Symbolic Regression

Maarten Keijzer
Chordiant Software Inc.

Overview

- What is Symbolic Regression?
- The Machine Learning Perspective
  - maximum likelihood/maximum posterior
  - The role of priors
- Inductive Inference
  - universal priors
- Implications for GP

Symbolic Regression (naïve view)

- Given a set of input data $x$ and a set of desired outputs $t$, find a function $f$ such that:

\[ t = f(x_1, \ldots, x_n) \]

<table>
<thead>
<tr>
<th>$x_1$</th>
<th>$x_2$</th>
<th>$x_3$</th>
<th>$t$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.64</td>
<td>0.81</td>
<td>0.94</td>
<td>0.31</td>
</tr>
<tr>
<td>0.08</td>
<td>0.25</td>
<td>0.41</td>
<td>0.38</td>
</tr>
<tr>
<td>0.92</td>
<td>0.97</td>
<td>0.87</td>
<td>0.77</td>
</tr>
<tr>
<td>0.63</td>
<td>0.22</td>
<td>0.47</td>
<td>0.05</td>
</tr>
<tr>
<td>0.16</td>
<td>0.18</td>
<td>0.21</td>
<td>0.24</td>
</tr>
<tr>
<td>0.17</td>
<td>0.46</td>
<td>0.64</td>
<td>0.92</td>
</tr>
<tr>
<td>0.5</td>
<td>0.34</td>
<td>0.29</td>
<td>0.12</td>
</tr>
<tr>
<td>0.21</td>
<td>0.13</td>
<td>0.5</td>
<td>0.41</td>
</tr>
</tbody>
</table>

Symbolic Regression

- Find function structure (+ coefficients) using Genetic Programming

\[
(\text{ADD} (\text{MULT} \ x \ 0.341) \ y)
\]

\[
0.341 \ x + y
\]
**Process**

- Normal GP choices:
  - Some representation
    - Tree, linear, graph, ...
  - Some fitness function
    - Error based: MAE, MSE, or something else
  - The regular stuff
- Do something with constants

**Applications**

- Physics / Engineering
  - empirical equations/differential equations
- Econometrics
  - empirical relations
- Finance
  - trading rules
- Industry
  - process control/identification
- ...

**What do you get?**

- Automatic variable selection
- Explicit symbolic results
  - Interpretation (gray box?)
  - Acceptance by engineers
  - Ease of implementation for resulting expressions
- Freedom to implement non-continuous cost functions
- Multi-Objective search

**The END**

- Without a proper framework to discuss issues in Symbolic Regression, this is about what can be said.
- However, let's turn to ML + Statistics and see if there's more
Basic Statistical Theory on Regression

Formal description of relationship between probability, error measures, likelihood, posterior distributions and prior distributions

Likelihood (definition)

The Likelihood of our gp-function $f$ is the probability that we will have observed targets $t$, given our estimation of $f$.

$$p(t|f)$$

If there is no noise in the problem (i.e., measurements are perfect), the likelihood of $f$ is necessarily 1 iff $t = f(x)$, and zero otherwise.

In other words. If we know that $t$ is noise-free, the likeliness of observing the data given our function is exactly the number of hits.

However, there's no discrimination between near hits and non-hits

Likelihood (computation)

What's the likelihood of observing these points, given this function?

Need some way of assigning probabilities: the 'noise model'

Symbolic Regression (correct view)

- Given a set of input data $x$ and a set of desired outputs $t$, find a function $f$ such that:

$$t = f(x_1, \ldots, x_n) + \epsilon$$

<table>
<thead>
<tr>
<th>$x_1$</th>
<th>$x_2$</th>
<th>$x_3$</th>
<th>$t$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.64</td>
<td>0.01</td>
<td>0.94</td>
<td>0.31</td>
</tr>
<tr>
<td>0.08</td>
<td>0.05</td>
<td>0.41</td>
<td>0.38</td>
</tr>
<tr>
<td>0.02</td>
<td>0.97</td>
<td>0.87</td>
<td>0.77</td>
</tr>
<tr>
<td>0.63</td>
<td>0.22</td>
<td>0.47</td>
<td>0.05</td>
</tr>
<tr>
<td>0.16</td>
<td>0.18</td>
<td>0.21</td>
<td>0.24</td>
</tr>
<tr>
<td>0.17</td>
<td>0.46</td>
<td>0.64</td>
<td>0.92</td>
</tr>
<tr>
<td>0.5</td>
<td>0.34</td>
<td>0.29</td>
<td>0.12</td>
</tr>
<tr>
<td>0.21</td>
<td>0.13</td>
<td>0.5</td>
<td>0.41</td>
</tr>
</tbody>
</table>
The role of epsilon

