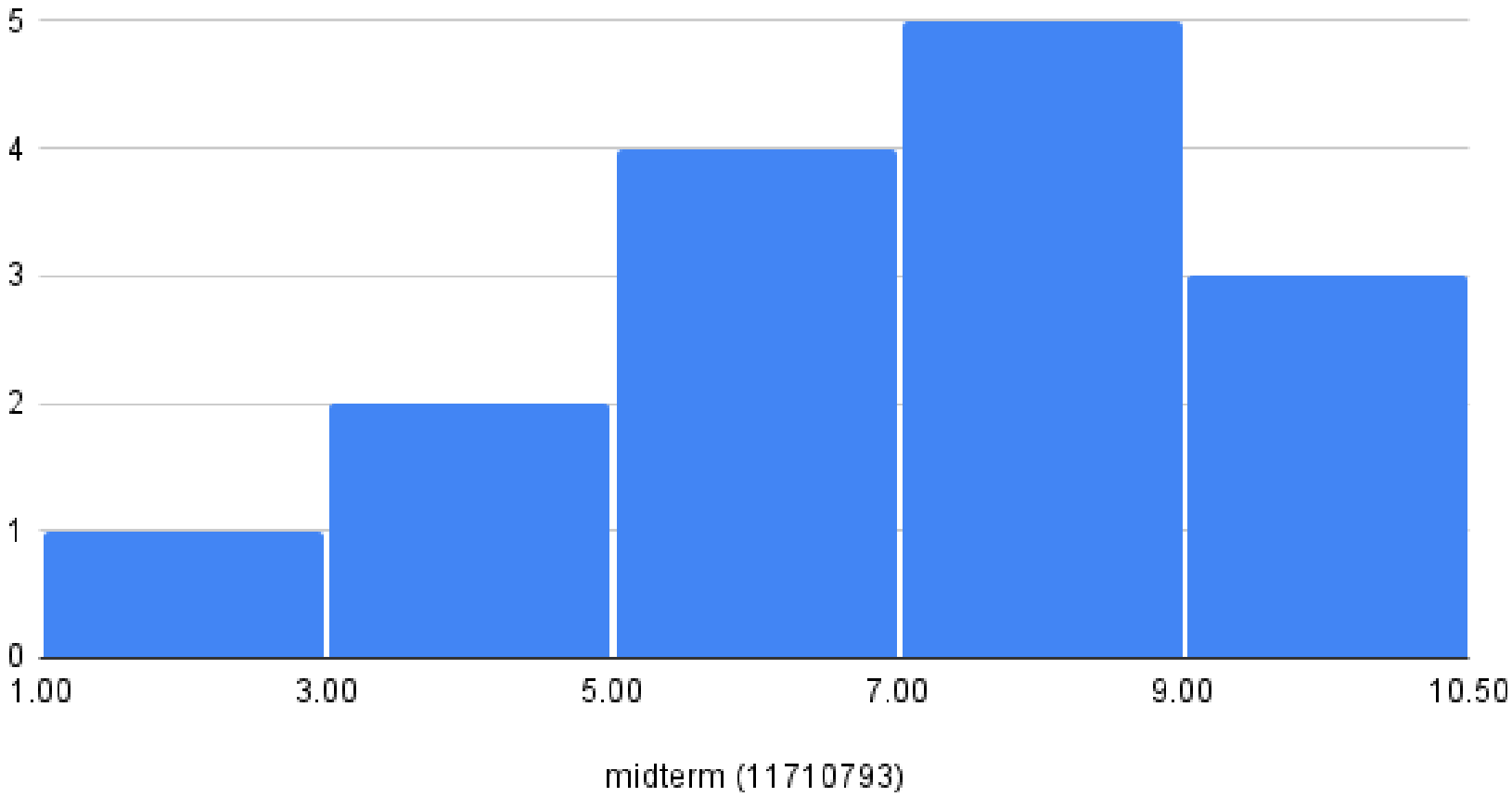
data science for (physical) scientists 6

fitting models to data - MCMC

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

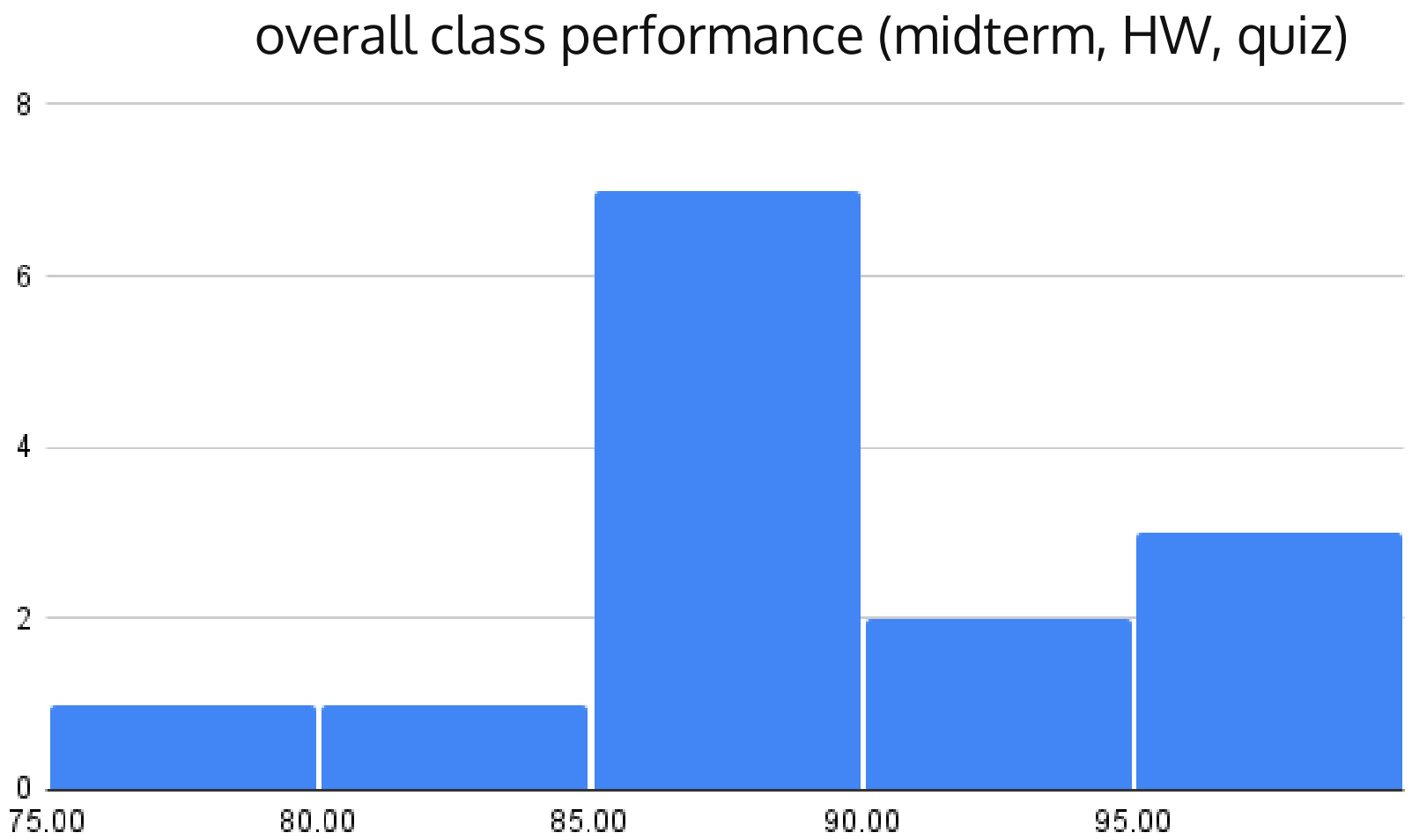
Grades are based on

- **5% pre-class questions**
- **15% class performance and participation (get up and code!!)**
- **40% homework**
- ~~**15% midterm**~~
- **30% final**



Grades are based on

- **5% pre-class questions**
- **15% class performance and participation (get up and code!!)**
- **40% homework**
- ~~**15% midterm**~~
- **30% final**



this slide deck

http://bit.ly/dsps__6

MIDTERM RULES:

WORK ALONE

Ask questions on Slack at
https://dsps21.slack.com/archives/C02JSD2GTPC

The instructions will be made available on
github at 4PM Friday (today)

The STRICT NO EXCEPTIONS delivery
deadline is Sunday 4PM

Delivery it by SHARING THE COLAB
NOTEBOOK - NOT POSTING ON GITHUB

Since you have 48 hours I expect a reasonably neat presentation including discussion of results, descriptions of figures, comments on choices you made along the way...

fitting models to data

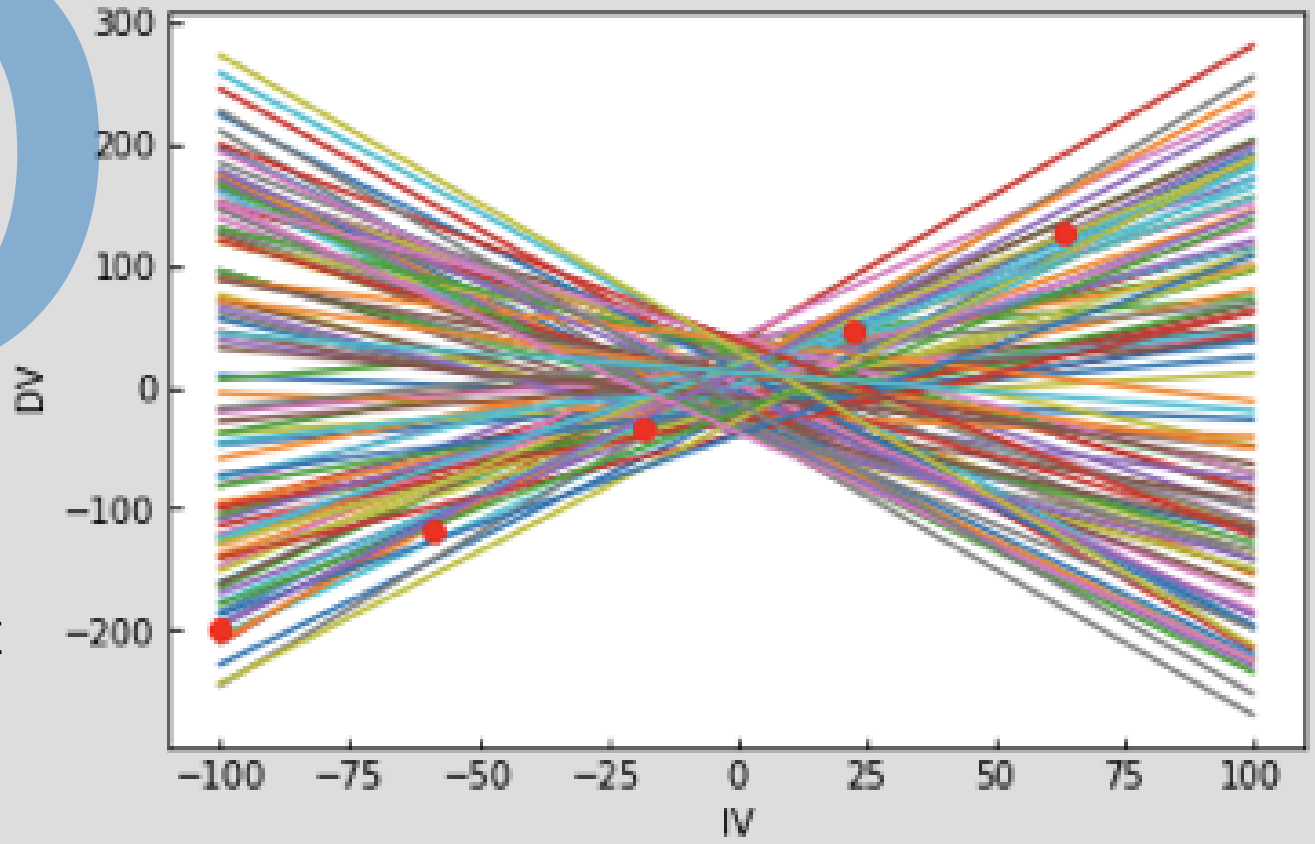
recap

1

choose your model :

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

line model: *ax+b*



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

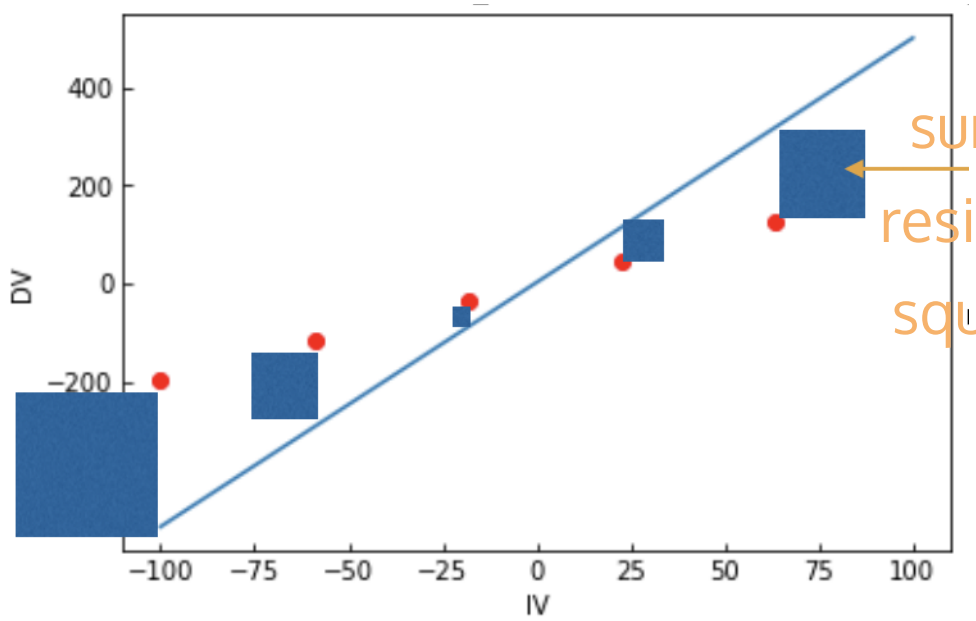
- no model is right
- some models are useful

