data science for (physical) scientists VI

fitting models to data - MCMC

dr.federica bianco | fbb.space | fedhere | fedhere

this slide deck

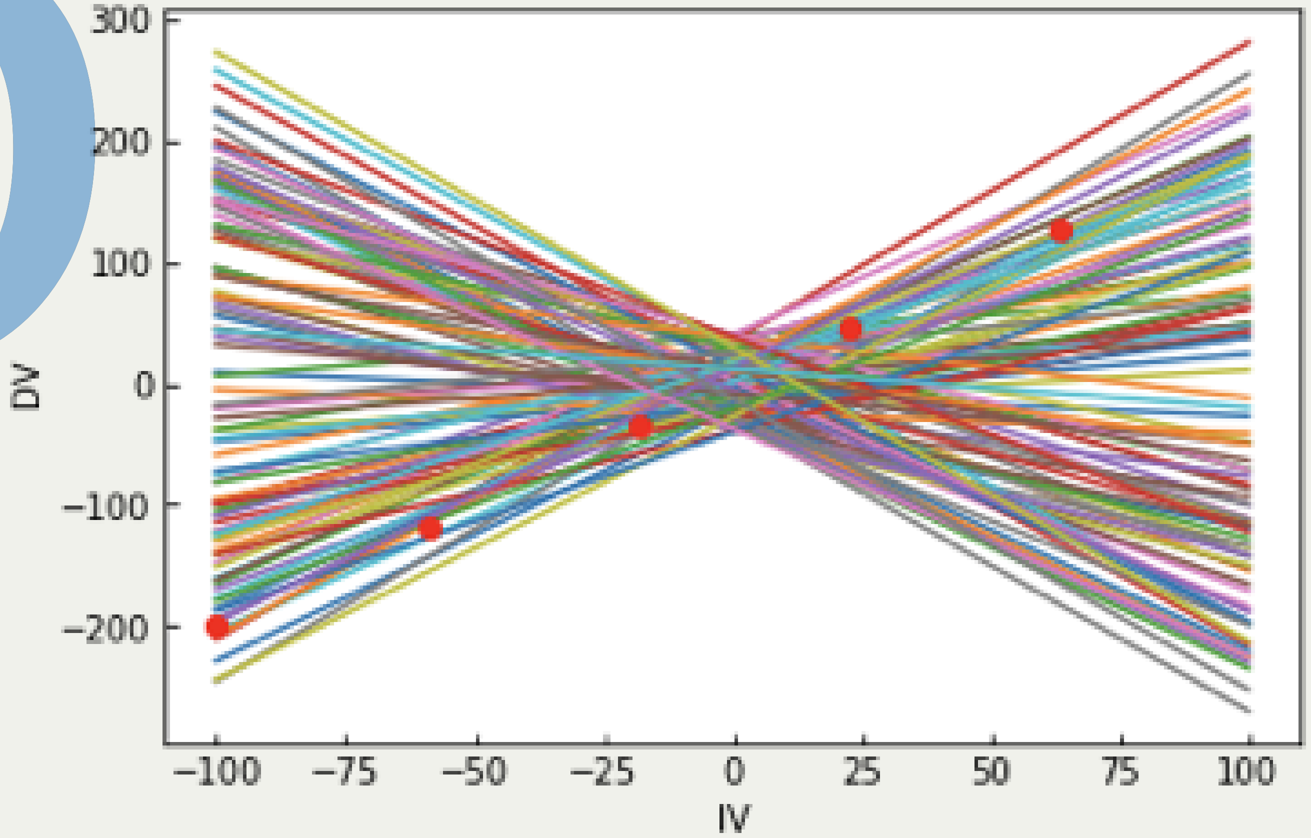
http://bit.ly/UDdsps6

recap

fitting models to data

recap 1

line model: *ax+b*



choose your model :

choose a mathematical formula to represent the behavior you see/expect in the data

recap 1

choose your model :

choose a mathematical formula to represent the behavior you see/expect in the data

a *mathematical* representastion of reality

In applying mathematics to subjects such as physics or statistics we make tentative assumptions about the real world which we know are false but which we believe may be useful nonetheless.

George Box, 1976

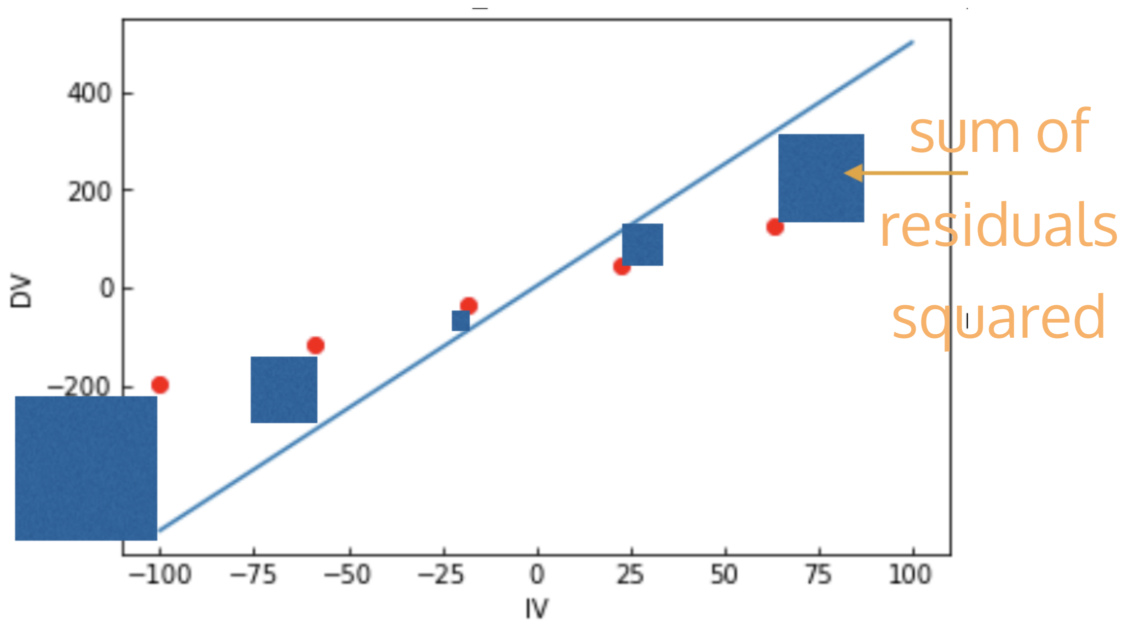
- no model is right
- some models are useful

recap

2

choose an objective function :

you need a plan to choose the parameters of the model: to "optimize" the model.
You need to choose something to be MINIMIZED or MAXIMIZED



$$\sum \frac{(y_i - (mx_i + b))^2}{2\sigma_i^2} \sim \chi^2(dof = DOF)$$

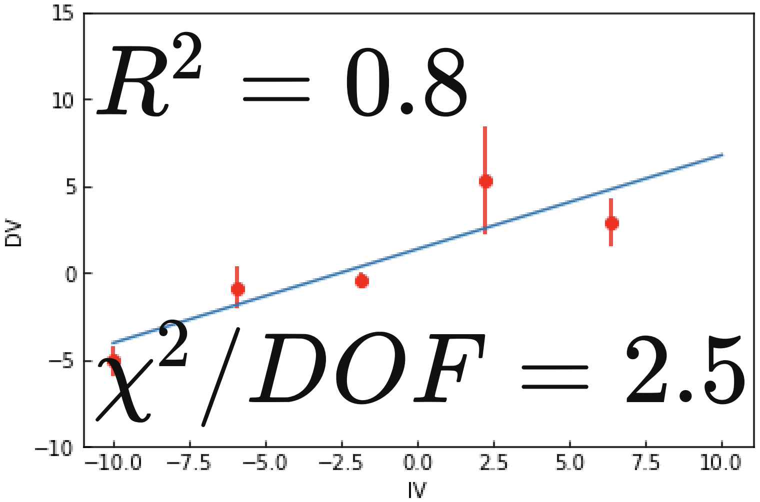
$$R^2 = 1 - \frac{\sum_i (y_i - f_i)^2}{\sum_i (y_i - \bar{y})^2}$$

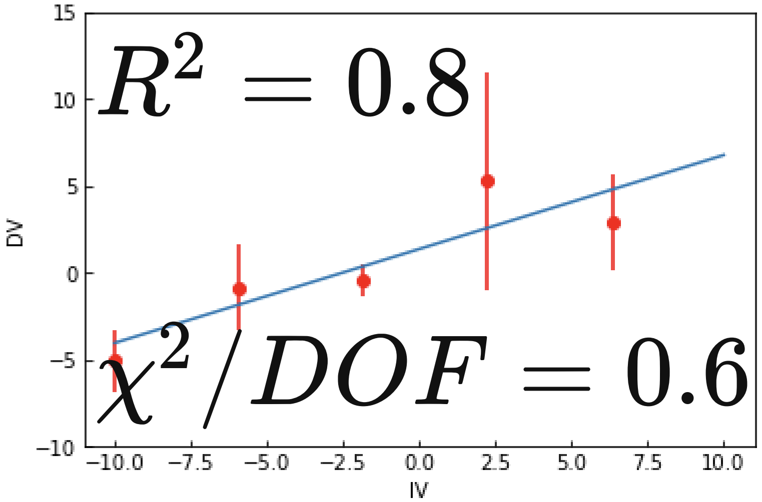
$$\frac{\sum \frac{(y_i - (mx_i + b))^2}{2\sigma_i^2}}{DOF} \sim \chi^2(dof = 1)$$

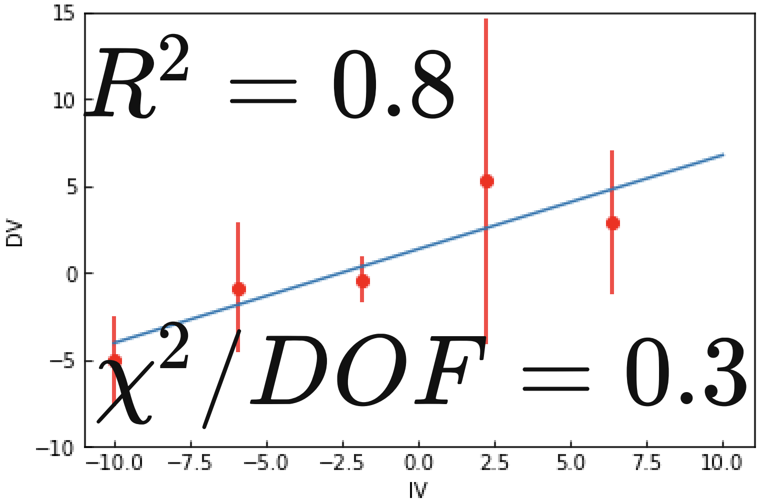
recap 3

evaluate the quality of your model

again: many options!







recap 3

evaluate the quality of your model

again: many options!

homoscedastic :

