Machine Learning for Molecular Engineering

Problem Set 6

Date: April 29, 2023
Due: 11:59 PM ET on Thursday,May 11, 2023

Instructions This is the final problem set for the undergraduate version of this course (3.C01, 10.C01, 20.C01). This exercise is a supervised learning competition hosted on Kaggle. Here is the link to the competition. For the submission, you will need to submit a notebook containing your code and a short writeup to describe your solutions; the writeup can be integrated with your notebook in a markdown cell. You can find the template notebook here. You can also submit a separate notebook as the solution to the machine learning competition. **Because this is the final pset, you will need to use your discretion in deciding how to tackle these problems; we have not specified exactly how each part should be approached.**

Background

Solvation Free Energies

Solvation free energies (SFEs) are the differences in the thermodynamic potential of a mixed solute molecule and a solvent reservoir and a separated solute molecule and solvent reservoir. The SFE can be used to describe the relative population of a chemical species in solution and vacuum at thermodynamic equilibrium. SFEs also provide insight into how a solvent behaves in different environments and how it partitions between two different environments, e.g., hydrophilic blood and hydrophobic tissue for a drug. Some of the most important physicochemical properties for drug metabolism are solubility, permeability, clearance, volume of distribution, and half-life [1], which all depend on partition coefficients that can be derived from SFEs. Improving our capability to estimate SFEs accurately will help us better predict these properties and design more efficient drugs.

However, the accuracy of predictions for solvation free energies can be poor. This is true of both *ab initio* and data-driven approaches. Predicting the solvation energy of ionic solutes (i.e., charged species) is particularly challenging [2]. In this pset, you will explore a machine learning solution to predict partition coefficients given a pair of solvent and solute. This is similar to other quantitative structure-property relationship tasks we've seen in the course, but crucially it depends on the *interaction* of two molecules, not just one molecule.

The partition coefficient data we will use are from the "Solv@TUM" database [3]. The authors calculated the partition coefficients from published experimental infinite dilution activity coefficients, Henry's law constants, and mole fraction solubility data using thermodynamic relationships. We have provided a training set with four columns: Solvent (SMILES), Solute (SMILES), $\log_{10} K$ (partition coefficient), ΔG (solvation free energy). The solvation free energy and $\log K$ are related by the follow mathematical relation:

$$\Delta G = -kT \log K \tag{1}$$

where K, the equilibrium constant, is defined as the relative population:

$$K = \frac{[\text{solute}]_{\text{solvent}}}{[\text{solute}]_{\text{air}}} \tag{2}$$

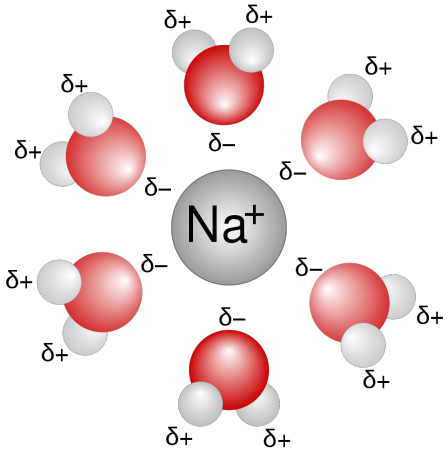

Figure 1: A sodium ion solvated by water molecules. Source: Wikipedia

Registering for the competition

Go to kaggle.com to register for an account. After you register successfully, click here to join the competition. You can submit your solutions to the test set and your result will be displayed on the leaderboard. If you prefer anonymity, you can choose a nickname like `molmlmaster` rather than your real name; if you do this, please include your nickname in your final writeup. Note that you are not graded based on your ranking but a combination of different factors which will be described below. Our competition will have no cash prizes, but you can explore other competitions on Kaggle for potential cash prizes and employment opportunities.

Part 1: (30 points) Baseline Regression Methods

This section provides instructions to get you started, and will help you train a couple of baseline ML methods for the prediction task. You don't need to submit your predictions to Kaggle for this part.

Part 1.1 (10 points) Prepare Dataset

To get the data, you can either go to the Data section in Kaggle or use the code we provided to download the data. There are 3 files you can download:

`solvation_train.csv`: the training set. There are four columns: `"Solvent"`, `"Solute"`, `"logK"`, `"delG_solv"`, which are the solvent molecule SMILES, solute molecule SMILES, partition coefficient, and SFE.

`solvation_test.csv`: The solvent/solute SMILEs used to generate solutions for the test set.

`mol_prop.csv`: For each molecule, we have also provided its molecular weight, dipole moment, and polarizabilities, which you can use to generate features to train a model.

We have provided some utility functions for you to load and generate features which you can use to train your models. Your first step is to load and get familiar with your data. Write code to build a feature set that combines physical descriptors for solvents and solutes. You will need

to concatenate solvent and solute features into a feature vector for each solvation/solute pair, or develop some other means of describing information about both solute and solvent. Be sure to check for NaNs!

