Machine Learning Engineer Nanodegree

Capstone Proposal

Predicting Molecular Properties - A Kaggle Competition

Submitted By :
Aditi Basu

UDACITY

Contents

1 Background 2

2 Problem Statement 2

3 Datasets & Inputs 2

4 Solution Statement 4

5 Benchmark Model 4

6 Evaluation Metrics 4

7 Project Design 5

7.1 Data Pre-processing 5

7.2 Data Cleaning 6

7.3 Domain Knowledge 6

7.4 Feature Engineering 6

7.5 Model Selection 6

1 Background

Nuclear Magnetic Resonance (NMR), a closely related technology to MRI, is typically used by researchers around the world to further their understanding of the structure and dynamics of molecules, across areas like environmental science, pharmaceutical science, and materials science.

One of the properties measured is the scalar coupling constant. These are effectively the magnetic interactions between a pair of atoms. The strength of this magnetic interaction depends on intervening electrons and chemical bonds that make up a molecule's three-dimensional structure.

2 Problem Statement

The objective of this project is to develop an algorithm that can predict the magnetic interaction between two atoms in a molecule (i.e. the scalar coupling constant).

Using state-of-the-art methods from quantum mechanics, it is possible to accurately calculate scalar coupling constants given only a 3D molecular structure as input. However, these quantum mechanics calculations are extremely expensive (days or weeks per molecule), and therefore have limited applicability in day-to-day workflows.

A fast and reliable method to predict these interactions will allow medicinal chemists to gain structural insights faster and cheaper, enabling scientists to understand how the 3D chemical structure of a molecule affects its properties and behavior.

Ultimately, such tools will enable researchers to make progress in a range of important problems, like designing molecules to carry out specific cellular tasks, or designing better drug molecules to fight disease.

3 Datasets & Inputs

The datasets used in this project are provided courtesy of Kaggle. The following data/files are provided:

- train.csv - the training set, where the first column (molecule_name) is the name of the molecule where the coupling constant originates (the corresponding XYZ file is located at ./structures/.xyz), the second (atom_index_0) and third column (atom_index_1) is the atom indices of the atom-pair creating the coupling and the fourth column (scalar_coupling_constant) is the scalar coupling constant that we want to be able to predict

- test.csv - the test set; same info as train, without the target variable

- structures.zip - folder containing molecular structure (xyz) files, where the first line is the number of atoms in the molecule, followed by a blank line, and then a line for every atom, where the first column contains the atomic element (H for hydrogen, C for carbon etc.) and the remaining columns contain the X, Y and Z cartesian coordinates (a standard format for chemists and molecular visualization programs)
- structures.csv - this file contains the same information as the individual xyz structure files, but in a single file

The following data is provided for the molecules in train.csv only.

- dipole_moments.csv - contains the molecular electric dipole moments. These are three dimensional vectors that indicate the charge distribution in the molecule. The first column (molecule_name) are the names of the molecule, the second to fourth column are the X, Y and Z components respectively of the dipole moment.
- magnetic_shielding_tensors.csv - contains the magnetic shielding tensors for all atoms in the molecules. The first column (molecule_name) contains the molecule name, the second column (atom_index) contains the index of the atom in the molecule, the third to eleventh columns contain the XX, YX, ZX, XY, YY, ZY, XZ, YZ and ZZ elements of the tensor/matrix respectively.
- mulliken_charges.csv - contains the mulliken charges for all atoms in the molecules. The first column (molecule_name) contains the name of the molecule, the second column (atom_index) contains the index of the atom in the molecule, the third column (mulliken_charge) contains the mulliken charge of the atom.
- potential_energy.csv - contains the potential energy of the molecules. The first column (molecule_name) contains the name of the molecule, the second column (potential_energy) contains the potential energy of the molecule.
- scalar_coupling_contributions.csv - The scalar coupling constants in train.csv (or corresponding files) are a sum of four terms. scalar_coupling_contributions.csv contain all these terms. The first column (molecule_name) are the name of the molecule, the second (atom_index_0) and third column (atom_index_1) are the atom indices of the atom-pair, the fourth column indicates the type of coupling, the fifth column (fc) is the Fermi Contact contribution, the sixth column (sd) is the Spin-dipolar contribution, the seventh column (pso) is the Paramagnetic spin-orbit contribution and the eighth column (dso) is the Diamagnetic spin-orbit contribution.

The datasets provided seem to be useful at this stage. Each 'id' feature is representative of a combination of 'molecule_name', 'atom_index_0' and 'atom_index_1'. The datasets are presented such that information about each 'id' is found in different files. In order to create a single, rich and powerful dataset that can be fed into a machine learning model, the information from the different files will need to be pre-processed and eventually combined.

Pre-processing of the data will include the following steps:

- Handling of missing data and outliers
- One-hot encoding of the non-numerical (or categorical) variables
- Splitting of the training set into independent and target variables
- Feature scaling of variables using either rescaling, mean normalization, or standardization

The results of this project will be presented in the form of a .csv file format that contains the prediction of the scalar coupling constant for each ‘id’ in the test.csv file (i.e. the resulting file will consist of ‘id’ and ‘scalar_coupling_constant’ columns).

4 Solution Statement

To solve the problem at hand, a couple of solutions will be evaluated. First, a random forest will be used as it is considered a very robust model in general. Secondly, a deep neural network will be used to compare its results against the random forest. Both models will output a single prediction of the scalar coupling constant. The model that produces the better result (evaluated using the metrics stated in the Evaluation Metrics section) will be submitted to the Kaggle leaderboard.

