DDI

A graph neural network model for drug drug interaction

Mario Correddu

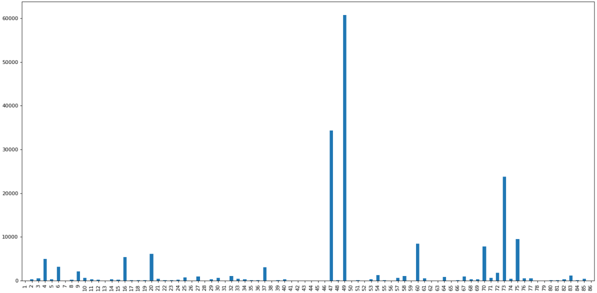
October 28, 2022

Introduction

- the aim of this project is, given a dataset made of couples of drugs if and what side effect taking the drugs together may cause
- Several existing methods predict drug interactions, but require detailed, but often unavailable drug information as inputs, such as drug targets. We aim to predict DDI types for given drug pairs and drug–food constituent pairs using only name and structural information as inputs. We remark that such framework can also be applied to drug food constituent pairs as long as their structural information is available.

Dataset

- the dataset consists of 1706 different drugs and 191,808 drug combinations, whose side effects are categorized in 86 different types of reaction
- the classes are heavily unbalanced:

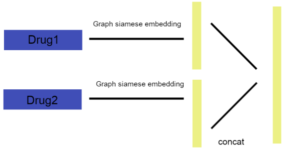


Data preparation

- in order to implement a GNN model from smiles we applied the function given by tdc that converted the smile representations into pytorch geometric data objects
- however we realised that there were a couple of drugs that weren't recognised by the function, so, knowing the Drunkbank Ids of such drugs we used the pubchem online converter at https://pubchem.ncbi.nlm.nih.gov/idexchange/idexchange.cgi to get another smile representation, now recognised by the converter
- another issue was that around 300 couples were repeated with different indexes. Given this small amount of drugs would have caused to change the problem from a multi class classification to a multi-label classification, it was decided to simply discard the couple that were double, aiming thus the analysis to ideally find the most probable side effect

The model

- two-step model: the idea is to firstly capture all the useful informations from the single molecules and later also add the information given by their interaction with other molecules.
- During the first step we aim to find a nice representation of our molecules as a vector, starting from the pytorch geometric representation. We deploy a graph neural network that takes as input our molecules and after a two layers of graph convolution and two fully connected ones returns our embedding through an operation of global max pooling.



- final linear layer that, after concatenating the two embeddings, acts as a classifier that aims to guess the right side effect, with loss given by the crossentropy.

The model

- second step: consider the fully connected graph made out by the 1706 drugs as nodes and connected by edges that can be of 1 of any of the 86 types according to the side effect.
- model is a graph neural network on the whole graph with the task of predicting edge labels. The kind of layer that is used is the l graph convolutional operator from the “Modeling Relational Data with Graph Convolutional Networks” paper (https://arxiv.org/abs/1703.06103), defined by equation:

$$x_i' = Wx_i + \sum_{rinR} \sum_{j \in N_r(i)} \frac{1}{|N_r(i)|} W_r j$$

where R is the set of relations (edge type).

The model

- dropout of parameter 0.4, skip connection
- the classification was made, by representing every edge through the concatenation of its connected nodes representations, which was later processed by a two layer fully connected neural network, which acts as our classifier.
- Everywhere in the network the activation function was the ReLu.
- The second network was trained using a special kind of minibatches that constructed a graph from a subset of nodes and edges.
- We applied our model using the GCN and the GAT operator for the first part of the model. We then confronted all the results together with the second half of the model trained also on the MACCS molecular descriptors.

Results

benchmark at https://academic.oup.com/nar/article/46/D1/D1074/4602867

/	acc	MAcroF1	Kappa
GCN embed	0.923	0.95	0.91
GAT embed	0.912	0.90	0.90
MACCS RGCN	0.938	0.91	0.926
GCN embed RGCN	0.97	0.936	0.964
GAT embed RGCN	0.971	0.939	0.965

Accuracy= $\frac{1}{N}\sum_{i\in L} TP_i$ where L is the set of all possible labels, TP_i is the number of points labeled correctly as i, and N is the sum of all points in the dataset.
MacroF1= $\frac{1}{\#L}\sum_{i\in L}\frac{TP_i}{N_i}$, where N_i is the total number of points whose true label is i, while $\#L$ is the number of classes
Kappa=$\frac{p_o - p_e}{1-p_e}$ where p_o is the empirical probability of agreement and p_e is the expected agreement when both annotators assign labels randomly. Alas p_e is estimated using a per-annotator empirical prior over the class labels.

Grazie per l'attenzione!