THE OHIO STATE UNIVERSITY

Genetic Programming:

exploring a new analysis tool to assist neutrino searches

Kaeli Hughes, with Professor Amy Connolly
Undergraduate: The Ohio State University
Graduate: The University of Chicago

Machine Learning Overview

A type of artificial intelligence that allows computers to solve complex problems

Many different types:

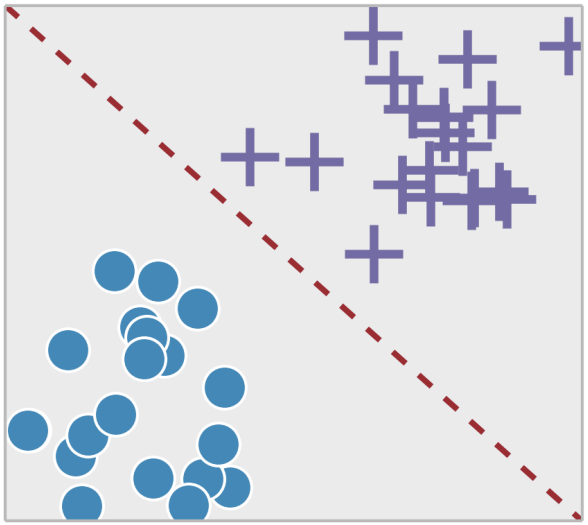
- Genetic programming
- Neural networks
- Boosted decision trees

Example: robotic vision

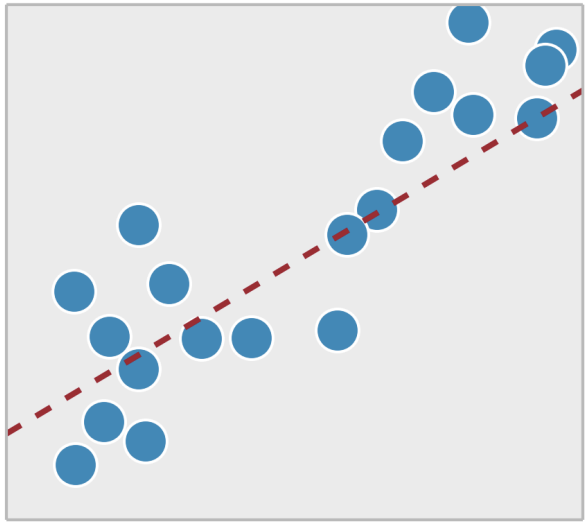
- Neural networks learned the difference between pictures with/without a white object (see right)

Two Main Types of Problems

Classification



Regression



Examples:

- Sorting dogs into breeds
- Recognizing handwritten text
- Sorting events into background/ signal
- Oindree will talk about this tomorrow

Examples:

- Determining both the form of a function and its coefficients
- Determining an unknown function to describe the relationship between a set of variables

Why this might be useful for ANITA

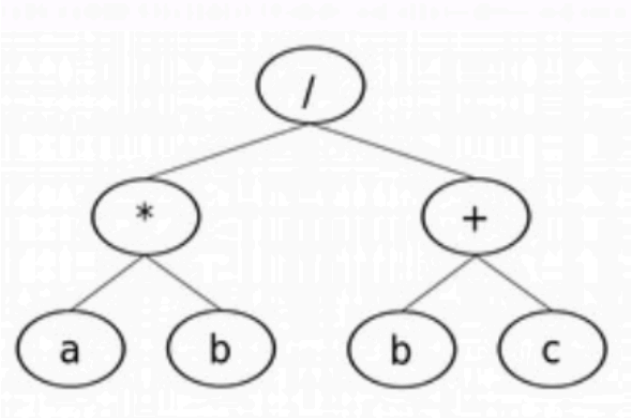
1. Anthropogenic noise is hard to model using standard techniques
   - Clustering with ABCD vs. exponential
   - Machine Learning may give us detailed model

2. Being able to model the background more precisely could allow optimizations of searches to improve

3. Currently we have tried 2 dimensional modeling (cross correlation and SNR), but in the future plan to try many dimensions

Genetic Programming

- Can solve both classification and regression problems
- Uses principles based on biological evolution to evolve solutions to problems
- GP algorithms create "populations" of functions called "trees" and then evolve them over multiple "generations"
- One program in particular: Karoo GP by Kai Staats



A function tree that Karoo GP creates, that represents the function (a*b)/(b +c). Credit: Kai Staats

Kai Staats:

- Masters in Applied Mathematics from University of Cape Town
- Documentarian
- Previous work includes SKA and LIGO

