Bayesian Machine Learning for Scientific Research

Maximizing information from data

Ezequiel Alvarez
sequi@unsam.edu.ar
ICTP-SAIFR October 2024

Previous lectures

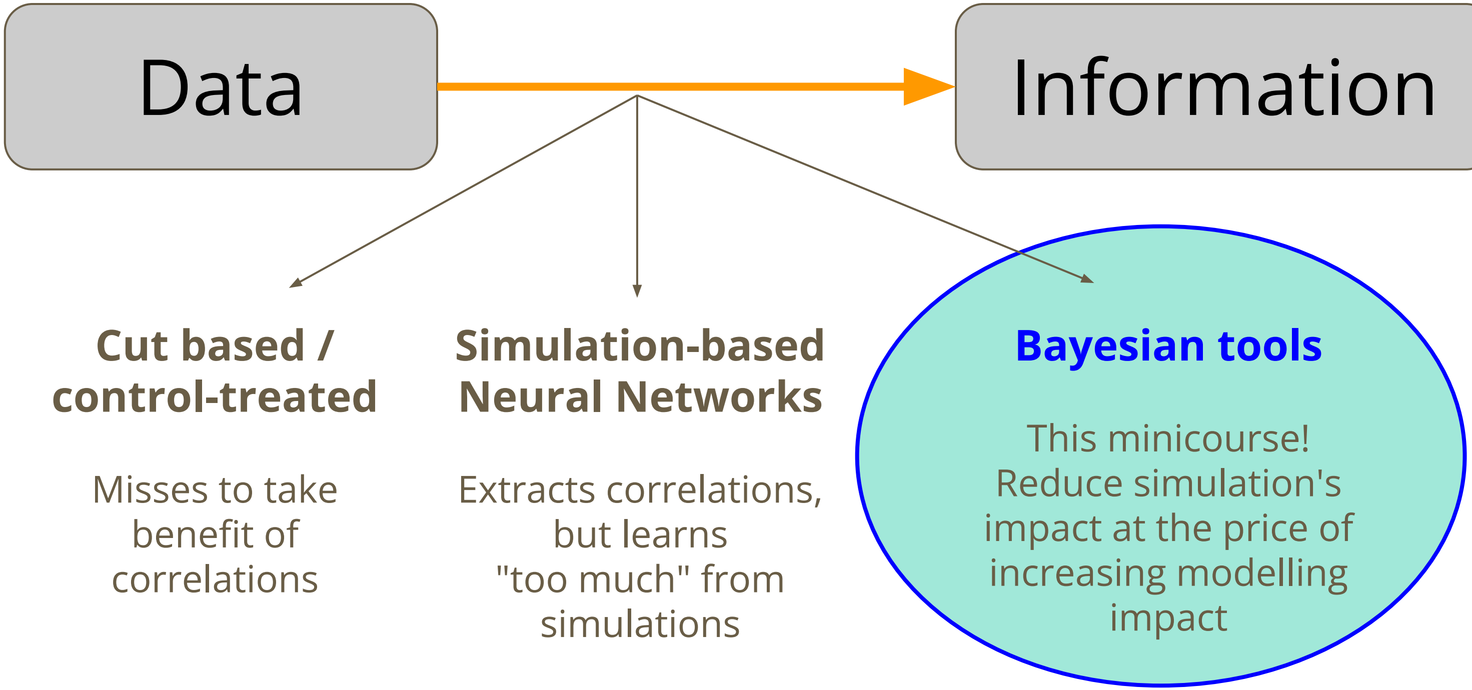
Lecture 1

Bayesian: assume data being sampled from a PDF, infer its parameters and learn the internal structure of the data

To learn the PDF of the data and then being able to assess, predict , generate, etc.

More scientific

Lecture 1

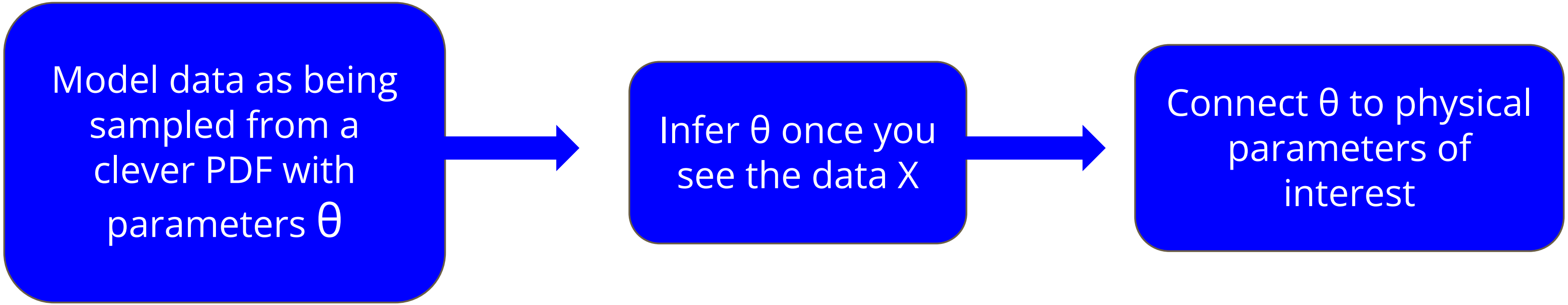

Data
Information
Cut based / control-treated
Misses to take benefit of correlations
Simulation-based Neural Networks
Extracts correlations, but learns "too much" from simulations
Bayesian tools
This minicourse!
Reduce simulation's impact at the price of increasing modelling impact

Lecture 1

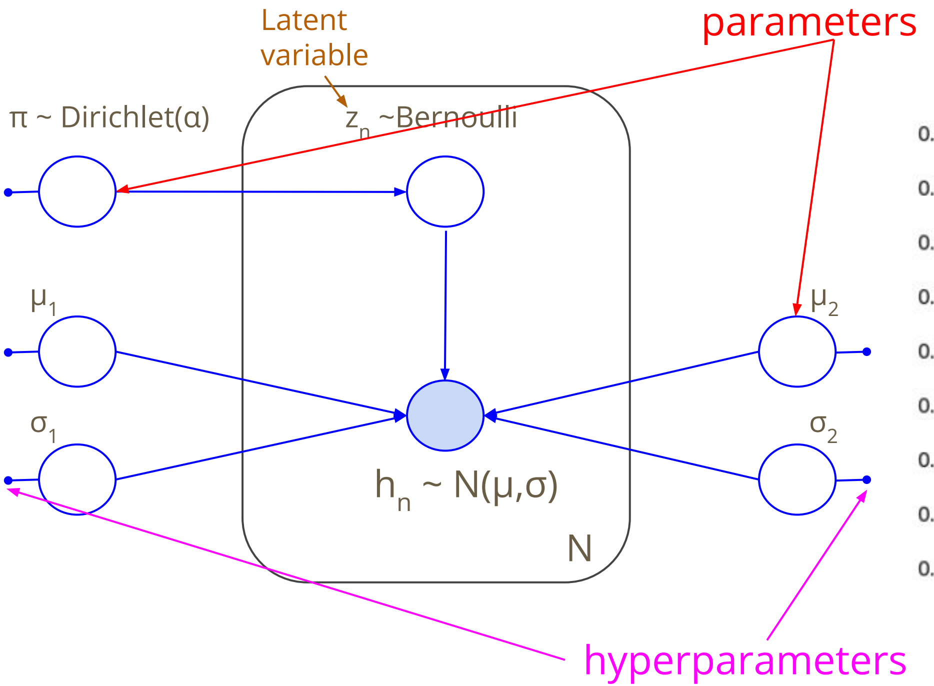
Bayes Theorem:

$$p(\theta | X) = \frac{p(X|\theta) * p(\theta)}{p(x)}$$

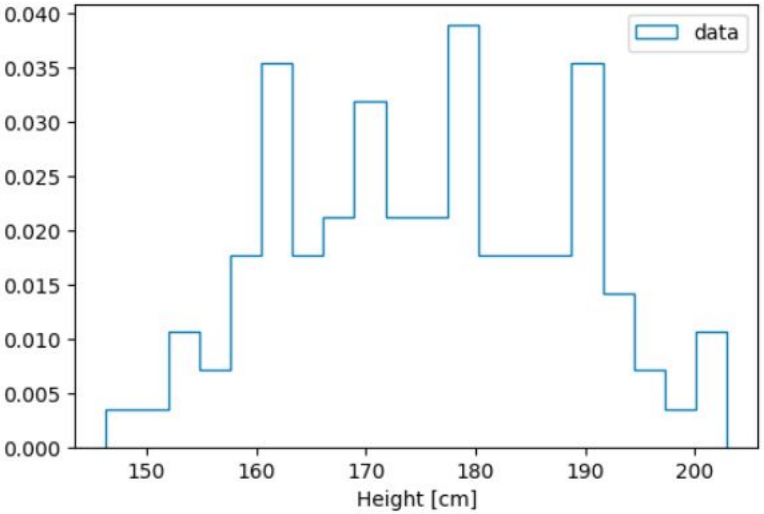
Our utility: X = data, θ=parameters



Lecture 1



Graphical Model



1D Gaussian Mixture

Lecture 2: The box factory

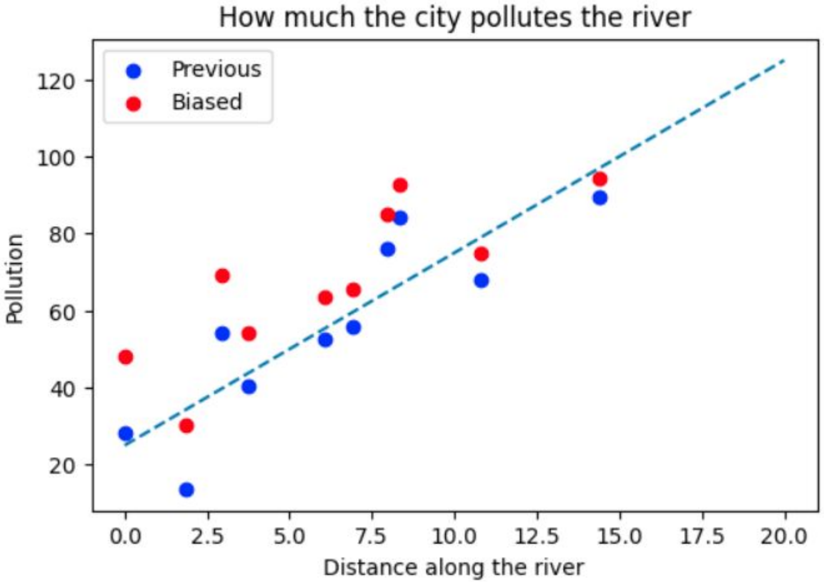
Box OK: *p*
Box wrong: *1-p*

Experiment: k boxes ok out of n

X = k, n
Θ = p

$$p(\theta|X) = \frac{1}{B(n'+n, k'+k)} p^{k'+k}(1-p)^{n'+n-k'-k}$$

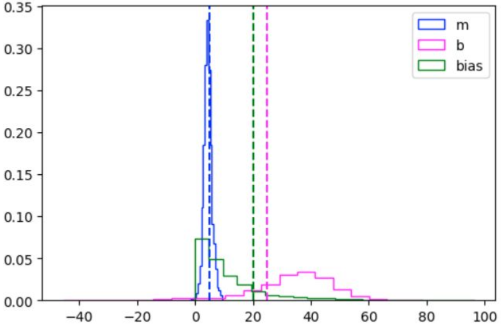
Lecture 2: Linear Regression

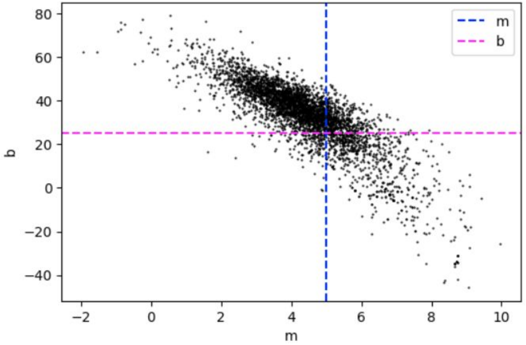


Expect more pollution from downtown

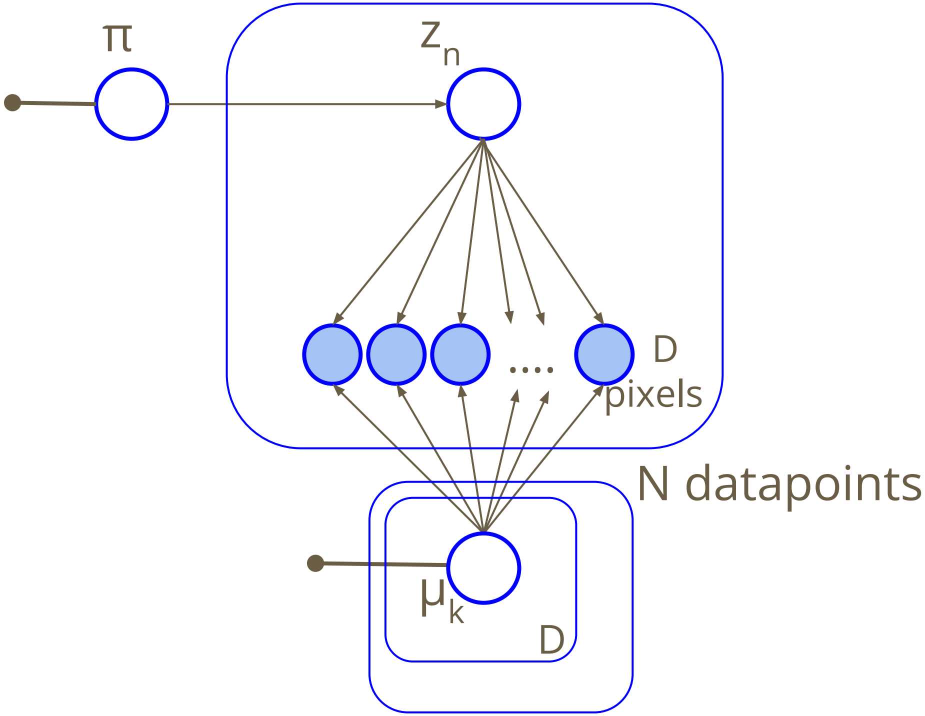
Lecture 2: Linear Regression

```
contamination_model_with_bias="""
data {
 int <lower=0> N;
 vector[N] x;
 vector[N] y;

}

parameters {
  real m;
  real b;
  real<lower=0> sigma;
  real<lower=0> bias;
}
model {
   bias ~ cauchy(0,10);  // we are using a half cauchy
                         // because bias >= 0
   y ~ normal(m * x + b + bias*exp(-x/10), sigma);
}
"""
```