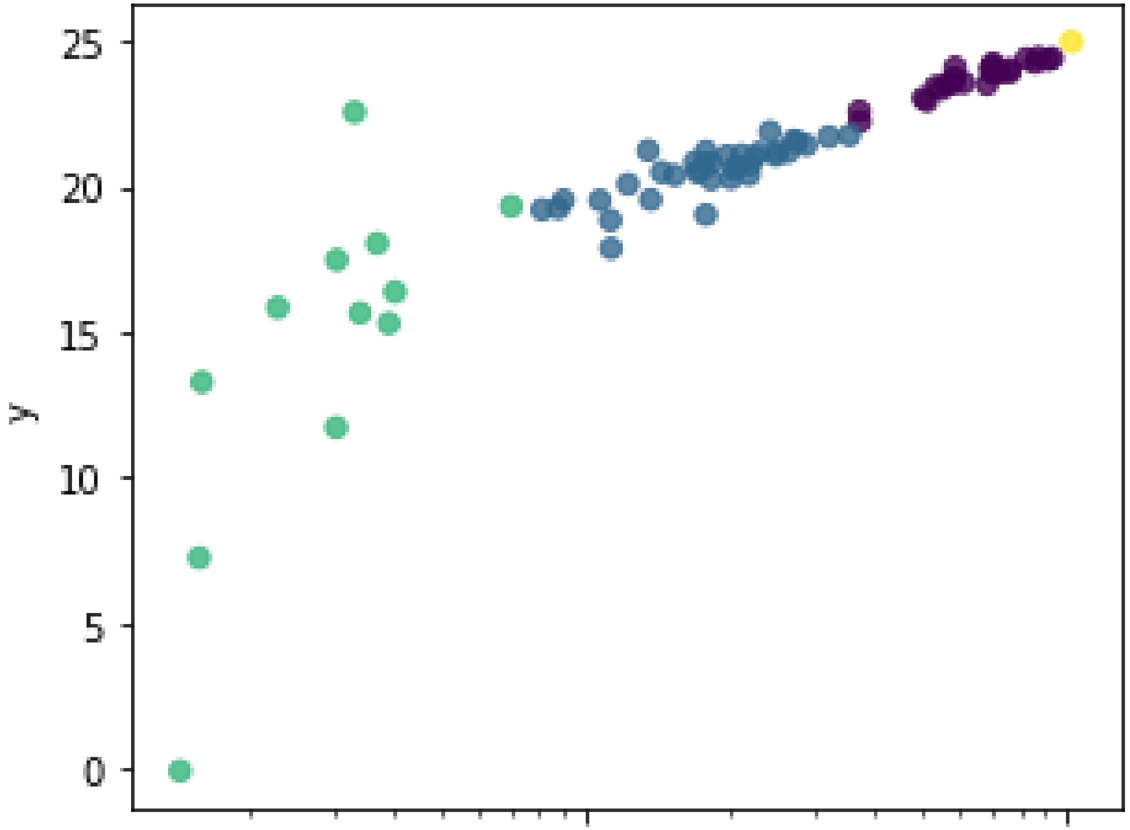
the uncertainty is the same for all data points

heteroscedastic:

the uncertainty different for each datapoint

(almost always the case in physics!)

recap 3



scatter may **dependent on exogenous variable**

(very difficult problem not well studied in statistics - very common in physics!)

evaluate the quality of your model

again: many options!

Stochastic vs Systematics

Systematic	**Statistical**
Biases the measurement *in one direction*	No preferred direction
Affects the sample regardless of the size	Shrinks with the sample size (typically as N)
Any distribution (usually we use Gaussian though	Typically Gaussian or Poisson

Fitting models in ML: Cross Validation

recap 3

1. Split data into a training subset and a test subset
2. Fit the model to the training data
3. Calculate the error of the model on the test data
4. REPEAT

WHY? you can find out how good your model is AND if it is OVERFITTING

Choosing a model: the principle of Parsimony

recap

4

William of Ockham (logician and Franciscan friar) 1300ca

but probably to be attributed to John Duns Scotus (1265–1308)

“Complexity needs not to be postulated without a need for it”

“Between 2 theories that perform similarly choose the ***one with fewer parameters***”

fitting a model to data

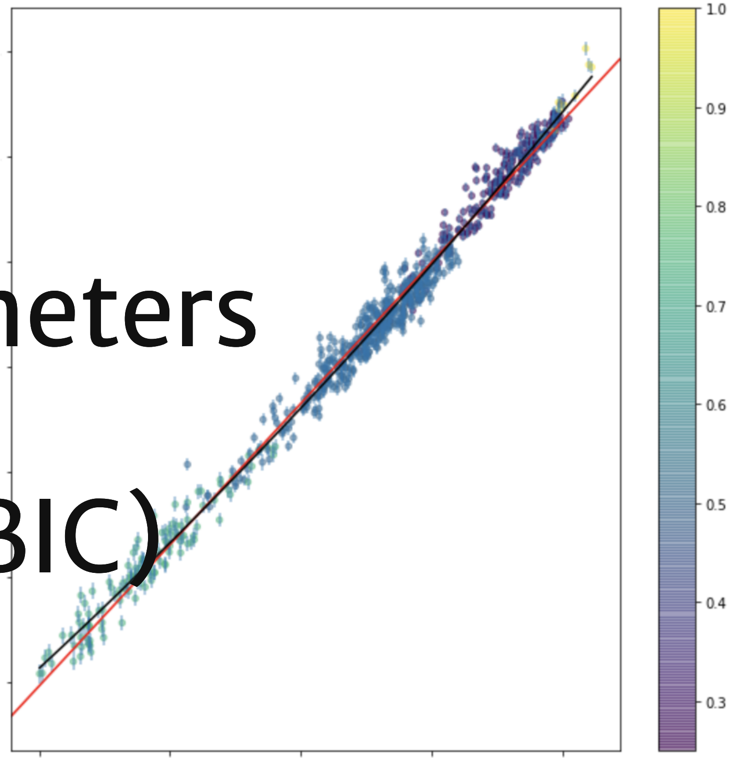
epistemology

1 - >1 order equation

2 - uncertainties in the fit parameters

3 - comparing models (LR, AIC, BIC)

4 - MCMC



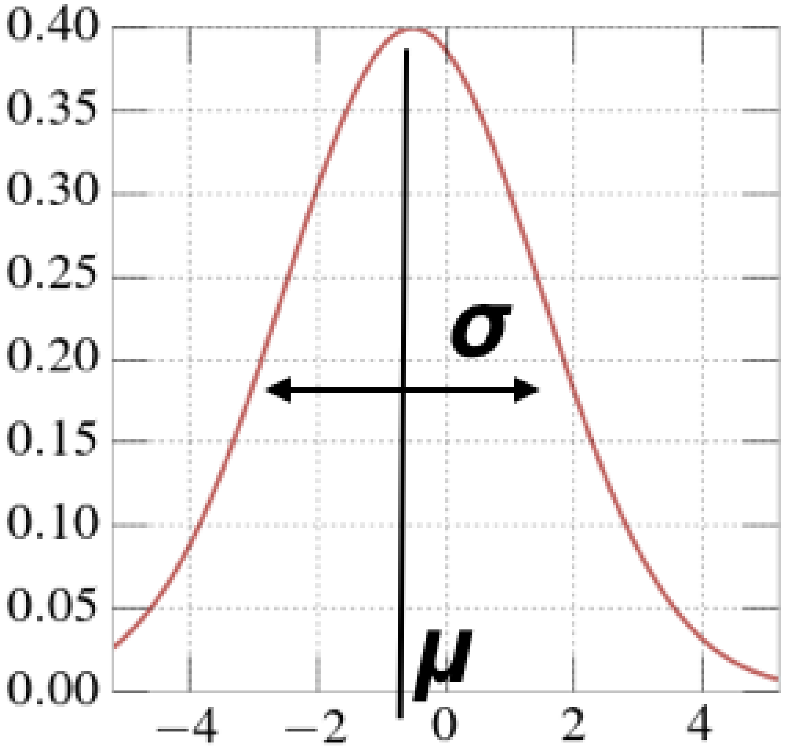
Probability vs Likelihood

Probability of data given model $P(x|\theta)$

Probability vs Likelihood

Probability of data given model $P(x|\theta)$

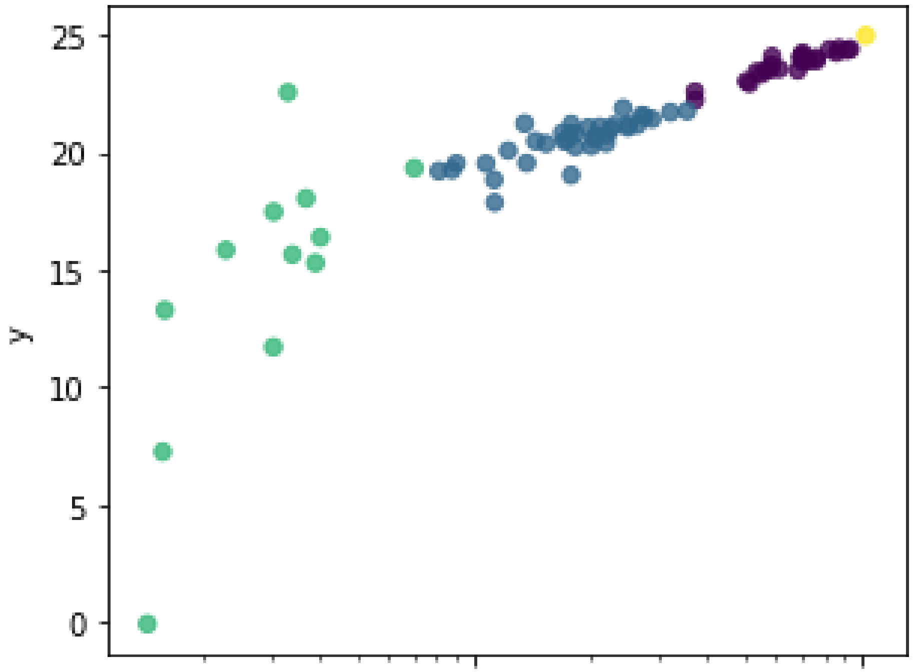
Gaussian distribution: $P(x|\mu, \sigma)$



Probability vs Likelihood

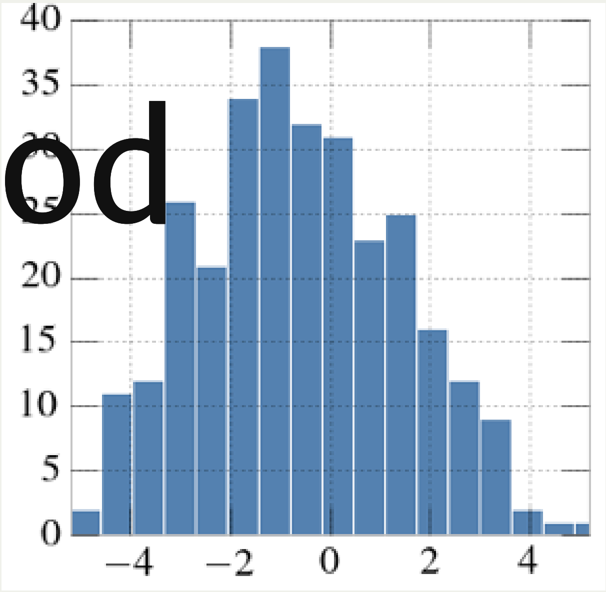
Probability of data given model $$P(x|\theta)$$