Karoo GP: How Functions Evolve

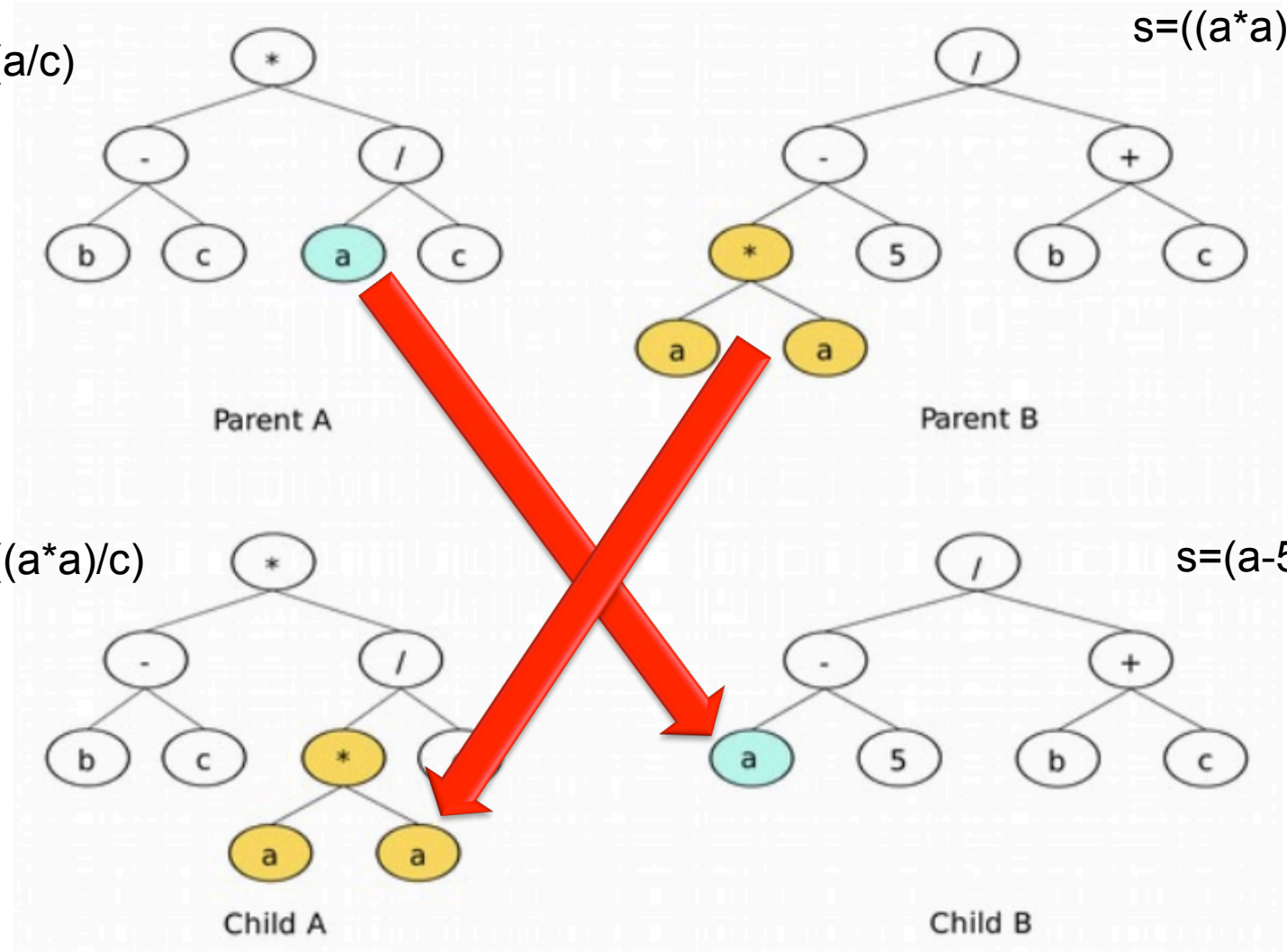
- After each generation, functions face off in “tournament selection”, in which the tree with the best fitness score continues on
- fitness score (regression):
  abs(expected value - model value)
- Functions that win the tournament selection go on to parent the next generation using a method shown

Method	Example Parent(s)	Example Child(ren)
Reproduction	(t*t)/(r+t)	(t*t)/(r+t)
Point Mutation	(t*t)/(r+**t**)	(t*t)/(r+**r**)
Branch Mutation	(t*t)/**(r+t)**	(t*t)/**(t/r)**
Crossover	Parent 1: (t*t)/**(r+t)** Parent 2: (r-t)***(t/r)**	Child 1: (t*t)/**(t/r)** Child 2: (r-t)***(r+t)**

Visual Example of Crossover

s=(b-c)*(a/c)

s=((a*a)-5)/(b+c)



Preparing the Data: Kepler's 3rd Law

Variables

Solution = t^2/r^3

r	t	F1 (=r^3)	F2 (=t^2)	0.1	2.7	3.1	0.5	s
0.241	0.39	0.0139	0.1521	0	0	0	0	0.98
0.615	0.72	0.2326	0.5184	0	0	0	0	1.01
1.00	1.00	1.00	1.00	0	0	0	0	1.00
1.88	1.52	6.6446	2.3104	0	0	0	0	1.01
11.8	5.2	1643.0	27.04	0	0	0	0	0.99
29.5	9.54	25672	91.012	0	0	0	0	1.00
84.0	19.18	5.9e5	367.87	0	0	0	0	1.00
165	30.06	4.4e6	903.60	0	0	0	0	1.00
248	39.44	1.5e7	1555.5	0	0	0	0	1.00

Features (Optional)

Constants (Optional)

How Karoo GP Works: Kepler's 3rd Law

1. The user provides a spreadsheet like the previous slide
2. Karoo GP asks user to select number of generations, number of trees, depth of tree, and minimum number of nodes (terms) per tree
3. That's it! Karoo runs and outputs the best answer in minutes

```
We have constructed a population of 100 Trees for Generation 1

Evaluate the first generation of Trees ...
        Tree 1 yields (sym): t**2
        Tree 2 yields (sym): r + t
        Tree 3 yields (sym): 3*r + t
        Tree 4 yields (sym): r + t
        Tree 5 yields (sym): r*t**2 - r*t + t + t/r
        Tree 6 yields (sym): r
        Tree 7 yields (sym): r**2*t - r*t + r + 1 + 2*t/r
        Tree 8 yields (sym): 1
        Tree 9 yields (sym): r**2/t - t**7 + t**2
        Tree 10 yields (sym): r*t
        Tree 11 yields (sym): 2*t
```

