Module 3
学習済みANNとMILPを使用した Branch-Heightが制限された 2-Lean環式化学グラフの推定

2021年2月14日

目 次

1 概要

この冊子では, 学習済み人工ニューラルネットワーク (ANN) の重みとバイアスと目標値を与えられた時に, デスクリプタ (記述子) のベクトルを推定できる混合整数線形計画法 (Mixed Integer Linear Programming; MILP) の使い方を説明する.

MILP を定義する機能は Python で実装されていて, COIN-OR パッケージの PuLP モデリングモジュール [5, 6, 7, 8] を使用する.

まず, この冊子に添付されているファイルの一覧を示す.

- フォルダ `source_code`
  与えられた ANN に対し, 環式化学グラフの特徴ベクトルを推定するための MILP を実装した四つの Python スクリプトと, MILP 定式化における各デスクリプタの最小値と最大値を含むファイルを含むフォルダ.

  - `ann_inverter.py`
    ANN の逆問題に対する定式化した MILP を実装したスクリプト [1].
  - `cyclic_graphs_MILP_ec_id_vector.py`
    変数を初期化し, 規定のトポロジ構造 [2] を持つ環式化学グラフを推定するための定式化した MILP の制約を用意する関数を含む Python スクリプト.
  - `infer_cyclic_graphs_ec_id_vector.py`
    データを用意し, 指定の入力データに対して定式化した MILP を実行する Python スクリプト. このスクリプトの使用に関する詳細は, 第 4 節に記載されている.
  - `read_instance_BH_cyclic_v05.py`
    与えられたテキストファイルからトポロジカル仕様を読み取るために必要な関数を含む Python スクリプト.
  - フォルダ `topological_description`
    論文 [2] で詳述されている化学仕様をそれぞれ提供する六つのテキストファイルを含むフォルダ.
    - `instance_a.txt`
    - `instance_b1.txt`
    - `instance_b2.txt`
    - `instance_b3.txt`
    - `instance_b4.txt`
    - `instance_c.txt`
    - `instance_d.txt`
  - フォルダ `ANN`
    密閉式引火点 (Flash point (closed cup); Fp), 脂溶性（Lipophilicity; Lp）, および水溶性（Solubility; Sl）の三つの目標特性の学習済み人工ニューラルネットワーク（ANN）に関する情報を含むフォルダ. 上記の三つの各特性 $\texttt{property} \in \{\texttt{FP}, \texttt{LP}, \texttt{SL}\}$ について三つのファイルが提供される.

* property_desc.csv
  ANN の学習で使用されるデスクリプタを含むコンマ区切り形式のファイル.
* property_biases.txt
  学習済み ANN のバイアスの値を含むファイル.
* property_weights.txt
  学習済み ANN の重みの値を含むファイル.

各ファイルのデータ形式は第 3 節で説明されており, 実際の例は第 4 節で説明されてる.

- fv4_cyclic_stdout.cpp
  SDF 形式で保存された化学グラフの特徴ベクトルを計算する C ++プログラム. このプログラムの入力と出力は, MILP の解として推定されるグラフのデスクリプタ値を検証するために Python スクリプト infer_cyclic_graphs_ec_id_vector.py で動作するようにカスタマイズされている（解が存在する場合）. このソースファイルは, fv という名前の実行ファイルにコンパイルする必要がある.
- fv
  上記の C ++プログラムの実行可能なバイナリファイル. この実行可能ファイルは, オペレーティングシステム Linux Mint 18.3 を実行している PC 上で gcc バージョン 5.4.0 によってコンパイルされている.

この冊子の残りの部分は, 次のように構成されている. 第 2 節では使用される用語と表記法を説明する. 第 3 節はプログラムの入力データと出力データを説明し, 第 4 節は入力データと計算結果の具体例を示す.

2 用語と表記法

この節では, この冊子で使用されている用語と表記法について説明する.

- **特徴ベクトル**
  特徴ベクトルは, デスクリプタと呼ばれる特定のパラメータの数値を格納する. 本研究では, 非水素原子の数, ある程度の頂点の数など, グラフ理論のデスクリプタを選択する.
- **人工ニューラルネットワーク - ANN**
  人工ニューラルネットワークは, 機械学習の方法の一つである. 入力の特徴ベクトルと出力の目標データのペア間の相関関数を構築する方法を提供する.
- **入力層, 中間層, 出力層**
  順伝播型ニューラルネットワークの多層パーセプトロンモデルを扱う. これらのニューラルネットワークは、いくつかの層で構成されている. 最初に入力層があり, 各ニューロンは特徴ベクトルの一つの値を入力として受け取る. 次に隠れ層がある. ここでは, 入力層からの値が順伝播的に伝播され, 一つの層の各ノードが次の層の全てのノードに接続される. 最後に, 出力は出力層で与えられる. 単一の目標値の予測を扱うため, 出力層は単一のノードで構成されていると仮定する.

