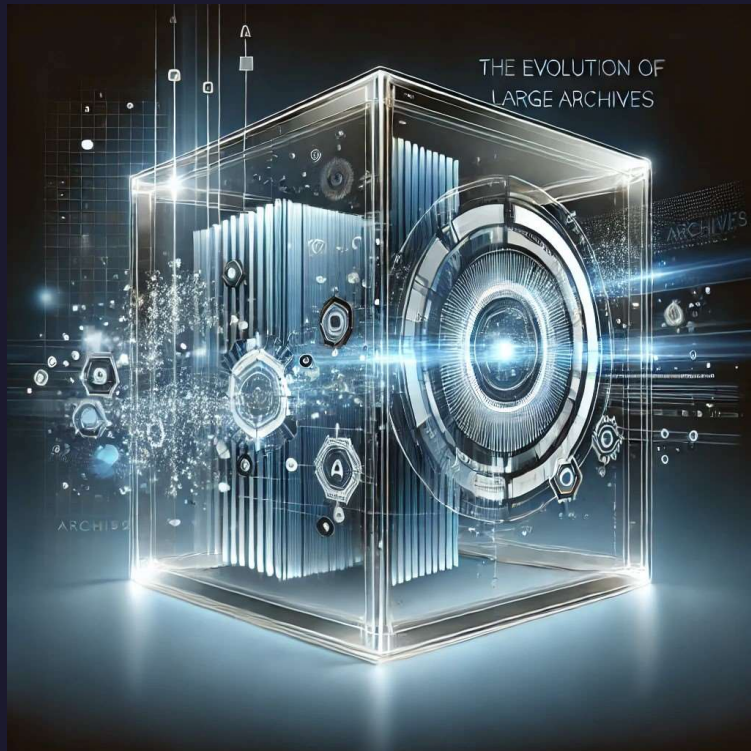


講演番号：1E-04



任意のカラムの組合せを比較基準にして行える,  
巨大表形式データ上の  
特異性類似レコードの高速検索法

2025.02.27

NNIテクノロジーズ株式会社

古庄 晋二

# はじめに

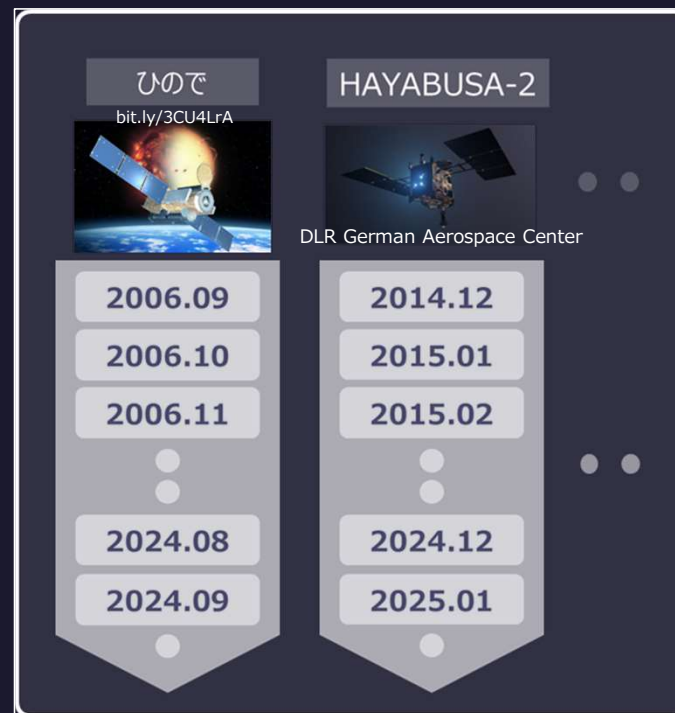
