THE UNIVERSITY
of NORTH CAROLINA
at CHAPEL HILL

A New Suite of Statistical Algorithms for Bayesian Model Fitting with Both Intrinsic and Extrinsic Uncertainties in Two Dimensions

Nicholas C. Konz

Senior Honors Thesis Defense

Department of Physics and Astronomy, UNC Chapel Hill

April 2020

The Goals of This Project

The TRK statistic, created by Trotter, 2011, is used to fit models to data with both intrinsic uncertainty (error bars) and extrinsic uncertainty (sample variance), both in two dimensions; the **worst case uncertainty** scenario.

Trotter implemented the TRK statistic as “science code”, in the form of a genetic algorithm.

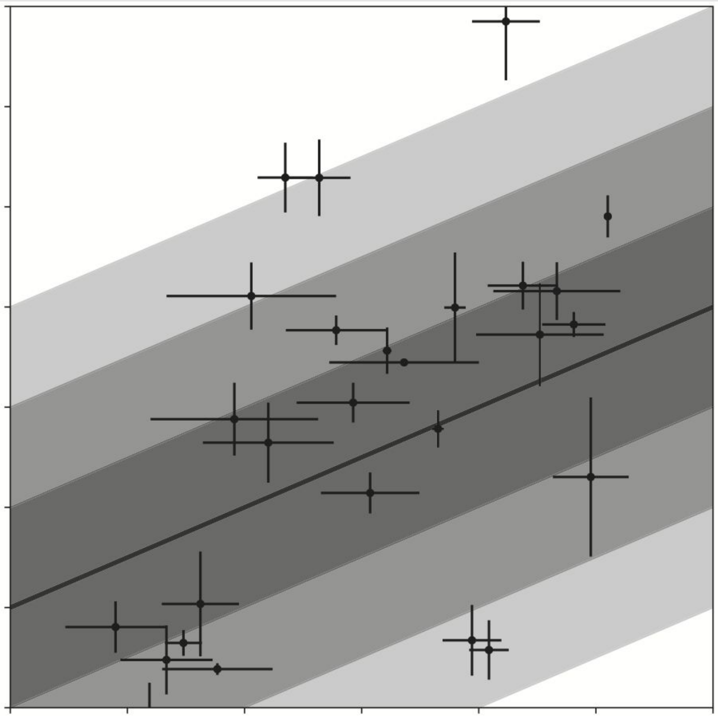
My job was to develop a production-quality (C++) code library for TRK from scratch: the “TRK Suite”, which:

1. Expands the original functionality of the TRK statistic, making it heavily-tested and applicable/generalizable to a wide range of problems, while also being:
2. Robust, but easy to use, including
    a. A user-friendly, well-documented and self-contained codebase, and
    b. A website for the statistic—including a web-based calculator—and code repository

Part I: The TRK Statistic

Setting the Stage

- N datapoints $\{x_n, y_n\}$ with intrinsic distributions defined by error bars $\{\sigma_{x,n}, \sigma_{y,n}\}$, assumed Gaussian.
- The dataset may also have (2D) *extrinsic* scatter, or *slop*, that cannot fully be accounted for by the error bars; also assumed Gaussian.
- Slop is defined by the parameters (σ_x, σ_y), and must be fit to alongside the model parameters $\{\vartheta_m\}$.

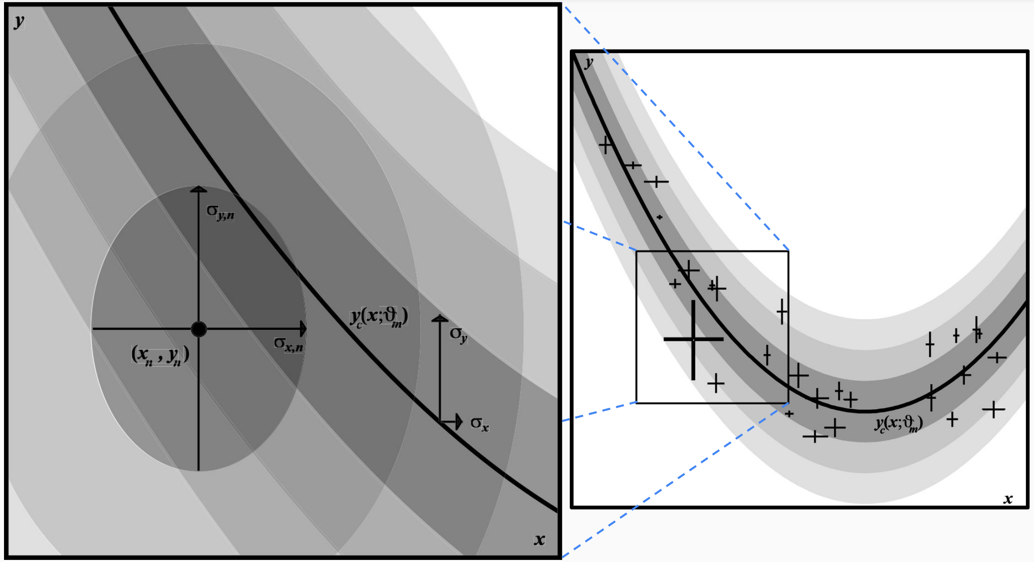


Pictured: a dataset and accompanying linear model distribution where the scatter of the data cannot solely be accounted for by the error bars, which must be parameterized as slop.

Defining the Model

- How do we define (the data-independent part of) a model distribution given this scenario?
    - Convolve some desntiy distribution $g(x,y)$ along some model *curve* $y_c(x;\vartheta_m)$ and then convolve *this* with a 2D Gaussian describing the slop.

- We can also represent y_c as a *delta function* along some arbitrary coordinate system(u_n, v_n), **that may differ between points.**



Pictured: some 2D dataset with accompanying model distribution (left), and then zoomed into the inset box (right).

Defining the Likelihood Function I

- Review: for a single n^{th} datapoint, we need to consider:
  a. The intrinsic uncertainty distribution (error bars), defined by $(\sigma_{x,n}, \sigma_{y,n})$.
  b. The extrinsic uncertainty distribution (slop), defined by (σ_x, σ_y).
  c. The “data-independent” portion of the model distribution: some distribution $g(x, y)$ convolved along some model curve $y_c(x; \vartheta_m)$.
     - Note that we can write this as $g(x,y)\delta(v_n, v_{c,n}(u_n; \vartheta_m))$, where (u_n, v_n) is the arbitrary coordinate system used for this datapoint.

