタの抽出**

工程パラメータの散布図で、k-meansの円形仮定では捉えきれない楕円形のクラスタを抽出。

### ▸ 長所

- **所属確率・異常スコア**が得られる
- 楕円形クラスタも扱える
- 確率モデルなので統計的解釈が明確
- 柔らかな境界（境界付近の点を識別可）

### ▸ 短所・注意点

- クラスタ数（成分数）の事前指定が必要
- ガウス分布の仮定が崩れると精度低下
- 初期値依存（局所解にハマる）
- 高次元データでは過学習しやすい

### こんな時に選ぶ

◆ **異常スコアを連続値で出したい**とき（閾値で異常判定）
◆ k-meansで「境界が無理矢理すぎる」と感じたとき
◆ 正常運転データから「未知異常」を統計的に検知したいとき

▸ **異常予知での価値** 「確率の低い領域＝異常候補」という解釈が、製造業の品質管理（管理限界線）と相性が良いのが最大の強みです。

GROUP 02 ・ 確率モデル系

06

# ベイズガウス混合モデル

Bayesian Gaussian Mixture Model（成分の重みをベイズ的に調整）

ひと言で言うと

**GMMの「クラスタ数を自動で決める」版。**
**不要な成分の重みを自動で0に近づけ、適切な数のクラスタに収束する。**

## 仕組みのイメージ

**たとえ話：**
「とりあえず最大10個のクラスタがある」と
多めに指定して始めると、ベイズ的な仕組みが
「不要な成分の重みを自動で0近くにする」ため、
実質的なクラスタ数を自動決定します。

**ポイント：**
普通のGMMは「成分数を間違えると精度が落ちる」のが弱点。
ベイズ版は**「多めに指定しておけばOK」**な堅牢性が魅力です。

## 現場ユースケース

### ■ クラスタ数が不明な異常検知

設備の運転モードがいくつあるか事前に分からない場合に、
自動で適切な数を発見する。

### ■ GMMよりロバストな運用

長期運用で「運転モードが少しずつ増える」状況でも、
ベイズ版なら自動で吸収しやすい。

## ▸ 長所

- **クラスタ数の自動決定**
- 過学習を抑制（不要な成分を消す）
- GMMの利点をすべて継承（異常スコア等）
- 長期運用での適応性が高い

## ▸ 短所・注意点

- GMMより**計算コストが高い**
- 事前分布の設定で挙動が変わる
- 解釈に統計の素養が要る
- 高次元データには弱い

## こんな時に選ぶ

◆ **クラスタ数が事前に分からない**とき
◆ GMMの結果がクラスタ数依存で不安定なとき
◆ 長期運用で運転モードが変動する設備の監視

▸ **使いどころ** GMMの「k指定の悩み」から解放されるのが最大の価値。少し遅くなる代償は十分釣り合います。

GROUP 03 ・ 密度ベース系

07

# DBSCAN

Density-Based Spatial Clustering of Applications with Noise

ひと言で言うと

**密度が高い領域をクラスタ、まばらな点を「ノイズ＝異常」として分離する、異常予知の本命モデル。**

## 仕組みのイメージ

**たとえ話：**
「半径εの中に最低N個の点があれば仲間」と判定。
仲間がたどれる連鎖はすべて同じクラスタ。
誰の仲間にもなれないポツンとした点は
**「ノイズ」として区別**します。

**ポイント：**クラスタ数を事前指定する必要がなく、
**複雑な形のクラスタ**も自然に扱えます。
「ノイズ＝異常候補」がアルゴリズムの設計に
組み込まれている。

## 現場ユースケース

### ■ 設備の異常運転パターン検出

正常運転データは密集したクラスタを形成、
異常時はその外れに点在 → DBSCANで自動分離。

### ■ 振動波形の外れ値検出

振動センサ波形の特徴量空間で、通常と違う波形を
「ノイズ」として浮かび上がらせる。

## ▸ 長所

- **外れ値（ノイズ）を明示的に分離**
- クラスタ数の事前指定が不要
