Evolutionary Computing

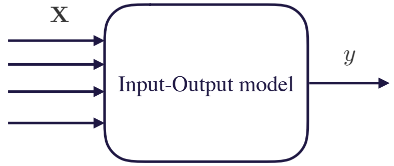
Fundamentals

Michela Mulas

A quick recap

Wrapping up...

Predictive models



In a typical scenario, we have an **outcome measurement**

- Quantitative or categorical

We wish to predict the outcome based on a set of features.

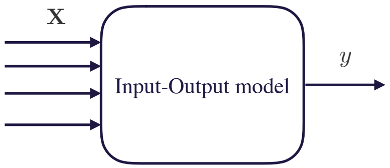
We have a **training set of data**

- We observe the outcome and feature measurements for a set of objects.

A quick recap

Wrapping up...

Predictive models



We have a **training set of data**

- We observe the outcome and feature measurements for a set of objects.

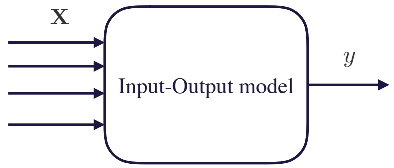
Using this data we build a prediction model, or learner.

- It will enable us to predict the outcome for new unseen objects.
  A good learner is one that accurately predicts such an outcome.

A quick recap

Wrapping up...

Predictive models



This is an example of the **supervised learning problem**.

- It is called "supervised" because of the presence of the outcome variable to guide the learning process.

In the **unsupervised learning problem**[1], we observe only the features and have no measurements of the outcome.

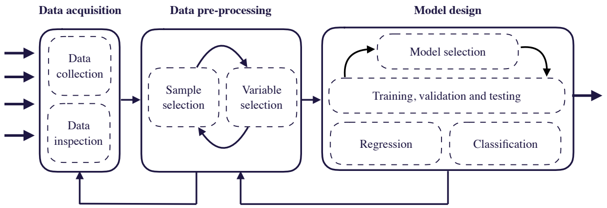
- Our task is rather to describe how the data are organized or clustered.

[1] We did not consider this problem during the course.

A quick recap

Wrapping up...

Predictive models

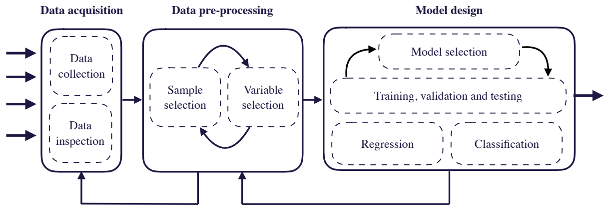


Predictive modeling is the process by which a model is created or chosen to try to best predict the probability of an outcome

A quick recap

Wrapping up...

Predictive models



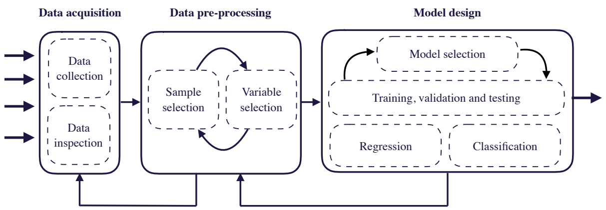
Data pre-processing

- Addition, deletion or transformation of training set data.
- Data transformation for individual predictors.
- Data transformation for multiple predictors.
  - ⇝ Principal component analysis.

A quick recap

Wrapping up...

Predictive models



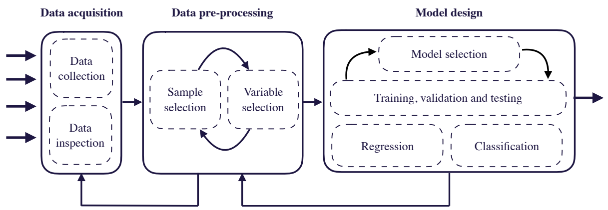
Data pre-processing

- Removing predictors: Between-predictors correlation.
- Adding predictors: create “dummy” variables.

A quick recap

Wrapping up...

Predictive models



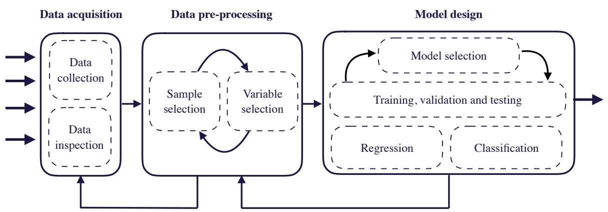
Data spending

- The problem of over-fitting.
  - ⇝ The **bias-variance trade-off**.
- Resampling techniques.
  - ⇝ k-fold cross validation
  - Repeated train/test splits
  - Bootstrap

A quick recap

Wrapping up...

