Hunting for signals using Gaussian Process Regression

Abhijith Gandrakota, Alexandre V. Morozov, Amit Lath, Sindhu Murthy

ACAT 2022
Bari, Italy

Reference: arxiv:2202.05856

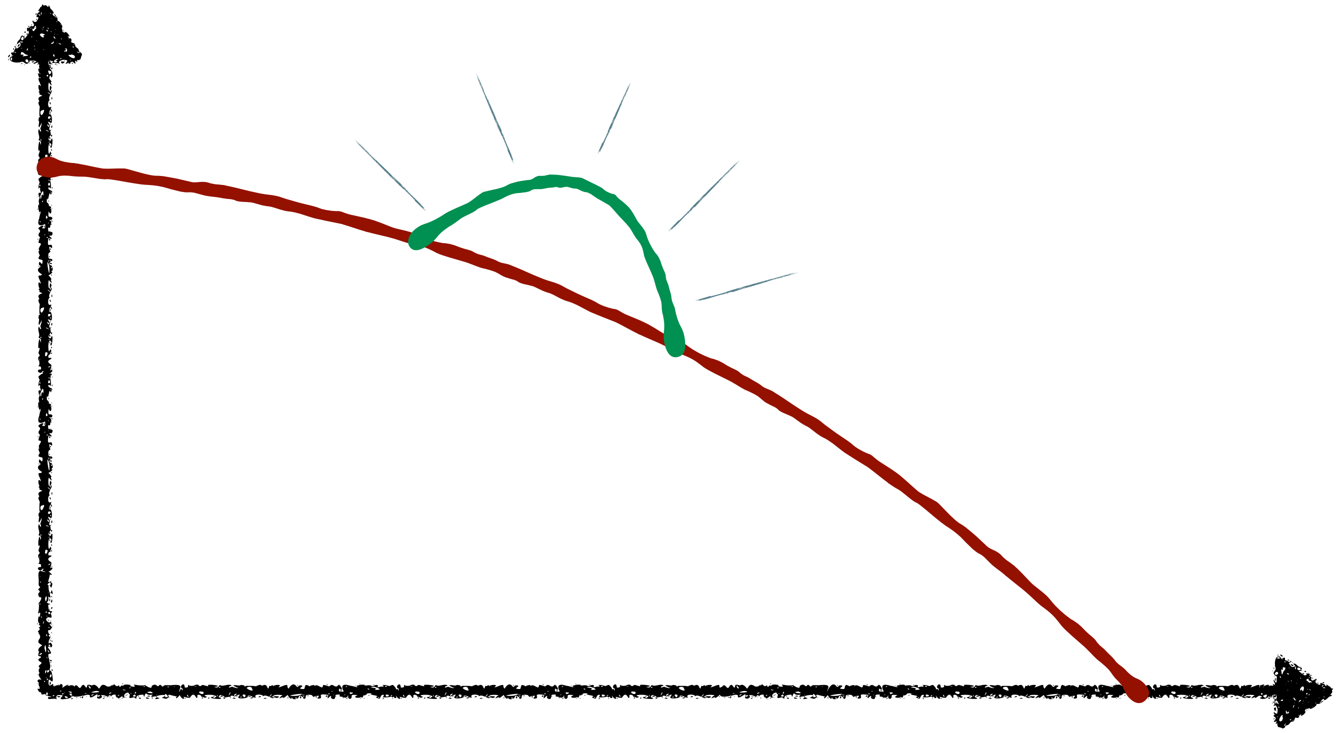
Introduction

- In many LHC searches, we often look for particle resonances
- These resonances are often manifested as local features in mass distributions
- One essential procedure we do to find signal / deviation from bkg

Introduction

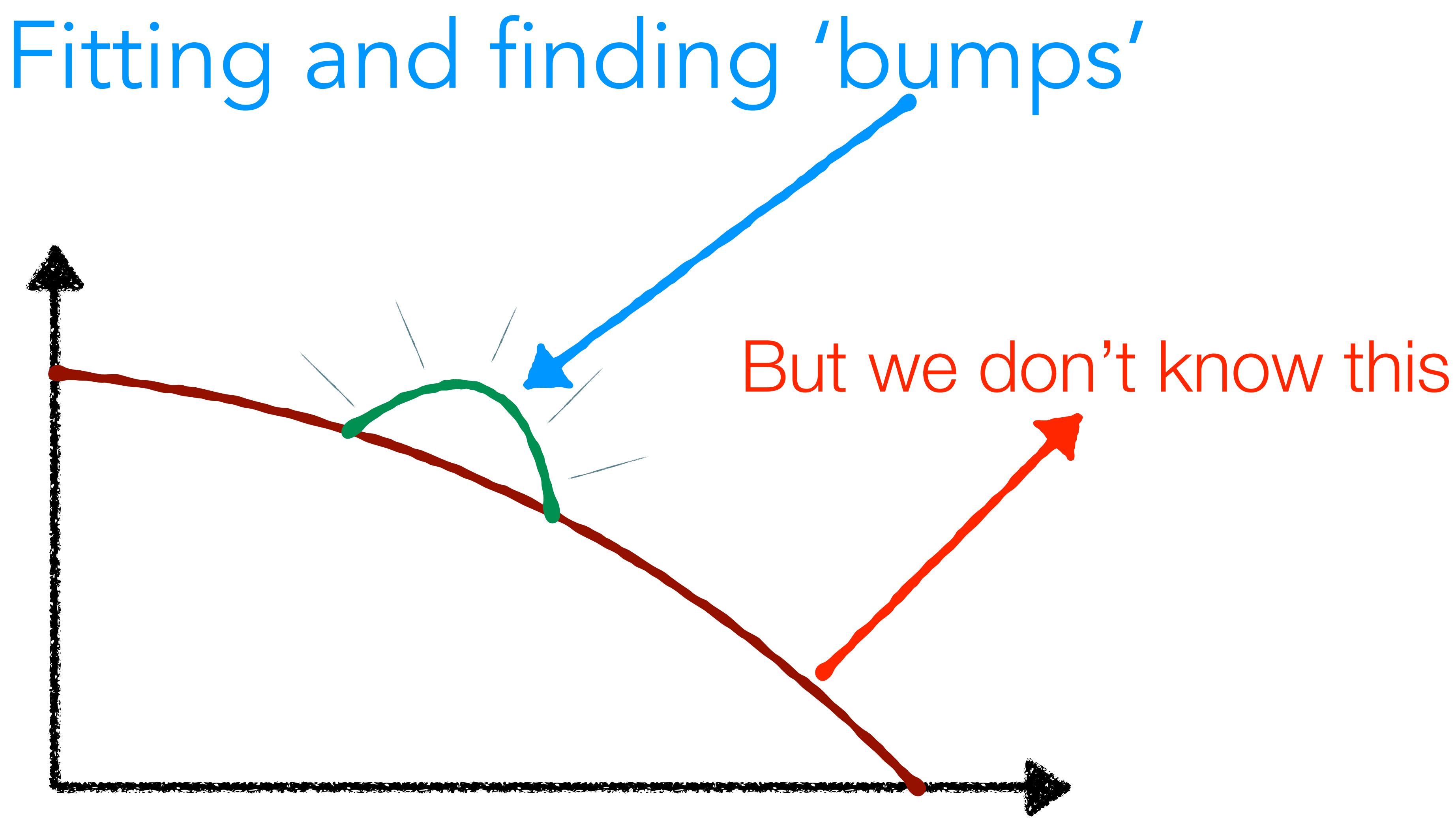
- In many LHC searches, we often look for particle resonances
- These resonances are often manifested as local features in mass distributions
- One essential procedure we do to find signal / deviation from bkg

