Evolutionary Feature Selection using Integer Encoding

Gregor Bankhamer, Tobias Buchner, Rene Maier, Christian Müller

Problem

- **Given**: Samples with n features and Knn classifier

- **Goal**: Find optimal feature subset w.r.t classification accuracy
- **Difficulty**: Huge search space of 2^n
- **Solution**:
    - Generate initial population of subsets
    - Compute fitness of each subset (classification (Knn) accuracy on training patterns)
    - Recombine best/good subsets to generate new subsets (+Mutation)

Solution Encoding (Revisited)

Solution == Subset

Assign to each feature an unique integer

I.e: Age = 1, Color=2, Length = 3

Possible solution:

- s =(3,2,2) … means classify only considering Length and Color as features

Old approach (previous group):

- Solution has to be a **valid permutation** of the features eg. $\{(x_1,x_2,x_3),(x_2,x_1,x_3),(x_3,x_2,x_1)...\}$

- **But**: order not relevant for euclidean distance

Distance between two patterns p1, p2 using solution (x_2,x_1,x_3):

$$(p1.x_2 - p2.x_2)^2 + (p1.x_1 - p2.x_1)^2 + (p1.x_3 - p2.x_3)^2$$

same classification, same fitness, no purpose for evolution

Distance between two patterns p1, p2 using solution (x_3,x_2,x_1):

$$(p1.x_3 - p2.x_3)^2 + (p1.x_2 - p2.x_2)^2 + (p1.x_1 - p2.x_1)^2$$

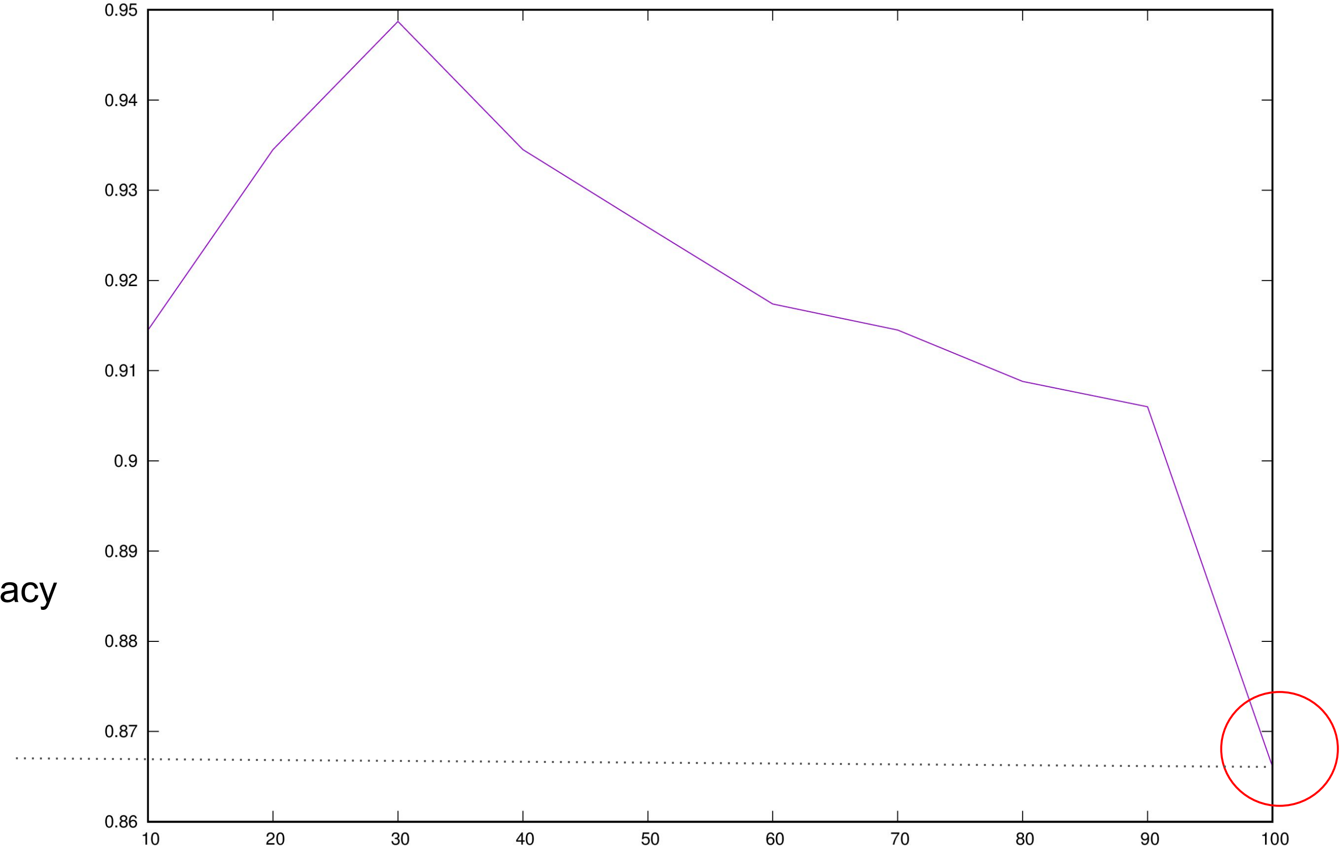
Fix

Solution has to be a **valid permutation** of the features eg.
$\{(x_1,x_2,x_3),(x_2,x_1,x_3),(x_3,x_2,x_1)...\}$