Predictive models



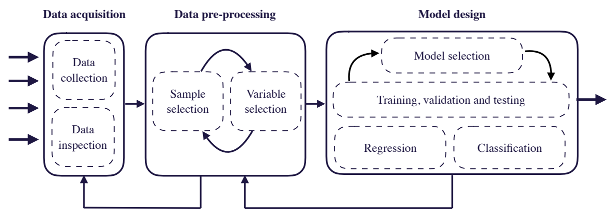
Models for regression

- Ordinary least squares models.
- Penalized models.
- Principal component regression.
- Partial least squares.
- Neural networks.

A quick recap

Wrapping up...

Predictive models



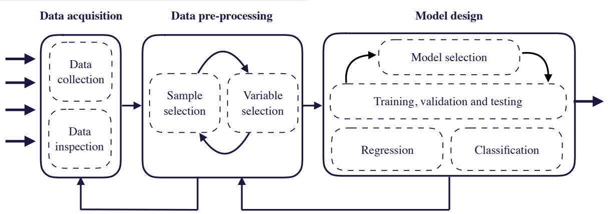
Models for regression – Measuring performance

- Root mean square error.
- Coefficient of determination.

A quick recap

Wrapping up...

Predictive models



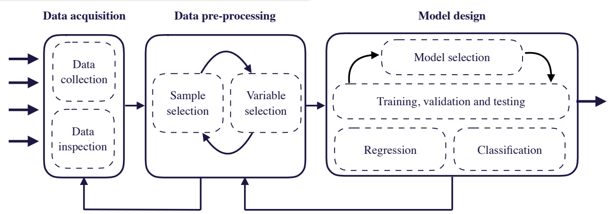
Models for classification

- Logistic regression.
- LDA and QDA.
- K-nearest neighbor.
- Support vector machines.
- (Neural networks).

A quick recap

Wrapping up...

Predictive models



Models for classification – Measuring performances

- Confusion matrix.
- Receiver operating characteristics (ROC) curve.

A quick recap

Wrapping up... Predictive models

- **Confusion matrix**: It is a simple cross-tabulation of the observed and predicted classes for the data.

		Predicted class		
		$-$ or Null	$+$ or Non-Null	Total
True Class	$-$ or Null	True Neg. (TN)	False Pos. (FP)	N
	$+$ or Non-Null	False Neg. (FN)	True Pos. (TF)	P
	Total	N^*	P^*	

⇝ It shows the possible results of a classification.
⇝ $+$ and $-$ are the two class that we are trying to classify.

A quick recap

Wrapping up... Predictive models

- **Confusion matrix**: Important measures for classification and diagnostic testing can be derived from it.

Name	Definition	Synonyms
False Pos. Rate	FP/N	Type I error, 1$-$Specificity
True Pos. Rate	TP/P	1$-$Type II error, power, sensitivity, recall
Pos. Pred. Value	TP/P*	Precision, 1$-$ false discovery proportion
Neg. Pred. value	TN/N*	Type I error - Specificity

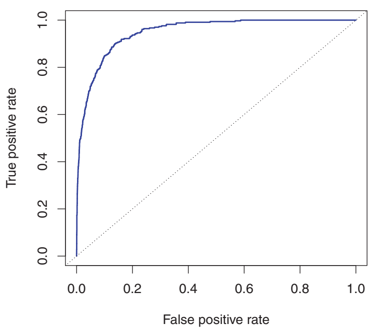
⇝ The denominators for the false positive and true positive rates are the actual population counts in each class.

⇝ The denominators for the positive predictive value and the negative predictive value are the total predicted counts for each class.

A quick recap

Wrapping up... Predictive models

- **Receiver operating characteristics (ROC) curve**: can be used for determining alternate cutoffs for class probabilities.



⇝ The overall performance of a classifier, summarized over all possible thresholds, is given by the area under the ROC curve (AUC).

⇝ An ideal ROC curve will hug the top left corner, so the larger the AUC the better the classifier.

⇝ ROC curves are useful for comparing different classifiers, since they take into account all possible thresholds.