$$P(x|\mu, \sigma)$$

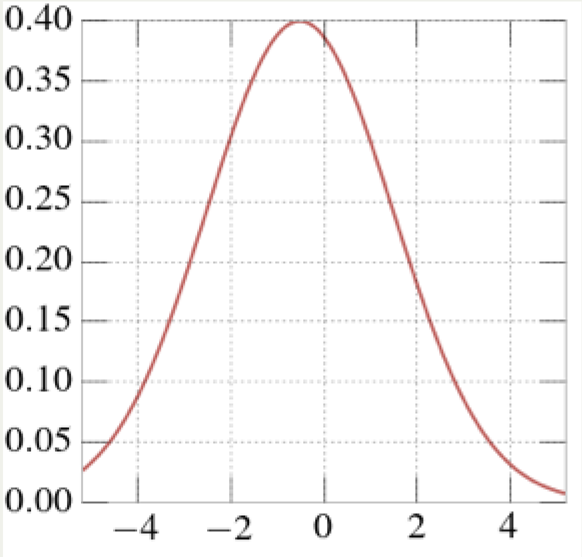


Noisy line function: $$P(\vec{y}|\vec{x}, a, b, \mu, \sigma(x))$$

Probability vs Likelihood

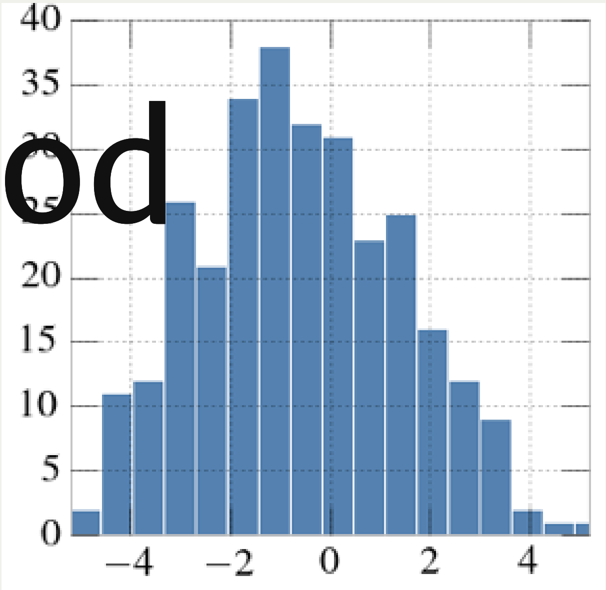


Probability of ***data*** given ***model*** $$P(x|\theta)$$

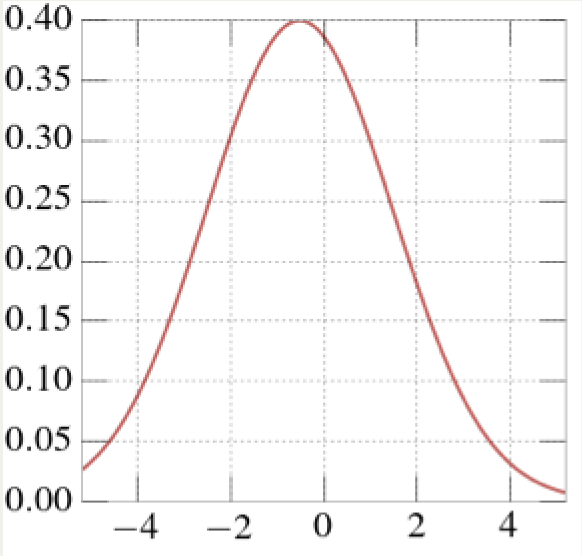


Probability of ***model*** given ***data*** $$L(\theta|x)$$

Probability vs Likelihood



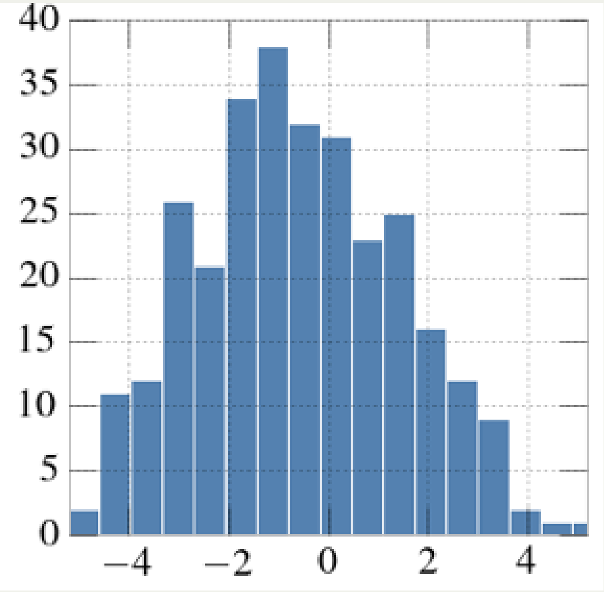
Probability of ***data*** given ***model*** $P(x|\theta)$



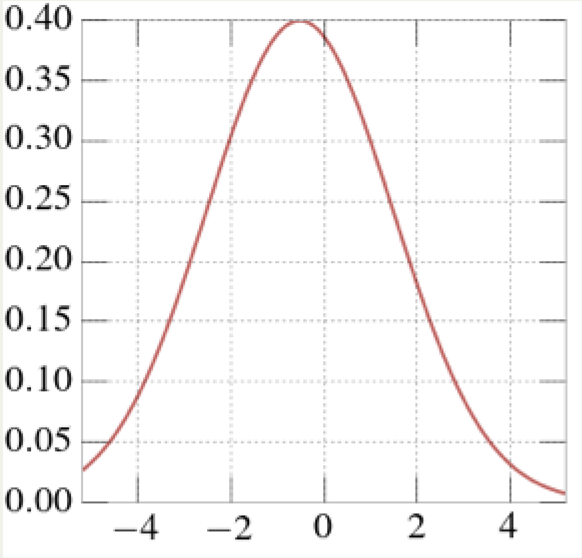
Probability of ***model*** given ***data*** $L(\theta|x)$

Same formula! different meaning

Likelihood



The likelihood is the probability of a model given the data - given what I measured (my observations) what is the probability that the data I observed is generated by a process such as the one described by my model



Probability of ***model*** given ***data***

$$L(\theta|x)$$

Likelihood

Assume the data is generated in a Gaussian distribution

Probability of ***data*** given ***model***

$$N(\mu, \sigma) \sim \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

Probability of ***model*** given ***data***

$$L_{\mu,\sigma}(x) \sim \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

Likelihood

Assume the data is generated in a Gaussian distribution

Probability of ***data*** given ***model***

$$N(\mu, \sigma) \sim \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

Probability of ***model*** given ***data***

$$L_{\mu,\sigma}(x) \sim \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

Likelihood

Assume the data is generated in a Gaussian distribution

Probability of ***data*** given ***model***

$$N(\mu, \sigma) \ \sim \ \Pi_i \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x_i-\mu)^2}{2\sigma^2}}$$

Probability of ***model*** given ***data***

$$L_{\mu,\sigma}(x) \ \sim \ \Pi_i \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x_i-\mu)^2}{2\sigma^2}}$$

Given some observations $\vec{x}$ we want to model them with the best function: the one that is MAXIMALLY LIKELY.

Likelihood

Assume the data is generated in a Gaussian distribution

Probability of ***data*** given ***model***

$$N(\mu, \sigma) \sim \Pi_i \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x_i-\mu)^2}{2\sigma^2}}$$

Probability of ***model*** given ***data***

$$L_{\mu,\sigma}(x) \sim \Pi_i \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x_i-\mu)^2}{2\sigma^2}}$$

Given some observations $\vec{x}$ we want to model them with the best function: the one that is MAXIMALLY LIKELY.

After we choose a functional form (*N*) for the model we want to choose the parameters μ, σ that maximize

Likelihood

Assume the data is generated in a Gaussian distribution

Probability of ***data*** given ***model***

$$N(\mu, \sigma) \sim \Pi_i \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x_i - \mu)^2}{2\sigma^2}}$$

Probability of ***model*** given ***data***

$$L_{\mu,\sigma}(x) \sim \Pi_i \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x_i - \mu)^2}{2\sigma^2}}$$

Given some observations $\vec{x}$ we want to model them with the best function: the one that is MAXIMALLY LIKELY.

After we choose a functional form (*N*) for the model we want to choose the parameters μ, σ that maximize

Likelihood

Assume the data is generated in a Gaussian distribution

Probability of ***data*** given ***model***

$$N(\mu, \sigma) \sim \Pi_i \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x_i - \mu)^2}{2\sigma^2}}$$

Probability of ***model*** given ***data***

$$L_{\mu,\sigma}(x) \sim \Pi_i \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x_i - \mu)^2}{2\sigma^2}}$$

$$\mathrm{Find}\ (\mu^*, \sigma^*)\ ||\ L_{\mu^*,\sigma^*} = max\,(L_{\mu,\sigma})$$

Given some observations $\vec{x}$ we want to model them with the best function: the one that is MAXIMALLY LIKELY.

After we choose a functional form (N) for the model we want to choose the parameters μ, σ that maximize

Likelihood

Assume the data is generated in a Gaussian distribution

Probability of ***data*** given ***model***

$$N(\mu, \sigma) \sim \Pi_i \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x_i-\mu)^2}{2\sigma^2}}$$

Probability of ***model*** given ***data***

$$L_{\mu,\sigma}(x) \sim \Pi_i \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x_i-\mu)^2}{2\sigma^2}}$$

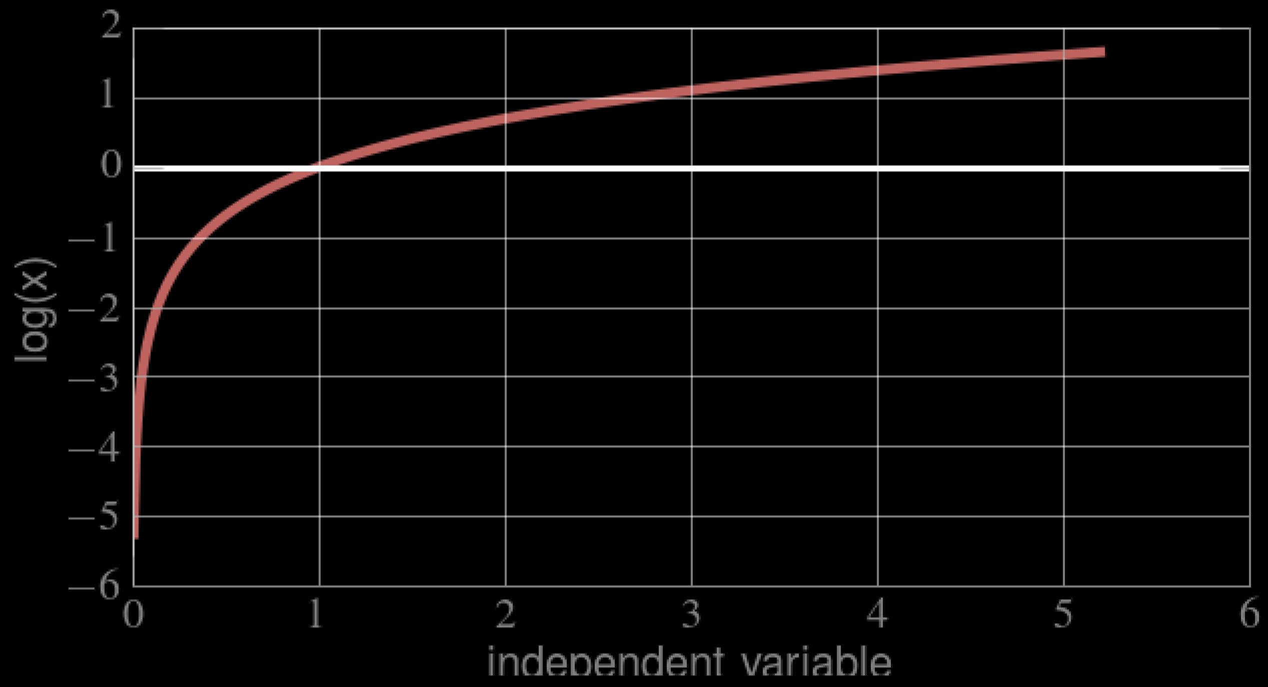
$$\text{Find } (\mu^*, \sigma^*) \; || - log\left(L_{\mu^*,\sigma^*}\right) = min\left(-log\left(L_{\mu,\sigma}\right)\right)$$