Example output to the screen of Karoo GP

How Karoo GP Works: Kepler's 3rd Law

```
Tree 81 yields (sym): r**(-2)
Tree 82 yields (sym): t/r
Tree 83 yields (sym): t**2/r**2
Tree 84 yields (sym): t**2/r**4
Tree 85 yields (sym): t/r**3
Tree 86 yields (sym): t/r
Tree 87 yields (sym): 1
Tree 88 yields (sym): t**2/r**4
Tree 89 yields (sym): t**2/r
Tree 90 yields (sym): t/r**4
Tree 91 yields (sym): t/r
Tree 92 yields (sym): t/r**3
Tree 93 yields (sym): t/r
Tree 94 yields (sym): t/r**3
Tree 95 yields (sym): r**(-3)
Tree 96 yields (sym): t
Tree 97 yields (sym): r**(-3)
Tree 98 yields (sym): r
Tree 99 yields (sym): t**2/r**3
Tree 100 yields (sym): t/r**2

  23 trees [ 1  3  6  7  8 12 15 21 30 33 34 39 41 42 49 52 59 60
 61 63 64 78 99] offer the highest fitness scores.

 Copy gp.population_b to gp.population_a


  "It is not the strongest of the species that survive, nor the mos
t intelligent,
   but the one most responsive to change." --Charles Darwin

 Congrats! Your multi-generational Karoo GP run is complete.
```

Generation	Karoo's best guess for answer
1	s=1
2	s=1
3	s=1
4	s=1
5	$s=t^2/r^3$
6	$s=t^2/r^3$
7	$s=t^2/r^3$
8	$s=t^2/r^3$
9	$s=t^2/r^3$
10	$s=t^2/r^3$

Comparing Other GP Algorithms

Karoo GP:
- Open Source
- Can customize fitness function
- Newly developed

Eureqa:
- Commercial
- Very well developed
- Cannot customize fitness function

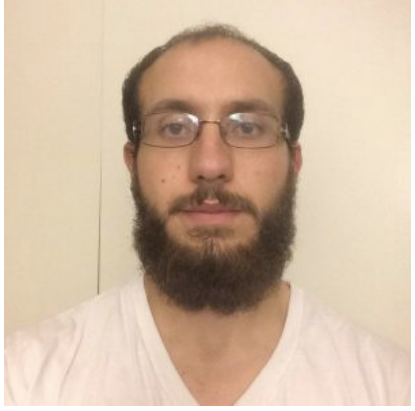
HeuristicLab:
- Open Source
- Only runs on Windows (limited Linux functionality)
- Unstable (crashes easily)

Karoo GP

Eureqa®

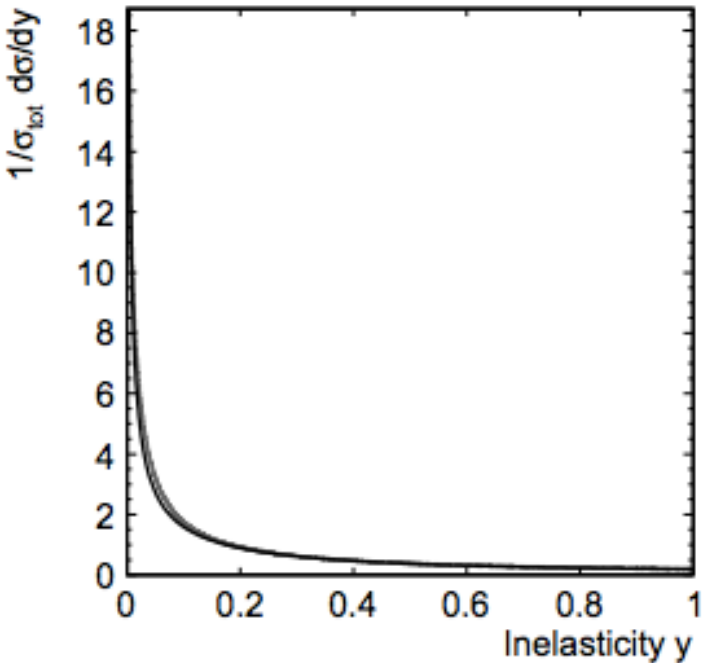
HL HeuristicLab
A Paradigm-Independent and Extensible Environment for Heuristic Optimization

Abdullah, an undergraduate student at OSU, has been doing lots of research into best GP techniques

