Faculty of Mathematics and Information Science

WARSAW UNIVERSITY OF TECHNOLOGY

Deliverable 1

Adapting the gips library for classification problem utilizing discriminant analysis – gipsDA

Norbert Frydrysiak, Antoni Kingston

version 1.0

22.10.2025

Table of Contents

1 Abstract

Classification methods such as Linear and Quadratic Discriminant Analysis (LDA and QDA) often face challenges in high-dimensional, low-sample-size settings, primarily due to the difficulty of accurately estimating the covariance matrix. This project proposes the development of gipsDA, a novel R package designed to address this limitation. The core methodology involves adapting classical discriminant analysis by incorporating permutation symmetry constraints on the covariance matrix. By leveraging the functionalities of the gips R package to identify these underlying symmetries, the number of parameters to be estimated is significantly reduced.

The primary objective is to create new, more robust classifiers that exhibit superior performance and are less prone to overfitting in data-scarce environments. The final deliverable will be a well-documented and rigorously tested R package with a user-friendly interface, providing a valuable new tool for researchers and practitioners. Ultimately, gipsDA aims to offer a specialized solution for "small data" challenges, making previously intractable classification problems more solvable.

1.1 History of changes

Date	Author	Description	Version
19.10.2025	Norbert Frydrysiak	First version without Project Schedule Section	0.6
20.10.2025	Antoni Kingston	Project Schedule and Minor Changes	0.8
22.10.2025	Norbert Frydrysiak, Antoni Kingston	Enhancing overall sentence style and Gantt chart modification	1.0

2 Vocabulary

LDA (Linear Discriminant Analysis) - A classic statistical method in machine learning used for classification and dimensionality reduction. As a classifier, it works by modeling the probability distribution of the data for each class. It assumes that distributions of all classes share the same covariance matrix. Based on this assumption, LDA finds a hyperplane constituting decision boundary that separates the classes, thus its name. It's efficiency is particularly high when categories are in fact linearly discriminable.

QDA (Quadratic Discriminant Analysis) - A close relative of LDA, also a probabilistic classification method. Has less strict assumptions than LDA i.e. distributions of classes do not share the same covariance matrix. This results in a quadratic decision boundary (e.g., a curve like a parabola or an ellipse), which can capture more complex relationships between classes. The increased flexibility comes at a cost - the method requires estimating more parameters, meaning it generally needs more training data to perform well.

gips - An R package, which name stands for *Gaussian model Invariant by Permutation Symmetry*. Its primary purpose is to analyze a dataset and discover hidden permutation symmetries within data. By identifying these it can reduce the number of degrees of freedom. This is of extreme value in high-dimensional scenarios, especially when the number of features is substantially larger than the number of samples, a situation that poses significant problems with estimation of covariance matrix. The package can be used for exploratory data analysis to find underlying data structures as well as for estimatory purposes.

gipsDA - The name of our proposed engineering project. It represents a novel library that combines the principles of gips with classical Discriminant Analysis (LDA and QDA). The core idea is to create new classification models that leverage the permutation symmetries found using the gips methodology. By enforcing these symmetry constraints on the covariance matrix, the models will require fewer parameters. This will make them theoretically more robust and accurate for classification problems where the amount of available training data is limited. Essentially, gipsDA aims to adapt LDA and QDA for better performance in high-dimensional environments.

3 Specification

3.1 Executive summary

This project, undertaken as an engineering thesis, focuses on the creation of gipsDA, a novel R library for machine learning classification. The core of this work is to develop new classification models by adapting classical LDA and QDA. The key innovation is the integration of permutation symmetry constraints on the covariance matrix, a concept borrowed from the gips R library. This modification is designed to significantly reduce the number of parameters that need to be estimated, which is theoretically expected to yield more robust and accurate classifiers, especially in scenarios where training data is scarce. The project's outcome will be a functional R library containing these new models, complete with a standard, user-friendly API featuring fit and predict methods, making it accessible for both academic research and practical application.