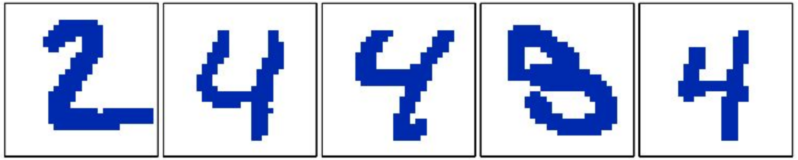


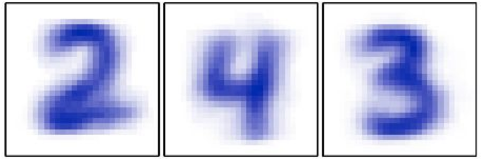


Lecture 2: Mixture of Bernoulli



Input





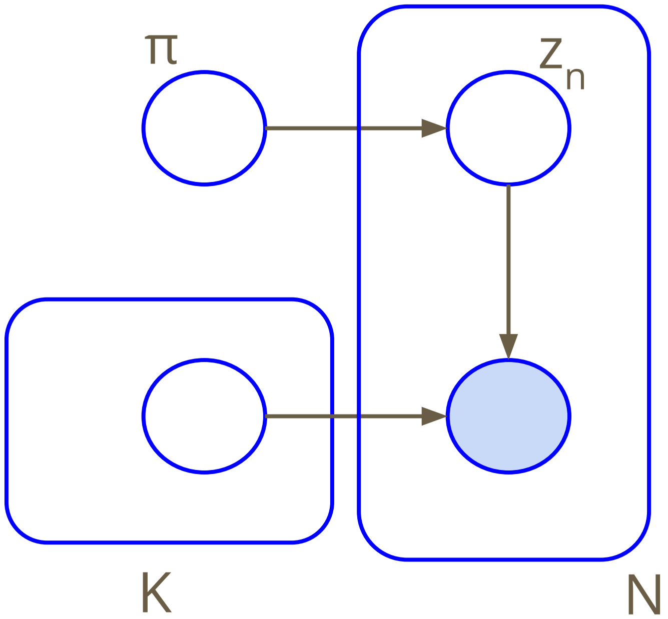
Result
(the μ's)

Scientifically: z_n is the probability of each class

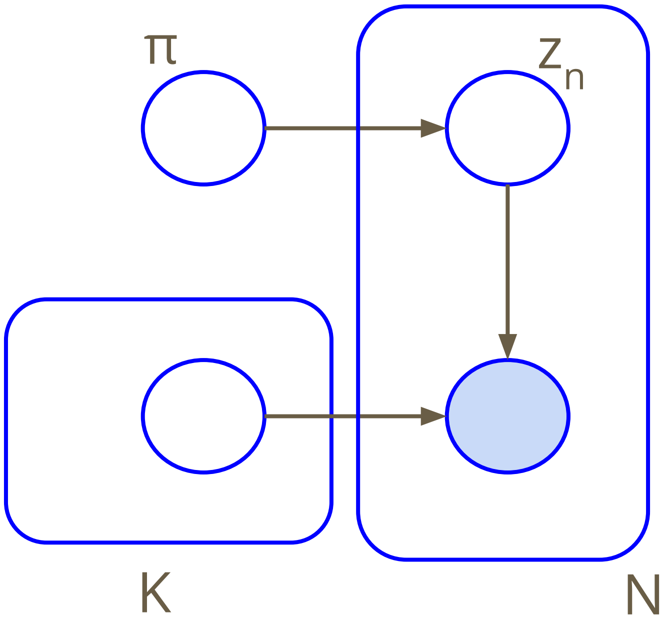
Mixture Models

Lecture 3

Mixture Models



Mixture Models



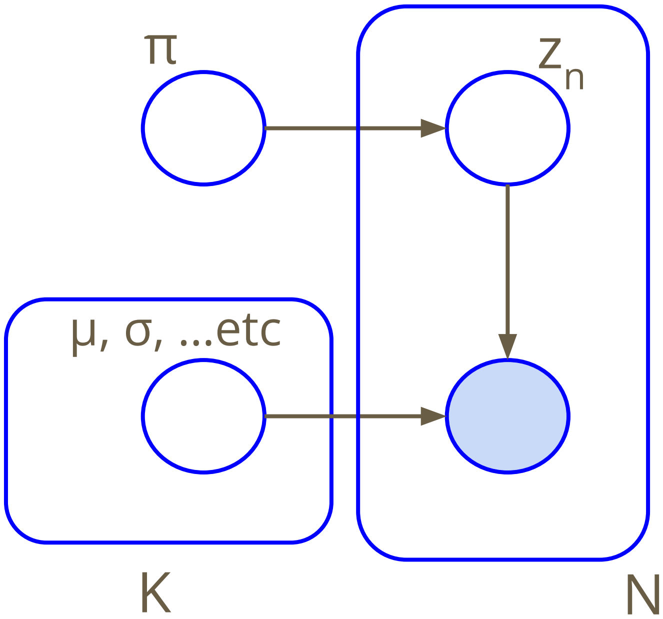
N : Number of datapoints

π : Mixture components

z_n : to which class belongs

K : Classes

Mixture Models



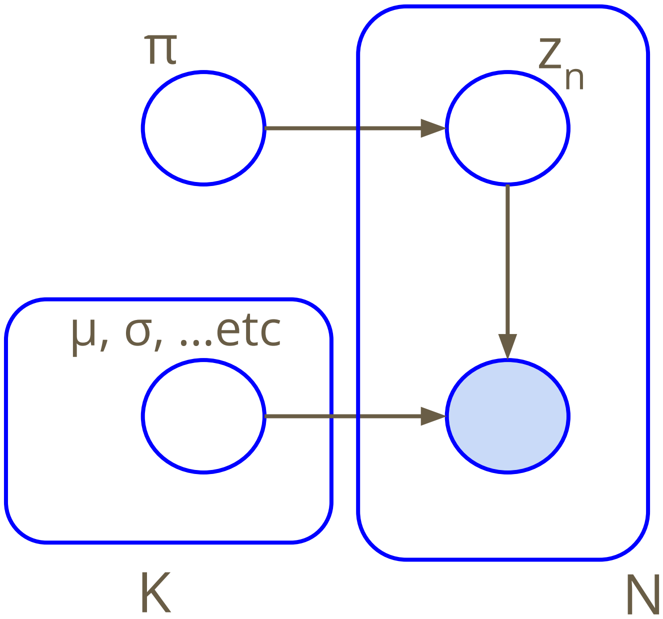
N : Number of datapoints

π : Mixture components

z_n : to which class belongs

K : Classes

Mixture Models



N : Number of datapoints

π : Mixture components

z_n : to which class belongs

K : Classes

Very complex data can be constructed from such a combination of simple basic PDFs

Mixture Models: Crypto price ?

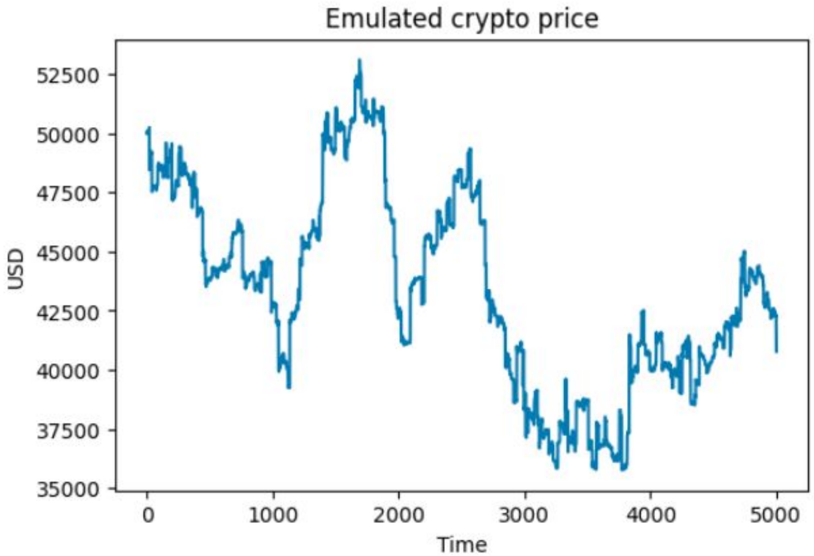
Real Bitcoin price Aug to Sep

Mixture Models: Crypto price ?

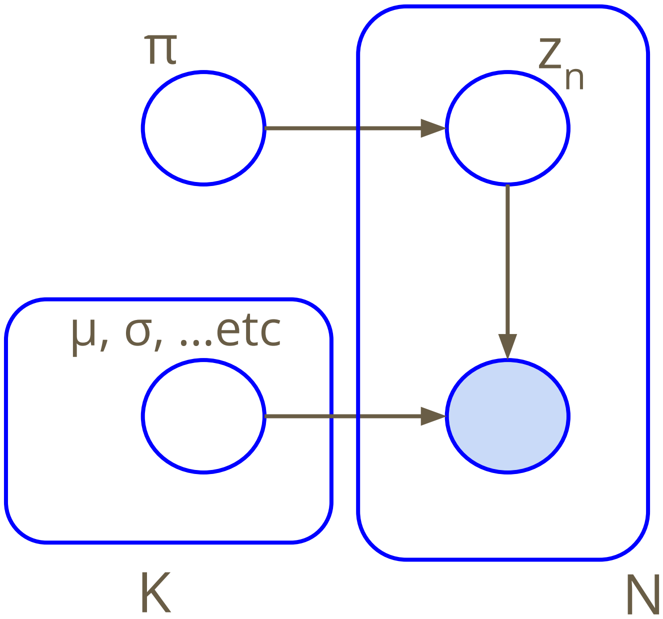
Real Bitcoin price Aug to Sep

```python
sigmas = [10,100,1000]
crypto_price = [50000]
np.random.seed(1)
z = np.random.choice([0,1,2], p=[0.7,0.25,0.05], size=5000)
for zn in z:
    crypto_price.append(cripto_price[-1]+np.random.normal(0,sigmas[zn]))
```

```python
plt.plot(cripto_price)
plt.title('Emulated crypto price')
plt.ylabel('USD')
plt.xlabel('Time')
plt.show()
```



