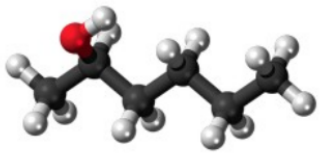
Implementation project
Lab Machine Learning for Data Science
Summer Semester 2023

G. Montavon
Institute of Computer Science
Department of Mathematics and Computer Science
Freie Universität Berlin

Lab ML for Data Science: Part II

Getting Insights into Quantum-Chemical Relations



The goal of Part II of the project is to extract quantum-chemical insights from a dataset of molecules. We will focus on the problem of understanding the relation between the geometry of a molecule and its electronic properties. Such understanding enables the principled design of new molecules (e.g. potentially facilitating the design of new drugs or materials). For the purpose of this project part, in particular, to ensure that the experiments can be carried out easily, we will consider a reduced scenario where the data consists of small molecules, and we will limit ourselves to a specific electronic property, the atomization energy. The atomization energy of a molecule is the energy generated by dissociating all atoms from the molecule, i.e. moving atoms far apart so that the bonds between atoms are broken. Because it consumes energy to break these bonds, the atomization energy is typically a negative quantity.

Unlike most real-world data, quantum chemical data does not need to be acquired through physical observation. There exists many physics-based computational methods to calculate the atomization energy associated to a given molecular structure (the atoms positions). These physics-based calculations, although computationally expensive, deliver very accurate results. However, the sequence of computations these physics-based methods perform is intrincate and does not allow for extracting an interpretation of the chemical relation of interest, for example, what substructures in the molecule, e.g. atoms, bonds or chemical groups, can be associated to low or high atomization energy.

The purpose of using a ML approach is therefore not only to provide a computationally effective way of predicting the chemical relation of interest but also to identify what is needed in the molecular structure to be able to predict, and, using Explainable AI, what exact substructures are being used for prediction.

1 The QM7 Dataset

Our starting point is a simple quantum-chemical dataset, the QM7 dataset, which can be downloaded here:

```
http://quantum-machine.org/data/qm7.mat
```

(It is provided as a Matlab file, and it can be loaded in Python using the function `scipy.io.loadmat`.) The QM7 dataset consists of 7165 organic molecules, each of which is composed of up to 23 atoms. The 3d coordinates of each atom in each molecule are available in the variable R. It is an array of size $7165 \times 23 \times 3$ containing for each molecule and atom a triplet representing the 3d coordinates. The variable Z is an array of size 7165×23 which gives for each molecule and atom of the molecule the corresponding atomic number. An atomic number of 1 corresponds to a hydrogen atom (H), the number 6 corresponds to carbon (C), the numbers 7 and 8 to nitrogen (N) and oxygen (O) respectively, and finally, the number 16 corresponds to sulfur (S). If the number is zero, then it indicates that there is no atom at this index, and the corresponding 3d coordinate should therefore be ignored. This allows for representing in the same array molecules of different sizes. In addition to these geometrical features of the molecule, the dataset also provides for each molecule its atomization energy (computed via quantum-chemical simulation). These atomization energy values are stored in the variable T, an array of size 7165.

1.1 Visualizing Molecules

There are a variety of libraries for rendering molecular geometries with various degrees of sophistication. A quick and dirty approach is to use the scatter plot function of matplotlib, where each point is an atom (e.g. plotted according to its xy-coordinates and discarding the z-coordinate). Note that bonds are not provided as part of QM7 because they are strictly speaking not needed to infer chemical properties (bonds can be derived from atom coordinates). To better visualize the molecule, one can draw connections between nearby atoms by plotting a line between two atoms if the Euclidean distance between the two is smaller than a fixed threshold.

2 Data Representation, ML Model and Explanations

To predict the relation between molecular geometry and atomization energy, we will consider a ridge regression model with a particular summing structure. This model will have the advantage of being reasonably accurate and enabling an insightful exploration of the predicted quantum-chemical relation.