- 複雑な形のクラスタも扱える
- 異常予知の標準モデル

## ▸ 短所・注意点

- **密度パラメータ（ε, minPts）の調整**が要
- クラスタごとに密度が違うと弱い
- 高次元データで距離計算が破綻しがち
- 大規模データで遅い

## こんな時に選ぶ

◆ **異常予知が主目的**のとき（ノイズ分離が要）
◆ クラスタ数が事前に分からない
◆ クラスタの形が複雑（円形仮定が成り立たない）

▸ **異常予知での立ち位置** 「異常予知ならまずDBSCAN」と言われる定番。パラメータ調整の難しさをHDBSCANが解決した、と理解すると流れがスムーズ。

GROUP 03 ・ 密度ベース系

08

# HDBSCAN

Hierarchical DBSCAN（階層型DBSCAN）

ひと言で言うと

**DBSCANの「密度パラメータ調整がいらない」進化版。**
**クラスタごとに密度が違っても自然に対応。**

## 仕組みのイメージ

**たとえ話：**
DBSCANは「半径εに最低N個」と全体共通の閾値でしたが、
HDBSCANは**「クラスタごとに密度が違っても、安定して存在し続けるクラスタを残す」**仕組み。

**ポイント：**
「最小クラスタサイズ」だけ指定すれば動く。
DBSCANより**堅牢で扱いやすい**ため、
近年は密度ベースの第一選択肢になっています。

## 現場ユースケース

### ■ クラスタ密度がばらつく異常検知

高負荷運転は点が密集、低負荷運転はまばら、というように**密度がモードごとに違う**設備データ。

### ■ DBSCANのパラメータに悩んだら

DBSCANで結果が安定しない、ハイパラ調整が大変というときの第一選択肢。

## ▸ 長所

- **密度パラメータ調整不要**
- クラスタごとに密度が違っても対応
- ノイズ分離（異常検知）の精度が高い
- 確率的なクラスタ所属度も得られる

## ▸ 短所・注意点

- DBSCANよりやや遅い
- 高次元データでは距離計算が不安定
- 大規模データではメモリ消費が大きい
- 最小クラスタサイズの感度が高い

## こんな時に選ぶ

◆ **異常予知の主力**として（迷ったらまずこれ）
◆ DBSCANで結果が不安定なとき
◆ クラスタ密度がモードごとに大きく違うとき

▸ **異常予知での立ち位置** 近年の事実上の標準。DBSCANの理論的優位を保ちつつ、実務上の使いにくさを解消した完成度の高いモデル。

GROUP 03 ・ 密度ベース系

09

# OPTICS

Ordering Points To Identify the Clustering Structure

ひと言で言うと

**DBSCANの「密度ばらつきに強い」派生版。**
**密度の濃淡をそのまま順序データとして可視化できる。**

## 仕組みのイメージ

**たとえ話：**
各点に「到達可能距離」というスコアを与え、
すべての点を順番に並べます。
**縦軸=到達距離のグラフ（リーチャビリティプロット）**で
「谷」の部分がクラスタ、「山」が境界やノイズと一目で
分かります。

**ポイント：**
DBSCAN/HDBSCANの「中間的存在」。
**可視化に強い**のが他にない特長。

## 現場ユースケース

### ■ クラスタ構造の探索的分析

設備データに「いくつクラスタがあるか分からない」状況で、
リーチャビリティプロットから直感的に判断。

### ■ 複雑な密度構造の可視化

品質会議で「データのまとまり具合」を説明したいとき、
視覚的にわかりやすい資料が作れる。

## ▸ 長所

- 密度のばらつきに強い
- **リーチャビリティプロットで可視化**
- 探索的分析に最適
- クラスタ数の事前指定が不要

## ▸ 短所・注意点

- DBSCAN/HDBSCANより**遅い**
- 結果からクラスタを抽出するのに追加処理が必要
- 大規模データに不向き
- 高次元データでは距離計算が破綻しがち

