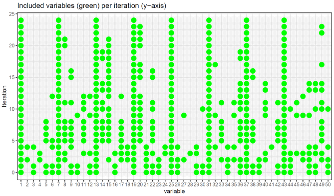
Optimization in Machine Learning

Evolutionary Algorithms - GA / Bit Strings



Learning goals

- Recombination
- Mutation
- Simple examples

BINARY ENCODING

- In theory: Each problem can be encoded binary
- In practice: Binary not always best representation (e.g., if values are numeric, trees or programs)

We typically encode problems with **binary decision variables** in binary representation.

Examples:

- Scheduling problems
- Integer / binary linear programming
- Feature selection
- ...

RECOMBINATION FOR BIT STRINGS

Two individuals $\mathbf{x}, \tilde{\mathbf{x}} \in \{0, 1\}^d$ encoded as bit strings can be recombined as follows:

- **1-point crossover:** Select crossover $k \in \{1, ..., d-1\}$ randomly. Take first k bits from parent 1 and last $d-k$ bits from parent 2.

$$\begin{array}{ccccc} \color{red}{1} & \color{blue}{1} & & \color{red}{1} \\ \color{red}{0} & \color{blue}{0} & & \color{red}{0} \\ \hline \color{red}{0} & \color{blue}{1} & \Rightarrow & \color{blue}{1} \\ \color{red}{1} & \color{blue}{1} & & \color{blue}{1} \\ \color{red}{1} & \color{blue}{0} & & \color{blue}{0} \end{array}$$

- **Uniform crossover:** Select bit j with probability p from parent 1 and $1-p$ from parent 2.

$$\begin{array}{ccccc} \color{red}{1} & \color{blue}{0} & & \color{red}{1} \\ \color{red}{0} & \color{blue}{0} & & \color{blue}{0} \\ \color{red}{0} & \color{blue}{1} & \Rightarrow & \color{blue}{1} \\ \color{red}{0} & \color{blue}{1} & & \color{blue}{1} \\ \color{red}{1} & \color{blue}{0} & & \color{red}{1} \end{array}$$

MUTATION FOR BIT STRINGS

Offspring $\mathbf{x} \in \{0, 1\}^d$ encoded as a bit string can be mutated as follows:

- **Bitflip:** Each bit j is flipped with probability $p \in (0, 1)$.

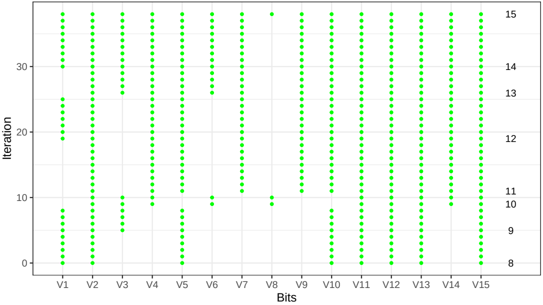
$$\begin{array}{ccc} 1 & & \color{red}{0} \\ 0 & & 0 \\ 0 & \Rightarrow & 0 \\ 0 & & \color{red}{1} \\ 1 & & 1 \end{array}$$

EXAMPLE 1: ONE-MAX EXAMPLE

$\mathbf{x} \in \{0, 1\}^d$, $d = 15$ bit vector representation.

Goal: Find the vector with the maximum number of 1's.

- Fitness: $f(\mathbf{x}) = \sum_{i=1}^{d} x_i$
- $\mu = 15$, $\lambda = 5$, $(\mu + \lambda)$-strategy, bitflip mutation, no recombination



Green: Representation of best individual per iteration. Right scale shows fitness.

EXAMPLE 2: FEATURE SELECTION

We consider the following toy setting:

- Generate design matrix $\mathbf{X} \in \mathbb{R}^{n \times p}$ by drawing $n = 1000$ samples of $p = 50$ independent normally distributed features with $\mu_j = 0$ and $\sigma_j^2 > 0$ varying between 1 and 5 for $j = 1, \ldots, p$.
- Linear regression problem with dependent variable $\mathbf{y}$:

$$\mathbf{y} = \mathbf{X}\boldsymbol{\theta} + \epsilon$$

with $\epsilon \sim \mathcal{N}(0, 1)$.

Parameter $\boldsymbol{\theta}$:

$$\theta_0 = -1.2$$
$$\theta_j = \begin{cases} 1 & \text{for } j \in 1, 7, 13, 19, 25, 31, 37, 43 \\ 0 & \text{otherwise} \end{cases}$$

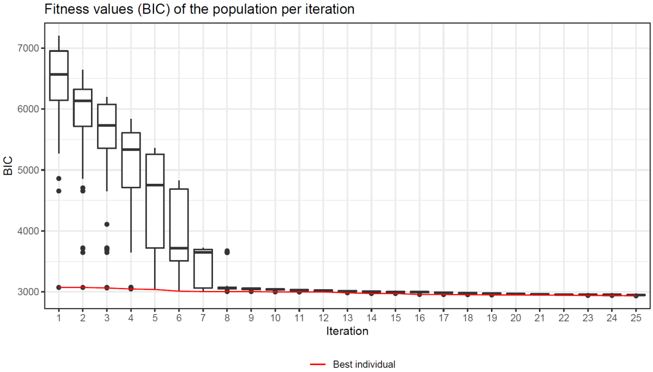
$\Rightarrow$ Only 8 out of 50 equally influential features

EXAMPLE 2: FEATURE SELECTION

- **Aim:** Find influential features
- **Encoding:** $\mathbf{z} \in \{0, 1\}^p$, $z_j = 1$ means θ_j included in model
- **Fitness function** $f(\mathbf{z})$: BIC of the model belonging to $\mathbf{z}$
- **Mutation:** Bit flip with $p = 0.3$
- **Recombination:** Uniform crossover with $p = 0.5$
- **Survival selection:** $(\mu + \lambda)$ strategy with $\mu = 100$ and $\lambda = 50$

```
## [1] "After 10 iterations:"
##  [1]  1  7 11 13 14 15 19 20 22 25 30 31 36 37 40 43 44 48
## [19] 49 50
## [1] "After 20 iterations:"
##  [1]  1  7  8 13 15 19 20 25 31 37 43
## [1] "Included variables after 24 iterations:"
## [1]  1  7 13 19 25 31 37 43
```

EXAMPLE 2: FEATURE SELECTION



EXAMPLE 2: FEATURE SELECTION