Mixture Models



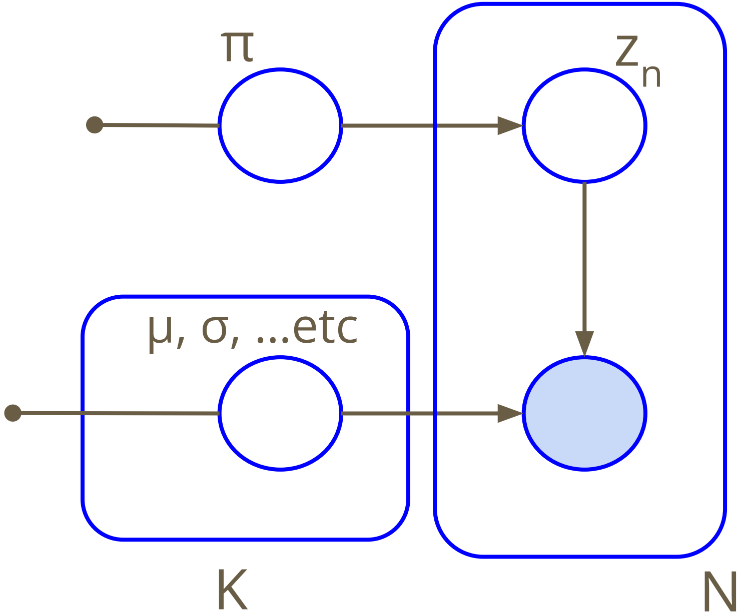
N : Number of datapoints

π : Mixture components

z_n : to which class belongs

K : Classes

Mixture Models



N : Number of datapoints

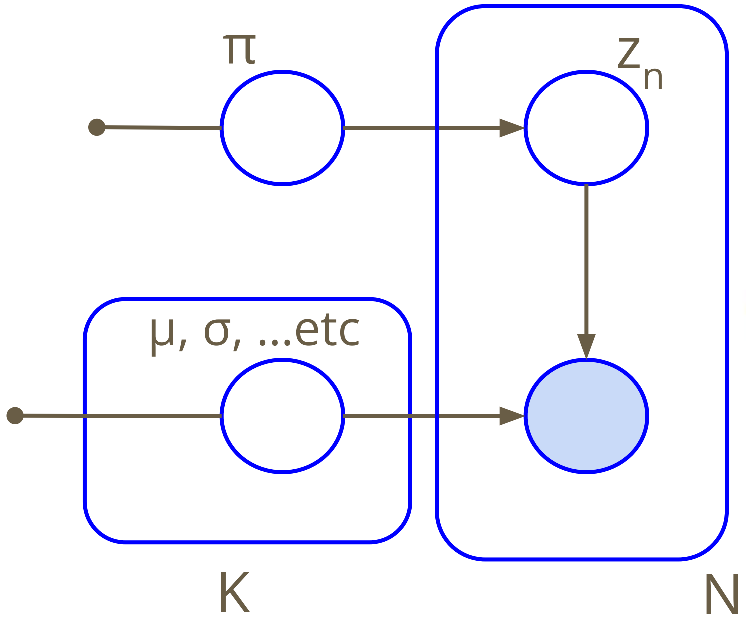
π : Mixture components

z_n : to which class belongs

K : Classes

Hyperparameters:
Input prior info

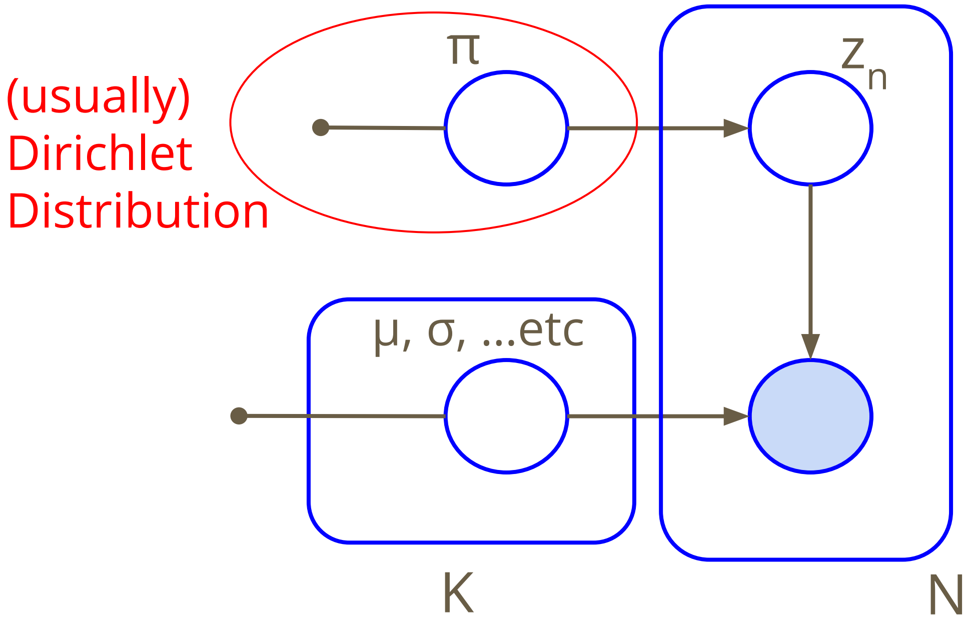
Mixture Models



$$p(x_n|\Theta) = \sum_{k=1}^{K} \pi_k \, p(x_n|\theta_k)$$

$$\log p(X|\Theta) = \sum_{n=1}^{N} \log(\sum_{k=1}^{K} \pi_k \, p(x|\theta_k) \,)$$

Mixture Models



Dirichlet Distribution

Samples D positive numbers that add to 1:

E.g. for D=5 : X = [0.15, 0.18, 0.37, 0.07, 0.23]

Dirichlet Distribution

Samples D positive numbers that add to 1:

E.g. for D=5 : X = [0.15, 0.18, 0.37, 0.07, 0.23]

With a probability

p(X) = *Dir*(X; α)

Dirichlet Distribution

Samples D positive numbers that add to 1:

E.g. for D=5 : X = [0.15, 0.18, 0.37, 0.07, 0.23]

With a probability

p(X) = *Dir*(X; α)

A Dirichlet distribution
samples Multinomial distributions!

Dirichlet Distribution

Samples D positive numbers that add to 1:

E.g. for D=5 : X = [0.15, 0.18, 0.37, 0.07, 0.23]

With a probability

p(X) = *Dir*(X; α) = $\frac{1}{\mathrm{B}(\boldsymbol{\alpha})} \prod_{i=1}^{K} x_i^{\alpha_i - 1}$

A Dirichlet distribution
samples Multinomial distributions!

Dirichlet Distribution

Samples D positive numbers that add to 1:

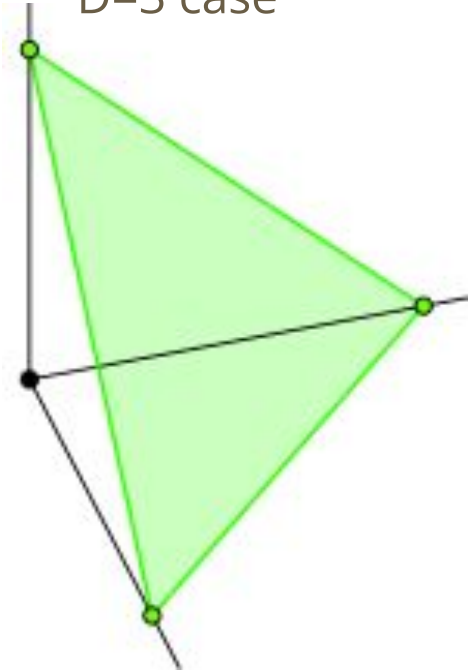
E.g. for D=5 : X = [0.15, 0.18, 0.37, 0.07, 0.23]

With a probability

p(X) = *Dir*(X; α) = $\frac{1}{\mathrm{B}(\boldsymbol{\alpha})} \prod_{i=1}^{K} x_i^{\alpha_i - 1}$

A Dirichlet distribution
samples Multinomial distributions!

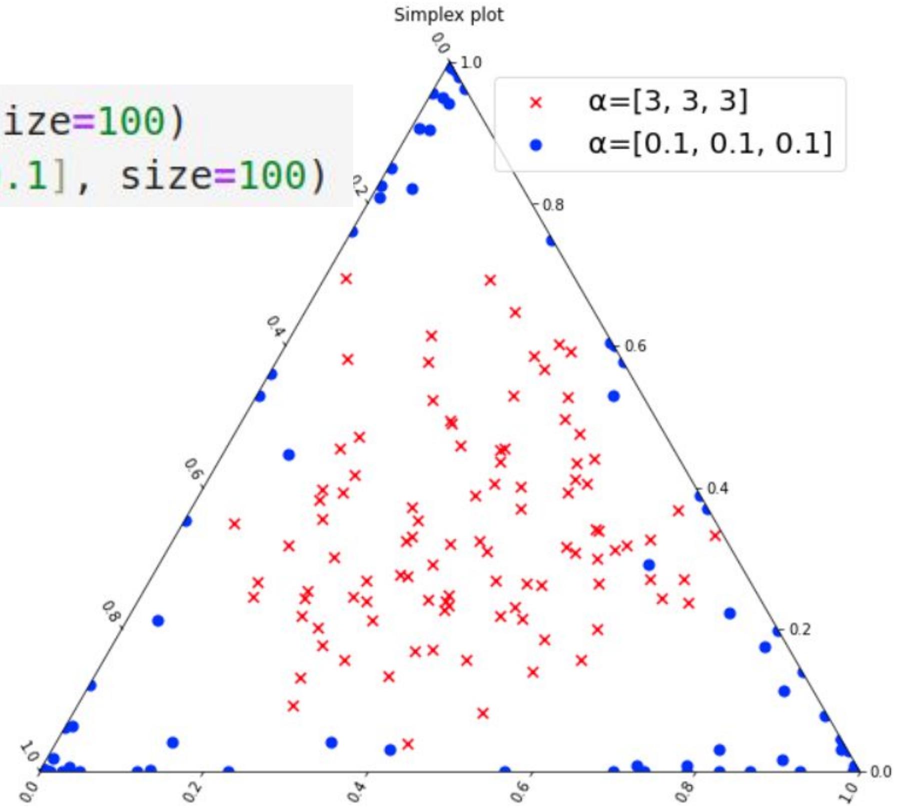
We can visualize the
D=3 case



Dirichlet Distribution

```
sample1 = np.random.dirichlet([3,3,3], size=100)
sample2 = np.random.dirichlet([0.1,0.1,0.1], size=100)
```

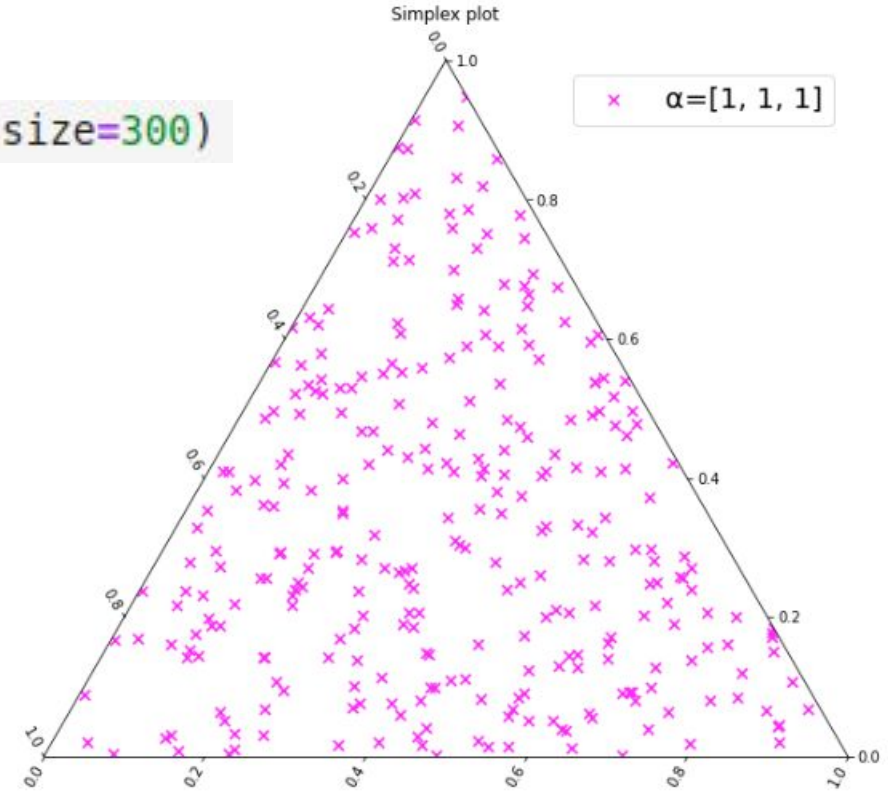
$$p(X) = Dir(X;\alpha) = \frac{1}{B(\boldsymbol{\alpha})}\prod_{i=1}^{K} x_i^{\alpha_i - 1}$$



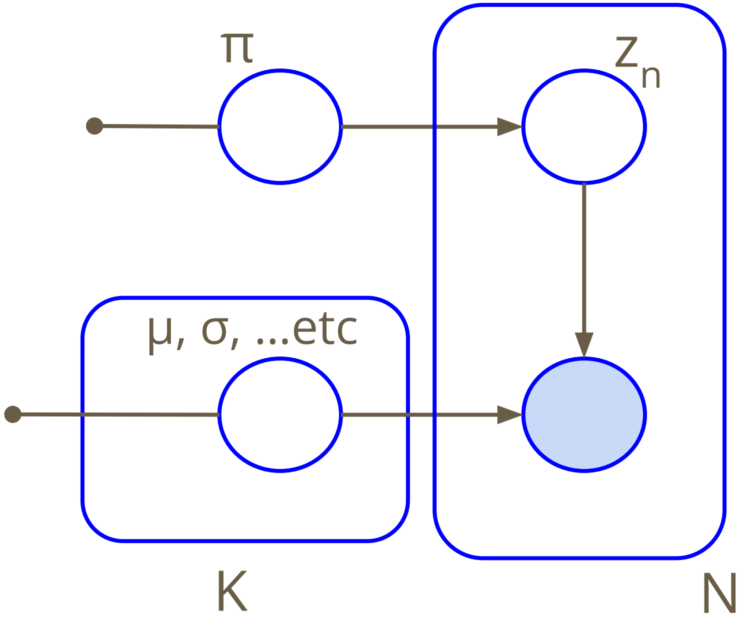
Dirichlet Distribution

```python
sample0 = np.random.dirichlet([1,1,1], size=300)
```

p(X) = *Dir*(X; α) = $\frac{1}{\mathrm{B}(\boldsymbol{\alpha})}\prod_{i=1}^{K} x_i^{\alpha_i - 1}$



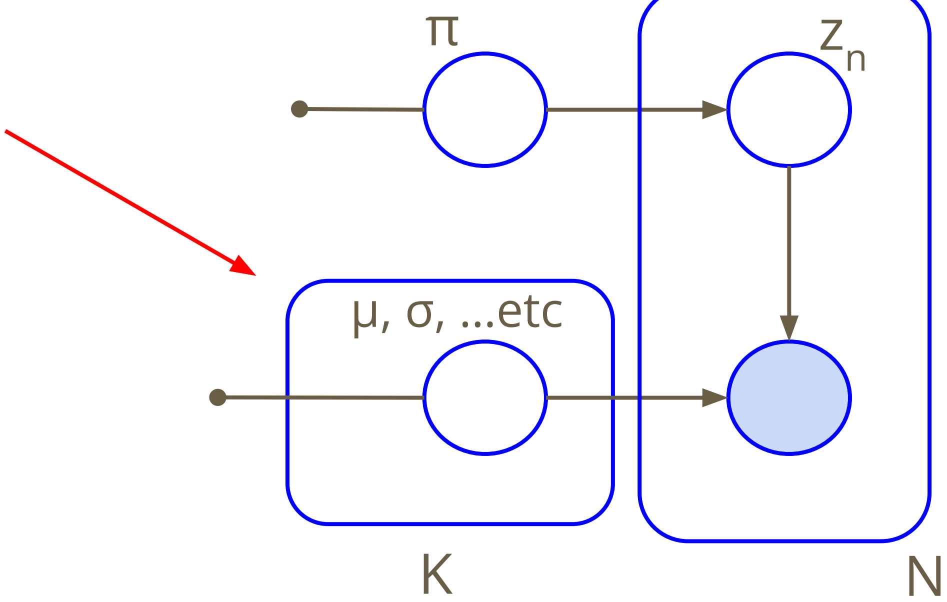
Mixture Models: where is the hack ?