## 特異性類似レコードの検索 とは、

事故や故障、歩留まり低下

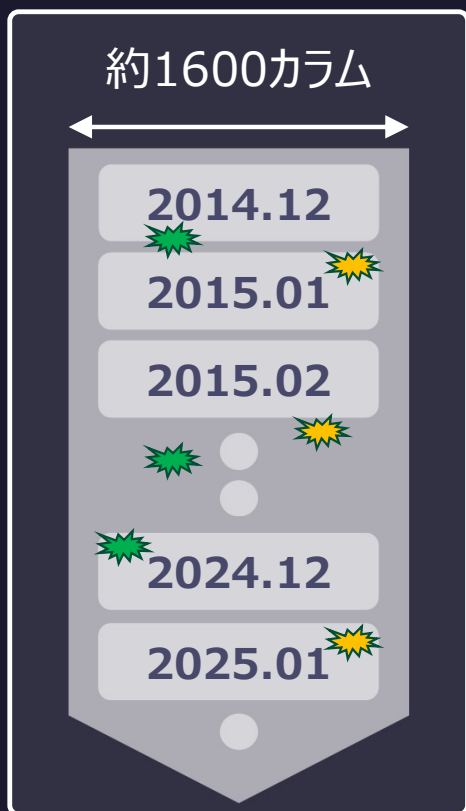
日々蓄積され巨大になる表形式データ：  
ビッグアーカイブ (BA) の検索

予兆や原因の究明

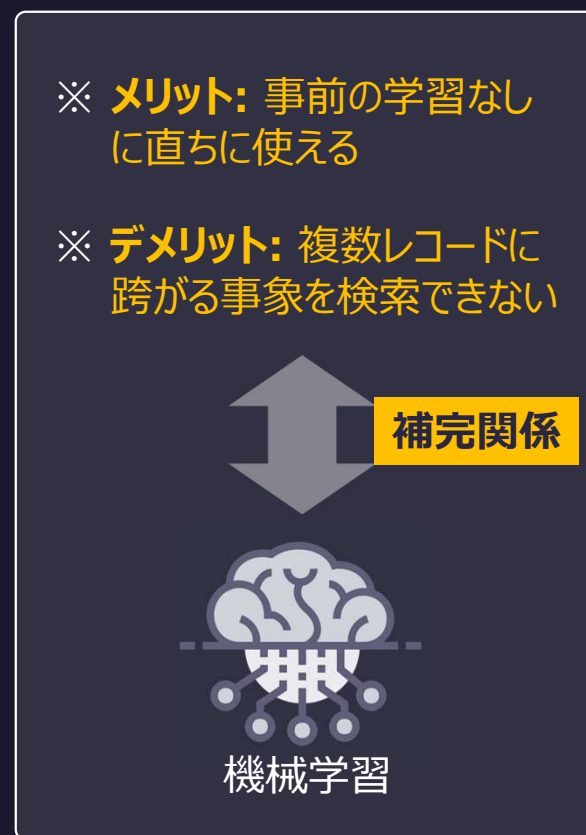
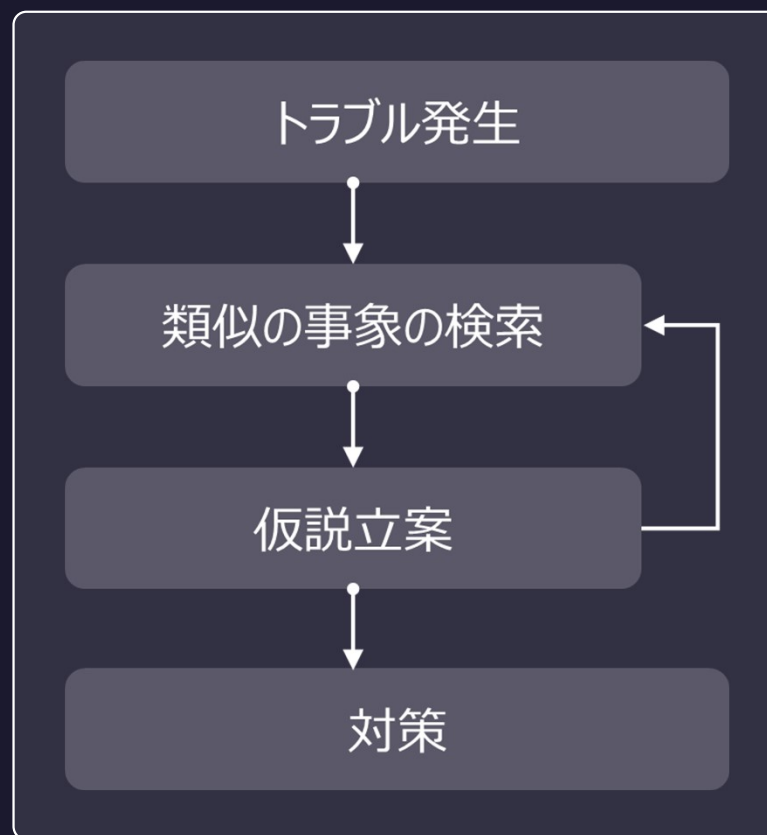
## ビッグアーカイブ (BA)



