Understanding Multimodality in Feature Selection Landscapes

Xavier F. C. Sánchez Díaz Ole Jakob Mengshoel

Norwegian University of Science and Technology

NorwAI Norwegian Research Center for AI Innovation

NTNU

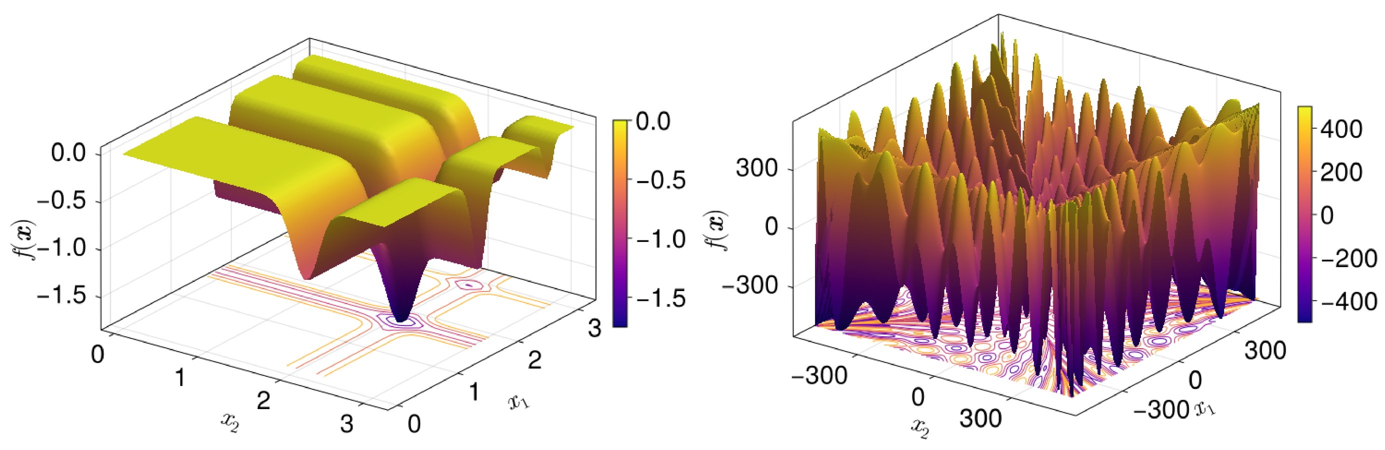
Abstract

We present different visualization techniques for **feature selection landscapes**, which generally are **multimodal**—they contain multiple solutions. We illustrate some of these visualization techniques and discuss how they help us gain insights about our feature selection problem. In this case study, each optimum is a subset of **features used to train** a Machine Learning (ML) model, represented as a bitstring. For example, the bitstring 0110 represents an ML model using the second and third features, and ignoring the other two.

What is a *Search Landscape?*

A *search landscape* is a visual representation of the solution space, as it is navigated by a search method or optimization algorithm. Any search landscape can be represented using a tuple $\mathfrak{L} = (\mathcal{X}, f, \mathcal{N})$, where $\mathcal{X}$ is the search space, f is a *fitness* (cost, error, or objective function), and $\mathcal{N}$ is any neighborhood or notion of accessibility from a state $\boldsymbol{b} \in \mathcal{X}$.

For example, a landscape could represent **the classification error** of an ML model over a specific dataset using a specific set of parameters. Studying the landscape can provide several insights: highlight feature importance, assess problem difficulty, or determine algorithm performance.



(a) Michalewicz function (b) Rana's function

Figure 1. The landscapes of some 2D test functions in the continuous domain [7].

Distance-Fitness Correlation and Number of Optima

How are optimal solutions distributed throughout the space?

A common approach to highlight multimodality in feature selection search spaces is to plot the correlation between the distance among optimal solutions and their classification error—either aggregated into bins or as a scatter plot.

