有機合成に最適な酵素候補提示のための特徴量エンジニアリングによる EC 番号予測

EC Number Prediction Using Feature Engineering to Present Optimal Enzyme Candidates in Organic Synthesis

武藤 克弥 (Katsuya Muto)
u255018@st.pu-toyama.ac.jp

富山県立大学大学院 電子・情報工学専攻 情報基盤工学部門

January 23, 2024

1. はじめに

1.1 研究背景

有機合成化学において，生体触媒の効率性や環境面から化学反応の設計に酵素を生体触媒として利用される機会が増加している．酵素は EC 番号によって分類されており，代謝経路の解析や新たな酵素反応設計のため，機械学習で EC 番号を予測し，酵素の性質を特定する研究が行われている．

1.2 本研究の目的

有機合成に用いる酵素を探索する実験コストや時間削減のため，化学反応に最適な酵素候補を EC 番号として予測できる EC 番号予測手法を開発する．

1. 代謝経路の解析 = 生体の機能の解明

未知のタンパク質配列
[MAKLLLLIFGVFIFVNSQAQTFPTILEKHN・・・]

どんな性質か知る
時間　大
コスト　大
→

まず大まかに知りたい
?

2. 新たな化合物の設計 = 医薬品など

酵素(生体触媒)
効率よく反応
環境にやさしい

A + B → C

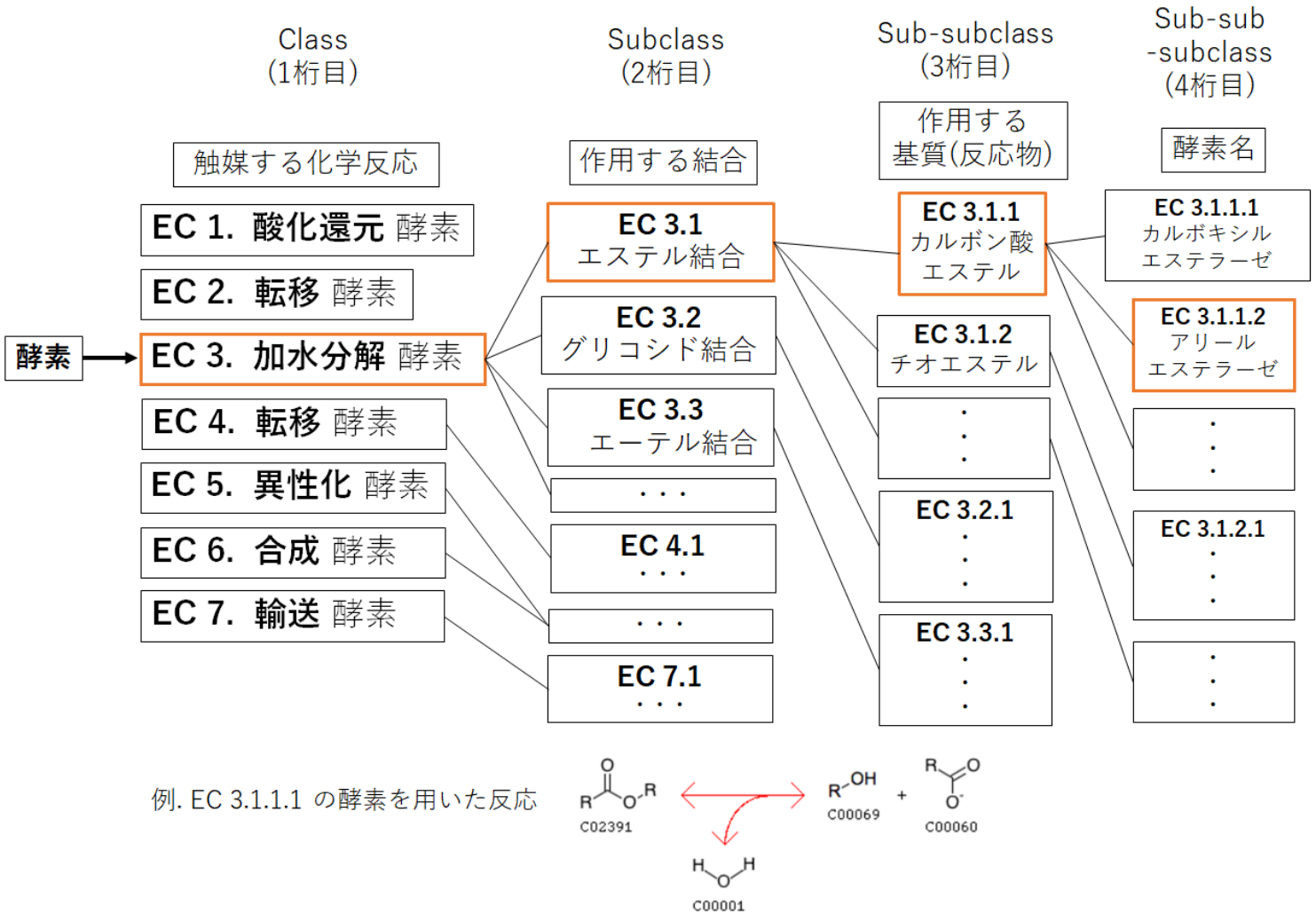
どの酵素最適か？
時間　大
コスト　大
→

候補絞りたい
?