## こんな時に選ぶ

◆ **クラスタ構造を視覚的に確認**したいとき
◆ 「いくつクラスタがあるか」を探索的に調べたいとき
◆ 品質会議で「データの密度構造」を説明したい場面

▸ **立ち位置** 本番運用というより「データ理解の道具」。最初にOPTICSで構造を見て、
本番はDBSCAN/HDBSCANに切り替える運用が現実的。

GROUP 04 ・ 階層 / グラフ系

10

# スペクトラルクラスタリング

Spectral Clustering（グラフ理論を使ったクラスタリング）

ひと言で言うと

**データ点を「グラフ（ネットワーク）」として捉え、
グラフ理論で複雑な形のクラスタも切り分ける。**

## 仕組みのイメージ

**たとえ話：**
各データ点を「ノード」、近い点同士を「エッジ」で結んだネットワークを作ります。
そのネットワークを**「最も自然な切れ目」**で分割するのがスペクトラルクラスタリング。

**ポイント：**
三日月形やリング状などk-meansが苦手な**複雑な形**でも、グラフの繋がりに沿って自然に分割できます。

## 現場ユースケース

### ■ 振動波形・画像の複雑形状クラスタ

特徴量空間で曲線的に分布するデータ（振動の周期パターンなど）を、形状に沿って分割。

### ■ k-meansで失敗する複雑な分布

k-meansでクラスタ境界が不自然になる場面で、最後の選択肢として有効。

## ▸ 長所

- **複雑な形のクラスタ**を切り分けられる
- 三日月・リング形状でも対応可
- 非凸（へこんだ形）のデータに強い
- k-meansが諦めた問題への最後の手段

