

HOCHSCHULE FÜR TECHNIK, WIRTSCHAFT UND KULTUR LEIPZIG

FAKULTÄT INFORMATIK UND MEDIEN M. SC. INFORMATIK

EVOLUTIONÄRE ALGORITHMEN

Symbolische Regression mit evolutionären Algorithmen

*Karl - Augustin Jahnel
21INM, 76299
geb. am 06.09.1998
in Halle (Saale)
karl-augustin.jahnel@stud.htwk-leipzig.de*

Inhaltsverzeichnis

1	Motivation	2
2	Problembeschreibung	2
3	Lösungsansatz	2
3.0.1	Crossover	4
3.0.2	Mutation	4
3.1	Implementation	4
4	Informationen zum Nachschlagen	4
5	Alternativer Vorschlag - Kantenerkennung mit evolutionären Algorithmen	5

1 Motivation

Anwendungen symbolischer Regression in der echten Welt:

1. Vorhersage der nächsten Werte einer Zeitreihe
Aktienpreisvorhersage, Umsatzvorhersage...
2. Daten komprimieren
Eine mathematische Formel ist vielleicht die kürzest mögliche Darstellung eines Datensatzes. Wenn die Zielvariable eine gewisse Regelmäßigkeit aufweist, kann die symbolische Regression Gigabytes von Daten in etwas verwandeln, das in einer Zeile gleichwertig ausgedrückt werden kann.
3. grafische Darstellung der Funktionsweise evolutionärer Algorithmen

2 Problembeschreibung

Die symbolische Regression ist eine Art der Regressionsanalyse, die den Raum der mathematischen Ausdrücke durchsucht, um das Modell zu finden, das am besten zu einem gegebenen Datensatz passt, sowohl in Bezug auf die Genauigkeit als auch auf die Einfachheit. Dem Algorithmus wird kein bestimmtes Modell als Startpunkt vorgegeben. Stattdessen werden die anfänglichen Ausdrücke durch zufällige Kombination mathematischer Bausteine wie mathematische Operatoren, analytische Funktionen, Konstanten und Zustandsvariablen gebildet.

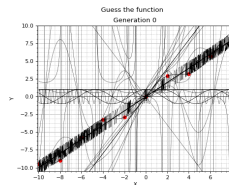


Abbildung 1: Symbolische Regression mit genetischer Programmierung

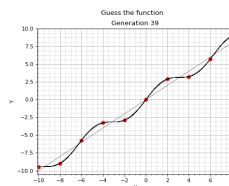


Abbildung 2: Symbolische Regression mit genetischer Programmierung nach 39 Generationen

3 Lösungsansatz

Jeder mathematische Ausdruck kann in Form eines Syntaxbaums dargestellt werden. Es gibt unendlich viele verschiedene Syntaxbäume, die semantisch äquivalenten Ausdrücken entsprechen. Zum Beispiel:

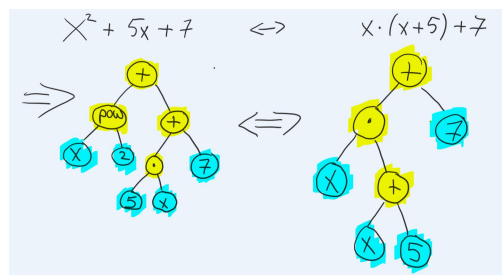


Abbildung 3: Mathematischer Ausdruck als Syntaxbaum

In Bezug auf den genetischen Algorithmus:

1. kann jeder Syntaxbaum als "Chromosom" behandelt werden (eine Einheit, die "mutieren" und sich durch "Crossover" mit anderen "Chromosomen" verändern kann).
2. Es ist notwendig, eine Fitnessfunktion zu definieren: die Funktion, die berechnet, wie gut jede Formel (die durch den Syntaxbaum kodiert wurde) - die vorhandenen Daten repräsentieren kann (z.B.: mit dem mittleren quadratischen Fehlerwert).