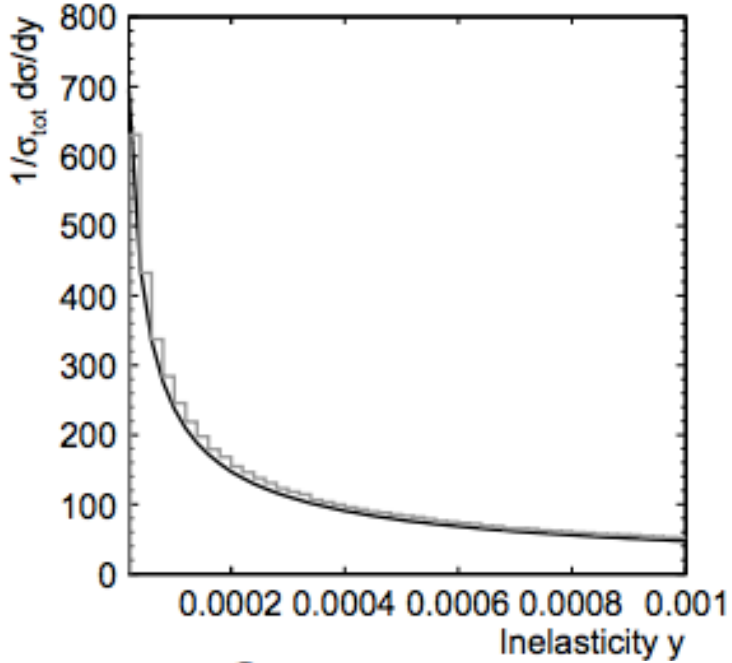


Spoorthi, also an undergraduate at OSU, helped eliminate some of the other options (notably TMVA)

Proof of Concept: Inelasticity



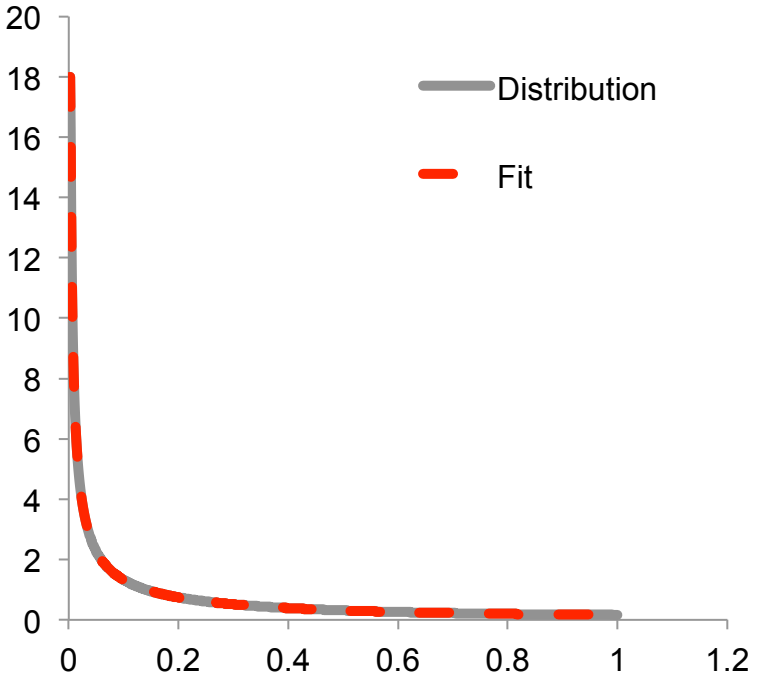
$$y_0 = \frac{(y_{\max} - C_1')^R}{(y_{\min} - C_1')^{R-1}} + C_1'$$



$$y_0 = C_1 + \Big[R(y_{\max} - C_1)^{(-1/C_2+1)} + \cdot (1-R)(y_{\min} - C_1)^{(-1/C_2+1)}\Big]^{C_2/(C_2-1)}$$

- Plots show theoretical vN CC inelasticity distributions (in black) and parameterized fit (grey). From (Connolly, 2011)
- Parameterization split into two regions

Proof of Concept: Inelasticity



- Eureqa: almost exact fit for y>0.0001
- R^2= 0.99999 (1 is a perfect fit)
- Caveat: have not yet included steep low y values yet

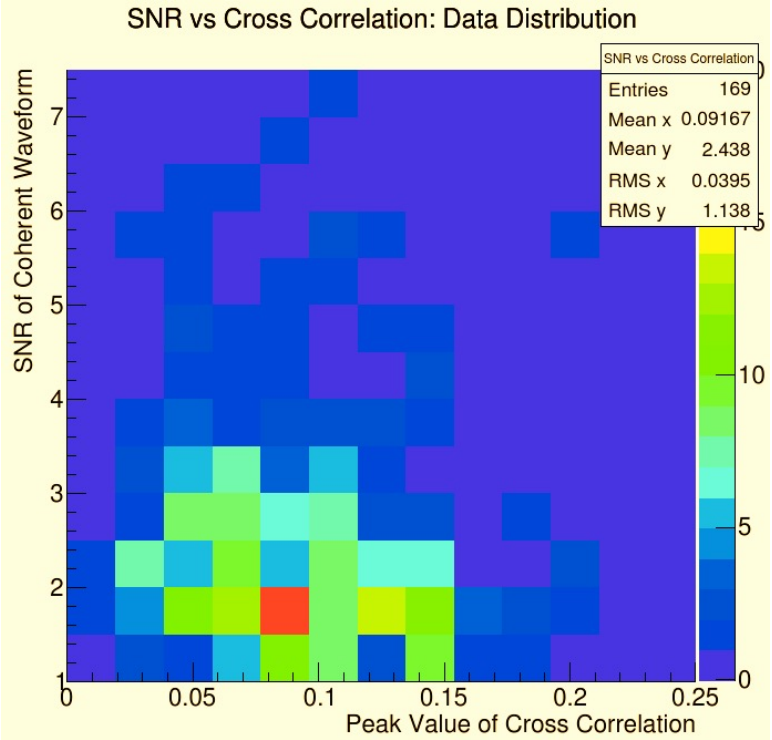
$$\frac{1}{x\sqrt{\tan^{-1}\left(x^{3/2}\right)}}0.0249826\left(-8.3929\,x\sqrt{\tan^{-1}\left(x^{3/2}\right)}+4.13317\sqrt{x}\sqrt{\tan^{-1}\left(x^{3/2}\right)}-0.0835953\sqrt{\tan^{-1}\left(x^{3/2}\right)}+x^3\sqrt{\tan^{-1}\left(x^{3/2}\right)}+8.73838\,x\right)$$