Putting all of this together, the joint probability for a *single datapoint* (is then

$$p_n(\vartheta_m, \sigma_x, \sigma_y | x_n, y_n, \sigma_{x,n}, \sigma_{y,n}) = \int_{x'} \int_{y'} \int_{u_n} \int_{v_n} g(x,y)\delta(v_n - v_{c,n}(u_n; \vartheta_m)) \times$$
$$\mathcal{N}(x'|x, \sigma_x)\mathcal{N}(y'|y, \sigma_y)\mathcal{N}(x'|x_n, \sigma_{x,n})\mathcal{N}(y'|y_n, \sigma_{y,n}) \mathrm{d}v_n \mathrm{d}u_n \mathrm{d}y' \mathrm{d}x'$$

* $\mathcal{N}(x|\mu, \sigma)$ *denotes a Gaussian distribution*

Defining the Likelihood Function II

- Given that the likelihood is the product of all N of the datapoints' joint probabilities p_n,

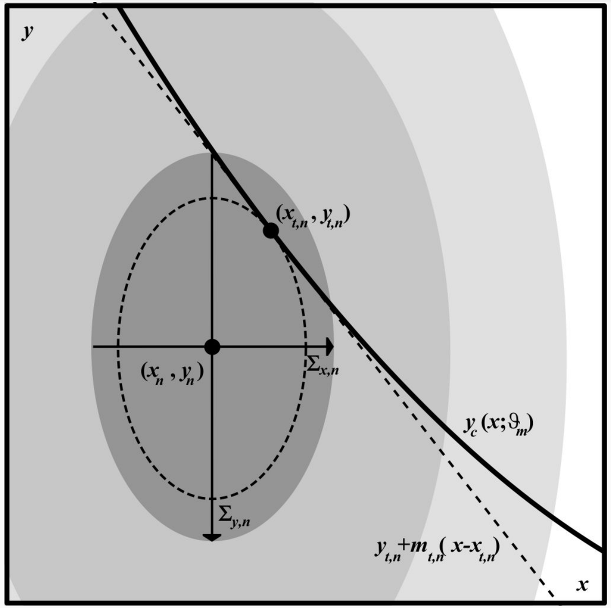
$$\mathcal{L} = \prod_{n=1}^{N} p_n(\vartheta_m, \sigma_x, \sigma_y | x_n, y_n, \sigma_{x,n}, \sigma_{y,n}),$$

this four-integral solution is clearly computational intractable.

As such, we will need to make some simplifying, but reasonable assumptions.

Making a Few Simplifications

1. By defining $\Sigma_{x,n} \equiv \sqrt{\sigma_{x,n}^2 + \sigma_x^2}$ and $\Sigma_{y,n}$ similarly, p_n can be written as the integral of the effectively 1D distribution along the model curve, convolved through a 2D Gaussian "error ellipse" with widths defined by $(\Sigma_{x,n}, \Sigma_{y,n})$ (see right).
2. Assume $g(x, y)$ is ~constant on the scale of the error ellipse, so it can be factored out of the integral.
3. Assume the model curve y_c is approximately linear at this same scale, with the line going through the point $(x_{t,n}, y_{t,n})$ where y_c is tangent to the error ellipse (see right), i.e. $y_c(x) \approx y_{t,n} + m_{t,n}(x - x_{t,n})$.



Pictured: visualization of the approximations 1 and 3, for a single datapoint.

The TRK Likelihood

- Given these assumptions, we can evaluate the integrals of p_n over u_n and v_n, to obtain

$$p_n(\vartheta_m, \sigma_x, \sigma_y | x_n, y_n, \sigma_{x,n}, \sigma_{y,n}) \approx g(x_n, y_n) \frac{\mathrm{d}u_n}{\mathrm{d}x} \mathcal{N}\left(y_n \,|\, y_{t,n} + m_{t,n}(x_n - x_{t,n}), \sqrt{m_{t,n}^2 \Sigma_{x,n}^2 + \Sigma_{y,n}^2}\right).$$

- As the likelihood $\mathcal{L} = \prod_{n=1}^{N} p_n$ is maximized to determine a best fit, the choice of arbitrary coordinate u_n will define the statistic.
- Different choices of u_n will create different statistics with varying properties. For some n^{th} datapoint, the TRK statistic takes u_n to be perpendicular to the line segment connecting the datapoint (x_n, y_n) with the tangent point $(x_{t,n}, y_{t,n})$ (see next slide). This gives a likelihood of

$$\mathcal{L}^{\mathrm{TRK}} \propto \prod_{n=1}^{N} \sqrt{\frac{m_{t,n}^2 \Sigma_{x,n}^2 + \Sigma_{y,n}^2}{m_{t,n}^2 \Sigma_{x,n}^4 + \Sigma_{y,n}^4}} \exp\left\{-\frac{1}{2} \frac{\left[y_n - y_{t,n} - m_{t,n}(x_n - x_{t,n})\right]^2}{m_{t,n}^2 \Sigma_{x,n}^2 + \Sigma_{y,n}^2}\right\}.$$

* *where $m_{t,n}$ is the slope of the linear approximation of the model curve through the error ellipse tangent point.*

A Visualization of the TRK statistic

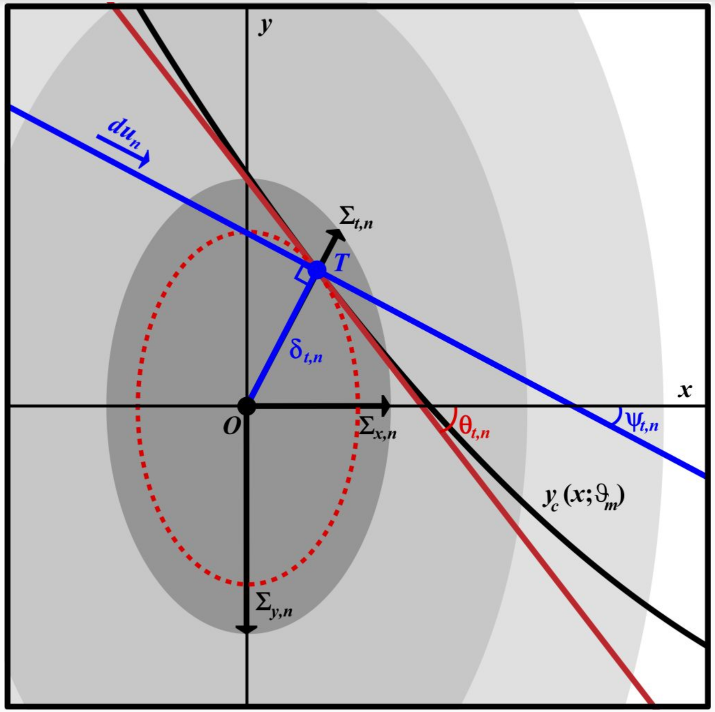
Pictured:

- Some datapoint (x_n, y_n)(point O) with error ellipse described by widths $(\Sigma_{x,n}, \Sigma_{y,n})$.
- The model curve y_c is tangent to the error ellipse at $(x_{t,n}, y_{t,n})$ (point T), and the red line is the linear approximation of y_c.
- The blue line indicated the rotated coordinate u_n for the TRK statistic.

- Note that following the figure, the TRK likelihood can be rewritten as

$$\mathcal{L}^{\mathrm{TRK}} \propto \prod_{n=1}^{N} \frac{1}{\Sigma_{t,n}} \exp\left(-\frac{1}{2}\frac{\delta_{t,n}^2}{\Sigma_{t,n}^2}\right).$$

So, this is really a χ^2-like statistic measured in the direction of closest approach of the model curve to the data point!



Part II: Properties of The TRK Statistic

Invertibility and Scalability

- Invertibility: whether fitting y vs. x gives same fit/likelihood as x vs y. **TRK is completely invertible.**
- Scalability: whether re-scaling the data (e.g. changing units) along one of the axes, i.e. scaling the y-axis by some factor s, does not alter the best fit.
  - **TRK is *not* scalable;** how do we get around this?
  - Nothing can be determined about the relative fitness between best fits at different scales, solely by comparing likelihoods, so s cannot be fit to as a model parameter.
  - However, there is still a way to compare fits performed on different scales...