## ▸ 短所・注意点

- **大規模データに不向き**（計算量が爆発）
- クラスタ数の事前指定が必要
- 類似度（カーネル）の選択が結果に影響
- 外れ値の影響を受けやすい

## こんな時に選ぶ

◆ **k-meansやGMMで境界が不自然**になるとき
◆ クラスタの形が球形・楕円ではないと分かっているとき
◆ データ数が中規模（数千〜数万）以内

▸ **立ち位置** 「形が複雑」が主要因のときの選択肢。データが大きい場合はHDBSCANの方が現実的なケースが多い。

GROUP 04 ・ 階層 / グラフ系

11

# 階層凝集クラスタリング

Hierarchical Agglomerative Clustering（HAC）

ひと言で言うと

**近い点・近いクラスタを下から順にまとめていく。**
**樹形図（デンドログラム）で全粒度の関係が見える。**

## 仕組みのイメージ

**たとえ話：**
最初は各データ点を1つのクラスタとし、「最も近い2つ」を繰り返し合体させていきます。
最終的には1つの大クラスタに収束。

**ポイント：**
合体の履歴を**デンドログラム（樹形図）**として描けば、「クラスタを何個に分けるか」を後から決められます。
生物分類・系統樹の発想に近い。

## 現場ユースケース

### ■ 不良モードの階層分類

不良サンプルを樹形図で整理し、
「大分類3種→中分類7種→小分類15種」と階層的に整理。

### ■ 試作データの探索的分析

試作段階の少量データで、
データの自然な階層構造を可視化する。

## ▸ 長所

- **樹形図（デンドログラム）で可視化**できる
- クラスタ数を後から自由に決められる
- 階層関係が明確に見える
- 結果が決定論的（毎回同じ）

## ▸ 短所・注意点

- **大規模データに不向き**（計算量爆発）
- 「合体の距離計算」方法選択で結果が変わる
- 外れ値に弱い
- 一度合体すると後戻りできない

## こんな時に選ぶ

◆ **階層的な分類体系**が欲しい（不良モードのカテゴリ整理など）
◆ 樹形図を品質会議の説明資料として使いたいとき
◆ データが少量（数千件以内）の試作段階

▸ **立ち位置** 「樹形図の可視化」を最大の武器とする古典手法。説明可能性を最優先する場面で根強い人気があります。

05 COMPARISON TABLE (1/2)

# アルゴリズム比較表

7つの観点から16アルゴリズムを ○ △ × で評価

| アルゴリズム | 大規模データ | クラスタ数事前不要 | 複雑な形状 | 外れ値・ノイズ検出 | 速度 | 異常スコア出力 | 可視化 |
|---|---|---|---|---|---|---|---|
| アイソレーションフォレスト | ○ | ○ | △ | ○ | ○ | ○ | △ |
| 局所外れ値因子(LOF) | × | ○ | ○ | ○ | × | ○ | △ |
| ワンクラスSVM | × | ○ | ○ | ○ | × | ○ | △ |
| MT法 | ○ | ○ | × | ○ | ○ | ○ | ○ |
| SGDワンクラスSVM | ○ | ○ | × | ○ | ○ | ○ | △ |
| k-means | ○ | × | × | × | ○ | △ | △ |
| ミニバッチk-means | ○ | × | × | × | ○ | △ | △ |
| 二分k-means | ○ | × | × | × | ○ | △ | ○ |
| BIRCH | ○ | △ | × | × | ○ | × | △ |
| ガウス混合モデル(GMM) | △ | × | △ | ○ | △ | ○ | △ |
| ベイズガウス混合モデル | △ | ○ | △ | ○ | × | ○ | △ |
| DBSCAN | △ | ○ | ○ | ○ | △ | ○ | △ |
| HDBSCAN | △ | ○ | ○ | ○ | △ | ○ | ○ |
| OPTICS | × | ○ | ○ | ○ | × | ○ | ○ |
| スペクトラル | × | × | ○ | △ | × | × | △ |
| 階層凝集 | × | ○ | △ | △ | × | × | ○ |

**凡例**

○ 得意　△ 条件付き　× 不得意

▸ **異常予知の主軸 = 「外れ値・ノイズ検出」「異常スコア出力」列**

この2列が両方○のモデルが異常予知の本命。**特に「アイソレーションフォレスト」「HDBSCAN」「GMM」**が両軸で○を取り、迷ったらまずこの3つを横並びで試すのが定石です。

▸ **異常検知系（PART 1）の特徴**

5モデルすべてで「異常スコア出力○」「外れ値検出○」を達成。
**アラート閾値の現場調整がしやすい**のが純粋な異常検知系の最大のメリットです。

▸ **読み方のコツ** 「○が多い＝最良」ではありません。**課題で重視すべき列だけ**を見て候補を絞ってください。

05 COMPARISON TABLE (2/2)

# 7つの観点 — どこを見るべきか

比較軸の意味と、現場でどう判断材料にするか

### ▸ 大規模データ

数万件を超える大規模データに耐えるか。
IoT・センサデータが多い現場なら最重要。
**iForest / SGD-OCSVM / MT法 / BIRCH**が強い。

