User Guide for CSE700 Final Project: Kernel Ridge Regression

Uriel Garcilazo Cruz and Asma Jamali

April 8, 2025

Contents

1 Sinopsis

Our project focuses on the implementation of global and local matrix representations of molecular properties (HOMO-LUMO gap and heat capacity) taken from the QM9 dataset. We use them in a comparative framework to evaluate the effects of trimming eigenvalues during the implementation of kernel ridge regression (KRR). Along with our documentation we have developed a Python package to go along with this implementation.

Our software **krr** is an implementation of Kernel Ridge Regression (KRR) with support for custom kernels (Tanimoto, Dice, Gaussian and Laplacian) to predict molecular properties (e.g., HOMO-LUMO gap and heat capacity) from the QM9 dataset. The code integrates eigenvalue truncation to optimize performance.

2 How to run Python package

A functional folder for our application — with all the elements required to run the code — should have the following arrangement:

```
.
|-- CSE700
|   |-- Dataset
|   |   |-- bob_rep.npy
|   |   |-- coulomb_matrix_rep.npy
|   |   `-- dataset.pkl
|   |-- krr
|   |   |-- Kernel_ridge_regression.py
|   |   `-- kernels_library.py
|   |-- main.py
|   `-- setup.py
```

Due to the size of the files contained in the folder Dataset, which is large relative to the conventional file size accepted by GitHub, the process of building the folder structure shown above requires two steps:

2.1 Clone the GitHub repository

In your terminal, navigate to the directory where you want to clone the repository and run:

```
git clone https://github.com/Asma-Jamali/CSE700.git
```

If you encounter any issues, please use your browser to navigate to the GitHub repository containing our KRR algorithm:

https://github.com/Asma-Jamali/CSE700/tree/KRR

In there, you will find the option to clone the repository. You can do this by clicking on the green button labeled "Code" and copying the URL provided. Keep in mind that you will need to have Git installed on your computer to clone

the repository. If you don't have Git installed, you can download the repository as a ZIP file by clicking on the "Download ZIP" option in the same menu. This is especially useful for Windows users who may not have Git installed.

2.2 Download the dataset

The dataset is too large to be uploaded to GitHub. To download the dataset, please use the link below:

LINK TO DATASET

The dataset is password protected. Due to one of our developer's expertise being arachnology, the password is **Maratus_volans**.

Once the dataset has been downloaded, place it inside the folder and at the same level as main.py (control module).

2.3 Installing the package

The program is now in our local directory, but it's still not ready to run. There are multiple libraries that need to be installed. Luckily, Python has a straightforward way to ensure the interpreter can install such dependencies. Navigate to the folder where setup.py is located in the terminal and run:

```
pip install -e .
```

This will create an editable installation of the package, allowing you to modify the code and see the changes immediately without needing to reinstall the package. To install the package, you need to have Python version ¿=3.9, and pip installed on your system. If you don't have them installed, you can download Python from the official website: Python Downloads. Pip is included with Python installations. We also recommend using a virtual environment to avoid conflicts with other packages.

To execute the code, navigate to the folder where main.py is located in the terminal and run:

```
python main.py
```

If all the dependencies are installed correctly, the program should run without any issues.

2.4 Inputs and Outputs

The program takes the following inputs:

For the current version of the code, the user only need to specify the location of the picked dataset file, and the location for the output files:

```
# in main.py:
dataset_path = 'Dataset/dataset.pkl'
output_path = '/results/'
```

This data is a type of numpy array; a highly efficient data structure for storing large amounts of data.

The program will then load the dataset and use it to train the KRR model. The model will be trained using the specified kernel (Tanimoto, Dice, Gaussian, or Laplacian) and the specified eigenvalue truncation method (if applicable). The program will then output two csv files:

- **results.csv**: This file contains the results of the KRR model, including the predicted values and the actual values.
- **eigenvalues.csv**: This file contains the eigenvalues used in the KRR model, including the trimmed eigenvalues (if applicable).