2. 酵素と EC 番号

酵素を 4 組の数字 (EC ○. ○. ○. ○) の組み合わせで分類したもの.

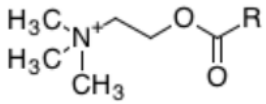
EC 番号の機械学習予測 = 酵素候補の絞り込み



3.1 化合物の構造表現法

計算機上で化学反応を表現する各種方法

構造式

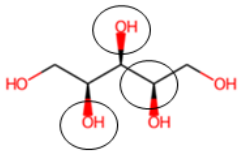


機械学習データ

SMILES *C(=O)OCC[N+](C)(C)C

フィンガープリント

100001011・・・ ＝



→官能基の有無を判定
2値のバイナリベクトル

物理・化学的特性値

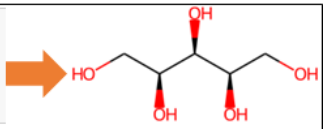
・分子量，電荷，疎水性など

・化合物A: (100, 8.32, -0.23, ・・・)
→特性値のベクトルで化学反応を表現

RDKit

化学構造を扱えるPythonライブラリ
・ 208種類の特性値(記述子)あり

```
from rdkit import Chem
Xylitol = Chem.MolFromMolFile('Xylitol.mol')
Xylitol
```



```
from rdkit.Chem import Descriptors
print("SMILES: " + Chem.MolToSmiles(Xylitol))
print("分子量: " + str(Descriptors.MolWt(Xylitol)))
print("LogP: "+ str(Descriptors.MolLogP(Xylitol)))
print("TPSA: "+ str(Descriptors.TPSA(Xylitol)))
```

```
SMILES: OC[C@H](O)[C@@H](O)[C@H](O)CO
分子量: 152.14600000000002
LogP: -2.9462999999999995
TPSA: 101.15
```

3.2 EC 番号予測手法 [123]

EC 番号予測の目的

酵素探索の短縮:：既存データで学習＆予測精度向上→ (将来的) 未知データ適用

(1)タンパク質配列[1]　**予測範囲：1桁目～4桁目**

画像処理(CNN),自然言語処理(Transformer)ベースの予測

[MAKLLLNSQAQTFPTILEKHN・・・]

→1. 代謝経路の解析

(2)化合物の物理・化学的特性値[2]
分子量，電荷，疎水性など

予測範囲
1桁目～3桁目
データ数小

(3)差分反応フィンガープリント[3]
SMILESをハッシュ化
→2048次元のバイナリベクトル

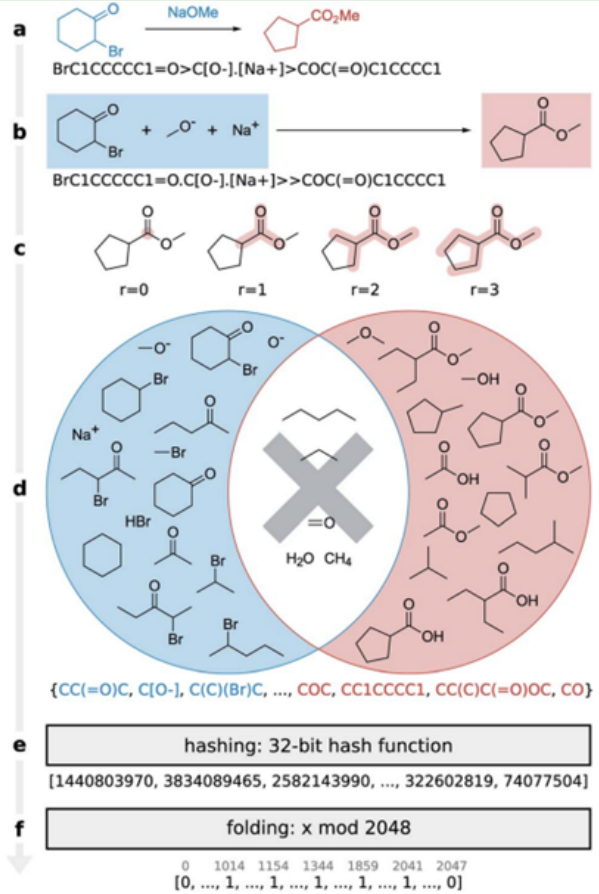
予測範囲
1桁目～3桁目
データ数大

EC 番号反応式： 反応物 1 + 反応物 2 ⟷ 生成物 1 + 生成物 2

$$\rightarrow RFP = FP_{\text{生成物 1+生成物 2}} - FP_{\text{反応物 1+反応物 2}}$$

反応物→生成物の変化
=フィンガープリントの変化

2.酵素を用いた化学反応の設計 =**有機合成目線**



SMILESの2値ベクトル化[3]

[1]Naoki Watanabe et al., 2023.
[2]Diogo A. R. S. Latino et al., 2009.
[3]Daniel Probstl., 2023.