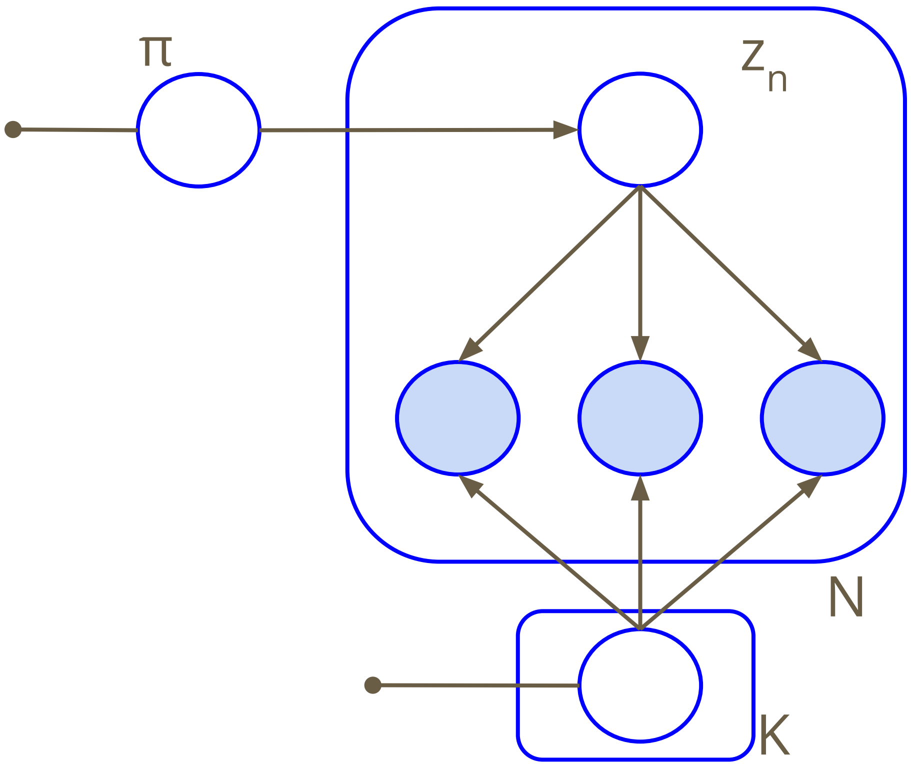
Mixture Models: where is the hack ?

Prior info:
restrict mixture to well defined family of curves

(AKA Template fit)

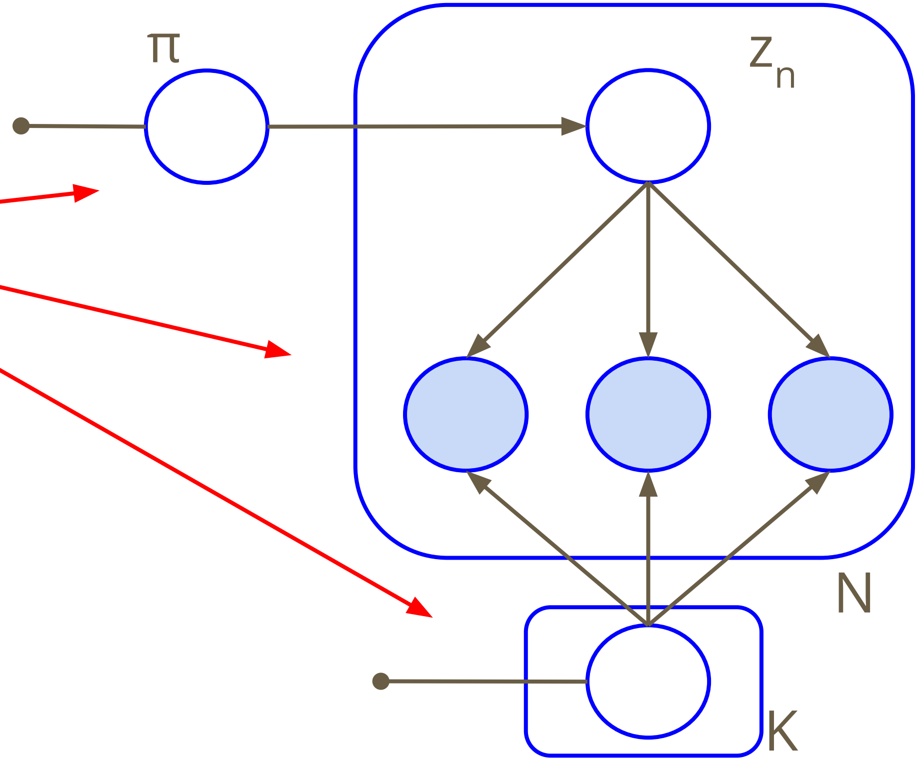


Mixture Models: where is the hack ?



Mixture Models: where is the hack ?

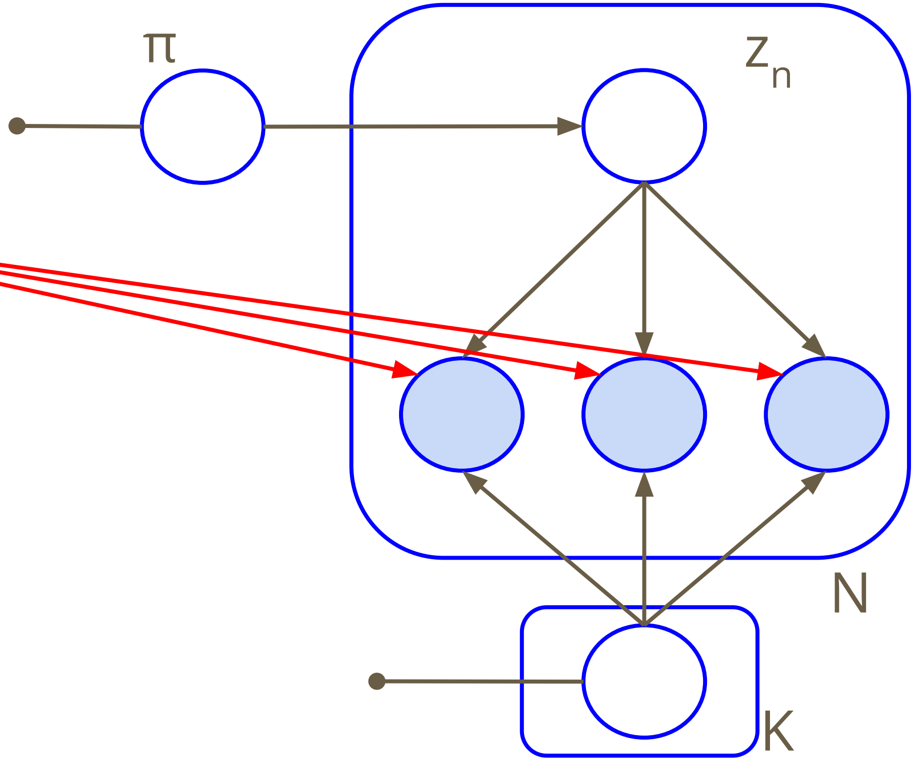
Exploit correlations at each datapoint simultaneously with all prior info!



Mixture Models: where is the hack ?

Conditionally independence

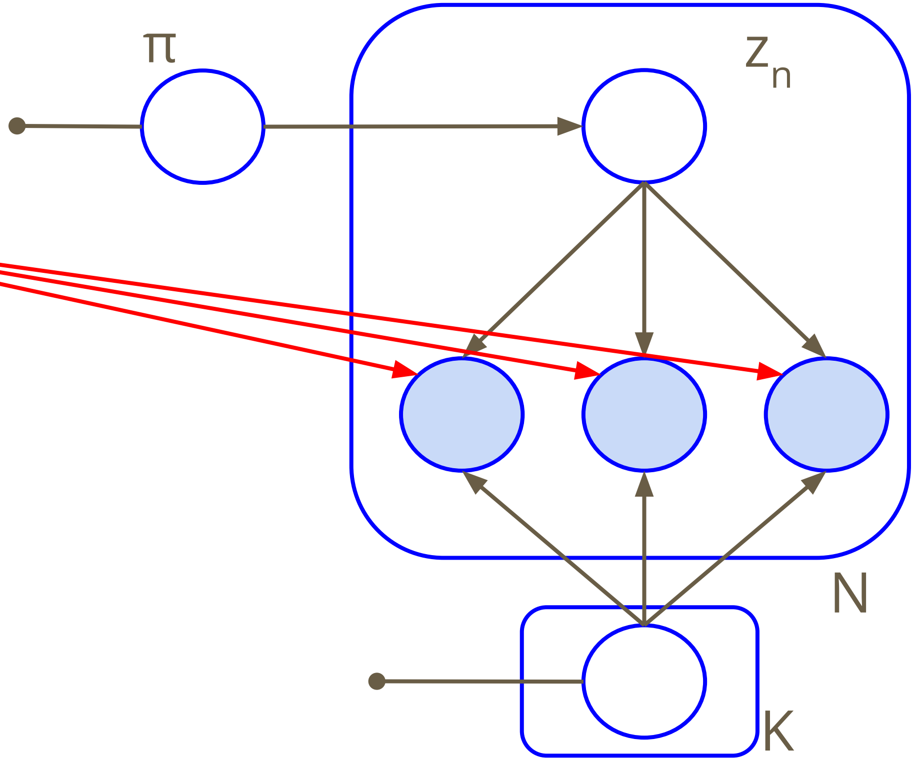
Once the class is defined,
they are independent



Mixture Models: where is the hack ?

Conditionally independence

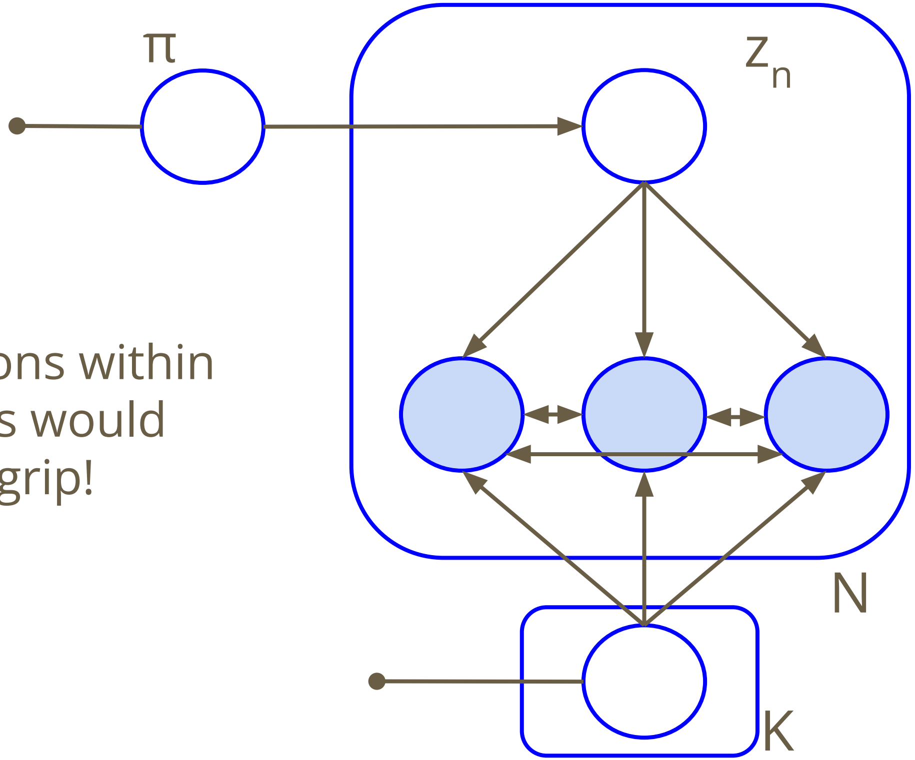
Once the class is defined,
they are independent



Huge reduction in number of parameters to infer!

better grip!
($bins^D$ vs bins*D)

Mixture Models: where is the hack ?