But: only part of evolved solution is used for classification:
$\{(x_1,x_2,x_3),(x_2,x_1,x_3),(x_3,x_2,x_1)...\}$ -> $\{(x_1,x_2),(x_2,x_1),(x_3,x_2)...\}$

Question: Where do we crop the solution?

Ionosphere

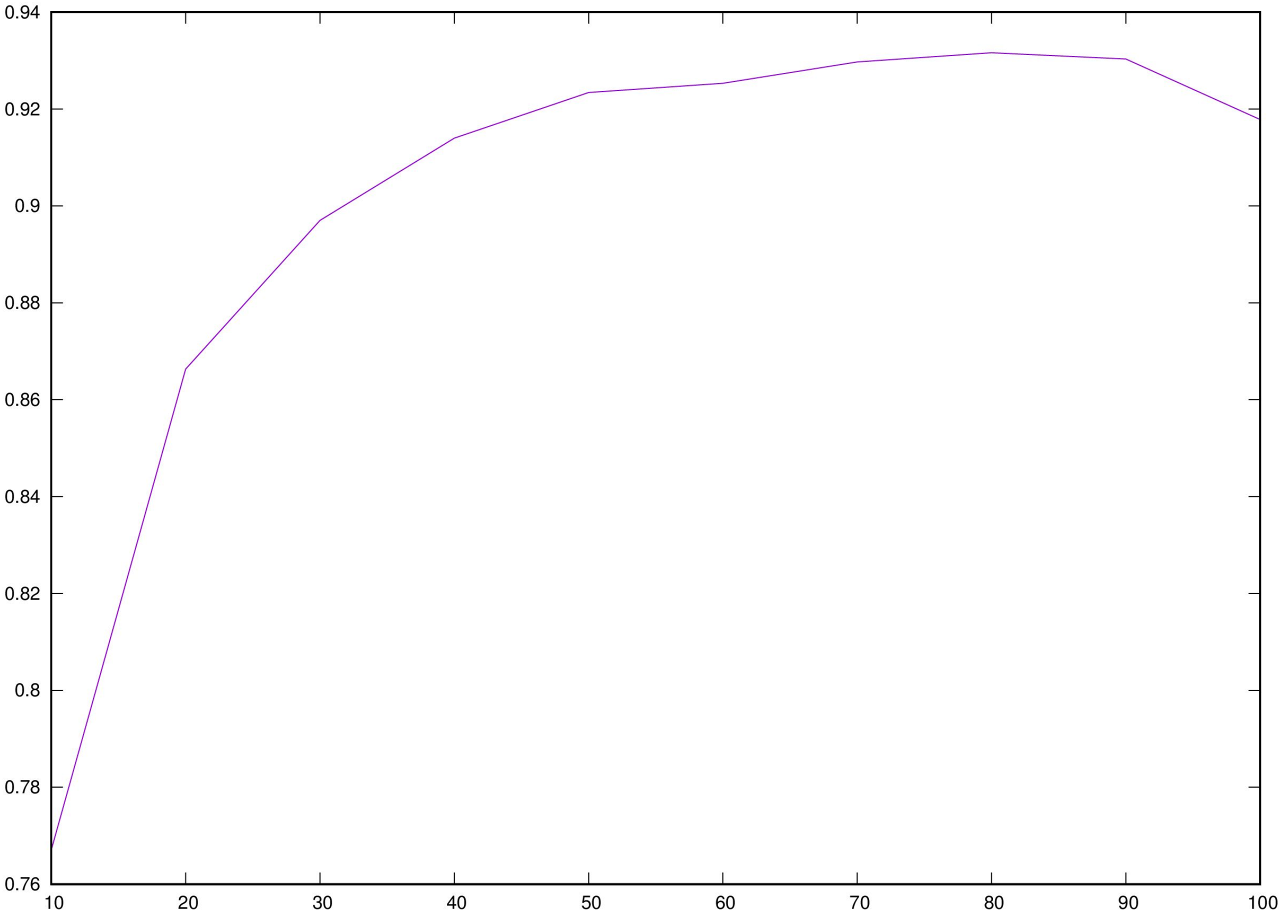


Classification accuracy

% features to use

=> crop solution at 30%
of features

Semeion



Classification accuracy

% features to use

New Solution

No real permutations eg. (3,2,2) is a valid solution
=> no cropping, number of used features is evolved

But: How should we calculate distance between patterns?