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



$$L^2 = \sum_i (y_i - f_i)^2$$

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

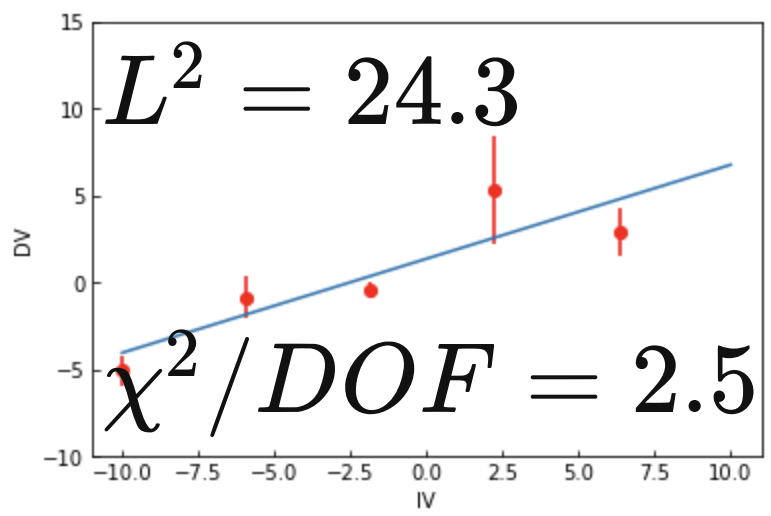
$$\frac{\sum \frac{(y_i - (mx_i + b))^2}{2\sigma_i^2}}{DOF} \sim \chi^2(dof = 1)$$

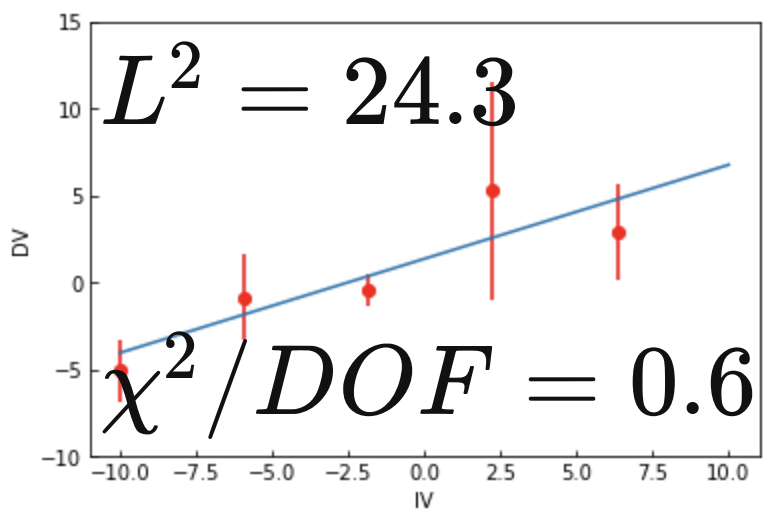
recap

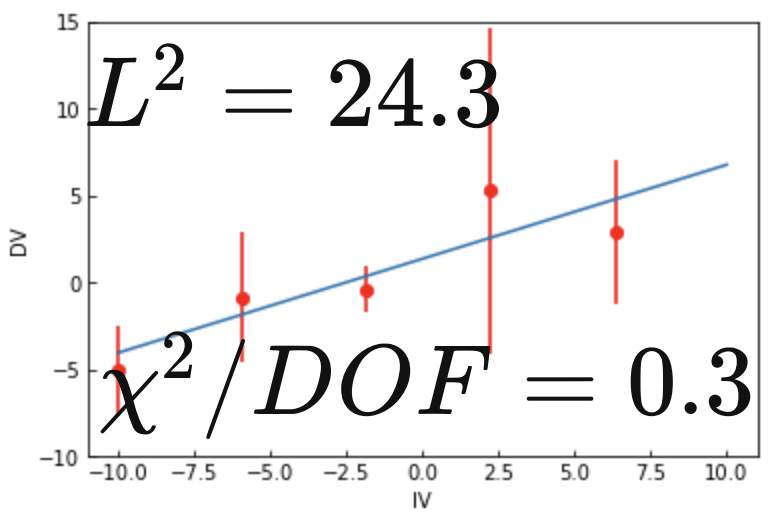
3

evaluate the quality of your model

again: many options!







recap

3

evaluate the quality of your model

again: many options!

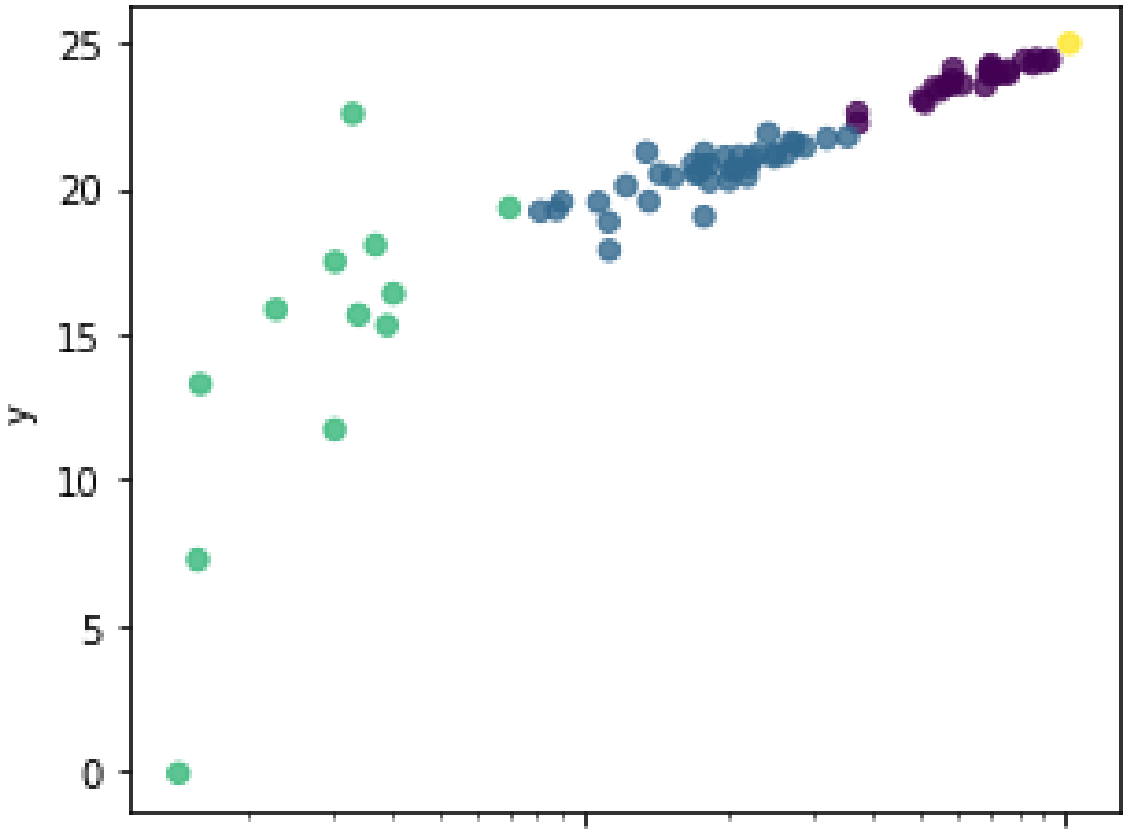
homoscedastic :

the uncertainty is the same for all data points

heteroscedastic:

the uncertainty different for each datapoint

(almost always the case in physics!)



scatter may **dependent on exogenous variable**

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

recap 3

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

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

how we minimize: gradient descent

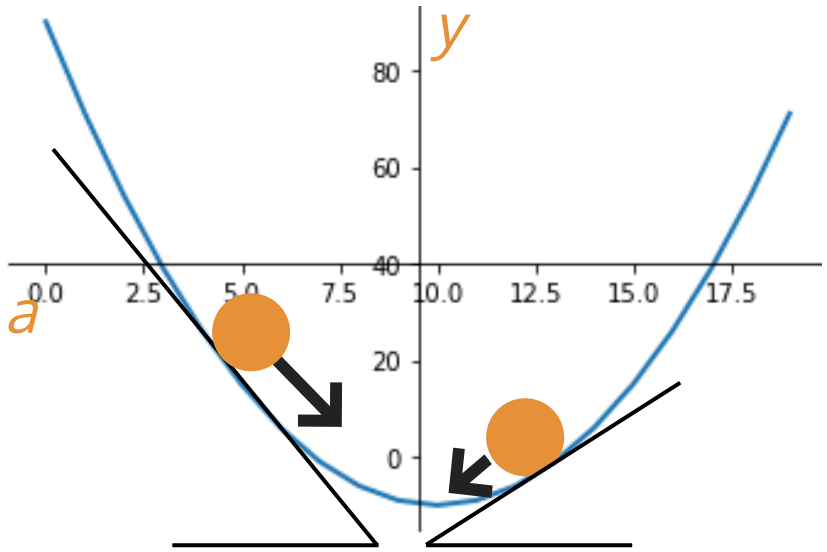
Optimization

recap

4

Gradient Descent

$$p_{\text{new}} := p_{\text{old}} - \eta \nabla Q(p)$$



how we minimize: gradient descent

Optimization

Gradient Descent

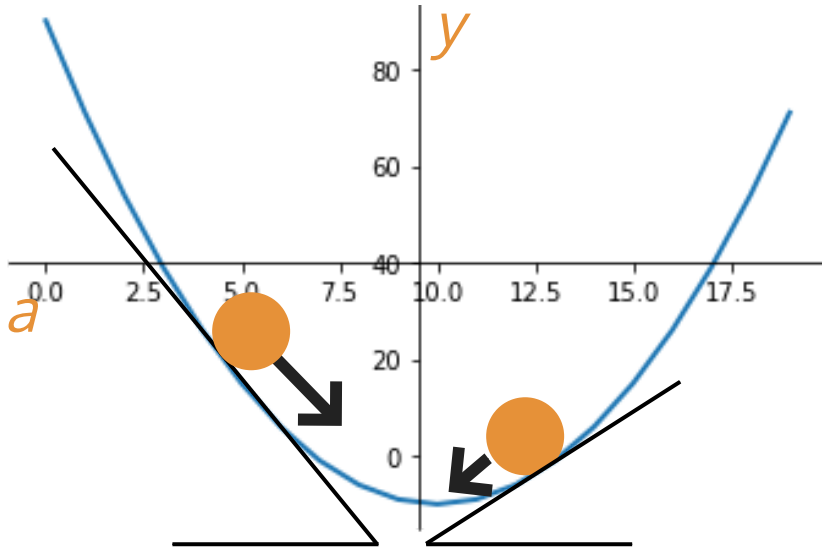
$$p_{\text{new}} := p_{\text{old}} - \eta \nabla Q(p)$$

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

WHILE gradient is negative:

calculate the loss function at the current position loss_curr

choose a new random position as **new =** $\theta_{\text{old}} - \eta \nabla Q(p)$



how we minimize: gradient descent

Optimization

Stochastic Gradient Descent

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

WHILE convergence criterion is me:

calculate the loss function at the current position loss_curr

choose a new random position **new = θn = (m,b)**

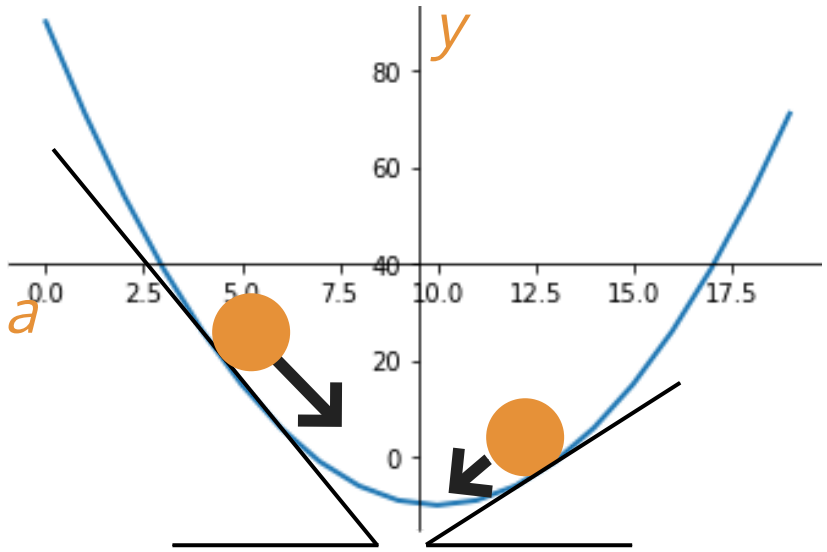
calculate the loss function at the current position loss_new

IF loss_curr > loss_new

move to **θn**

ELSE

pass //do nothing



1 - how to choose a model: principle of parsimony

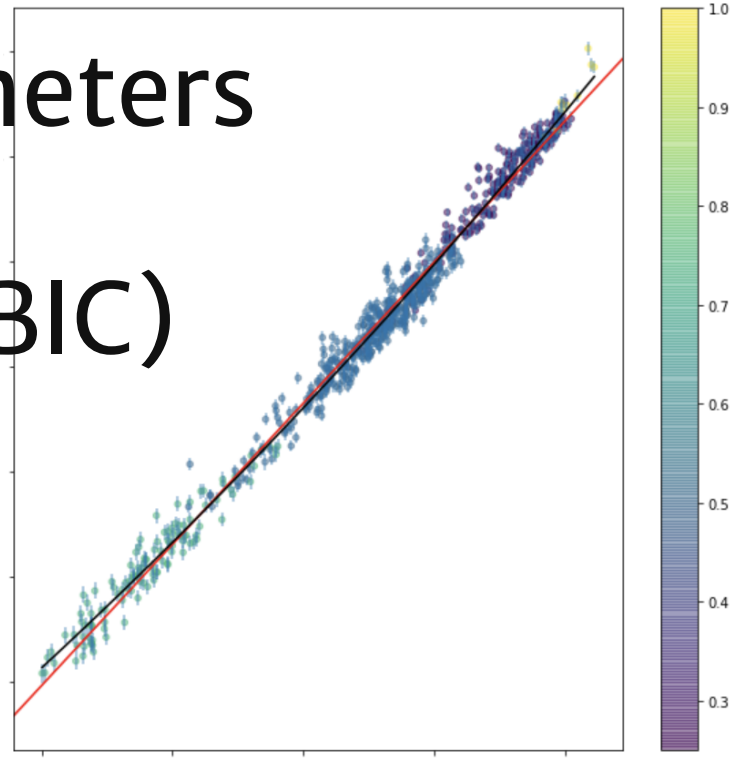
fitting a model to data

2 ->1 order equation

3 - uncertainties in the fit parameters

4 - comparing models (LR, AIC, BIC)

