Introduction to Program Synthesis (WS 2024/25)

Chapter 3.2 - Traditional Methodologies (Exemplification and Problems)

Dr. rer. nat. Roman Kalkreuth

Chair for AI Methodology (**AIM**), Department of Computer Science,
RWTH Aachen University, Germany

Center for Artificial Intelligence

RWTH AACHEN UNIVERSITY

Traditional Methodologies
Exemplification and problem domains

- Search spaces in program synthesis → Prone to combinatorial explosion
  - Naive approach with brute-force search or random walk is a dead end
- Popular deterministic algorithmic paradigms to efficiently tackle combinatorial problems
  - Backtracking
  - Divide and conquer
  - Branch and bound
  - Dynamic programming → Memoization, Tabulation
- Popular heuristic methods:
  - Local search
  - Greedy search
- Commonly used either base (*"vanilla"*) or extended form in combinatorial optimization

Traditional Methodologies

Exemplification and problem domains

- Practical demonstration and application of such techniques to combinatorial problems
- Problems from games $\rightarrow$ Chess
- Common problem domains of program synthesis $\rightarrow$ Symbolic regression, logic synthesis, algorithm design

Traditional Methodologies

Exemplification and problem domains: Knights Tour Problem (KTP)

- Classic puzzle in chess where the goal is to visit every field on a chessboard
  - ⤳ Combinatorial problem
  - ⤳ Each square is exactly visited once
  - ⤳ The knight must make a legal move in L shape

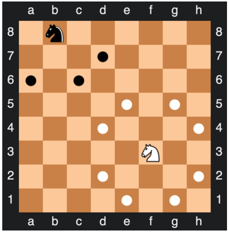


Figure: Examples of valid knight moves (Source: Wikimedia

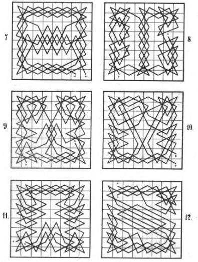


Figure: Example of open knight tours (Source: G. Mann, 120 neue Rösselsprünge, 1859)

Traditional Methodologies

Exemplification and problem domains: Knights Tour Problem (KTP)

- A tour is called closed if the end square of the knight is one knight move away from the start square
  - $\rightsquigarrow$ Otherwise the path is called open
- Finding a closed tour $\rightarrow$ Similar instance of the *Hamiltonian cycle problem*
- Determining whether a directed or undirected graph G contains a Hamiltonian cycle
  - $\rightsquigarrow$ NP-Complete problem
  - $\rightsquigarrow$ However, KTP can be solved in linear time
- Hamiltonian cycle on the *knight graph*

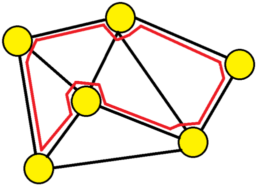


Figure: Example of a Hamiltonian cycle)

Definition (Hamiltonian Path)

A **Hamiltonian path** is a path between two vertices v_1, v_2 of a graph G that visits each vertex **exactly once**. If endpoints of a Hamiltonian path are **adjacent**, then the resulting cycle is called a **Hamiltonian cycle**. Let $G = (V, E)$ a graph with $|V| = n$ vertices and $|E| = m$ edges. G is *hamiltonian* if there exists a cycle in G that contains all vertices of V. If G has Hamiltonian paths but no Hamiltonian cycle, G is called *semi-hamiltonian*

- A knight's tour exists on an $n \times n$ board iff $n \geq 5$
- Number of possible knight moves grows exponentially as the board size increases

Board size	Number of tours
5x5	1,728
6x6	6,637,920
7x7	165,575,218,320
8x8	19,591,828,170,979,904

Traditional Methodologies

Exemplification and problem domains: Symbolic Regression (SR)

- Symbolic regression $\rightarrow$ type of regression analysis
  - Located in the wider domain of statistical learning
- Regression analysis $\rightarrow$ **statistical processes** that estimates **relationships** between **dependent** and **independent variables**
- Popular representatives:
  - $\leadsto$ **Linear regression** $\rightarrow$ Predicting responses to data in linear correlated data
  - $\leadsto$ **Logistic regression** $\rightarrow$ Prediction of probabilities based on a logistic function
  - $\leadsto$ **Symbolic regression**

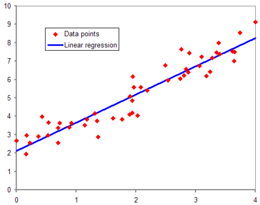


Figure: Linear Regression (Source: Wikimedia)

Traditional Methodologies

Exemplification and problem domains: Symbolic Regression (SR)