- **重み**
  ANN 内の二つのノードを接続する各辺には, 重みと呼ばれる実数値が割り当てられる. ANN の学習プロセスの一部で, 既知の特徴ベクトルと目標値のペアに基づいて, それぞれの重みの値を決定する.

- **バイアス**
  入力層のノードを除く ANN の各ノードには, バイアスと呼ばれる実数値が割り当てられる. バイアスは辺の重みと同様に, 学習プロセスを通じて決定される.

- **活性化関数**
  ANN では, 各ノードは入力の活性化関数と呼ばれる関数として出力を生成する. 各ノードには, 活性化関数として Rectified Linear-Unit（ReLU）関数があると仮定する. これは ANN の逆問題の MILP 定式化 [1] で正確に表すことができる.

- **混合整数計画問題 (MILP)**
  すべての制約が線形式として与えられ, いくつかの決定変数が整数値のみを取る必要がある数理計画問題の一種. MILP の標準的な文献として, [3] を挙げておく.

- **グラフ**
  頂点の有限集合とエッジの有限集合で構成される抽象的な組合せ構造. ここで各辺は頂点のペアである. 無向グラフを扱う. すなわち, 辺が順序付けられていない頂点のペアになっているグラフを扱う. グラフ理論の初等的な内容に関しては様々な文献 (たとえば [4] など) が知られている.

3　プログラムの入力と出力

この節では, プログラムの入力と出力の形式について説明する. 第 3.1 節ではプログラムの入力形式の例を示し, 第 3.2 節では具体的な計算例を示す. 次に, 第 3.3 節ではプログラムの出力形式の例を示し, 第 3.4 節では具体的な計算例を示す.

3.1　プログラムの入力

この節では、プログラムへの入力について説明する. 最初の入力には、以下を含む三つのテキストファイルが必要となる.

- CSV 形式のデスクリプタ名,
- テキスト形式の学習済み ANN の重みと
- テキスト形式の学習済み ANN のバイアス.

プログラム実行のコマンドラインパラメータとして用いる共通の接頭辞が `TT` の場合, 学習済み ANN のディスクリプタ名, 重み, バイアスが格納されたファイルはそれぞれ `TT_desc.csv`, `TT_weights.txt`, `TT_biases.txt` というファイル名で保存されていなければならない.

次に, 学習済み ANN に基づいて化学グラフを推定するための目標値を指定する. その次は, [2] で説明されているように、テキストファイルで指定された化学仕様と、第 3.4 節で説明されている出力ファイルのファイル名の接頭辞を指定する.

最後に, 使用する MILP ソルバープログラムを選択する. 以下から選択できる.

- 1: CPLEX (商用 MILP ソルバー) [9].

ファイル `infer_acyclic_graphs_ec_id_vector.py` のパラメータ `CPLEX_PATH` は、CPLEX プログラム実行可能ファイルの正しいパスに設定する必要があることに注意せよ.

- 2: CBC (無償のオープンソース MILP ソルバー). Python 用の PuLP パッケージ [5] に付属している.

3.2 入力データ形式

この節では, プログラムの実際の入力例を示す. 特に, 第 3.1 節で述べた三つの入力ファイルの具体例を示す.

このプログラムの目的は, 与えられた学習済み ANN から所望の出力を生成する特徴ベクトルを計算することである. 図 1 は学習済み ANN の例を示す.

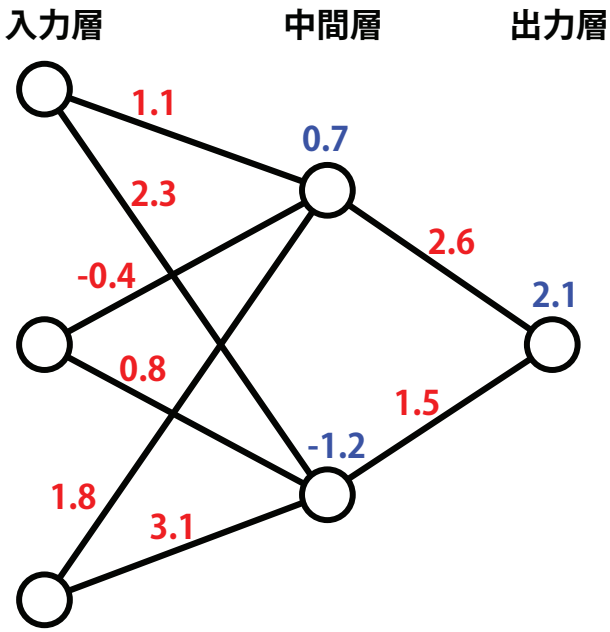

図 1: 学習済み ANN の例. 赤い数字は ANN の重みを示し, 青い数字はバイアスを示す.