# 想定利用法



HAYABUSA2 HK-Data



# 特異性 類似度計量法に望まれる 5つの性質

## 1. 名義・順序・間隔・比例 の全尺度に対応

動作モード、装置の設定などの非数値も比較基準

## 2. 値の大小・スケーリングに影響されにくい、ロバスト性

Pa, dB などスケーリングが異なっても同じ検索結果が欲しい

## 3. 精度を低下させる、ノーマライズが不要

… ノーマライズは値の精度に重大なダメージ

## 4. 各カラムの類似度を加算できる、加算性

… 類似度の加算は難しい

## 5. 類似度のシャープネス、明瞭性

… 明瞭性がなければ検索は困難

# 既存の比較方法

## 候補1

Gower距離

1. 名義尺度 (Categorical Data)

- 計算方法: 一致すれば0, 一致しなければ1.
- $A_{ij}^{(1)} = 1$  ならば  $i$  と  $j$  の値が異なる.

2. 順序尺度 (Ordinal Data)

- 計算方法: ランクを正規化して距離を計算.
- ランク値を  $i$  と  $j$  にスケーリングし、その差を計算する.

3. 区間尺度 (Interval Data)

- 計算方法: 値の差を最大値と最小値で正規化.
- $A_{ij}^{(3)} = \frac{|x_i - x_j|}{\max(x) - \min(x)}$

4. 比率尺度 (Ratio Data)

- 計算方法: 区間尺度と同様.

Copyright © NNI Technologies, Co., Ltd. All Rights Reserved. 17

17P に Jump

## 候補2

分位数

原データの乱数	ソート後
0.178	0.0
1.360	1.21
2.374	2.22
3.236	3.33
...	...
...	22.242
...	23.244 (22%分位数)
...	24.264
...	...
...	46.428 (44%分位数)
...	47.444 (47%分位数)
...	...
98.980	...
99.844	99.995

ソートを行う

順位の開きを評価

Copyright © NNI Technologies, Co., Ltd. All Rights Reserved. 18

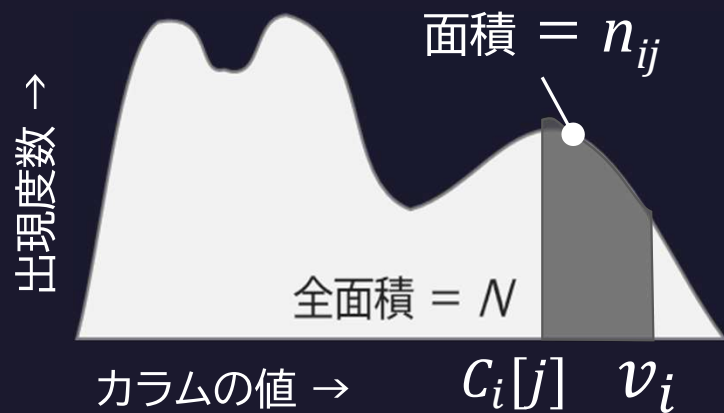
18P に Jump

※ COS類似度などの、  
・ 数値しか使えない、  
・ 固定の組合せしか使えない  
方法は不可

1. [最大値-最小値] による正規化  
外れ値の影響大
2. 加算の正当性 に疑問
3. 類似度の明瞭性がない

1. 加算の正当性 に疑問
2. 類似度の明瞭性がない
3. BA ではソートのコストが高すぎる

# 自然数化された分布情報を使った 類似度 の提案



1. 値を順位に変換（自然数化）して使う。
2. そのため、順序～比例尺度のいずれも区別無く対応でき、**各種の尺度に対応**できる。 **…性質1**
3. 値の大小、スケーリングに影響されず、**ロバスト**になる。 **…性質2**
4. 同じく、**ノーマライズも不要**になる。 **…性質3**

**…性質3**

$v_i$ : カラム  $i$  の目標値  
 $C_i[j]$ : カラム  $i$  の  $j$  番目のセルの値  
 $n_{ij}$ :  $v_i$  と  $C_i[j]$  の間に出現する値の出現度数

$s_{ij}$ : カラム  $i$  の  $j$  番目のセルと目標値との類似度

$$s_{ij} = -\log_2 \left( \frac{n_{ij}}{N} \right) \text{ (bit)} \quad \dots \text{ 自己情報量}$$