The TRK Correlation Coefficient

- We want to compare two fits done at different scales $s = a$ and $s = b$ ($a < b$).
- Define the *scale-dependent TRK correlation coefficient* to be

$$R^2_{\mathrm{TRK}}(a,b) \equiv \tan^2\left(\frac{\pi}{4} - \frac{|\theta_a - \theta_b|}{2}\right) \text{ (linear model case)} \longrightarrow \qquad R^2_{\mathrm{TRK}}(a,b) \equiv \frac{1}{N}\sum_{n=1}^{N}\tan^2\left(\frac{\pi}{4} - \frac{|\theta_{t,n;a} - \theta_{t,n;b}|}{2}\right) \text{ (general case),}$$

For the linear case, θ_a is the slope angle of the best fit line at a, and θ_b is defined similarly. For the nonlinear, general case, $\theta_{t,n;a}$ is the angle of the derivative of the best-fit curve at the tangent point to the n^{th} datapoint at a. ***Note that these fits, although performed at different scales, have their angles compared within the original, s=1 space.***

- Therefore, for the linear case, if the two lines have the same slope, $R^2_{\mathrm{TRK}} = 1$, and if they are orthogonal, $R^2_{\mathrm{TRK}} = 0$.
- But, how does this help us determine an *optimum fitting scale*?

Scale Optimization

- Consider the slop parameters (σ_x, σ_y). Trotter (2011) showed that there exists scales a, b with $a < b$ such that

$$\lim_{s \to a^+} \sigma_x = 0 \text{ and } \lim_{s \to b^-} \sigma_y = 0.$$

- We consider the *physically meaningful* range of fits to correspond to $s \in [a, b] \subset [0, \infty)$, as best fits outside of this range require imaginary slop.
- Therefore, there must exist some optimum fitting scale $s_0 \in [a, b]$.
- The process used to actually determine a, b and s_0 in practice will be covered shortly.

Part III: Core Algorithms of the TRK Statistic

THE UNIVERSITY
of NORTH CAROLINA
at CHAPEL HILL

THE UNIVERSITY
of NORTH CAROLINA
at CHAPEL HILL

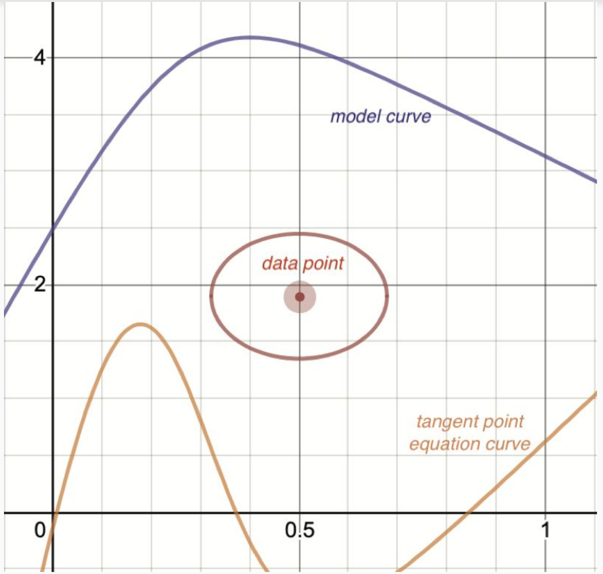
Finding Tangent Points I

- Recall that the for each of N datapoints, the TRK likelihood includes terms from the linear approximation of the model curve y_c through the point $(x_{t,n}, y_{t,n})$ where y_c is tangent to the n^{th} datapoint's error ellipse described by $(\Sigma_{x,n}, \Sigma_{y,n})$.
- But, how can we determine these $(x_{t,n}, y_{t,n})$?
- The condition describing such a tangent point can be written as

$$(y_c(x) - y_n)\frac{\mathrm{d}y_c(x;\vartheta_m)}{\mathrm{d}x}\Sigma_{x,n}^2 + (x - x_n)\Sigma_{y,n}^2 = 0,$$

which can be solved implicitly to obtain $x_{t,n}$ and $y_{t,n} = y_c(x_{t,n})$.

- In practice, the Two-Point Newton-Raphson rootfinder of Tiruneh et. al, 2013 to solve this equation.
- However, as shown to the right, a datapoint can have multiple tangent points…how do we find them all?

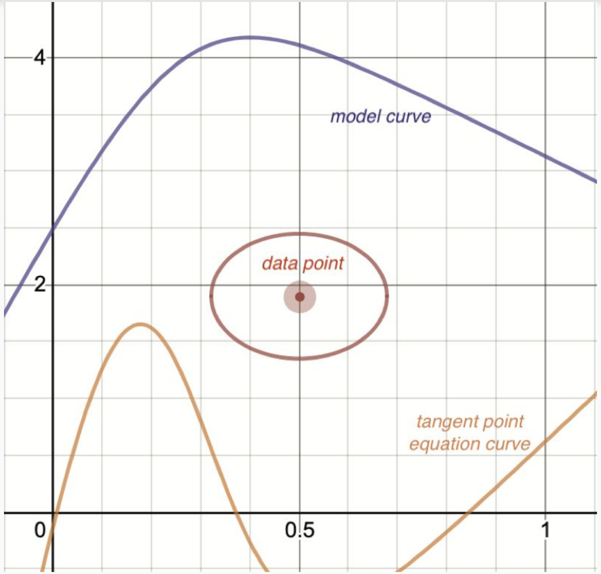


Pictured: example of a model curve and datapoint with error ellipse where there are multiple tangent points, found by solving the equation in orange.

Finding Tangent Points II

- Certain non-monotonic models can frequently have multiple tangent points for a single datapoint.
- Given such a scenario, we take the tangent point that maximizes the joint posterior probability p_n of that datapoint.

- The root finder will always converge to the same tangent point given some initial guess; so, in order to find all tangent points, we created an algorithm to supply various guesses to the root finder until all tangent points are found.
- The core of this algorithm is a quadratic approximation of the tangent point equation through the initially found tangent point, that is used to estimate guesses for additional points, if possible.

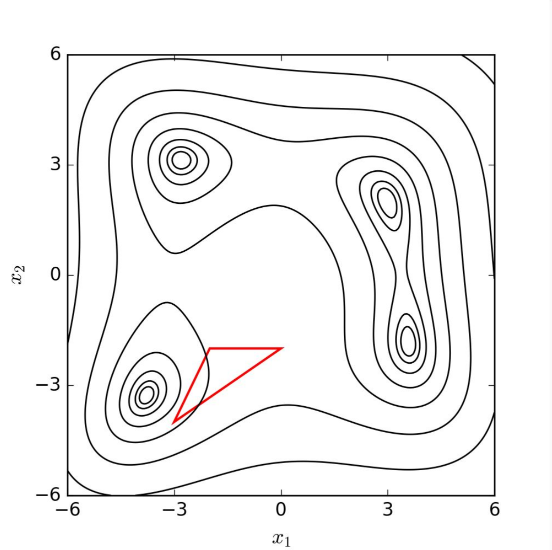


Pictured: example of a model curve and datapoint with error ellipse where there are multiple tangent points, found by solving the equation in orange.

Likelihood Maximization

- In practice, the Downhill Simplex method of Nelder and Mead, 1965 is used to maximize the TRK likelihood, due to:
    a. it not requiring derivatives of the likelihood with respect to the model *and* slop parameters, and
    b. the ability to implement bounded priors on the parameters, in the form of “hard walls” in parameter space.