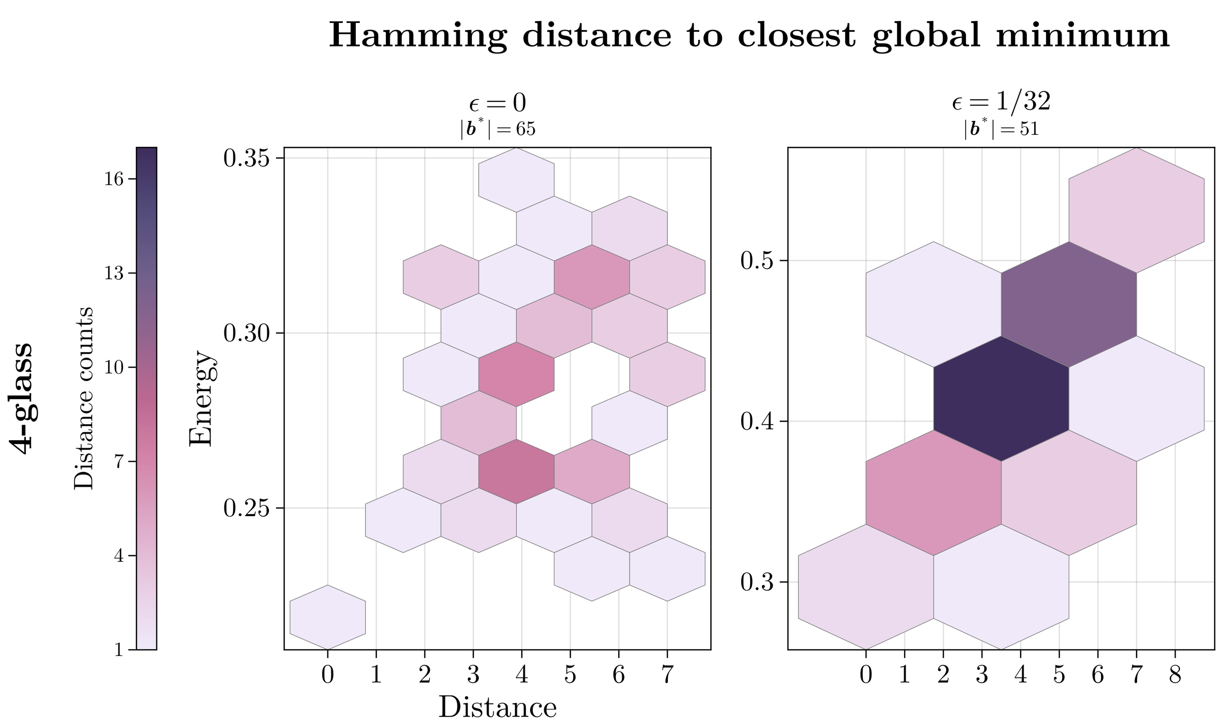
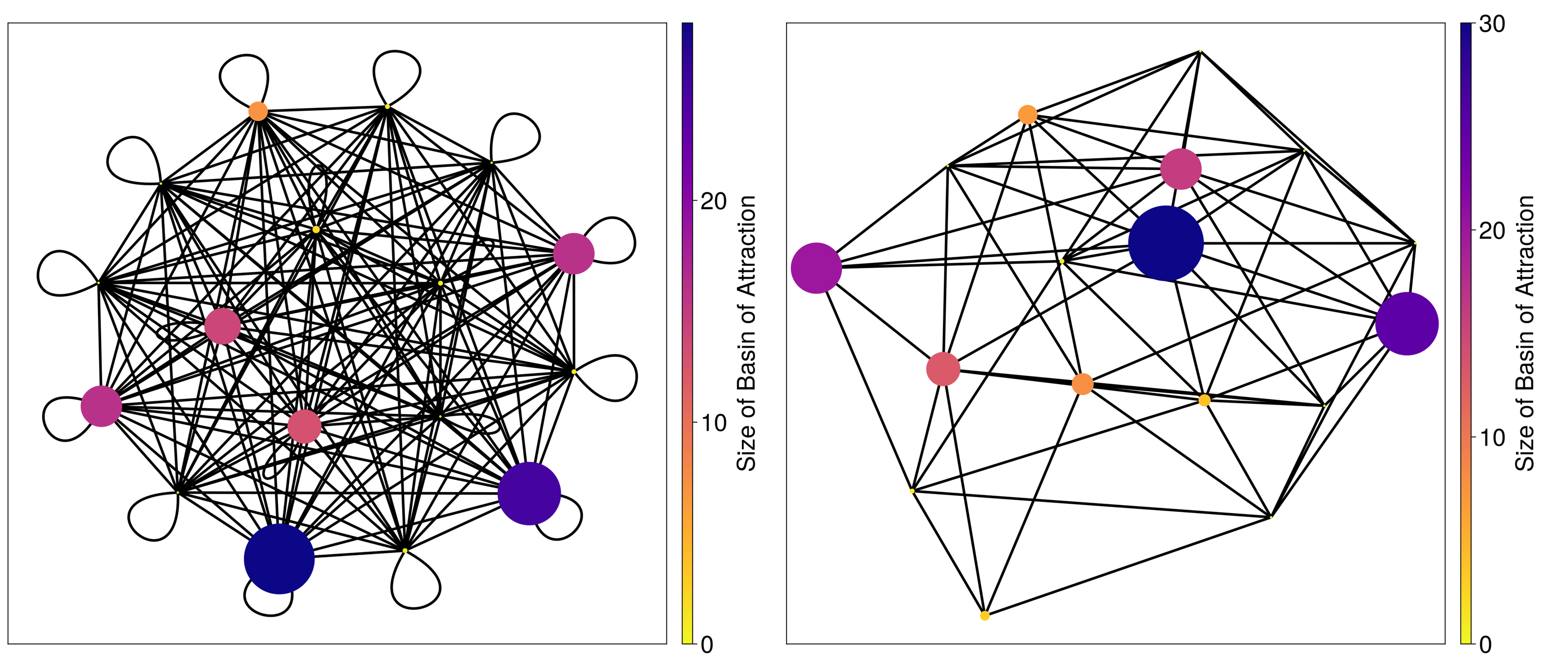


