A global feature selection algorithm for the model selection step in the identification of cell signaling networks

Gustavo Estrela[1,2,3], Lulu Wu[1,2], Vincent Noël[1,3], Carlos Eduardo Ferreira[2], Hugo A. Armelin[1,3], Marco Dimas Gubitoso[2], Junior Barrera[1,2], and Marcelo S. Reis[1]

[1]Center of Toxins, Immune-response and Cell Signaling (CeTICS), Instituto Butantan, Brazil
[2]Instituto de Matemática e Estatística, Universidade de São Paulo, Brazil
[3]Laboratrio Especial de Ciclo Celular (LECC), Instituto Butantan, Brazil
[4]Instituto de Química, Universidade de São Paulo, Brazil

Motivation

In the context of Machine Learning, the feature selection problem consists in choosing a subset of features that best explains the classification with minimum redundancy. The space of solutions of this problem induces a Boolean lattice and the cost function commonly describes U-shaped curves on chains of this lattice, what is explained by the increase of estimation error as we include more features. Hence, we can approximate this problem to the U-Curve problem, which is a special case of the feature selection problem where every chain of the search space describes U-shaped curves. Some algorithms in the literature exploit this approximation; still, they show limitations regarding scalability, which might be a problem for the feature selection step in the identification of cell signaling networks. To tackle this issue, we developed the PUCS (Parallel U-Curve Search) algorithm (Fig. 1–2).

The algorithm

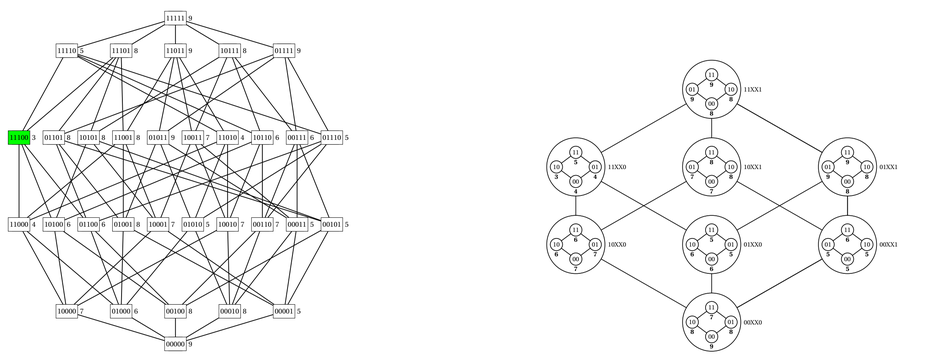


Fig. 1: An instance of the U-curve problem with $|S| = 5$ (left) and induced search space when the third and fourth elements are regarded as don't care (right).