Logarithms

MONOTONICALLY INCREASING

if x grows, log(x) grows, if x decreases, log(x) decreases

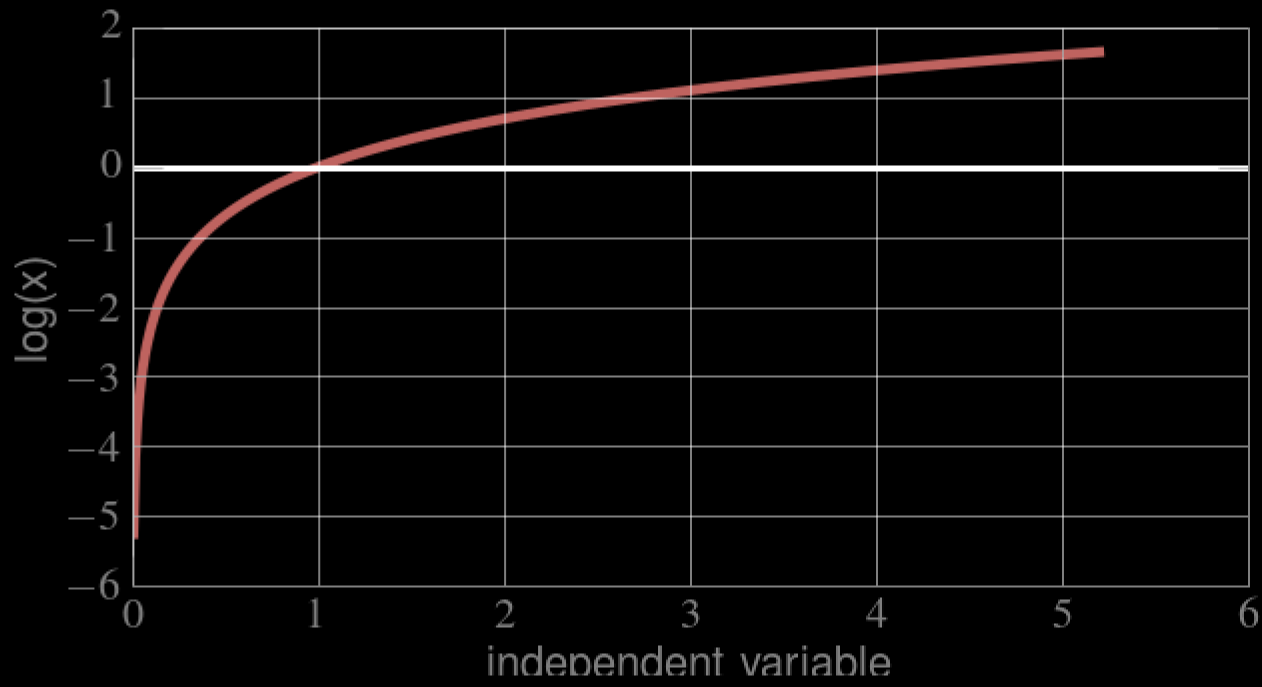
the location of the maximum is the same!



Logarithms

MONOTONICALLY INCREASING

SUPPORT : $(0, \infty]$

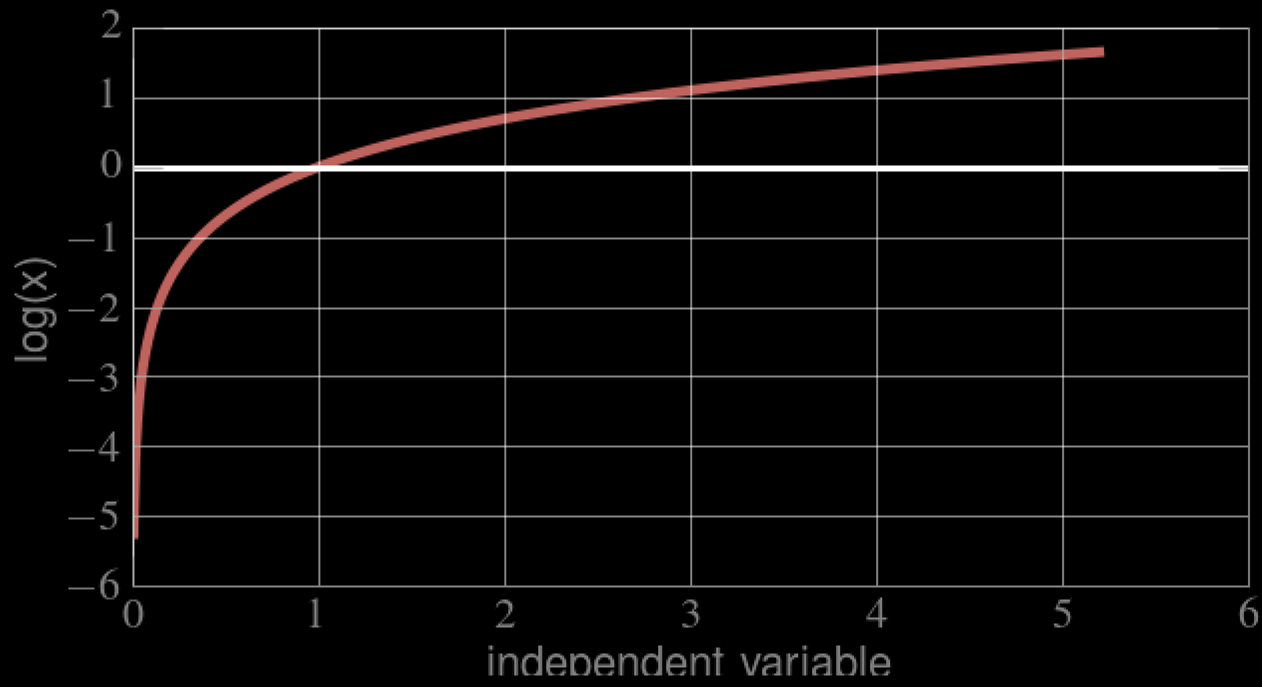


Logarithms

MONOTONICALLY INCREASING

SUPPORT : $(0, \infty]$

Not a problem cause L like P is positive defined



Data analysis recipes: Fitting a model to data*

David W. Hogg
Center for Cosmology and Particle Physics, Department of Physics, New York University
Max-Planck-Institut für Astronomie, Heidelberg

Jo Bovy
Center for Cosmology and Particle Physics, Department of Physics, New York University

Dustin Lang
Department of Computer Science, University of Toronto
Princeton University Observatory

https://arxiv.org/pdf/1008.4686.pdf

In the case of the straight line fit in the presence of known, Gaussian uncertainties in one dimension, one can create this generative model as follows: Imagine that the data *really do* come from a line of the form $y = f(x) = m\,x + b$, and that the only reason that any data point deviates from this perfect, narrow, straight line is that to each of the true y values a small y-direction offset has been added, where that offset was drawn from a Gaussian distribution of zero mean and known variance σ_y^2. In this model, given an independent position x_i, an uncertainty σ_{yi}, a slope m, and an intercept b, the frequency distribution $p(y_i|x_i, \sigma_{yi}, m, b)$ for y_i is

$$p(y_i|x_i, \sigma_{yi}, m, b) = \frac{1}{\sqrt{2\,\pi\,\sigma_{yi}^2}}\,\exp\left(-\frac{[y_i - m\,x_i - b]^2}{2\,\sigma_{yi}^2}\right) \quad , \qquad (9)$$

where this gives the expected frequency (in a hypothetical set of repeated experiments[13]) of getting a value in the infinitesimal range $[y_i, y_i + \mathrm{d}y]$ per unit $\mathrm{d}y$.

The generative model provides us with a natural, justified, scalar objective: We seek the line (parameters m and b) that maximize the probability of the observed data given the model or (in standard parlance) the *likelihood of the parameters*.[14] In our generative model the data points are independently drawn (implicitly), so the likelihood $\mathscr{L}$ is the product of conditional probabilities

$$\mathscr{L} = \prod_{i=1}^{N} p(y_i|x_i, \sigma_{yi}, m, b) \quad . \qquad (10)$$

likelihood, probability, and objective functions

$$L(m, b|\vec{y}) = \prod_i^N p_i(y_i|x_i, m, b)$$

$$L(m, b|\vec{y}) = \prod_i^N p_i(y_i|x_i, \sigma_i, m, b)$$

$$p(y_i) = \frac{1}{\sigma_i\sqrt{2\pi}} \exp - \frac{(y_i-(mx_i+b))^2}{2\sigma_i^2}$$

$$\log a \cdot b = \log a + \log b$$

$$\ln L(m, b|\vec{y}) = \ln \prod_i^N \frac{1}{\sigma_i\sqrt{2\pi}} \exp - \frac{y_i-(mx_i+b)}{2\sigma_i^2}$$

likelihood, probability, and objective functions

$$L(m,b|\vec{y}) = \prod_i^N p_i(y_i|x_i,m,b)$$

$$L(m,b|\vec{y}) = \prod_i^N p_i(y_i|x_i,\sigma_i,m,b)$$

$$p(y_i) = \frac{1}{\sigma_i\sqrt{2\pi}} \exp - \frac{(y_i-(mx_i+b))^2}{2\sigma_i^2}$$

$$x^a \cdot x^b = x^{(a+b)}$$

$$\ln L(m,b|\vec{y}) = \ln \prod \frac{1}{\sigma_i\sqrt{2\pi}} + \ln\left(\prod_i^N e^{-\frac{y_i-(mx_i+b)}{2\sigma_i^2}}\right)$$

likelihood, probability, and objective functions

$$L(m,b|\vec{y}) = \prod_i^N p_i(y_i|x_i,m,b)$$

$$L(m,b|\vec{y}) = \prod_i^N p_i(y_i|x_i,\sigma_i,m,b)$$

$$p(y_i) = \frac{1}{\sigma_i\sqrt{2\pi}} \exp -\frac{(y_i-(mx_i+b))^2}{2\sigma_i^2}$$

$$\ln L(m,b|\vec{y}) = \ln \prod \frac{1}{\sigma_i\sqrt{2\pi}} + \ln\left(e^{-\sum_i^N \frac{y_i-(mx_i+b)}{2\sigma_i^2}}\right)$$

likelihood, probability, and objective functions

$$L(m,b|\vec{y}) = \prod_i^N p_i(y_i|x_i,m,b)$$

$$L(m,b|\vec{y}) = \prod_i^N p_i(y_i|x_i,\sigma_i,m,b)$$

$$p(y_i) = \frac{1}{\sigma_i\sqrt{2\pi}} \exp -\frac{(y_i-(mx_i+b))^2}{2\sigma_i^2}$$