Fitting and finding 'bumps'



Introduction

- In many LHC searches, we often look for particle resonances
- These resonances are often manifested as local features in mass distributions
- One essential procedure we do to find signal / deviation from bkg

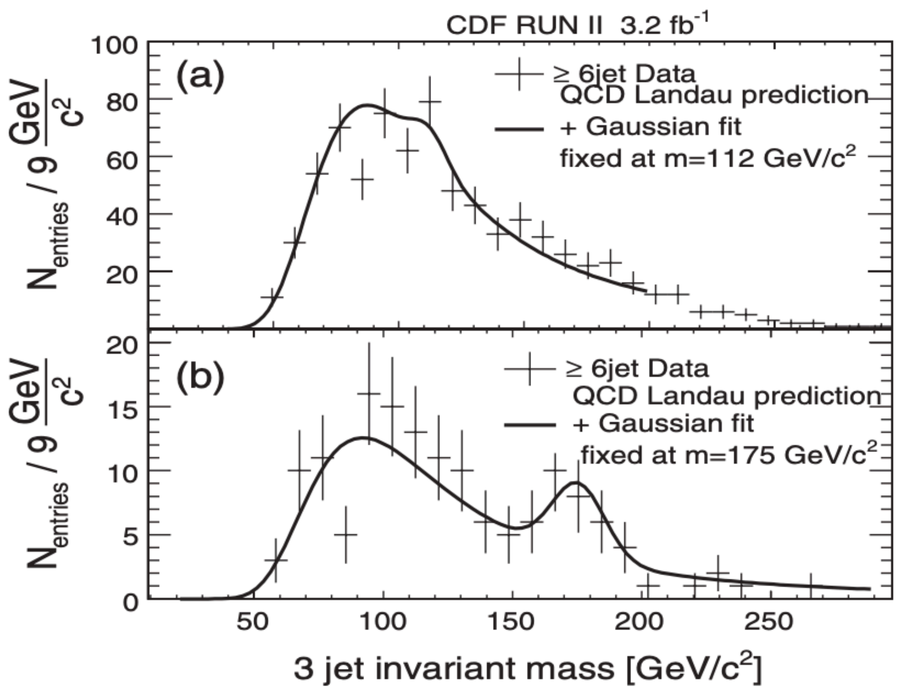


Finding signal bumps

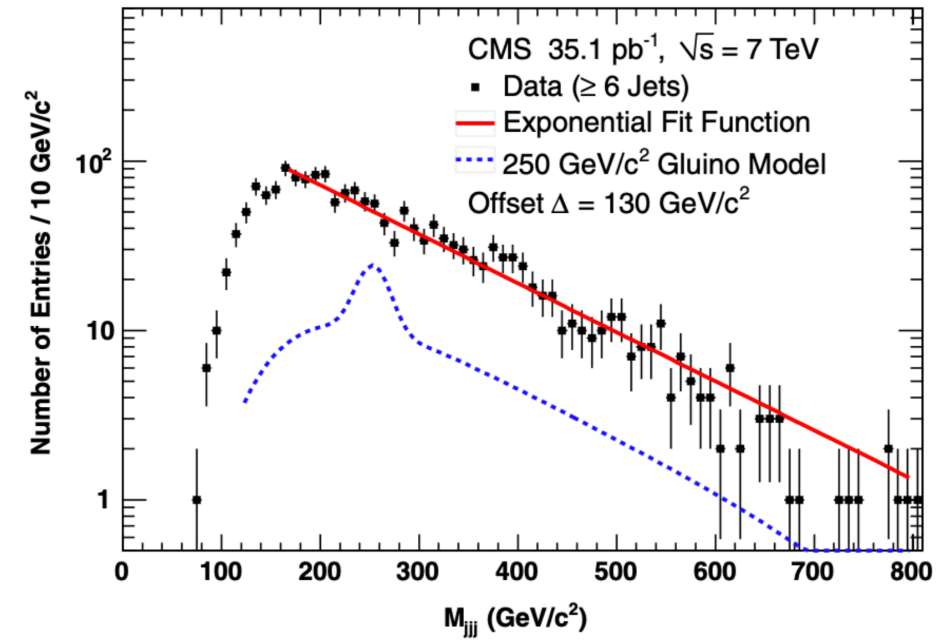
- There is one essential procedure we do to find localized signal / deviation from bkg
- Procedure to Fit:
  - Option 1: Use data driven methods + Signal template
    - Hard to find a method that works and very specific to the analysis
  - Option 2: Fit a smooth function + Gaussian to the data
    - How are we choosing this smooth function? it's Ad-hoc !

But what function to choose ?

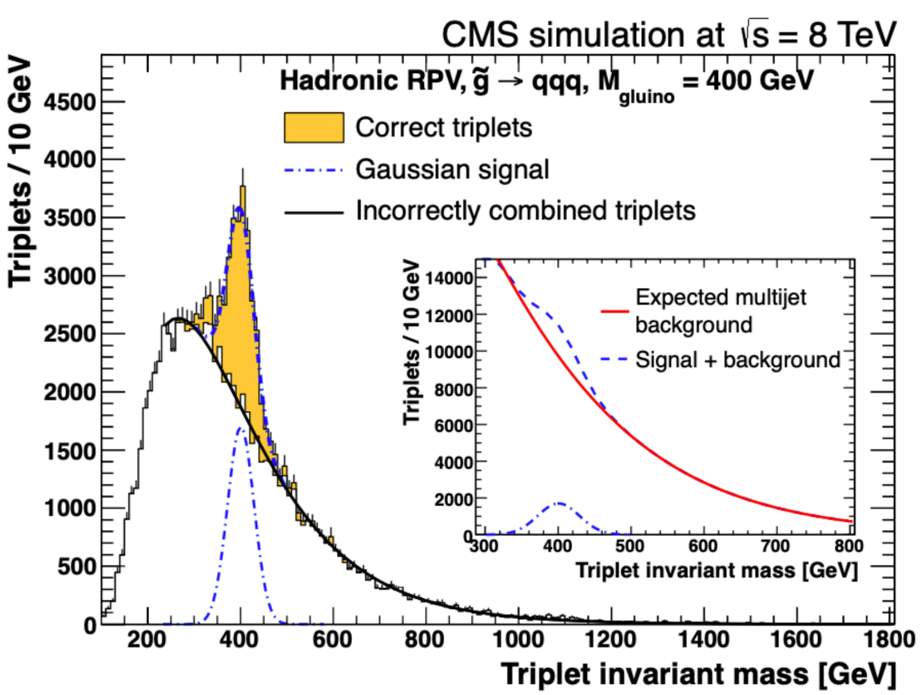
- **In the history of search for RPV gluinos, The fit function changed with every search !**

