RESEARCH

Paper 7 in applied machine learning

Aakash Nepal and Cuong Gia Pham

Full list of author information is available at the end of the article
*Equal contributor

Abstract

Goal of the project: In this project, we were able to get familiar with proteomics datasets. Using the PyOpenMS library[1] and binning feature engineering method, the information of LC-MS/MS (Liquid Chromatography with tandem mass spectrometry) is converted into an appropriate form, which could be applied in machine learning models.

Main results of the project: The performance of GradientBoostingClassifier surpassed the PassiveAggressiveClassifier model in all the criteria.

Personal key learning :

1 Pham Gia Cuong: PyOpenMS, feature engineering method.
2 Aakash Nepal: Data augmentation with SMOTE, binning, feature importance, PyOpenMS

Estimation working hours:

1 Pham Gia Cuong: 7 hours per week
2 Aakash Nepal: 7 hours per week.

Project evaluation: 2.7 out of 5

Number of words : Approx. 1550(without abstract, captions, references)

1 Goal of the project

The main goal of the project is to develop two machine learning models for the classification of the extracted features which are derived from the LC-MS/MS proteomics dataset by using the FeatureFinder algorithm of pyOpenMS. For this, the relevant classes were selected and two ML classifiers, namely GradientBoostingClassifier and PassiveAggressiveClassifier were trained on those extracted features from the pyOpenMS to classify the samples from the given benchmark dataset from the paper of Wessels et al.[2]. The models were evaluated using the performance measures such as accuracy, sensitivity, specificity, and F1-score described in Table 1 of the paper by Rauschert et al.[3]. Furthermore, a feature importance analysis was also performed to understand the biological relevance of the most important features found by using those machine learning methods. Moreover, the classification model can play an important role in computational proteomics in many areas such as biomarker discovery, and drug development. With feature importance analysis, we can gain insights into the relationships within data and help understand the potential proteins which might aid in personalized medicine.

2 Data

A well-designed comprehensive LC-MS/MS collected data is an essential basis for developing and benchmarking innovative analytical techniques. In this project, we worked with a benchmark dataset containing LC-MS/MS measurements from 50

Escherichia coli K12 protein samples. Different amounts of protein bovine carbonic anhydrase II (CA) and/or chicken ovalbumin (OVA) were added to the samples giving rise to 9 different classes. The measurements were made with nanoflow reversed-phase LC and a linear ion trap Fourier-transform ion cyclotron resonance MS. The raw data were converted to mzXML format, and Mascot-compatible peak lists were generated for database searches as part of the preprocessing steps. For this project, we chose 10 samples without CA and OVA (CA = 0, OVA = 0) as the control class and 20 samples with CA and OVA (CA = H or L, OVA = H or L) as CA/OVA class referring to the Table 2 which is available in the data deposition documentation[4].

3 Methods

3.1 Extracting features, features engineering, and data augmentation

To get LC-MS/MS proteomics data, biological samples(e.g., blood and urine) are collected from the patients. After that, the workflow involves preparing the samples, chromatographic separation such as LC, ionization with methods such as electrospray ionization, mass analysis, LC-MS/MS, data acquisition and processing, and subsequent data analysis and interpretation. Upon downloading the LC-MS/MS proteomics benchmark data and comprehending the requirements, we utilized the PyopenMS library to extract the peaks, also known as features, in mass spectrometry. By scanning through each mzXML file, PyopenMS detected these peaks. Each peak was then encoded based on its retention time (RT) and mass-to-charge ratio (m/z). Next, employing a binning feature engineering technique with 25 bins in RT and 25 bins in m/z, we organized the values of each RT and m/z pair into their respective bins. Figure 1 illustrates this process: Figure 1A depicts a scatter plot of the peak data frame for one sample(EC10), while Figure 1B showcases the plot after binning the data. In this representation, each bin corresponds to an interval

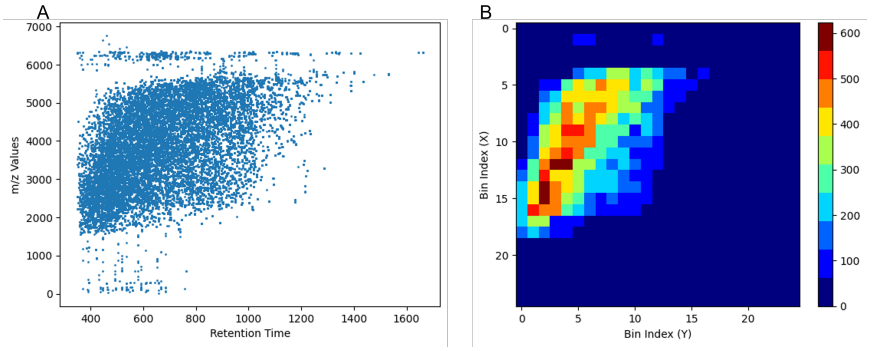

Figure 1 Scatter plots: Figure 1A is a scatter plot of RT and m/z values from the extracted data(sample EC10), Figure 1B is its color-bin plot of RT and m/z values.

of RT and m/z values. The value within each bin indicates the number of RT and m/z pairs that fall within that interval. The intensity of a bin reflects the density of RT and m/z pairs within that specific interval. After that, each bin matrix is flattened and applied to train and test machine learning models. From the original paper, we know that the dataset is small and the classes are imbalanced, where the control group has only 10 samples and the group which used CA/OVA has only 20

samples, hence to overcome this burden, we used SMOTE library to generate more samples. After that, each class has 60 samples. The dataset is then split to train and test set with a ratio of 3:2.