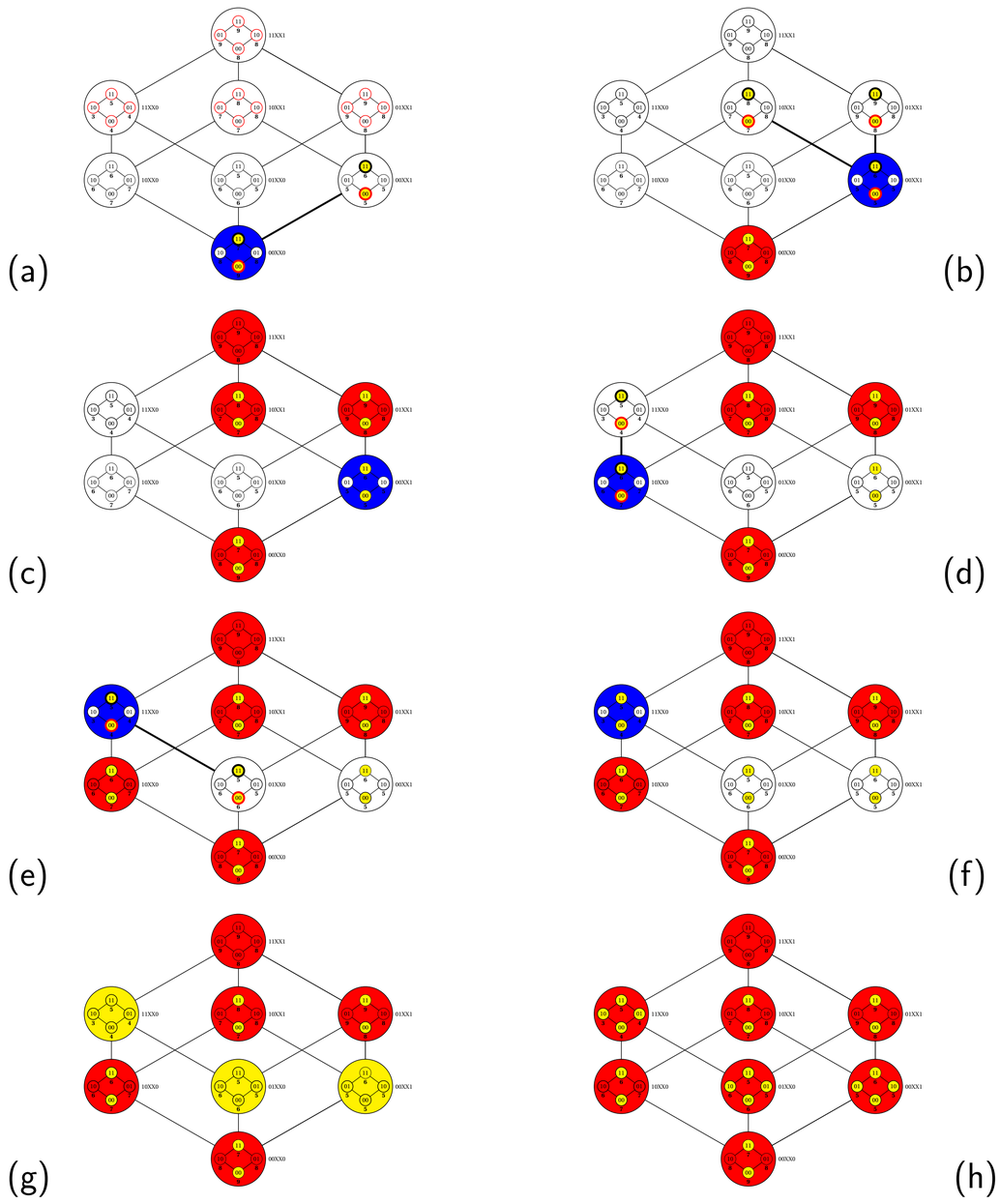
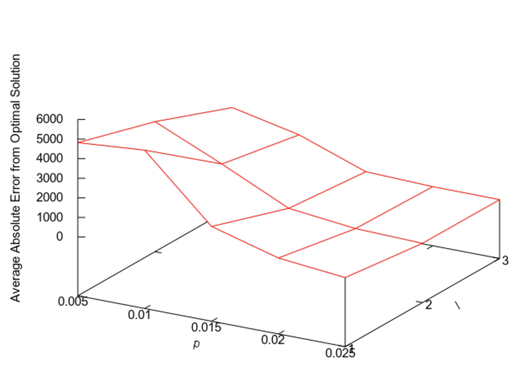


Fig. 2: Dynamics of the PUCS algorithm (Fig. 2(a)–(h)). In the outer lattice, blue, yellow and red colors mean a node being processed, solved or removed from the search space, respectively. In the inner lattice, yellow color on a given node means that the cost function was computed for its subset.

Results

PUCS has two parameters: p and l determine the size of the outer lattice and the number of recursive calls before calling the base case solver, respectively. In our implementation, we were able to confirm our expectations that the algorithm can find solutions as good as it possible (be optimal) as long as we increase the parameters p and l.



We used the *featsel* (github.com/msreis/featsel), a C++ framework to implement and benchmark PUCS with other algorithms, such as Exhaustive Search (ES), Sequential Forward Selection (SFS) and Backward Feature Selection (BFS). These experiments were carried out in a 64-core, 256 GB RAM server (Fig. 3–4).