- Simplex: higher-dimensional generalization of a triangle
    - Evolves organically through parameter space along the hypersurface of the likelihood
    - Uses likelihood evaluated at its vertices to find where likelihood is maximized

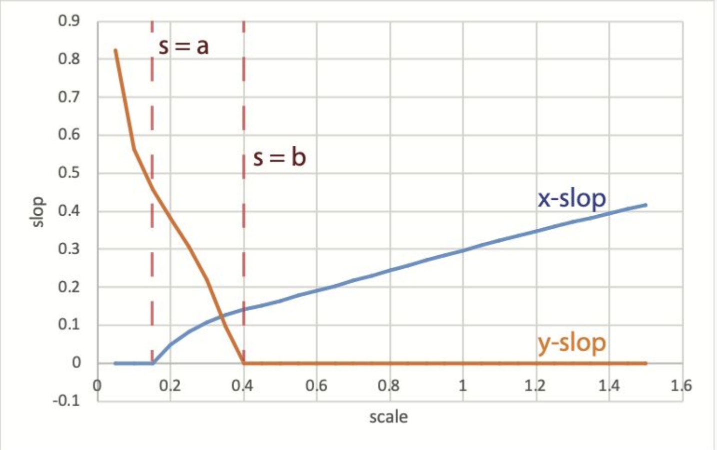


https://en.wikipedia.org/wiki/Nelder%E2%80%93Mead_method#/media/File:Nelder-Mead_Himmelblau.gif

Pictured: example animation of the Downhill-Simplex method

Implementing Scale Optimization I

- Recall that as the TRK statistic is *not* scalable, we must determine the optimum scale s_0 at which to run fits/maximize the likelihood, given a dataset and model.
- To do so, we must determine the minimum scale a and maximum scale b at which the x-slop σ_x goes to zero and the y-slop σ_y goes to zero, respectively.



Pictured: example of the dependence of the best fit slop parameters on the fitting scale s.

- I've found that in practice, (see right), these relationships have shown to be monotonic.
- As such, a type of bracketing/bisection method is used to determine a and b, by running fits to determine best fit slop parameters at various scales, until the exact scales at which these slops go to zero are found.

Implementing Scale Optimization II

- To determine s_0, Trotter defined an iterative approach, with the first approximation $s_0^{(1)}$ found with

$$R^2_{\mathrm{TRK}}(a, s_0^{(1)}) = R^2_{\mathrm{TRK}}(s_0^{(1)}, b).$$

In practice, this equation is solved using a bisection method.

- Further iterations are found by
  a. shifting from $s = 1$ space to $s = s_0^{(1)}$ space when solving the above equation, such that the angles θ within the expression follow the transformation $\theta \to \arctan\left(s_0^{(1)} \tan\theta\right)$, and then
  b. repeating the above equation until the successive $s_0^{(1)}$ converge to s_0.

Part IV: Parameter Uncertainty Computation

THE UNIVERSITY *of* NORTH CAROLINA *at* CHAPEL HILL

Bayes' Theorem

- In order to compute uncertainties of model parameters $\{\vartheta_m\}$and slop parameters (σ_x, σ_y)most accurately for some M-parameter model, we need to compute the *posterior* probability distribution $P(\Theta|D)$ of the model parameters, *(where Θ is the total $M + 2$ parameters and D is the dataset)*
- Given Bayes' Theorem, we have that

$$P(\Theta|D) \propto \mathcal{L}^{\mathrm{TRK}}(D|\Theta)p(\Theta),$$

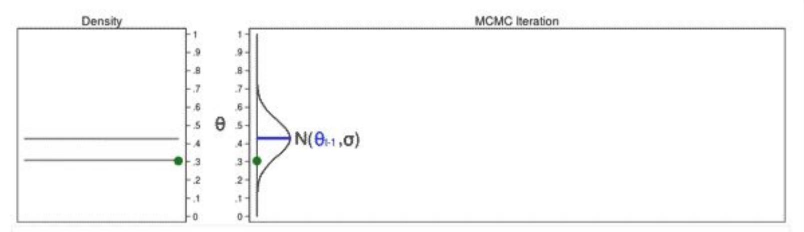
given any priors $p(\Theta)$.

- So, we can compute the value of the posterior at some set of parameters given the likelihood and any priors.
- However, to calculate uncertainties, we need to characterize the posterior distribution *as a whole*, in order to determine its variance. In practice, we can do this by *sampling*, or generating a histogram of, $P(\Theta|D)$.
- This ushers in Monte Carlo methods...

THE UNIVERSITY of NORTH CAROLINA at CHAPEL HILL

The Adaptive Metropolis-Hastings Method

- In practice, we sample the posterior $P(\Theta|D)$ with the *adaptive* Metropolis-Hastings Markov Chain Monte Carlo (MCMC) method (Haario et. al, 2001).
- This algorithm utilizes the ability of computers to rapidly generate random numbers, by iteratively generating R samples from $P(\Theta|D)$.
- The "*Markov Chain*" describes how given some i^{th} sample Θ_i, the *proposal* distribution Q of the next potential sample Θ_t depends on Θ_i, i.e. $Q(\Theta_t|\Theta_i)$.
- The *adaptive* part describes how Q self-updates over time, as the overall sampling is being generating.
- Essentially, the algorithm works by
    a. Given some previous Θ_i, sample $\Theta_t \sim Q$.
    b. Accept Θ_t (set $\Theta_{i+1} = \Theta_t$) with probability $\min[1, P(\Theta_t)/P(\Theta_i)]$.
    c. If accepted, update location/mean μ and size/covariance matrix Σ of Q given quality of overall sample.
    d. Even if not accepted, add Θ_t to overall sampling.



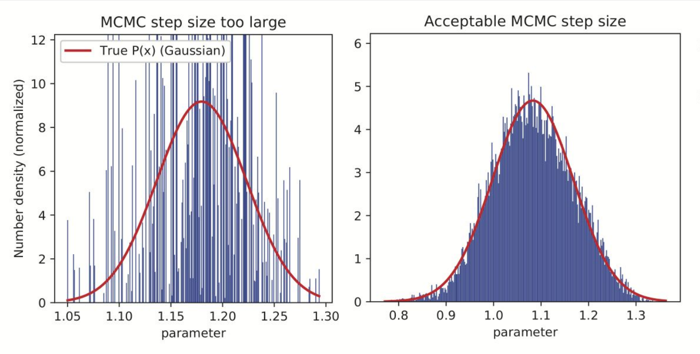
Pictured: animation of MH sampler running over time, sampling a single-parameter posterior distribution with a Gaussian proposal distribution.

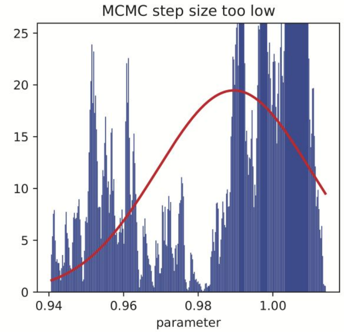
https://blog.stata.com/wp-content/uploads/2016/11/animation3.gif

THE UNIVERSITY *of* NORTH CAROLINA *at* CHAPEL HILL

But Why Use *Adaptive* MCMC?

- With traditional, non-adaptive Metropolis-Hastings, the covariance matrix Σ, i.e. the standard deviations σ for each of the model/slop parameters of the proposal distribution Q are chosen in advance.
    - These σ can be considered as "step-sizes" for the Markov Chain sampler in the algorithm, and as shown to the right, erroneous values for these can lead to terrible samplings.
    - The *Adaptive* addition to the MH algorithm automatically optimizes these step sizes while the sampling is being completed, such that no user input is required.
- Given a set of R parameters sampled from the posterior distribution, for each parameter, we compute the *possibly asymmetric* intervals that contain 68.27%, 95.45% and 99.73% of the samples, i.e. $\pm 1\sigma$, $\pm 2\sigma$ and $\pm 3\sigma$.
    - These intervals are determined with a bracketing/bisection-type algorithm.