σ_i not part of the model

$$\ln L(m,b|\vec{y}) = \ln \prod \frac{1}{\sigma_i\sqrt{2\pi}} + \ln\left(e^{-\sum_i^N \frac{y_i-(mx_i+b)}{2\sigma_i^2}}\right)$$

likelihood, probability, and objective functions

$$L(m,b|\vec{y}) = \prod_i^N p_i(y_i|x_i,m,b)$$

$$L(m,b|\vec{y}) = \prod_i^N p_i(y_i|x_i,\sigma_i,m,b)$$

$$p(y_i) = \frac{1}{\sigma_i\sqrt{2\pi}} \exp - \frac{(y_i-(mx_i+b))^2}{2\sigma_i^2}$$

$$\ln L(m,b|\vec{y}) = K - \sum \frac{(y_i-(mx_i+b))^2}{2\sigma_i^2}$$

likelihood, probability, and objective functions

$$L(m, b|\vec{y}) = \prod_i^N p_i(y_i|x_i, m, b)$$

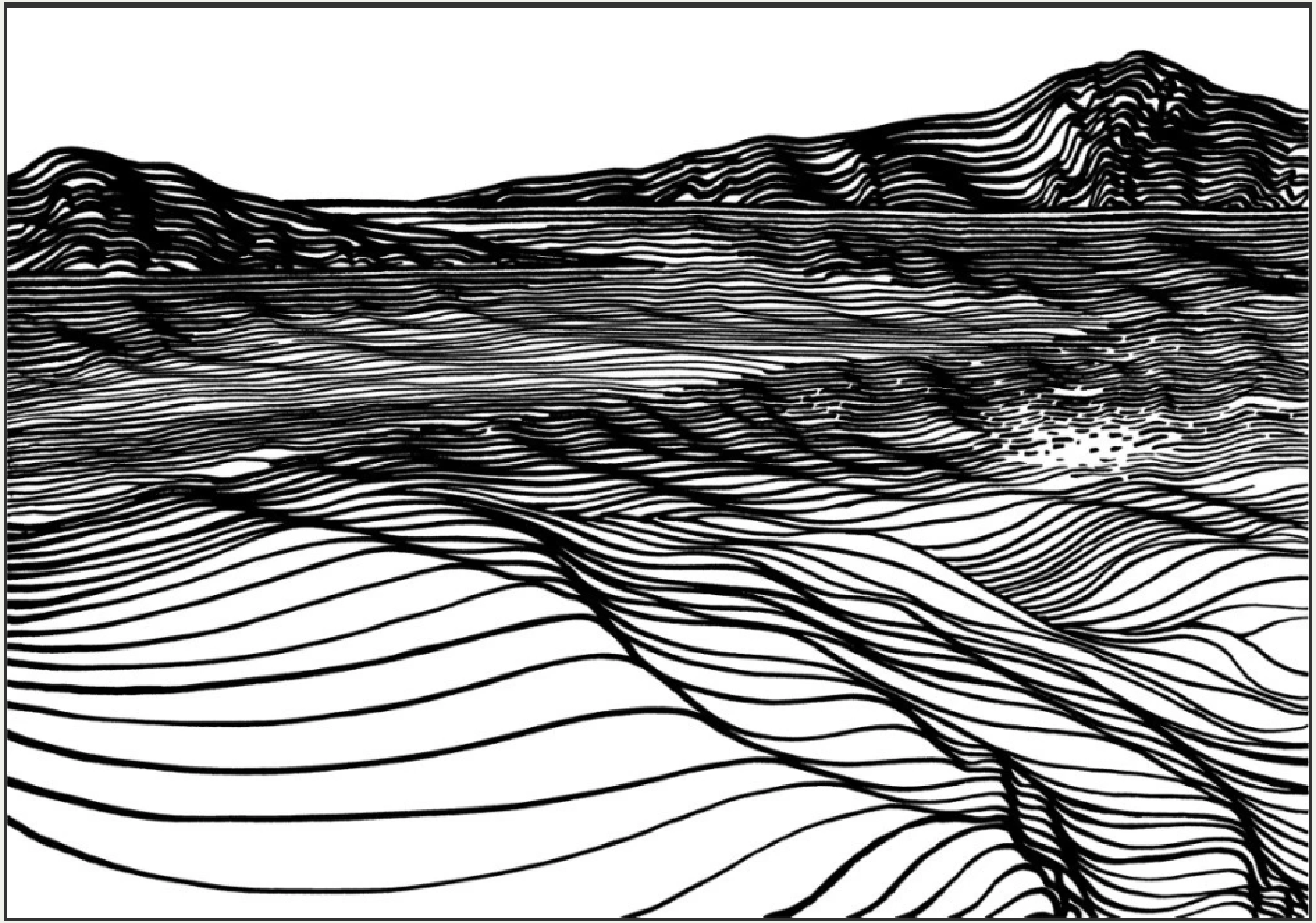
$$L(m, b|\vec{y}) = \prod_i^N p_i(y_i|x_i, \sigma_i, m, b)$$

$$p(y_i) = \frac{1}{\sigma_i\sqrt{2\pi}} \exp - \frac{(y_i - (mx_i + b))^2}{2\sigma_i^2}$$

$$\ln L(m, b|\vec{y}) = K - \sum \frac{(y_i - (mx_i + b))^2}{2\sigma_i^2} = K - \frac{1}{2}\chi^2$$

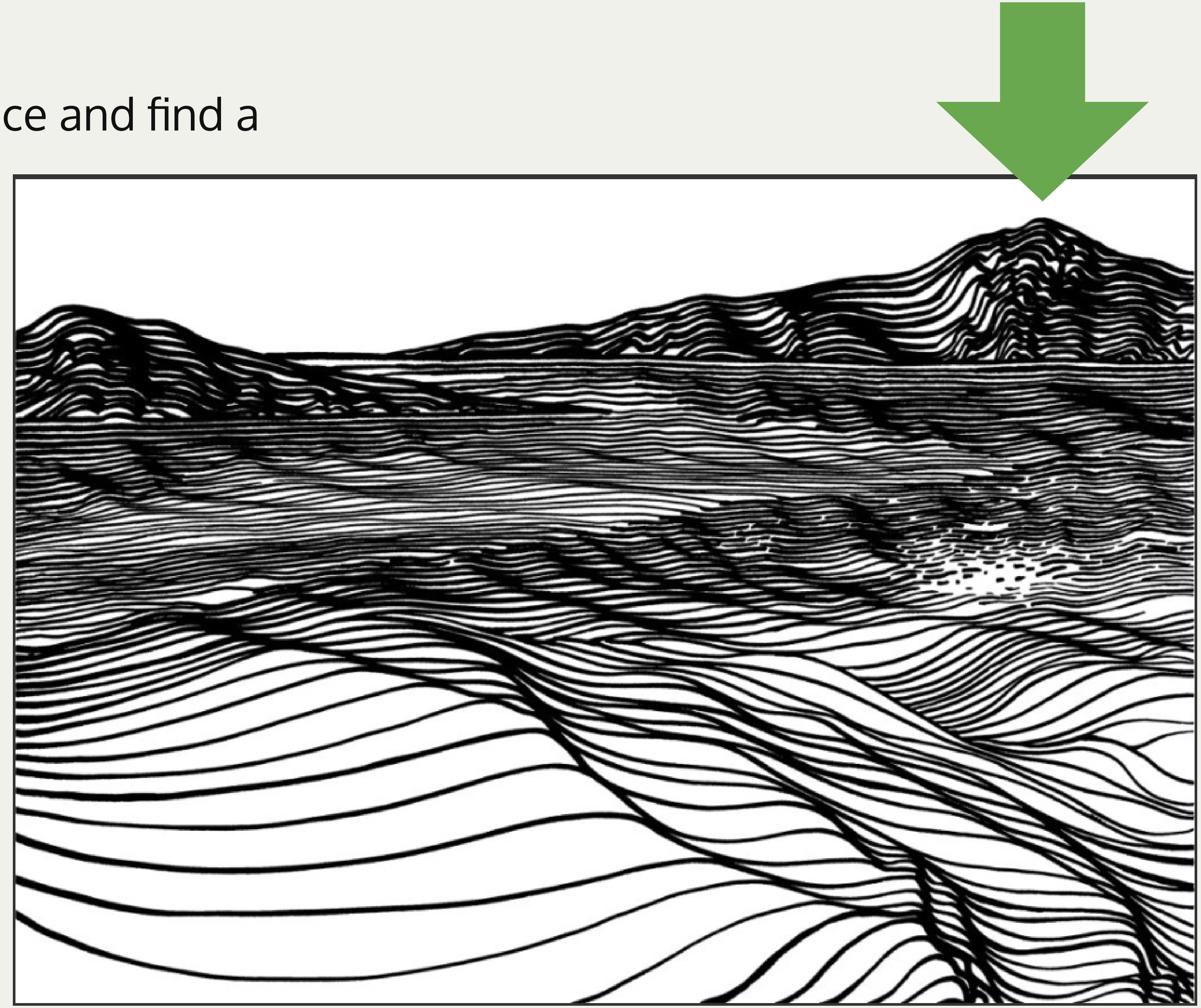
think about the likelihood surfece...

you want to explore the surface and find a peak



think about the likelihood surfece...

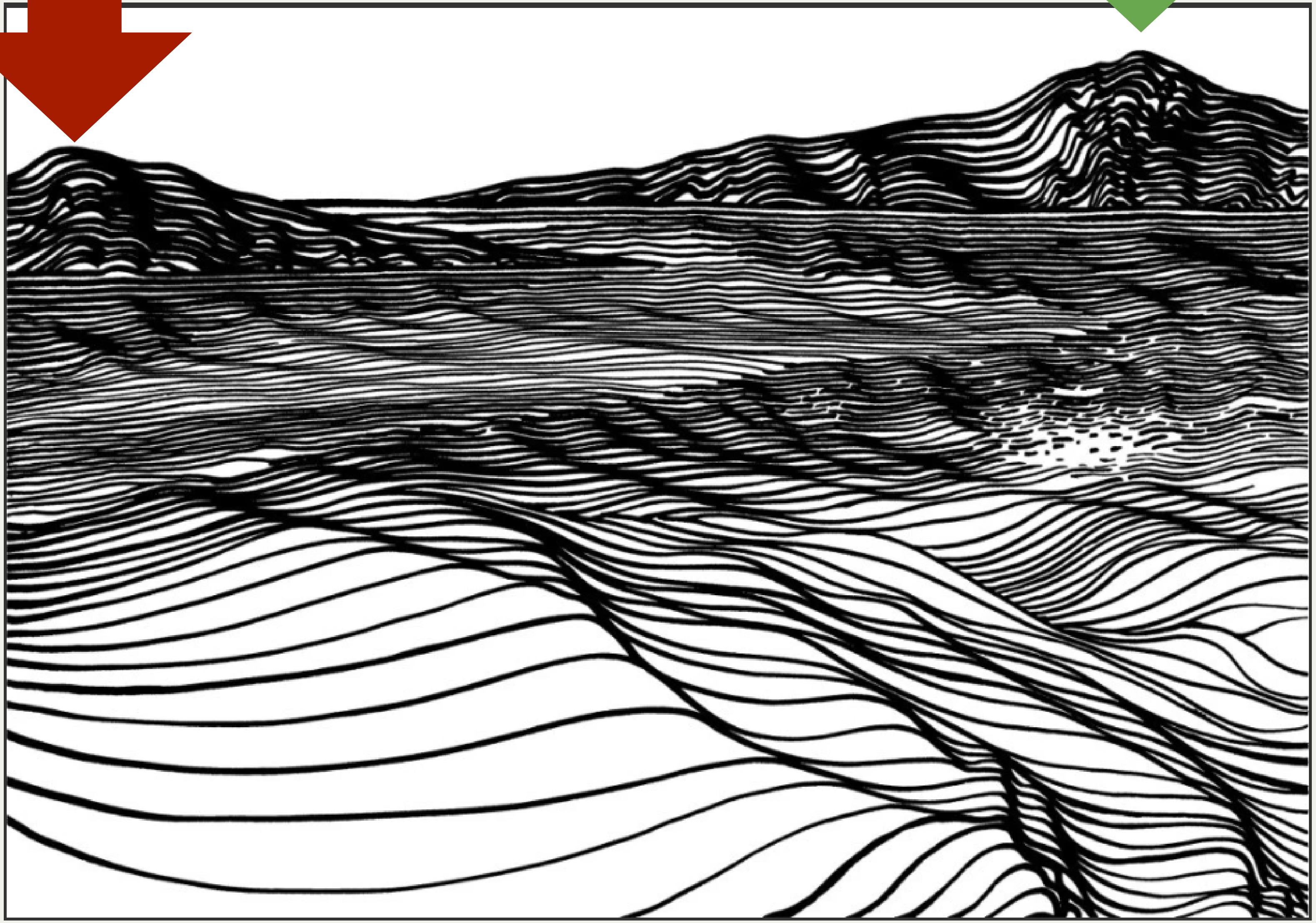
you want to explore the surface and find a peak



think about the likelihood surfece...