2.1 Data Representation

Let $\mathcal{M} = (\mathcal{E}_1, \ldots, \mathcal{E}_{|\mathcal{M}|})$ be a molecule viewed as a collection of $|\mathcal{M}|$ elements. It can be for example the collection of atoms forming the molecule. Let ϕ be a function that maps each element into a vector representation $\phi(\mathcal{E}_i) \in \mathbb{R}^h$. When the elements are individual atoms, a possible choice for ϕ is a one-hot encoding of the atom type i.e. $\phi(\mathcal{E}_i) = (1_{\mathcal{E}_i=\mathrm{H}}, 1_{\mathcal{E}_i=\mathrm{C}}, \ldots, 1_{\mathcal{E}_i=\mathrm{S}})$. For a given molecule, one can then build an overall vector representation by summing the representation of each element:

$$\boldsymbol{x} = \sum_{i=1}^{|\mathcal{M}|} \phi(\mathcal{E}_i) \tag{1}$$

An interesting property of this feature vector is that it is *invariant* to the order of indexing of elements. For example, if we shuffle the atoms, the representation remains the same. If we would have used a concatenation operator instead of a summing operator in Eq. (1), such invariance property would not be present.

2.2 Ridge Regression Model

Once the representation has been built, it can be used as input for a ML model, for example, a simple linear model:

$$f(\boldsymbol{x}) = \boldsymbol{w}^\top \boldsymbol{x}$$

where $\boldsymbol{w} \in \mathbb{R}^h$ is a parameter vector that needs to be learned. We propose to learn this vector with ridge regression, which consists of defining the minimization objective:

$$J(\boldsymbol{w}) = \mathbb{E}[(\boldsymbol{w}^\top \boldsymbol{x} - t)^2 + \lambda \|\boldsymbol{w}\|^2]$$

where $\mathbb{E}$ is an expectation over all molecules of the training set. In the ridge regression framework, it is necessary to center the data as well as the target values. One can then show that minimizing this objective yields the closed form solution

$$\boldsymbol{w} = (\Sigma_{\boldsymbol{xx}} + \lambda I)^{-1} \Sigma_{\boldsymbol{x}t}$$

where $\Sigma_{\boldsymbol{xx}} = \mathbb{E}[\boldsymbol{x}\boldsymbol{x}^\top]$ and $\Sigma_{\boldsymbol{x}t} = \mathbb{E}[\boldsymbol{x}t]$ are auto- and cross-covariance matrices respectively. Note that there are several choices one needs to make in this framework. First, the way the molecule is decomposed into individual elements (e.g. whether the elements are atoms of the molecule or pairs of atoms). Second, the way the elements are encoded into vectors, i.e. the choice of the feature map ϕ. Lastly, one needs to choose a good regularization parameter λ. This can be achieved by testing multiple values of λ on a logarithmic scale, and retaining the value of λ that minimizes the prediction error on a set of data disjoint from that used for learning the model.

2.3 Deeper Insights with Explanations

While the procedure above would allow us to determine whether the molecular property is predictable from the data given as the input, an important insight on its own, this result is of limited use as it does not tell us *what* in the molecule has contributed to the low/high energy prediction. Interestingly, this an explanation to this question is readily given by looking at the structure of the model:

$$\begin{aligned} f(\boldsymbol{x}) &= \boldsymbol{w}^\top \boldsymbol{x} \\ &= \boldsymbol{w}^\top \Big(\sum_{i=1}^{|\mathcal{M}|} \phi(\mathcal{E}_i) \Big) \\ &= \sum_{i=1}^{|\mathcal{M}|} \underbrace{\boldsymbol{w}^\top \phi(\mathcal{E}_i)}_{R_i} \end{aligned} \tag{2}$$

The term R_i indentifies the contribution of the element i of the molecule to the overall prediction. Lastly, expressing R_i as a function of the element's property, we can identify how a modification of the given element (e.g. a change of atom type) affects the contribution of that element to the molecular property.

3 Experiments

We now would like to apply the method above in practice on the QM7 dataset. For this, one needs to decide how the molecule should be decomposed into distinct elements, and how these elements should be represented. The first experiment will consider a decomposing the molecule into individual atoms, and

it allows us to produce a classifier with a mean absolute error of approximately 15 kcal/mol. The second experiment will consider a decomposition in terms of pairs of atoms. This will enable the prediction error to go down to 6 kcal/mol.