4.1 提案手法の概要

RDKit 特性値を用いた EC 番号予測

反応物から生成物に変化するときの 208 種類の特性値変化量を計算

特性値(RDKit)　分子量，電荷，疎水性　官能基1，官能基2，官能基3

化合物　A = (100, 8.32, -0.23, ・・・, 1, 0, 0, ・・・)

208種

B = (99, 9.32, -6.23, ・・・, 0, 0, 0, ・・・)

C = (100, 8.32, -0.23, ・・・)

D = (89, 7.32, 0, ・・・)

反応式：A + B ⟶ C + D

特徴ベクトル：(A + B) - (C + D)

= (10, 2, -6.23, ・・・, 1, 0, -1, ・・・)

208次元

有意性

(2)特性値ベース

+

(3)フィンガープリント(一部)の組み合わせ

(3)との比較

1~3桁目の予測(同様)

+

4桁目の予測(卒論)

→4桁目の予測まで実施

RDKit特性値 →

特徴ベクトル↓

	MaxEStateIndex	MinEStateIndex	MinAbsEStateIndex	qed	
4.1.1.74	-7.449074	-2.629630	-0.064815	-0.360209	-1.
1.2.1.8	-0.197403	1.307870	0.405116	-0.079826	0.
2.5.1.85	0.593569	0.196239	-2.312624	0.488055	-4.
1.4.1.4	0.234718	-0.413194	0.418052	0.389325	0.
1.1.1.3	-0.155930	-0.317778	0.059255	0.016389	-2.
...	...	...	...	...	
4.4.1.13	-3.236897	-0.282721	-1.272102	-0.270358	5
2.3.1.-	-0.286151	0.039395	0.454936	-0.386083	0.
2.3.1.57	-0.286151	0.039395	0.454936	-0.386083	0.

RDKit 記述子の中から必要なものだけを選択する

ラッパー法による記述子選択 (SequentialFeatureSelector(SFS)[4])