A quick recap

Wrapping up... Predictive models

- **Reading list**

Max Kuhn and Kjell Johnson.
Applied Predictive Modeling, Springer (2014)

Jerome H. Friedman, Robert Tibshirani, and Trevor Hastie.
The Elements of Statistical Learning, Springer (2017)

Gareth James, Daniela Witten, Trevor Hastie and Robert Tibshirani.
An Introduction to Statistical Learning with Applications in R, Springer (2017)

Simon Haykin.
Neural Networks: A Comprehensive Foundation, Pearson (2008)

Today's goal

Today, we going to do...

Evolutionary computing

- A bit of history
- Basic concepts
  - Naive evolution theory
  - Different paradigms
  - Main components
- Applications

Reading list

- A.E. Eiben and J.E. Smith, Introduction to Evolutionary Computing, Springer, 2007
- A.P. Engelbrecht *Computational Intelligence – An Introduction*, Wiley 2007.

Why Evolutionary Computing?

Motivations

Evolutionary Computing (EC) refers to **computer-based problem solving systems that use computational models of evolutionary processes**, such as natural selection, survival of the fittest and reproduction, as the fundamental components of such computational systems.

- EC is part of computer science
- EC is not part of life sciences/biology
- Biology delivered inspiration and terminology

Evolution	Problem Solving
Environment	Problem
Individual	Candidate solution
Fitness	Quality

Why Evolutionary Computing?

Motivations

- **Conceptual Simplicity**. A primary advantage of evolutionary computing is that it is conceptually simple.
- **Broad Applicability**. EC algorithms can be applied to virtually any problem that can be formulated as a function optimisation task. It requires a data structure to represent solutions, a performance index to evaluate solutions, and variation operators to generate new solutions from old solutions.
- **Outperform Classic Methods on Real Problems**. Real-world function optimisation problems often (i) impose nonlinear constraints, (ii) require payoff functions that are not concerned with least squared error, (iii) involve non-stationary conditions, (iv) incorporate noisy observations or random processing.

Why Evolutionary Computing?

Motivations

- **Non-smooth response surfaces**. Most real-world problems have non-smooth response surfaces.

  $\rightsquigarrow$ Traditional methods usually do not provide quality results.

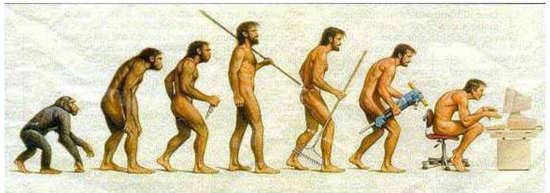
  Evolutionary techniques can be applied successfully to many problems of this type, since no gradient information is needed.
- **Size of the search space**. For most problems, the size of the search space of feasible solutions is huge: too large to consider exhaustive search.

  For such large search spaces, heuristic search methods are of importance.
- **Robust to dynamic changes**. Evolutionary algorithms can be used to adapt solutions to changing circumstance.

Imitate evolution

Attempts to define the term **biological evolution** still cause numerous debates, with the Lamarckian and Darwinian views being the most popular and accepted.

- **Jean-Baptiste Lamarck** (1744-1829) was the first to theorize about **biological evolution**. His main idea is base on heredity: the inheritance of acquired traits.
- **Charles Darwin** (1809-1882) is generally considered as the founder of both **theory of evolution and principle of common descent**.



Somewhere, something went wrong (Anonymous).

Imitate evolution

Pioneers

1948 Turing proposes **genetical** or **evolutionary search**.
1962 Bremermann optimisation through **evolution** and **recombination**.
1964 Rechenberg introduces **evolution strategies**.
1965 Fogel, Owens and Walsh introduce **evolutionary programming**.
1975 Holland introduces **genetic algorithms**.
1985 First international conference on genetic algorithms
1992 Koza introduces **genetic programming**.
1993 First scientific EC journal (MIT Press)
1997 Launch of European EC Research Network EvoNet
...

Applications

Today

Evolutionary computing is used successfully in many practical applications: data mining, combinatorial optimisation, fault diagnosis, classification, clustering, scheduling, and time series approximation.