5 - MCMC



1 Likelihood

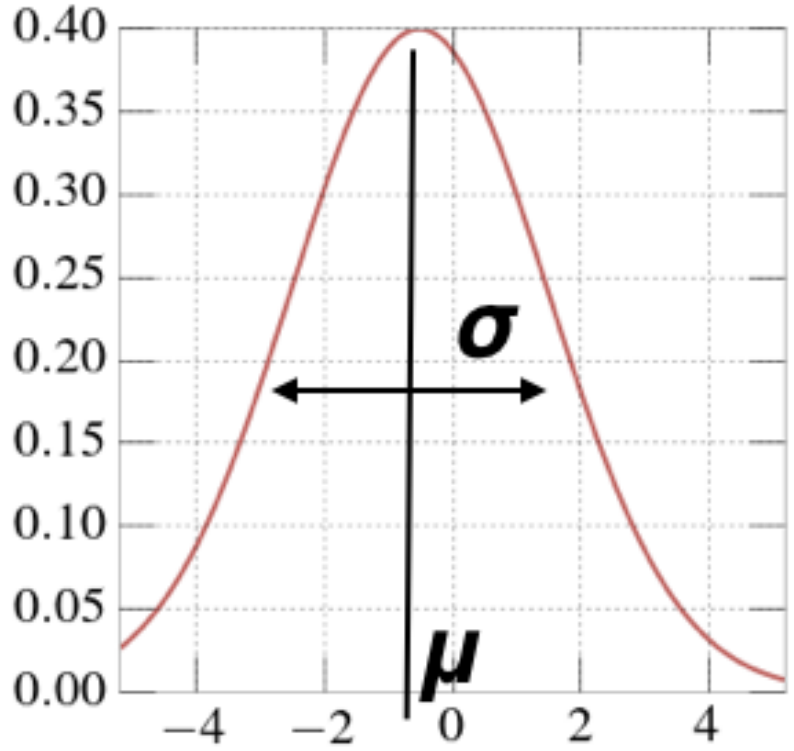
Probability vs Likelihood

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

Probability vs Likelihood

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

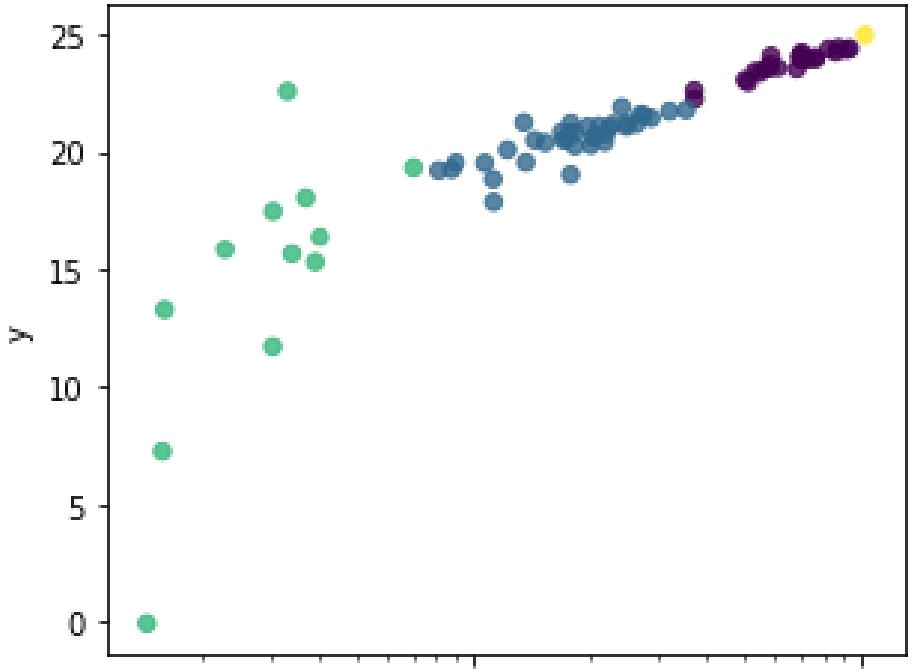
Gaussian distribution: $P(x|\mu,\sigma)$



Probability vs Likelihood

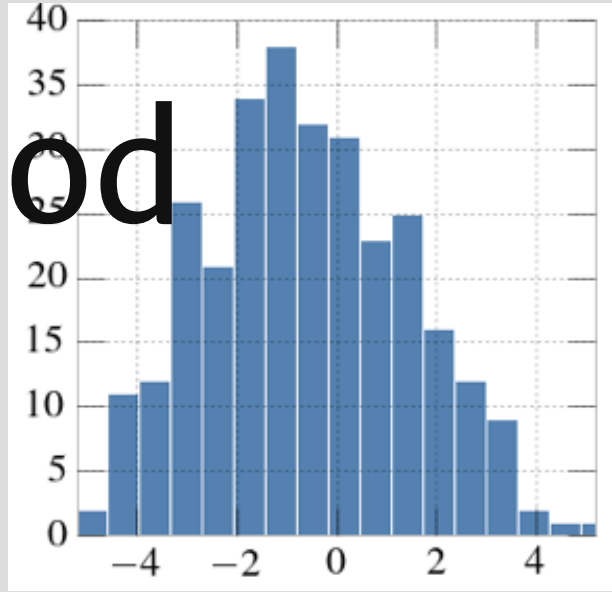
Probability of data given model $P(x|\theta)$

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

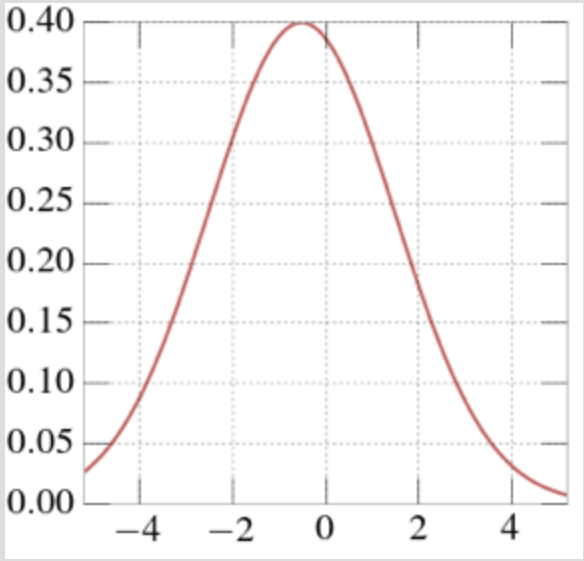


Noisy line function: $P(\vec{y}|\vec{x}, a, b, \mu, \sigma(x))$

Probability vs Likelihood

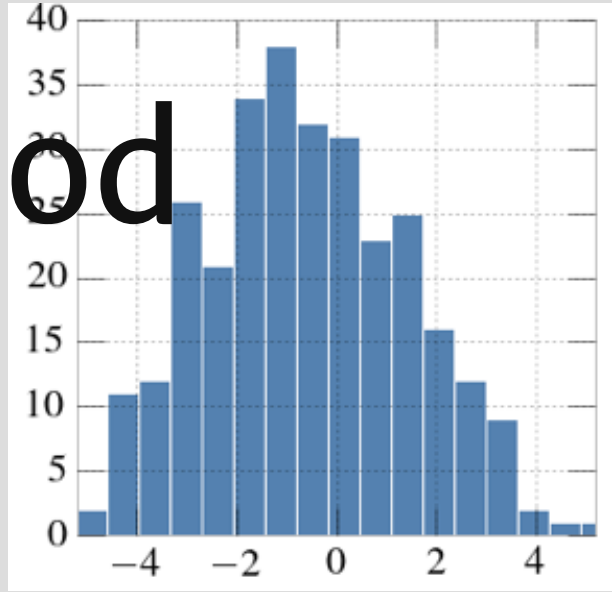


Probability of ***data*** given ***model*** $P(x|\theta)$

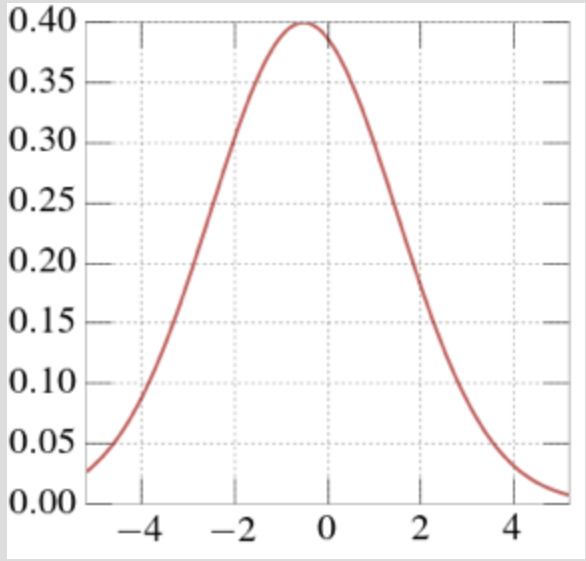


Probability of ***model*** given ***data*** $L(\theta|x)$

Probability vs Likelihood



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



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

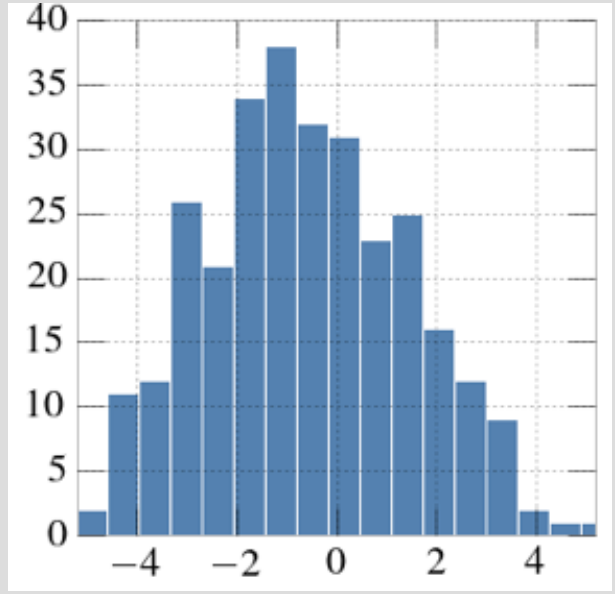
Same formula! different meaning

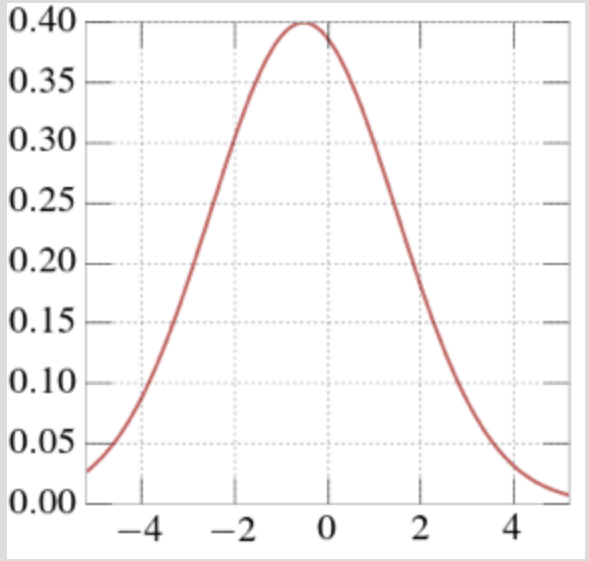
Likelihood

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***

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

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***

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

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

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***

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

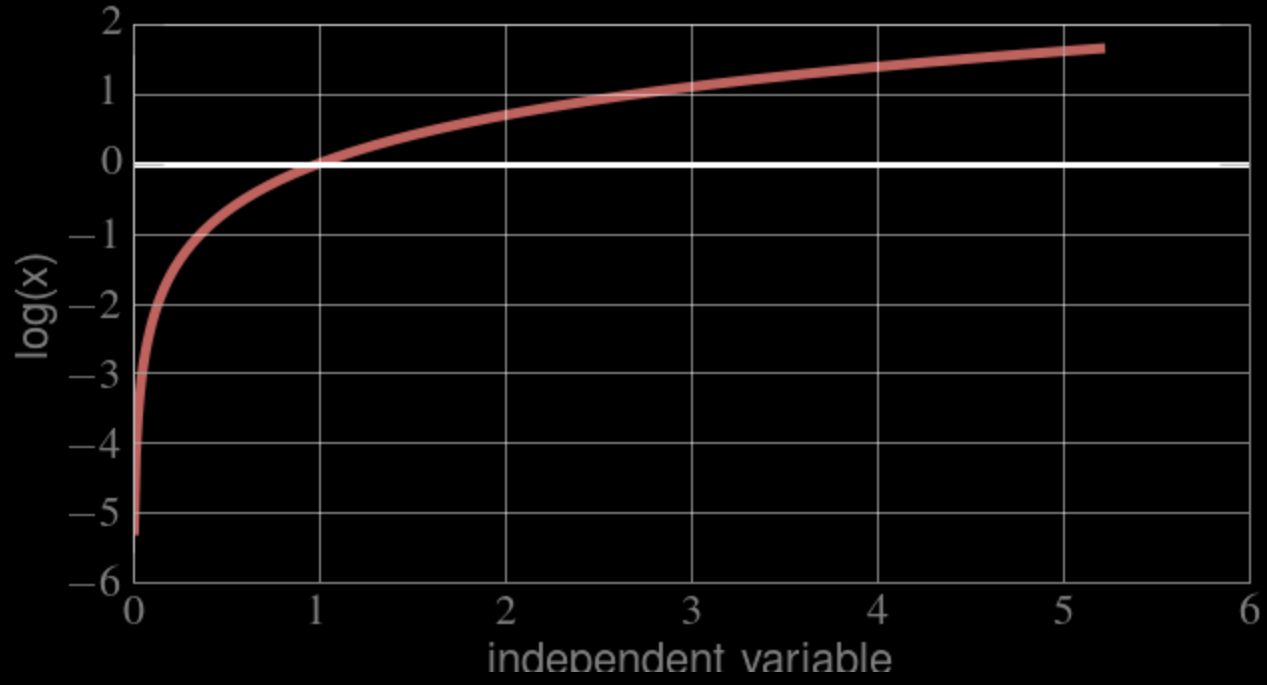
$$\text{Find } (\mu^*,\sigma^*) \; || - log\left(L_{\mu^*,\sigma^*}\right) = min\left(-log\left(L_{\mu,\sigma}\right)\right)$$