Using Karoo GP to model background

- Building off work done in Brian Dailey's thesis work on ANITA-2
- Rather than assume exponential, let GP evolve a form
- Data was binned so that Karoo GP could find a function with the form:

$$f(x,y) = \frac{dN}{dxdy}$$

- "dN": number of background events per bin
- "dx": bin width along x-axis (cross correlation)
- "dy": bin height along y-axis (SNR)

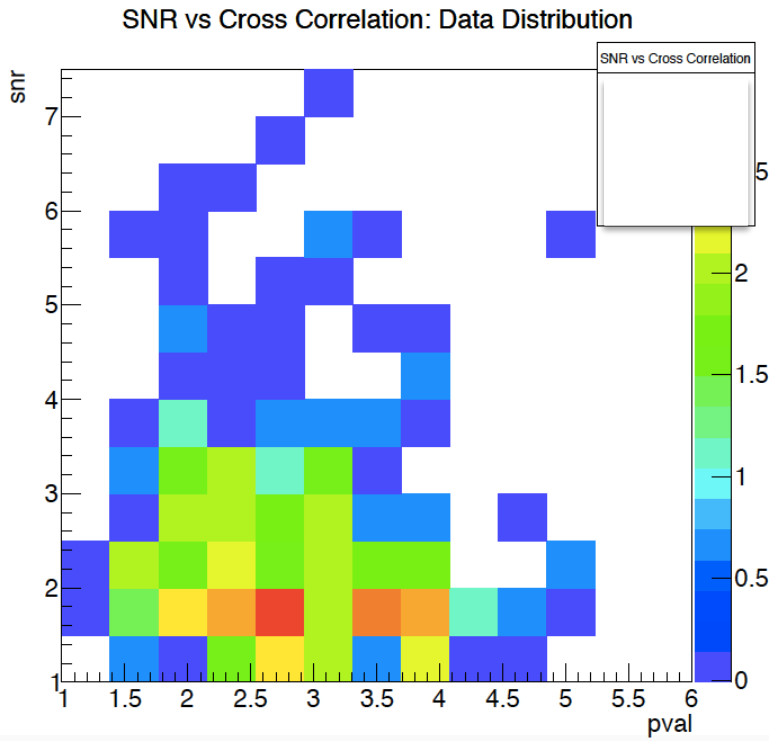


Binned data from singular Antarctic bin

Preparing the data for Karoo

1. Increase Cross Correlation range:
   - Two variables should be on the same scale
   - Therefore, range was increased from 0-0.25 to 0-5.5 (similar to SNR)

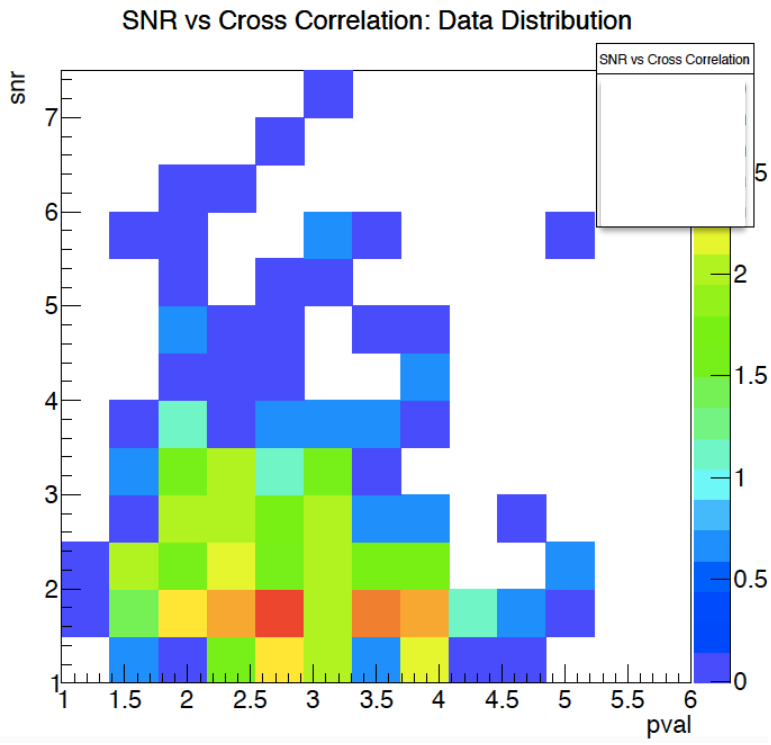
2. Take the logarithm of the number of events
   - Data naturally has sharp peak which is hard for Karoo GP to guess
   - Taking logarithm makes the peak less steep