... and tomorrow

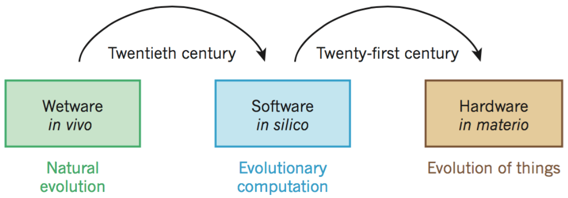
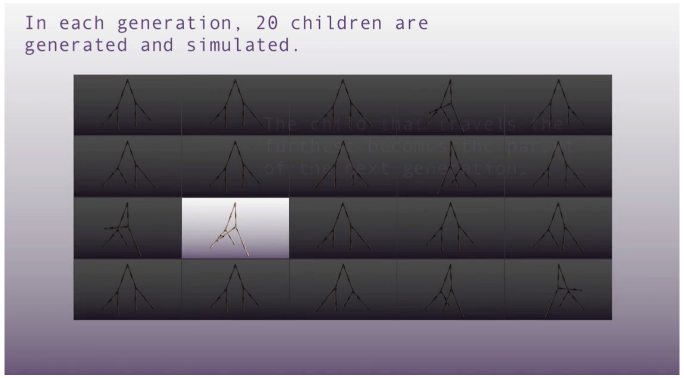


Image credits: A.E. Eiben and J. Smith. *From evolutionary computation to the evolution of things*, Nature, 521:476-482, 2015

Applications



- **Eugénie von Tunzelmann: Tunzelbots (2014)**[1]

[1] Source: https://vimeo.com/85053197

Why Evolutionary Computing?

Paradigms

- **Genetic algorithms** (GAs), which model genetic evolution.
- **Genetic programming** (GP), which is based on genetic algorithms, but individuals are programs (represented as trees).
- **Evolutionary programming** (EP), which is derived from the simulation of adaptive behaviour in evolution (i.e. phenotypic evolution).
- **Evolution strategies** (ESs), which are geared toward modelling the strategic parameters that control variation in evolution, i.e. the evolution of evolution.
- **Differential evolution** (DE), which is similar to genetic algorithms, differing in the reproduction mechanism used.
- **Cultural evolution** (CE), which models the evolution of culture of a population and how the culture influences the genetic and phenotypic evolution of individuals.
- **Co-evolution** (CoE), where initially “dumb" individuals evolve through cooperation, or in competition with one another, acquiring the necessary characteristics to survive.

Evolutionary Algorithm definitions

Biological inspiration

Darwin’s evolutionary idea

- A **population** of individuals exists in an environment with limited resources.
- Competition for those resources causes selection of those fitter individuals that are better adapted to the environment.
- These individuals act as seeds for the generation of new individuals through **recombination and mutation**.
- The new individuals have their **fitness** evaluated and compete (possibly also with **parents**) for survival.
- Over time **natural selection** causes a rise in the fitness of the population.

Evolutionary Algorithm definitions

General scheme of an evolutionary algorithm

Evolution via natural selection of a randomly chosen population of individuals can be thought of as a search through the space of possible chromosome values.

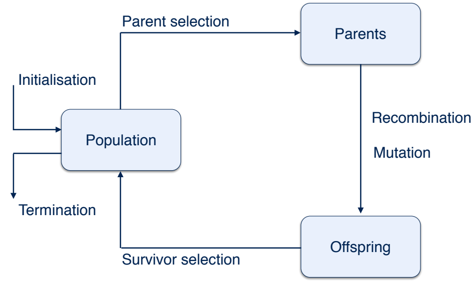
- An evolutionary algorithm (EA) is a **stochastic search for an optimal solution** to a given problem.
- EA falls into the category of “**generate and test**” algorithms.
- Variation operators (**recombination** and **mutation**) create the necessary diversity and thereby facilitate novelty.
- **Selection** reduces diversity and acts as a force pushing quality.

Evolutionary Algorithm definitions

General scheme of an evolutionary algorithm

The evolutionary search is influenced by:

- An **encoding** of solutions to the problem, as a chromosome;
- A function to evaluate the **fitness** or survival strength of individuals;
- **Initialisation** of the initial population;
- **Selection** operators;
- **Reproduction** operators.



Evolutionary Algorithm definitions
General scheme of an evolutionary algorithm

Algorithm 1 Generic Evolutionary Algorithm

Let $t = 0$ be the generation counter;
Create/initialize the n_x-dimensional population, $C(0)$, to consist of n_s individuals

```
1: while stopping condition(s) not true do
2:     Evaluate the fitness, f(x_i(t)), of each individual, x_i(t) (select the parents);
3:     Perform reproduction to create offsprings (reproduction and mutation);
4:     Select a new population, C(t+1) (survivor selection);
5:     Advance to the new generation, i.e., t = t+1;
6: end while
```

Darwin's theory is encapsulated within this algorithm:

- Natural selection occurs within the reproduction operation where the "best" parents have a better chance of being selected to produce offspring, and to be selected for the new population.
- Random changes are effected through the mutation operator.