学習済み ANN の重みとバイアスの情報はそれぞれ二つのテキストファイルに書き込まれる. まず, 重みの情報を含むテキストファイルの構造を示す. このテキストファイルの最初の行には, ANN のアーキテクチャに関する情報 (各レイヤーのノードの数) が書き込まれる. 2 行目以降は、ANN の重みが書き込まれる. 各行は, ＡＮＮの 1 つのノードに接続する辺の重みが書き込まれており, 最初は入力層のノードの重み, 次に隠れ層のノードの重みを表している. 以下に, 図 1 に示されてる ANN のテキストファイルの例を示す.

重みのデータ形式

3 2 1

1.1 2.3
-0.4 0.8
1.8 3.1
2.6
1.5

次は ANN のバイアスのテキストファイルである. Fig. 1 のバイアスを以下に示す.

バイアスのデータ形式

0.7
-1.2
2.1

最後に, 特徴ベクトルのデータが含まれたテキストファイルの形式について説明する. このファイルの最初の行には, 特徴ベクトルで使用されるデスクリプタの名前が書き込まれている. 2 行目以降は, トレーニングデータセット内の各化学グラフのデスクリプタの数値が書き込まれている. 一つの化学グラフ毎に 1 行書き込まれる. 例えば, ANN フォルダの TT_desc.csv を確認せよ. ここで, TT $\in$ {FP, LP, SL} である.

3.3 プログラムの出力

この節では, プログラムの出力について説明する. 与えられた学習済み ANN の予測として与えられた目標値となるような特徴ベクトルを持つ非環式化合グラフが存在する場合, プログラムは特徴ベクトルを出力する. そのような化学グラフが存在しない場合, 存在しないと出力する. 次の節では, プログラムの出力について説明する.

3.4 出力データ形式

この節では, パソコン上で得られたプログラムの出力データについて説明する. プログラムを実行すると, ターミナルに表示される標準エラーストリームにいくつかのメッセージが出力される. MILP ソルバーが計算を完了すると, 計算のステータスがターミナルに表示される.

ターミナル上の出力結果のデータ形式

Initializing Time: 0.809	# `stderr` への書き込み
Start Solving Using CPLEX...	# `stderr` への書き込み
Status: Feasible	# 解のステータス
MILP y*: 197.922	# MILP で計算された目標値
ANN propagated y*: 197.922	# 学習済み ANN によって計算された目標値
All descriptors match	# MILP と推定されたグラフによるデスクリプタの値
Solving Time: 16.711	# MILP ソルバーの所要時間

最後に, 実行可能解が存在する場合、プログラムは二つのテキストファイルを出力する. プログラムは出力ファイルのファイル名を入力の一部として要求することに注意せよ. 出力ファイル名として与えられたパラメータが `filename` であると仮定する. 結果として得られる二つのテキストファイルのファイル名を以下に示す.

- `filename.sdf`
- `filename_partition.sdf`.

`filename.sdf` は推定された化学グラフの情報が SDF(Structure Data File) 形式で書き込まれている. 詳細については, 公式ドキュメント（英語）を参照せよ.

`http://help.accelrysonline.com/ulm/onelab/1.0/content/ulm_pdfs/direct/reference/ctfileformats2016.pdf`

`filename_partition.sdf` には, [2] で説明されているように, `filename.sdf` から非巡回部分グラフへの閉路グラフの分解が含まれている.

4 プログラムの実行と計算例

この節では, プログラムを実行する方法と具体的な計算例について説明する. 以下は,

`infer_cyclic_graphs_ec_id_vector.py`

を実行する例である.