### ▸ クラスタ数事前不要

「いくつクラスタがあるか」を事前に決めなくてよいか。
**異常検知系すべて・密度系・ベイズGMM**が強い。

### ▸ 複雑な形状

三日月形・リング形などk-meansの円形仮定で扱えない形状に対応できるか。
**LOF / ワンクラスSVM / 密度系**が強い。

### ▸ 外れ値・ノイズ検出

**異常予知では最重要。**
「クラスタに属さない点＝異常」と分離できるか。
**異常検知系全種・密度系・GMM**が強い。

### ▸ 速度

学習・推論の速さ。
リアルタイム監視や頻繁な再学習が要件なら重要。
**iForest / MT法 / SGD-OCSVM / k-means系**が速い。

### ▸ 異常スコア出力

「異常度」を連続値で出力できるか。
アラート閾値調整に必須。
**異常検知系すべて / GMM / HDBSCAN / OPTICS**が強い。

### ▸ 可視化

樹形図・リーチャビリティプロット等で、クラスタ構造を視覚的に把握できるか。
**階層凝集・OPTICS・HDBSCAN・MT法**が強い。

### ▸ 異常予知の優先軸

①「外れ値・ノイズ検出」を最優先
②「異常スコア出力」で連続値判定
③「クラスタ数事前不要」で自動化
④「速度」で大規模対応
⑤「可視化」で現場説明

## 標準的な比較セット例（実務で「まずこの3〜5個」）

**異常検知の本命**

アイソレーションフォレスト / LOF / ワンクラスSVM

**パターン発見の本命**

HDBSCAN / GMM / k-means

**特殊用途**

MT法（品質工学）/ OPTICS（可視化）/ 階層凝集（分類体系）

▸ **SFI活用** 上記の組合せをSFIで「一括実行→比較」するのが最効率です。

06 QUICK REFERENCE (1/3)

# こんな時はこのアルゴリズム①

データの性質から逆引きする

## 01 外れ値・異常を直接検出したい

**▶ アイソレーションフォレスト / LOF / ワンクラスSVM**
純粋な異常検知の本命3兄弟。
SFIで横並び比較してデータに合うものを選ぶ。

## 02 運転モードをパターン分類したい

**▶ HDBSCAN / GMM / k-means**
クラスタリング系で運転モードを発見。
クラスタに名前を付けて「正常」を定義する。

## 03 大規模IoTデータ（数百万行以上）

**▶ アイソレーションフォレスト / SGD-OCSVM / ミニバッチk-means**
大規模対応の3兄弟。リアルタイム監視も視野。

## 04 複雑な形のクラスタ（三日月・リング形状）

**▶ LOF / ワンクラスSVM / 密度系**
球形仮定に縛られないモデル群。
振動波形特徴量などで威力発揮。

## 05 異常スコアを連続値で出したい

**▶ 異常検知系すべて / GMM / HDBSCAN**
「所属確率の低さ」「密度の低さ」が
自然な異常スコアになる。閾値調整可。

## 06 クラスタ密度がモードごとに違う

**▶ HDBSCAN / OPTICS / LOF**
密度の階層構造に対応。「全体異常 vs 局所異常」の
両方を捕捉する組合せ。

## 07 運転モードを確率的に解釈したい

**▶ ガウス混合モデル (GMM)**
「モードAに70%、Bに30%」のような確率的所属が
得られる。境界域の判定に有効。

## 08 少数の試作データを探索的に分析したい

**▶ 階層凝集 / OPTICS**
樹形図・リーチャビリティプロットでデータ構造を
視覚化できる。

▸ 次ページ ▸ 「目的・運用条件から選ぶ」「ベテラン判断との照合」も整理しています。

06 QUICK REFERENCE (2/3)

# こんな時はこのアルゴリズム②

目的と運用条件から逆引きする

### 09 「正常運転」を定義したい

**▶ GMM / k-means → ワンクラスSVM**
クラスタリングでモード抽出 → 異常検知で境界学習、の2段構えが定石。

### 10 「未知異常」を検出したい

**▶ アイソレーションフォレスト / HDBSCAN**
過去事例にない異常パターンを「外れ値」「ノイズ」として自動検出。