Evolutionary Algorithm components
The chromosome

Nature

- Organisms have certain characteristics represented by long strings of information contained in the **chromosomes**.
- Chromosomes are structures of compact intertwined molecules of DNA, found in the nucleus of organic cells.
- Each chromosome contains a large number of **genes**, which are the unit of heredity.

Evolutionary Algorithm

- These characteristics refer to the variables of the optimisation problem, for which an optimal assignment is sought.
- Each individual represents a candidate solution to an optimisation problem. Candidate solutions (individuals) exist in **phenotype** space.
- They are encoded in **chromosome**, also referred to as a **genome**, which exist in **genotype** space.

Evolutionary Algorithm components
The chromosome

Nature

- Genes determine many aspects of anatomy and physiology through control of protein production.
- Each individual has a unique sequence of genes, or **allele**.

Evolutionary Algorithm

- Chromosomes contain **genes**, which are in (usually fixed) positions called **loci** (sing. locus) and have a value (**allele**).

Evolutionary Algorithm components
The chromosome

- A genotype describes the genetic composition of an individual, as inherited from its parents; it represents which allele the individual possesses.
- A phenotype is the expressed behavioural traits of an individual in a specific environment; it defines what an individual looks like.

> In order to find the global optimum, every feasible solution must be represented in genotype space.

Evolutionary Algorithm components

Representation

An important step in the design of an EA is to **find an appropriate representation of candidate solutions** (i.e., chromosomes).

The efficiency and complexity of the search algorithm greatly depends on the representation scheme.

- Binary strings: Genetic Algorithms
- Real-valued vectors: Evolution Strategies
- Finite state Machines: Evolutionary Programming
- LISP trees: Genetic Programming

The best strategy is **choose representation to suit problem** and to **choose variation operators to suit representation**

Evolutionary Algorithm components

Initial population

The first step in applying an evolutionary algorithm to solve an optimisation problem is to **generate an initial population**.

- The standard way of generating an initial population is to assign a **random value from the allowed domain** to each of the genes of each chromosome.
- The goal of random selection is to **ensure that the initial population is a uniform representation** of the entire search space. If regions of the search space are not covered by the initial population, chances are that those parts will be neglected by the search process.

Evolutionary Algorithm components

Initial population

The first step in applying an evolutionary algorithm to solve an optimisation problem is to **generate an initial population**.

- The **size of the initial population** has consequences in terms of computational complexity and exploration abilities.
  - Large numbers of individuals **increase diversity**, improving the exploration abilities of the population.
    - ⇝ The more the individuals, the **higher the computational complexity per generation**.
    - ⇝ While the execution time per generation increases, it may be the case that fewer generations are needed to locate an acceptable solution.
  - A small population will represent a **small part of the search space**.
    - ⇝ While the time complexity per generation is low, the EA may need more generations to converge than for a large population.
    - ⇝ The EA can be forced to explore more of the search space by increasing the rate of mutation.

Evolutionary Algorithm components

Fitness function

Nature

- In the Darwinian model of evolution, **individuals with the best characteristics** have the best chance to survive and to reproduce. The main concept is survival of the fittest.

Evolutionary Algorithm

- To determine the ability of an individual of an EA to survive, a **fitness function** is used to quantify how good the solution represented by a chromosome is.

Usually, the fitness function provides an **absolute measure of fitness**

The solution represented by a chromosome is directly evaluated using the objective function.

Evolutionary Algorithm components
Fitness function

The formulation of the fitness function is influenced by the optimisation problem:

- **Unconstrained** optimisation problems: the fitness function is simply the objective function.
- **Constrained** optimisation problems: some EAs change the fitness function to contain two objectives:
  1. The original objective function
  2. The other is a constraint penalty function
- **Multi-objective** optimisation problems can be solved by using a weighted aggregation approach, where the fitness function is a weighted sum of all the sub-objectives or by using a Pareto-based optimisation algorithm.
- **Dynamic and noisy** problems, where function values of solutions change over time. Dynamic fitness functions are time-dependent whereas noisy functions usually have an added Gaussian noise component.

Evolutionary Algorithm components
Selection operators

The main objective of selection operators is to emphasise better solutions.

This is achieved in two of the main steps of an EA:

- **Selection of the new population**: at the end of each generation to serve as the population of the next generation.
  - ⇝ The new population can be selected from only the offspring, or from both the parents and the offspring.
  - ⇝ The selection operator should ensure that good individuals do survive to next generations.