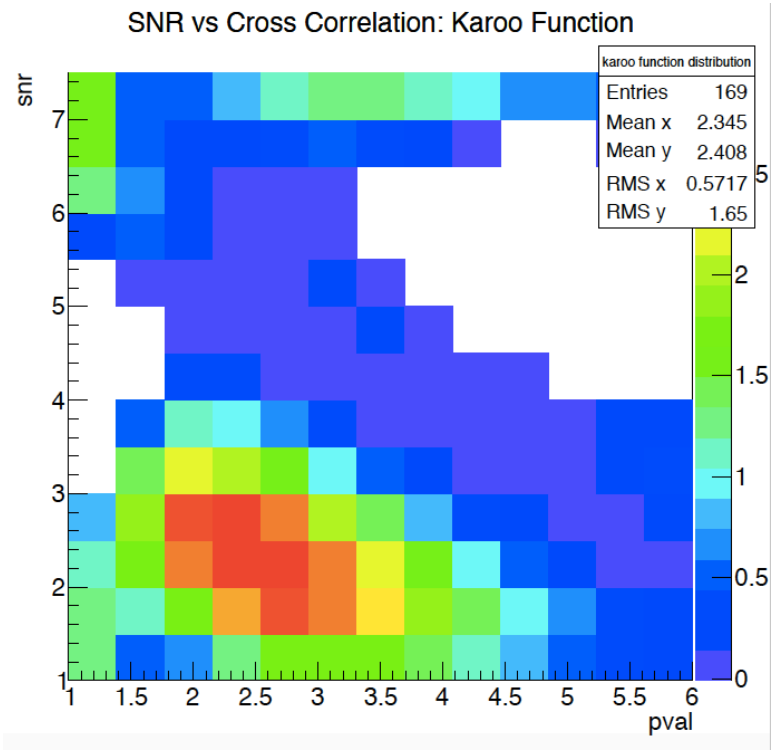
Data distribution after taking the logarithm of the bin number and increasing the cross correlation range

Finding a Best Fit Function:

Binned Background Data

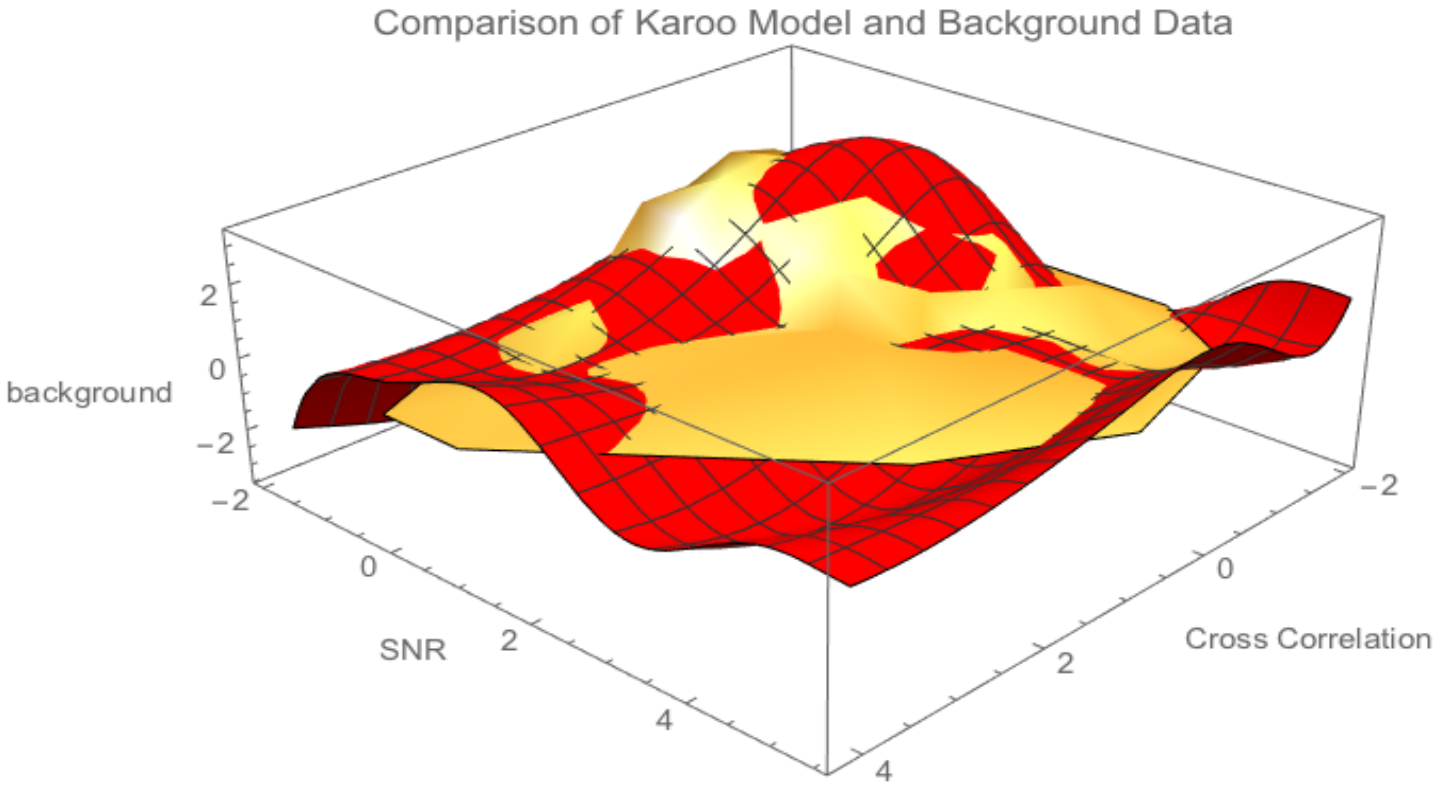


Karoo Function to describe data



Finding a Best Fit Function:

$$f_2(x,y) = \log\frac{dN}{dxdy} = 0.29 * \sin(x+3) - \cos(y+3) * \cos(x+y) - 0.819 * \cos(y+3) + 0.40 - 0.71 * e^{-y-3} * \cos(x+3) - 0.88 * e^{-y} * e^{-y-3} - e^{-y-3} * \cos(x+3) * \cos(y+3) - 2.0 * e^{-x-3} * \cos(x+3) - 0.59 * e^{-x-3} - 0.61 * e^{-x} * \cos(x+3) * \cos(x+y)$$