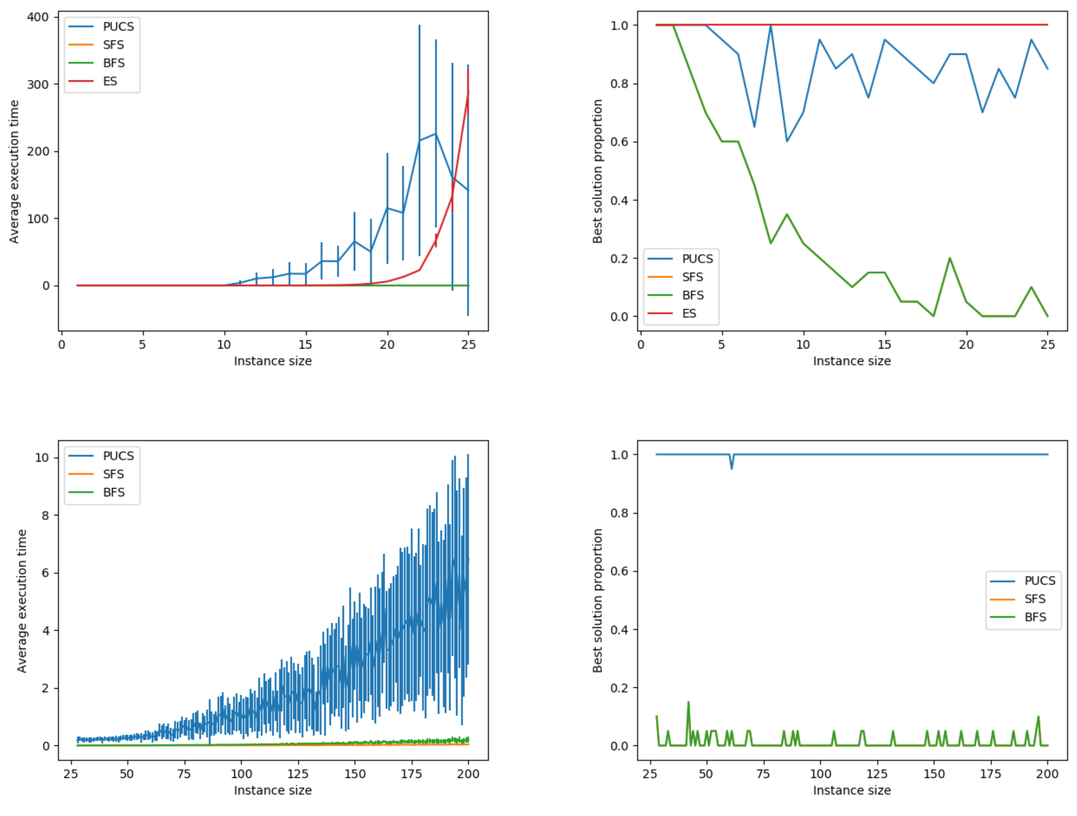


Fig. 3: Average execution time and average number of times each algorithm found the best solution. For these instances, we used SFS to solve recursion base cases of the PUCS algorithm.

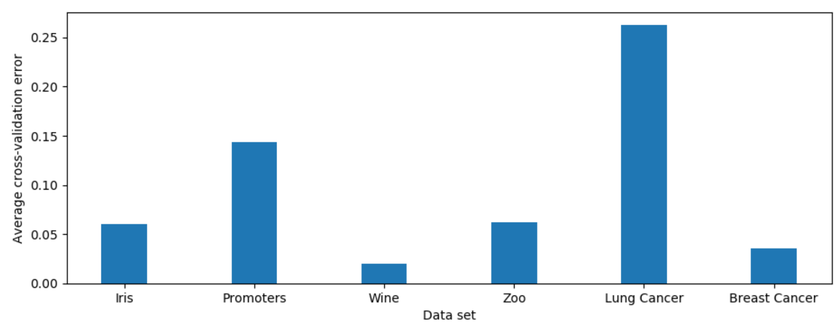


Fig. 4: Average cross-validation error when using the features selected by PUCS to design SVM classifiers for some biological databases available at the UCI Machine Learning Repository.

Conclusion

Results showed that PUCS has a better time performance than optimal solvers such as ES, and also can find better solutions than heuristics such as SFS and BFS. Future and ongoing work on this research line includes:

- Design of new algorithms, especially ones that generalize the search space for poset forests;
- Application of PUCS on the feature selection step in identification of cell signaling pathways.

{marcelo.reis, gustavo.matos }@butantan.gov.br

Acknowledgements

FAPESP CNPq CAPES