TP : Fouille de données de type graphe et chémoinformatique

2016-10-06 M2BI Chloé-Agathe Azencott

1 Analyse du jeu de données Mutag

Le but de ce TP est d'analyser le jeu de données de mutagénicité Mutag. Ce jeu de données contient 188 molécules étiquettées 1 ou -1 selon qu'elles sont mutagènes ou non. Ce TP est conçu en Python.

Données Téléchargez le jeu de données Mutag depuis http://tinyurl.com/mutag2015.
Le fichier `mutag_188_data.smi` contient les 188 molécules au format SMILES.
Le fichier `mutag_188_target.txt` contient leurs étiquettes, dans le même ordre.

Fingerprints Les fichiers `mutag_188_*.fingerprints` contiennent des représentations des molécules du jeu de données Mutag précalculées. Le format est le suivant :

- Chaque ligne commençant par # représente une sous-structure, autrement dit, un sous-graphe moléculaire ;
- Chaque ligne ne commençant pas par # correspond à une molécule, représentée par sa chaîne de caractères SMILES, suivie de son nom, suivie de sa fingerprint dans un format type dictionnaire (`<indice>:<valeur>` pour tous les indices pour lesquels la valeur n'est pas 0.)

En ce qui concerne les noms des fichiers, `PATH_d<depth>` correspond à l'extraction de tous les sous-chemins de taille allant jusqu'à `<depth>` et `CIRC_d<depth>` correspond à l'extraction de tous les sous-arbres de profondeur allant jusqu'à `<depth>`.

1. Quelle est la longueur des fingerprints pour chacun des fichiers ?

Lab notebook La suite de ce TP est disponible sous forme de notebook Jupyter (anciennement ipython).

Pour y accéder, lancer

```
jupyter notebook TP\ Mutag.ipynb
```

ou (systèmes plus anciens) :

```
ipython notebook TP\ Mutag.ipynb
```

Si le notebook ne marchait pas, vous en trouverez une version non-interactive en HTML dans le fichier `TP Mutag.html`.

2 Informations complémentaires

scikit-learn La bibliothèque `scikit-learn`[1] fournit un grand nombre d'outils permettant d'effectuer les tâches que nous venons d'implémenter. De nombreux algorithmes de classifica-

1. http://scikit-learn.org/

tion, régression et clustering y sont implémentés, ainsi que des méthodes facilitant la validation croisée, l'évaluation des performances, la normalisation des données, la sélection de variables...

scikit-learn permet par exemple aussi de tracer des courbes ROC (taux de vrais positifs, ou sensibilité, en fonction du taux de faux positifs, ou 1-sensitivité, pour tous les seuils possibles) et de calculer l'aire sous cette courbe.

scikit-learn repose sur `NumPy` (Numerical Python), une extension de Python destinée à la manipulation de matrices et tableaux multidimensionels, et `SciPy` (Scientific Python), une collection de bibliothèques de calcul scientifique, ainsi que sur `matplotlib`, une bibliothèque de visualisation de données sous forme graphiques.

OpenBabel Open Babel [2] est une toolbox libre conçue pour manipuler des données chimiques. OpenBabel est capable de calculer un certain nombre de représentations "fingerprints" prédéfinies.

Le paquet **Pybel** [3] est un wrapper Python pour OpenBabel :

```
import pybel
```

Pour lire le fichier de molécules et créer une liste d'objets de la classe `pybel.Molecule` :

```
chemicals = list(pybel.readfile("smi", "mutag_188_data.can"))
```

Nous pouvons ensuite extraire une représentation vectorielle ("fingerprint") pour chacune des molécules. Par défaut, Pybel propose d'indexer les fragments linéaires (ou sous-chemin) d'une longueur variant de 1 à 7 atomes. Pour plus d'information, voir : http://openbabel.org/docs/2.3.1/Fingerprints/fingerprints.html?highlight=fp2.

```
fingerprints = [pybel.Molecule.calcfp(chem, "FP2") for chem in chemicals]
```

`fingerprints` est une liste d'objets de type `pybel.Fingerprint`. Pour accéder aux indices des bits à 1 dans ces fingerprints, on peut utiliser l'attribut `bits`. Par exemple :

```
fingerprints[0].bits
```

Pour calculer d'autres types de fingerprints, il faut utiliser d'autres logiciels, tels que ChemCPP [4], ou les implémenter soi-même. Noel O'Boyle propose un exemple d'implémentation de fingerprints circulaires (ECFP) ici :
http://baoilleach.blogspot.fr/2008/02/calculate-circular-fingerprints-with.html.

2. http://openbabel.org
3. http://openbabel.org/docs/2.3.1/UseTheLibrary/Python_Pybel.html
4. http://chemcpp.sourceforge.net/html/index.html