GA

Four types of problems

- black box model
- search problems
- optimization vs constraint satisfaction
- NP problems

black box model

- optimization: find input
- modelling: find model
- simulation: find output

search problems

Search space: collection of all objects of interest including the desired solution

optimization vs constraint satisfaction

Objective function: a way of assigning a value to a possible solution that reflects its quality on scale

Constraint: binary evaluation telling whether a given requirement holds or not

	Objective function	
Constraints	Yes	No
Yes	Constrained optimisation problem	Constraint satisfaction problem
No	Free optimisation problem	No problem

NP problems

- class P: algorithm can solve the problem in polynomial time
- class NP: problem can be solved, and any solution can be verified within polynomial time by some o algorithm (P subset of NP)
- Class NP-complete: problem belongs to class NP and any other problem in NP can be reduced to th problem by an algorithm running in polynomial time
- Class NP-hard: problem is at least as hard as any other problem in NP-complete but solution cannot necessarily be verified within polynomial time

Evolutionary Algorithm

Individuals are “units of selection”
Population is the “unit of evolution”

Phenotypic traits
Genotype (DNA inside) determines phenotype

Scheme of an EA

Part1 P35

Typical EA behaviour: Stages

- Early stage: quasi-random population distribution
- Mid-stage: population arranged around/on hills
- Late stage: population concentrated on high hills

SGA-Step

- SGA-Step 1: Encoding the Decision Variable
- SGA - Step 2: Initial Population
- SGA-Step 3: Computing Fitness Proportions
- SGA - Step 4: Reproduction
- SGA - Step 5: Crossover
- SGA-Step 6: Mutation

Representation, Mutation, and Recombination

Most common representation of genomes:

- Binary
- Integer
- Real-Valued or Floating-Point
- Permutation
- Tree

Binary Representation: 1-point crossover

- Choose a random point on the two parents
- Split parents at this crossover point

Performance with 1-point crossover depends on the order that variables occur in the representation

More likely to keep together genes that are near each other

Binary Representation: n-point crossover

still some positional bias

Binary Representation: Uniform crossover

Flip a coin for each gene of the first child
Make an inverse copy of the gene for the second child

Binary Representation: Crossover OR mutation?

in general, it is good to have both
mutation-only-EA is possible, xover-only-EA would not work

Only crossover can combine information from two parents

Only mutation can introduce new information (alleles)

Integer Representation

Nowadays it is generally accepted that it is better to encode numerical variables directly (integers, floati variables)

Real-Valued or Floating-Point Representation

Real-Valued or Floating-Point Representation: Single arithmetic crossover

$(x_1, \ldots, x_n)$

$(y_1, \ldots, y_n)$

$(x_1, \ldots, x_k, \alpha \cdot y_k + (1-\alpha) \cdot x_k, \ldots, x_n)$

Real-Valued or Floating-Point Representation: Simple arithmetic crossover

$(x_1, \ldots, x_n)$

$(y_1, \ldots, y_n)$

$(x_1, \ldots, x_k, \alpha \cdot y_{k+1} + (1-\alpha) \cdot x_{k+1}, \ldots, \alpha \cdot y_n + (1-\alpha) \cdot x_n)$

Real-Valued or Floating-Point Representation: Whole arithmetic crossover

$\alpha \cdot \mathbf{x} + (1-\alpha) \cdot \mathbf{y}$

Real-Valued or Floating-Point Representation: Blend Crossover

这张图片介绍了遗传算法中实数值或浮点数表示的混合交叉（Blend Crossover）方法。该方法的步骤如下：

1. 确定两个父本的基因序列，表示为 $(x_1, \ldots, x_n)$ 和 $(y_1, \ldots, y_n)$。

2. 假设 $x_i < y_i$（对每一个 i 都成立）。
3. 定义 $d_i = y_i - x_i$，即两个父本在第 i 个位置的基因值之差。
4. 通过随机采样得到新的子代基因值 z_i，采样范围为 $[x_i - \alpha d_i, x_i + \alpha d_i]$。

在这里，α 是一个扩展因子，用于确定采样范围的宽度，原始作者发现 $\alpha = 0.5$ 时能够得到最佳结果

Permutation Representations: Swap mutation

Pick two alleles at random and swap their positions

Permutation Representations: Insert Mutation

Pick two allele values at random

Move the second to follow the first, shifting the rest along to accommodate

Permutation Representations: Scramble mutation

Pick a subset of genes at random

Randomly rearrange the alleles in those positions

Permutation Representations: Inversion mutation

Pick two alleles at random and then invert the substring between them.