5 Benchmark Model

Currently, there exists a physical method, called NMR, that allow measurements of many vicinal coupling constants in peptides, proteins, and other molecules. Coupling constants measured in these spectra can be used to determine backbone and side[U+2010]chain conformations, to obtain stereospecific resonance assignments of prochiral atoms, and to characterize conformational distributions of dihedral angles. Combined with information obtained from nuclear Overhauser effect measurements, these data will provide more precise determinations of protein solution structures by NMR spectroscopy.

Results from this project’s data-based approach can be compared to pre-existing results from NMR spectroscopy.

6 Evaluation Metrics

Kaggle has specified the evaluation metric for this competition to be the Log of the Mean Absolute Error (MAE), calculated for each scalar coupling type, and then averaged across types, so that a 1% decrease in MAE for one type provides the same improvement in score as a 1% decrease for another type.

$$score = \frac{1}{T}\sum_{t=1}^{T} log(\frac{1}{n_t}\sum_{i=1}^{n_t}|y_i - \hat{y_i}|)$$

7 Project Design

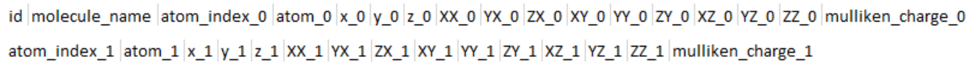
This section outlines theoretical work flow for the proposed solution.

7.1 Data Pre-processing

The datasets provided by Kaggle are useful but will need to be pre-processed before feeding them into a machine learning algorithm. The following will be performed on the datasets:

1. For every 'id' (a combination of molecule_name, 'atom_index_0' and 'atom_index_1'), the bits of information from the different files will be combined to form one single file in a .csv format. This file will have as its columns:
   - id
   - molecule_name
   - atom_index_0
   - atom_index_1
   - dipole_moments_x (from dipole_moments.csv)
   - dipole_moments_y (from dipole_moments.csv)
   - dipole_moments_z (from dipole_moments.csv)
   - potential_energy (from potential_energy.csv)
   - fc (from scalar_coupling_contributions.csv)
   - sd (from scalar_coupling_contributions.csv)
   - pso (from scalar_coupling_contributions.csv)
   - dso (from scalar_coupling_contributions.csv)
   - scalar_coupling_constant (from train.csv)
2. The magnetic shielding tensors (XX, YX, ZX, XY, YY, ZY, XZ, YZ, ZZ), the mulliken_charge, and the XYZ molecular structure, found in magnetic_shielding_tensors.csv, mulliken_charges.csv and structures.csv respectively, are relevant to each atom_index in a molecule_name, rather than being relevant to the 'id'. As such, in order to combine these variables into the dataset discussed in point #1 above, atom_index_0 and

atom_index_1 will each have their own set of magnetic shielding tensors, mulliken charges and XYZ structure. For example, a row in the dataset will look something like this, along with the columns mentioned above:



id molecule_name atom_index_0 atom_0 x_0 y_0 z_0 XX_0 YX_0 ZX_0 XY_0 YY_0 ZY_0 XZ_0 YZ_0 ZZ_0 mulliken_charge_0
atom_index_1 atom_1 x_1 y_1 z_1 XX_1 YX_1 ZX_1 XY_1 YY_1 ZY_1 XZ_1 YZ_1 ZZ_1 mulliken_charge_1

7.2 Data Cleaning

1. Now that all the data is in a single file, the next step is to look for missing values in the data. The preferred way to handle missing data will be to delete any row of data that has missing or NaN values. However, if this results in more than 50% of the data being removed, then other methods of handling missing data will need to be evaluated (such as replacement using the mean/min/max of that particular column, or copying the value of the column from the next row).
2. Outliers for each column will be identified and the respective row of data will be removed.
3. All independent variables will scaled by one of the following methods - rescaling, mean normalization, or standardization
4. The dataset will then be split into dependent (target) and independent variables. The target variable that we are trying to predict is the scalar_coupling_constant, and all other variables are the independent variables.

7.3 Domain Knowledge

Due to my lack of domain knowledge in this field, some time will be spent on improving my understanding of molecular structure and dynamics with extra attention to scalar coupling.

7.4 Feature Engineering

Based on the domain knowledge gained, new features will be engineered from the existing data. The specifics of what features will be engineered are undecided at this stage; these will be decided with the aid of domain-specific knowledge.

7.5 Model Selection

To solve the problem at hand, two models will be created and evaluated:

1. Random Forest
2. Deep neural network

The hyperparameters of the above models will be tuned and refined during the course of the implementation.

Bibliography

[1] CHAMPS (CHemistry And Mathematics in Phase Space). (2019, May 31). Predicting Molecular Properties
Retrieved from *https://www.kaggle.com/c/champs-scalar-coupling/overview*

[2] Gaetano T. Montelione, S. Donald Emerson, Barbara A. Lyons. (1992, April). A general approach for determining scalar coupling constants in polypeptides and proteins.
Retrieved from *https://onlinelibrary.wiley.com/doi/abs/10.1002/bip.360320406*

[3] Microsoft Azure Machine Learning: Algorithm Cheat Sheet. (2015).
Retrieved from *https://docs.microsoft.com/en-us/azure/machine-learning/studio/algorithm-cheat-sheet*

[4] Best Practices for Feature Engineering. (2017, July 26).
Retrieved from *https://elitedatascience.com/feature-engineering-best-practices*