分類モデルの予測精度を評価し，最高評価となる組み合わせの記述子を選択

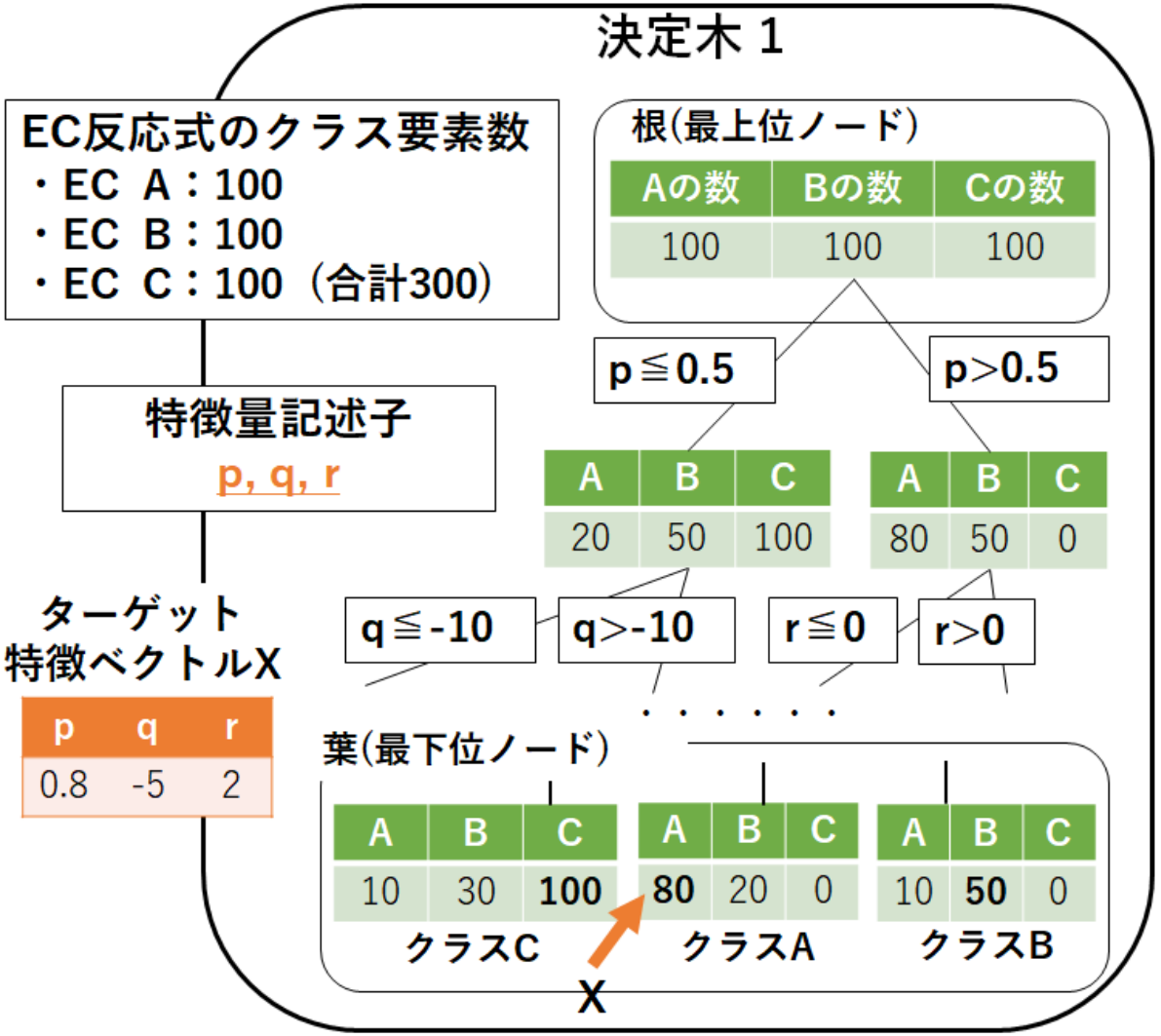
用いる手法: Step Forward 法

1. n 個の記述子から 1 つ選択し，n 種類の分類モデルを作成
2. 最も評価の高いモデルに用いられている，記述子を選択
3. $n-1$ 個の記述子から 1 つ選択し，先ほど選択されたモデルに追加することで，新たな分類モデルを作成
4. $n-1$ 個のモデルで最も評価の高いものに用いられている，記述子の組み合わせを選択
5. 指定した特徴数になるまで 3 と 4 を繰り返す.

モデルの評価基準： F1 スコア平均 (層化 5 分割交差検証)

[4] Mlxtend.feature selection, http://rasbt.github.io/mlxtend/api_subpackages/mlxtend.feature_selection/

4.3 提案手法 ランダムフォレスト (RF) による EC 番号分類 (1)



4.3 提案手法ランダムフォレスト (RF) による EC 番号分類 (2)

各ノードで IG が最大となるように，特徴量 f とデータを分割する閾値を決定

$$IG(D_P, f) = I_{imp}(D_p) - \frac{N_{left}}{N_p} I_{imp}(D_{left}) - \frac{N_{right}}{N_p} I_{imp}(D_{right})$$

D_p：上位ノード内のデータ (特徴ベクトル)

f：分割時に用いられる特徴量

$I_{imp}(t)$：ジニ不純度

D_{left}，D_{right}：分割先の下位ノード内のデータ

N_p，N_{left}，N_{right}：上位ノード，下位 (左右) ノードのデータ数

$$I_{imp}(t) = \sum_{i=1}^{c} p(i|t)(1 - p(i|t)) = 1 - \sum_{i=1}^{c} p(i|t)^2$$

$p(i|t)$：ノード t のクラス i のデータ割合，c：ノード t 内のクラス数

パラメータ調整(グリッドサーチ)→各決定木の最大深さ、決定木数

5.1 数値実験の概要

10〜18p は卒論+修士中間までの内容

数値実験の流れ

1. KEGG の (EC 番号 x 反応式) データで RF モデル作成
→特徴量エンジニアリング (記述子選択，RF パラメータ調整)
2. KEGG テストデータに対してモデルの精度予測
3. BRENDA や文献の (EC 番号 x 反応式) データにする予測

データベース (KEGG と BRENDA) の違い [5]

KEGG: 天然の酵素反応中心
BRENDA: 天然だけでなく非天然 (人工基質) も多く含む

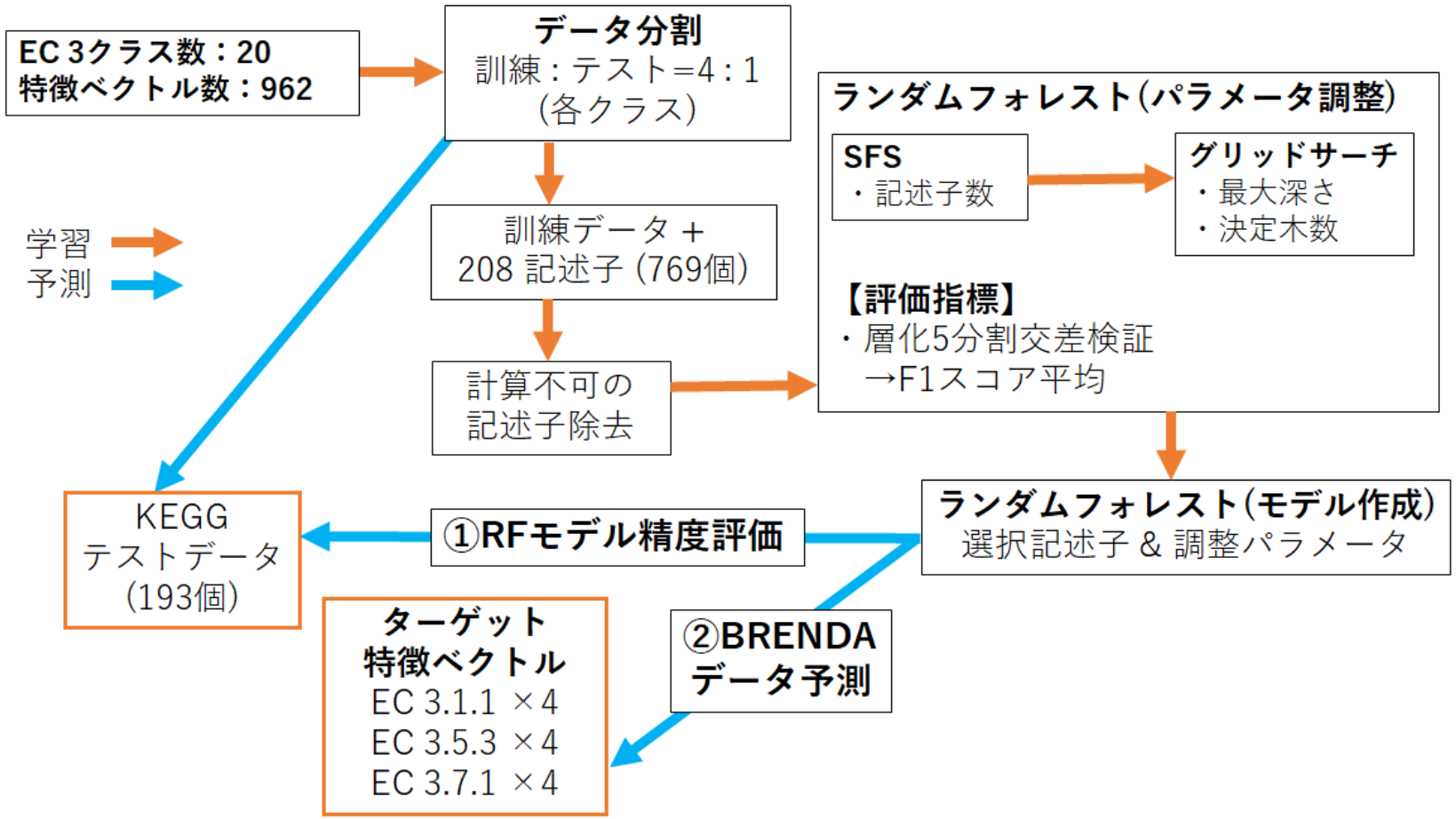
有機合成では非天然の化合物も使用
→ BRENDA(文献) の反応に対する予測が重要