Pictured: Three different sets of 100,000 samples generated through MH, with proposal distribution "step sizes" of 10, 0.1, and 0.0001, clockwise. The target posterior is in red.

Part V: Additional Algorithms of the TRK Statistic

Pivot Points

- Model parameters of linear (or linearizable) models are often correlated; however, there is a way to remove such correlation.
- Such models are linearizable because they can be written in the form $y = b + m(x - x_p)$, where x_p is defined to be the *pivot point* of the model. For example, the power-law model

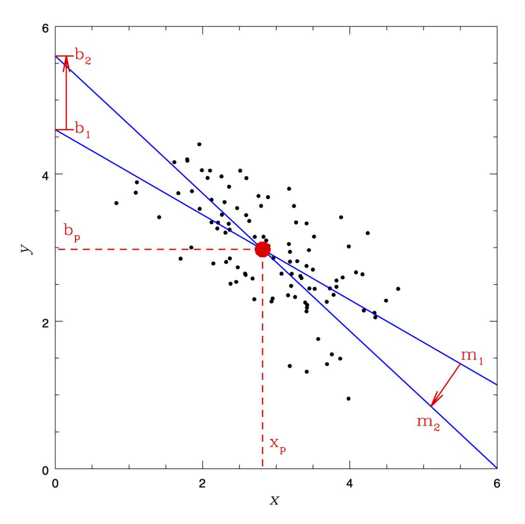
$$y(x) = a_0 \left(\frac{x}{10^{x_p}} \right)^{a_1}$$

can be written as

$$\begin{aligned} \underbrace{\log_{10} y(x)}_{y} &= \log_{10} a_0 + a_1 \left[\log_{10} x - \log_{10} 10^{x_p} \right] \\ &\equiv \underbrace{\log_{10} a_0}_{b} + \underbrace{a_1}_{m} \left[\underbrace{\log_{10} x}_{x} - \underbrace{(\log_{10} x)_p}_{x_p} \right]. \end{aligned}$$

$$y = b + m(x - x_p)$$

- Given such a model, various fits, perturbed from some best fit, will intersect through the same pivot point x_p. If this x_p is chosen to be the x_p of the model, the correlation between the intercept and slope parameters b and m, respectively, will be removed (Trotter 2011).
- So, how can we reliably determine this optimum pivot point?



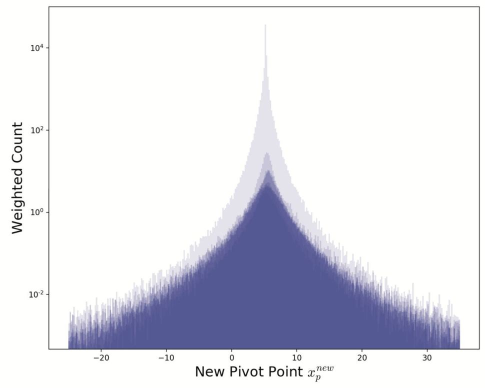
Pictured: two different linear fits of the same dataset. Both intersect at some optimum pivot point x_p.

Finding Pivot Points

- Consider two linear(ized) fits with parameters b_1, m_1 and b_2, m_2, with some chosen shared pivot point x_p^{old}. Shown on the previous figure, the *optimal* pivot point x_p^{new} should be where the lines intersect. This leads to the relation

$$x_p^{\mathrm{new}} = x_p^{\mathrm{old}} + \frac{b_1 - b_2}{m_2 - m_1}.$$

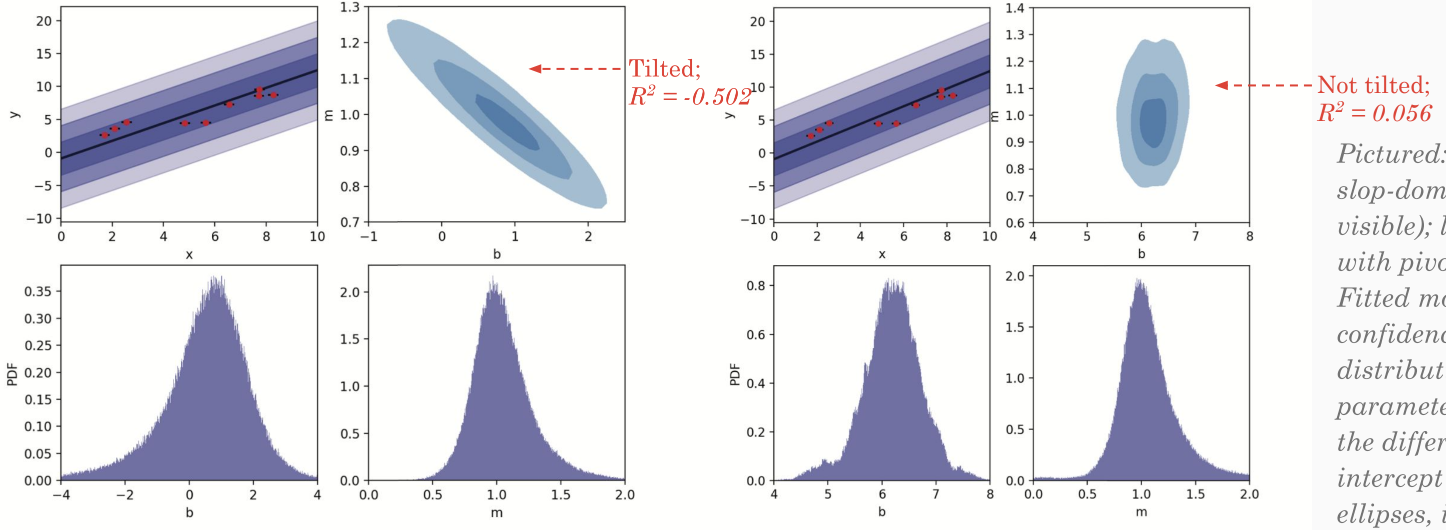
- However, which sets of parameters should we actually use to compute x_p^{new}? None are known *a priori*, so we take a Monte Carlo approach:
  a. Generate ~10,000 parameter samples with MH sampler, and then ~100,000 2-combinations of sets of (b, m) from this sample.
  b. Compute ~100,000 possible pivot points using these combinations and the above equation, and assign weights to each using standard propagation of uncertainty.
  c. Take the weighted half-sample mode (Bickel et. al 2005) of this distribution as the final x_p^{new}.
  d. Repeat above steps until convergence, setting $x_p^{\mathrm{old}} \rightarrow x_p^{\mathrm{new}}$.



Pictured: ten different iterations of MCMC-generated weighted distributions of possible new pivot points x_p^{new}.