3.1 Simple atom-based Representation

We decompose each molecule $\mathcal{M}$ into its set of atoms. Then for each atom $\mathcal{E}_i$, we generate a feature representation. A reasonable choice is the so-called one-hot representation which generates different feature vectors for different atom types:

$$\phi(\mathcal{E}_i) = \begin{pmatrix} I(\mathcal{E}_i = \mathrm{H}) \\ I(\mathcal{E}_i = \mathrm{C}) \\ I(\mathcal{E}_i = \mathrm{N}) \\ I(\mathcal{E}_i = \mathrm{O}) \\ I(\mathcal{E}_i = \mathrm{S}) \end{pmatrix} \in \mathbb{R}^5$$

For example, a carbon atom is represented by the vector $(0, 1, 0, 0, 0)$, and a sulfur atom is represented by the vector $(0, 0, 0, 0, 1)$. This atom representation can now be used to generate the feature vector $\boldsymbol{x}$ following Eq. (1). For example, the simple molecule CH_4 (methane) is mapped to the vector $\boldsymbol{x} = (4, 1, 0, 0, 0)$.

Now, one can learn the ridge regression model, choose an appropriate regularization parameter λ, measure and report the prediction error one some test set disjoint from the training and validation sets. Finally, deeper insights on the structured of the prediction function can be obtained identify the summands given in Eq. (2). Using the method of Section 2.3, you can try to identify how the different atom types contribute individually to the predicted energy. Compare the produced insights with existing chemical knowledge or the literature.

3.2 Models with Pairs of Atoms

We now decompose each molecule into its set of *pairs* of atoms, i.e. $\mathcal{E}_i$ now denotes a *pair* of atoms, and a molecule contains $\frac{\#\text{atoms}\cdot(\#\text{atoms}-1)}{2}$ such pairs. A reason to consider pairs of atoms instead of individual atoms is to take into consideration the mutual distances between atoms. Like for atoms, it is possible to generate a one-hot encoding of distances, i.e. binning them into mutiple intervals $[\theta_1, \theta_2], [\theta_2, \theta_3], \ldots, [\theta_{m-1}, \theta_m]$, and generating the vector:

$$\phi^A(\mathcal{E}_i) = \begin{pmatrix} I(\theta_1 \leq \mathrm{dist}(\mathcal{E}_i) < \theta_2) \\ I(\theta_2 \leq \mathrm{dist}(\mathcal{E}_i) < \theta_3) \\ \vdots \\ I(\theta_{m-1} \leq \mathrm{dist}(\mathcal{E}_i) < \theta_m) \end{pmatrix}$$

In practice, to avoid introducing unnatural discontinuities into the model, the hard indicator function I can be replaced by a soft indicator function (e.g. a Gaussian function with mean at the center of the interval and fixed variance). Like for the regularization parameter λ, the size of intervals and the scale of the Gaussian function (if applicable) need to be chosen appropriately, e.g. searching for the configuration that minimizes the error on validation data.

The newly defined distance-based feature representation so far does not look at the atom types. In

order to incorporate atom types, one can generate another feature representation of atom types:

$$\phi^B(\mathcal{E}_i) = \begin{pmatrix} I(\text{type}(\mathcal{E}_i) = \text{HH}) \\ I(\text{type}(\mathcal{E}_i) = \text{HC}) \\ I(\text{type}(\mathcal{E}_i) = \text{HN}) \\ \vdots \\ I(\text{type}(\mathcal{E}_i) = \text{SS}) \end{pmatrix}$$

i.e. a feature map of $6 \cdot 5/2 = 15$ dimensions. Here, we have considered unordered pairs of atom types in order to implemenent invariance to the indexing.

Lastly, the feature maps ϕ_A and ϕ_B can be combined into one 'big' feature map consisting e.g. of all products between elements of the two feature maps, i.e. a matrix $\phi(\mathcal{E}_i)$ with elements

$$[\phi(\mathcal{E}_i)]_{jk} = \phi^A_j(\mathcal{E}_i) \cdot \phi^B_k(\mathcal{E}_i)$$

for all j and k. Note that the matrix $\phi(\mathcal{E}_i)$ is then flattened into a vector for the subsequent processing steps. A ridge regression model can now be trained on the newly defined molecule representation. To get a good model parameters, you can proceed as in Section 3.1.

Furthermore, in comparison to the insights one could get in Section 3.1, additional insights can now be built from the enhanced feature representation. We no longer have a single contribution per atom type, but for each pair of atom type. Furthermore, each pair of atom type contributes as a function of the distance between the two atoms. These are called *pairwise potentials* and they can be plotted as a function of the distance. Make these plots, and check whether the pairwise potentials you obtain for each pair of atom types are consistent with chemical knowledge, e.g. whether they reflect typical bond lengths between different atoms.