$s_j$ :  $j$  番目のレコードと目標値との類似度

$$s_j = \sum_i s_{ij} \text{ (bit)} \quad \dots \text{ 情報量の加算}$$

**…性質4**

# なぜBAの 分布情報を全て 使おうとしているのか

BA は巨大であるために、その分布は信頼できる。

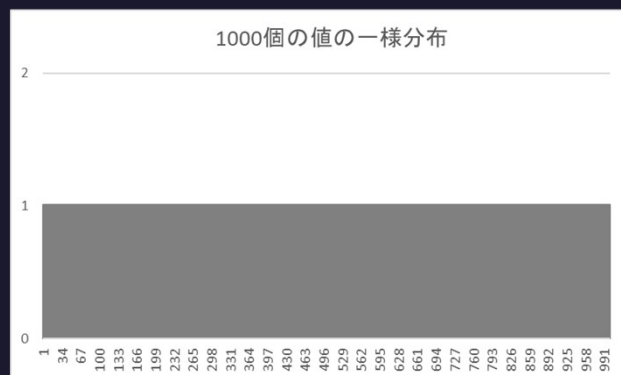


そのため、分布情報を精密に扱えば、特異性が検索可能に。



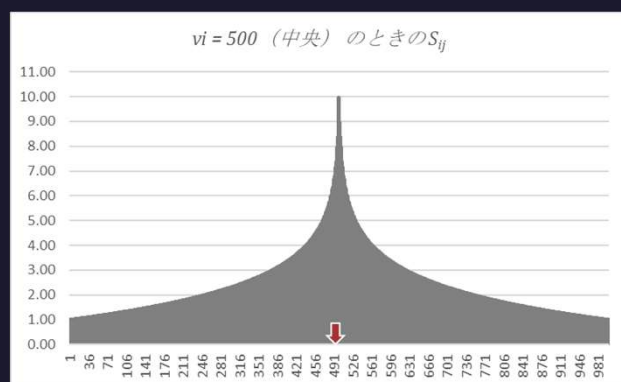
※ ただし、分布情報を精密に扱おうとすると膨大な計算量が必要になる。  
後に、それがどのように効率化出来るか、も説明する。

