Assignment 1

Intelligent Information Systems - Labs

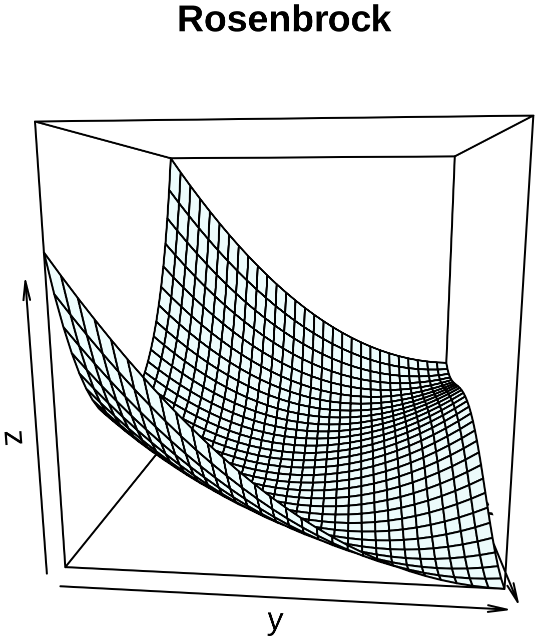
Description and overview

This is the first assignment you need to accomplish as part of the Intelligent Information Systems course. The material relevant for the following tasks is either provided as separate methods, or was considered during practical labs. Both tasks account together for 100% of the first 25% of the (practical labs) grade. Additional 5% can be obtained as a part of some tasks, yet maximum percentage (100%) cannot be exceeded.

The assignment must be submitted as a R Markdown file (file -> new file -> R Markdown), where you write both the code, as well as the theoretical answers (echo = T option must be enabled for each code block). Markdown files can easily be exported to PDF using ("Knit") button in R Studio; please submit also the PDF (along with the R markdown).

Task 1: Maximization of a non-convex function.

A variation of the Rosenbrock function we are interested in is defined as $f(x, y) = (1-x)^2 + e \cdot (y - x^2)^2$ (e is the base of the natural logarithm). It is non-convex. As a warm-up, what does it mean for a function to be non-convex?



Your task is to:

1. Code a method $f(x, y)$ that computes a value z, given an input tuple (x, y).
2. Code a method that visualizes the Rosenbrock function in 3D, where $x \in [-1, 1]$ and $y \in [-1, 1]$. (15%)
   - (HINT: there exists a *persp* function)
   - (HINT: Compute at least 400 values of z)
3. Code a genetic algorithm, that attempts to find the *global maximum* of this function. Plot the maximum value it finds in (2.) (15% + 5%)

- Try at least three different mutation and/or crossover settings.
- Extra 5%: Plot the trace of evolution (each evaluation) within 2.)

4. Discuss the results. (20%)
   - How does performance vary when you are increasing the number of iterations?
   - What about population size?
   - Explain the difference between local and global maxima.

Task 2: Genetic feature selection

High-dimensional data sets are quite common nowdays. The task of *feature selection* adresses the issue of identifying such features (= columns), that are the most relevant for a given e.g., classification problem. Individual solution, however you choose to represent it, must be evaluated as follows.

Evaluation of solutions

First, select your preferable classifier. The classifier shall be evaluated using three-fold cross validation. The output of this part is the average performance (justify the measure of your choice). Represent the final solution so that you take into account both the classifier's performance, as well as *the number of features.*

The dataset you are to use is *DLBCL.csv*, and represents a set of genes associated with B-cell Lymphoma. The target variable is called *class*. The hard constraint on the minimum number of features is 2. The hard constraint on the maximum number of features is 1000.

Hints:

- Separate the columns that are not the target from the target
- TrainControl from *caret* package offers simple evaluation procedures
- Train method from *caret* package offers simple model training
- Fitness must take into account both the number of features, as well as the classifier's performance

Grading for this task consists of:

- Solution encoding (5%)
- Fitness function specification (15%)
- Evolution of the optimal solution (15%)

Discussing the results (15%)

1. Was the feature selection successful? Report how many features were needed for the final performance. Plot the trace of evolution and comment on the change of fitness through generations.
2. Suggest how to improve the selection process.
3. The procedure might overfit the data set. How would you prevent it?
4. What are some of the key properties of the fitness function?
5. Compare the final result (set of features) with the same number of features, that correlate the most with the target variable. What do you observe?
   - HINT: There exists a *cor* function in R.