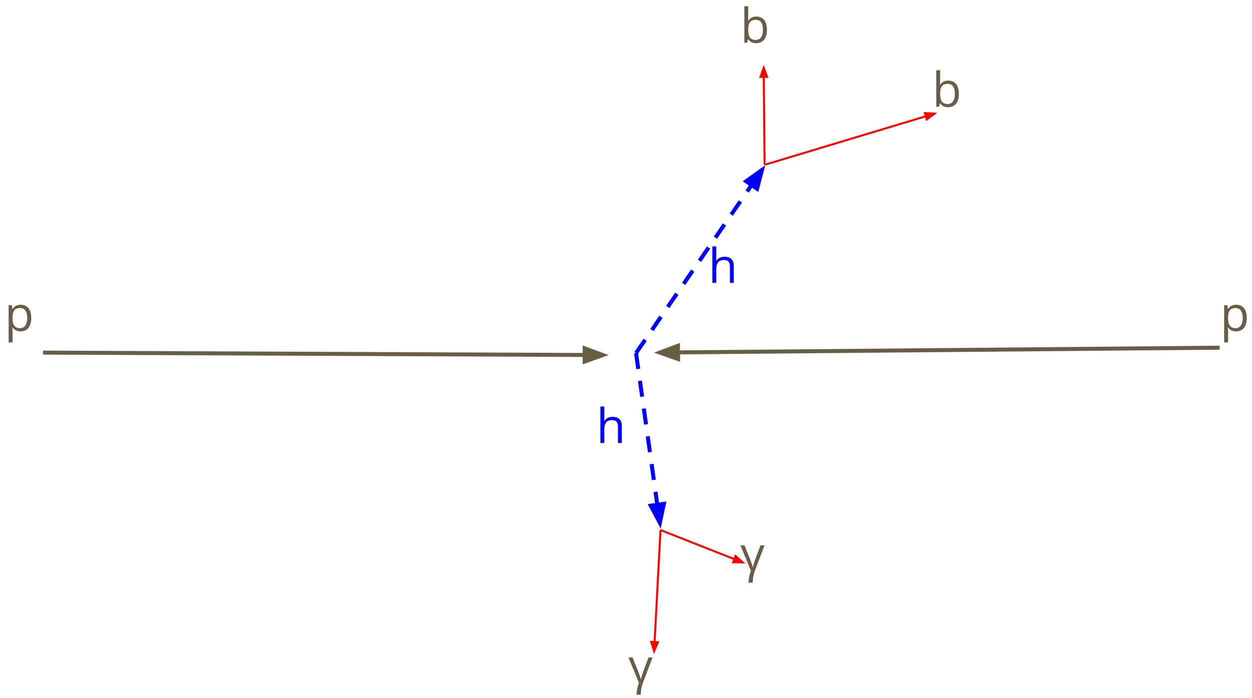
Correlations within each class would spoil the grip!

2D Mixture Model

di-Higgs production hh → bbγγ

(also hh → bbbb)

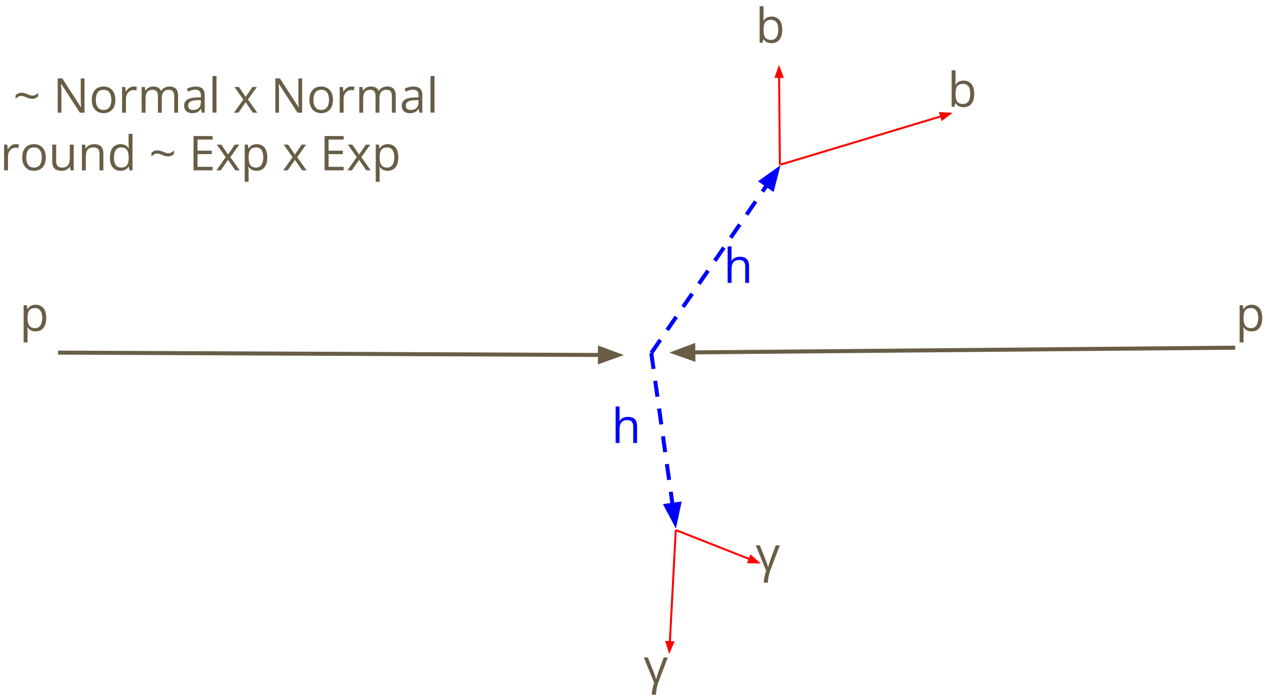
2D Mixture Model: $hh \longrightarrow bb\gamma\gamma$



2D Mixture Model: hh → bbγγ

Invariant masses:

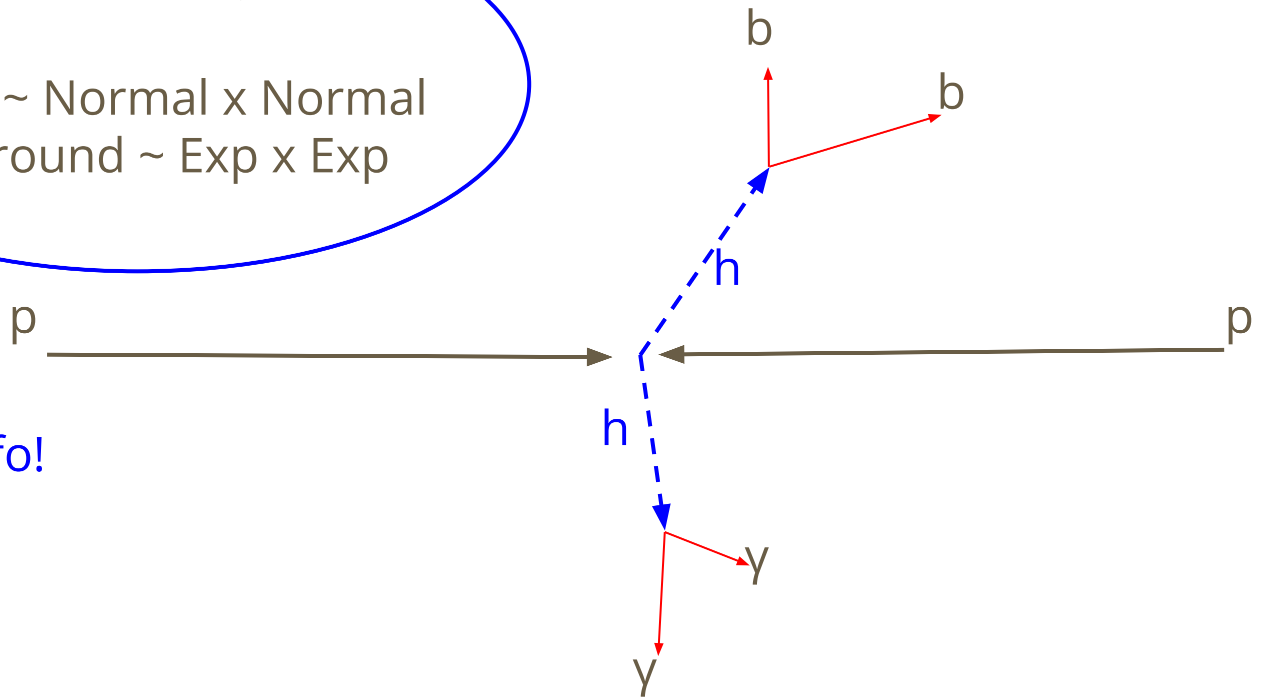
Signal ~ Normal x Normal
Background ~ Exp x Exp



2D Mixture Model: hh → bbγγ

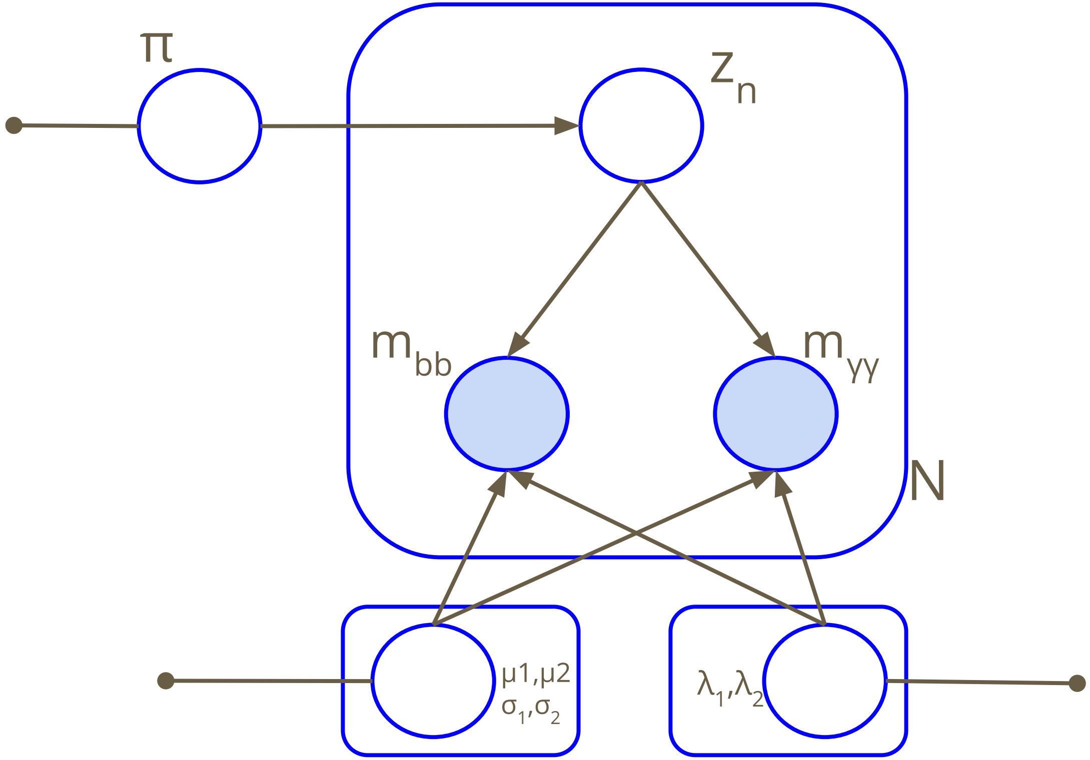
Invariant masses:

Signal ~ Normal x Normal
Background ~ Exp x Exp

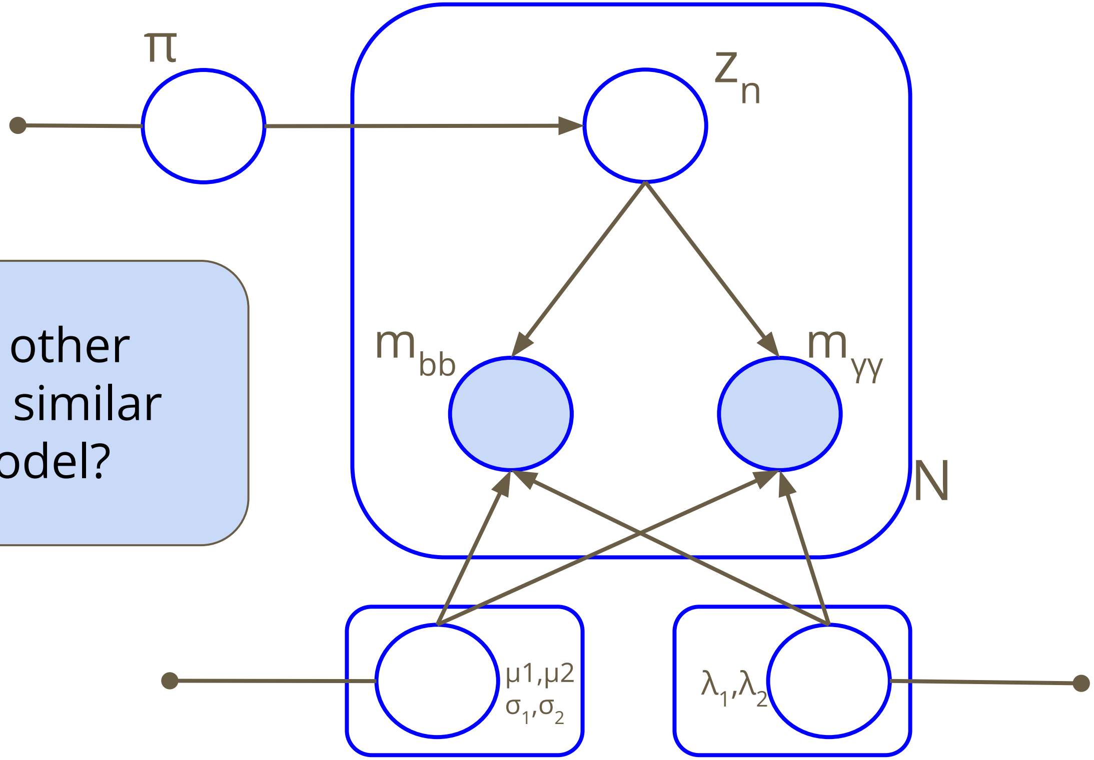


Prior info!