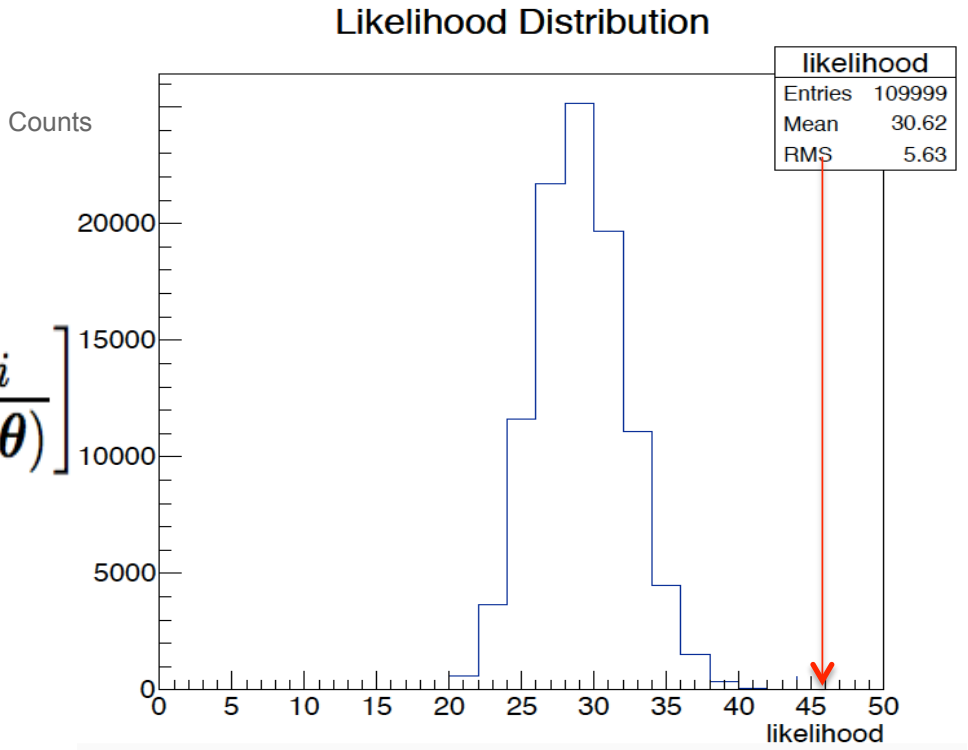
Three dimensional plot showing the data (yellow) and the Karoo GP model (red)

Calculating Likelihood

- Generated pseudo-experiments that did model data and calculate likelihood based on the following equation:

$$-2\ln\lambda(\boldsymbol{\theta}) = 2\sum_{i=1}^{N}\left[\mu_i(\boldsymbol{\theta}) - n_i + n_i \ln\frac{n_i}{\mu_i(\boldsymbol{\theta})}\right]$$

- "n_i": value from pseudo-data
- "μ_i": value from Karoo GP model
- Determined that more models must be tested before continuing with analysis