3.2 Machine learning models

For this project, we employed two machine-learning models, including the GradientBoostingClassifier and the PassiveAggressiveClassifier.

GradientBoostingClassifier is an ensemble learning method, which combines many weak learning models together to create a strong predictive model. Hence, first, the algorithm starts by initializing the "ensemble" as an empty model, typically a decision tree with a single node. The first weak model is then trained on the training data to predict the target labels. The difference between the actual target labels and the predicted values is then calculated, this step is also called the calculation of the loss function. The model is then optimized by minimizing the loss function through gradient descent, adjusting the model's parameters to reduce errors, and building the next node based on that.

The PassiveAggressiveClassifier is a machine learning algorithm, which is mostly used for online learning tasks, particularly in scenarios where data arrives sequentially or in a streaming fashion It belongs to the family of linear classifiers and is effective for binary classification. The algorithms try to minimize a hinge loss function, which measures the margin between the predicted class label and the true class label. It aims to maximize the margin while penalizing misclassifications. The key characteristic of the PassiveAggressiveClassifier lies in its adaptive update mechanism, which allows it to handle changing data distributions and update the model incrementally.

3.3 Training and model evaluation

We used the gridsearchcv function from the sci-kit learn package to perform 5-fold cross-validation for identifying optimal hyperparameters for each machine learning model in this project. A ROC curve with confidence intervals was created by using 5-fold cross-validation to assess the performance of the model on the training and testing datasets. The metrics such as precision, sensitivity, and F1-score were also calculated for evaluating the model. Briefly, the precision measures the accuracy of the positive predictions, sensitivity indicates the capacity of the model to properly identify the instances that are positive, and the F1-score combines both precision and sensitivity into one measure which provides a balanced assessment of the accuracy of the model.

3.4 Features importance

For finding features importance, we first used the build-in function of sklearn called coef_ to have the coefficient of each column from the flattened matrix. Each column in the flattened matrix represents a bin in the bin matrix, which is equal to a range of values, in which many RT and m/z pairs fall in as we explained above. From that bin, we can trace back to the original RT and m/z data frame to find the peaks, which are inside that interval. With the value of m/z, we can find the important peptides.

4 Results and Discussion

In this section, we present the experimental results obtained from applying the PassiveAggressiveClassifier and GradientBoostingClassifier model to the dataset, providing an analysis of its performance and comparing it to other baseline classifiers.

4.1 Model evaluation

From Figure 2, we can observe the remarkable performance of both the GradientBoostingClassifier and the PassiveAggressiveClassifier model in discriminating between the control class and others. The mean AUC value of the GradientBoostingClassifier model on the test set, measuring an impressive 0.98, signifies its superior ability to accurately classify instances compared to the PassiveAggressiveClassifier model, which achieved a mean AUC value of 0.90. This substantial difference in AUC values clearly demonstrates the superior discriminative power of the GradientBoostingClassifier model in capturing the distinguishing features that separate the control class from the others.

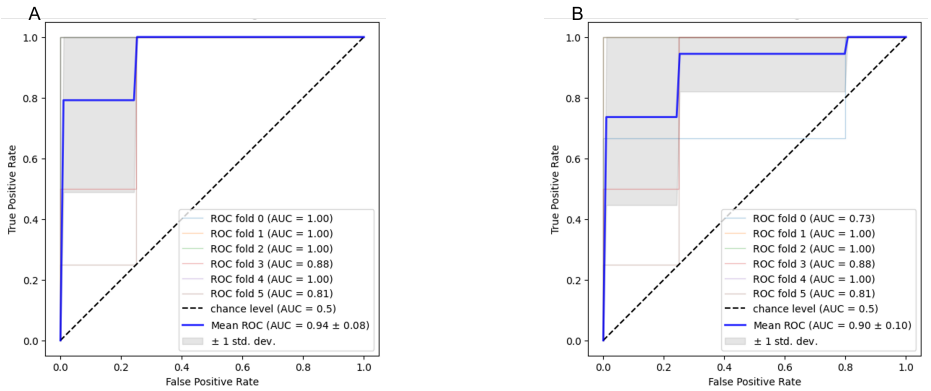

Figure 2 ROC curves for each model: Figure 2A is ROC curve for GradientBoostingClassifier model and Figure 2B is a ROC curve for PassiveAggressiveClassifier model. The blue line in each ROC indicates the mean AUC.