Business Goal:

The primary business goal of the gipsDA project is to unlock new data analysis capabilities in high-value domains where data acquisition is inherently difficult or expensive. By providing a tool that excels in low-data environments, gipsDA aims to reduce the time and cost associated with data collection in fields such as medical diagnostics, rare-event detection, and industrial quality control. The objective is to create a specialized, high-performance tool that gives organizations a competitive advantage by enabling them to build reliable predictive models from datasets that are too small for traditional machine learning techniques. The project's success will be measured by its ability to provide a demonstrable improvement in classification accuracy on small sample size datasets compared to standard LDA/QDA, thus offering a clear return on investment for its adoption.

System Vision:

The vision for gipsDA is to establish it as the go-to specialized library for classification on small sample size datasets within the R ecosystem. We envision gipsDA becoming a benchmark tool in academic research for discriminant analysis and a trusted component in the commercial data science toolkit for high-dimensional problems. In the future, the core concept of permutation-invariant covariance matrices will be extended beyond LDA and QDA to a broader suite of statistical models, creating a comprehensive framework for symmetry-constrained machine learning. The ultimate vision is for gipsDA to be recognized not just as a library, but as a pioneering approach that makes previously intractable classification problems solvable, thereby pushing the boundaries of what is possible with limited data.

3.2 Functional requirements

The functional requirements for the gipsDA library are defined by the interactions between the system and the primary actors - the Data Scientist and the Machine Learning Engineer. The Data Scientist is primarily focused on model experimentation, training, and analysis, while the Machine Learning Engineer is focused on integrating and deploying the trained models into larger applications.

Use cases

The following UML Use Case diagram illustrates the key functionalities of the gipsDA library and the actors who will interact with them.

Actor Descriptions:

Data Scientist - The primary user, responsible for exploring data, training classifiers to find predictive patterns, and evaluating their performance.

Machine Learning Engineer - Responsible for taking a validated, trained model and integrating it into a production system.

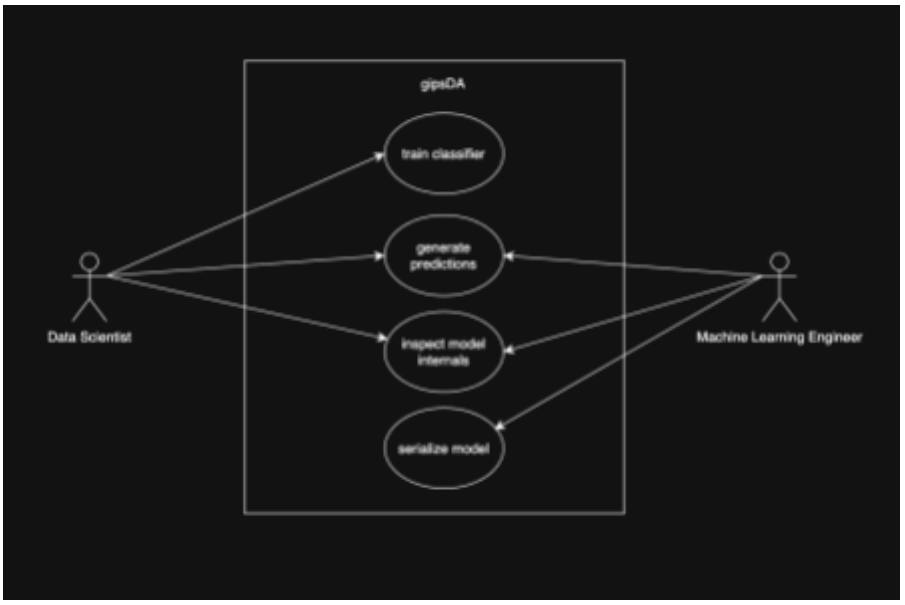

Figure 1 UML Use Case diagram

Table 1 Description of use cases for actors and system

Actor	Name	Description	System response
Data Scientist	Train Classifier	The user trains a gipsDA classification model by calling the fit() method with a feature matrix and a target vector.	The system validates the input data, runs the training algorithm to estimate all model parameters, and stores these parameters internally, making the model object "trained".
Data Scientist, Machine Learning Engineer	Generate Predictions	The user provides new, unseen data to a trained model by calling the predict() or predict_proba() methods.	The system applies the learned decision function and returns an array of predicted class labels or class probabilities.
	Inspect Model Internals	The user accesses the attributes of a trained model object to understand its learned characteristics.	The system returns the stored values of the requested parameters. If the model is not yet trained, it returns none.
Machine Learning Engineer	Serialize Model for Deployment	The user saves a trained model object to a json file and later loads it back into memory for use in an application.	The system writes the model's state to a json file. Upon loading, it reconstructs the trained model object in memory, ready for inference.

