Fundamentals of Machine Learning
Support Vector Machines, Practical Session

Masters DSC/MLDM

Assignment Details

- **Due date:** Short summary of your data: February 24, 2025, 23:59 on Moodle.

  Project report: March 17, 2025, 23:59 on Moodle (hard deadlines).
- **Groups:** 3 students per group.
- **Format:** ZIP archive containing your work for Part 1 and Part 2:
  - PDF report (4 to 5 pages, including figures).
  - Jupiter notebook or python script. Requirements file with the libraries used.

1 Directed Practical Session

1.1 Little Warm-Up

To familiarize yourself with the behavior of SVM, you can look at the following demos:

- `http://cs.stanford.edu/people/karpathy/svmjs/demo/`
- `https://www.csie.ntu.edu.tw/~cjlin/libsvm/`

1.2 SVM with Scikit-Learn

We will use the `SVC` class from Scikit-Learn. Important options:

- `C`: penalty parameter (default 1.0).

- kernel: options are linear, poly, rbf, sigmoid.
- gamma: option for RBF kernel (default: 1/n_features).
- degree: polynomial kernel degree (default: 3).
- decision_function_shape: ovr or ovo for multiclass.

1.3 Datasets

1.3.1 Random Datasets

```
# Example Python code to generate synthetic data
from sklearn.model_selection import train_test_split
import numpy as np
import pandas as pd

# Generate data...
np.save('./data/generated_data.npy', data)
pd.DataFrame(data).to_csv('./data/generated_data.csv', index=False)
```

1.3.2 Real Dataset: Pulsar Classification

- Apply cross-validation to find the best hyperparameters.
- Use the SimpleImputer to handle missing values.

2 Go imbalanced!

Find a dataset with an imbalanced problem (predominant label in an imbalanced dataset is called the majority class; the less common label is called the minority class). Run experiments using SVM to find the best predictions.

Or choose one of the data sets below.

ATTENTION: Some datasets have multiclass target variables. You can choose one of the classes to make it binary (keep in mind it has to be an imbalanced dataset), or you use ovr or ovo for multiclass.

- Credit card fraud detection

  https://www.kaggle.com/datasets/samuelcortinhas/credit-card-approval-clean-data

- Dry Beans Classification

  https://archive.ics.uci.edu/dataset/602/dry+bean+dataset

- Forest Cover Types

  `https://archive.ics.uci.edu/dataset/31/covertype`

- Email Spam Detection

  `https://www.kaggle.com/datasets/balaka18/email-spam-classification-dataset-csv/data`

- KDD Cup 1999 Data

  `https://www.kaggle.com/datasets/galaxyh/kdd-cup-1999-data`

3 Deliverables

- Dataset exploration and justification (due before February 24th). If you have chosen of the suggested datasets, you must present a summary of the dataset.
- Report (4-5 pages) detailing methodology, experiments, and results.
- Jupyter notebook (or python script), requirements file.

It is recommended to use the libraries numpy, pandas, matplotlib, seaborn, scikit-learn, imbalanced-learn.