[5] 荒木 通啓, 2014

5.1 数値実験の概要

モデル作成・予測手順

EC3 クラスに限定 & 2・3 桁目までを分類予測



5.1 数値実験の概要

化学反応データの取得と整形

KEGG より EC3 クラスの構造ファイルを取得し，962 個 ×128 記述子の特徴ベクトルを作成

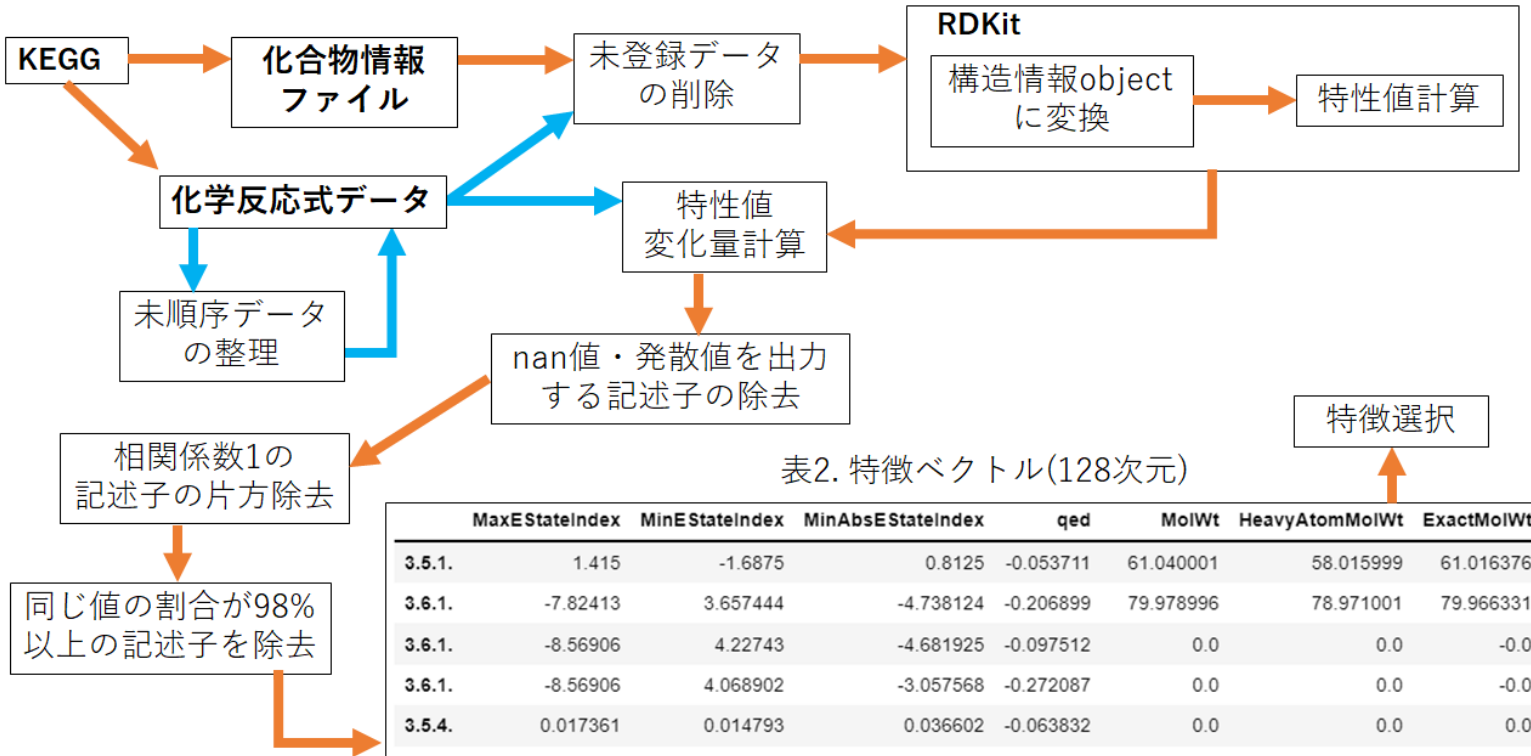


表2. 特徴ベクトル(128次元)