User Stories

1. As a Data Scientist, I want to train a classifier with a simple fit() method, so that I can rapidly prototype and test different models on my datasets.
2. As a Data Scientist, I want to access the learned parameters: covariance matrices and classes' means after training, so that I can analyze the underlying structure and discovered symmetries in the data.
3. As a Data Scientist, I want to get class probability scores for my predictions, so that I can evaluate the model's confidence and set appropriate decision thresholds.
4. As a Machine Learning Engineer, I want to save a trained model to a single json file, so that I can easily deploy it as an asset in a production application, potentially in different programming environments.
5. As a Machine Learning Engineer, I want to load a saved model and use it for prediction without needing the original training data, so that I can build efficient and scalable inference services.
6. As a Data Scientist, I want the library to raise errors for invalid input data, so that I can quickly debug my data preparation pipeline.

3.3 Non-functional requirements

Table 1 List of non-functional requirements grouped into URPS categories

Requirements area	Requirement No.	Description
Usability	1	**API Consistency:** The library's interface must be consistent with common R/Python modeling packages, for instance predict() method for generating predictions.
	2	**Comprehensive Documentation:** The package must be fully documented using the standard R documentation system. It should include detailed help files for every exported function.
	3	**Informative Error Messaging:** When the user provides invalid input (e.g., incorrect data types, mismatched dimensions), the system must throw errors with clear, actionable messages that help the user diagnose the problem.
Reliability	4	**High Test Coverage:** The package must include a comprehensive test suite using the testthat framework. The tests must cover the core logic, edge cases, and user inputs to ensure mathematical correctness and robustness.
	5	**Reproducibility:** The training process can be made deterministic by setting a random seed. This guarantees that when a classifier is trained on the same data and with the same parameters, it will produce the exact same model.
	6	**Graceful Handling of Edge Cases:** The library must handle known edge cases in the input data gracefully. This includes datasets with only one class, features with zero variance, pairwise linear dependence columns, throwing clear errors instead of crashing or producing NA/NaN results without warning.

Performance	7	**Predictable Performance on Target Datasets:** For datasets within the library's target scope (e.g., up to 1,000 samples and 100 features), model training and prediction should complete within a reasonable timeframe on standard consumer hardware.
	8	**Efficient Memory Usage:** The library's algorithms should be implemented to minimize unnecessary data copying, a common performance bottleneck in R. It must not hold multiple redundant copies of the full dataset in memory.
	9	**Scalability Clarification:** The documentation must clearly state the practical limitations of the algorithms.
Supportability	10	**Code Quality and Readability:** The source code must adhere to a consistent and readable style. The code should be well-commented and logically structured to facilitate future maintenance and contributions.
	11	**Minimized Dependencies:** The package should depend on a minimal, well-defined set of stable packages available on CRAN. This simplifies installation and reduces the risk of dependency conflicts in users' environments.
	12	**Standardized Packaging:** The library must be structured as a standard R package, passing R CMD check without errors or warnings, or significant messages. The ultimate goal is for the package to be suitable for submission to CRAN, allowing users to install it with `install.packages("gipsDA")`.

4 Project schedule

We do not plan to follow a strict division of fields of responsibility — every task will be jointly managed by both writers.

Color	Norbert Frydrysiak	Antoni Kingston
	0.7	0.3
	0.3	0.7
	0.5	0.5

Color-coded tasks correspond to the Gantt chart below.

