CafsPy: A Covering Array Feature Selection Python Library

Abstract

CafsPy is a feature selection Python library grounded on Covering Array (CA) theory. It enables the discovery of minimal yet highly informative subsets of features or wavelengths, optimizing classification accuracy in high-dimensional datasets. CafsPy is particularly valuable in domains like chemometrics, where hundreds or thousands of spectral bands require reduction before classification.

> *"CafsPy was developed as a Python-based feature selection module that generates multiple high-accuracy feature subsets. It provides practitioners and researchers with the ability to select subsets that maintain an adequate F1-score."*
>
> (Romo et al., 2025, p. 3)

Overview

CafsPy is a Python-based library tailored for feature and waveband selection through the application of **Covering Arrays (CA)**. It specifically addresses the challenge of high-dimensional data by modeling interactions among features using a structured combinatorial approach. The library implements two main algorithms—**Covering Array Feature Selection (CAFS)** and **Iterative Covering Array Feature Selection (ICAFS)**—both of which are grounded in the concept of *t*-way interaction testing commonly used in software engineering.

Covering Arrays are represented as **CA(N, k, t, v)**, where:

- **N** denotes the number of test cases (rows),
- **k** is the number of parameters (features),
- **t** represents the interaction strength,
- **v** is the size of the alphabet (typically binary for CafsPy).

Each row in the CA corresponds to a feature subset which is then evaluated using a classification model. The **CAFS algorithm** utilizes a fixed CA to iteratively reduce the feature set, while **ICAFS** regenerates a new CA at each iteration based on the updated feature pool using the **IPOG algorithm**, thus preserving interaction coverage.

The ICAFS implementation is highly configurable: users can select the classifier (e.g., kNN, SVM, MLP), define the interaction strength *t*, set the number of iterations *T*, and apply seed-based shuffling for randomized yet reproducible subset generation. Unlike traditional feature selection techniques, CafsPy emphasizes discovering **subsets that maintain or improve classification performance** (measured via F1-score), not merely identifying the most individually relevant features.

Algorithms Implemented

The library includes two core algorithms:

1. CAFS (Covering Array Feature Selection): Uses a static binary covering array to generate multiple feature subsets and evaluates classification performance on each of them. It is limited to the size of the initial covering array (1400 features at most). Besides, it assumes fixed feature dependencies and uses a truncated array as features to be reduced.
2. ICAFS (Iterative CAFS): It is an improvement over CAFS, given that it generates a new covering array at each iteration using the IPOG algorithm. The advantage is that it maintains t-way interactions dynamically as features are reduced, improving adaptability. Besides, it works with any sklearn classifier, like kNN, SVM, MLP, OPF, etc, providing classifier independence.

How It Works

Example usage in Python:

```
from cafspy import ICAFS, CAFS
from sklearn.neighbors import KNeighborsClassifier

# Data preparation
X = (data - data.min()) / (data.max() - data.min())
y = labels

# Set the classifier and hyperparameters
clf = KNeighborsClassifier(n_neighbors=3)

# Run ICAFS
scores_list, feature_list = ICAFS(X, y, t=2, T=10, lr=clf, print_logs=True)
```

Applications

CafsPy has demonstrated strong performance in chemometrics, hyperspectral imaging, and general classification tasks. For example, in the classification of **Amazonian cacao-clone nibs**, CAFS was able to reduce 1401 NIR wavelengths to only five, achieving an F1-score of

98.89%, outperforming other techniques such as Eigenvector Centrality Feature Selection (ECFS) and Multi-Cluster Feature Selection (MCFS) (Castro et al., 2022).

Feature Selection Taxonomy

Feature selection (FS) is a vital pre-processing step in machine learning and data analysis, aiming to reduce data dimensionality while preserving or enhancing model accuracy and interpretability. Traditional FS methods fall into **three broad categories**:

1. **Filter Methods.** Evaluate the importance of features based on statistical measures (e.g., correlation, mutual information) independent of any learning algorithm.
2. **Wrapper Methods.** Utilize the performance of a specific learning model to evaluate different feature subsets, often involving search strategies like forward selection, backward elimination, or metaheuristics.
3. **Embedded Methods.** Perform feature selection as part of the model training process.

CafsPy is a **wrapper-based method** under the **subset selection** category, specifically using **combinatorial search strategies (covering arrays)** to explore the feature space. Each subset is evaluated using a machine learning model with cross-validation, and the best-performing subset is selected. This makes CafsPy a **model-specific subset selector**, as defined in more recent FS taxonomies such as those proposed by **Li et al. (2017)**.

Unlike **metaheuristic wrappers** (e.g., Genetic Algorithms, Binary Bat Algorithm), which may lack reproducibility or theoretical coverage guarantees, CafsPy provides **systematic and theoretically grounded subset generation** through CAs. Moreover, it fills a gap in the Python ecosystem, which largely lacks tools based on CA-driven feature selection despite their success in software testing and system design.

Installation & Requirements

Install via pip:

pip install cafspy

References

Castro, W., De-la-Torre, M., Avila-George, H., Torres-Jimenez, J., Guivin, A., & Acevedo-Juárez, B. (2022). Amazonian cacao-clone nibs discrimination using NIR spectroscopy coupled to naïve Bayes classifier and a new waveband selection approach. Spectrochimica Acta Part A: Molecular and Biomolecular Spectroscopy, 270, 120815. https://doi.org/10.1016/j.saa.2021.120815

Castro, W., Seminario, R., Nauray, W., Acevedo-Juárez, B., De-la-Torre, M., & Avila-George, H. (2025). Multispectral drone imagery dataset for plus and non-plus Neltuma pallida trees in northern Peru. Data in Brief, 60, 111645. https://doi.org/10.1016/j.dib.2025.111645

Li, J., Cheng, K., Wang, S., Morstatter, F., Trevino, R. P., Tang, J., & Liu, H. (2017). Feature selection: A data perspective. *ACM Computing Surveys (CSUR)*, *50*(6), 1–45. https://doi.org/10.1145/3136625

Romo Macías, S., Avila-George, H., Torres-Jimenez, J., Castro, W., & De-la-Torre, M. (2025). CafsPy: A Covering Array Feature Selection Python Library. In Proceedings of the International Conference on Software Process Improvement (under review).