	MaxEStateIndex	MinEStateIndex	MinAbsEStateIndex	qed	MolWt	HeavyAtomMolWt	ExactMolWt	NumValer
3.5.1.	1.415	-1.6875	0.8125	-0.053711	61.040001	58.015999	61.016376	
3.6.1.	-7.82413	3.657444	-4.738124	-0.206899	79.978996	78.971001	79.966331	
3.6.1.	-8.56906	4.22743	-4.681925	-0.097512	0.0	0.0	-0.0	
3.6.1.	-8.56906	4.068902	-3.057568	-0.272087	0.0	0.0	-0.0	
3.5.4.	0.017361	0.014793	0.036602	-0.063832	0.0	0.0	0.0	
...	...	...	...	...	...	...	...	
3.1.1.	-6.828515	-0.668523	-1.021632	-0.087708	0.0	0.0	0.0	
3.2.1.	-8.720595	1.384253	-0.518534	-0.069449	0.0	0.0	-0.0	
3.2.1.	-8.708548	1.381274	-0.495988	-0.069449	0.0	0.0	-0.0	

5.1 数値実験の概要

EC 3 クラスのデータ内訳

5-fold 交差検証を行うため，反応式 (特徴ベクトル) 数が 6 以上のクラス 20 種を使用

クラス名	反応式数	クラス名	反応式数	クラス名	反応式数
3.1.1	122	3.2.2	24	3.5.5	12
3.1.2	59	3.3.2	6	3.5.99	11
3.1.3	152	3.4.13	6	3.6.1	94
3.1.4	29	3.4.19	7	3.7.1	35
3.1.6	14	3.5.1.	155	3.8.1	16
3.1.7	8	3.5.3	25	3.13.1	9
3.2.1	131	3.5.4	47	合計	962

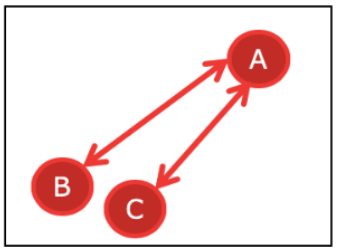
問題点

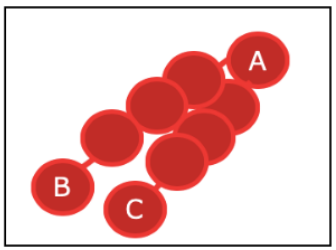
クラスによってデータ数が不均衡 (予測のばらつき)

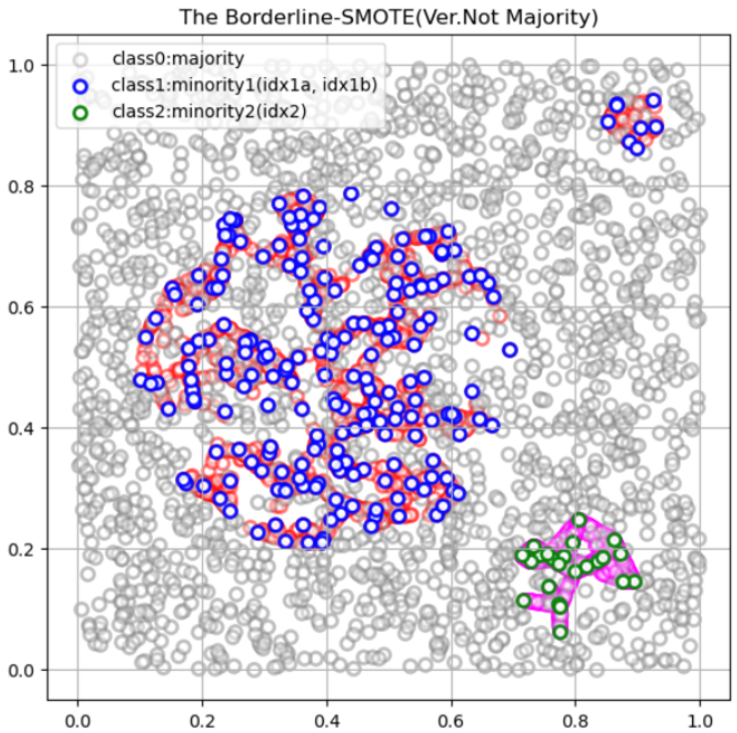
オーバーサンプリング (SNOTE) でデータ数を水増し

SMOTE[6] によるオーバーサンプリング

不均衡なクラスデータを調整 [7,8]







[6] Nitesh V. Chawla et al., 2002.