Part 1.2 (10 points) Linear Regression

Apply linear regression on the training data to predict $\log K$. You may want to normalize your feature set during training, like you did in Problem Set 1. Report the 5-fold cross-validated R^2 score.

Part 1.3 (10 points) MLP Regression

Use the same input features as the previous part to train an MLP regressor to predict $\log K$. Report the 5-fold cross-validated R^2 score.

Part 2: (70 points) Machine Learning Competition and Report

In this part, you will apply the techniques you learned in this class to propose machine learning solutions to a supervised regression problem. Based on the training data we provided to you, you will try to make predictions for the held-out test data. We have provided the test feature set in `solvation_test.csv`. You will use your model to predict $\log K$ using this test feature set. The evaluation of test performance will be handled by Kaggle, and you can see your results in a leaderboard. The goal is find the best model possible.

You can submit up to 20 solutions to Kaggle per day. We have provided a utility function for you to generate a submission file. Your answers will be compared with the true test labels on Kaggle and your model performance will be ranked in a leaderboard. The metric we use is the R^2 score, ranging from 0 to 1. The reported public score on Kaggle is calculated with only 40% percent of the test data, but your performance will be graded on the private score based on the other 60% of the test data; this is to prevent you from tuning performance to the test data (which is bad practice!). We will release the final performance values when the competition is over.

Please note that in your representation or in your model architecture, you will need some way of learning about the *interactions* between the solute and solvent molecules. How you've accomplished this must be addressed explicitly in your writeup.

You need to submit your commented code (as a `.ipynb` notebook) with a writeup that addresses the following points:

1. Data preprocessing
2. Choice of feature representations

   You are provided with molecular SMILES and a very small number of physical properties for both solvent and solute molecules. You can represent them with SMILES strings, circular fingerprints, molecular graphs, other RDKit descriptors, or look at descriptor-calculating packages like Mordred. It's up to you! Describe your choice of molecular representation and briefly explain your reason.
3. Choice of model architecture

You are expected to try at least two models and select the best-performing model to submit the best solution. If you are designing a novel model in PyTorch, describe your model architecture. For all your models, report cross-validation scores or some other rigorous metric of performance.[1] You need to provide a brief description of your model choice or design and mention any open-source code/packages you used. If you decide to adopt a model architecture or method from a paper, please include the reference in your writeup.

4. Model evaluation and selection

   Describe how you performed hyperparameter search. Describe how the model performance is evaluated to select the best hyperparameter.

Grading Rubric

(25 points) Creativity

There are many modeling choices for regression problems like this. You can use linear regression, random forests, support vector machines, gradient boosted trees, and neural networks. You are encouraged to survey the literature for inspiration. For example, in [4], the authors applied two separate solvent and solute encoder networks that quantify structural features of the given compounds via word embedding and recurrent layers. The two networks are also coupled with an attention mechanism. Another graph-based approach is described here. **Warning:** You should not merely copy publically available code.

We encourage you to use PyTorch in your solutions. This will let you build more complex models that are optimizable by gradient descent. For example, you can augment your MLP architecture with an attention mechanism, or play around with how solute and solvent features are combined (e.g., concatenated? added? outer product?).

(35 points) Technical correctness

We will examine your code and text to evaluate your solution on technical correctness and the appropriateness of your chosen methods. Please comment and document your code so that we understand your approach to data preprocessing, train/validation/test split, hyperparameter optimization, and cross-validation. It is always important to compare your model performance to that of appropriate baseline methods. While we don't expect every idea you come up with to outperform the baseline, we do expect you to make this comparison and accurately report the resulting performance of your models.

(15 points) Model performance (10 + 5 points)

We have provided a linear regression baseline and 3 additional TA-prepared models as benchmarks. The linear regression baseline is similar to something you might have done for Part 1.2. You can earn up to 10 points plus 5 bonus points on this part, depending on your performance as follows:

1. Beat or tie the linear baseline - between 2 and 6 points.
2. Beat or tie 1 TA benchmark - between 6 and 10 points.

[1] If you have a very good model but cross-validation is too compute-intensive, you can report a test set performance instead.

3. Beat or tie 2 TA benchmarks - 10 points, with up to 5 bonus points possible.

4. Beat or tie 3 TA benchmarks - 10 points plus 5 bonus points; potentially more bonus points if you impress us.

References

[1] Kroger, L. C., Muller, S., Smirnova, I. & Leonhard, K. Prediction of solvation free energies of ionic solutes in neutral solvents. *The Journal of Physical Chemistry A* **124**, 4171–4181 (2020).

[2] Subramanian, V. *et al.* Multisolvent models for solvation free energy predictions using 3d-rism hydration thermodynamic descriptors. *Journal of chemical information and modeling* **60**, 2977–2988 (2020).

[3] Hille, C. *et al.* Generalized molecular solvation in non-aqueous solutions by a single parameter implicit solvation scheme. *The Journal of chemical physics* **150**, 041710 (2019).

[4] Lim, H. & Jung, Y. Delfos: deep learning model for prediction of solvation free energies in generic organic solvents. *Chemical science* **10**, 8306–8315 (2019).