2D Mixture Model: hh → bbγγ



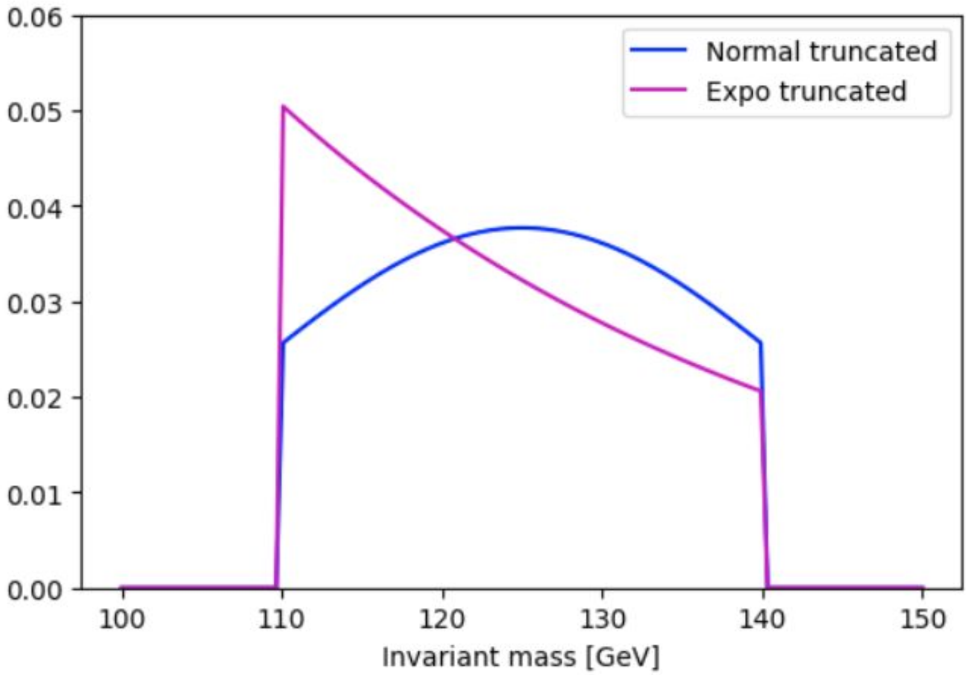
2D Mixture Model: $hh \to bb\gamma\gamma$



Let's imagine other systems with a similar Graphical Model?

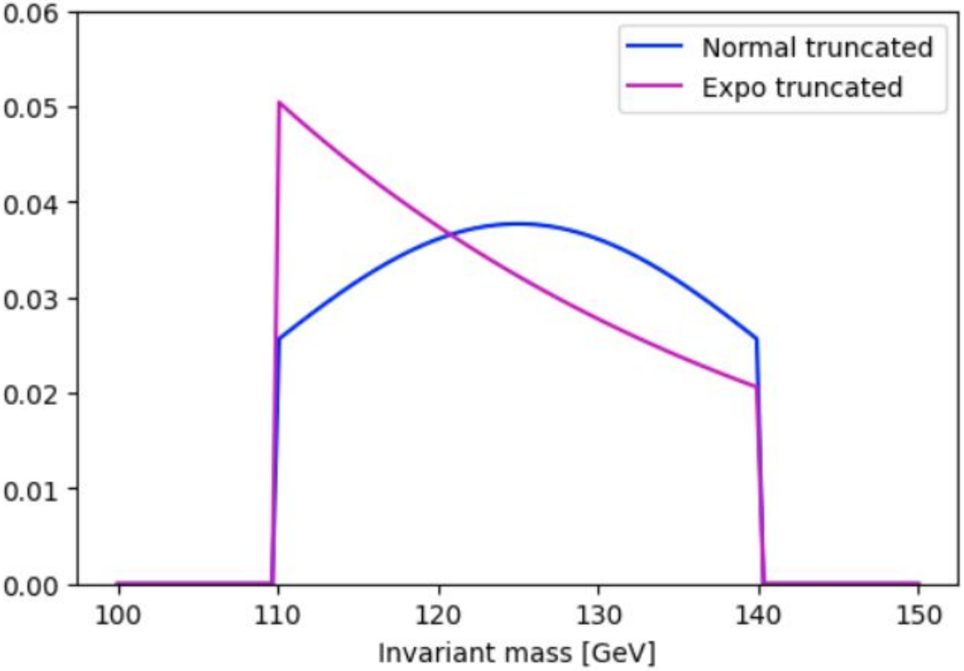
2D Mixture Model: hh → bbγγ

Truncated distributions!



2D Mixture Model: hh → bbγγ

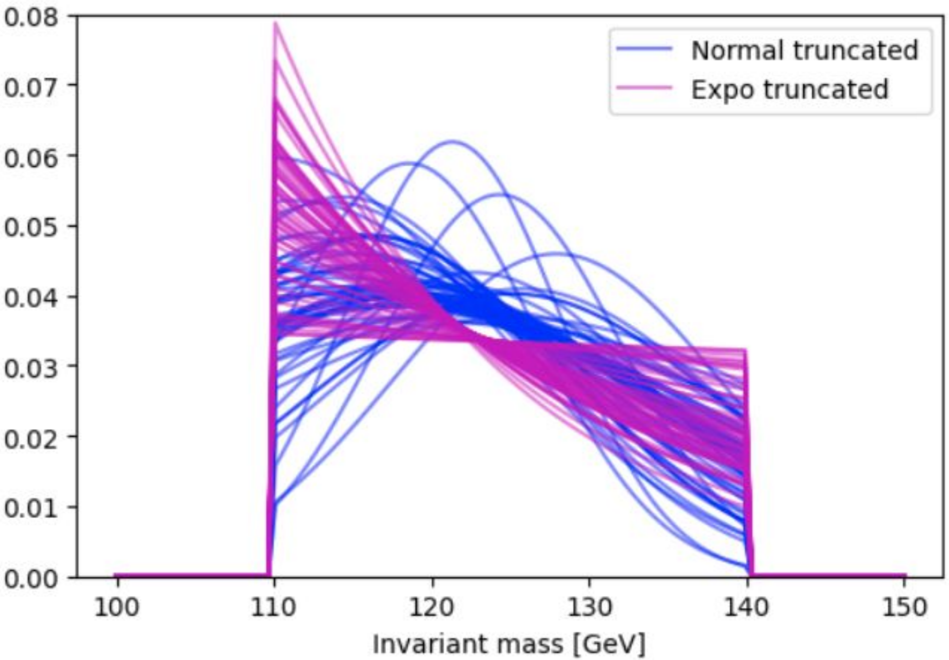
Truncated distributions!



Hey…. but they are quite different!

2D Mixture Model: hh → bbγγ

Truncated distributions!

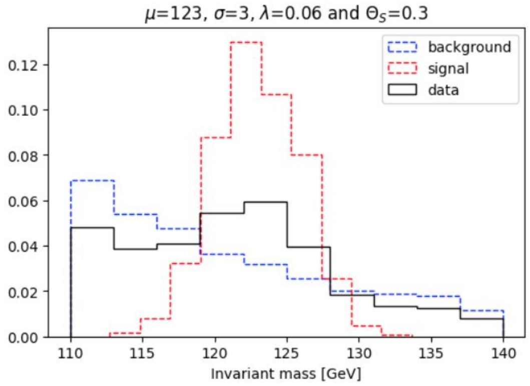


Hey…. but they are quite different!

Difficult to disentangle this mixture if parameters are unknown!

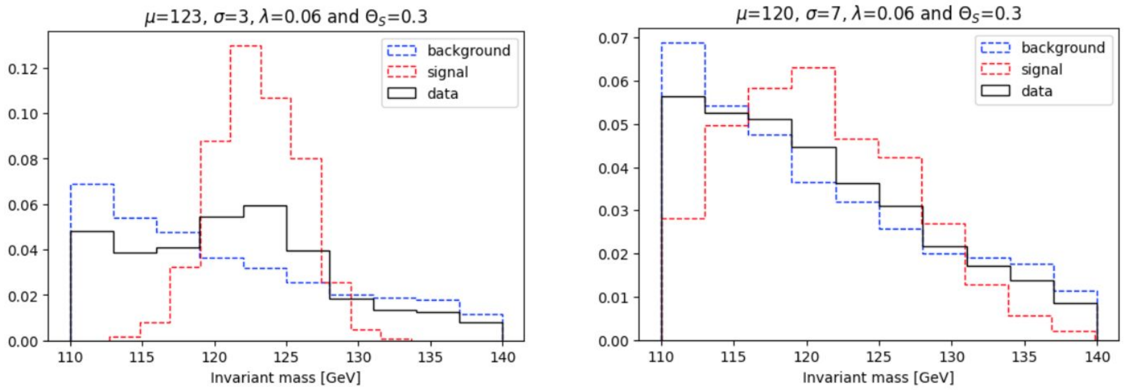
2D Mixture Model: hh → bbγγ

Too evident



2D Mixture Model: hh → bbγγ

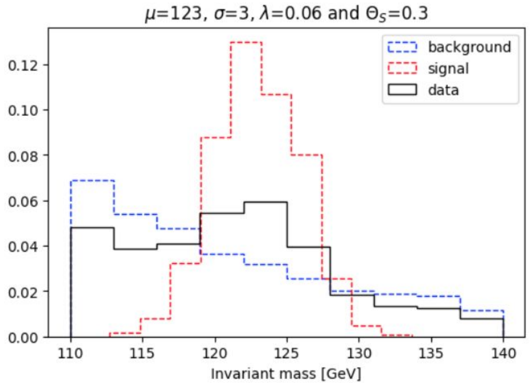
Too evident

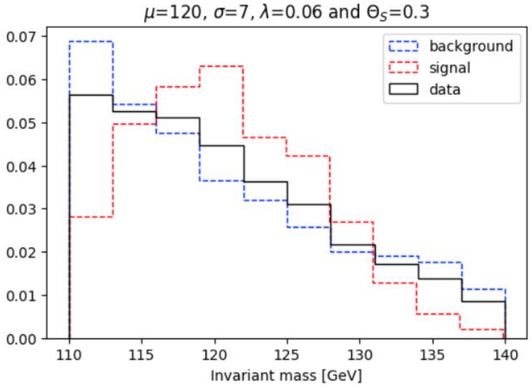


Mmm this is harder...

2D Mixture Model: hh → bbγγ

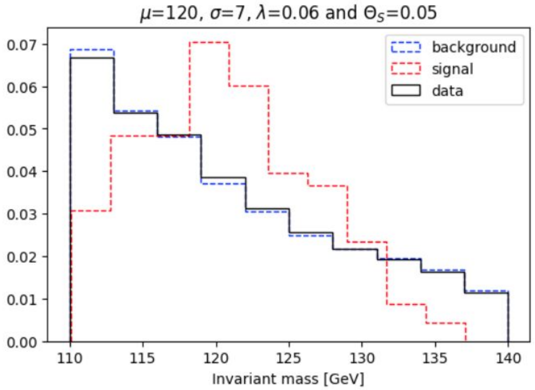
Too evident





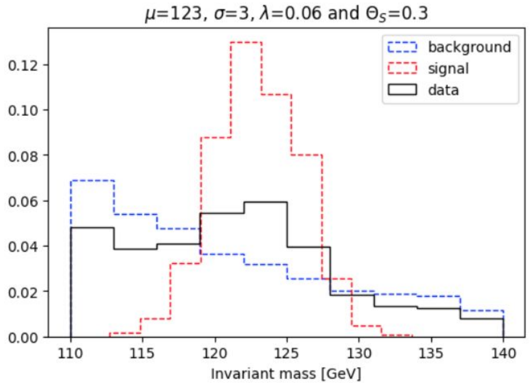
Mmm this is harder...

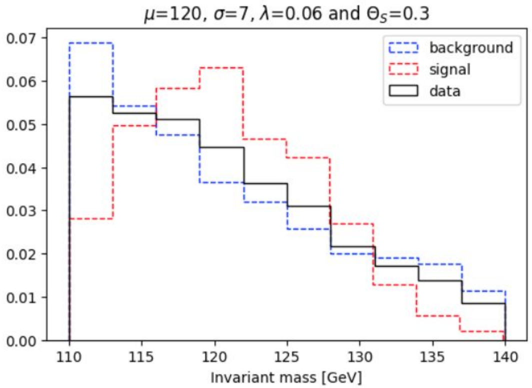
Bahh come on.. You won't get this!



2D Mixture Model: hh → bbγγ

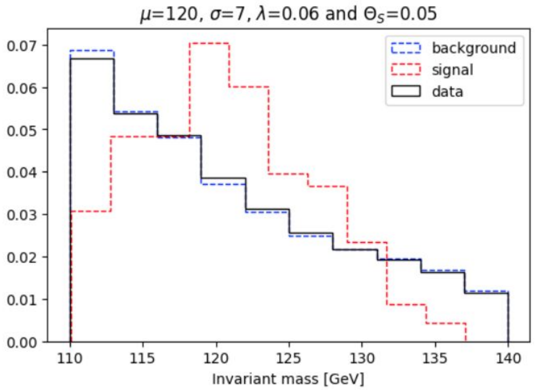
Too evident





Mmm this is harder...

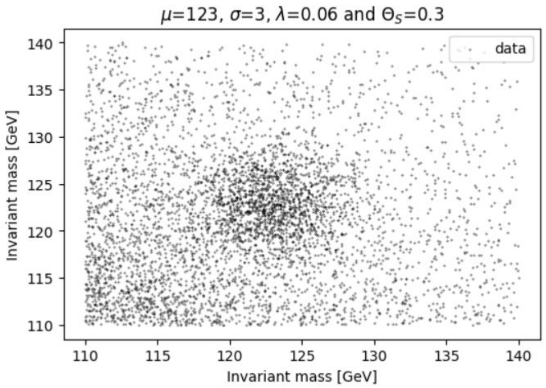
Bahh come on.. You won't get this!