[7] 【リレー連載】わたしの推しノード –機械学習時代の申し子「SMOTE ノード」が不均衡データの壁を突破する, https://www.ibm.com/blogs/solutions/jp-ja/spssmodeler-push-node-10/

[8] BorderlineSMOTE(Ver.Multiclass_Classification).ipynb., https://github.com/hkosho/pimientitosML/blob/main/

SMOTE 実行結果

データ数最大の EC 3.5.1 クラス (155 個) になるようオーバーサンプリング

(a) Before applying SMOTE

EC Class	Num of Equation	EC Class	Num of Equation	EC Class	Num of Equation
3.1.1	122	3.2.2	24	3.5.5	12
3.1.2	59	3.3.2	6	3.5.99	11
3.1.3	152	3.4.13	6	3.6.1	94
3.1.4	29	3.4.19	7	3.7.1	35
3.1.6	14	3.5.1	155	3.8.1	16
3.1.7	8	3.5.3	25	3.13.1	9
3.2.1	131	3.5.4	47	Total	962

(b) After applying SMOTE

3.1.1	155	3.2.2	155	3.5.5	155
3.1.2	155	3.3.2	155	3.5.99	155
3.1.3	155	3.4.13	155	3.6.1	155
3.1.4	155	3.4.19	155	3.7.1	155
3.1.6	155	3.5.1	155	3.8.1	155
3.1.7	155	3.5.3	155	3.13.1	155
3.2.1	155	3.5.4	155	Total	3100

5.2. 実験結果 特徴量エンジニアリング

特徴選択・パラメータ調整の結果

RF モデル SMOTE 前「記述子数：14, 各決定木の最大深さ：20, 決定木数 200」
SMOTE 後「記述子数：23, 最大深さ：15, 決定木数 800」

(a) Before applying SMOTE

EC Num	Num of Equation	Precision	Recall	F1 Score	EC Num	Num of Equation	Precision	Recall	F1 Score
3.1.1	25	0.96	0.96	0.96	3.4.19	1	1.00	1.00	1.00
3.1.2	12	0.92	1.00	0.96	3.5.1	31	0.94	0.97	0.95
3.1.3	31	0.91	1.00	0.96	3.5.3	5	0.83	1.00	0.91
3.1.4	6	0.86	1.00	0.92	3.5.4	9	0.89	0.89	0.89
3.1.6	3	1.00	1.00	1.00	3.5.5	2	1.00	1.00	1.00
3.1.7	2	0.00	0.00	0.00	3.5.99	2	1.00	0.50	0.67
3.2.1	26	0.96	0.96	0.96	3.6.1	19	0.86	0.95	0.90
3.2.2	5	0.83	1.00	0.91	3.7.1	7	1.00	0.71	0.83
3.3.2	1	1.00	1.00	1.00	3.8.1	3	1.00	0.67	0.80
3.4.13	1	0.00	0.00	0.00	3.13.1	2	1.00	0.50	0.67

	Num of Equation	Precision	Recall	F1 Score
Total	193			
Average		0.85	0.80	0.81
Accuracy				0.92

(b) After applying SMOTE

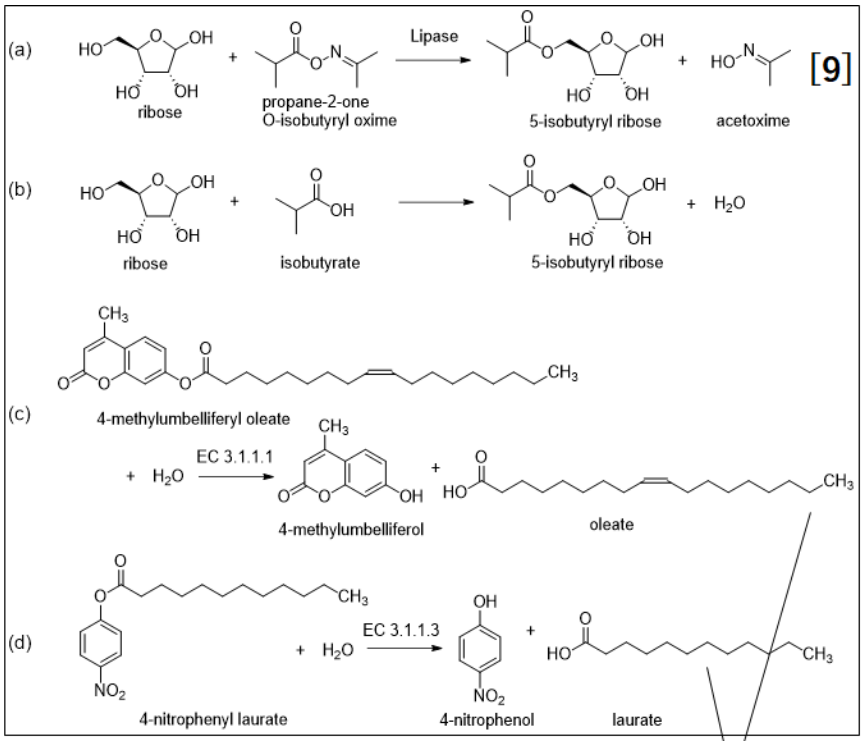
3.1.1	31	1.00	0.94	0.97	3.4.19	31	1.00	1.00	1.00
3.1.2	31	1.00	1.00	1.00	3.5.1	31	1.00	0.97	0.98
3.1.3	31	0.97	1.00	0.98	3.5.3	31	0.97	0.97	0.97
3.1.4	31	1.00	0.97	0.98	3.5.4	31	1.00	0.97	0.98
3.1.6	31	1.00	1.00	1.00	3.5.5	31	0.89	1.00	0.94
3.1.7	31	1.00	1.00	1.00	3.5.99	31	1.00	1.00	1.00
3.2.1	31	1.00	1.00	1.00	3.6.1	31	1.00	0.97	0.98
3.2.2	31	0.97	1.00	0.98	3.7.1	31	1.00	1.00	1.00
3.3.2	31	1.00	1.00	1.00	3.8.1	31	1.00	1.00	1.00
3.4.13	31	1.00	1.00	1.00	3.13.1	31	1.00	1.00	1.00

