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 growth of estimation errors as we add more features.

The U shaped curves justifies the reduction to the U-Curve problem: a special case of the feature selection problem where every chain of the search space describe U shaped curves. Many algorithms in the literacy exploit this reduction and yet they show limitations regarding scalability, and that shows the need for new approaches on solving the U-Curve problem. To this end, we developed the Parallel U-Curve Search (PUCS).

Method

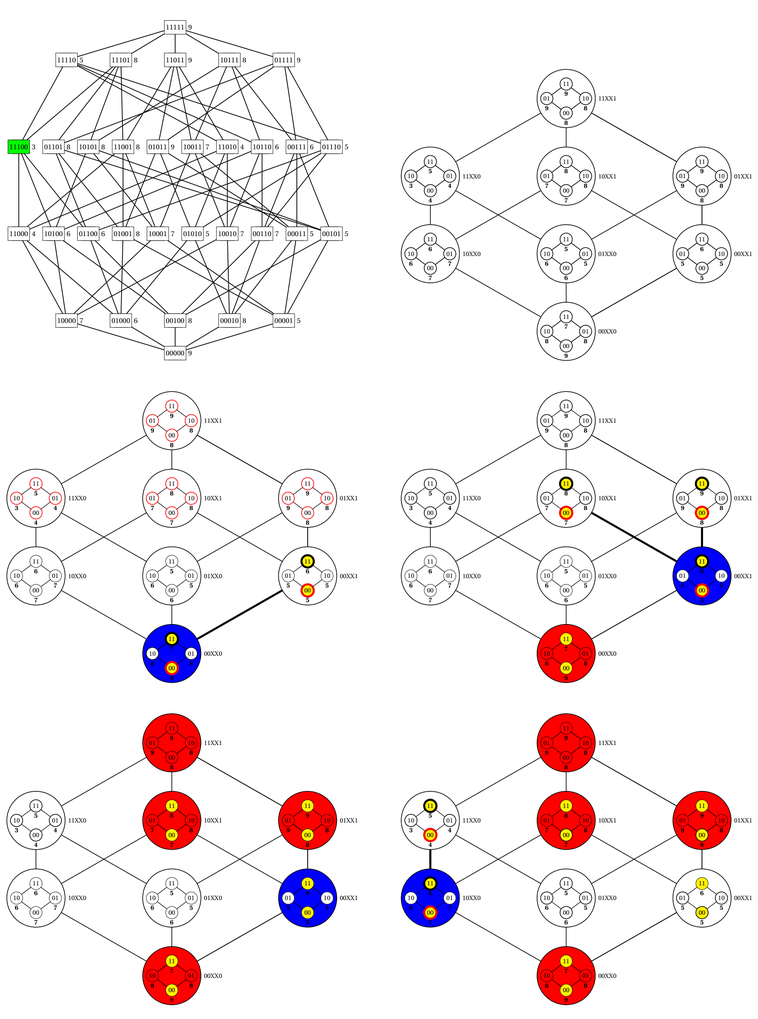
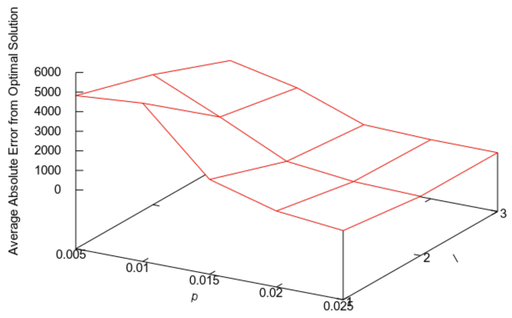


Figure: PUCS dynamics on an instance of the U-Curve problem.

Results

We implemented the PUCS algorithm on C++ language and used OpenMP to parallelize the code. Using a server with 64 cores and 256 gigabytes of memory 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* framework to benchmark PUCS with other algorithms, such as Exhaustive Search (ES), Sequential Forward Selection (SFS) and Backward Feature Selection (BFS).