Logarithms

MONOTONICALLY INCREASING

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

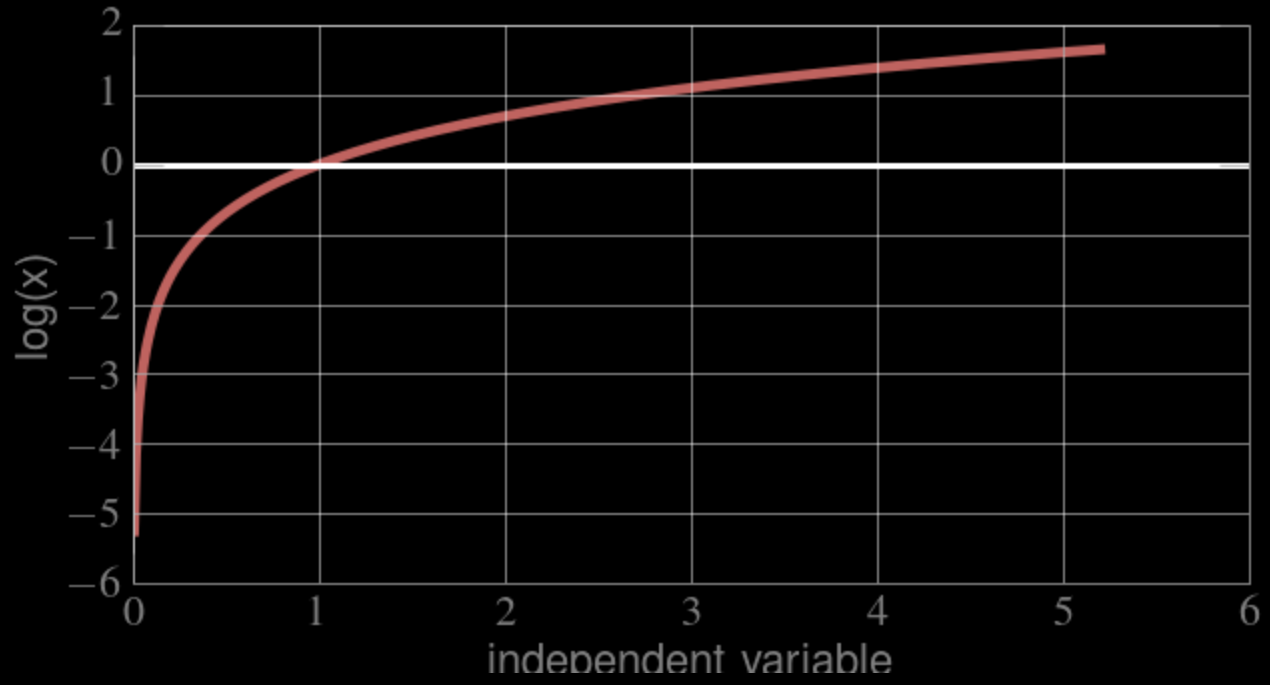
the location of the maximum is the same!



Logarithms

MONOTONICALLY INCREASING

SUPPORT : $(0, \infty]$

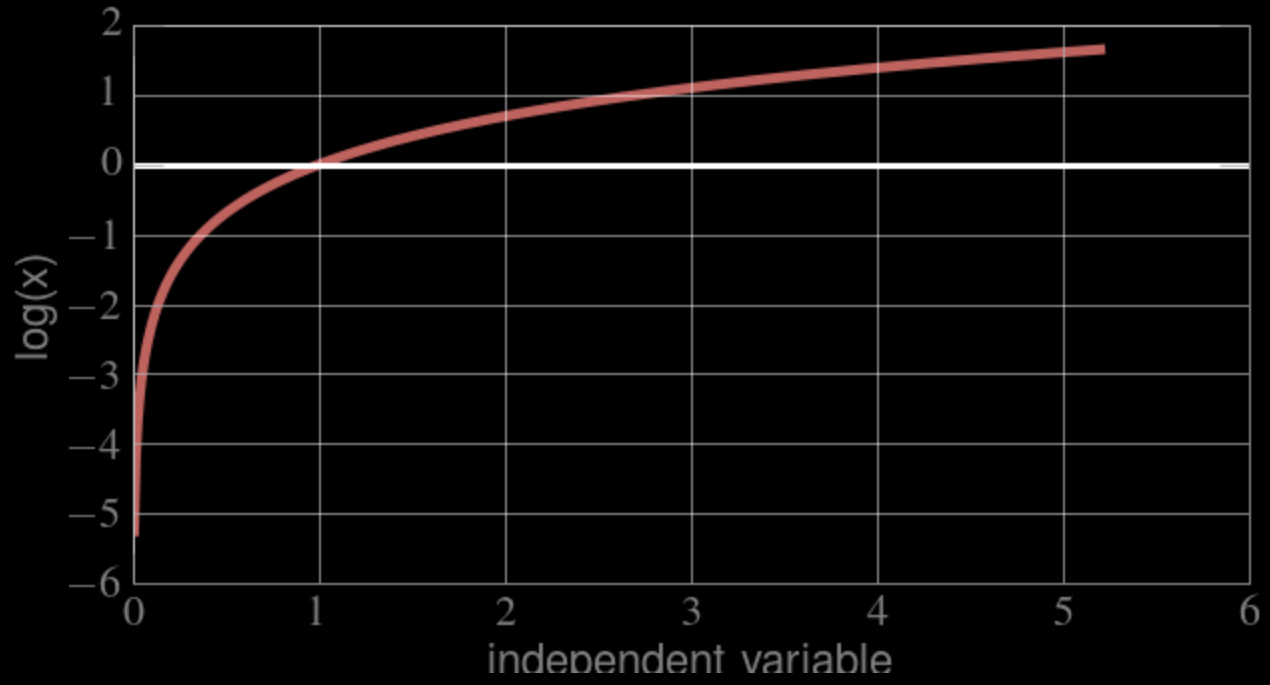


Logarithms

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

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

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

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

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

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

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

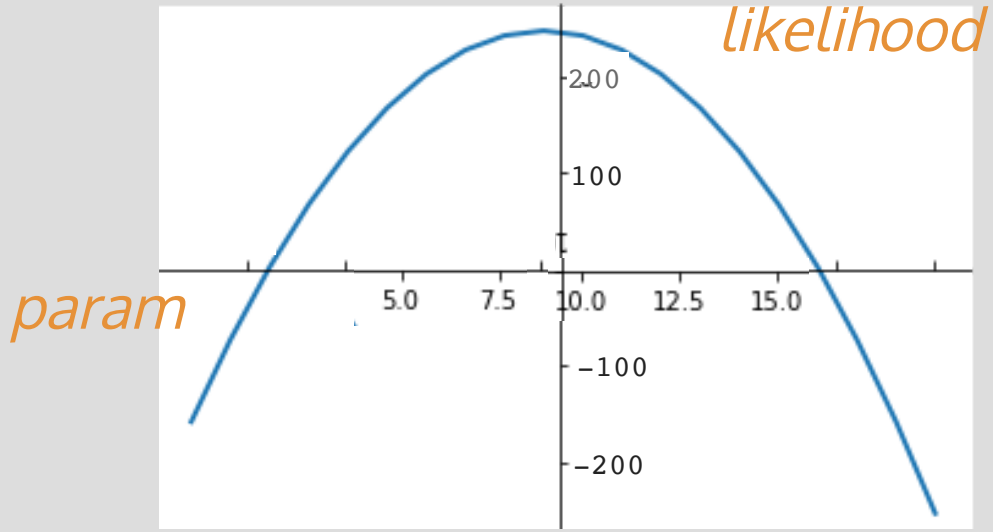
$$L(m, b|\vec{y}) = \prod_i^N p_i(y_i|x_i, \sigma_i, m, b)$$

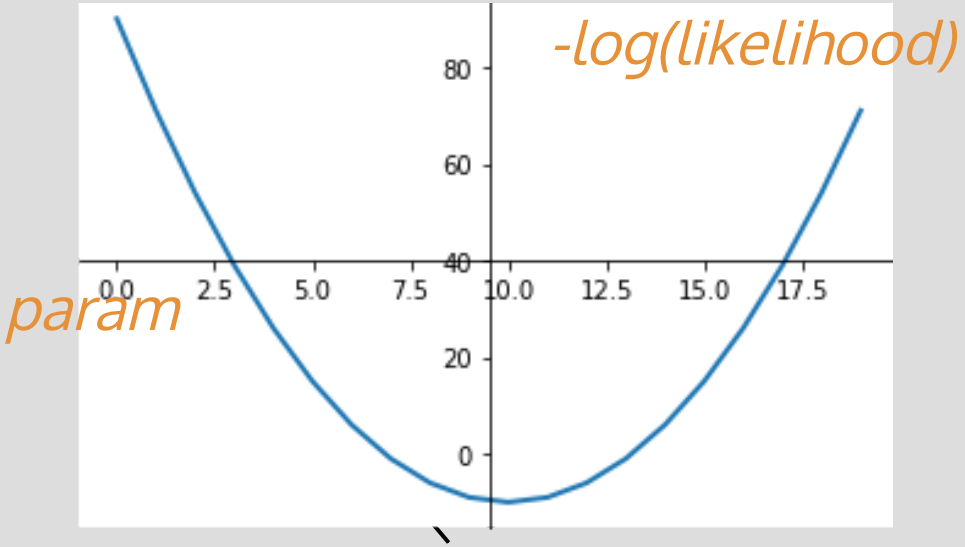
$$p(y_i) = \frac{1}{\sigma_i\sqrt{2\pi}} \exp - \frac{(y_i - (mx_i + b))^2}{2\sigma_i^2}$$

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

think about the likelihood surfece...

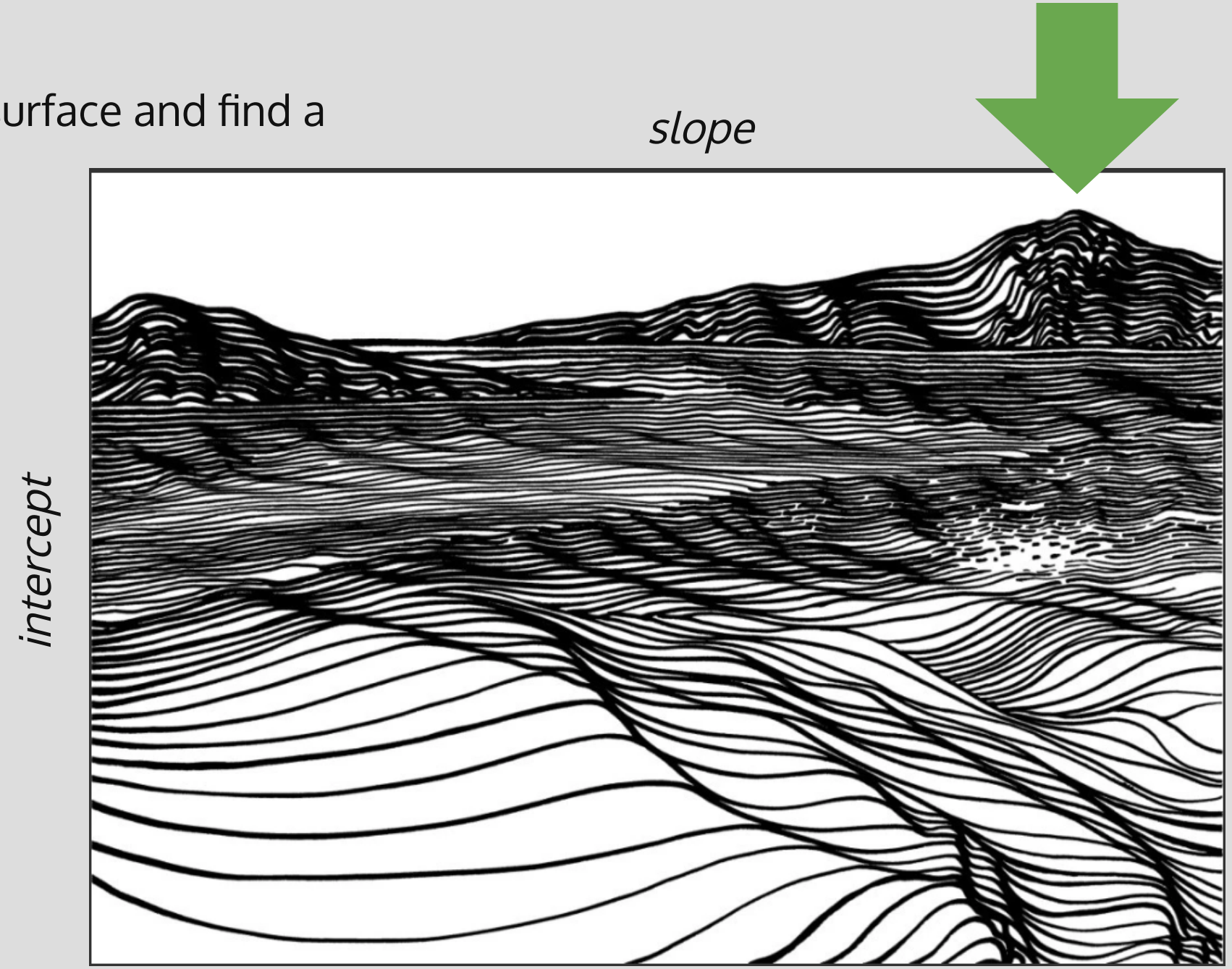
you want to explore the surface and find a peak





think about the likelihood surfece...