4.1 プログラムの実行

まず第 1 節で説明されているように, ターミナルが `source_code` フォルダの場所に正しく移動されていることを確認する.

```
python infer_cyclic_graphs_ec_id_vector.py trained_ann_filename_prefix target_value
       chemical_specification output_file_name solver_type
```

例として目標特性に密閉式引火点を使用する. ファイル名の接頭辞が FP となる学習済み ANN、目標値を 200, 化学仕様のためのファイルを `instance_a.txt`, MILP ソルバー (パラメータ値は 1) を CPLEX[9] とする.

```
python infer_cyclic_graphs_ec_id_vector.py ANN/FP 200
      chemical_specification/instance_a.txt result 1
```

上記のコマンドを実行すると, ターミナルプロンプトに次のテキストが表示される.

ターミナル上の出力

Initializing Time: 0.809
Start Solving Using CPLEX...
Status: Feasible
MILP y*: 197.922
ANN propagated y*: 197.922
All descriptors match
Solving Time: 16.711

出力ファイル `result.sdf` および `result_partition.txt` の内容を以下に示す.

File `result.sdf`

```
 1
MILP_cyclic
ec_id_vector
 43 46  0  0  0  0  0  0  0  0999 V2000
    0.0000    0.0000    0.0000 C   0  0  0  0  0  0  0  0  0  0  0  0
    0.0000    0.0000    0.0000 C   0  0  0  0  0  0  0  0  0  0  0  0
    0.0000    0.0000    0.0000 C   0  0  0  0  0  0  0  0  0  0  0  0
    0.0000    0.0000    0.0000 C   0  0  0  0  0  0  0  0  0  0  0  0
    0.0000    0.0000    0.0000 C   0  0  0  0  0  0  0  0  0  0  0  0
    0.0000    0.0000    0.0000 O   0  0  0  0  0  0  0  0  0  0  0  0
    0.0000    0.0000    0.0000 C   0  0  0  0  0  0  0  0  0  0  0  0
    0.0000    0.0000    0.0000 C   0  0  0  0  0  0  0  0  0  0  0  0
    0.0000    0.0000    0.0000 N   0  0  0  0  0  0  0  0  0  0  0  0
    0.0000    0.0000    0.0000 C   0  0  0  0  0  0  0  0  0  0  0  0
    0.0000    0.0000    0.0000 C   0  0  0  0  0  0  0  0  0  0  0  0
    0.0000    0.0000    0.0000 C   0  0  0  0  0  0  0  0  0  0  0  0
    0.0000    0.0000    0.0000 C   0  0  0  0  0  0  0  0  0  0  0  0
    0.0000    0.0000    0.0000 N   0  0  0  0  0  0  0  0  0  0  0  0
    0.0000    0.0000    0.0000 C   0  0  0  0  0  0  0  0  0  0  0  0
```

```
    0.0000    0.0000    0.0000 C   0  0  0  0  0  0  0  0  0  0  0  0
    0.0000    0.0000    0.0000 C   0  0  0  0  0  0  0  0  0  0  0  0
    0.0000    0.0000    0.0000 C   0  0  0  0  0  0  0  0  0  0  0  0
    0.0000    0.0000    0.0000 C   0  0  0  0  0  0  0  0  0  0  0  0
    0.0000    0.0000    0.0000 C   0  0  0  0  0  0  0  0  0  0  0  0
    0.0000    0.0000    0.0000 C   0  0  0  0  0  0  0  0  0  0  0  0
    0.0000    0.0000    0.0000 C   0  0  0  0  0  0  0  0  0  0  0  0
    0.0000    0.0000    0.0000 C   0  0  0  0  0  0  0  0  0  0  0  0
    0.0000    0.0000    0.0000 C   0  0  0  0  0  0  0  0  0  0  0  0
    0.0000    0.0000    0.0000 C   0  0  0  0  0  0  0  0  0  0  0  0
    0.0000    0.0000    0.0000 C   0  0  0  0  0  0  0  0  0  0  0  0
    0.0000    0.0000    0.0000 C   0  0  0  0  0  0  0  0  0  0  0  0
    0.0000    0.0000    0.0000 C   0  0  0  0  0  0  0  0  0  0  0  0
    0.0000    0.0000    0.0000 C   0  0  0  0  0  0  0  0  0  0  0  0
    0.0000    0.0000    0.0000 N   0  0  0  0  0  0  0  0  0  0  0  0
    0.0000    0.0000    0.0000 C   0  0  0  0  0  0  0  0  0  0  0  0
    0.0000    0.0000    0.0000 C   0  0  0  0  0  0  0  0  0  0  0  0
    0.0000    0.0000    0.0000 C   0  0  0  0  0  0  0  0  0  0  0  0
    0.0000    0.0000    0.0000 O   0  0  0  0  0  0  0  0  0  0  0  0
    0.0000    0.0000    0.0000 C   0  0  0  0  0  0  0  0  0  0  0  0
    0.0000    0.0000    0.0000 C   0  0  0  0  0  0  0  0  0  0  0  0
    0.0000    0.0000    0.0000 C   0  0  0  0  0  0  0  0  0  0  0  0
    0.0000    0.0000    0.0000 O   0  0  0  0  0  0  0  0  0  0  0  0
    0.0000    0.0000    0.0000 C   0  0  0  0  0  0  0  0  0  0  0  0
    0.0000    0.0000    0.0000 C   0  0  0  0  0  0  0  0  0  0  0  0
    0.0000    0.0000    0.0000 O   0  0  0  0  0  0  0  0  0  0  0  0
    0.0000    0.0000    0.0000 C   0  0  0  0  0  0  0  0  0  0  0  0
    0.0000    0.0000    0.0000 C   0  0  0  0  0  0  0  0  0  0  0  0
  1  2  2  0  0  0  0
  1 24  1  0  0  0  0
  2 25  1  0  0  0  0
  2 41  1  0  0  0  0
  3  4  1  0  0  0  0
  3 23  2  0  0  0  0
  3 38  1  0  0  0  0
  4  5  2  0  0  0  0
  4 33  1  0  0  0  0
  5 24  1  0  0  0  0
  6 12  1  0  0  0  0
  6 18  1  0  0  0  0
```

```
  7  9  1  0  0  0  0
  7 11  2  0  0  0  0
  7 28  1  0  0  0  0
  8  9  1  0  0  0  0
  8 10  1  0  0  0  0
 10 11  1  0  0  0  0
 11 12  1  0  0  0  0
 12 17  1  0  0  0  0
 13 14  1  0  0  0  0
 13 17  1  0  0  0  0
 13 19  1  0  0  0  0
 14 15  1  0  0  0  0
 14 16  1  0  0  0  0
 17 18  2  0  0  0  0
 18 22  1  0  0  0  0
 19 20  1  0  0  0  0
 19 21  1  0  0  0  0
 19 23  1  0  0  0  0
 22 23  1  0  0  0  0
 24 25  1  0  0  0  0
 25 26  1  0  0  0  0
 25 27  1  0  0  0  0
 28 29  1  0  0  0  0
 28 30  1  0  0  0  0
 30 31  1  0  0  0  0
 30 32  1  0  0  0  0
 33 34  1  0  0  0  0
 34 35  1  0  0  0  0
 35 36  1  0  0  0  0
 35 37  1  0  0  0  0
 38 39  1  0  0  0  0
 39 40  3  0  0  0  0
 41 42  1  0  0  0  0
 42 43  3  0  0  0  0
M  END
$$$$
```