	Num of Equation	Precision	Recall	F1 Score
Total	620			
Average		0.99	0.99	0.99
Accuracy				0.99

5.2 実験結果 BRENDA(文献) 反応式に対する予測

・EC 3.1.1, EC 3.7.1, EC 3.5.3 からそれぞれ 4 つの Target 反応
→合成実験の文献 [9] や BRENDA[10] から取得

EC 3.1.1



EC 3.7.1 Target 1 ～ Target 4

EC 3.5.3 Target 1 ～ Target 4

BRENDA記載の文献反応[10]

[9] Tamas Benkovics et al., 2020
[10] BRENDA, https://www.brenda-enzymes.org/index.php

5.2 実験結果 BRENDA(文献) 反応式に対する予測

12 個中 9 個正しい EC 番号を 1 番目に予測

(a) Target Equation (registered as EC 3.1.1)

	1st	2nd	3rd
Target1	3.2.1.	3.1.1.	3.8.1.
Probability	0.349167	0.095	0.08375
Target2	3.2.1.	3.1.1.	3.7.1.
Probability	0.230313	0.179375	0.1525
Target3	3.1.1.	3.7.1.	3.2.1.
Probability	0.561849	0.094382	0.067454
Target4	3.1.1.	3.7.1.	3.5.1.
Probability	0.922124	0.052175	0.007679

(b) Target Equation (registered as EC 3.7.1)

	1st	2nd	3rd
Target1	3.13.1.	3.7.1.	3.5.99.
Probability	0.251406	0.127435	0.124286
Target2	3.7.1.	3.1.2.	3.5.1.
Probability	0.293299	0.2725	0.119097
Target3	3.7.1.	3.1.1.	3.1.2.
Probability	0.992454	0.002546	0.0025
Target4	3.7.1.	3.1.2.	3.5.1.
Probability	0.991954	0.0025	0.001625

(c) Target Equation (registered as EC 3.5.3)

	1st	2nd	3rd
Target1	3.5.3.	3.13.1.	3.7.1.
Probability	0.952604	0.009063	0.007774
Target2	3.5.3.	3.5.99.	3.5.4.
Probability	0.7375	0.07625	0.07
Target3	3.5.3.	3.5.4.	3.5.99.
Probability	0.90625	0.05	0.01625
Target4	3.5.3.	3.5.99.	3.1.1.
Probability	0.99875	0.00125	0

修論でやること

修士中間までの問題点

・モデル作成 (学習) で SMOTE の仮想データを使っていたこと
・BRENDA のデータ不足 (予測の信頼性)

新たにやること

EC 番号 1〜3 桁の予測

修論でやること

新たなデータの作成

4 つのデータベース (Rhea, BRENDA, MetaNetX, PathBank) からなる約 5 万の SMILES データセット [11] を使用

→ 50000×200 次元の特徴ベクトル作成

データ数に対する問題点

データ数が以前の 15 倍→特徴選択に膨大な時間 (推定 75 時間)

対策 (修論の予測の流れ)

1 桁，2 桁, 3 桁ごとに分けて特徴選択
トレーニングデータ (80%: 5-ford 交差検証)(1)1 桁目の予測 (EC1〜7)
(2)7 クラスで 2 桁目特徴選択
(3)2 桁目各クラスで 3 桁目特徴選択
(4)2 桁目 3 桁目で選んだ特徴の共通部分を選択
(5) 選んだ特徴に対してテストデータで F 値を評価

[11] Daniel Probstl., 2023.

おわりに

- 現状はデータセットの整形が終わり，特徴選択を実行中 (下図)
- 各桁の特徴選択で SMOTE の適用方法を検討中

```
selected_feat3 = XtrainT6.columns[list(sfs3.k_feature_idx_)]
```

```
numOfTrainData: 37900
numOfTestData: 9475
```

```
[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurrent workers.
[Parallel(n_jobs=1)]: Done   1 out of   1 | elapsed:  3.9min remaining:    0.0s
[Parallel(n_jobs=1)]: Done 199 out of 199 | elapsed: 216.8min finished

[2024-01-18 17:38:27] Features: 1/25 -- score: 0.28867226584401423[Parallel(n_jobs=1)]: Using backend SequentialBackend with 1 concurr
ent workers.
```