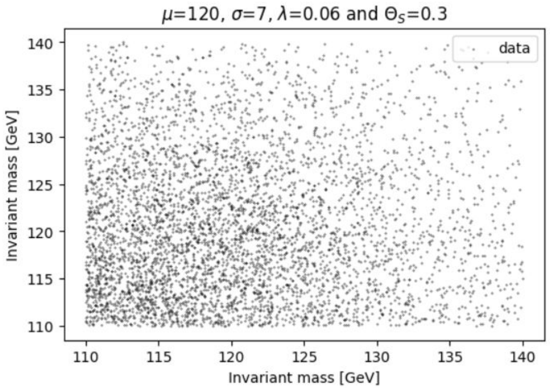


The hack is to think multi-dimensional!

2D Mixture Model: hh → bbγγ (bbbb)

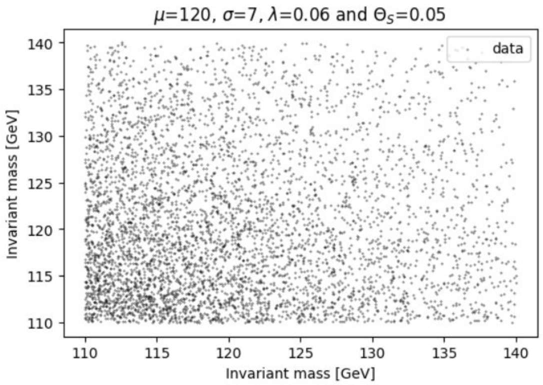
Too evident





Mmm this is harder...

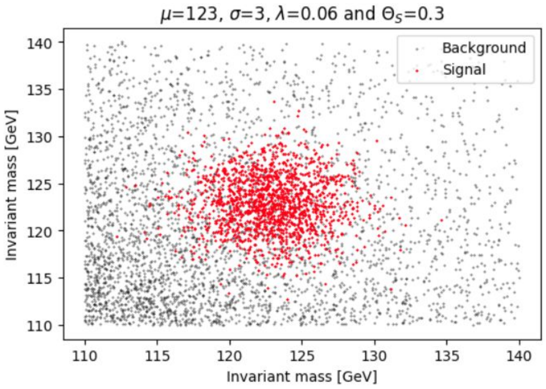
Bahh come on.. You won't get this!

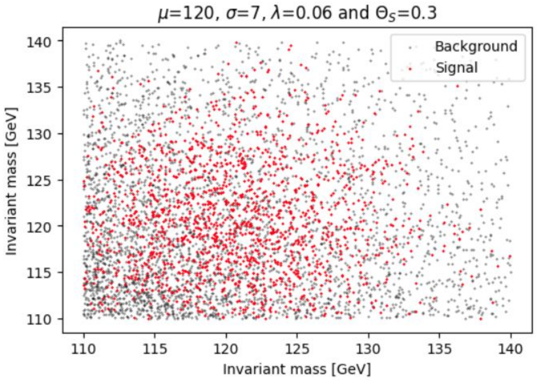


The hack is to think multi-dimensional!

2D Mixture Model: hh → bbγγ (bbbb)

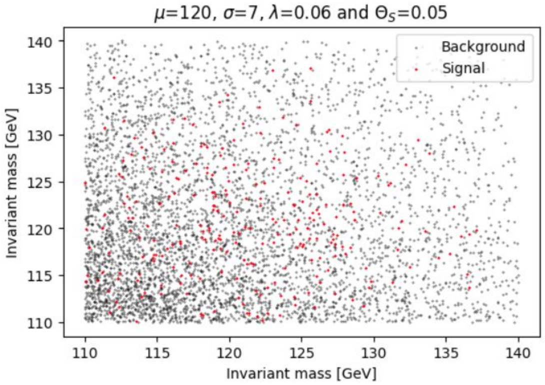
Too evident





Mmm this is harder...

Bahh come on.. You won't get this!



The hack is to think multi-dimensional!

2D Mixture Model: hh → bbγγ (bbbb)

Stan
mixture model
of ***truncated***
distributions

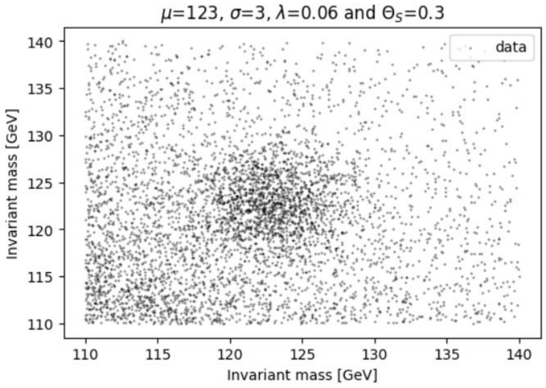
N x N

mixed with

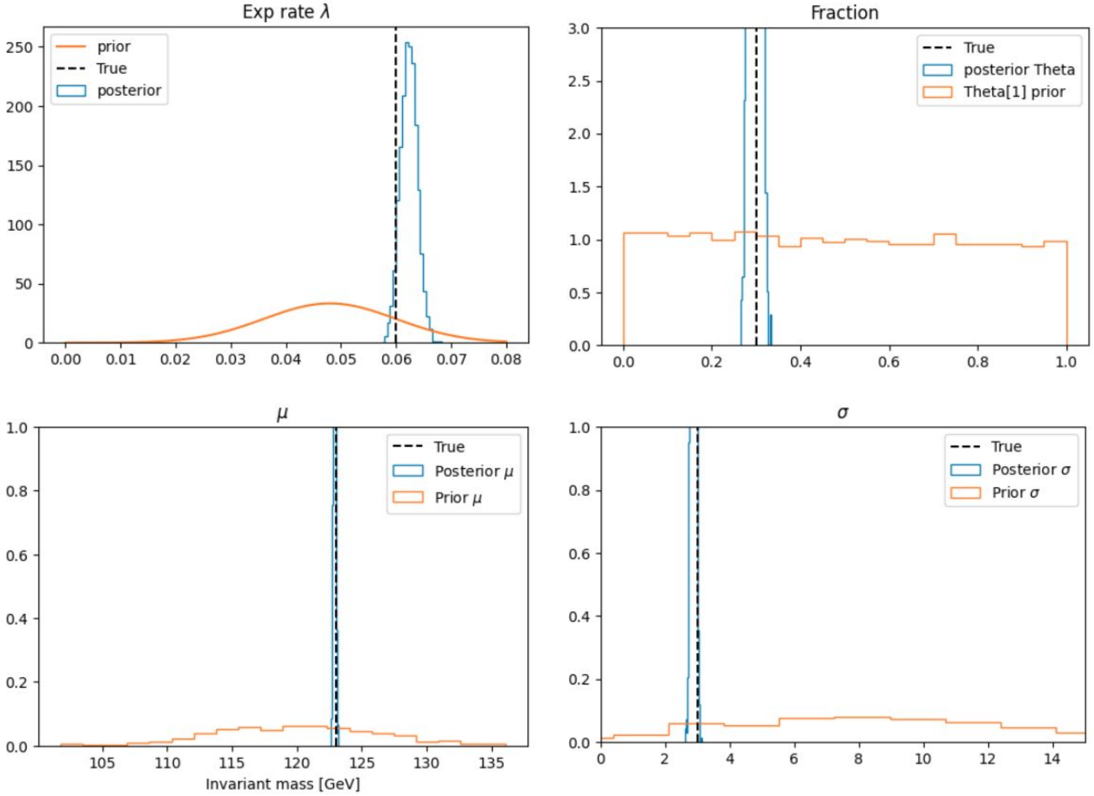
Exp x Exp

```
model {
    vector[2] lp;
    mu ~ normal(mu0, sigma0);
    sigma ~ cauchy(mu1, sigma1);
    lambda0 ~ normal(mu2, sigma2);
    theta ~ dirichlet([t1,t2]);

   for (n in 1:N) {
     lp[1] =   normal_lpdf(y[n,1] | mu, sigma)
             - log_diff_exp( normal_lcdf( U | mu, sigma), normal_lcdf( L | mu, sigma) )
             + normal_lpdf(y[n,2] | mu, sigma)
             - log_diff_exp( normal_lcdf( U | mu, sigma), normal_lcdf( L | mu, sigma) );
     lp[2] =   exponential_lpdf( y[n,1] - L | lambda0)
             - exponential_lcdf( U - L | lambda0)
             + exponential_lpdf( y[n,2] - L| lambda0)
             - exponential_lcdf( U - L | lambda0);
     target += log_mix(theta, lp);
     };
}
```

Simple case

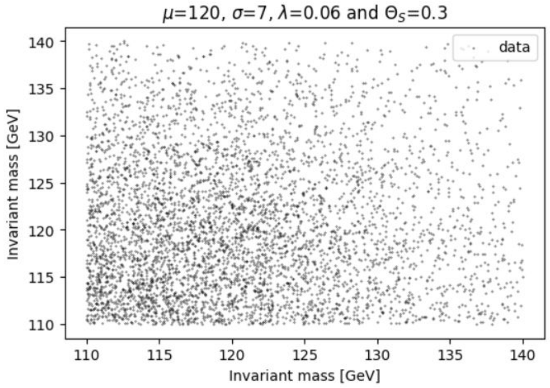


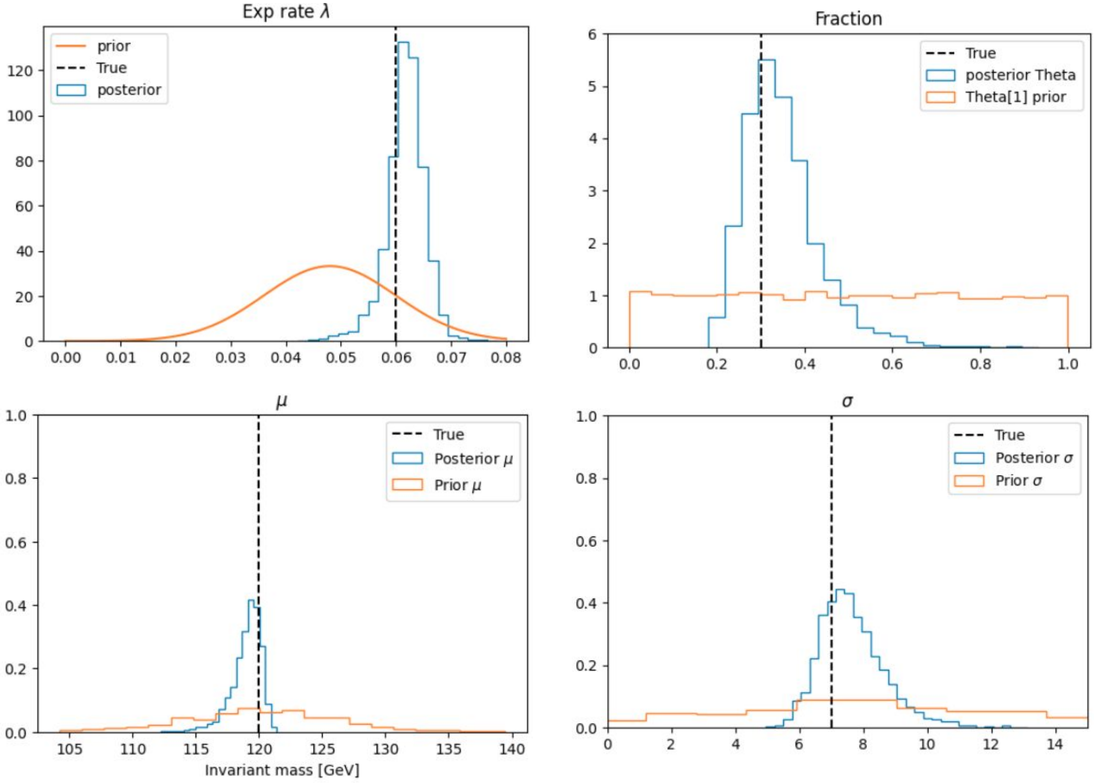
μ=123, σ=3, λ=0.06 and Θ_S=0.3
data
Invariant mass [GeV]
Invariant mass [GeV]



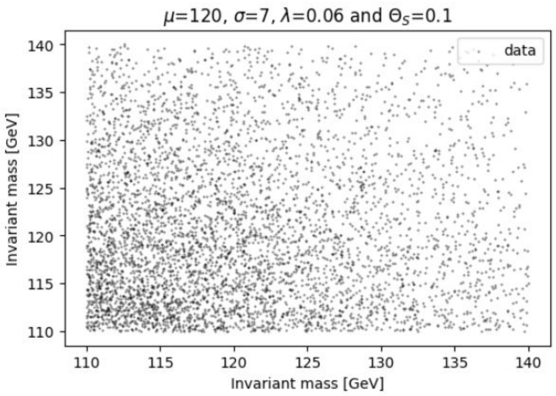
Exp rate λ
prior
True
posterior
Fraction
True
posterior Theta
Theta[1] prior
μ
True
Posterior μ
Prior μ
Invariant mass [GeV]
σ
True
Posterior σ
Prior σ