- Distance between two patterns p1, p2 using solution (x_3,x_2,x_2):

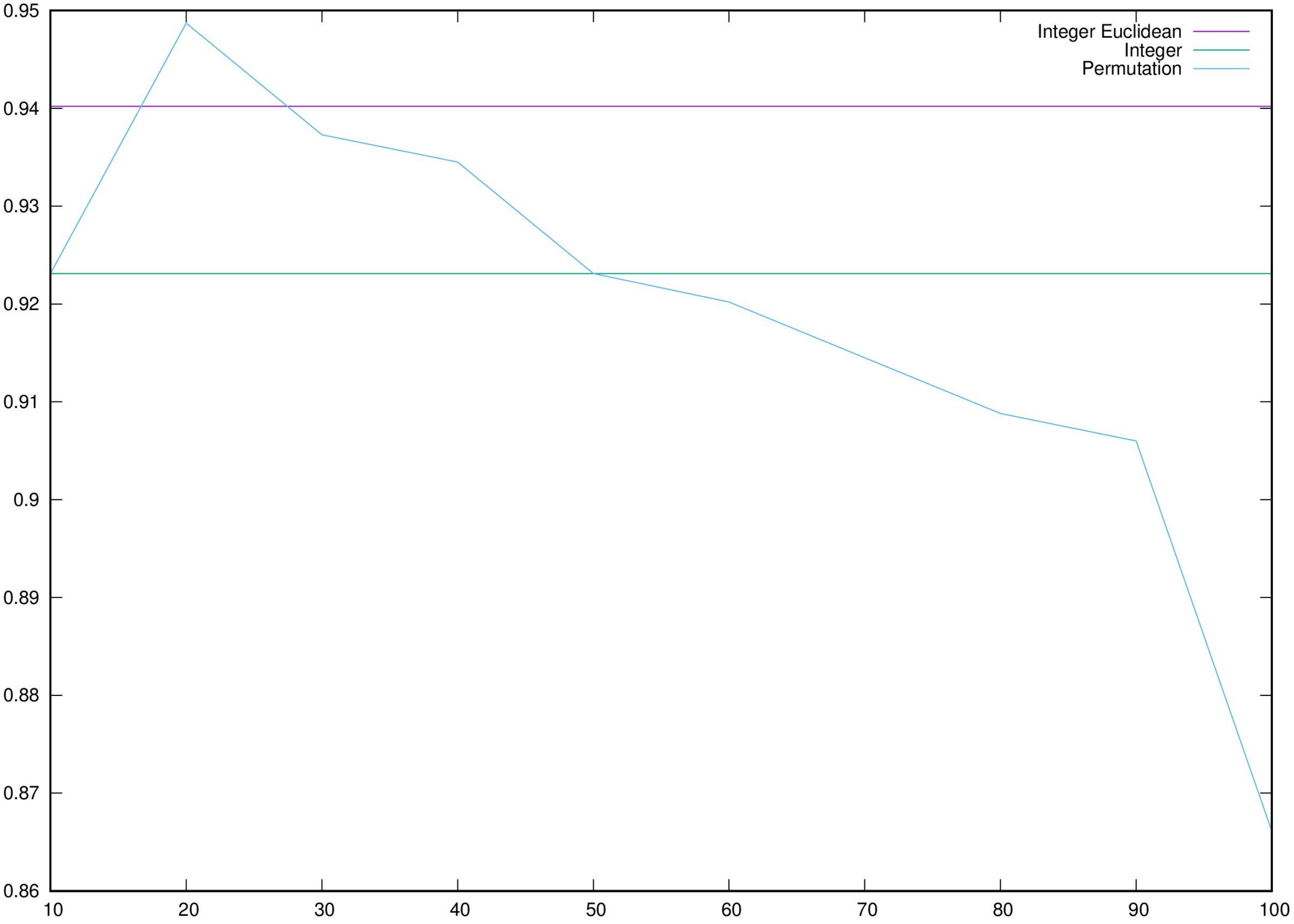
  $(p1.x_3 - p2.x_3)^2 + (p1.x_2 - p2.x_2)^2 + (p1.x_2 - p2.x_2)^2$, add “weight”

- Distance between two patterns p1, p2 using solution (x_3,x_2,x_2):

  $(p1.x_3 - p2.x_3)^2 + (p1.x_2 - p2.x_2)^2$, disallow duplicates

Ionosphere

100 generations



Outlook

- More generations with other datasets (not only ionosphere)
- How do evolved solutions look like? Analyse evolved individuals
- Summarize results and key findings