Permutation Representations: Order 1 crossover

- Choose an arbitrary part from the first parent
- Copy this part to the first child
- Copy the numbers that are not in the first part, to the first child:
  - starting right from cut point of the copied part,
  - using the order of the second parent
  - and wrapping around at the end

1. **选择父代A的一个任意部分**

2. **将这部分复制到子代**

3. **从父代B开始，将剩余的数字按照父代B的顺序填入子代**，但要跳过已经在子代中的数字

Parent 1: 8 4 7 **3 6 2 5 1** 9 0

Parent 2: 0 1 2 3 4 5 6 7 8 9

Child 1: 0 4 7 **3 6 2 5 1** 8 9

Permutation Representations: Partially Mapped Crossover (PMX)

P125/63

Permutation Representations: Cycle crossover

ML2

Handling outliers

- Remove outliers: If the number of records which are outliers is not many, a simple approach may be remove them
- Imputation: One other way is to impute the value with mean or median or mode. The value of the mc data element may also be used for imputation.
- Capping: For values that lie outside the 1.5 times of IQR limits, we can cap them by replacing those observations below the lower limit with the value of 5th percentile and those that lie above the uppe with the value of 95th percentile.

Handling missing values

- Eliminate records having a missing value of data elements
- Imputing missing values
- Estimate missing values

Normalization

Normalization

$x' = \frac{x - x_{\min}}{x_{\max} - x_{\min}}$

Standardization

$x' = \frac{x - \mu}{\sigma}$

ML3

$p(x|\omega) = \frac{1}{\sqrt{2\pi}\sigma}\exp\left(-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2\right)$

$\mu = \int_{-\infty}^{+\infty} xp(x|\omega)dx$

$\sigma^2 = \int_{-\infty}^{+\infty} (x-\mu)^2 p(x|\omega)dx$

$p(x|\omega_j) = \frac{1}{(2\pi)^{d/2}|\Sigma_j|^{1/2}}\exp\left(-\frac{1}{2}(x-\mu_j)^T\Sigma_j^{-1}(x-\mu_j)\right)$

ML4

$g(x) = W^T x + w_0$

- 如果 $g(x) > 0$，则决定 ω_1。
- 如果 $g(x) < 0$，则决定 ω_2。

$S_W = S_1 + S_2$

m_i is the d-dimensional sample mean of class i

scatter of data s

scatter matrices S

$w = S_w^{-1}(m_1 - m_2)$

$w = (S_w + \beta I)^{-1}(m_1 - m_2)$

$w_0 = -\frac{w^T(m_1+m_2)}{2}$

ML5. Support Vector Machines

ML6

ML7

1. **准确率（Accuracy）**：
   $\text{Accuracy} = \frac{TP+TN}{TP+FP+TN+FN}$ 表示所有分类正确的样本占总样本的比例。
2. **错误率（Error Rate）**：
   $\text{Error Rate} = \frac{FP+FN}{TP+FP+TN+FN} = 1 - \text{Accuracy}$ 表示所有分类错误的样本占总样本的比例。
3. **灵敏度（Sensitivity）或召回率（Recall）**：
   $\text{Sensitivity (Recall)} = \frac{TP}{TP+FN}$ 表示所有实际为正类的样本中被正确识别为正类的比例。
4. **特异性（Specificity）**：
   $\text{Specificity} = \frac{TN}{TN+FP}$ 表示所有实际为负类的样本中被正确识别为负类的比例。
5. **精确度（Precision）**：
   $\text{Precision} = \frac{TP}{TP+FP}$ 表示所有预测为正类的样本中实际为正类的比例。
6. **F分数（F-score）**：

   - 特别地，F1分数是精确度和召回率的调和平均：$\text{F1 Score} = 2 \times \frac{\text{Precision}\times\text{Recall}}{\text{Precision}+\text{Recall}}$ 表示精确度和召衡点。

ML9

subset selection algorithm

Search algorithm

Exhaustive search method

Sequential forward selection

Sequential backward elimination

Evaluation criterion

Classification performance of a classifier

Separability measures

Method

P9 P24

ML10