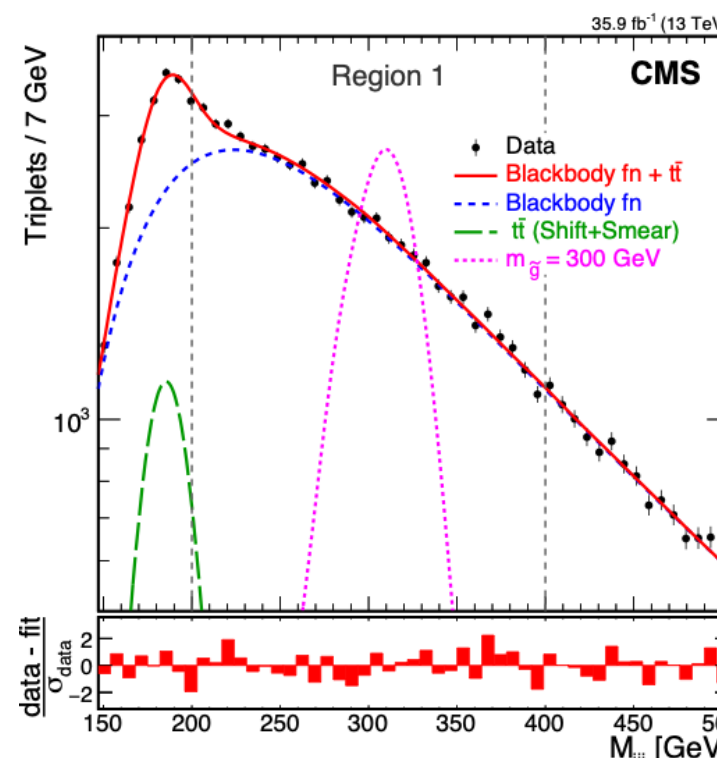


CDF: Landau (x) gaussian



CMS 7 TeV - Exponential





CMS 8 TeV $$\frac{dN}{dx} = P_0 \frac{\left(1-\frac{x}{\sqrt{s}}\right)^{P_1}}{\left(\frac{x}{\sqrt{s}}\right)^{P_2+P_3\log\frac{x}{\sqrt{s}}}},$$

CMS 13 TeV $$\frac{dN}{dx} = \frac{1}{(x+c)^{5+d\ln\frac{x}{\sqrt{s}}}}\,\frac{a}{e^{\frac{b}{x+c}}-1}$$

Finding signal bumps

- There is one essential procedure we do to find localized signal / deviation from bkg
- Procedure to Fit:
  - Option 1: Use data driven methods + Signal template
    - Hard to find a method that works and very specific to the analysis
  - Option 2: Fit a smooth function + Gaussian to the data !
    - How are we choosing this smooth function? it's Ad-hoc !

Anything better on the menu ?

Finding signal bumps

- There is one essential procedure we do to find localized signal / deviation from bkg
- Procedure to Fit:
  - Option 1: Use data driven methods + Signal template
    - Hard to find a method that works and very specific to the analysis
  - Option 2: Fit a smooth function + Gaussian to the data !
    - How are we choosing this smooth function? it's Ad-hoc !
  - New Option : Bkg estimation method that works with only few assumptions, Can we use ML techniques to infer it directly from data ?

Finding signal bumps

- There is one essential procedure we do to find localized signal / deviation from bkg
- Procedure to Fit:
  - Option 1: Use data driven methods + Signal template
    - Hard to find a method that works and very specific to the analysis
  - Option 2: Fit a smooth function + Gaussian to the data !
    - How are we choosing this smooth function? it's Ad-hoc !
  - New Op ly few assumptions, Can we

Gaussian Prossess Regression !

Gaussian Process Regression

"A Gaussian process is a probability distribution over possible functions that fit a set of points"

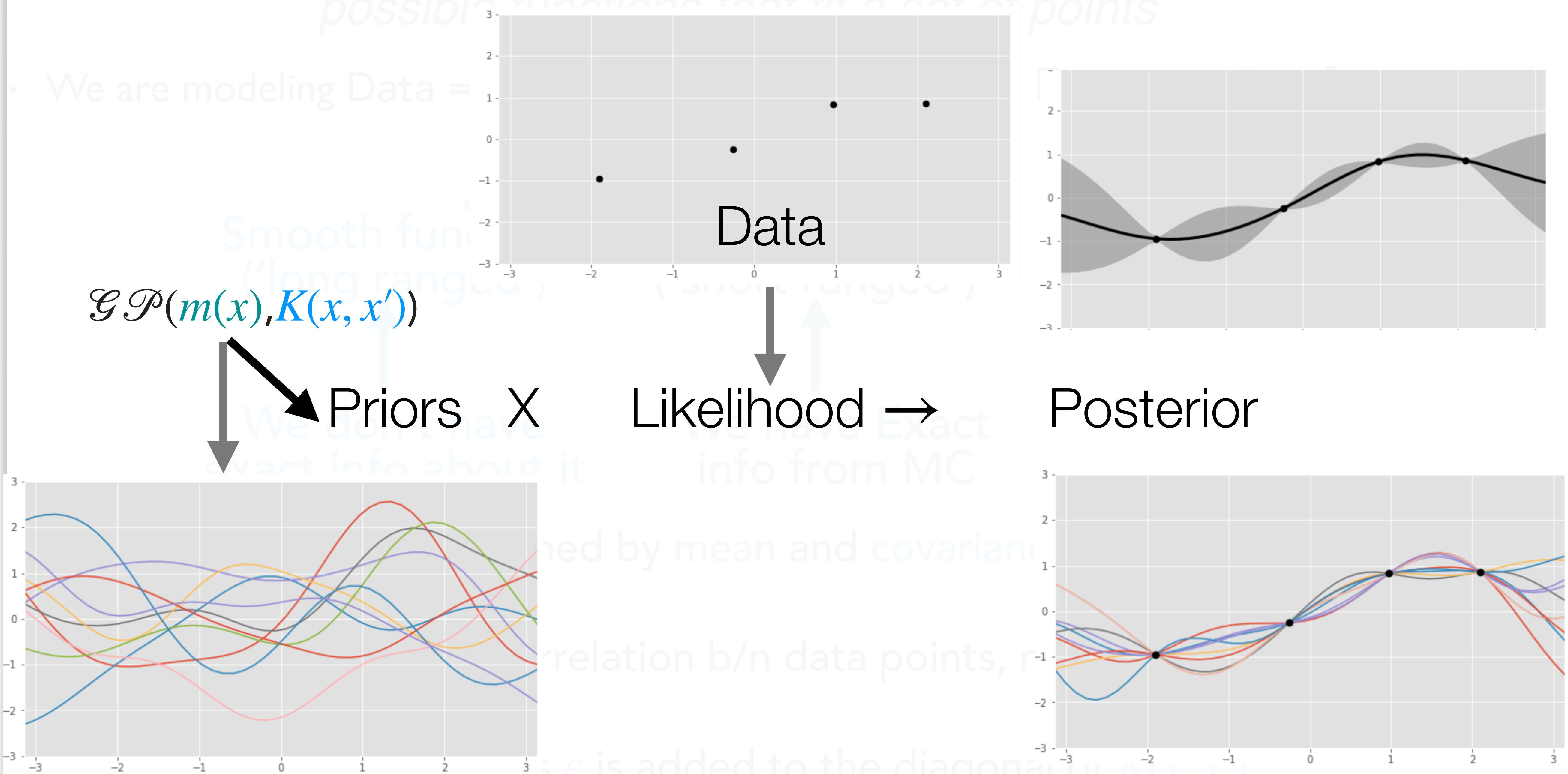
- We are modeling Data = Bkg(x) + Sig(x) + ϵ ← Error coming from experiment