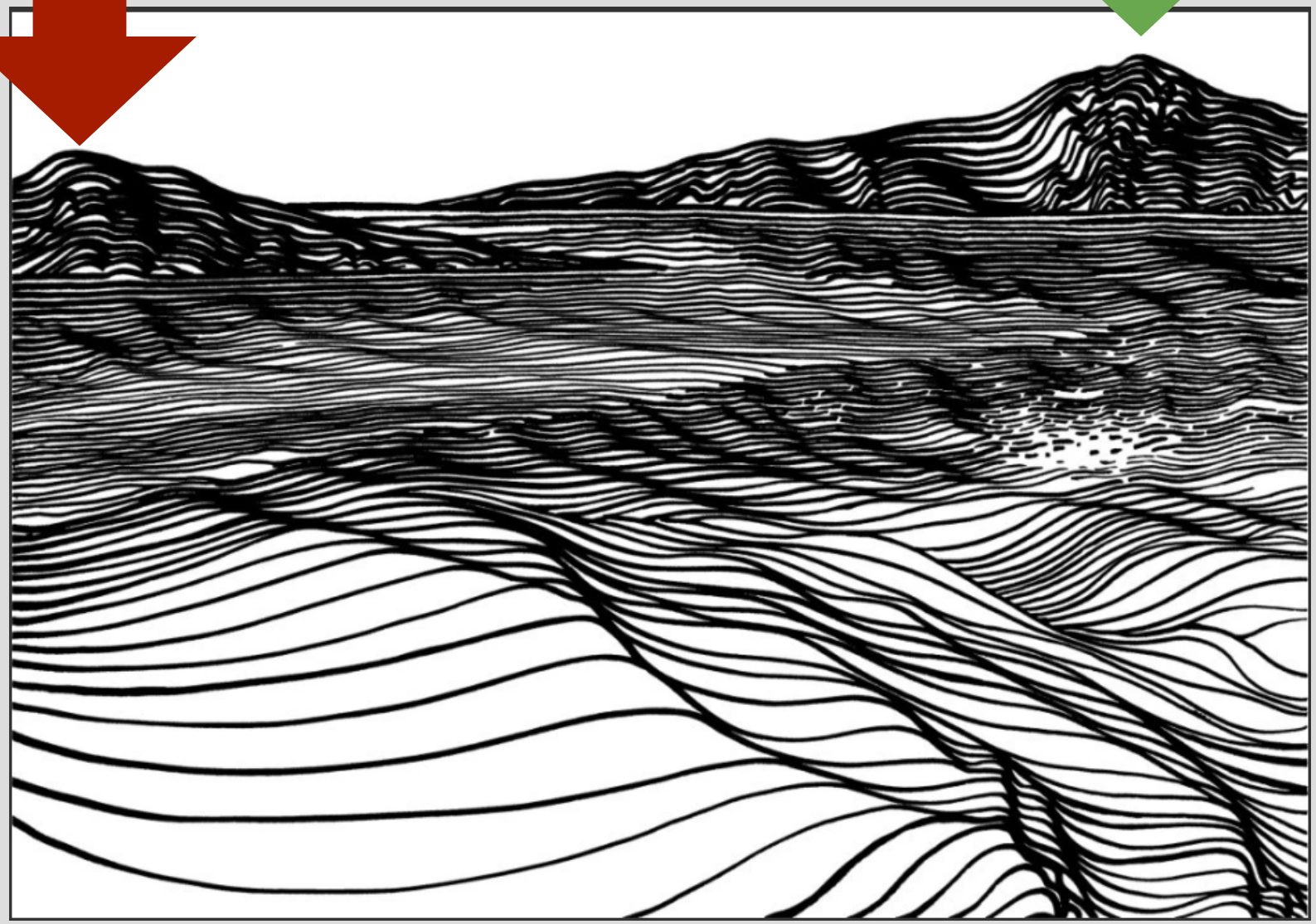
you want to explore the surface and find a peak



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

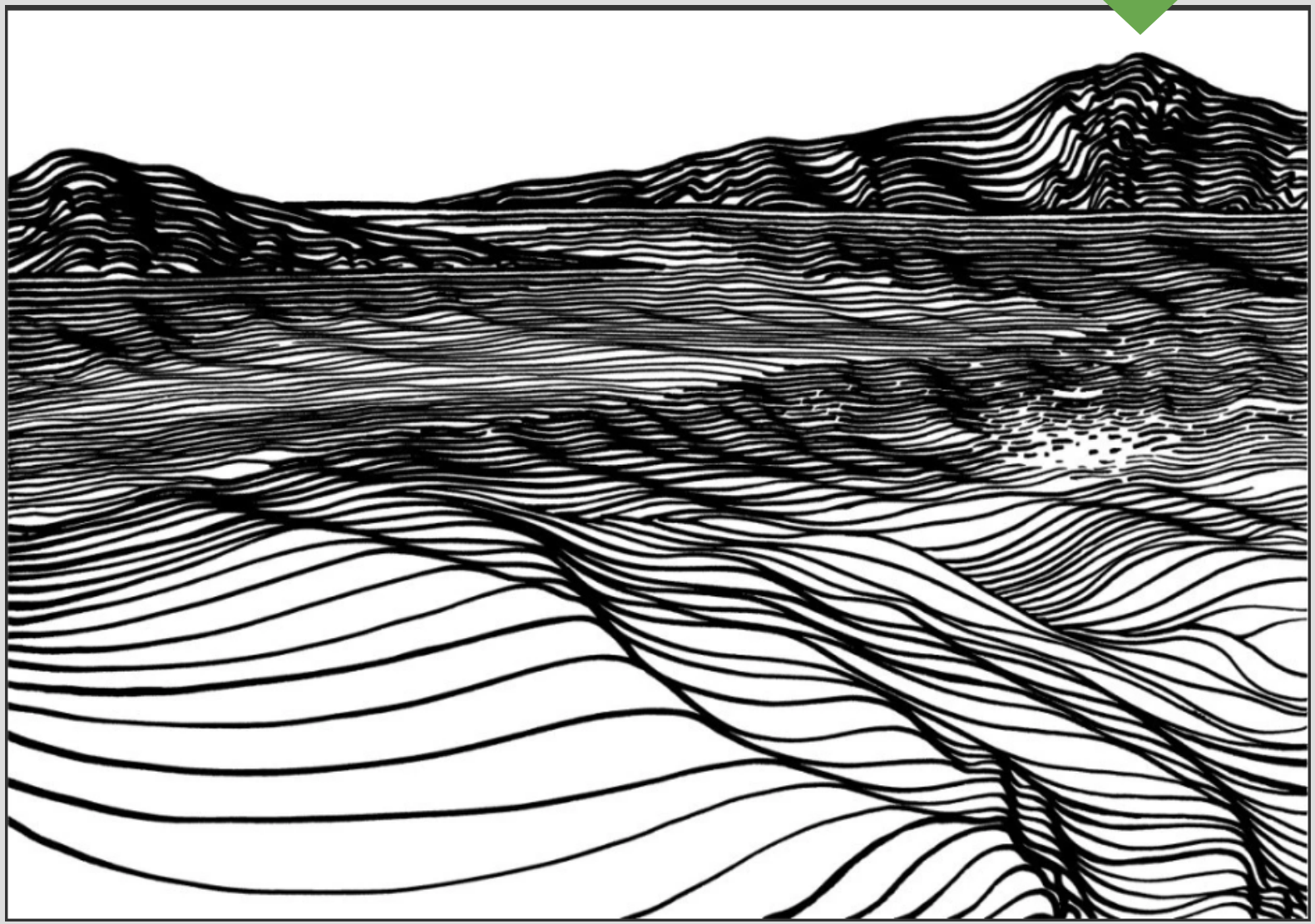
The problem of fitting models to data reduces to finding the

maximum *likelihood*

of the data given the model

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

-log(*likelihood*)



21

Model Selection principles

what model should I choose?

the answer truly depends on what you are modeling
for and what domain knowledge is available

except:

the principle of parsimony

the principle of parsimony

or Ockham's razor

Pluralitas non est ponenda sine neccesitate

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”

the principle of parsimony

the earth is round,

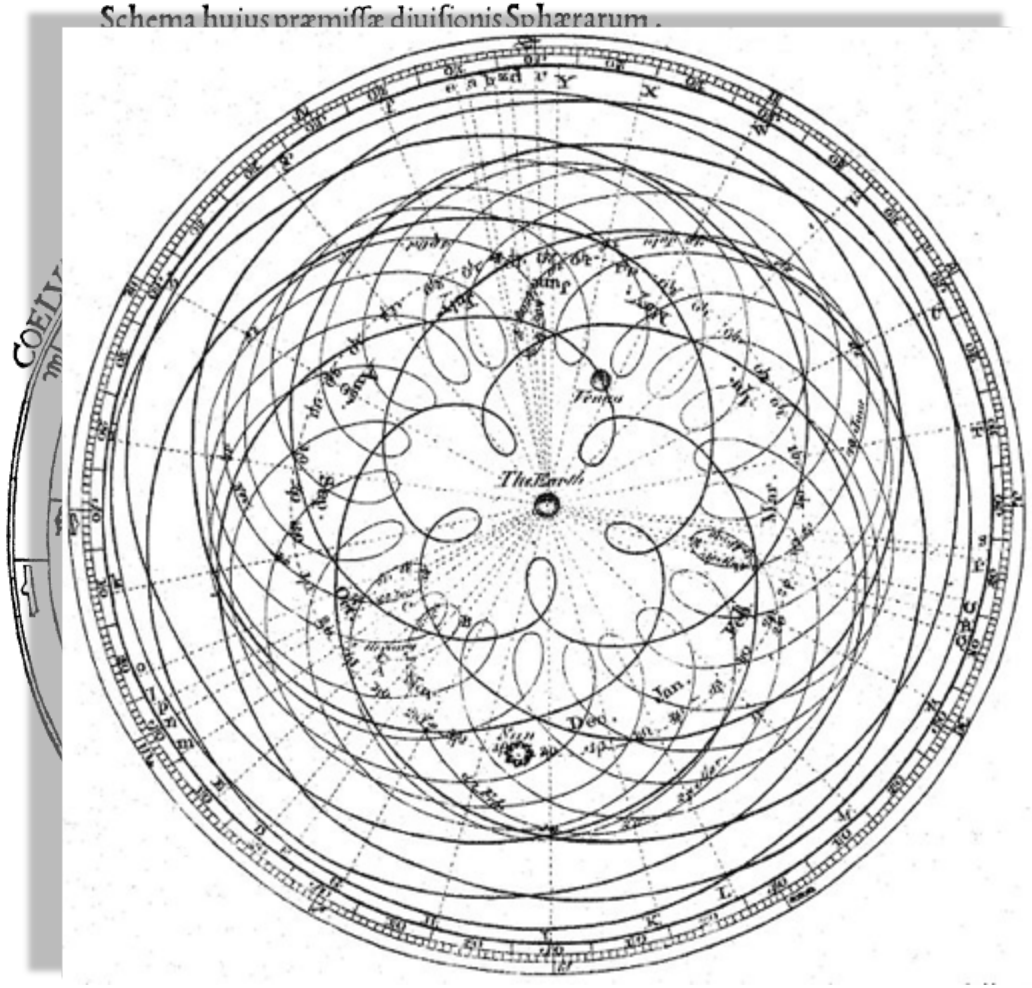
and it orbits around the sun

Geocentric models are intuitive:

from our perspective we see the Sun moving, while we stay still

Peter Apian, *Cosmographia*, Antwerp, 1524 from Edward Grant, "Celestial Orbs in the Latin Middle Ages", *Isis*, Vol. 78, No. 2. (Jun., 1987).

the principle of parsimony



Encyclopaedia Brittanica 1st Edition

Dr Long's copy of Cassini, 1777

the earth is round,

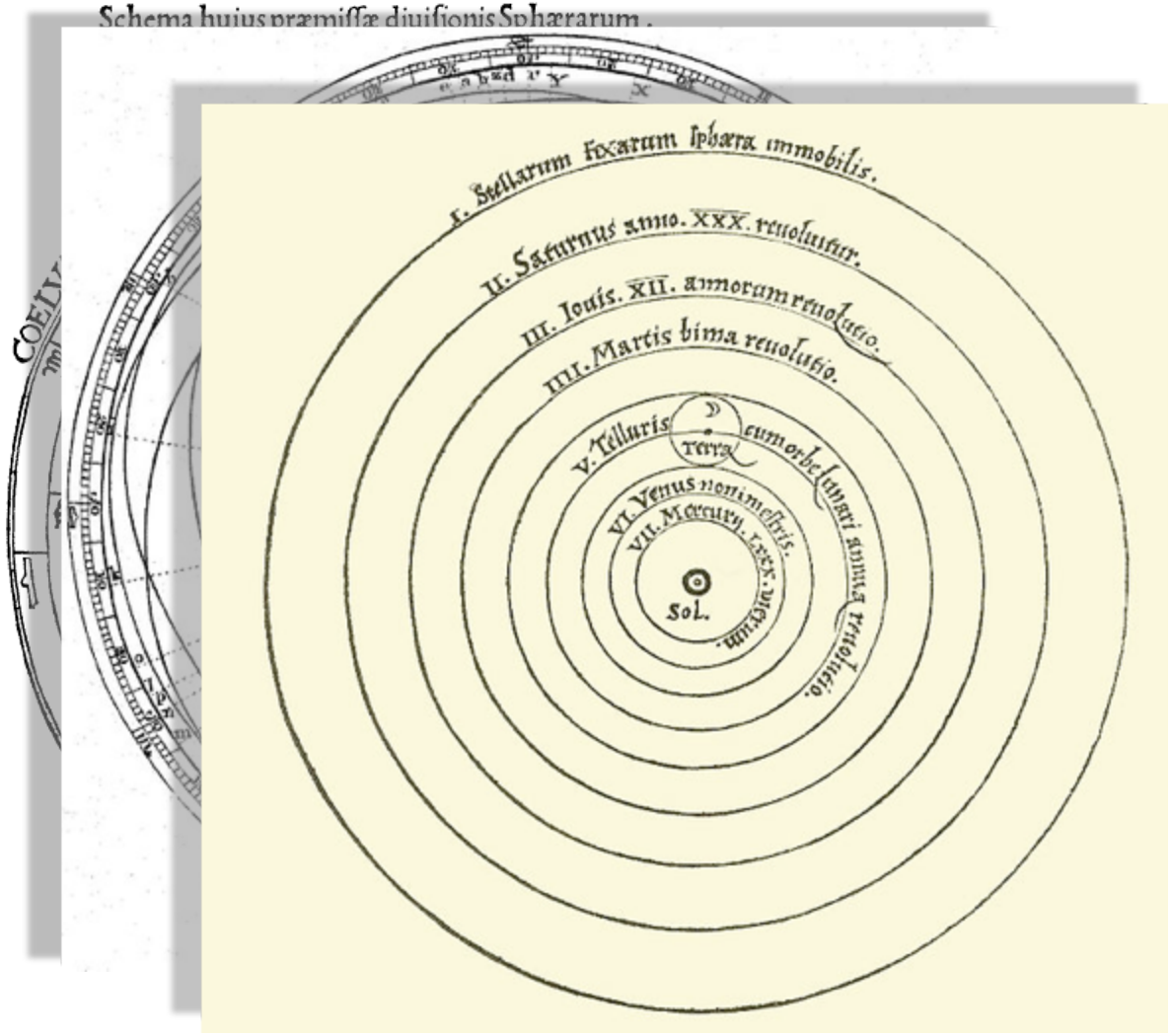
and it orbits around the sun

As observations improve

this model can no longer fit the data!

not easily anyways...

the principle of parsimony



the earth is round,

~~and it orbits around the sun~~

A new model that is much simpler fit the data just as well

(perhaps though only until better data comes...)

Heliocentric model from Nicolaus Copernicus' *De revolutionibus orbium coelestium*.

the principle of parsimony

or Ockham's razor

Pluralitas non est ponenda sine neccesitate

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”

the principle of parsimony

or Ockham's razor

Pluralitas non est ponenda sine neccesitate

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 ***simpler one***”

the principle of parsimony

or Ockham's razor

Pluralitas non est ponenda sine neccesitate

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***”

the principle of parsimony

Science and Statistics George E. P. Box (1976)

Journal of the American Statistical Association, Vol. 71, No. 356, pp. 791-799.