\[ t = f(x) + \epsilon \]

Denotes the *noise* in the measurements \( t \)

\[ p(t \mid f) = p(f + \epsilon \mid f) = p(\epsilon \mid f) \]

Maximum likelihood function is the one with minimal residual error

Corollary: perfect fit of the 'sextic' polynomial is of very limited interest

Likelihood and Noise

If we have reason to assume that the noise is normally distributed (if one assumes variance is finite, this is the maximum entropy choice), our likelihood function will become:

\[ p(t \mid f) = \prod_i \frac{1}{\sqrt{2\pi}\sigma} \exp\left[ -\frac{(t_i - f(x_i))^2}{2\sigma^2} \right] \]

\( \text{noise was 0.1} \)
\( p(t \mid f(x)) = 4\% \)

Log-Likelihood / Squared Error

Maximizing Likelihood is equivalent with minimizing negative log-likelihood. After taking the logarithm and deleting constants, we end up with

\[ \arg\max_f p(t \mid f) = \arg\min_f \sum_i (t_i - f(x_i))^2 \]

Conclusion: assuming *noise* is distributed normally, leads to squared error minimization

Robust Error Measures

Sometimes you do have an idea of the nature of the noise. If you expect outliers, you might want to consider one of these robust measures

- absolute error \(|t - f(x)|\)
- double exponential distribution
- Lorentzian \( \log(1 + (t - f(x))^2) \) Lorentzian (Cauchy) distribution
- Pearson limit \( \log(\sqrt{(1 + (t - f(x))^2)}) \) --- Pearson limit distr.

All these measures translate into an assumption on the *noise*
Maximum Likelihood is limited

In general we are not interested in the probability of observing the data given the (correctness of the) function.

We want the probability (correctness) of the function given the data itself (Because we want to find the best function)

Maximizing this probability is called maximizing the posterior.

Maximize Posterior

maximizing posterior equals maximizing likelihood times prior

\[ \text{argmax}_f p(f|t) = \text{argmax}_f p(t|f)p(f) \]

The method of maximum likelihood assumes that the prior is uniform, i.e., all functions are equally likely

A uniform prior on an infinite space is ill-defined!

Bayes Rule

\[ p(f|t) = \frac{p(t|f)p(f)}{p(t)} \]

normalizer, generally unknown/uncomputable

A Prior for GP

Take the probability of generating the function at random as the prior. For instance under grow initialization 

\[ T = \{x\} \text{ and } F = \{+\} \]

generate \( x \) with probability 0.5

plus

with probability 0.5

plus

with probability 0.125

\[ p(f) = 2^{-|f|} \]
A Prior for GP

\[ p(f) = c^{-|\mathcal{F}|} \]

\( c \) depends on terminal/function set

Maximum Posterior for GP

\[
\arg\max_f p(f|t) = \arg\max_f p(t|f) p(f) \\
= \arg\max_f \prod_i \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(t-f(x_i))^2}{2\sigma^2}} c^{-|\mathcal{F}|} \\
= \arg\min_f \sum_i \frac{(t_i-f(x_i))^2}{2\sigma^2} + |\mathcal{F}| \log(c) \\
= \arg\min_f \sum_i (t_i-f(x_i))^2 + y |\mathcal{F}| \quad y = \log(c) \cdot \lambda \sigma^2
\]

Exponential Prior on size

\[
\arg\min_f \sum_i (t_i-f(x_i))^2 + y |\mathcal{F}|
\]