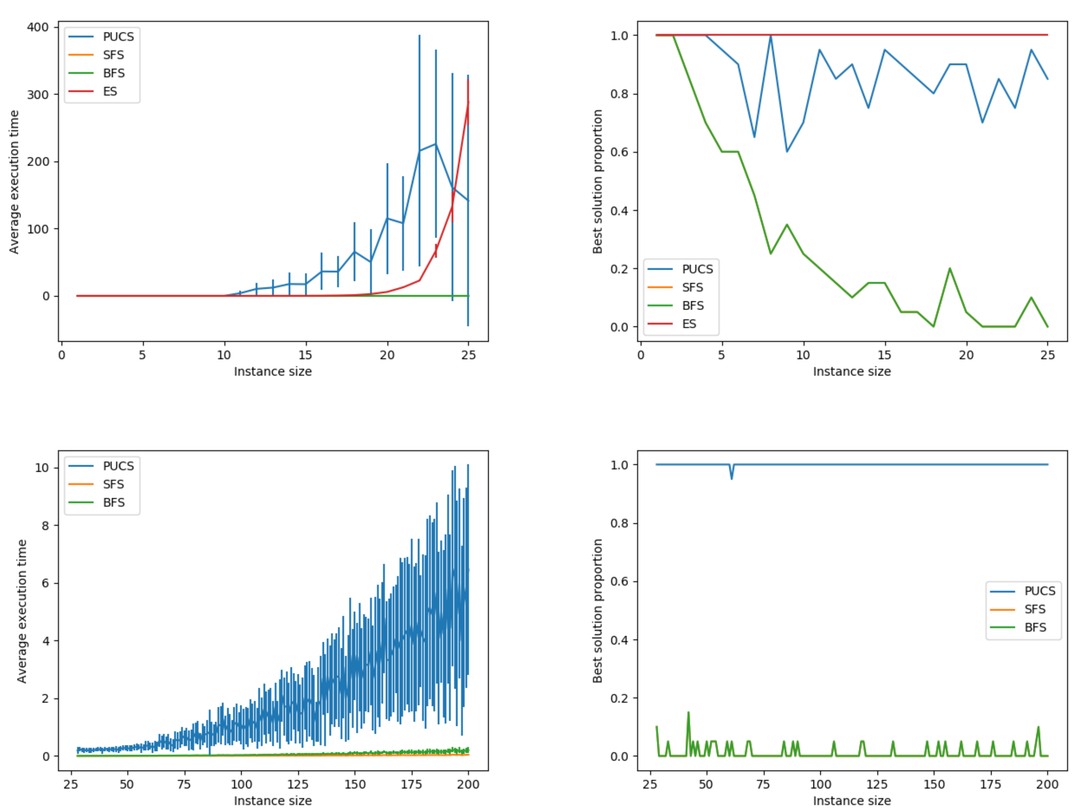


Figure: Average execution time and average number of times each algorithm found the best solution. For these instances we used an SFS as a base for the PUCS algorithm.

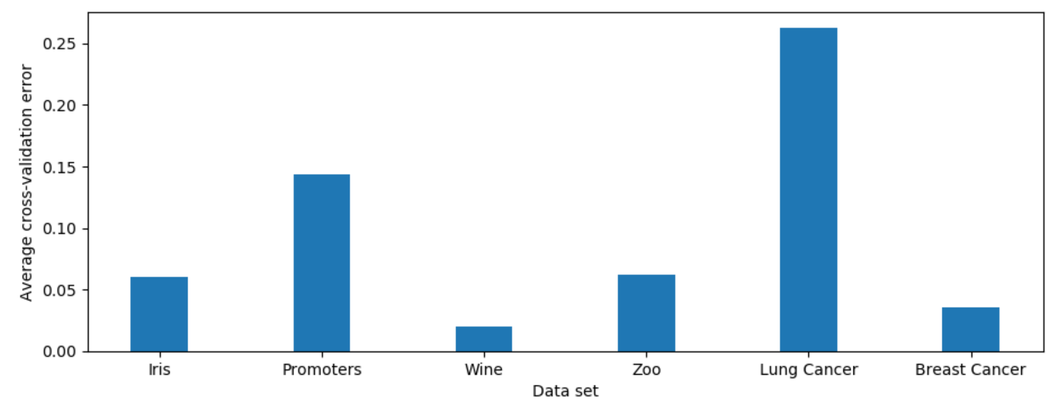


Figure: Average cross-validation error when using the features selected by PUCS to project a support vector machine learning model.

Conclusion

On our implementation of PUCS we could see that this algorithm has better time performance than optimal solvers (such as ES) and also finds better solutions than heuristics (such as SFS and BFS). Future and ongoing work on this project includes:

- design of poset forest-based algorithms;
- applications on cell signaling pathways.

Acknowledgements