# 類似度 $s_{ij}$ の性質



1. 類似度  $s_{ij}$  は  $v_i$  で単一のピークを持つ。

2.  $v_i$  付近の値の重複が少なければ、ピークは鋭いものになる。



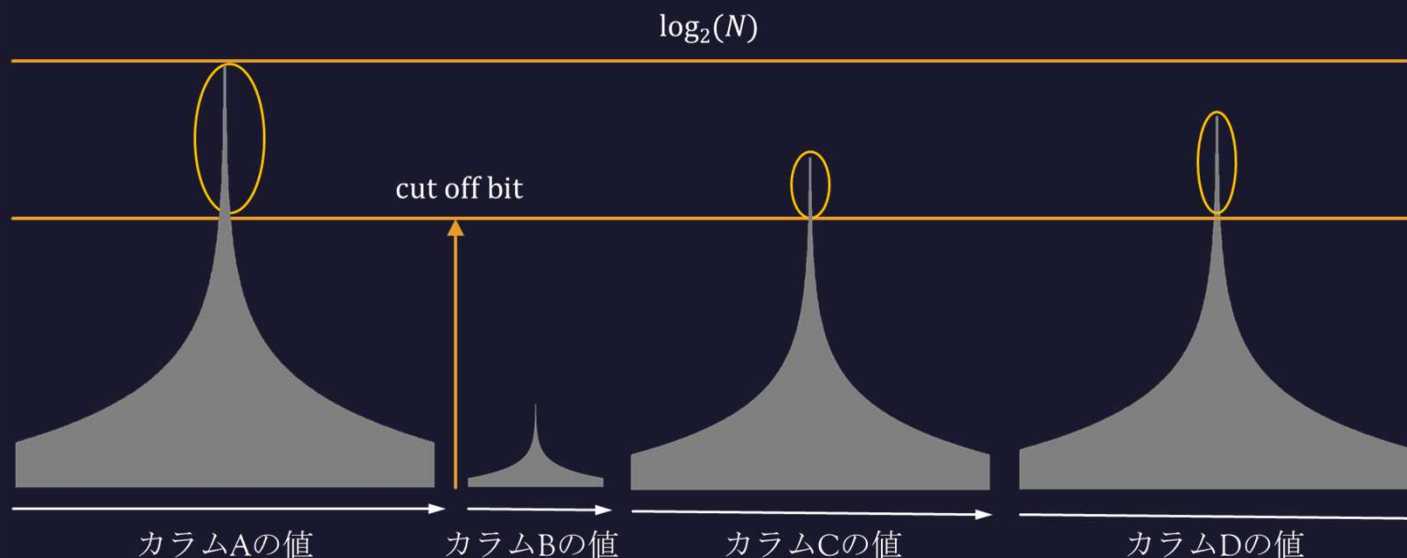
明瞭性

…性質5



# 類似レコードの高速検索手順

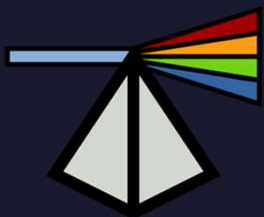
1. 各カラム毎に cut off bit よりも類似度の高いレコード番号リストを抽出する。
2. 上記で各カラム毎に得られたレコード番号リストの OR を取り、重複を無くす。
3. レコード番号リストのレコード全てについて、レコードとしての類似度を算出する。
4. レコードとしての類似度の最も高いものから順番に必要な個数<sup>\*1</sup>のレコードを取得する。



# 5つの性質を満たす $S_{ij}$

1. 値の順位で比較でき、全尺度に対応できる
2. 値の順位で比較でき、ロバスト性を備える
3. 値の順位で比較でき、ノーマライズが不要
4. 自己情報量であるため、加算性がある
5. 類似度がシャープで、明瞭性がある

# 自然数インデックス (NNI)

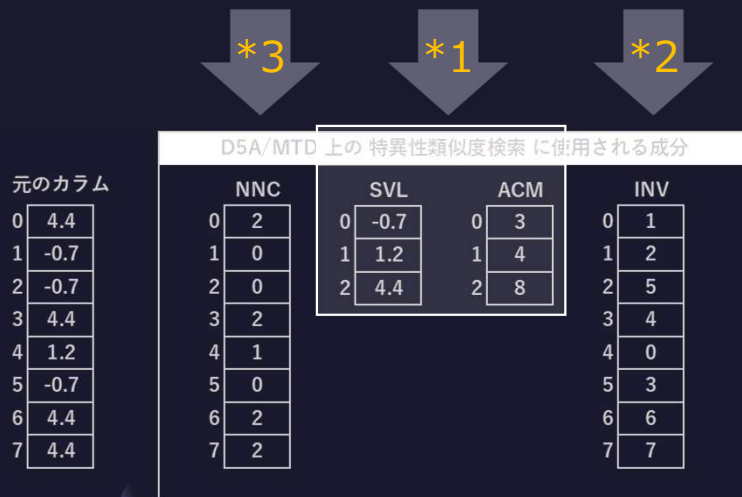


表形式データは、さまざまな **成分** に分解できる。  
分解すると、どのカラムも同じ構造になる。

成分を組み合わせて、高速なアルゴリズム群を作れる。  
このアプローチを **自然数インデックス (NNI)** と呼ぶ。

成分はどのカラムでも同じになるため、  
**全てのカラム、全てのカラムの組合せ、全ての部分集合**  
の高速化に寄与できる。

# NNIによる検索高速化



**NNC:** Natural Numbered Column

元のカラムの値を SVL 上の位置に置き換えたもの

**SVL:** Sorted Value List

カラムの出現値を昇順・ユニークにしたリスト

**ACM:** Accumulated occurrence

SVL上の値の累積度数

**INV:** Inverted record number

転置レコード番号

\* 1.  $SVL + ACM$  で  $n_{ij}$  を簡単に求められる。

+

\* 2.  $INV$  で **ピーク付近のレコード番号**を簡単に検索できる。

+

\* 3.  $NNC$  で **レコード番号から  $n_{ij}$**  を簡単に求められる。

# 実測-1 TOPIX4

## 目的:

- Cut off bit をどうとるか、
- 検索結果の上位 x % を採用するか、  
上記を変化させると、抽出すべきレコードの  
何% を得られるか？ を計測する。

## 用いたデータ:

TOPIX4(1989~2020)  
全 7,744 (12.919 bit) レコード  
5カラム(日付、{始・高・低・終}値)