Smooth function ('long ranged') ← We don't have exact info about it

Local feature ('short ranged') ← We have Exact info from MC

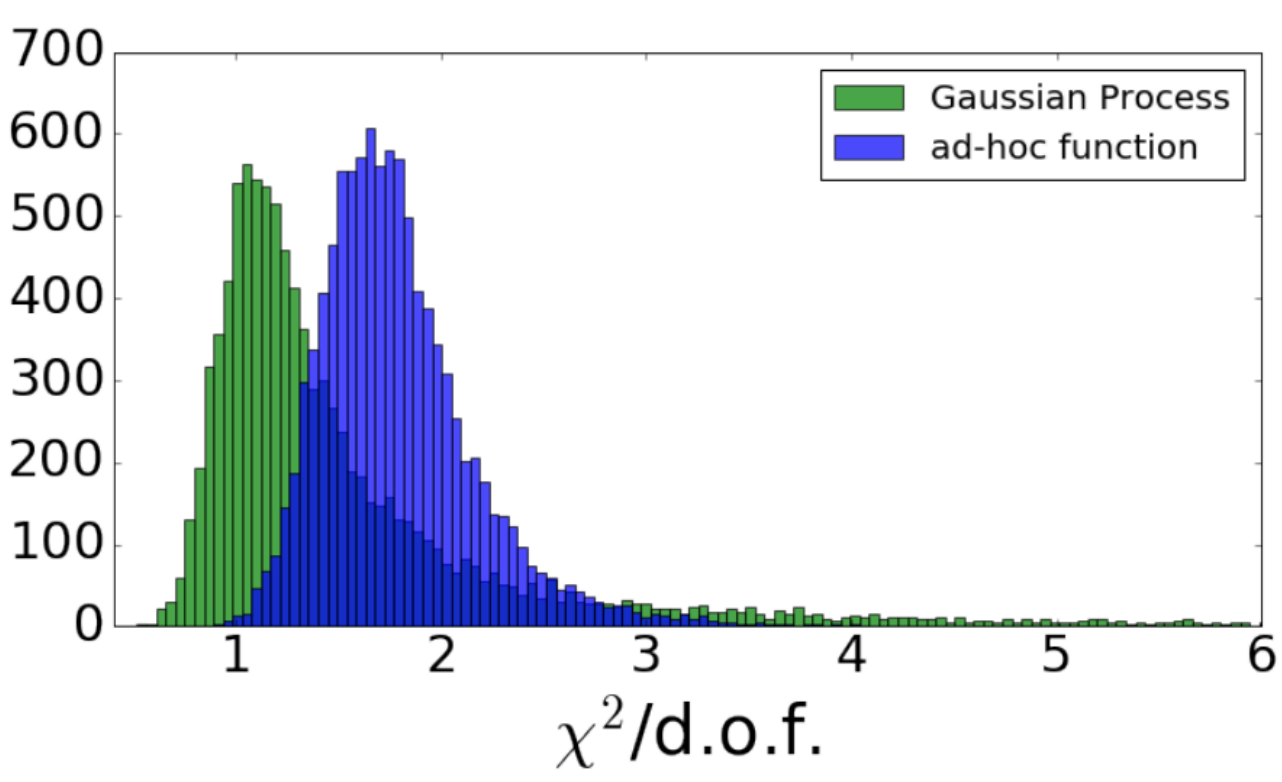
- Like a gaussian, GP is defined by mean and covariance fn ~ $\mathcal{GP}(m(x), K(x, x'))$
- The $K(x, x')$ defines the correlation b/n data points, models smooth background
  - Error in our observations ϵ is added to the diagonal of $K(x, x')$
- The $m(x)$ is used to add additional *interpretability:* extracting signal parameters

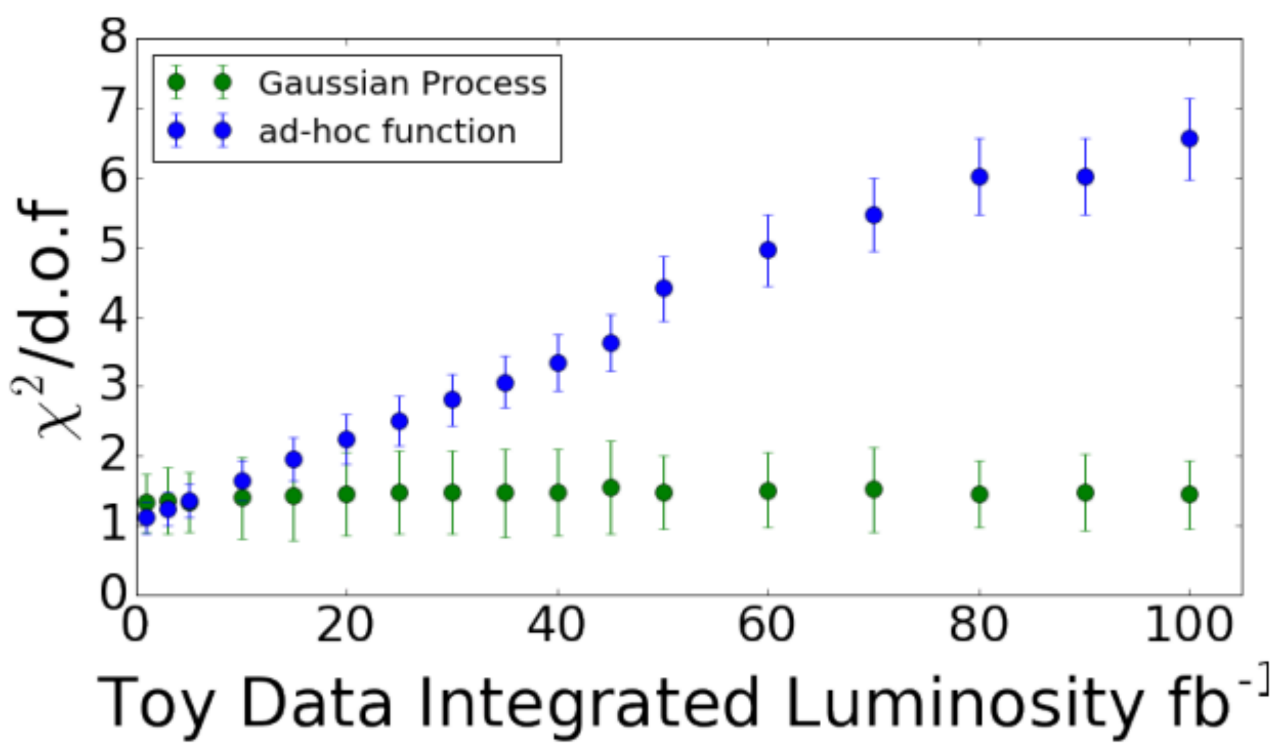
Gaussian Process Regression



Why GP ?

- Very well understood kernel based ML technique and used in various fields
- Use of GP for HEP background modeling is first illustrated in arxiv:1709.05681
  - Tests were performed on toys based on LHC dijet distribution
  - It leads to a constant performance with increasing statistics





[1]: Arxiv: 1709.05681

Why GP ?

- Very well understood kernel based ML technique and used in various fields
- Use of GP for HEP background modeling is first illustrated in illustrated arxiv:1709.05681
  - Tests were performed on toys based on LHC dijet distribution
  - It leads to a constant performance with increasing statistics

- **But what's the catch ?**
  - Choice for $m(x)$ ~ Gaussian / etc $\ldots$, But how do we pick $K(x, x')$?
  - How do we best extract the parameters of signal ~ $m(x)$?
  - A simple prescription for extracting limits and tests on real data
    - Can we add a bit of poisson statistics flavor to it ?