Since all models are wrong the scientist cannot obtain
a "correct" one by excessive elaboration. On the
contrary following William of Occam he should seek
an economical description of natural phenomena

Since all models are wrong the scientist must be alert
to what is importantly wrong.

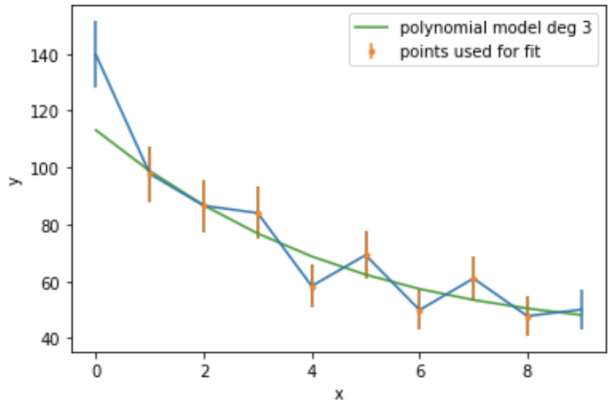
the principle of parsimony

Careful!

Increasing the model's degrees freedom allows a "better fit" in the in-sample set

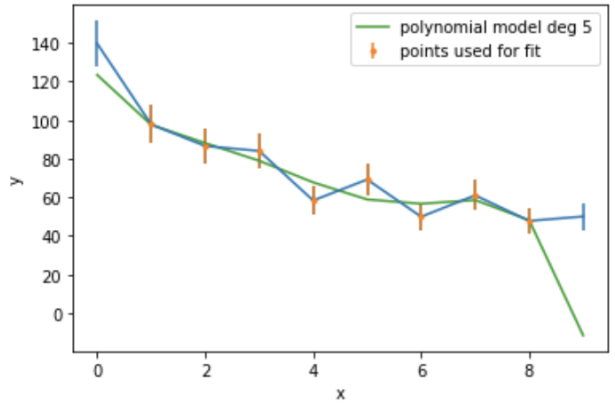
```
fitpoly(3, x, y, yerr);
```

```
L2 insample 3.3e+02
L2 outofsample 709.80
```



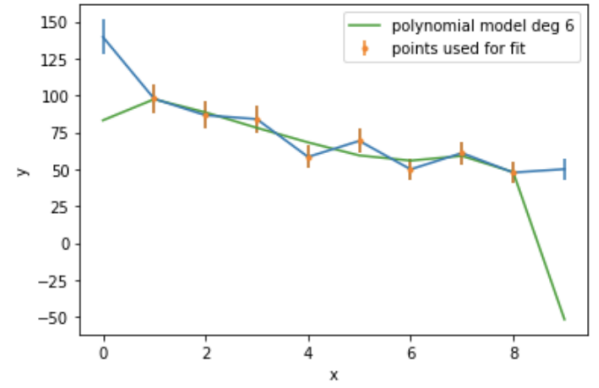

```
[ ] fitpoly(5, x, y, yerr);
```

```
L2 insample 2.7e+02
L2 outofsample 263.94
```




```
[ ] fitpoly(6, x, y, yerr);
```

```
L2 insample 2.7e+02
L2 outofsample 3189.61
```



22

Model Selection methods

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()

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

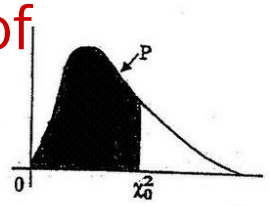
```
from scipy.stats.distributions import chi2
def likelihood_ratio(llmin, llmax):
    return(-2*(llmax-llmin))

LR = likelihood_ratio(L1,L2)

p = chi2.sf(LR, dof)
# dof: difference in number of parameters
print ('p: %.30f' % p)
# LR is chi squared distributed:
# p represents the probability that this result
# (or a more extreme result than this)
# would happen by chance
```

HOW DO I CHOOSE A MODEL?

difference in number of parameters between the 2 models



Given two models which is preferable?

Likelihood-ratio tests

likelihood ratio statistics LR

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

statsmodels.model.compare_lr_ratio()

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

The table below gives the value x_0^2 for which $P[x^2 < x_0^2] = P$ for a given number of degrees of freedom and a given value of P.

Degrees of Freedom	Values of P									
	0.005	0.010	0.025	0.050	0.100	0.900	0.950	0.975	0.990	0.995
1	---	---	0.001	0.004	0.016	2.706	3.841	5.024	6.635	7.879
2	0.01	0.020	0.051	0.103	0.211	4.605	5.991	7.378	9.210	10.597
3	0.072	0.115	0.216	0.352	0.584	6.251	7.815	9.348	11.345	12.838
4	0.207	0.297	0.484	0.711	1.064	7.779	9.488	11.143	13.277	14.860
5	0.412	0.554	0.831	1.145	1.610	9.236	11.070	12.833	15.086	16.750
6	0.676	0.872	1.237	1.635	2.204	10.645	12.592	14.449	16.812	18.548
7	0.989	1.239	1.690	2.167	2.833	12.017	14.067	16.013	18.475	20.278
8	1.344	1.646	2.180	2.733	3.490	13.362	15.507	17.535	20.090	21.955
9	1.735	2.088	2.700	3.325	4.168	14.684	16.919	19.023	21.666	23.589
10	2.156	2.558	3.247	3.940	4.865	15.987	18.307	20.483	23.209	25.188
11	2.603	3.053	3.816	4.575	5.578	17.275	19.675	21.920	24.725	26.757
12	3.074	3.571	4.404	5.226	6.304	18.549	21.026	23.337	26.217	28.300
13	3.565	4.107	5.009	5.892	7.042	19.812	22.362	24.736	27.688	29.819
14	4.075	4.660	5.629	6.571	7.790	21.064	23.685	26.119	29.141	31.319
15	4.601	5.229	6.262	7.261	8.547	22.307	24.996	27.488	30.578	32.801
16	5.142	5.812	6.908	7.962	9.312	23.542	26.296	28.845	32.000	34.267
17	5.697	6.408	7.564	8.672	10.085	24.769	27.587	30.191	33.409	35.718
18	6.265	7.015	8.231	9.390	10.865	25.989	28.869	31.526	34.805	37.156
19	6.844	7.633	8.907	10.117	11.651	27.204	30.144	32.852	36.191	38.582
20	7.434	8.260	9.591	10.851	12.443	28.412	31.410	34.170	37.566	39.997

MLTSA: model selection

model selection is also based on the minimization of a quantity. Several quantities are suitable:

AIC Optimism and likelihood maximization on the training set

BIC Bayese theorem

MLD Shannon 1948: A Mathematical Theory of Communication

a theory to find fundamental limits on signal processing and communication operations such as data compression

MLTSA:

AIC, BIC, & MDL

Likelihood: **Model Performance**.

$$\mathrm{AIC} = -\frac{2}{N}\log(L) + \frac{2}{N}k$$

number of parameters:

Model Complexity

Akaike information criterion (AIC) .

Based on $\lim_{N\to\infty}(-2E(\log Pr_{\hat{\theta}}(Y))) = -\frac{2}{N}E\ \log(L) + d\frac{2}{N}$

where $Pr_{\hat{\theta}}(Y)$ is a family of function (=densities) containing the correct (=true) function and $\hat{\theta}$ is the set of parameters that maximized the likelihood *L*

***L* is the likelihood of the data, *k* is the number of parameters,**

***N the number of variables*.**

MLTSA:

AIC, BIC, & MDL

Likelihood: **Model Performance**.

$$\mathrm{AIC} = -\frac{2}{N}\log(L) + \frac{2}{N}k$$

number of parameters:

Model Complexity

Akaike information criterion (AIC) .

Based on $\lim_{N\to\infty}(-2E(\log Pr_{\hat{\theta}}(Y))) = -\frac{2}{N}E\ \log(L) + \frac{2}{N}d$

where $Pr_{\hat{\theta}}(Y)$ is a family of function (=densities) containing the correct (=true) function and $\hat{\theta}$ is the set of parameters that maximized the likelihood *L*

***L* is the likelihood of the data, *k* is the number of parameters,**

***N the number of variables*.**

"-" sign in front of the log-likelihood: AIC shrinks for better models,

AIC ~ k => is linearly proportional to the number of parameters

MLTSA:

AIC, BIC, & MDL

Likelihood: **Model Performance**.

$$\mathrm{BIC} = -2\log(L) + \log(N)k$$

number of parameters:

Model Complexity

Bayesian information criterion (BIC) .

***L* is the likelihood of the data, *k* is the number of parameters,**

***N the number of variables*.**

stronger penalization of complexity (as long as N> e^2)

Bayes Factor

The derivation is very different:

$$\frac{P(M_m|D)}{P(M_l|D)} = \frac{P(M_m)}{P(M_l)} \cdot \frac{P(D|M_m)}{P(D|M_l)}$$

MLTSA:

AIC, BIC, & MDL

$$\mathrm{MDL} = -\log(L(\theta)) - \log(L(y|X,\theta))$$

Minimum Description Length (MDL) .

negative log-likelihood of the model parameters (*θ*) and the negative log-likelihood of the target values (*y*) given the input values (*X*) and the model parameters (*θ*).

also: log(L(θ)): number of bits required to represent the model,

log(L(y| X,θ)): number of bits required to represent the predictions on observations

minimize the encoding of the model and its predictions

derived from Shannon's theorem of information

MLTSA:

AIC, BIC, & MDL

$$\text{AIC} = -\frac{2}{N}\log(L) + \frac{2}{N}k$$

$$\text{BIC} = -2\log(L) + \log(N)k$$

$$\text{MDL} = -\log(L(\theta)) - \log(L(y|X,\theta))$$

Mathematically similar, though derived from different approaches. All used the same way: the preferred model is the model that minimized the estimator

implementation https://machinelearningmastery.com/probabilistic-model-selection-measures/

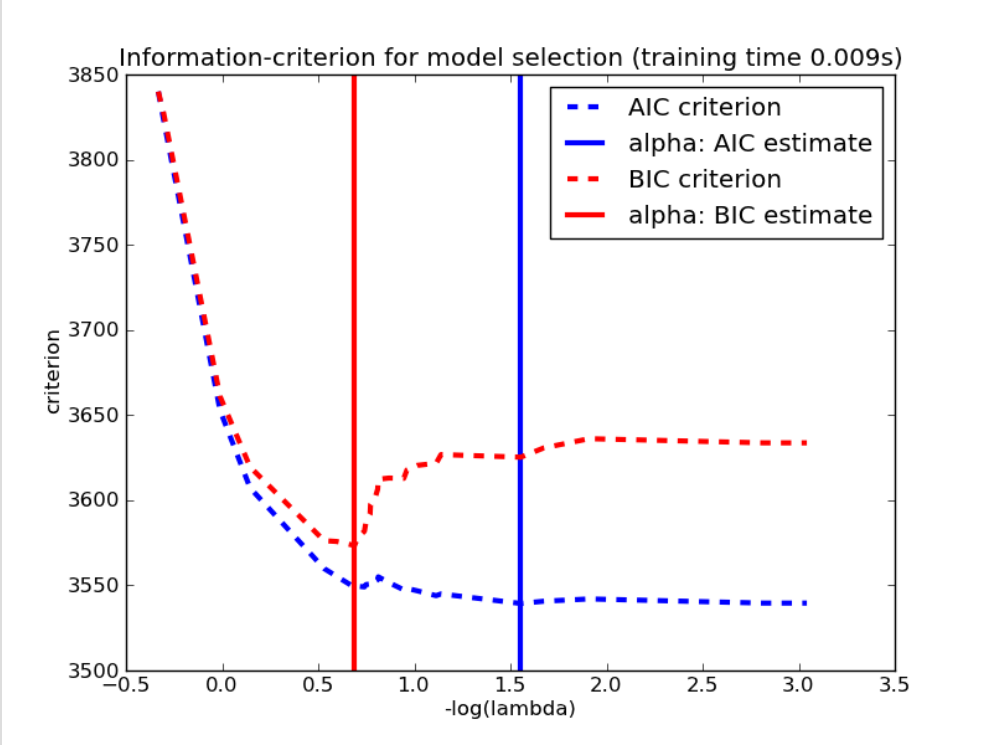
MLTSA:

AIC, BIC, & MDL

$$\mathrm{AIC} = -\frac{2}{N}\log(L) + \frac{2}{N}k$$

$$\mathrm{BIC} = -2\log(L) + \log(N)k$$

$$\mathrm{MDL} = -\log(L(\theta)) - \log(L(y|X,\theta))$$



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***

AIC - BIC

also consider at Akaike and Bayesian Information Criteria for not nested models: both are returned in a statsmodel fit