## 結論:

- Cut Off:  $\log_2(N)$ -6.0 bit、
- 検索結果の上位 50% を採用すれば  
100件以上の信頼できる結果を得られる。

全レコードそれぞれに対し、特異性類似レコードを検索した

Cut Off Bit	上位10%	上位20%	上位30%	上位40%	上位50%	上位60%	上位70%	上位80%	上位90%	上位100%
12.0	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%
11.0	82.75%	72.96%	61.55%	53.79%	46.95%	40.74%	35.67%	28.65%	23.32%	15.54%
10.0	95.65%	90.60%	85.14%	79.95%	74.17%	67.70%	60.38%	52.21%	42.71%	26.16%
9.0	99.38%	98.18%	96.60%	94.20%	91.06%	86.68%	80.86%	73.19%	62.52%	37.64%
8.0	99.93%	99.77%	99.51%	98.97%	97.99%	96.21%	93.18%	88.15%	79.26%	47.90%
7.0	100.00%	99.99%	99.96%	99.88%	99.74%	99.42%	98.61%	96.52%	91.07%	56.22%
6.0	100.00%	100.00%	100.00%	99.99%	99.98%	99.95%	99.86%	99.46%	97.12%	63.08%
5.0	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	99.99%	99.96%	99.42%	69.86%
4.0	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	99.99%	99.96%	77.76%
3.0	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	86.35%
2.0	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	92.74%
1.0	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	97.81%
0.0	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%
	100.00%	99.98%	95.97%	90.94%	80.89%	70.79%	60.69%	0.49%		

得られた特異性類似レコードの数

Cut Off Bit	上位10%	上位20%	上位30%	上位40%	上位50%	上位60%	上位70%	上位80%	上位90%	上位100%
12.0	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
11.0	1.82	2.97	4.57	6.16	7.74	9.37	10.72	12.56	13.93	15.53
10.0	4.06	8.28	12.48	16.55	20.70	24.84	29.02	33.11	37.23	41.39
9.0	8.28	16.54	24.82	33.10	41.38	49.65	57.92	66.20	74.47	82.75
8.0	14.03	28.06	42.09	56.11	70.15	84.17	98.20	112.22	126.26	140.28
7.0	22.30	44.60	66.90	89.20	111.51	133.80	156.10	178.40	200.70	223.00
6.0	35.56	71.12	106.68	142.24	177.80	213.36	248.93	284.49	320.04	355.60
5.0	59.54	119.07	178.60	238.14	297.67	357.20	416.74	476.27	535.81	595.34
4.0	105.78	211.55	317.32	423.10	528.88	634.66	740.43	846.20	951.98	1057.76
3.0	193.27	386.54	579.82	773.09	966.35	1159.63	1352.91	1546.18	1739.44	1932.72
2.0	349.26	698.51	1047.77	1397.03	1746.29	2095.55	2444.81	2794.07	3143.33	3492.58
1.0	587.44	1174.90	1762.34	2349.79	2937.24	3524.68	4112.13	4699.58	5287.03	5874.47
0.0	774.00	1548.00	2322.00	3098.00	3872.00	4646.00	5421.00	6195.00	6970.00	7744.00

# 実測-2 ひので

## 太陽観測衛星「ひので」(SOLAR-B)

[https://www.jaxa.jp/projects/sas/solar\\_b/index\\_j.html](https://www.jaxa.jp/projects/sas/solar_b/index_j.html)



(画像)

[https://www.jaxa.jp/projects/sas/solar\\_b/images/solar\\_b\\_main\\_001.jpg](https://www.jaxa.jp/projects/sas/solar_b/images/solar_b_main_001.jpg)

太陽観測衛星「ひので」のハウスキーピングデータ (HKデータ) を使ったデモ。

HK2 : 396カラム、約2.9億レコードを使い、特異性類似レコードの検索を実行。

all.d5a (286,721,092)

File 編集 操作 調査 移動

2_ERROR_ANC	HK2_GAS_TEMI	HK2_GAS_X	HK2_GAS_X_SF	HK2_GAS_Y	HK2_GAS_Y_SF	HK2_GAS_Z	HK2_GAS_Z_SF	K2
1	0	117.891	-5	-60000	-5	-60000	-5	-60000
2								-60000
3								-60000
4								-60000
5								-60000
6								-60000
7								-60000
8								-60000
9								-60000
10								-60000
11								-60000
12								-60000
13								-60000
14								-60000
15								-60000
16								-60000
17								-60000
18								-60000
19	0	117.891	-5	-60000	-5	-60000	-5	-60000
20	0	117.891	-5	-60000	-5	-60000	-5	-60000