Furthermore, it is noteworthy that ROC curves of both classifiers lie comfortably within the confidence interval, providing a strong indication of the reliability and robustness of the results. This implies that the observed performance of the classifiers is statistically significant and not merely a result of chance or random variation. Therefore, we can have confidence in the reported AUC values and the corresponding discrimination capabilities of the models.

In addition, the results from Table 1 provide clear evidence of the superior performance of the GradientBoostingClassifier model compared to the PassiveAggressiveClassifier model. The GradientBoostingClassifier model achieved an accuracy of 0.96, which is significantly higher than the accuracy of 0.85 obtained by the PassiveAggressiveClassifier model. This substantial difference in accuracy underscores the GradientBoostingClassifier model's superior ability to classify instances correctly. Besides that, the sensitivity of the GradientBoostingClassifier model with

Table 1 Performance Metrics Comparison

Classifier	Accuracy	Sensitivity	Specificity	F1-score
GradientBoostingClassifier	0.96	0.96	0.95	0.96
PassiveAggressiveClassifier	0.85	0.92	0.78	0.94

0.96 also surpasses that of the PassiveAggressiveClassifier model, which achieved a sensitivity of 0.92. The higher sensitivity of the GradientBoostingClassifier model indicates its proficiency in correctly identifying a larger proportion of positive instances from the control class. The specificity value, which indicates how accurately identifying true negatives and minimizing the occurrence of false positives, shows also significantly better performance of the GradientBoostingClassifier model in comparison with the PassiveAggressiveClassifier model. Additionally, when considering the F1-score, a metric that balances precision and recall, the GradientBoostingClassifier model achieves a higher score of 0.96 compared to the F1-score of 0.94 obtained by the PassiveAggressiveClassifier model. This reinforces the overall superiority of the GradientBoostingClassifier model in terms of capturing relevant instances and achieving a balance between precision and recall.

4.2 Feature importance

Order	GradientBoostingClassifier	PassiveAggressiveClassifier
1	mz_668_rt_822	mz_774_rt_551
2	mz_721_rt_280	mz_880_rt_1093
3	mz_668_rt_1636	mz_721_rt_551
4	mz_827_rt_2449	mz_1198_rt_1636
5	mz_615_rt_2720	mz_1251_rt_1364
6	mz_615_rt_1907	mz_668_rt_551
7	mz_668_rt_1364	mz_880_rt_1636
8	mz_721_rt_822	mz_880_rt_1364

Table 2 The 8 most important features with m/z and RT values identified by GradientBoostingClassifier and PassiveAggressiveClassifier model which correspond to a peptide. For example, mz_668_rt_822 means a peptide having 668 m/z and 822 RT values.

Table 2 shows the 8 most important features with m/z and RT values which correspond to a peptide that is listed in the identification file listed in datasets description[5]. For example, mz_668_rt_822 which is identified as important by GradientBoostingClassifier could represent the nearest peptide with a similar m/z value, namely "K.DFPIANGERQSPVDIDTK.A" as listed in the documentation which was spiked by CA. Other examples from the feature importance of PassiveAggressiveClassifier are mz_1198_rt_1636 and mz_1251_rt_1364 which could represent the peptide "K.DEDTQAMPFR.V" and "R.ADHPFLFCIK.H" which were spiked by OVA, respectively. Moreover, further verification of these peptides may offer important insights into proteins and the development of diagnostic methods may be aided by further analysis.

5 Contributions

1 Pham Gia Cuong: Overall 50% report and did feature extraction, and feature importance.

2 Nepal Aakash: Overall 50% report and did data augmentation, binning, and ML models, the idea of naming the features.

References

[1] Hannes L. Röst et al. "pyOpenMS: A Python-based interface to the OpenMS mass-spectrometry algorithm library". In: *Proteomics* 14.1 (2014), pp. 74–77. DOI: 10.1002/pmic.201300246.

[2] Hans J.C.T. Wessels et al. "A comprehensive full factorial LC-MS/MS proteomics benchmark data set". In: *Proteomics* 12.14 (2012), pp. 2276–2281. DOI: 10.1002/pmic.201100284.

[3] Sebastian Rauschert et al. "Machine learning and clinical epigenetics: a review of challenges for diagnosis and classification". In: *Clinical Epigenetics* 12 (1 2020), p. 51. DOI: 10.1186/s13148-020-00842-4. URL: https://doi.org/10.1186/s13148-020-00842-4.

[4] E. coli Dataset. *E. coli Dataset README*. https://cac.science.ru.nl/research/data/ecoli/README.pdf. Accessed: July 7, 2023. 2012.

[5] E. coli Dataset. *E. coli Dataset*. https://cac.science.ru.nl/research/data/ecoli/. Accessed: July 7, 2023. 2012.