An Example of Correlation Removal

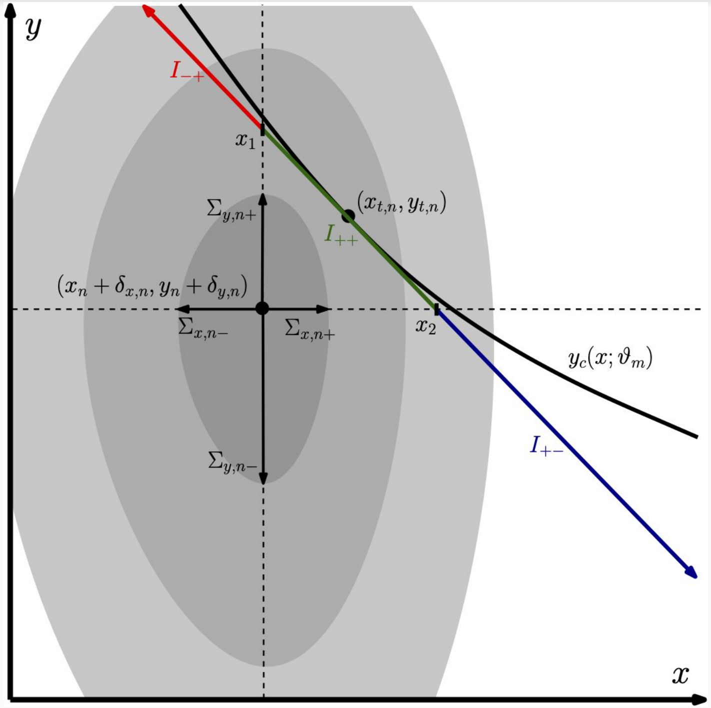
Before pivot point optimization, and… *After*



Pictured: Two linear best fits on slop-dominated dataset (error bars not visible); left and right are without and with pivot point optimization, respectively. Fitted models are shown with 1-, 2- and 3σ confidence regions, and MCMC-generated distributions for intercept and slope parameters are shown at the bottom. Note the difference in tilt/correlation between intercept vs. slope 1-, 2- and 3σ confidence ellipses, indicating the removal of correlation from the pivot point optimization.

Fitting with Asymmetric Uncertainties

- Up until now, we've only considered datasets and models with error bars and slop that follow *symmetric*, ordinary Gaussian distributions.
- However, what if we have *asymmetric* error bars (i.e. difference between -σ and +σ), and/or asymmetric slop?
- My code also allows for this; however, I won't get into this can of worms for now, as it requires re-doing essentially all of the previous analysis...



Part VI: Applications and Examples

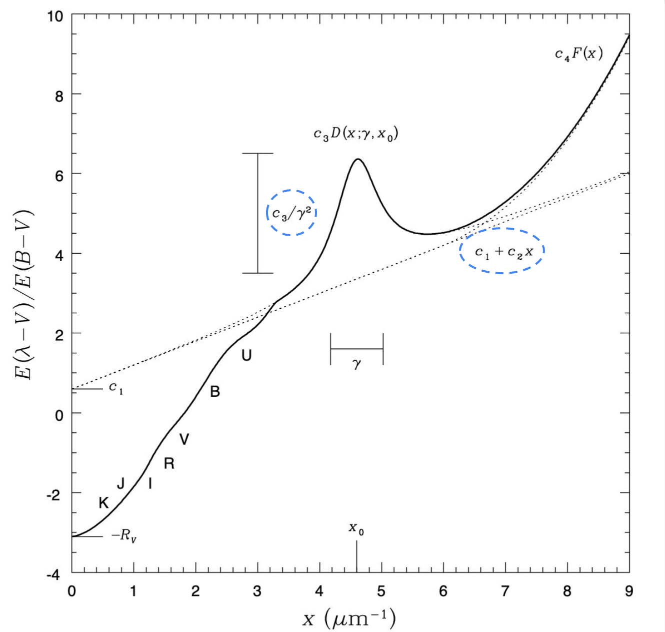
Comparison to Similar 2D Statistics

- Other Bayesian methods that can account for both 2D error bars and slop:
  - Recall that in our 2D uncertainty scenario, the choice of the rotated coordinate u_n will *define* the statistic.
  - For TRK, we chose u_n to be perpendicular to the line segment connecting the datapoint with the tangent point.
  - **D'Agostini 2005** chose $u_n \equiv x$; while this statistic is scalable, it is ***not invertible.***
  - **Reichart 2001** chose $u_n \equiv \sqrt{x^2 + y^2}$, and while it is invertible, it is ***not scalable.***

- Non-Bayesian methods:
  - **Orthogonal Distance Regression/Total Least Squares:**
  - A least squares method that minimizes the distances between the data points and the fitted curve along some directions determined by the error bars.
  - While useful in a pinch there are a number of downsides:
    1. No method to fit extrinsic scatter/slop,
    2. No general way to make it Bayesian, e.g. including priors
    3. ODR is ***not scalable*** (Isobe et al. 1990)

Interstellar Extinction TRK Fits

- Trotter 2011 did various fits to describe the relationship between parameters that define the CCM/FM (Cardelli et al. 1989 and Fitzpatrick et. al 1988) interstellar dust extinction model, shown to the right.
- I redid all of the fits with the TRK Suite that I wrote, as proof of concept of my own code.
- The CCM/FM model parameters that I explored are c_1, c_2, c_3, and γ, circled to the right.
- I updated Trotter's dataset of ~441 stars with a thorough literature search, so that the dataset I used has ~729 stars.



Pictured: the CCM/FM model, with our parameters of interest circled.

The c_1 vs. c_2 Model Fit

- The parameters c_1 and c_2 are strongly linearly correlated, as they describe the intercept and slope of a linear component of the CCM/FM model. As such, I ran a TRK fit on the dataset with a model curve of the form

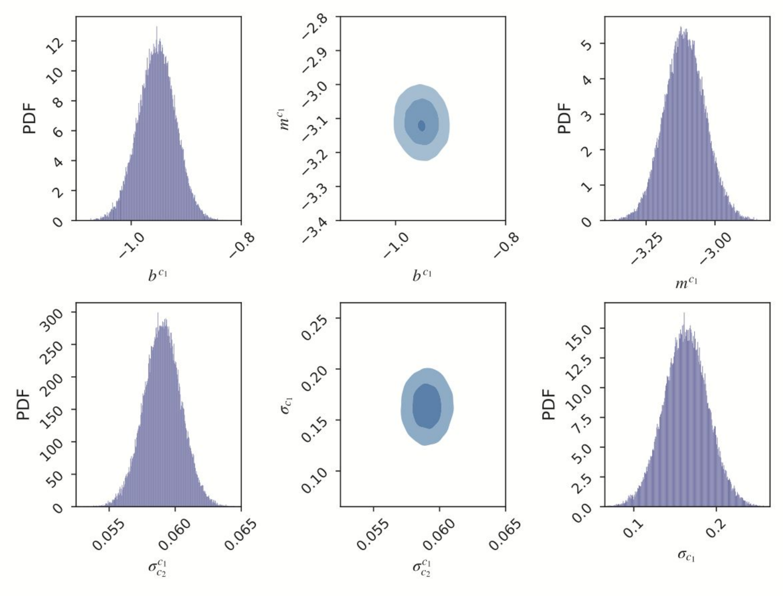
$$c_{1,c}(c_2;\vartheta_m) = b^{c_1} + m^{c_1}\left(c_2 - c_2^{p_{c_1}}\right),$$

  including scale optimization, pivot point-finding (for $c_2^{p_{c_1}}$), finding the best fit with the Downhill Simplex method, and MCMC-generated parameter uncertainties.