\( y = \log(c) \cdot \lambda \sigma^2 \)

Coding bias induced by primitive set

Intrinsic problem noise
(generally unknown, can also be more complex; per case uncertainty, weights)

Conclusions

- Noise assumption determines error function
  - Likelihood
- In GP maximum posterior not equal to maximum likelihood
  - Infinite space, assuming uniform prior is wrong
- Introducing a prior creates penalty function
  - Free parameter(s) value(s) inherently unknown
**Theory of Inductive Inference**

MDL & Universal priors (brief)

**Occam’s Razor**

- *Objects should not be multiplied beyond necessity*
  - What is necessity?
  - If any improvement in error is a *good thing*, Occam does not lead to penalty based parsimony pressure:
    - does lead to lexicographical parsimony pressure

  Occam’s razor as such does not provide justification for balancing size and error

**Minimum Description Length (Rissanen)**

- Minimize the total length in bits to transmit:
  - The model
  - The exceptions

$$\text{tree\_coding\_length} + \text{exception\_coding\_length}$$

$$\text{tree size} \quad \text{error}$$

*Problem: coding bias. True MDL is undecidable*

**What's tree coding length?**

- Our simple prior on size translates to a coding of $x$ bits per node
- Formally, we’re searching in the space of programs of variable length
- Every program can be described as a *prefix* (self-delimiting) sequence for a Universal Turing Machine
Solomonoff's Universal Prior

The universal prior probability of any prefix $p$ of a computable sequence $x$ is the sum of the probabilities of all programs (for a universal computer) that compute something starting with $p$

$$p(f) = 2^{-L(f)}$$

$L$ is the function that returns the length of the shortest program that can compute $f$. It is provably uncomputable.

Inductive inference and GP

- Our simple prior on size is a particular assumption about the universal coding function $L$
- when using the universal prior, the maximum posterior function becomes

$$\arg\min_f \sum_j (t_j - f(x_j))^2 + \gamma L(f)$$

In GP, we **always** have two components!!

Statement

Bloat is not an inherent problem in GP

Bloat is purely caused by ignoring program complexity in the objective function definition.

Solomonoff's theory of inductive inference shows that the prior is necessarily (a) complexity based, and (b) exponentially weighted
The true objective function for SR

The true objective function is given by:

\[ \arg\min_f \sum_j E(\epsilon_j) + L(f) \]

- **Error function** contains the assumptions on the noise.
- **Complexity function** contains our 'size'-based assumptions.

Penalty based optimization

The MDL-Fitness is defined as:

\[ \sum_j (t_j - f(x_j))^2 + \gamma |f| \]

- Particular form of MDL (only looking at size). One free parameter: \( \gamma \).
- Estimate using cross validation, then fix for entire set.
- Low initially, stronger later (Zhang et. al. 1995).
- Reports that penalty functions on size often work very well, yet leads to quite a few 'failed' runs. (Soule & Foster)

A practical objective function

A practical objective function is:

\[ \arg\min_f \sum_j E(\epsilon_j) + \gamma C(f) \]

- \( \gamma \) captures all constants involved in the tradeoff between size and goodness-of-fit.
- \( E \) is a simple error function.
- \( C \) is a practical function to get complexity information (e.g., size).

Multi Objective

Evolve a front of individuals that uniquely balance size and performance.
Pareto Dominance can be Overkill

Consider two expressions, with error $e_1$ and $e_2$ of size $s_1$ and $s_2$

- flip a coin, selecting $e_1,s_1$ with probability $p$, $e_2,s_2$ otherwise

Expected error of combination $p e_1 + (1-p) e_2$

Expected size of active expression $p s_1 + (1-p) s_2$

**Linear interpolation between members of front**

Convex hull optimization

- Because the 'true' objective function is additive, but with an unknown tradeoff the convex hull of the Pareto set contains the solution
Conclusions

- For symbolic regression we perform maximum posterior search
  - Priors cannot be ignored
- Maximum likelihood search is wrong, and leads to issues with bloat
- Penalty based search on 'complexity'/error' trade-off is difficult due to lack of knowledge
- Multi-objective search towards the convex hull in more promising