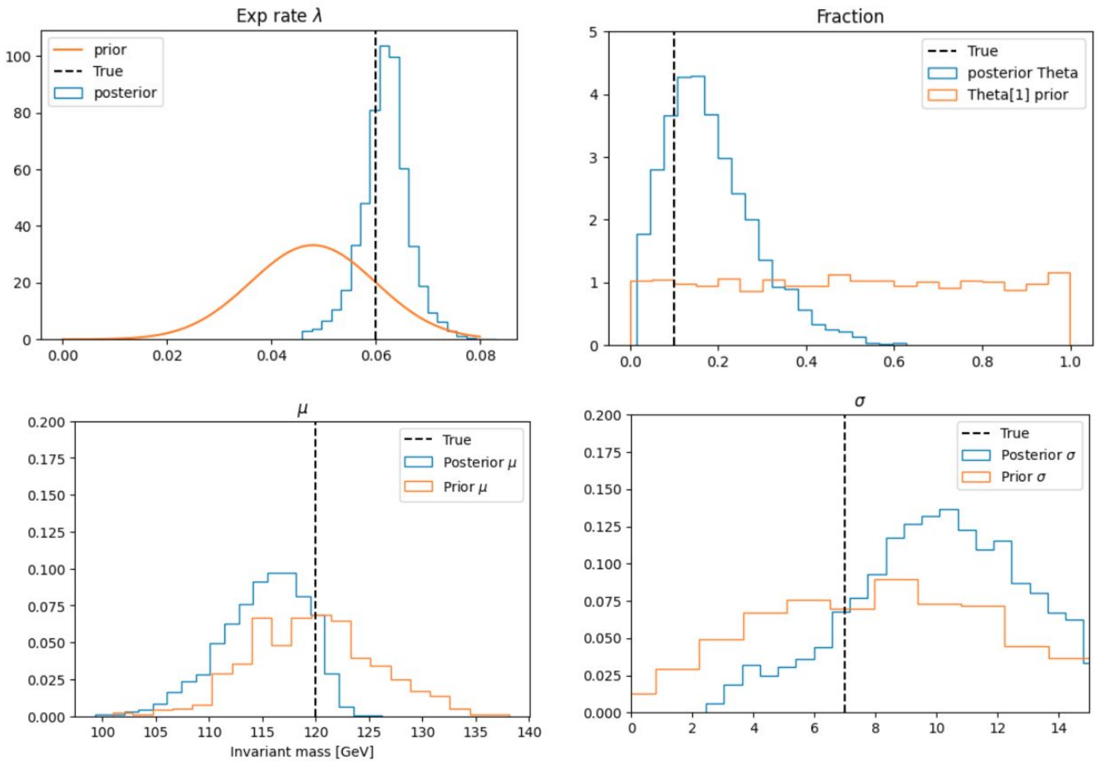
Mmm… this is harder



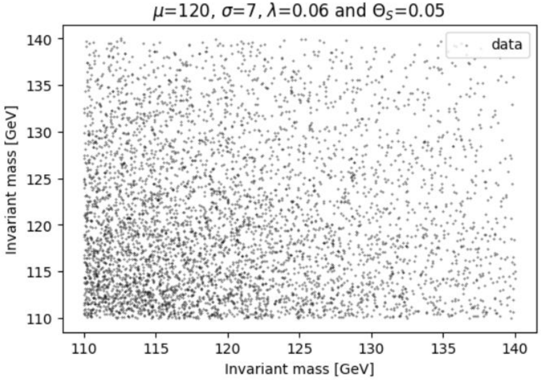


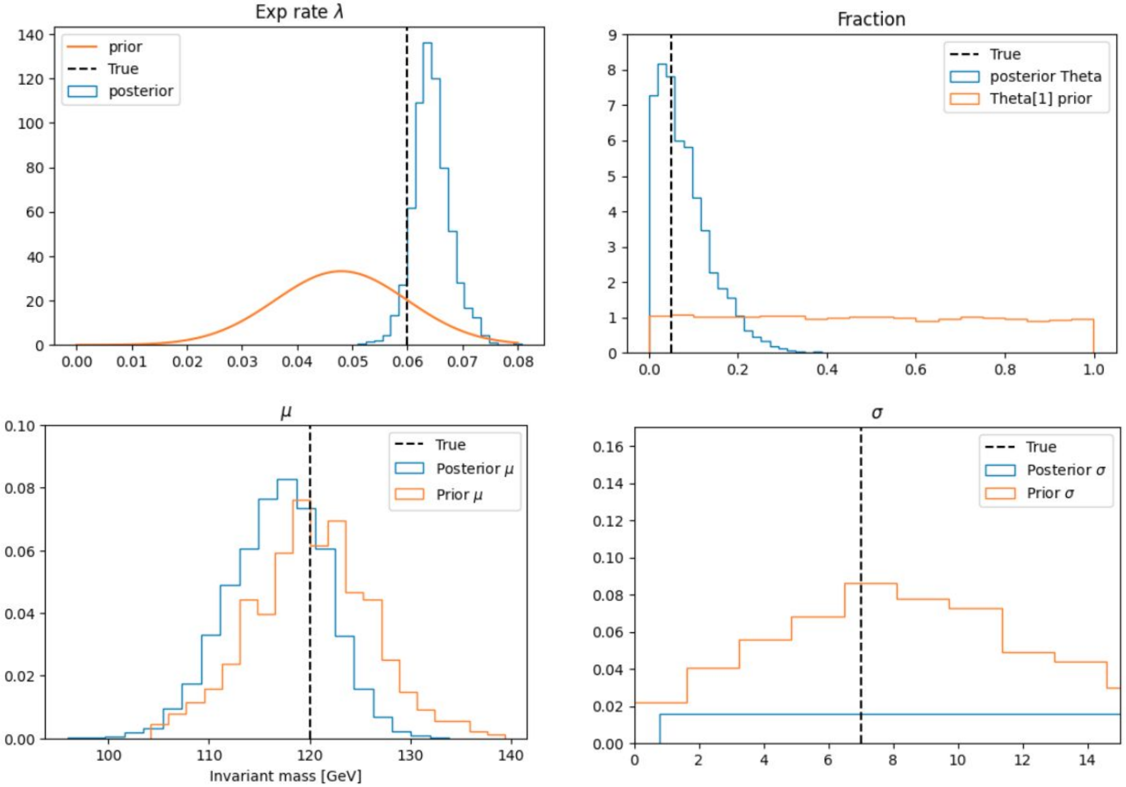
The impossible.... @10%



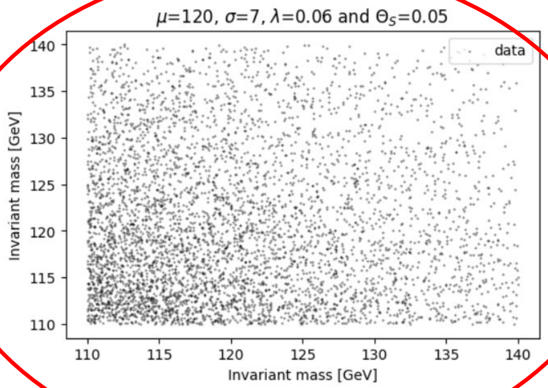


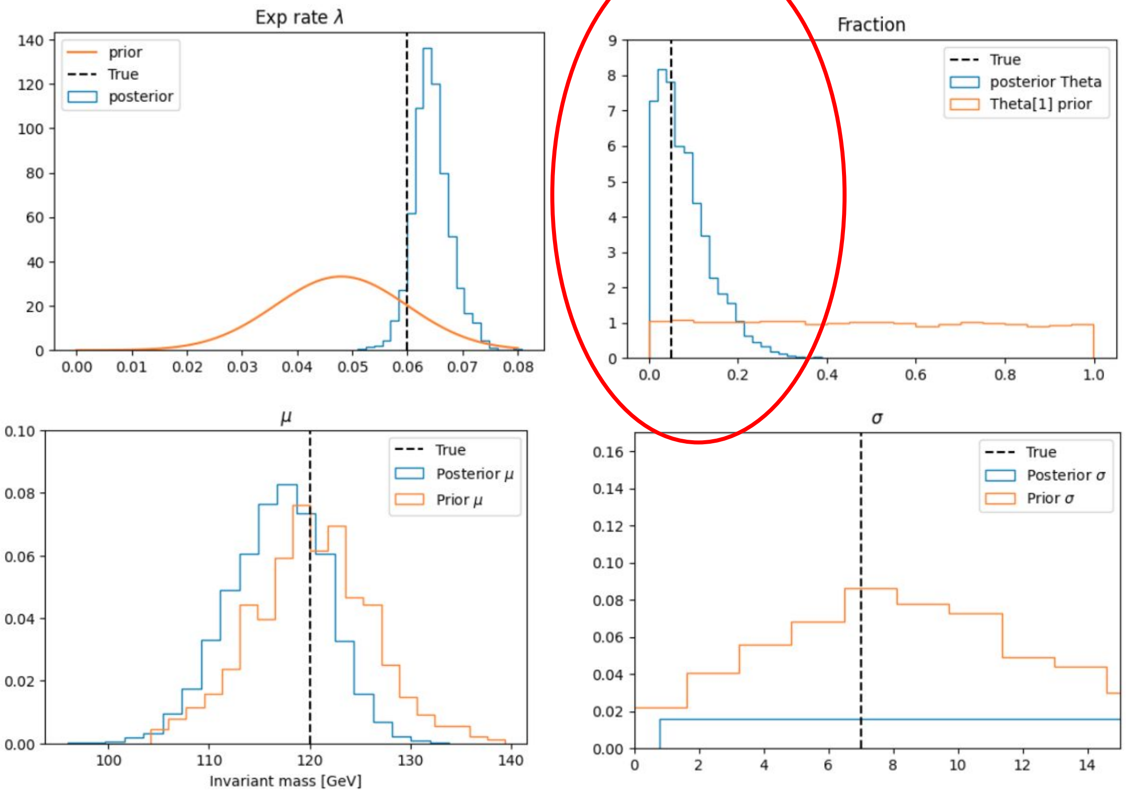
The impossible.... @5%!





The impossible.... @5%!



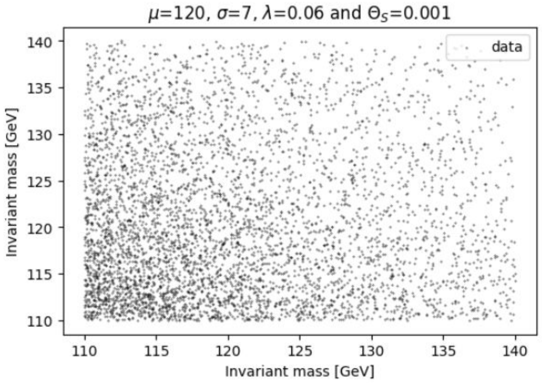


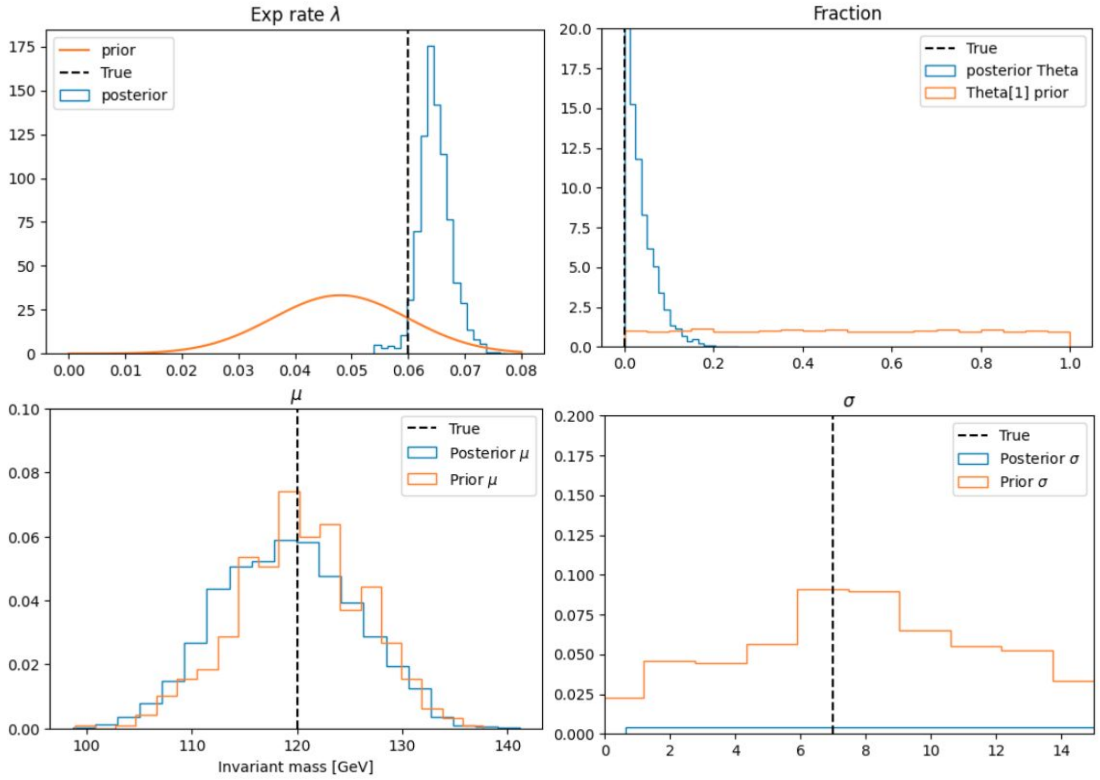
Correlation

Correlation

Correlation!

The impossible.... 0% signal!





Final remarks @ Lecture 3

Final remarks @ Lecture 3

- Mixture Models : Important tool for Science!
- Dirichlet Distribution : Important tool for Statistics in Science
- 2D Mixture Model
  - Stan simple programming
  - The power of correlations
  - The power of prior knowledge
  - $hh \rightarrow bb\gamma\gamma$ very useful… also in other studies!

Open questions @ Lecture 3

- How can we say we are right if we don't know the trues ?
- And if data does not follow a parametric curve (Normal, Exponential, etc)?