Gaussian Process Regression

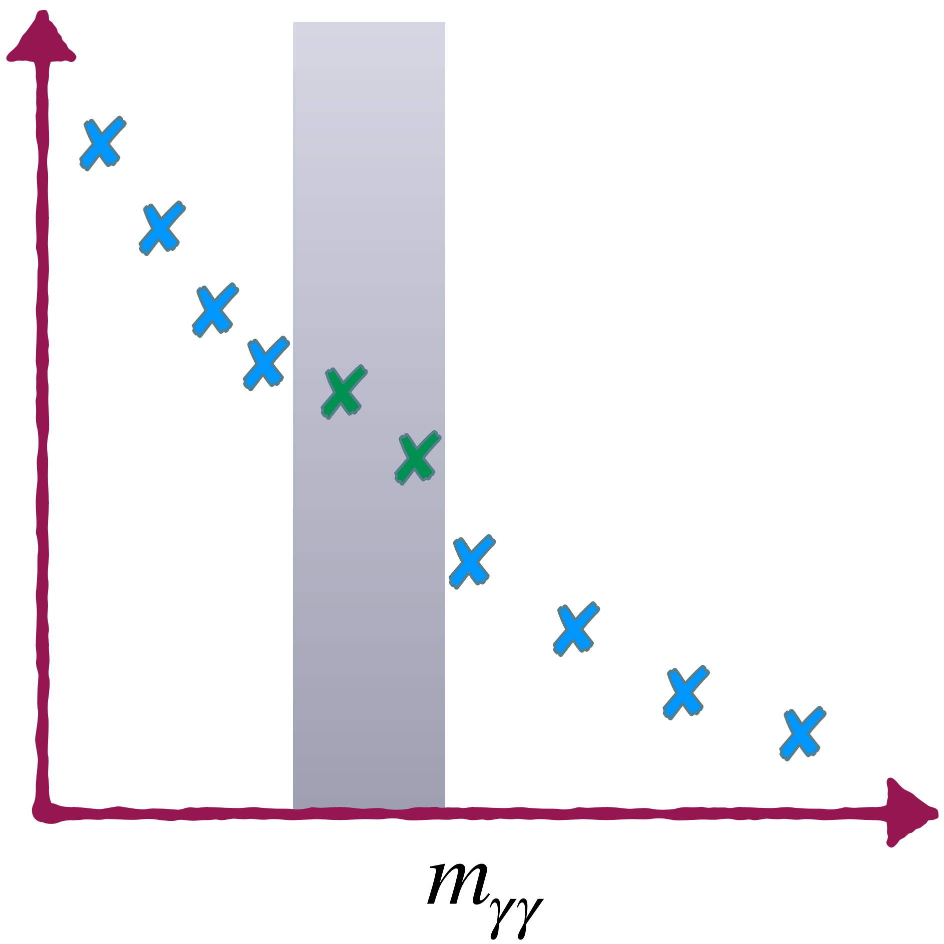
Fermilab

- We are modeling Data = Bkg(x) + Sig(x) + ϵ
- Lets take di-photon data from ATLAS @ LHC, Sig(x) we are keen in finding out is $H \to \gamma\gamma$
- We are more interested in figuring out the shape of Bkg(x),
- Mask expected signal region in data, so Data $\sim$ Bkg(x) ~ masking out $\pm 2\sigma$ from expected signal mean
  - No expected signal here so $m(x)$ ~ 0
- For a covariance, say $K(x, x') = A^2 \exp\left(-\frac{(x-x')^2}{2l^2}\right)$ optimize Hyper-Parameters (θ) : A, l by minimizing likelihood

$$\log p(y|X) = -\frac{1}{2} y^T (K + diag(\sigma^2))^{-1} y - \frac{1}{2} \log|K + \sigma_n^2 I| - \frac{n}{2} \log 2\pi$$

Goodness of fit — Complexity penalty

  - Use this to get predicted Bkg(x) distribution
- We can repeat it for different $K(x, x')$, How do we pick the best one out ?



Abhijith Gandrakota

GP : Model selection

- We applied various kernels for modeling Bkg(x) in masked di-photon data

  $k_{\text{Poly2}}(x, x')$, $k_{\text{RBF}}(x, x')$ and $k_{\text{Matern}}(x, x')$ [definitions of kernels in backup]

- Using optimized θ, calculate metrics to compare kernels

- Some of the main ingredients to calculate comparison metrics

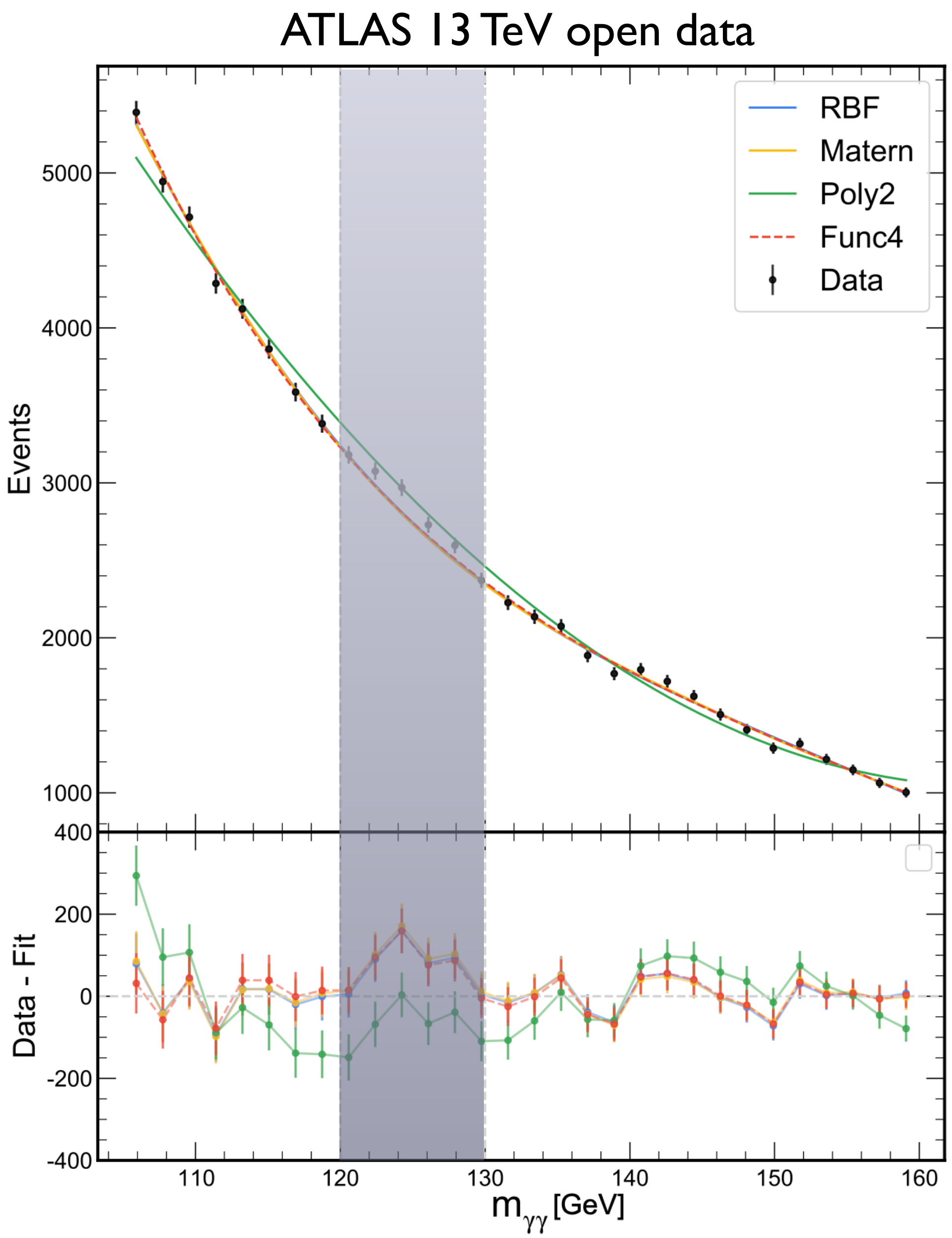
  - Poison Likelihood: $$\log L_{\mathcal{P}} = \sum_{i=1}^{N} \left[y_i - f(x_i) - y_i \log \left(\frac{y_i}{f(x_i)} \right) \right]$$