Figure 2. Juxtaposition of two Hex-bin plots of the distance-fitness correlation for the Glass Identification dataset [1], using a decision tree classifier under two levels of regularization (ϵ). Each bin aggregates different number of local optima, and a darker shade means a higher concentration of optima [5].

Connectedness and Local Optima Networks (LONs)

How difficult is it to go from one optimal solution to another?

LONs illustrate the connectivity of the landscape, highlighting *basins of attraction*—solutions that are found more often by a solver or classifier [3]. These visualizations help us estimate learning difficulty, which can in turn be useful for parameter tuning of similar problems.



(a) A LON with all basin transition edges (b) A LON with escape edges with $D = 3$

Figure 3. LON-LON: Juxtaposition of two LONs, representing the feature selection problem on the E-coli dataset [2], using a decision tree classifier. In 3b, the edges are only kept if the Hamming distance between local optima is less or equal than D, the amount of jumps needed to *escape*.

Hinged-Bitstring Maps (HBMs)

Are optimal solutions sparse? Are there more important features than others?

HBMs plot the entire search space by splitting its binary representation into two halves, each half associated with an axis—the first half uses the x-coordinate, and the second uses the y-coordinate. Solutions are plotted using their (x, y) coordinates and colored by their classification error. Optimal solutions are highlighted with a colored outline.