Dep. Variable:	y	**R-squared:**	0.982
Model:	OLS	**Adj. R-squared:**	0.982
Method:	Least Squares	**F-statistic:**	1816.
Date:	Tue, 22 Oct 2019	**Prob (F-statistic):**	1.97e-30
Time:	01:27:47	**Log-Likelihood:**	-78.850
No. Observations:	35	**AIC:**	161.7
Df Residuals:	33	**BIC:**	164.8
Df Model:	1		
Covariance Type:	nonrobust		

they are calculated combining the likelihood with a penalization for the extra parameters

generally both decrease with increasing increasing likelihood but you would look for the place where they start decreasing slowly as the "sweet spot" for your model

https://www.methodology.psu.edu/resources/AIC-vs-BIC/

MCMC

Mote Carlo Markov Chain

MCMC

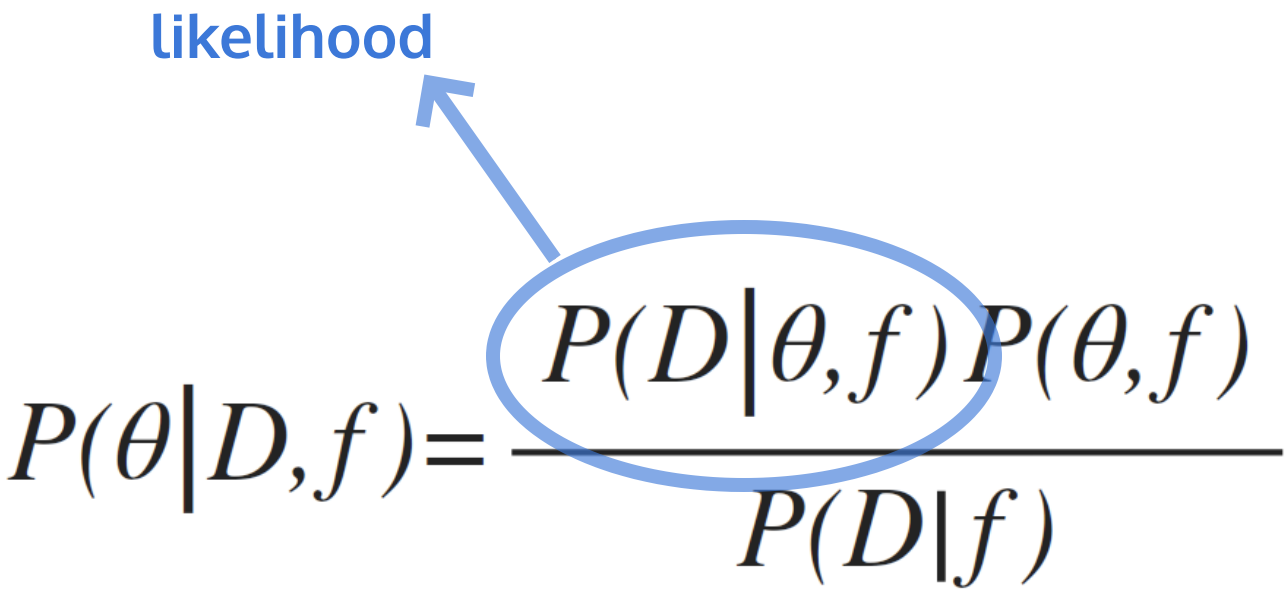
Mote Carlo Markov Chain

part 1: Bayes Theorem

Bayes Theorem

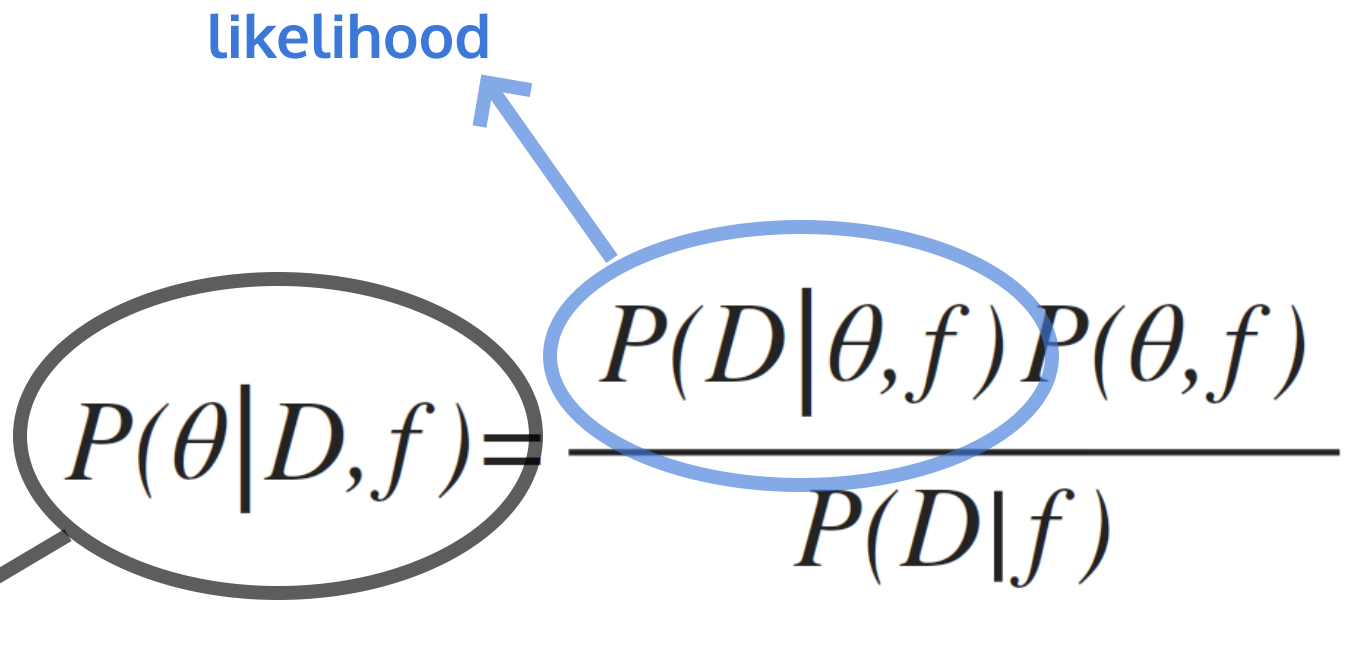
$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

Bayes Theorem



we are going to sample the likelihood:

Bayes Theorem

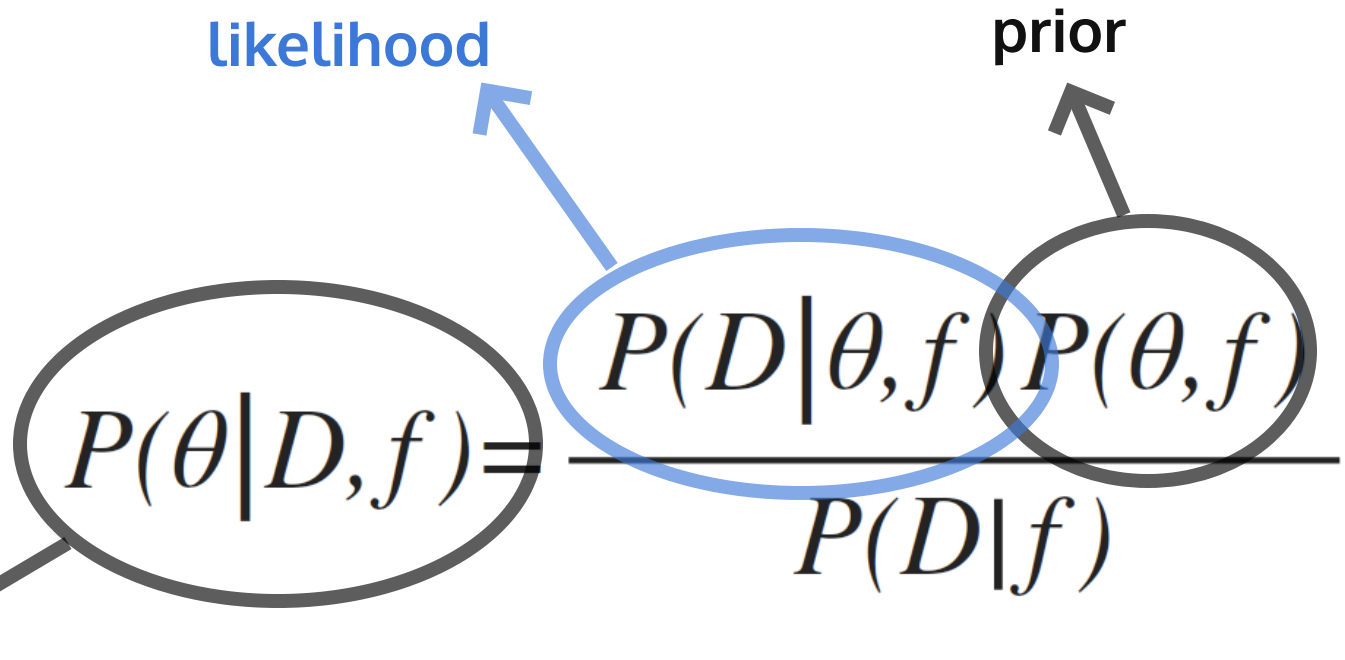


Definitions:

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



Definitions:

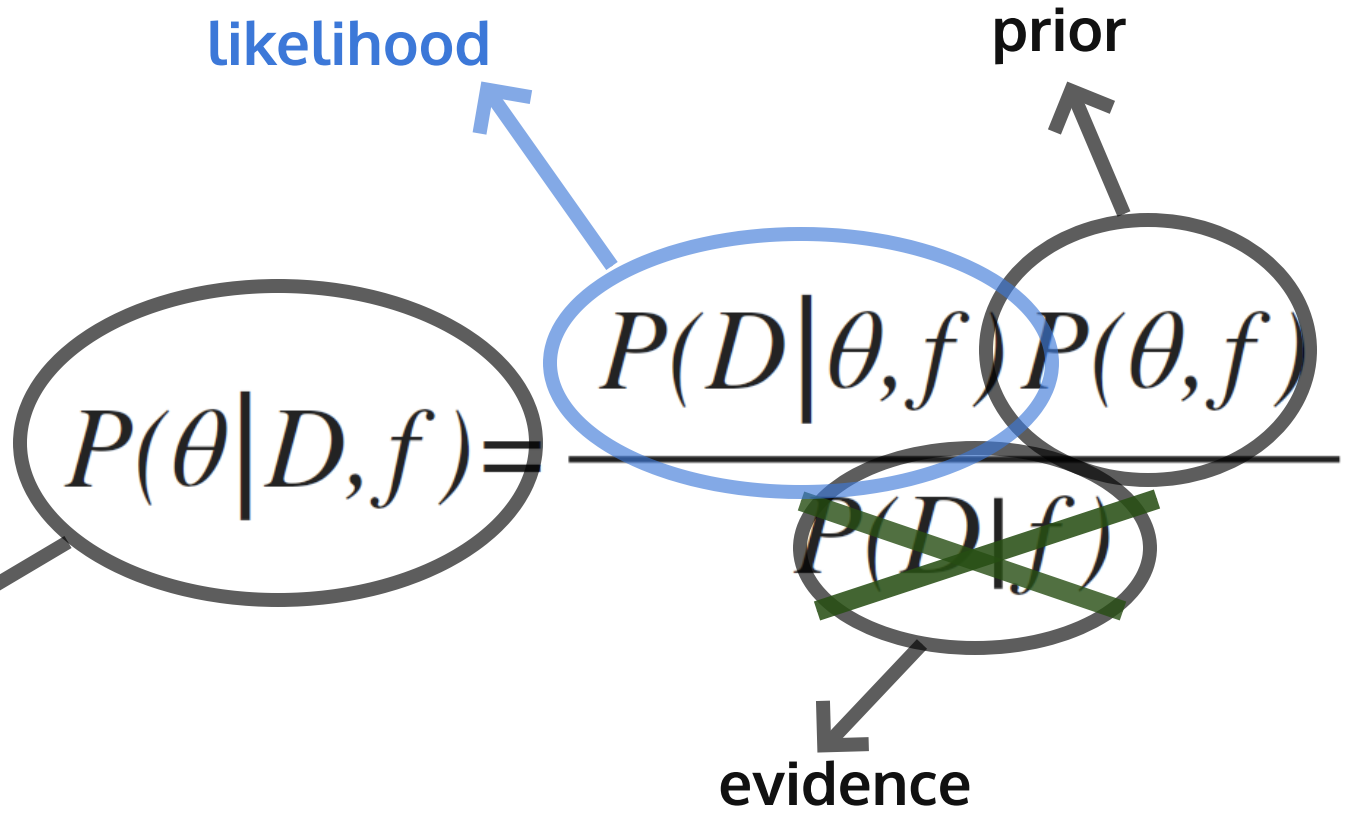
posterior: joint probability distributin of a parameter set (*m, b*) condition upon some data *D* and a model hypothesys *f*

prior: "intellectual" knowledge about the model parameters

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

$$P(\theta, f)$$

Bayes Theorem



Definitions:

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

prior: "intellectual" knowledge about the model parameters

evidence: marginal likelihood of data under the model

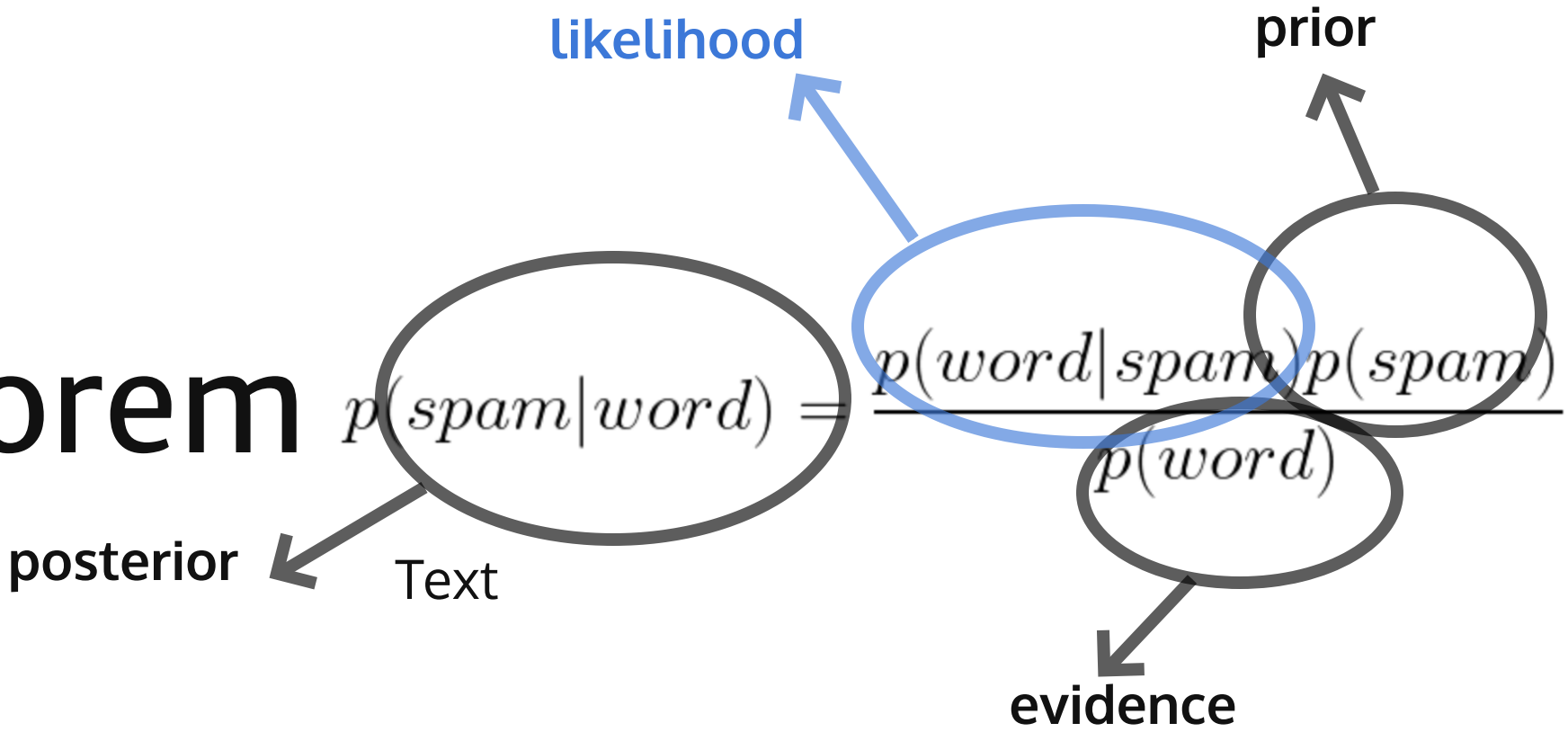
$$P(D|\theta, f)$$

$$P(\theta, f)$$

$$P(D|f) = \int P(D|\theta,f)P(\theta|f)d\theta$$

its constant in θ so we can ignore it!