  - Effective d.o.f : $$d_{eff}(\hat{\theta}) = \text{tr}[K(\hat{\theta})(K(\hat{\theta}) + \sigma^2 I)^{-1}]$$

- Calculate information criteria: $\text{AIC}_{\text{PL}} \equiv -2 \log L_{\mathcal{P}} + d_{eff}$

- We compare results w/ traditional functions: 4th order polynomials

Model	log $\lvert H \rvert$	n	d	-log(PL)	-log(GL)	$\text{BIC}_{\text{GL}}^{\text{naive}}$	BIC_{GL}	AIC_{PL}	$\text{BIC}_{\text{PL}}^{\text{naive}}$
Poly2	-0.531	1	2.99	38.02	87.52	89.22	87.25	82.02	39.72
RBF	0.417	2	4.68	8.95	72.15	75.55	72.36	27.26	12.35
Matern	2.906	2	5.67	8.69	72.30	75.70	73.75	28.72	12.09
Func4	–	5	5	8.65	–	–	–	27.30	17.15



Signal extraction

- With the Bkg(x) figured out, now let's hunt for the signals using $m(x)$

  Best suited kernel : $k_{\mathrm{RBF}}(x, x')$

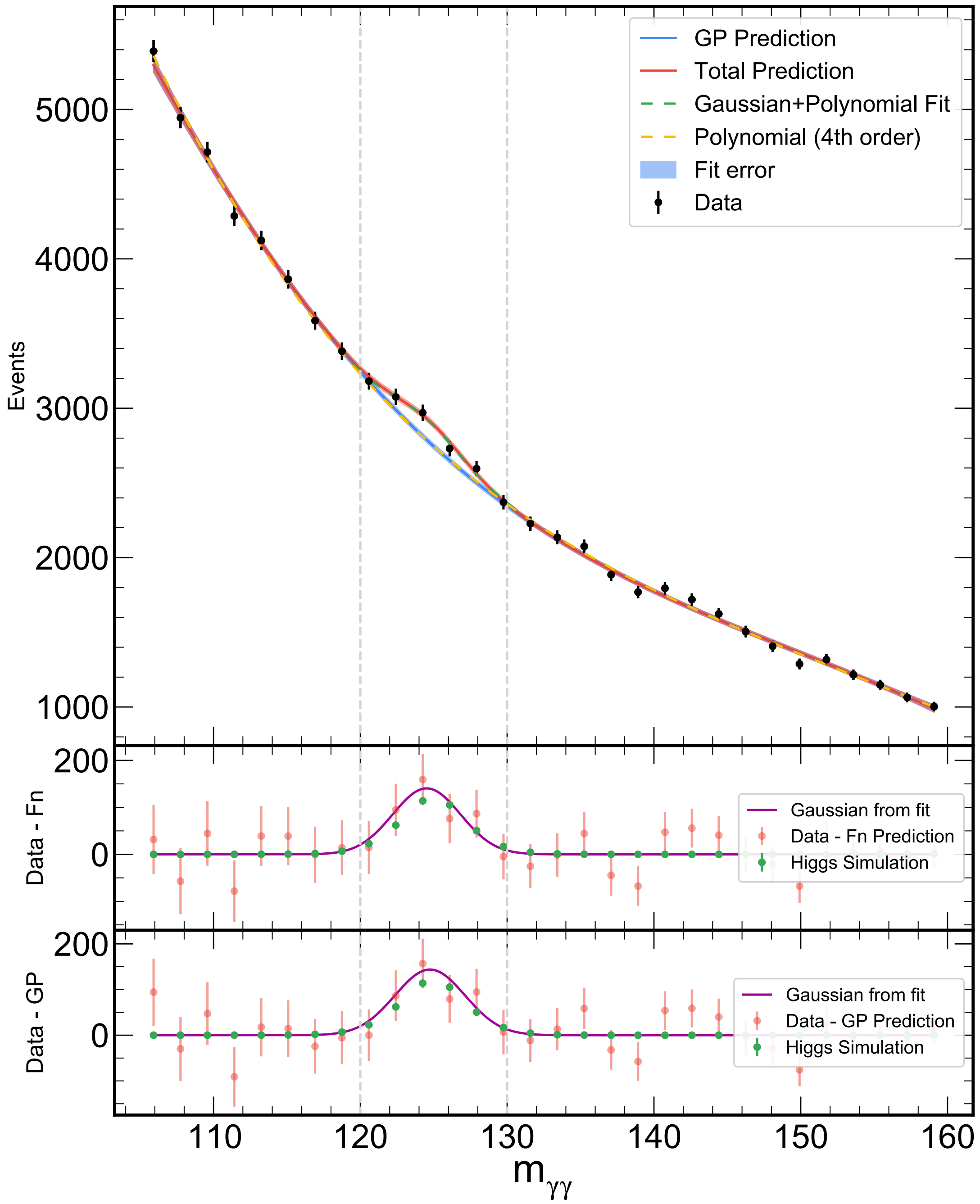
- Signal we are looking for is Higgs ($H \rightarrow \gamma\gamma$)

- For signal we take $m(x_i) = \frac{A}{\sqrt{2\pi\sigma}} \exp\left(-\frac{(x_i-\mu)^2}{2\sigma^2}\right)$

- We take the optimized $\hat{\theta}$, fit for signal parameters using poison likelihood

- Using the GP fits we find signal parameters to be
  - $A_{\mathrm{RBF}}, \mu_{\mathrm{RBF}}, \sigma_{\mathrm{RBF}} = \{473 \pm 123,\ 124.7 \pm 0.6,\ 2.4 \pm 0.4\}$