https://github.com/fedhere/DSPS/blob/master/lab7/lineFitLab.ipynb

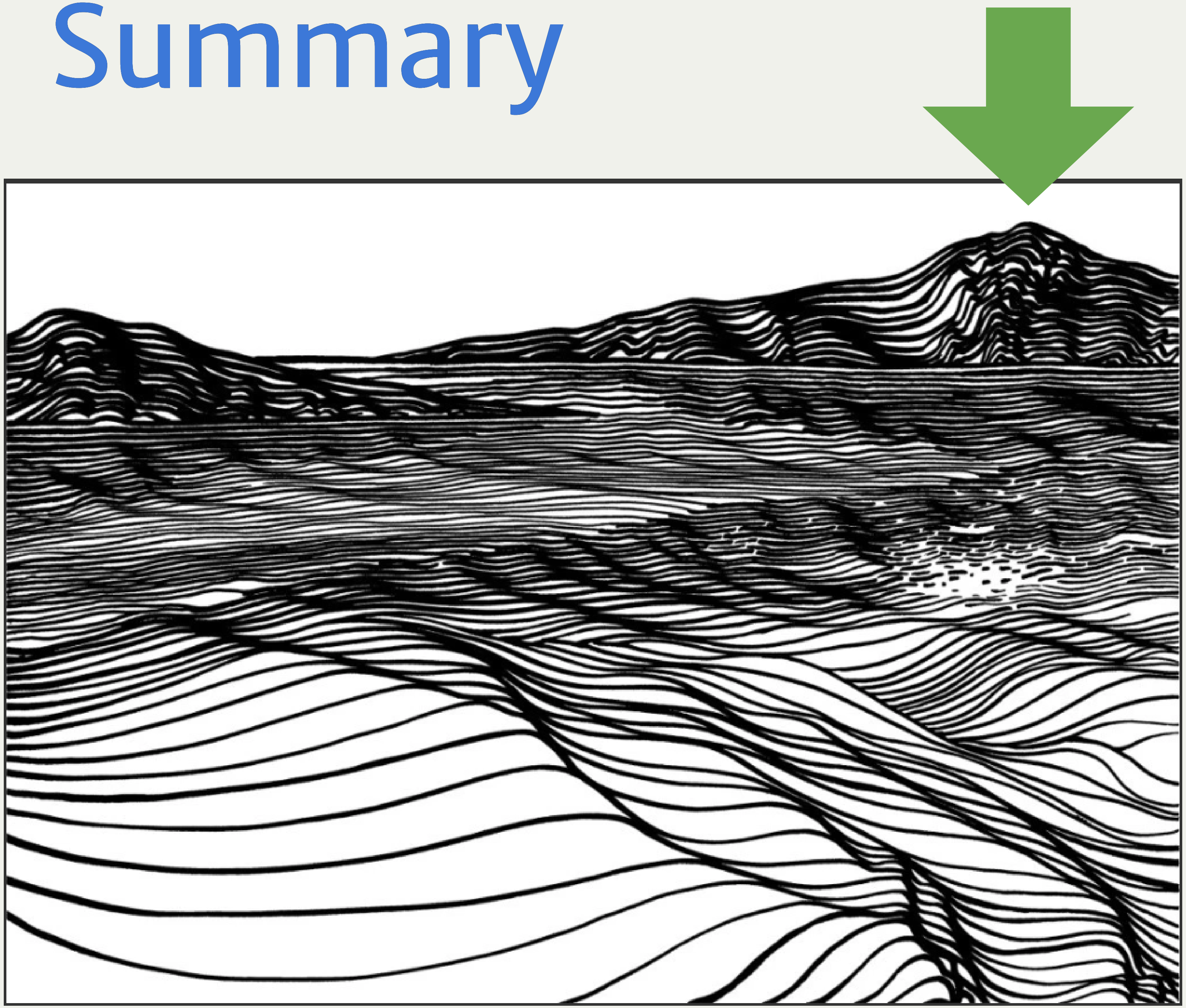
you want to explore the surface and find a peak



possible issues:

- how do I efficiently explore the whole surface?
- how do I explore the WHOLE survey at all?
- how do I avoid getting stuck in a local minimum (maximum)?

Summary



The problem of fitting models to data reduces to finding the

maximum *likelihood*

of the data given the model

This is effectively done by finding the **minimum** of the

-log(*likelihood*)

HOW DO I CHOOSE A MODEL?

NESTED MODELS (one model contains the other one, e.g.

y= ***mx + l*** is contained in

*y=ax**2+* ***mx + l***

Given two models which is preferable?

Likelihood-ratio tests

likelihood ratio statistics LR

$$LR = -2log_e \frac{L(\text{complex model})}{L(\text{simple model}}$$

A *rigorous* answer (in terms of NHST) can be obtained for ***2 nested* models**

This directly answers the question:

"is my more complex model *overfitting* the data?"

The LR statistics is expected to follow a χ^2 distrbution under the *Null Hypothesis* that the ***simpler model is preferable***

statsmodels.model.compare_lr_ratio()

HOW DO I CHOOSE A MODEL?

Given two models which is preferable?

A *rigorous* answer (in terms of NHST) can be obtained for ***2 nested*** **models**

This directly answers the question:

"is my more complex model *overfitting* the data?"

The LR statistics is expected to follow a χ^2 distrbution under the *Null Hypothesis* that the ***simpler model is preferable***

NESTED MODELS (one model contains the other one, e.g.

y= ***mx + l*** is contained in

*y=ax**2+* ***mx + l***

Likelihood-ratio tests

likelihood ratio statistics LR

$$LR = -2log_e \frac{L(\text{complex model})}{L(\text{simple model}}$$

statsmodels.model.compare_lr_ratio()

follow a χ^2 with **degrees of freedom equal to the difference in the number of degrees of freedom between the two models** (i.e., the number of variables added to the model).

MCMC

Mote Carlo Markov Chain

MCMC

Mote Carlo Markov Chain

part 1: Bayes Theorem

Bayes Theorem

$$P(A|B) = \frac{P(B|A)P(B)}{P(B)}$$

Bayes Theorem

likelihood

$$P(\theta|D,f) = \frac{P(D|\theta,f)\,P(\theta,f)}{P(D|f)}$$

we are going to sample the likelihood:

Bayes Theorem

likelihood

$$P(\theta|D,f) = \frac{P(D|\theta,f)\,P(\theta,f)}{P(D|f)}$$

posterior

Definitions:

posterior: joint probability distributin of a parameter set (*m*, *b*) condition upon some data *D* and a model hypothesys *f*

$$P(D|\theta, f)$$

Bayes Theorem

likelihood

prior

$$P(\theta|D,f) = \frac{P(D|\theta,f)\,P(\theta,f)}{P(D|f)}$$

posterior

Definitions:

posterior: joint probability distributin of a parameter set (*m*, *b*) condition upon some data *D* and a model hypothesys *f*

$$P(D|\theta, f)$$

prior: “intellectual” knowledge about the model parameters

$$P(\theta, f)$$

likelihood prior

Bayes Theorem

$$P(\theta|D,f) = \frac{\overbrace{P(D|\theta,f)}^{\text{likelihood}}\overbrace{P(\theta,f)}^{\text{prior}}}{\underbrace{\cancel{P(D|f)}}_{\text{evidence}}}$$

posterior

evidence

Definitions:

posterior: joint probability distributin of a parameter set (*m*, *b*) condition upon some data *D* and a model hypothesys *f*

$$P(D|\theta, f)$$

prior: “intellectual” knowledge about the model parameters

$$P(\theta, f)$$

evidence: marginal likelihood of data under the model $P(D|f) = \int P(D|\theta,f)P(\theta|f)d\theta$

its constant in θ so we can ignore it!

MCMC

Mote Carlo Markov Chain

part 2: Sampling a posterior

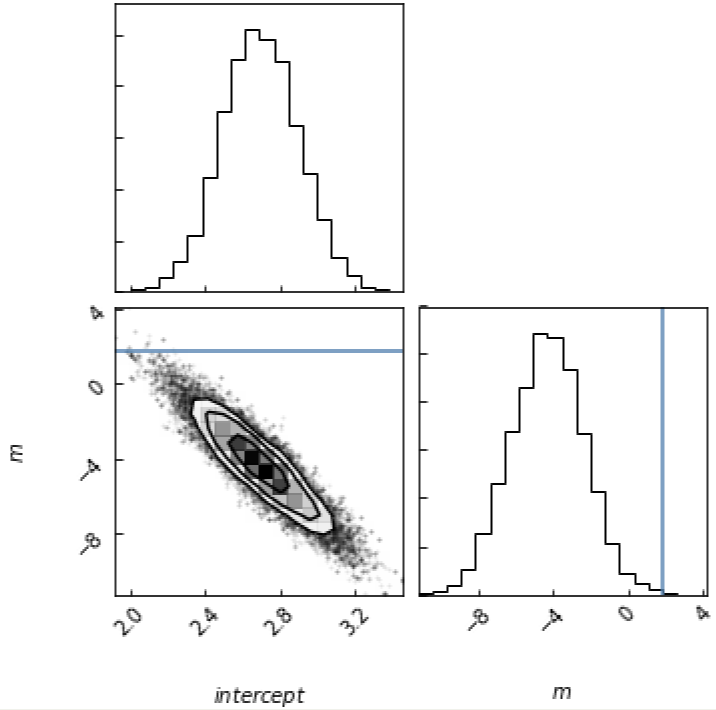
MCMC

$$P(\theta|D,f) \propto P(D|\theta,f)\,P(\theta,f)$$

posterior

posterior: joint probability distributin of a parameter set (*m, b*) condition upon some data *D* and a model hypothesys *f*

triangle plot

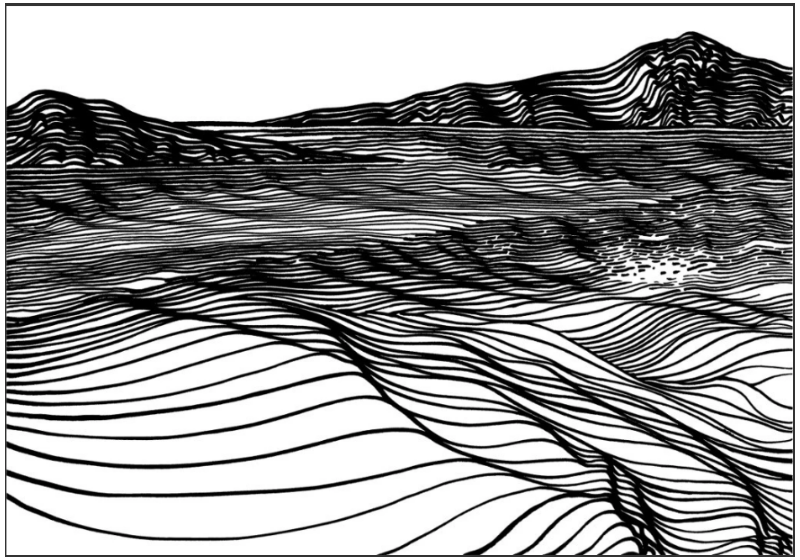


MCMC

$$P(\theta|D,f) \propto P(D|\theta,f)P(\theta,f)$$

posterior

Goal: sample the posterior distribution



choose a starting point **current** = ***θ₀= (m,b)***

WHILE convergence criterion is met:

calculate current posterior **post_curr = *P(D|θ,f)***

*/*proposal*/*
chose a new set of parameters **new = *θ_new= (m,b)***

calculate the new posterior **post_new=*P(D|θ_new,f)***

IF ***post_new > post_curr:***
current = new
ELSE***:***
/*accept with probability ***P(D|θ_new,f) / P(D|θ,f)*** */
r = random uniform number [0,1]
IF **r < *post_new / post_orig:***
current = new
ELSE:
pass //do nothing

MCMC

$$P(\theta|D,f) \propto P(D|\theta,f)P(\theta,f)$$

posterior

Goal: sample the posterior distribution

Questions:

0. how do I choose a starting point?

we arent even going to talk about it…

```
choose a starting point current = θ0= (m,b)
WHILE convergence criterion is met:
        calculate current posterior post_curr = P(D|θ,f)
        /*proposal*/
        chose a new set of parameters new = θnew= (m,b)
        calculate the new posterior post_new=P(D|θnew,f)
        IF post_new > post_curr:
            current = new
        ELSE:
            /*accept with probability P(D|θnew,f) / P(D|θ,f) */
            r = random uniform number [0,1]
            IF r < post_new / post_orig:
                current = new
            ELSE:
                pass  //do nothing
```

MCMC

$$P(\theta|D,f) \propto P(D|\theta,f)\,P(\theta,f)$$

posterior

Goal: sample the posterior distribution

Questions:

1. how do I choose the next point?