### 11 アラート閾値を現場で調整したい

**▶ アイソレーションフォレスト / GMM / MT法**
異常スコアが連続値で出るので、
過検知/見落としのバランスを現場で調整できる。

### 12 日本の品質工学に整合させたい

**▶ MT法（マハラノビス・タグチ）**
SN比・MD値など、タグチメソッドと親和性が高い。
社内規格・監査対応に強い。

### 13 不良モードを階層的に整理したい

**▶ 階層凝集 / 二分k-means**
「大分類→中分類→小分類」の階層構造を抽出。
不良モード辞書の構築に。

### 14 リアルタイム監視に組み込みたい

**▶ アイソレーションフォレスト / SGD-OCSVM / ミニバッチk-means**
逐次更新が可能で、
ストリーミングデータでも軽量に動作。

### 15 局所的な異常を見つけたい

**▶ 局所外れ値因子 (LOF)**
運転モードごとに正常値の幅が違うとき、
モード内での「異常」を捕捉できる唯一のモデル。

### 16 迷ったらまず何を試すか

**▶ アイソレーションフォレスト＋HDBSCAN**
異常検知＋クラスタリングの両面から眺めるのが定石。
SFIで両方一括実行。

▸ **重要** どの推奨も「絶対」ではありません。SFIで2〜3候補を横並びで比較し、
現場で最も納得感のあるものを選ぶのが定石です。

06 QUICK REFERENCE (3/3)

# 工程別・課題別レシピ集

「うちの工程ではどれから試すか」の現場別レシピ

## ⚡ 予知保全（設備異常検知）

振動・電流・温度・音響センサ

**異常検知系：アイソレーションフォレスト**（本命）／LOF（局所異常）
**クラスタ系：**HDBSCAN（未知異常）／GMM（正常モード定義）
**2段構え：**HDBSCANで正常クラスタ → ワンクラスSVMで境界学習

## 🛠️ 加工（運転モード抽出・異常検知）

送り速度・主軸回転・工具摩耗・振動

**大規模IoTデータ：SGDワンクラスSVM** / アイソレーションフォレスト
**クラスタ系：**ミニバッチk-means / BIRCH（運転モード抽出）
**複雑形状：**スペクトラル（振動波形特徴量が曲線分布する場合）

## ⚙️ 射出成形（不良モード分類・ロット品質）

温度・圧力・速度・保圧・サイクル

**運転モード抽出：**k-means / 二分k-means
**確率的判定：ベイズGMM**（モード数を自動決定）
**監査対応：**MT法（タグチ流で社内規格と整合）

## 🔍 品質管理（ロットばらつき・外れ値検出）

寸法・強度・色・厚みなど検査値

**外れロット検出：アイソレーションフォレスト** / MT法
**クラスタ系：**HDBSCAN / DBSCAN（密度の薄い領域の検査値を発見）
**少量データ：**階層凝集（試作・ベンチマーク段階）

## 🌐 大規模IoTデータ・リアルタイム監視

PLC・センサからの数百万行のストリーミングデータ

**本命：アイソレーションフォレスト** / SGDワンクラスSVM
**逐次クラスタ：**ミニバッチk-means / BIRCH（省メモリ）
**2段構え：**BIRCHで要約 → アイソレーションフォレストで詳細異常検知

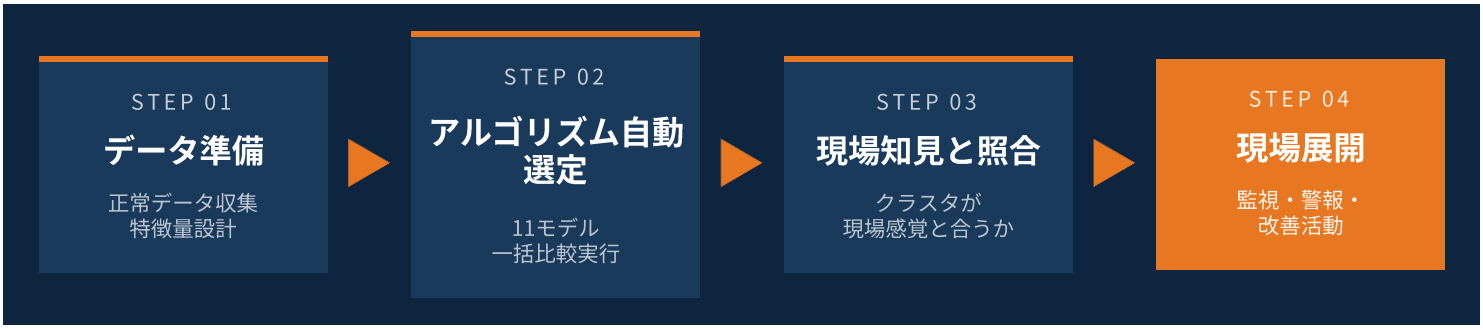
▸ **多変量検査・受入検査**

MT法（多変量を1つのMD値に集約）/ アイソレーションフォレスト（高次元データに強い）の組合せが定番。

07 SFI WORKFLOW (1/2)

# SFI活用の流れ

データ準備から現場展開まで




STEP 01

## データ準備

◆ **正常運転データの収集**（少なくとも数週間分）
◆ センサ値・PLCログ・帳票の統合
◆ 特徴量設計（生波形→統計量・FFT等）
◆ スケーリング（クラスタリング必須）
◆ 「明らかな異常」の事前除外

▸ **最重要：**クラスタリングは「教師なし」。
**特徴量設計の質**が結果の8割を決めます。

STEP 02

## アルゴリズム自動選定

◆ **16アルゴリズムを一括実行**
◆ シルエットスコア等の評価指標で比較
◆ クラスタ数の妥当性を複数指標で確認
◆ ノイズ点・外れ値の検出率を確認
◆ 候補3〜5モデルに絞り込み