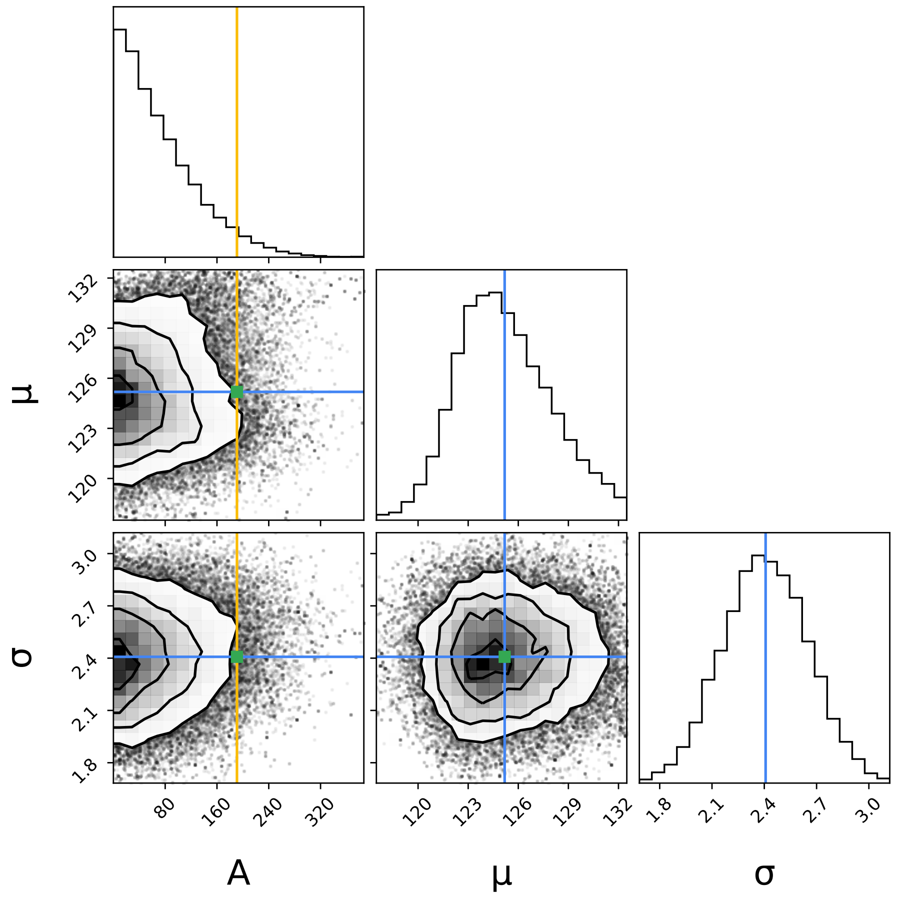
- Using the traditional functional fits we get
  - $A_{\mathrm{Func4}}, \mu_{\mathrm{Func4}}, \sigma_{\mathrm{Func4}} = \{443 \pm 199,\ 124.5 \pm 0.8,\ 2.3 \pm 0.9\}$



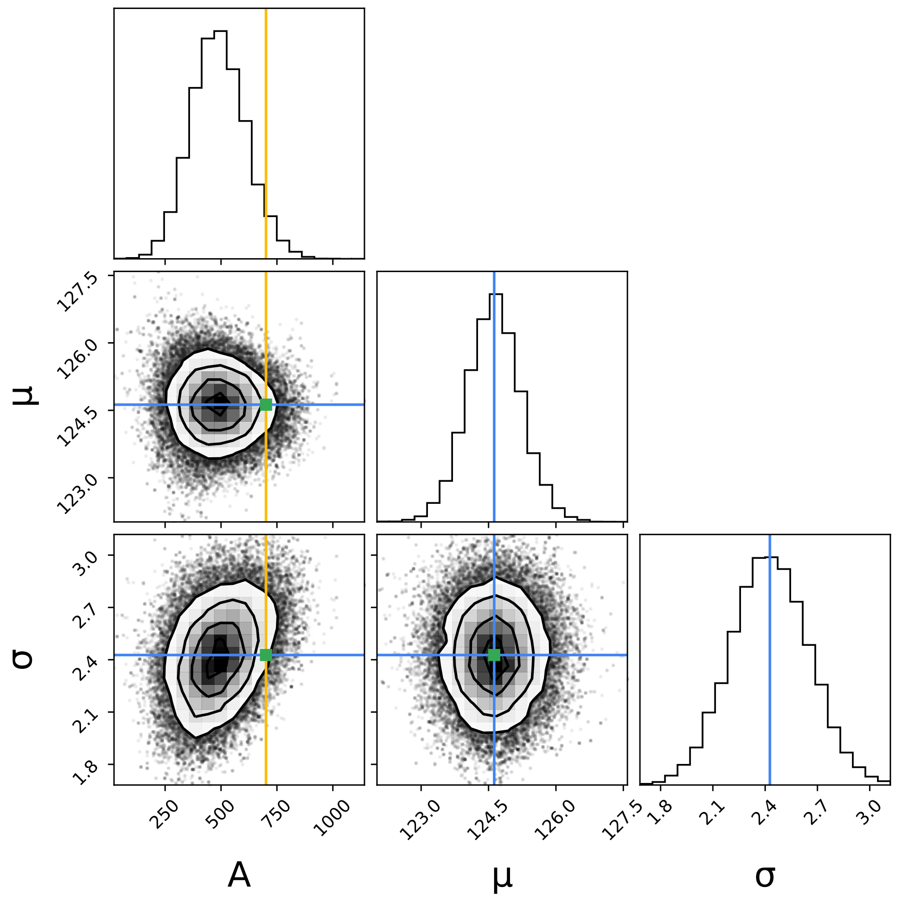
Estimating signal significance

- For significance we need the posterior distributions of signal parameters
  - Estimated by carrying out Markov Chain Monte Carlo (MCMC) of Poison likelihood
  - With systematic uncertainties as priors on signal parameters
- We integrate the amplitude posterior distribution (A) to get 95% CL value