Any *Markovian* process

```
choose a starting point current = θ0= (m,b)
WHILE convergence criterion is met:
        calculate current posterior post_curr = P(D|θ,f)
        /*proposal*/
        chose a new set of parameters new = θnew= (m,b)
        calculate the new posterior post_new=P(D|θnew,f)
        IF post_new > post_curr:
            current = new
        ELSE:
            /*accept with probability P(D|θnew,f) / P(D|θ,f) */
            r = random uniform number [0,1]
            IF r < post_new / post_orig:
                current = new
            ELSE:
                pass  //do nothing
```

Definition: A Markovian Process

A process is Markovian if the next state of the system is determined stochastically as a perturbation of the current state of the system, and *only* the current state of the system, i.e. the system has no memory of earlier states (a *memory-less* process).

Definition: A Markovian Process

A process is Markovian if the next state of the system is determined stochastically as a perturbation of the current state of the system, and *only* the current state of the system, i.e. the system has no memory of earlier states (a *memory-less* process).

A state being a stochastic perturbation of the previous state means that given the conditions of the state at time t (e.g. $A(t)$ = (position+velocity)) the *next* set of conditions $A(t+1)$ (updated position+velocity) will be drawn from a distribution related to the earlier state. For example the *next* velocity can be a sample from a Gaussian distribution with mean equal to the *current* velocity.

$$A(t+1) \sim \mathrm{N}(A(t), s)$$

MCMC

$$P(\theta|D,f) \propto P(D|\theta,f)P(\theta,f)$$

posterior

Goal: sample the posterior distribution

choose a starting point **current** = ***θ_0= (m,b)***

WHILE convergence criterion is met:

calculate current posterior **post_curr = *$P(D|\theta,f)$***

*/*proposal*/*
chose a new set of parameters **new = *θ_{new}= (m,b)***

calculate the new posterior **post_new=*$P(D|\theta_{new},f)$***

IF ***post_new > post_curr:***
current = new
ELSE***:***
/*accept with probability ***$P(D|\theta_{new},f) / P(D|\theta,f)$*** */
r = random uniform number [0,1]
IF **r < *post_new / post_orig:***
current = new
ELSE:
pass //do nothing

Questions:

1. how do I choose the next point?

Any *Markovian* process

Any *ergodic* process

Definition: An ergodic Process

(given enough time) the entire parameter space would be sampled.

Detailed Balance is a sufficient condition
for ergodicity
Metropolis Rosenbluth Rosenbluth Teller 1953 - Hastings 1970

At equilibrium, each elementary process should be equilibrated by its reverse process.

reversible Markov process

$$\pi(x_1)P(x_2|x_1) = \pi(x_2)P(x_1|x_2)$$

MCMC

$$P(\theta|D,f) \propto P(D|\theta,f)P(\theta,f)$$

posterior

Goal: sample the posterior distribution

DYI_MCMC.ipynb

choose a starting point **current** = ***θ_0= (m,b)***

WHILE convergence criterion is met:

calculate current posterior **post_curr** = ***$P(D|\theta,f)$***

*/*proposal*/*

chose a new set of parameters **new** = ***θ_{new}= (m,b)***

calculate the new posterior **post_new**=***$P(D|\theta_{new},f)$***

IF ***post_new > post_curr:***

current = new

ELSE***:***

/*accept with probability ***$P(D|\theta_{new},f) / P(D|\theta,f)$*** */

r = random uniform number [0,1]

IF **r <** ***post_new / post_orig:***

current = new

ELSE:

pass //do nothing

MCMC

$$P(\theta|D,f) \propto P(D|\theta,f)P(\theta,f)$$

posterior

Goal: sample the posterior distribution

Examples of how to choose the next point

Gibbs sampling:

Metropolis-Hastings proposal distribution with change *along a single direction at a time => always accept*

must know the integral P(D|f) along that direction

choose a starting point **current** = $\boldsymbol{\theta_0}$**= (m,b)**

WHILE convergence criterion is met:

    calculate current posterior **post_curr =** $\boldsymbol{P(D|\theta,f)}$

    /*proposal*/
    chose a new set of parameters **new =** $\boldsymbol{\theta_{new}}$**= (m,b)**

    calculate the new posterior **post_new=**$\boldsymbol{P(D|\theta_{new},f)}$

    IF ***post_new > post_curr:***
        current = new
    ELSE***:***
        /*accept with probability $\boldsymbol{P(D|\theta_{new},f) / P(D|\theta,f)}$ */
        r = random uniform number [0,1]
        IF **r < *post_new / post_orig:***
            current = new
        ELSE:
            pass //do nothing

MCMC

$$P(\theta|D,f) \propto P(D|\theta,f)\,P(\theta,f)$$

posterior

Goal: sample the posterior distribution

```
choose a starting point current = θ_0= (m,b)
WHILE convergence criterion is met:
        calculate current posterior post_curr = P(D|θ,f)
        /*proposal*/
        chose a new set of parameters new = θ_new= (m,b)
        calculate the new posterior post_new=P(D|θ_new,f)
        IF post_new > post_curr:
            current = new
        ELSE:
            /*accept with probability P(D|θ_new,f) / P(D|θ,f) */
            r = random uniform number [0,1]
            IF r < post_new / post_orig:
                current = new
            ELSE:
                pass  //do nothing
```

Examples of how to choose the next point

Other options:

simulated annealing (good for multimodal)
parallel tempering (good for multimodal)
differential evoution (good for covariant spaces)

MCMC

$$P(\theta|D,f) \propto P(D|\theta,f)P(\theta,f)$$

posterior

Goal: sample the posterior distribution

Examples of how to choose the next point

affine invariant : EMCEE package

1

```
choose a starting point current = θ_0= (m,b)
WHILE convergence criterion is met:
        calculate current posterior post_curr = P(D|θ,f)
        /*proposal*/
        chose a new set of parameters new = θ_new= (m,b)
        calculate the new posterior post_new=P(D|θ_new,f)
        IF post_new > post_curr:
            current = new
        ELSE:
          /*accept with probability P(D|θ_new,f) / P(D|θ,f) */
          r = random uniform number [0,1]
          IF r < post_new / post_orig:
              current = new
          ELSE:
              pass  //do nothing
```

MCMC

$$P(\theta|D,f) \propto P(D|\theta,f)P(\theta,f)$$

Examples of how to choose the next point

Other options:

simulated annealing (good for multimodal)
parallel tempering (good for multimodal)
differential evoution (good for covariant spaces)

https://www.youtube.com/embed/m1xN-iOGFPQ?enablejsapi=1

MCMC

https://www.youtube.com/embed/Vv3f0QNWvWQ?enablejsapi=1

Examples of how to choose the next point

Other options:

simulated annealing (good for multimodal)
parallel tempering (good for multimodal)
differential evoution (good for covariant spaces)

MCMC

Examples of how to choose the next point

affine invariant : EMCEE package

1

1

MCMC convergence

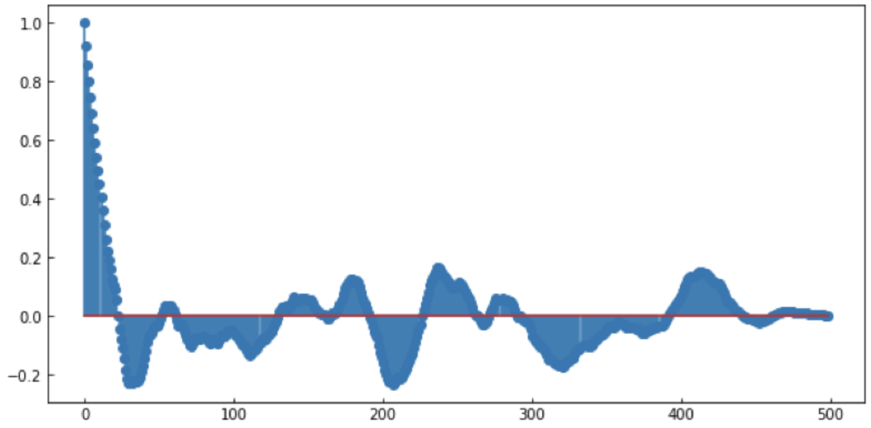
Goal: sample the posterior distribution

Questions:

1. how do I choose the next point?
2. when have I sampled the posterior adequatly?

   has your MCMC *converged* ?

MCMC convergence



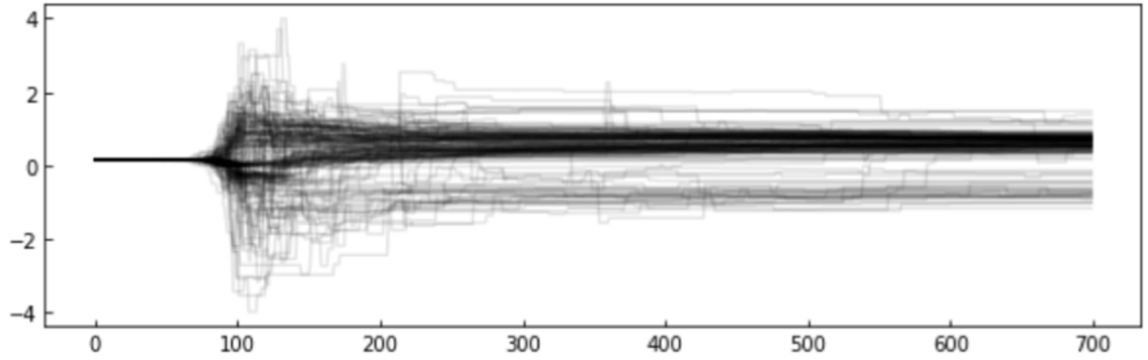
Goal: sample the posterior distribution

Questions:

1. how do I choose the next point?
2. when have I sampled the posterior adequatly?
   has your MCMC *converged*?

