Automated Algorithm Selection

Sami Hatoum

Abstract

Algorithm selection is a meta-algorithmic strategy that aims to identify the most suitable algorithm from a portfolio for a given problem instance. This report evaluates the effectiveness of various models—linear regression, logistic regression, and neural networks—in predicting the best algorithm using instance-specific features. Through empirical comparison, performance metrics such as average predicted cost and SBS-VBS gap are analysed before and after scaling and hyperparameter tuning.

1 Introduction

Automated algorithm selection (AS) aims to optimise performance by selecting the most appropriate algorithm from a predefined portfolio based on instance-level features. Given a cost matrix and a set of instance features, the goal is to predict the algorithm that minimises cost per instance. This process is critical in domains such as SAT solving, combinatorial optimisation, and machine learning, where no single algorithm performs best across all problem instances.

2 Datasets

Two datasets were used:

- **Performance Data**: A cost matrix $M \in \mathbb{R}^{n \times m}$, where $M[i][j]$ is the cost of using algorithm j on instance i.
- **Instance Features**: A matrix $V \in \mathbb{R}^{n \times d}$, where each row describes the features of one instance.

3 Baseline Evaluation: SBS and VBS

3.1 Definitions

Single Best Solver (SBS):

$$\text{SBS} = \min_j \left(\frac{1}{n} \sum_{i=1}^{n} M[i][j] \right)$$

Virtual Best Solver (VBS):

$$\text{VBS} = \frac{1}{n} \sum_{i=1}^{n} \min_j M[i][j]$$

3.2 Results

$$\begin{aligned} \text{SBS (Train)} &= 2308.18 \\ \text{VBS (Train)} &= 1434.86 \\ \text{SBS (Test)} &= 1248.93 \\ \text{VBS (Test)} &= 917.35 \end{aligned}$$

Random model performance:

$$\begin{aligned} \text{Avg Cost (Test)} &= 4863.26 \\ \text{SBS-VBS Gap} &= \frac{4863.26 - 917.35}{1248.93 - 917.35} = 11.90 \end{aligned}$$

4 Model Architectures

Four models were trained:

- **Logistic Regression (Classification)**
- **Linear Regression**
- **MLP Classifier (Neural Network Classification)**
- **MLP Regressor (Neural Network Regression)**

Classification models map instance features to an algorithm index. Regression models estimate cost directly, selecting the algorithm with lowest predicted cost.

4.1 Preprocessing: Scaling

Standardisation transforms each feature to have zero mean and unit variance. This ensures all features contribute equally and improves convergence.

5 Performance Results

5.1 Without Scaling

Model	Train Cost	Test Cost	SBS-VBS Train	SBS-VBS Test
Logistic Regression	3419.57	2735.83	2.27	5.48
Linear Regression	1577.98	1560.45	0.16	1.94
MLP Classifier	4359.72	3707.47	3.35	8.41
MLP Regressor	2675.41	1792.67	1.42	2.64

Table 1: Average Cost and SBS-VBS Gap (Not Scaled)

5.2 With Scaling

Model	Train Cost	Test Cost	SBS-VBS Train	SBS-VBS Test
Logistic Regression	2078.73	2719.55	0.74	5.44
Linear Regression	1577.98	7269.41	0.16	19.16
MLP Classifier	2259.63	2540.52	0.94	4.90
MLP Regressor	2420.56	1571.11	1.13	1.97

Table 2: Average Cost and SBS-VBS Gap (Scaled)

6 Evaluation Metrics

6.1 Classification

Key metrics include accuracy, precision, recall, and F1 score. Accuracy was prioritised for evaluating classification models. MLPClassifier with scaling achieved 59.15% accuracy on the training set.

6.2 Regression

Regression models were evaluated using MAE, MSE, RMSE, and R^2. The lowest MAE was achieved by Linear Regression (scaled): 3149.91.

7 Hyperparameter Tuning

7.1 Logistic Regression

Best Parameters (Scaled): C=10.0, solver=lbfgs, multi_class=ovr

- Accuracy (Test): 36.67%
- Avg Cost (Test): 3919.69
- SBS-VBS Gap (Test): 9.05

7.2 MLP Models

MLPClassifier (Scaled): Hidden layer = (100,), activation = tanh, learning rate = constant (0.001)

- Accuracy (Train): 68.58%
- Accuracy (Test): 33.33%

MLPRegressor (Scaled): Hidden layer = (100,), activation = relu, learning rate = adaptive (0.001)

- MAE (Test): 4613.71
- R^2 (Test): -0.62

8 Conclusion

Linear regression consistently achieved the lowest prediction error and smallest SBS-VBS gap, indicating robust generalisation performance. Neural networks showed improvement with scaling and tuning but remained less consistent. The evaluation framework demonstrated how basic preprocessing and parameter tuning significantly impact AS model performance.