▸ **SFIの本質的価値：**「迷ったら全部試せる」。
1モデルだけ選ぶより、横並び比較の意思決定の方が誠実です。

STEP 03

## 現場知見との照合

◆ **「このクラスタは何モード？」**を現場で命名
◆ ベテランの感覚と一致するか確認
◆ 一致しない場合：特徴量を見直し
◆ ノイズ点が「本当に異常か」を検証
◆ 「想定外のクラスタ」が見つかったら掘る

▸ **注意：**クラスタリングに「正解」はありません。
現場知見との突合せこそが、クラスタの意味付けを決めます。

STEP 04

## 現場展開

◆ 監視ダッシュボードへの組込み
◆ 異常スコアによる警報閾値設定
◆ 現場説明・OJT
◆ **「クラスタの追加・変化」のモニタリング**
◆ 定期的な再学習サイクルの設計

▸ **本質：**モデルは「作って終わり」ではなく、
現場の暗黙知を吸い上げ続ける仕組み。

▸ 次ページ SFIが「迷ったら全部試せる」プラットフォームである価値を、さらに掘り下げます。

07 SFI WORKFLOW (2/2)

# なぜ「迷ったら全部試せる」が価値か

クラスタリングこそ「複数手法の比較」が本質

▸ KEY MESSAGE

**クラスタリングには「正解」がない。**
**ならば、16個のアルゴリズムで切り取った**
**複数の視点を比べて、現場知見と照合する。**

## 01

### アルゴリズム依存の結果のブレを把握

k-meansとHDBSCANで結果が大きく違うなら、それは「データに固有の構造がない」サイン。
1モデルだけ見ると気付けない。

## 02

### 「異常」の多角的検証

複数アルゴリズムが「同じサンプルを異常と判定」したら、それは強い異常シグナル。
誤検知を減らす最良の手段。

## 03

### 試行錯誤の属人化を防ぐ

「あの担当者しか分からない選定理由」を排除。
再現性と引継ぎ可能な意思決定プロセスを残せる。

▸ ANTI-PATTERN —— SFIを使わない場合の典型的な失敗

① k-meansだけで運用 → 球形仮定で説明できない異常を見逃す
② クラスタ数を勘で決める → 同じデータで何度実行しても結果がブレる
③ ノイズ点を無視 → 「ノイズ＝異常の宝庫」を捨ててしまう

▸ SFIの存在意義

**アルゴリズムを「選ぶ」のではなく、「並べて比較する」。**
**その上で、現場知見と突き合わせて意思決定する。これがSFIの設計思想です。**

08 COMMON MISUNDERSTANDINGS (1/2)

# クラスタリングでよくある誤解

「教師なし学習」だからこそ陥りやすい落とし穴

## ✕ 「クラスタ＝意味のあるグループ」

▸ **なぜ問題か：**
アルゴリズムは数学的に「似ているもの」を集めるだけで、
それが現場にとって意味のある区別かは保証しません。
「データを3つに分けた」は事実ですが、その3つに名前を付けられなければ無価値です。
▸ **本質：**
クラスタリング結果は**「現場が意味付けして初めて完成」**します。
ベテランと一緒に「これは何モード？」を議論する工程が必須です。

## ✕ 「クラスタリングには『正解』がある」

▸ **なぜ問題か：**
教師なし学習なので、答え合わせができません。
同じデータでも、k-meansとHDBSCANでは違う結果が出ます。
どちらが「正解」かは、現場の目的によって変わります。
▸ **本質：**
シルエットスコアなどの数学的指標は**「参考値」**にすぎません。
最終判断は現場で「使える結果か」で決めます。

## ✕ 「k-meansだけ使っておけば十分」

▸ **なぜ問題か：**
k-meansは「球形クラスタ」を前提とした特殊なモデル。
製造業の異常予知では、振動波形・センサパターンは複雑な形状を持つことが多く、
k-meansでは「異常」を捉えきれません。
▸ **本質：**
異常予知ならまず**HDBSCAN**。k-meansは「正常モード抽出のベースライン」
として使い分けるのが正しい姿勢です。