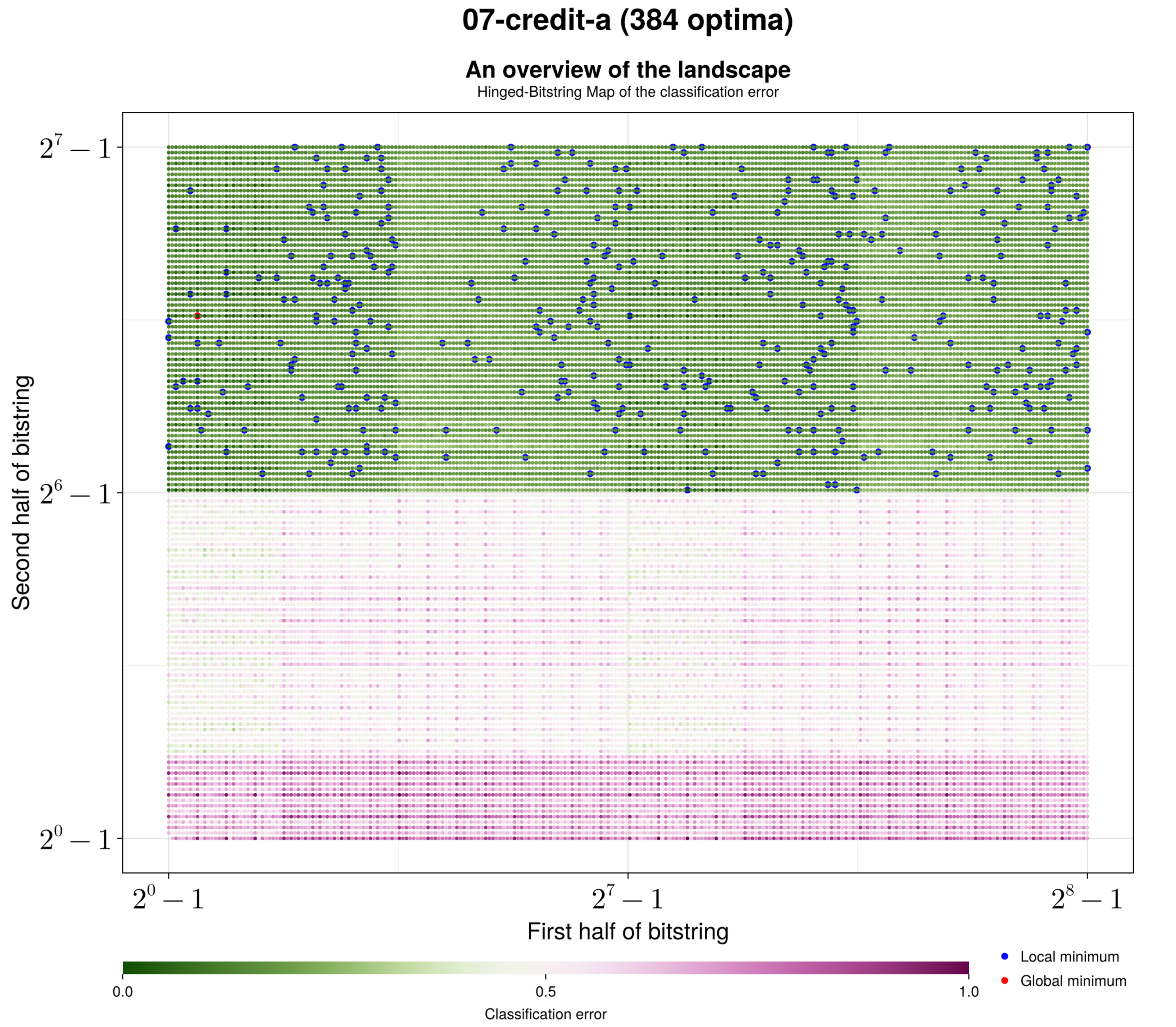


Figure 4. An HBM for the Credit Approval dataset [4], using the average of 32 individual decision trees. The greener a solution is, the better its classification accuracy. Locally optimal solutions are highlighted with a blue outline, and the global optimum with a red outline.

Feature Importance: Understanding an HBM

Since the position of a pixel in an HBM is relative to its binary representation, we can point to a specific feature being set by *folding* each axis in halves, fourths, eighths (or any $\log_2$ magnitude). For example, in Figure 4, the most significant bit (all the way to the left) of the second half of the bitstring is clearly impactful for classification error—all optimal solutions include the **ninth feature** (bold in Eq. 1).

Can you work out the position of the global optimum in Figure 4? Start on the leftmost part of each half of Eq. 1 and fold in two, recursively:

$$\boldsymbol{b}^* = [0, 0, 0, 0, 1, 0, 0, 0, \mathbf{1}, 1, 0, 0, 0, 0, 0] \quad (1)$$

- For the first half, a 0 means *left* while a 1 means *right* (x-axis)
- For the second half, a 0 means *bottom* while a 1 means *top* (y-axis)

Powered by High-Performance Computing

We can carry out these kind of experiments thanks to high-performance computing. Although analyzing the full search space is computationally demanding, similar analyses are also possible by sampling the space, offering practical insights about our feature selection problems.

norwegian open ai lab

We would like to thank the **Norwegian Open Artificial Intelligence Lab** for the promotion and hosting of our framework and tools in its GitHub repository, as well as **IDI** for access to **IDUN**; NTNU's High-Performance Computing Cluster [6].

Further Reading

More visualization methods, in-depth explanations and references can be consulted in our full report: `https://s.ntnu.no/visual-landscape`.