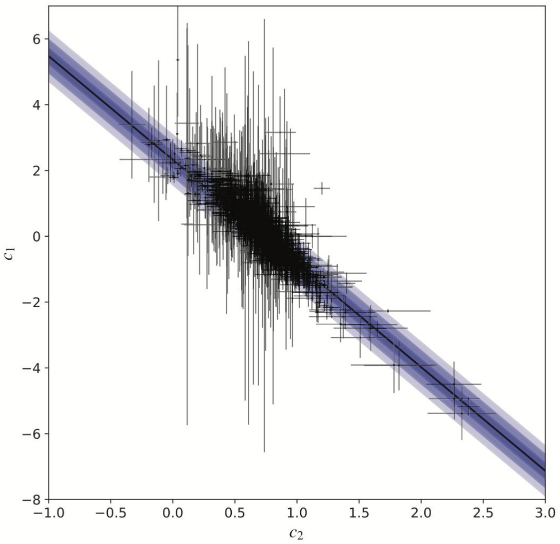
- Weights are also applied to each datapoint for this, and the next fit, that essentially "balance out" the occasional high c_2 datapoint.

Results of the c_1 vs. c_2 Model Fit

Pictured right: MCMC-generated posterior distributions for the slope and intercept model parameters, as well as the c_1- and c_2-slop parameters. Note that due to the pivot point-finding algorithm, the intercept and slope are uncorrelated. Also note that as expected, the slop parameters are uncorrelated.



$$c_{1,c}(c_2;\vartheta_m) = b^{c_1} + m^{c_1}\left(c_2 - c_2^{p_{c_1}}\right)$$



Pictured above: my c_1 vs. c_2 linear fit, with plotted 1-, 2- and 3σ slop confidence regions.

The BH vs. c_2 Model Fit

- The "bump height" parameter $\mathrm{BH} \equiv c_3/\gamma^2$ of the CCM/FM model is only loosely correlated with c_2; moderate values of c_2 tend to have higher values of BH, while low or high values tend to have low BH. This relationship can be parameterized as a *"smoothly-broken linear" model* of the form

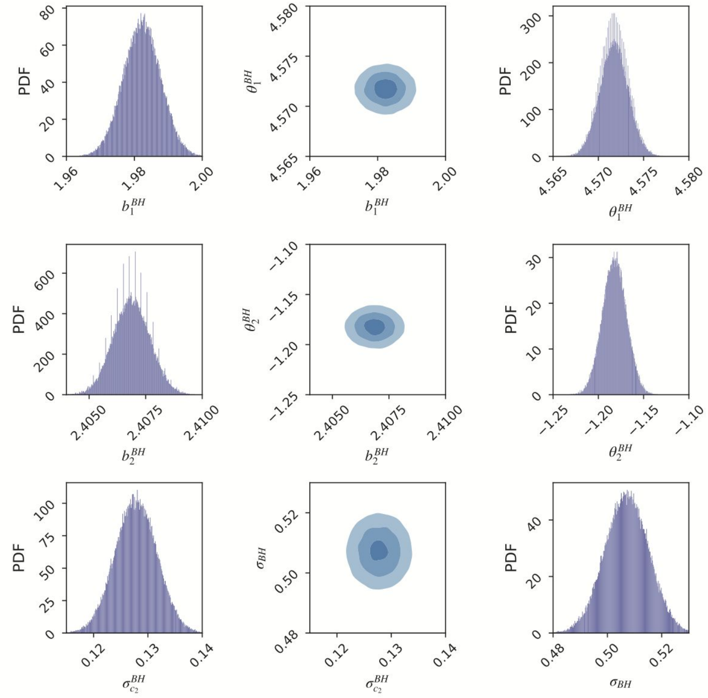
$$\mathrm{BH}_c(c_2;\vartheta_m) = -\ln\left[\exp\left\{-b_1^{\mathrm{BH}} - \tan\theta_1^{\mathrm{BH}}\left(c_2 - c_2^{p_1,\mathrm{BH}}\right)\right\} + \exp\left\{-b_2^{\mathrm{BH}} - \tan\theta_2^{\mathrm{BH}}\left(c_2 - c_2^{p_2,\mathrm{BH}}\right)\right\}\right]$$

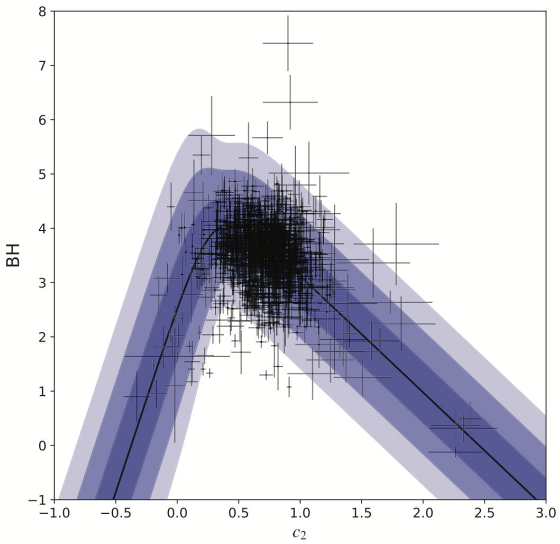
which describes two smoothly-joined lines with intercept and slope *angle* pairs of $(b_1^{\mathrm{BH}}, \theta_1^{\mathrm{BH}})$ and $(b_2^{\mathrm{BH}}, \theta_2^{\mathrm{BH}})$.

- I ran a TRK fit on the dataset with this model curve, including scale optimization, pivot point-finding (for $c_2^{p_1,\mathrm{BH}}$ and $c_2^{p_2,\mathrm{BH}}$), finding the best fit with the Downhill Simplex method, and MCMC-generated parameter uncertainties.
- Note that this model has *two pivot points*; as such, I modified the pivot-point finding algorithm to essentially work on both linear "legs" of the model simultaneously.

Results of the BH vs. c_2 Model Fit

Pictured right: MCMC-generated posterior distributions for the two sets of slope angle and intercept model parameters, as well as the BH- and c_2-slop parameters. Note that due to the pivot point-finding algorithm, both sets of intercept and slope (angle) are uncorrelated. Also note that as expected, the slop parameters are uncorrelated.



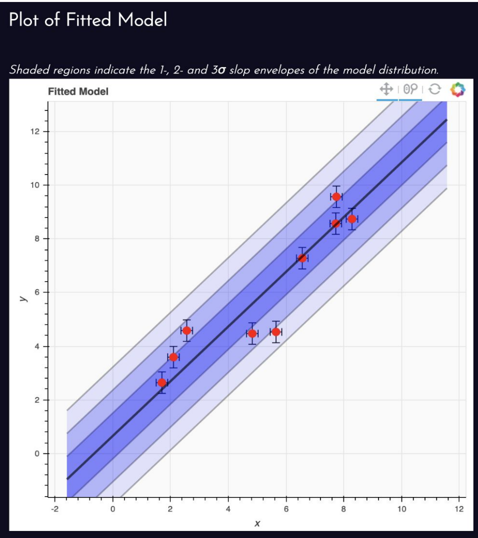


Pictured above: my c_1 vs. c_2 linear fit, with plotted 1-, 2- and 3σ slop confidence regions.