Evolutionary Algorithm components
Selection operators

The main objective of selection operators is to emphasise better solutions. This is achieved in two of the main steps of an EA:

- **Reproduction**: Offspring are created through the application of **crossover** and/or **mutation** operators.
  - **Crossover**: “superior" individuals should have more opportunities to reproduce to ensure that offspring contain genetic material of the best individuals.
  - **Mutation**: “weak" individuals should mutate and result in better traits (increasing the chances of survival to weak individuals).

Evolutionary Algorithm components
Selection operators

Selective pressure

- It is also referred as **takeover time**.
- It relates to the time it requires to produce a uniform population.
- It is defined as the speed at which the best solution will occupy the entire population by repeated application of the selection operator alone.
- An operator with a high selective pressure decreases diversity in the population more rapidly than operators with a low selective pressure, which may lead to premature convergence to suboptimal solutions.
- A high selective pressure limits the exploration abilities of the population.

Evolutionary Algorithm components
Parent selection mechanisms

The role of parent selection or **mating selection** is to distinguish among individuals based on their quality, to **allow the better individuals to become parents** of the next generation.

> The parent selection assigns variable probabilities of individuals acting as parents depending on their fitnesses. An individual is a parent is it has been selected to undergo variation in order to create offspring.

In EC, **parent selection is usually probabilistic**:

- Higher quality individuals get higher chances to become parents than those with lower quality.
- Low quality individuals are often given a small, but positive chance, otherwise the whole search would become to greedy and get stuck in a local optimum.
- This stochastic nature can aid escape from local optima.

Evolutionary Algorithm components
Variation operators

The role of variation operators is to create new individuals from old ones. Variation operators are divided into two types based on their **arity** (i.e., the number of objects that it takes as inputs).

Mutation: Arity = 1.

- Mutation is applied to one genotype and delivers a (slightly) modified mutant, the child or offspring of it.
- A mutation operator is always stochastic: its output, the child, depends on the outcomes of a series of random choices.
- Element of randomness is essential and differentiates it from other unary heuristic operators

Evolutionary Algorithm components
Variation operators

The role of variation operators is to create new individuals from old ones. Variation operators are divided into two types based on their **arity** (i.e., the number of objects that it takes as inputs).

Recombination: Arity > 1

- Recombination merges information from two parent genotypes into one or two offspring genotypes.
- The choice of what information to merge is stochastic.
- Most offspring may be worse, or the same as the parents. Hope is that some are better by combining elements of genotypes that lead to good traits.
- Principle has been used for millennia by breeders of plants and livestock.

Variation operators are representation and paradigm dependent: **for different representations different variation operators have to be defined**.

Evolutionary Algorithm components
Survivor selection mechanism

The role of survivor selection or environmental selection is to **distinguish among individuals based on their quality**. It is similar to parent selection, but it is used in a different stage of the evolutionary cycle.

- Most EAs use fixed population size so need a way of going from (parents + offspring) to next generation.
- Survivor selection is usually called **replacement**. In many cases, the two terms are interchangeably.
- It is often deterministic:
  - **Fitness based**: e.g., rank parents+offspring and take best;
  - **Age based**: make as many offspring as parents and delete all parents.
- Sometimes do combination (**elitism**).

Evolutionary Algorithm components

Survivor selection mechanism

The role of survivor selection or environmental selection is to **distinguish among individuals based on their quality**. It is similar to parent selection, but it is used in a different stage of the evolutionary cycle.

- Elitism refers to the process of **ensuring that the best individuals of the current population survive to the next generation**.
  - The best individuals are copied to the new population without being mutated.
  - The more individuals that survive to the next generation, the **less the diversity of the new population**.

Stopping conditions

The evolutionary operators are iteratively applied in an EA until a stopping condition is satisfied.

The **simplest stopping condition is to limit the number of generations that the EA is allowed to execute**, or alternatively, a limit is placed on the number of fitness function evaluations. This limit should not be too small, otherwise the EA will not have sufficient time to explore the search space.

A **convergence criterion** is usually used to detect if the population has converged. Convergence is loosely defined as the event when the population becomes stagnant. In other words, when there is no genotypic or phenotypic change in the population.

Stopping conditions

The evolutionary operators are iteratively applied in an EA until a stopping condition is satisfied.

The following convergence criteria can be used:

- Terminate when no improvement is observed over a number of consecutive generations.
- Terminate when there is no change in the population.
- Terminate when an acceptable solution has been found.
- Terminate when the objective function slope is approximately zero.