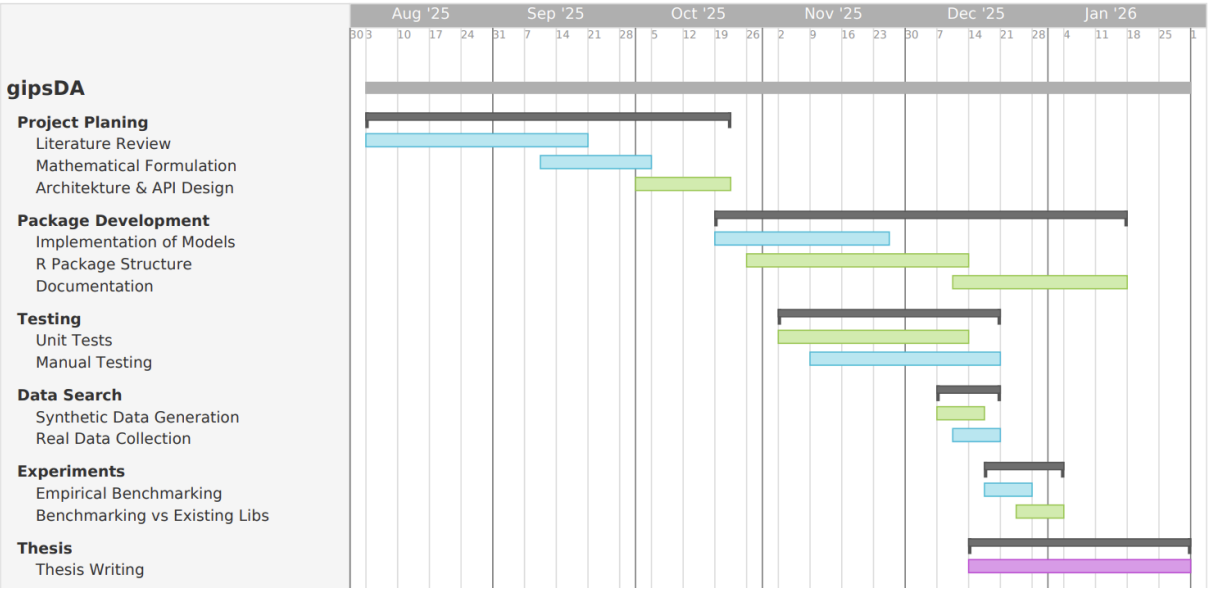

Figure 1 Project schedule, Gantt diagram.

5 Risk Analysis

SWOT	Threats	Opportunities
Internal	**Strengths**	**Weaknesses**
External	**Opportunities**	**Threats**

Strengths

- Innovative approach with the potential for high impact in niche area (high-dimensional dataset classification).
- Strong theoretical foundation based on the established gips library.

Opportunities

- High demand for machine learning models that perform well on small datasets, for instance in medicine or biology.
- Potential for publication in scientific journals and presentation at conferences.
- The library can be extended to other statistical machine learning models that rely on a covariance matrix (e.g., Gaussian Mixture Models).

Weaknesses

- The entire project is dependent on the gips library. Any bugs, changes in gips will directly impact gipsDA.
- The assumption of permutation symmetry in the data is a strong one and may not hold for many real-world problems. This makes the library a highly specialized tool rather than a general-purpose one.

Threads

- Analysts and scientists tend to stick with proven, familiar tools. Convincing them to try a new, specialized library instead of well-known packages might be a significant challenge.
- The field of machine learning is evolving rapidly. A new, more universal technique for high-dimensional datasets might emerge that proves more effective and overshadows gipsDA.
- Modern, heavily regularized models like XGBoost, CatBoost, or even glmnet can sometimes perform surprisingly well on high-dimensional datasets, potentially achieving comparable results without the specific assumptions.

6 Bibliography

[1] Hastie, T., Tibshirani, R., & Friedman, J. (2009). *The Elements of Statistical Learning: Data Mining, Inference, and Prediction* (2nd ed.). Springer. https://doi.org/10.1007/978-0-387-84858-7

[2] Chojecki, A., Morgen, P., & Kołodziejek, B. (2025). Learning Permutation Symmetry of a Gaussian Vector with gips in R. Journal of Statistical Software, 112(7), 1–38. https://doi.org/10.18637/jss.v112.i07