BKG only toy dataset

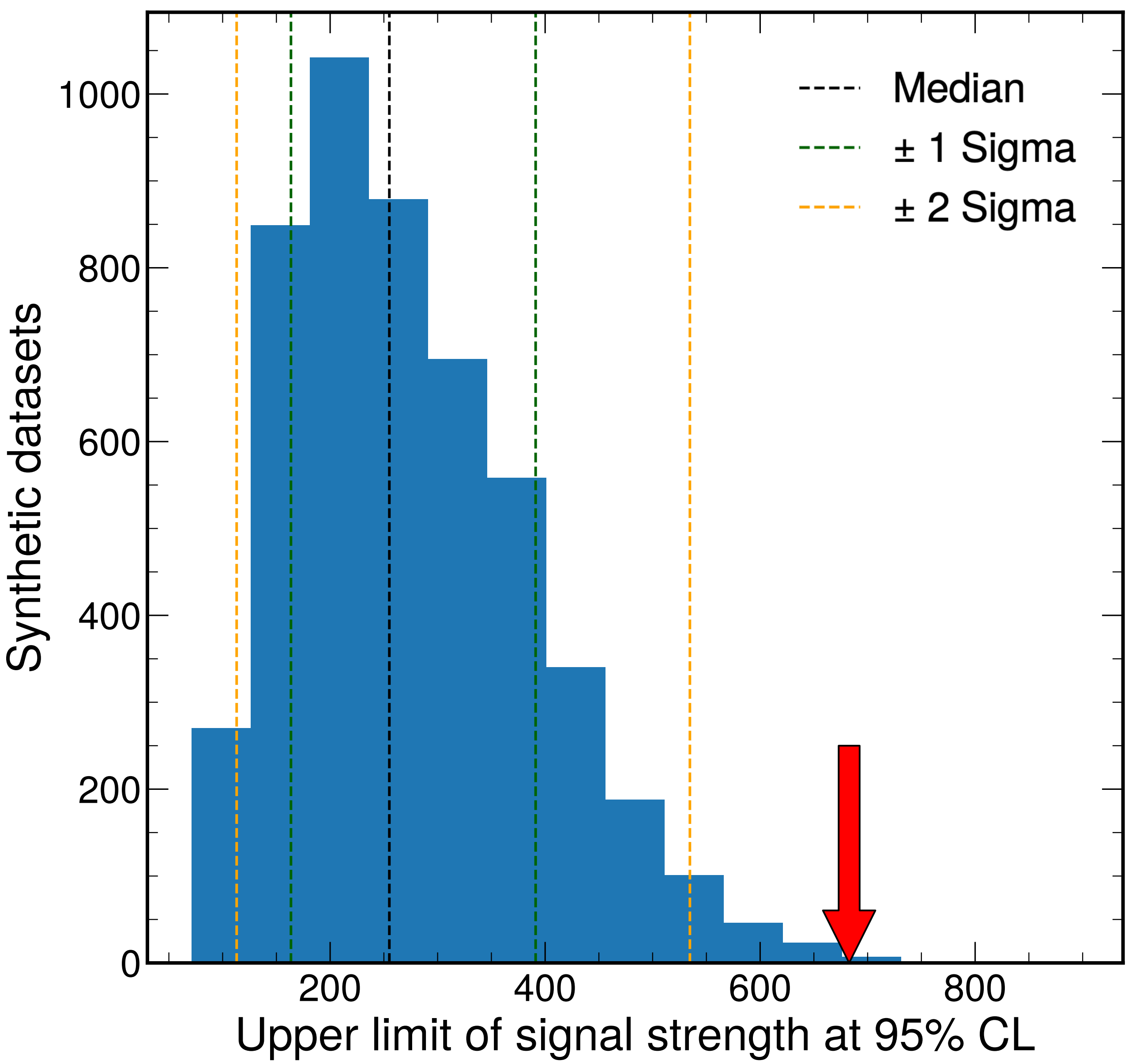


Observed data



Estimating signal significance

- We generate 5000 toy datasets by conditionally sampling from GP posterior
- Ran MCMC analysis on these toys
- Signal amplitude @ 95% CL from these toys gives us *sensitivity estimates*
- The same from *observed data* gives us the *significance* of the signal
- Results:
  - Observed signal strength: 485 ± 121
  - Significance: 3.15$\tilde{\sigma}$ or 99.84 percentile



Summary

- Non-parametric methods like GP can automate the background estimation
    - GP proves handy when fitting for smooth background distributions
    - Very relevant and essential for modeling data collected in RUN-3 and HL-LHC
- We provide a model selection framework for choosing GP covariance functions
- A method to extract localized signal parameters with minimal bias
- Prescription to estimate the sensitivity and the signal significance

For a more detailed information refer to: arxiv:2202.05856

Back-up slides

GP : Model selection

Fermilab

- We applied various kernels for modeling Bkg(x) in masked di-photon data

$$k_{\text{Poly2}}(x, x') = (\sigma_0^2 + x \cdot x')^2,$$

$$k_{\text{RBF}}(x, x') = \sigma_0 \exp\left[-\frac{(x - x')^2}{2l^2}\right]$$

$$k_{\text{Matern}}(x, x') = \sigma_0 \left[1 + \frac{\sqrt{5}}{l} d(x, x') + \frac{5}{3l} d(x, x')^2\right] \exp\left[-\frac{\sqrt{5}}{l} d(x, x')\right]$$

- $-\log p(y \mid X, K_i) \simeq -\log p(y \mid X, \hat{\theta}, K_i) + \frac{1}{2} \log |H| \equiv \text{BIC},$ H is the Hessian

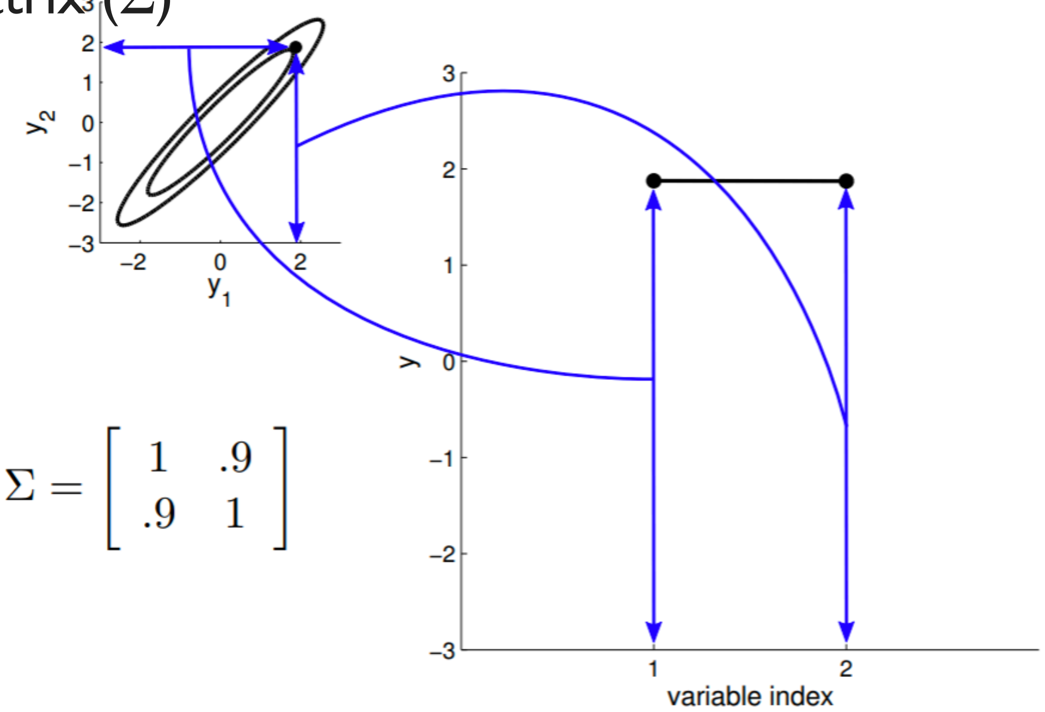
- $-\log p(y \mid X, K_i) \simeq -\log p(y \mid X, \hat{\theta}, K_i) + \frac{n}{2} \log N \equiv \text{BIC}^{\text{naive}},$ n is # parameters in model

N is # data points

Model	$\log \lvert H \rvert$	n	d	-log(PL)	-log(GL)	$\text{BIC}_{\text{GL}}^{\text{naive}}$	BIC_{GL}	AIC_{PL}	$\text{BIC}_{\text{PL}}^{\text{naive}}$
Poly2	-0.531	1	2.99	38.02	87.52	89.22	87.25	82.02	39.72
RBF	0.417	2	4.68	8.95	72.15	75.55	72.36	27.26	12.35
Matern	2.906	2	5.67	8.69	72.30	75.70	73.75	28.72	12.09
Func4	–	5	5	8.65	–	–	–	27.30	17.15

GP in a Nut shell

- At each bin X_i we have a bin content of $Y_i \in \mathcal{N}(\mu, \sigma)$ => (~ gaussian like errors)
- We can describe the correlation between the Y values using a matrix (Σ)
- In this 2 bin example both bins are very correlated.
  - The correlation structure of Y_1 and Y_2 is visualized as a 2D gaussian
  - All the randomly sampled points from this 2D gaussian show us the possible values of Y_i
  - By taking the weighted average, we can get mean and variance



$$\Sigma = \begin{bmatrix} 1 & .9 \\ .9 & 1 \end{bmatrix}$$

- GP is defined by a Mean function [m(x)] and a kernel matrix [K]
- In our case we have a higher bin count, we define this covariance matrix using a kernel
  - We factor in the noise (as each observation inherent error) by taking $\Sigma(x_i, x_j) = k(x_i, x_j) + I\sigma_y^2$
  - We do know the error on the each bin content, which is used in turn.
  - Using this Kernel, we can extrapolate prediction to any values of x