Module 1における特徴ベクトル生成の手順

mol-infer/2L-model

2021 年 3 月 2 日

目 次

1　はじめに

本稿では, 本プロジェクト (`mol-infer/2L-model`) における Module 1 の手順を解説する. この Module 1 の入力と出力は以下の通りである.

入力: 化学グラフの集合 $D = \{G_1, G_2, \ldots, G_p\}$.

出力: 特徴ベクトルの集合 $\mathcal{F}(D) \triangleq \{f(G_1), f(G_2), \ldots, f(G_p)\}$. ただし f は化学グラフを特徴ベクトルに変換する関数で, 論文 [1] において提案されたものである.

なお入力される化学グラフはいくつかの条件を満たさなければならない. たとえば四つ以上の炭素原子を持つ必要がある. (したがって水 (H_2O) は不可である.) また電荷のある原子, 価数が我々の定める標準値と異なる原子を持ってはならない. 詳しくは第 3 節を参照されたい. ただしこれまでのバージョンと異なり, **連結でさえあれば, 環構造・非環構造の別を問わずに取り扱うことができる**.

出力は, 具体的には以下の三つのファイルで与えられる.

(1) 外縁木情報が記されたテキストファイル.

(2) 特徴ベクトルが記された csv ファイル.

(3) 正規化された特徴ベクトルが記された csv ファイル.

このうち (3) は, Module 2 (ニューラルネットワークによる順方向予測) の入力として必要となる. また (1), (2), (3) のすべてが, Module 3 (混合整数線形計画法による逆方向予測) の入力として必要となる.

本稿の構成は以下の通りである.

- 第 2 節: 基本的な用語, およびパッケージのファイル構成の説明.
- 第 3 節: 実行例.
- 第 4 節: プログラムの入出力に関する詳細.

2 準備

2.1 用語の説明

化学グラフ． **節点** (node) の集合と, 節点と節点を結ぶ**辺** (edge) の集合の対を**グラフ** (graph) という. グラフにおける**閉路** (cycle) とは, ある節点を出発し, 辺を次々になぞって得られる節点の系列 (ただし始点と終点以外の節点は二度以上訪れない) のうち, 始点と終点が一致するものをいう.

節点に対する元素 (炭素, 窒素, 酸素など) の割当, 辺に対する多重度 (一般に 1 以上 4 以下の整数) の割当が与えられたグラフを**化学グラフ** (chemical graph) という.

記述子． 化学グラフの文脈における**記述子** (descriptor) とは, 化学グラフの特徴を表す指標をいう. 一般に, 化学グラフは一つの記述子に対して一つの数値を取る. 本プロジェクトで用いられる記述子の例として, 水素を除く原子の数, コアに属する原子の数, コアの高さ, などがある. 詳細は論文 [1] を参照のこと.

特徴ベクトル． 化学グラフと記述子の系列が与えられたとき, その化学グラフが各記述子に対して取る数値を順に並べたベクトルを**特徴ベクトル** (feature vector) という.

2.2 ファイル構成

パッケージに含まれるファイルとその役割は以下の通りである.

- `Makefile`: コンパイルのための makefile.
- `eliminate.py`: 炭素原子の数が 4 未満など, 本プロジェクトが想定していない化学グラフを除去するための Python スクリプト.
- `2L_fv_main.cpp`: **特徴ベクトル生成を行うメインプログラム**のソースコード (C++).
- `fringe`: 特徴生成プログラムにインクルードされる hpp ファイルを含むサブディレクトリ. 中身は以下の通り.
  - `ChemicalGraph.hpp`
  - `RootedGraph.hpp`
  - `TopologyGraph.hpp`
  - `commonused.hpp`
  - `cross_timer.h`

- bin: 特徴生成プログラムの実行形式ファイルを含むサブディレクトリ. 中身は以下の通り.
  - 2L_FV_macos.bin: macOS.
  - 2L_FV_ubuntu.bin: Linux (Ubuntu).
  - 2L_FV_win.exe: Windows.
- sample_data: 実行テストのためのデータファイルを含むサブディレクトリ. 中身は以下の通り.
  - Sl_all.sdf: 水可溶性に関するデータ (http://moleculenet.ai/datasets-1) に記された 1128 個の化合物に関するデータ.
  - Sl_all_eli.sdf: Sl_all.sdf に eliminate.py を適用して得られた SDF ファイル. 922 個の化合物を含む.
  - Sl_all_eli_fringe.txt: Sl_all_eli.sdf に特徴生成プログラムを適用して得られたファイルで, 外縁木情報が記されたテキストファイル.
  - Sl_all_eli_desc.csv: Sl_all_eli.sdf に特徴生成プログラムを適用して得られたファイルで, 特徴ベクトルが記された csv ファイル.
  - Sl_all_eli_desc_norm.csv: Sl_all_eli.sdf に特徴生成プログラムを適用して得られたファイルで, 正規化された特徴ベクトルが記された csv ファイル.