File `result_partition.sdf`

```
12
9
0 1
10
0 0
11
0 0
12
0 0
13
1 3
17
0 0
18
0 1
19
0 1
22
0 0
23
0 1
24
0 2
25
0 4
15
9 7 11
1 3
9 8 10
0 3
10 11
0 0
11 12
0 0
12 6 18
0 1
12 17
```

```
0 0
13 17
0 0
13 19
0 0
17 18
0 0
18 22
0 0
19 23
0 0
22 23
0 0
23 3 4 5 24
4 6
24 1 2 25
3 5
24 25
0 2
```

参考文献

[1] T. Akutsu and H. Nagamochi. A Mixed Integer Linear Programming Formulation to Artificial Neural Networks, in Proceedings of the 2019 2nd International Conference on Information Science and Systems, pp. 215–220, https://doi.org/10.1145/3322645.3322683.

[2] T. Akutsu and H. Nagamochi. A Novel Method for Inference of Chemical Compounds with Prescribed Topological Substructures Based on Integer Programming. Arxiv preprint, arXiv:2010.09203

[3] J. Matousek and B. Gärtner. Understanding and Using Linear Programming. Springer, 2007.

[4] M. S. Rahman. Basic Graph Theory. Springer, 2017.

[5] A Python Linear Programming API, `https://github.com/coin-or/pulp`.

[6] Optimization with PuLP, `http://coin-or.github.io/pulp/`.

[7] The Python Papers Monograph, `https://ojs.pythonpapers.org/index.php/tppm/article/view/111`.

[8] Optimization with PuLP, `https://pythonhosted.org/PuLP/`.

[9] IBM ILOG CPLEX Optimization Studio 12.8 User Manual. https://www.ibm.com/support/knowledgecenter/SSSA5P_12.8.0/ilog.odms.studio.help/pdf/usrcplex.pdf.