1. **check autocorrelation within a chain (*Raftery*)**
2. check that all chains coverged to same region (a stationary distribution *GelmanRubin*)
3. mean at beginning = mean at end (*Geweke*)
4. check that entire chain reached stationary distribution (or a final fraction of the chain, *Heidelberg-Welch* using Cramer-von-Mises statistic)

MCMC convergence



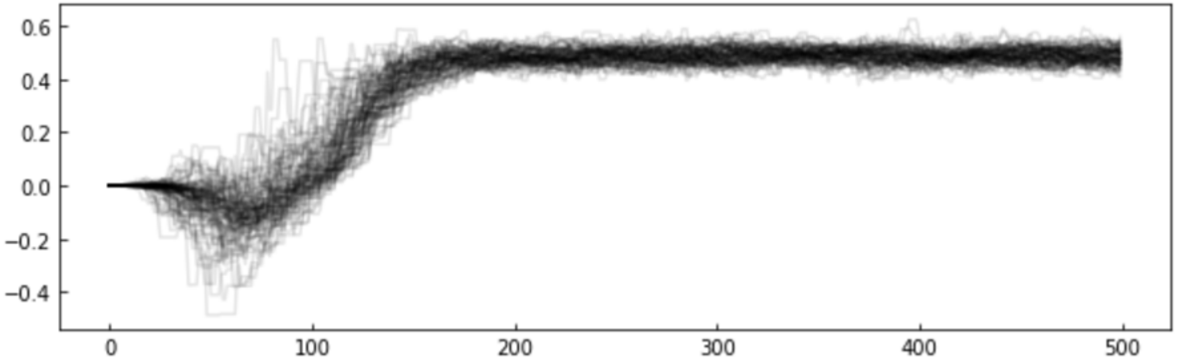
Goal: sample the posterior distribution

Questions:

1. how do I choose the next point?
2. when have I sampled the posterior adequatly?
   has your MCMC *converged* ?

1. check autocorrelation within a chain (*Raftery*)
2. **check that all chains coverged to same region (a stationary distribution *GelmanRubin*)**
3. mean at beginning = mean at end (*Geweke*)
4. check that entire chain reached stationary distribution (or a final fraction of the chain, *Heidelberg-Welch* using Cramer-von-Mises statistic)

MCMC convergence



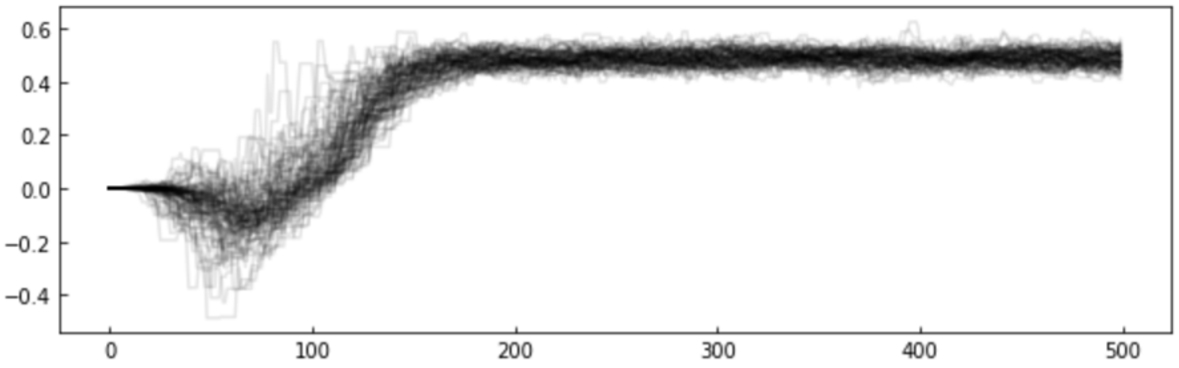
Goal: sample the posterior distribution

Questions:

1. how do I choose the next point?
2. when have I sampled the posterior adequatly?
   has your MCMC *converged*?

1. check autocorrelation within a chain (*Raftery*)
2. check that all chains coverged to same region (a stationary distribution *GelmanRubin*)
3. **mean at beginning = mean at end (*Geweke*)**
4. **check that entire chain reached stationary distribution (or a final fraction of the chain, *Heidelberg-Welch* using Cramer-von-Mises statistic)**

MCMC convergence



Goal: sample the posterior distribution

Questions:

1. how do I choose the next point?
2. when have I sampled the posterior adequatly?
   has your MCMC *converged*?
3. how can it be-the samples are *not independent!*

good point!…

1. check autocorrelation within a chain (*Raftery*)
2. check that all chains coverged to same region (a stationary distribution *GelmanRubin*)
3. mean at beginning = mean at end (*Geweke*)
4. check that entire chain reached stationary distribution (or a final fraction of the chain, *Heidelberg-Welch* using Cramer-von-Mises statistic)

Stochastic Processes in Science Inference: with the advent of computers (1940s), simulations became a valuable alternative to analytical derivation to solve complex scientific problems, and the only way to solve non-tractable problems. Events that occur with a known probability can be simulated, the possible outcomes would be simulated with a frequency corresponding to the probability.

Applications: Instances of the evolution of a complex systems can be simulated, and from this synthetic (simulated) sample solutions can be generalized as they would from a sample observed from a population:

Physics example: simulate multibody interactions (e.g. asteroids or particles in large systems) or nuclear reaction chains

Urban e.g.. *simulate traffic flow to determine the average trip duration instead of measuring many trips to estimate the trip duration,*

or a better scheme would be: *simulate traffic flow and validate your simulation by comparing the average trip duration for a synthetic sample and from a sample observed from the real system, then simulate proposed changes to traffic to validate and evaluate planning options before implementing them.*

Simulations require drawing samples from distributions.

We did not cocer this but it is important - you wont need to do it cause python numpy/scipy does it for you... but you should know this

Drawing samples from a distribution can be done directly if the probability PDF *P(X)* can be integrated *analytically* to find a CDF *F(x)* and if this CDF is invertible (*F-1(u) can be calculated analytically)* . The algorithm is:

1. draw a *uniformly distributed* number ***u*** between [0-1]
2. invert the CDF of your distribution evaluated at *u*: ***x=F-1(u)*** *is a sample from the desired PDF (i.e. x*'s are drawn at a frequency *P(x))*

If *F(x)* or *F-1(u)* cannot be calculated analytically **Rejection Sampling** allows to sample from the desired *P(x)* . The algorithm is:

1. find a function *Q(x)* that is larger than *P(x)* for every x and that has an analytical, integrable, invertible form
2. draw a sample x from *Q(x)* (see above)
3. draw a *uniformly distributed* number ***u*** between [0-Q(x)]
4. only accept x where *u <= P(x)*

If your proposal distribution is poorly chosen (much higher than P(x) in some regions) this can be an extremely wasteful process. The higher the problem dimensionality the more this issue becomes a concern. Alternatives include Importance sampling where the integral of the PDF

Markovian processes: A process is Markovian if the next state of the system is determined stochastically as a perturbation of the current state of the system, and only the current state of the system, i.e. the system has no memory of earlier states (a *memory-less* process).

A state being a stochastic perturbation of the previous state means that given the conditions of the state at time *t* (e.g. *A(t)* = (position+velocity)) the *next* set of conditions *A(t+1)* (updated position+velocity) will be drawn from a distribution related to the earlier state. For example the *next* velocity can be a sample from a Gaussian distribution with mean equal to the *current* velocity. *A(t+1)* ~ N*(A(t), s)*

Bayes theorem: relates observed data to proposed models by allowing to calculate the *posterior distribution of model parameters* for a given prior and observed dataset (see glossary for term definition).

$$\textit{Posterior(data, model-parameters)} = \frac{\textit{Likelihood(data, model-parameters)} * \textit{Prior(model-parameters)}}{\textit{Evidence(data)}}$$

$$P(\theta|D,f) = \frac{P(D|\theta,f)\,P(\theta,f)}{P(D|f)}$$

keyconcepts

Markov Chain Monte Carlo: Is a method to sample a parameter space that is based on Bayes theorem. The MCMC samples the ***joint posterior*** of the parameters in the model (up to a constant, the *evidence*, probability of observing your data under any model parameter choice, which is generally not calculable). Thus we can get posterior median, confidence intervals, covariance, etc... The algorithm is:

1. starting at some location in the parameter space propose a new location as a Markovian perturbation of the current location

2. if the proposal posterior is better than the posterior at the current location update your position (and save the new position in the chain)

3. if the proposal posterior is worse than the posterior at the current location update your position with some probability $\boldsymbol{\alpha}$

The choice of the proposal distribution and rule $\boldsymbol{\alpha}$ for accepting the new step in the chain have to satisfy the ***ergodic*** condition, that is: given enough time the entire parameter space would be sampled. (***Detailed Balance*** is a sufficient condition for ergodicity)

If the chain is Markovian and the proposal distribution is *ergodic the entire parameter space is sample, given enough time, with sampling frequency proportional to the posterior distribution*

Different MCMC algorithms: while all MCMC algorithm share the structure above the choice of proposal and the acceptance probability are different for different MCMC algorithms.

Metropolis Hastings MCMC is the first and most common MCMC with acceptance proportional to the ratio of posteriors: ***α~posteriorNew/posteriorCurrent.*** This becomes problematic when the posterior has multiple peaks (may not explore them all) or parameter are highly covariant (may take a very long time to converge)

Convergence: It is crucial to confirm that your chains have converged and your parameter space is properly sampled, but it is also very difficult to do it. Methods include checking for stationarity of the chain means and low auto correlation in the chains. The beginning of the chain is typically removed as the chains require a minimum number of steps to move away from the initial position effectively.

- **Stochastic**: random, following any distribution
- **PDF**: probability distribution function *P(x)* describes the *relative* likelihood of sample *x* compared
- **CDF**: cumulative distribution function - the probability that a value drawn from a distribution will be smaller than *x* $F(x)= \int_{-}^{x} P(x)$
- **Marginalize**: integrate along a dimension
- **Gaussian distribution**: a distribution with PDF $N(\mu,\sigma)= \frac{1}{\sqrt{2\pi}\,\sigma} e^{-\frac{(x-\mu)^2}{\sigma^2}}$
- **Chi Squared χ2:** a model fitting method based on the provable fact that the function $\sum_{i=1}^{N} \frac{(M-D)^2}{\sigma^2} \sim \chi^2_{DOF}$

(under proper assumption) follows a χ2 distribution

- **Likelihood**: in Bayes theorem its the term indicating the probability of the data under the model for a choice of parameters. More generally it can be thought of the probability of the parameters given the data
- **Posterior**: the probability of data given model calculated by Beyes theorem as likelihood * prior / evidence
- **Evidence**: the probability of the data given a model marginalized over all parameters
- **Prior**: prior, or otherwise obtained, knowledge about the problem which indicates how likeli the model parameter are for any value
- **Markovian process**: a process whose next stage depends stochastically on the current state only
- **Ergodic**: a process that given enough time would visit all location of the space
- **Markov Chain**: an N dimensional sequence of values of each parameter of the N-dim parameter space that is explored by an MCMC

glossary

While My MCMC Gently Samples

Bayesian modeling, Computational Psychiatry, and Python

A blog by

https://twiecki.io

VP of data science at Quantopian

resources

resources

Information Theory, Inference, and Learning Algorithms

David J.C. MacKay, 2003

Numerical Recipes

Bill Press+ 1992 (+)

Ensemble samplers with affine invariance

Jonathan Goodman and Jonathan Weare 2010

Slides on sampling from distributions

Paul E. Johnson 2015

Bill Press (Numerical Recipes) Video

proving how Metropolis-Hastings sutisfied Detail Balance

EMCEE readme

provides high level discussion, references, suggestion on parameter choices

D. Foreman-Mackey, D. Hogg, D. Lang, J. Goodman+ 2012

dan.iel.fm/emcee/current/

reading

emcee

The MCMC Hammer

emcee is an extensible, pure-Python implementation of Goodman & Weare's Affine Invariant Markov chain Monte Carlo (MCMC) Ensemble sampler. It's designed for Bayesian parameter estimation and it's really sweet!