2014.08.07

|  |  |
| --- | --- |
| **Feature (特徵)** | |
| C5  C6 | C5、C6基本上是同一條曲線，在多數情況下用**MI演算法運算皆得到第一名**，但用原始數據直接與dp\_filter作圖並看不出任何關連。細究原因**明顯受到是否對cycle離散化影響**。從目前結果看來，此兩個變數要對cycle做離散化/正規化後，**相似度極高，但實際上可能無用**。詳見後面-觀察分析1。 |
| P21 | P21\_max**在Regression-Based中表現不錯**，且實際畫圖觀察也確實十分相似，可能需要多研究MI與Regression的差異，未來對於其他變數才能更精準地分析。 |
|  |  |
| Others | 暫無發現顯著關係。 |
|  |  |

**MI-Based Algorithm**

|  |  |
| --- | --- |
| **目標值離散化方式** (目前設 dp\_filter\_max) | |
| manual\_cut  手動固定離散化 | 目前設定用(5,15,20)把 dp\_filter\_max切分出4個等級，與原始資料對照大致符合。但由於固定數值，不會受資料切割的設定而變化，像是在 noSeg 與 seg4\_2中的MI排名變動極大，在seg4\_2中已經把C6\_max與P21參數剔除前五名。故推論**手動固定離散方式不適用於切割過的資料**。 |
| ew\_cycle  自動等分離散化 | 自動依照 cycle做equal-weight離散化。在noSeg情況下結果與手動切割基本上相同，而隨segmentation增加，排名結果僅些微改變，猜測**自動等分離散化一致性高**。不過目前離散間隔的計算是以interval = (max-min)/N，容易受max影響而浮動，不見得有效離散化，故未來應設法改進。 |

**Regression-Based Algorithm**

**觀察分析1 – MI, ew\_cycle=10, C5, C6**

MI：ew\_cycle=10, seg10\_1, 排名第一(100%) C5\_seg9\_max

MI：ew\_cycle=10, seg10\_2, 排名第七(48.33%) C5\_max

MI：ew\_cycle=10, seg10\_2, 排名第一(100%) C6\_min

MI：ew\_cycle=10, seg10\_2, 排名未列入 C6\_mean

這四張圖看起來幾乎是一模一樣，但排名迥異。

MI：ew\_cycle=10, seg10\_2, 排名第二(74.26%) C5\_skewness

C5\_skewness雖然排名第二，但其實看不出任何關聯。

回到原始數據來看，C5、C6基本上是同一條曲線。

因為目前feature是以每個cycle的max, min做分等依據，故造成相似度很高的錯覺，實際原始資料毫無周期性變化。排名是相對的，且僅供參考，若將不適當的feature放入演算法一起跑，會造成結果錯亂。然後是否對cycle做離散化也有極大的影響。

**觀察分析2 – Regression, P21**

首先觀察原始資料

Regs: noSeg

Regs：seg10\_1

Regs: seg10\_2

觀察到P21\_max 變數在noSeg中表現與P21\_seg9\_max幾乎相同，而seg10\_2的切割方式看起來比原本更差。顯該變數主要是受到後段影響，提高segmentation卻用直擺細切的方式，反而是有害無益的。