- **Regression Model** $\rightarrow$ applied to parameter estimation
- Regression models typically consists of
  - $\rightsquigarrow$ Unknown parameters $\rightarrow \beta$
  - $\rightsquigarrow$ Independent variables (known) $\rightarrow X$
  - $\rightsquigarrow$ Dependent variables (known) $\rightarrow Y$
  - $\rightsquigarrow$ Error terms or residuals (unobserved) $\rightarrow \epsilon$
- The regression model can be considered as a function $\rightarrow$ $Y_i = f(X_i, \beta) + \epsilon_i$
  - $\rightsquigarrow$ Reformulation as an optimization problem that minimizes the error terms: $\epsilon = Y_i - f(X_i, \beta)$

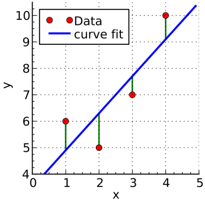


Figure: Least squares fitting (Source: Wikimedia)

Traditional Methodologies

Exemplification and problem domains: Symbolic Regression (SR)

- Let $Y \approx \beta_0 + \beta_1 * X + \epsilon$ be a linear regression model and let $T = \{y_i, x_i\}_{i=1}^{n}$ a dataset
- Corresponding minimization problem $\epsilon = Y - \beta_0 + \beta_1 * X$
- Cost function $\rightarrow$ residual sum of squares (RSS)
  - $\rightsquigarrow$ $\text{RSS} = \epsilon_1^2 + \epsilon_2^2, ..., \epsilon_n^2 = \sum_{i=1}^{n} \epsilon_i^2$
  - $\rightsquigarrow$ Residual calculation $\rightarrow \epsilon_i = Y_i - \hat{Y}$
  - $\rightsquigarrow$ Parameter estimation $\rightarrow$ (linear) least-squares method
    - $\rightsquigarrow$ Minimization of the sum of squared residuals

Traditional Methodologies

Exemplification and problem domains: Symbolic Regression (SR)

- Symbolic regression needs a representation for a symbolic model $\rightarrow$ expression trees
- Fitting a symbolic expression that can predict responses to a given dataset
- Corresponding mathematical expression is obtained
  - Can fit slopes and curves $\rightarrow$ applicable to linear and non-linear regression

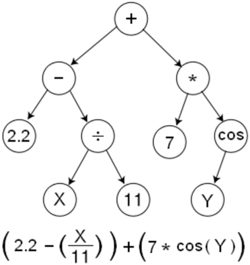


Figure: Expression tree representing a mathematical expression (Source: Wikimedia)

Traditional Methodologies

Exemplification and problem domains: Symbolic Regression (SR)

- Training dataset generation $\rightarrow$ Random points are drawn from uniform distribution
  - $\rightsquigarrow$ $\mathcal{X} = \{x | x \sim U(a, b), x \in \mathbb{R}\}$
- $\mathcal{X}$ is applied to the objective function $\mathcal{F}$
  - $\rightsquigarrow$ $\hat{\mathcal{Y}} = \{\hat{y} | \hat{y} = \mathcal{F}(x), \forall x \in \mathcal{X}\}$
- Candidate expression tree t is evaluated against $\hat{\mathcal{Y}}$ with $t(\mathcal{X}) \mapsto \mathcal{Y}$
  - $\rightsquigarrow$ Measure is needed to obtain the distance to the global optimum
  - $\rightsquigarrow$ Various distance metrics are used as cost function in SR:

Mean squared error	MSE	$\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$
Root mean squared error	RMSE	$\sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}$
Mean absolute error	MAE	$\frac{1}{n}\sum_{x=1}^{n}\lvert y_i - \hat{y}_i\rvert$

- **Cost function** $\mathcal{C} \rightarrow$ performance evaluation of candidate models
  - Candidate models $\rightarrow$ locally sampled expression trees
- **Loss function** $\mathcal{L} \rightarrow$ Comparison of actual (target) and predicted output values
  - Distance metric that is applied to each data point

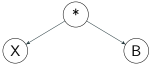
Definition (Symbolic Regression)

Let t be an expression tree in *tree space* $\mathcal{T}$. Find $t(\mathcal{X}) \mapsto \mathcal{C}_{min}$.
$t_{min} = \underset{t \in T}{\operatorname{argmin}} \mathcal{C}(t) := \{t \in \mathcal{T} \mid \mathcal{C}_{min}(t)\} := \{t_1 \in \mathcal{T} : \mathcal{C}(t_2) \geq \mathcal{C}(t_1) \; \forall t_2 \in \mathcal{T}\}$.

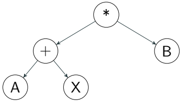
Traditional Methodologies

Exemplification and problem domains: Symbolic Regression (SR)

- **Neighbourhood function** $\mathcal{N}$ $\rightarrow$ replace a leaf node with a randomly generated subtree: $\mathcal{N}(\Psi) \mapsto \Psi'$



$\Psi = X * B$



$\Psi' = (A + X) * B$

- Stochastic hill climbing with restart is used as a search algorithm:
  - $\mathcal{R}$ $\rightarrow$ **Replacement function**: Selects a neighbour **with better cost** value **uniformly at random**
  - If no neighbour has a better cost value $\rightarrow$ Return and keep $\mathcal{P}$ (no replacement occurs)