\*\*\* 類似レコード検索 \*\*\*

\*\*\* 類似レコード検索 \*\*\*

record count:286,721,092 max bit: 28.095073

カラム:[HK2\_GAS\_Y] Vi:[-4.960784312337637] peak\_bit:[ 28.095073]

カラム:[HK2\_GAS\_Z] Vi:[-4.882352828979492] peak\_bit:[ 25.287718]

カラム:[HK2\_GAS\_X] Vi:[-3.588235378265381] peak\_bit:[ 28.095073]

カットオフビット数: 24.0

Hit count:[1] カラム:[HK2\_GAS\_Y] Vi:[-4.960784312337637] peak\_bit:[ 28.095073]

Hit count:[8] カラム:[HK2\_GAS\_Z] Vi:[-4.882352828979492] peak\_bit:[ 25.287718]

Hit count:[12] カラム:[HK2\_GAS\_X] Vi:[-3.588235378265381] peak\_bit:[ 28.095073]

[3] 個のカラムの検索結果を統合します

Merged Count [20]

Elapsed time: 4.361851 msec

1.rec:[237,651,593], Sj:74.999816

2.rec:[295], Sj:45.417814

3.rec:[296], Sj:45.417814

4.rec:[297], Sj:45.417814

5.rec:[298], Sj:45.417814

6.rec:[299], Sj:45.417814

7.rec:[300], Sj:45.417814

8.rec:[301], Sj:45.417814

9.rec:[10,846], Sj:30.067800

10.rec:[10,846], Sj:30.067800

クリア 設定表示 カットオフビット数 24 実行 自動計算

00:18.17

# まとめ

1. 自然数化された分布情報を使った類似度により、特異性類似度に対する5つの望ましい性質を実現した。
2. 類似度の高いレコードを各カラム単独で検索し、それらを合併し、それらの中から、類似度の高いもの上位を検索結果とする。
3. 上記の検索は 自然数インデックス(NNI)によって効率的に行える。
4. 株価データによってその有用性が確認された。

5. 提案した特異性類似レコードの検索方法は機械学習を補完できる。

以上で、特異性類似レコードの検索で、事故や故障、などの特異な事象を、ビッグアーカイブから効率よく検索できることが分かった。

ビッグアーカイブ（BA）を活用した故障予知や故障の原因究明などの支援が可能になる。

# Thank you

NNIテクノロジーズ株式会社

古庄 晋二

[shinji.furusho@nni-tech.com](mailto:shinji.furusho@nni-tech.com)

[www.nni-tech.com](http://www.nni-tech.com)





# Gower距離

## 1. 名義尺度 (Categorical Data)

- 計算方法: 一致すれば 0、一致しなければ 1。
- $s_{ij}^{(k)} = 1$  ならば  $i$  と  $j$  の値が異なる。

## 2. 順序尺度 (Ordinal Data)

- 計算方法: ランクを正規化して距離を計算。
- ランク値を  $[0, 1]$  にスケールリングし、その差を計算する。

## 3. 間隔尺度 (Interval Data)

- 計算方法: 値の差を最大値と最小値で正規化。
- $$s_{ij}^{(k)} = \frac{|x_i^{(k)} - x_j^{(k)}|}{\max(x^{(k)}) - \min(x^{(k)})}$$

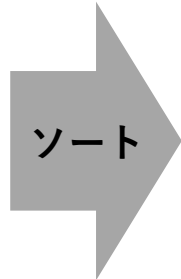
## 4. 比例尺度 (Ratio Data)

- 計算方法: 間隔尺度と同様。

# 分位数

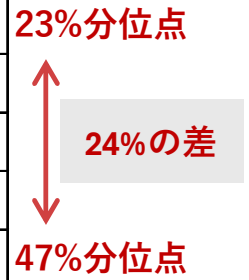
百個の乱数

0	78
1	960
2	974
3	246
...	...
...	...
...	...
...	...
...	...
...	...
...	...
...	...
...	...
98	990
99	444



ソート後

0	0
1	21
2	22
3	33
...	...
...	...
22	241
23	246
24	264
...	...
46	430
47	444
48	455
...	...
99	991



ソートを行う

順位の開き を評価