3　実行例

3.1　データの妥当性を確認

化合物データは標準的な SDF ファイルによって与えられなければならない.
また各化学グラフ $G \in D$ は以下の条件を満たす必要がある.

(i) G は 4 つ以上の炭素原子を持つ. また電荷のある原子, 価数が標準値[1]と異なる原子を持たない.

(ii) 芳香辺 (aromatic edge) を含まない.

(i) については, すべての化学グラフが条件を満たすかどうか, パッケージ内のプログラムを用いて判定し, 条件を満たさないものを除去することができる.
(ii) について, 芳香辺は事前に単結合もしくは二重結合に変換しておく必要がある.

対象外の化学グラフを除去. すべての化学グラフが条件 (i) を満たすか否かを判定し, 満たすもののみを別の SDF ファイルにまとめるには `eliminate.py` を用いる.

```
$ python eliminate.py input.sdf
```

もし条件 (i) を満たさない化学グラフがあれば, その化学グラフの CID が警告として出力される.
実行後, 条件 (i) を満たす化学グラフのみをまとめた `input_eli.sdf` という SDF ファイルが生成される. もしすべての化学グラフが (i) を満たす場合は, `input.sdf` と `input_eli.sdf` の中身は同一となる.

3.2　特徴ベクトルの生成

すべての化学グラフが上記 (i), (ii) を満たすような SDF ファイルに対して特徴ベクトルを生成するには `2L_fv_main.cpp` をコンパイルして用いる. なお当方の maxOS, Linux (Ubuntu), Windows 環境においてコンパイルした実行形式ファイルを `bin` ディレクトリに置いているので, それを用いても構わない.
ユーザが自分でコンパイルする場合, `Makefile` が使用可能な環境では,

```
$ make 2L_FV
```

とすればよい. そうでなければ

[1]第 4.3 節を参照せよ.

```
$ g++ -O2 -Wall -std=c++11 -o 2L_FV 2L_fv_main.cpp
```

とする.

input_eli.sdf に対して特徴ベクトルを生成し, その結果を output_fringe.txt, output_desc.csv, output_desc_norm.csv に出力するには,

```
$ ./2L_FV input_eli.sdf output
```

とする. 引数を与えずに実行すれば (もしくは引数が適切に与えられなかった場合), 指定すべき引数が出力されて終了する.

4　プログラムの入出力に関する詳細

ここでは特徴ベクトル生成のメインプログラム 2L_FV (ソースコードは 2L_fv_main.cpp) の入出力および実行に関する注意を述べる.

4.1　入力

このプログラムは入力のフォーマットとして標準的な SDF (Structure Data File) を使用している. SDF の書式について, 以下を参考資料として挙げておく.

- http://help.accelrysonline.com/ulm/onelab/1.0/content/ulm_pdfs/direct/reference/ctfileformats2016.pdf (2021 年 2 月 1 日 アクセス確認)
- https://www.chem-station.com/blog/2012/04/sdf.html (2021 年 2 月 1 日 アクセス確認)

4.2　出力

プログラムは三つのファイルを出力する. 具体的には,

```
$ ./2L_FV input_eli.sdf output
```

と実行した場合,

(1) 外縁木情報が記されたテキストファイル (output_fringe.txt);

(2) 特徴ベクトルが記された csv ファイル (output_desc.csv);

(3) 正規化された特徴ベクトルが記された csv ファイル (output_desc_norm.csv);

が出力される. このうち (3) は, Module 2 (ニューラルネットワークによる順方向予測) の入力として必要となる. また (1), (2), (3) のすべてが, Module 3 (混合整数線形計画法による逆方向予測) の入力として必要となる.

なお本プログラムが出力する (2), (3) の csv ファイルは, 独自の FV フォーマットを採用している. このテキストファイルは, カンマで区切った CSV ファイルであり, Excel などの表計算ソフトで開くことができる. 具体的には, 一行目には特徴ベクトルの記述子が, 二行目以降の各行には特徴ベクトルの数値データが記入される.

4.3 注意

原子の質量の標準値は, プログラムの中に記載するハードコード仕様になっている. 具体的には

- `2L_fv_main.cpp` 内の関数 `init_mass()`
- `fringe/commonused.hpp` 内のオブジェクト `map_atomic_number` および `map_int2atomic`

で定められているが, 質量の変更や, ほかの原子が必要な場合には, 編集して再度コンパイルすることで利用できる.

参考文献

[1] Y. Shi, J. Zhu, N.A. Azam, K. Haraguchi, L. Zhao, H. Nagamochi and T. Akutsu. A Two-layered Model for Inferring Chemical Compounds with Integer Programming. 2021, submitted.