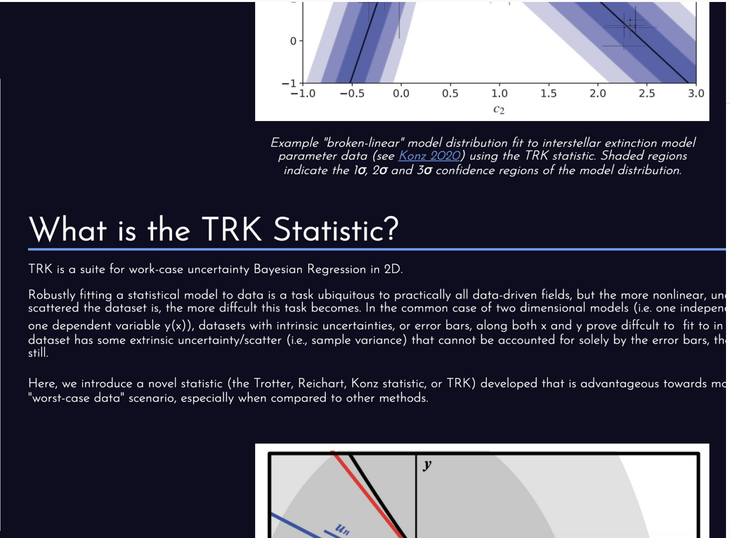
The TRK Website, Repo and Calculator

I also built a website (https://skynet.unc.edu/rcr/calculator), GitHub code repository (https://github.com/nickk124/TRK/) with code documentation and an easy to use, end-to-end web calculator for running TRK fits (https://skynet.unc.edu/rcr/calculator/trk).

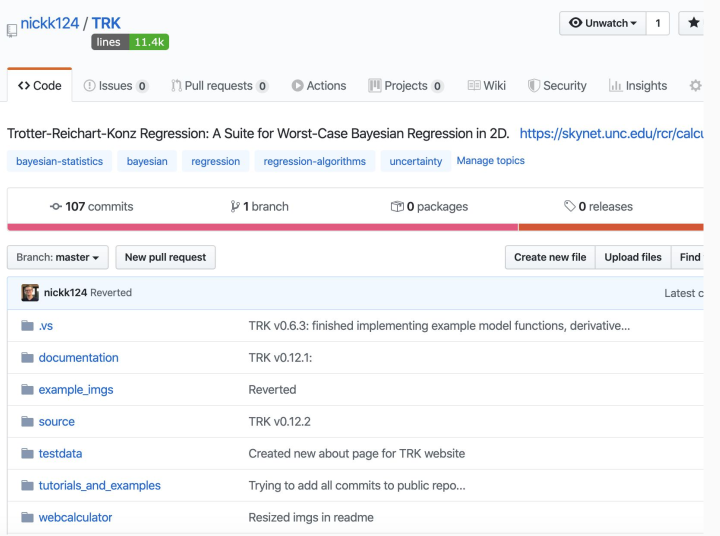
Calculator



Website

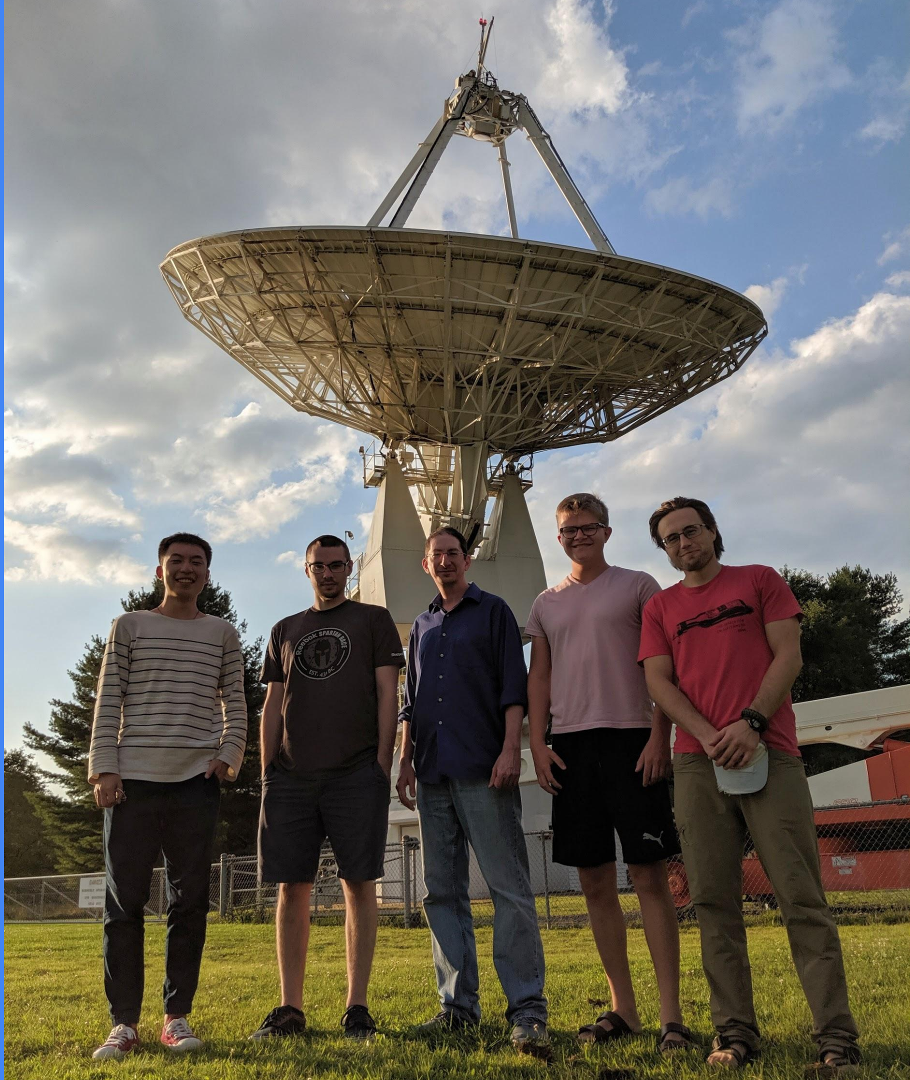


Repo



Epilogue: Future Endeavors

Thanks to Everyone!



THE UNIVERSITY of NORTH CAROLINA at CHAPEL HILL

Bibliography

- Bickel, David R and Rudolf Fruehwirth (2005). “On a fast, robust estimator of the mode: Comparisons to other robust estimators with applications.” In: arXiv preprint math/0505419.
- Cardelli, Jason A, Geoffrey C Clayton, and John S Mathis (1989). In: The Astrophysical Journal 345, pp. 245-256
- D'Agostini, G (2005). “Fits, and especially linear ts, with errors on both axes, extra variance of the data points and other complications.” In: arXiv preprint physics/0511182
- Fitzpatrick, Edward L and Derck Massa (1988). In: The Astrophysical Journal 328, pp. 734-746
- Haario, Heikki, Eero Saksman, Johanna Tamminen, et al. (2001). “An adaptive Metropolis algorithm.” In: Bernoulli 7.2, pp. 223-242 (cit. on p. 36).
- Isobe, Takashi et al. (Nov. 1990). “Linear Regression in Astronomy. I.” In: 364, p. 104
- Nelder, J. A. and R. Mead (1965). In: Computer Journal 7, pp. 308-313.
- Reichart, Daniel E (2001). “Dust extinction curves and Lyα forest flux decits for use in modeling gamma-ray burst afterglows and all other extragalactic point sources.” In: The Astrophysical Journal 553.1
- Tiruneh, Ababu Teklemariam, William N Ndlela, and Stanley J Nkambule (2013). “A two-point Newton method suitable for nonconvergent cases and with super-quadratic convergence.” In: Advances in Numerical Analysis 2013.
- Trotter, Adam S. (2011). “The Gamma-Ray Burst Afterglow Modeling Project: Foundational Statistics and Absorption & Extinction Models.” PhD thesis. The University of North Carolina at Chapel Hill.