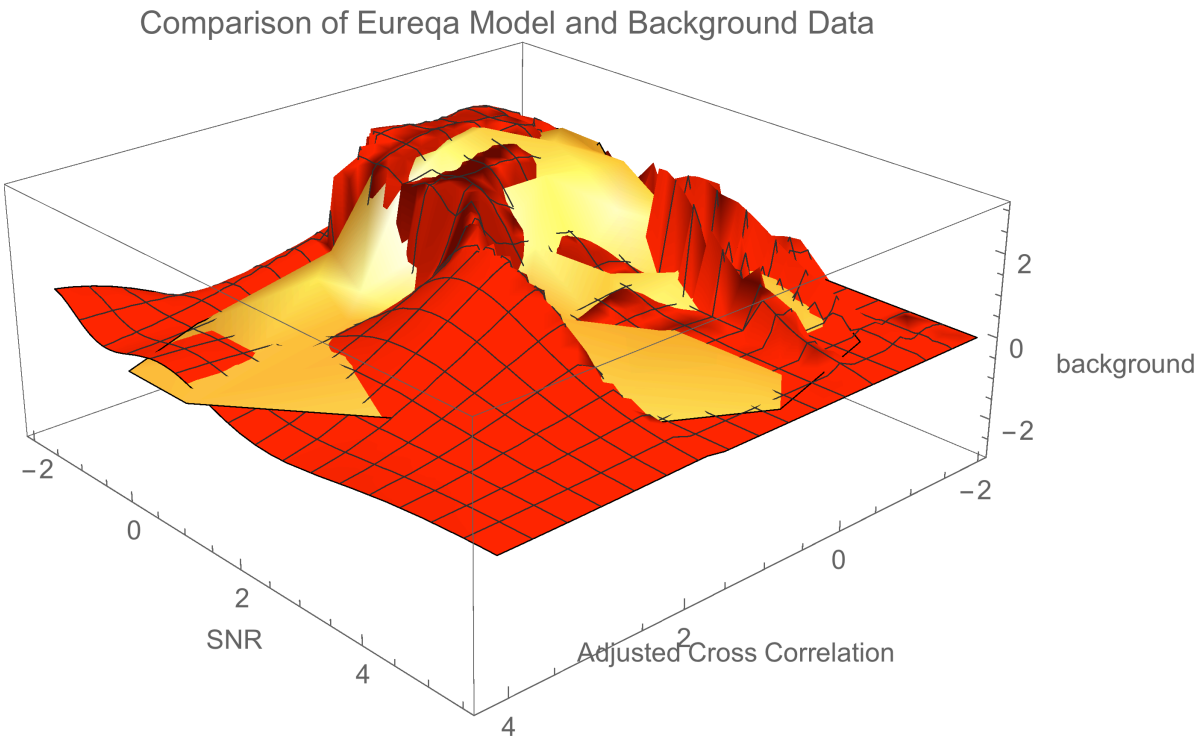


Likelihood Distribution of pseudo-experiments (blue) and data (red) each compared to the Karoo GP model

How do other GP algorithms perform?

Eureqa:

- Right general shape
- Not a clear peak
- $R^2 = 0.92$
- Still need to run psuedo- experiments to fully measure performance

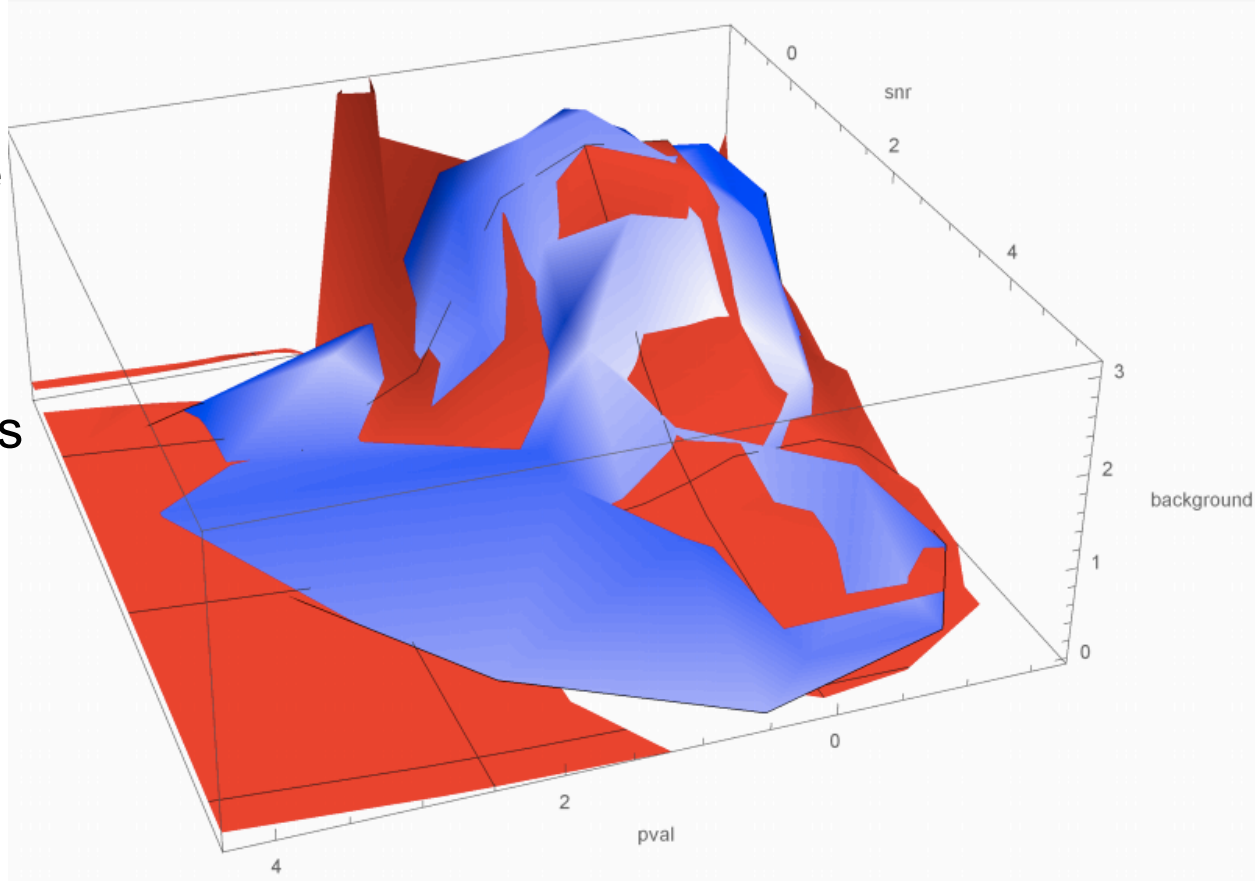


Yellow: data
Red: function

How do other GP algorithms perform?

HeuristicLab:

- Right general shape
- Clearer peak
- $R^2 = 0.94$
- Still need to run psuedo- experiments to fully measure performance



Blue: data
Red: function

What will we do after finding a good background model?

1. Design the best optimization cut based on background model from GP
   - Would no longer have to be exponential

2. Use that result to complete analysis

3. Continue modeling background using many variables (not just SNR/cross correlation)

4. Begin using GP algorithms on other multivariable problems (payload blasts vs non payload blasts, for example)