## ✕ 「特徴量はそのまま放り込めばよい」

▸ **なぜ問題か：**
クラスタリングは「距離」に強く依存します。
温度（℃）と圧力（kPa）を生の値で投入すると、単位の大きい方ばかりが効いて結果がゆがみます。
▸ **本質：**
**スケーリング（標準化・正規化）**はクラスタリングの大前提。
「生波形 → FFT → ピーク周波数」のような特徴量変換も精度を大きく左右します。

▸ **続き** 次ページ：クラスタ数の決め方／ノイズの解釈／長期運用での落とし穴

08 COMMON MISUNDERSTANDINGS (2/2)

# 運用フェーズで起きる誤解

モデルは作って終わりではなく、運用が本番

### ✕ 「クラスタ数は経験で決めればよい」

▸ **なぜ問題か：**
「3個のはず」と決め打ちすると、本当は5個ある運転モードを無理矢理3個に
押し込めることになり、異常と正常の境界が崩れます。
逆に多すぎても1つのモードが分裂して「異常っぽい」と誤検知されます。
▸ **本質：**
エルボー法・シルエット法など複数指標で確認。
さらに**クラスタ数を自動決定するモデル（HDBSCAN/ベイズGMM）**と比較すれば、
人間の思い込みを補正できます。

### ✕ 「ノイズ点はゴミなので捨てる」

▸ **なぜ問題か：**
DBSCAN/HDBSCANが「ノイズ」と判定した点こそ、実は**異常予知では「最も貴重な情報」**です。
「いつもと違うパターン」を機械が見つけてくれた結果なのに、捨ててしまっては本末転倒。
▸ **本質：**
ノイズ点はすべて記録し、現場で**「本当に異常か」「単なる計測ミスか」**を一つひとつ精査する。
これが異常予知運用の中核です。

### ✕ 「一度学習したら、ずっと使える」

▸ **なぜ問題か：**
設備の経年劣化、材料ロットの変更、季節要因などで、**「正常」の定義自体が時間とともに変動**します。
1年前のモデルが今でも「正常」を正しく表しているとは限りません。
▸ **本質：**
定期的な再学習サイクル（月次・四半期）と、**「新しいクラスタが出現していないか」のモニタリング**
を運用設計に組み込むことが必須です。

### ✕ 「異常検知できれば、原因も分かる」

▸ **なぜ問題か：**
クラスタリングは「いつもと違う」を見つける仕組みであって、
「なぜいつもと違うのか」を答える仕組みではありません。
クラスタの所属を「原因」と取り違えると、改善打ち手を間違えます。
▸ **本質：**
異常検知はあくまで**「現象の発見」**。
原因分析は別工程（特徴量重要度・要因分析・実験計画）として明確に分けます。

▸ **道具はあくまで道具です。使い手の規律と現場感覚こそが、本質的な異常検知を作ります。**

09 SUMMARY

# まとめ

## アルゴリズムは道具。
## 意味付けは現場知見が決める。

16種類のアルゴリズム（異常検知5種＋クラスタリング11種）は、
どれも「正解」ではなく「特性の違う道具」です。
教師なし学習である以上、**クラスタに意味を与え、異常を解釈するのは現場知見の役割**。
それが製造業データ解析の本質です。

### ① 数値指標だけで選ばない

シルエットスコアは参考値。
最終判断は「現場で使える結果か」
「ベテランの感覚と合うか」で
決めます。

### ② 迷ったら全部試して比較

SFIは「16モデル一括実行・横並び比較」
のためのプラットフォーム。
これがSFIの存在意義です。

### ③ ノイズ点こそ異常の宝庫

DBSCAN/HDBSCANが「ノイズ」と
判定した点を、すべて記録し精査する。
これが運用の中核です。

CONTACT

## お問い合わせ

SFI導入・運用・データ活用についてのご相談は、Vertys（ヴァーティス）まで。
製造業のための、本質的に機能するデータ解析を。

S SFI
VERTYS Inc.

SFI 異常予知アルゴリズム解説資料
Vertys Inc. | 2026