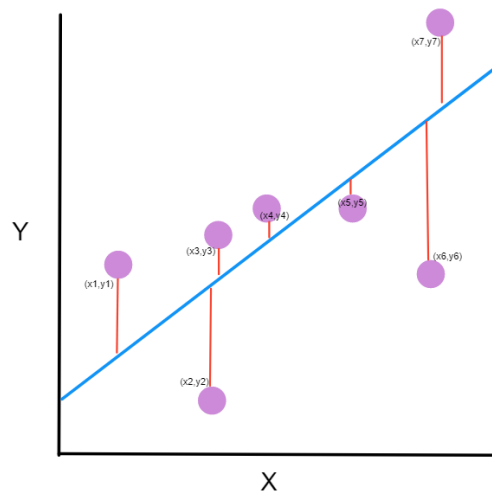


Abbildung 4: Mittlerer quadratischer Fehlerwert

3.0.1 Crossover

Beim "Crossover" wird der Syntaxbaum durch die Substitution seines Teilbaums durch einen Teilbaum eines anderen Syntaxbaums verändert.

Das folgende Bild erklärt die Implementierung der "CrossoverOperation" über Syntaxbäume:

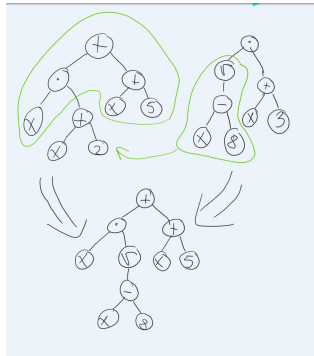


Abbildung 5: Crossover Funktionsweise

3.0.2 Mutation

Bei der Mutation wird es verschiedene Operationen geben:

- Ersetzen eines Knotens des Syntaxbaums durch einen Knoten, der einer anderen Rechenoperation entspricht
- Ersetzen eines Teilbaums durch einen zufällig erzeugten Teilbaum
- Entfernen eines Zwischenknotens aus dem Syntaxbaum
- Erweitern des Baums von der Wurzel aus
- Vertauschen von Teilbäumen bei nicht-kommutativen Operationen

Beispielhaft habe ich mal das Ersetzen eines Sub Baumes skizziert

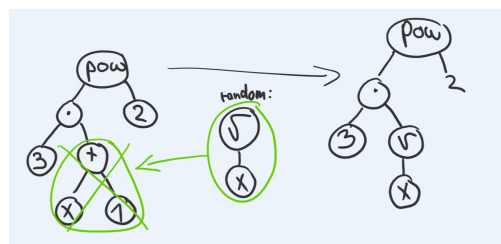


Abbildung 6: Crossover Funktionsweise

3.1 Implementation

Meine erste Wahl wäre Python gewesen, da ich in Python die Besten Kenntnisse besitze. Python bietet eine Menge interessanter Bibliotheken für genetische Algorithmen und dezente Plotting-Funktionen. Einige der beliebtesten Bibliotheken sind Pyvolution, deap, pySTEP, PyRobot, DRP und mehr. Diese Bibliotheken sind in der Lage, eine interaktive Grafik-Demo-Anwendung bereitzustellen. Ich habe aber auch die Idee es eventuell irgendwann als ReactJs Anwendung zur Verfügung zu stellen, da ich noch eine Domain besitze und es auch hosten könnte.

4 Informationen zum Nachschlagen

Symbolic regression via genetic programming, by D.A. Augusto; H.J.C. Barbosa : <https://ieeexplore-ieee-org.ezproxy.htwk-leipzig.de/stamp/stamp.jsp?tp=&arnumber=889734>

5 Alternativer Vorschlag - Kantenerkennung mit evolutionären Algorithmen

In dem Paper "Edge detection using evolutionary algorithms" by C.M. Ng; W.N. Leung; F. Chun wurde vorgestellt, wie man mithilfe evolutionärer Algorithmen Kanten erkennen kann. Eine grafische Oberfläche dazu zu bauen könnte in der Lehre eingesetzt werden und würde bestimmt Spaß machen. <https://ieeexplore.ieee.org/abstract/document/812522>

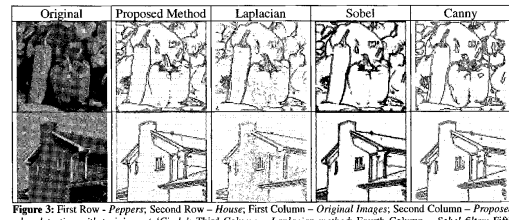


Abbildung 7: Kantenerkennung mit evolutionären Algorithmen

Sind diese Projektideen passend? Welche wäre Ihrer Erfahrung nach realistischer und besser geeignet als Projekt? Wenn keines der Beiden in Frage kommt, können sie mir eines zuweisen.