Bayes Theorem



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

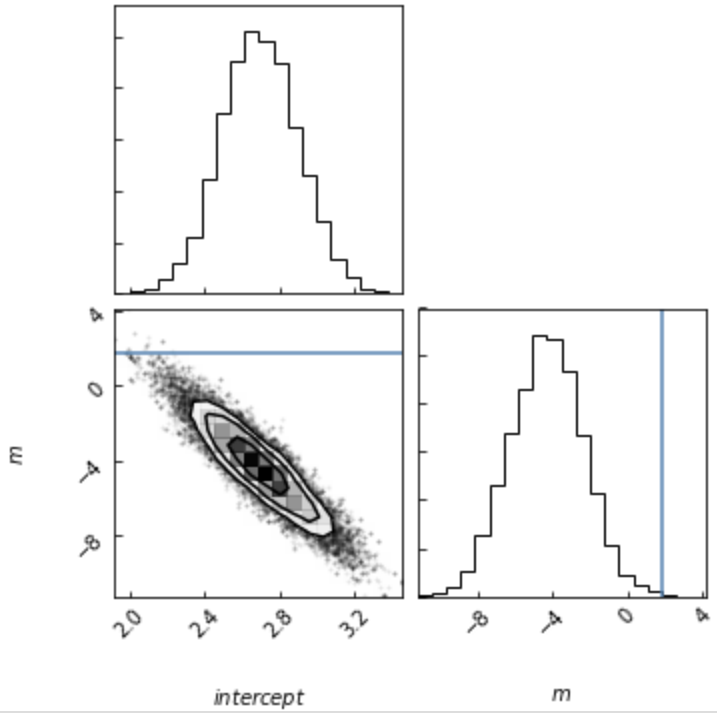
MCMC

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

posterior

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

triangle plot

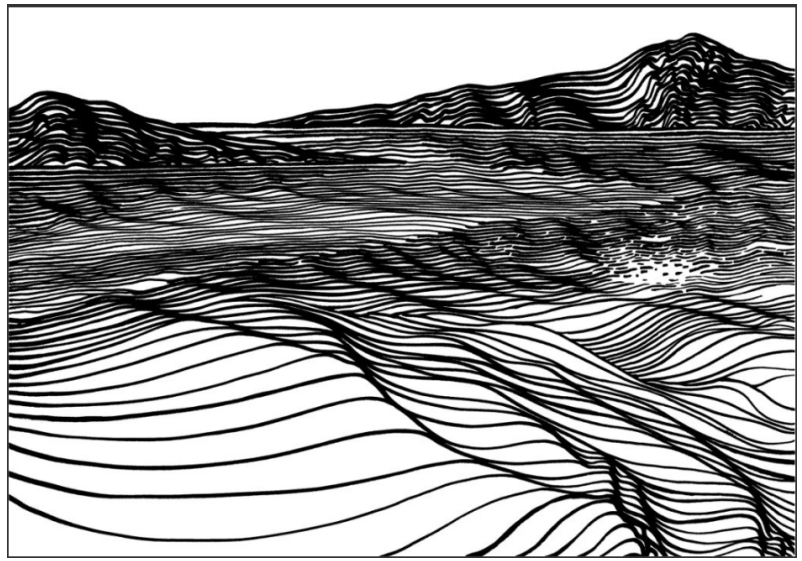


MCMC

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

posterior

Goal: sample the posterior distribution



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

WHILE convergence criterion is met:

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

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

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

Questions:

0. how do I choose a starting point?

we arent even going to talk about it...

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

WHILE convergence criterion is met:

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

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

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

Questions:

1. how do I choose the next point?

Any *Markovian* process

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

WHILE convergence criterion is met:

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

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

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

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.

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

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

Any *ergodic* process

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

WHILE convergence criterion is met:

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

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

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

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

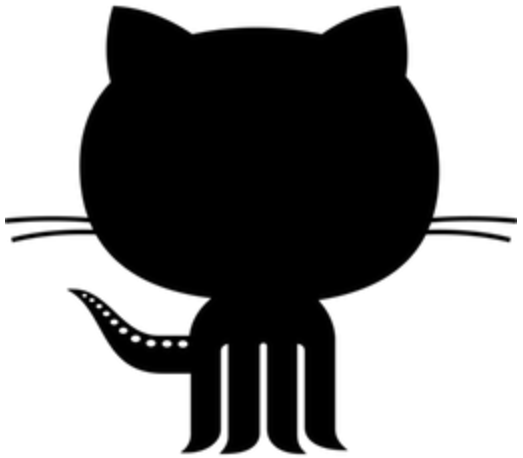
$$\pi(x_1)P(x_2|x_1) = \pi(x_2)P(x_1|x_2)$$

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** = $\theta_0 = (m,b)$

WHILE convergence criterion is met:

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

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

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 questions

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

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

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

posterior

Goal: sample the posterior distribution

Examples of how to choose the next point

Metropolis-Hasting

Gaussian random walk proposal distribution

choose a random starting point **current = *θo = (m,b)***
WHILE convergence criterion is met:
    calculate the current posterior **post_curr = *P(D| θo,f)***
    /*proposal*/
    draw a new set of parameters **new = *θ_{new} = (m,b)***
    calculate the current posterior **post_new = *P(D| θ_{new},f)***
    IF **post_new > post_curr:**
        current = new
    ELSE
        /*accept with probability ***P(D| θnew,f)/P(D| θo,f)*** */
        r = random uniform number [0,1]
        IF ***r < post_new / post_curr:***
            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 conditional distribution of each variable

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

WHILE convergence criterion is met:

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

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

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

[Submitted on 13 May 2009 (v1), last revised 30 Nov 2009 (this version, v2)]

Sampling from the thermal quantum Gibbs state and evaluating partition functions with a quantum computer

David Poulin, Pawel Wocjan

We present a quantum algorithm to prepare the thermal Gibbs state of interacting quantum systems. This algorithm sets a universal upper bound D^alpha on the thermalization time of a quantum system, where D is the system's Hilbert space dimension and alpha < 1/2 is proportional to the Helmholtz free energy density of the system. We also derive an algorithm to evaluate the partition function of a quantum system in a time proportional to the system's thermalization time and inversely proportional to the targeted accuracy squared.

Subjects: **Quantum Physics (quant-ph)**
Journal reference: Phys. Rev. Lett. 103 220502 (2009)

Journal of the Royal Statistical Society: Series B (Methodological) /
Volume 55, Issue 1 / p. 39-52

Article | Free Access

Modelling Complexity: Applications of Gibbs Sampling in Medicine

W. R. Gilks, D. G. Clayton, D. J. Spiegelhalter, N. G. Best, A. J. McNeil, L. D. Sharples, A. J. Kirby,

First published: September 1993
https://doi.org/10.1111/j.2517-6161.1993.tb01468.x
Citations: 17

Address for correspondence: Medical Research Council Biostatistics Unit, Institute of Public Health, University Forvie Site, Robinson Way, Cambridge, CB2 2SR, UK.

About

SUMMARY

We review applications of Gibbs sampling in medicine, involving longitudinal, spatial, covariate measurement and survival models. Applications in immunology, pharmacology, transplantation, cancer screening, industrial epidemiology and genetic epidemiology are discussed.

THE ASTROPHYSICAL JOURNAL
SUPPLEMENT SERIES

Power Spectrum Estimation from High-Resolution Maps by Gibbs Sampling

H. K. Eriksen[1], I. J. O'Dwyer[2], J. B. Jewell[3], B. D. Wandelt[4], D. L. Larson[5], K. M. Górski[6], S. Levin[7], A. J. Banday[8], and P. B. Lilje[9]

The Astrophysical Journal Supplement Series, Volume 155, Number 2

Citation H. K. Eriksen *et al* 2004 *ApJS* **155** 227

Article PDF View article

References

+ Article information

Abstract

We revisit a recently introduced power spectrum estimation technique based on Gibbs sampling, with the goal of applying it to the high-resolution *WMAP* data. In order to facilitate this analysis, a number of sophistications have to be introduced, each of which is discussed in detail. We have implemented two independent versions of the algorithm to cross-check the computer codes and to verify that a particular solution to any given problem does not affect the scientific results. We then apply these programs to simulated data with known properties at intermediate (N_{side} = 128) and high (N_{side} = 512) resolutions, to study effects such as incomplete sky coverage and white versus correlated noise. From these simulations we also establish the Markov chain correlation length as a function of signal-to-noise ratio and give a few comments on the properties of the correlation matrices involved. Parallelization issues are also discussed, with emphasis on real-world limitations imposed by current supercomputer facilities. The scientific results from the analysis of the first-year *WMAP* data are presented in a companion letter.

Multiple source localization using a maximum *a posteriori* Gibbs sampling approach

The Journal of the Acoustical Society of America **120**, 2627 (2006); https://doi.org/10.1121/1.2354027

Zoi-Heleni Michalopoulou[a)]

View Affiliations

PDF ABSTRACT FULL TEXT FIGURES CITED BY TOOLS

TOPICS

- Acoustical oceanography
- Image processing
- Probability theory
- Acoustic modeling, simulation and analysis
- Acoustic signal processing
- Monte Carlo methods
- Speed of sound
- Acoustic wave propagation
- Hydrophone
- Acoustic source localization

ABSTRACT

Multiple source localization in underwater environments is approached within a matched-field processing framework. A maximum *a posteriori* estimation method is proposed that estimates source location and spectral characteristics of multiple sources via Gibbs sampling. The method facilitates localization of weak sources which are typically masked by the presence of strong interferers. A performance evaluation study based on Monte Carlo simulations shows that the proposed maximum *a posteriori* estimation approach is superior to simple coherent matched-field interference cancellation. The proposed method is also tested on the estimation of the number of sources present, providing probability distributions in addition to point estimates for the number of sources.

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

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

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

WHILE convergence criterion is met:

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

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

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

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

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

Other options:

Hamiltonian MC: proposing moves to distant states which maintain a high probability of acceptance due to the approximate energy conserving properties of the simulated Hamiltonian

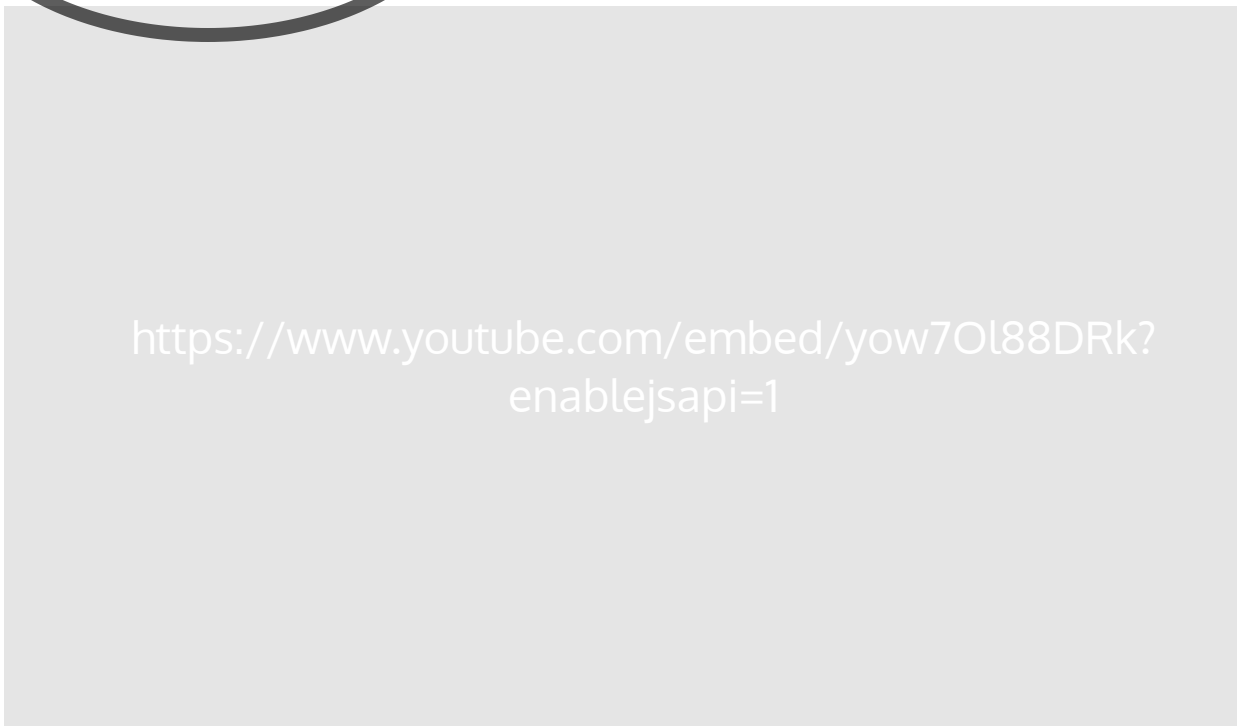
simulated annealing (good for multimodal)
parallel tempering (good for multimodal)
differential evoution (good for covariant spaces)

MCMC

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

Examples of how to choose the next point

affine invariant : EMCEE package



1

1

MCMC convergence

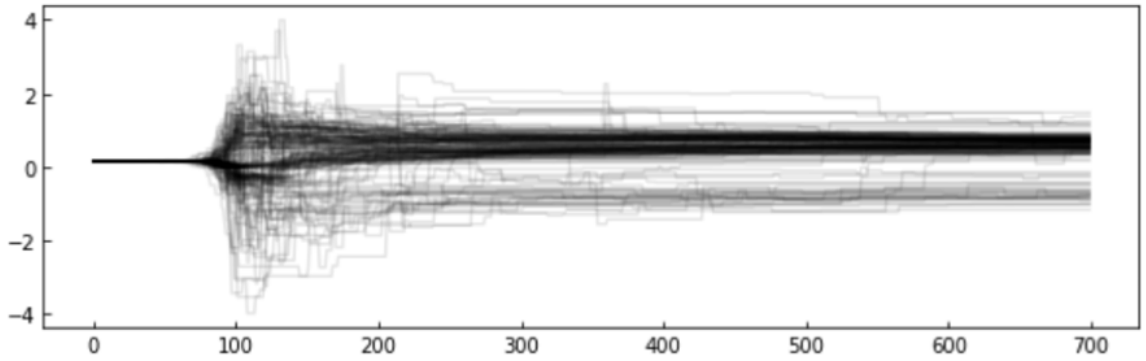
Goal: sample the posterior distribution

Questions:

1. how do I choose the next point?

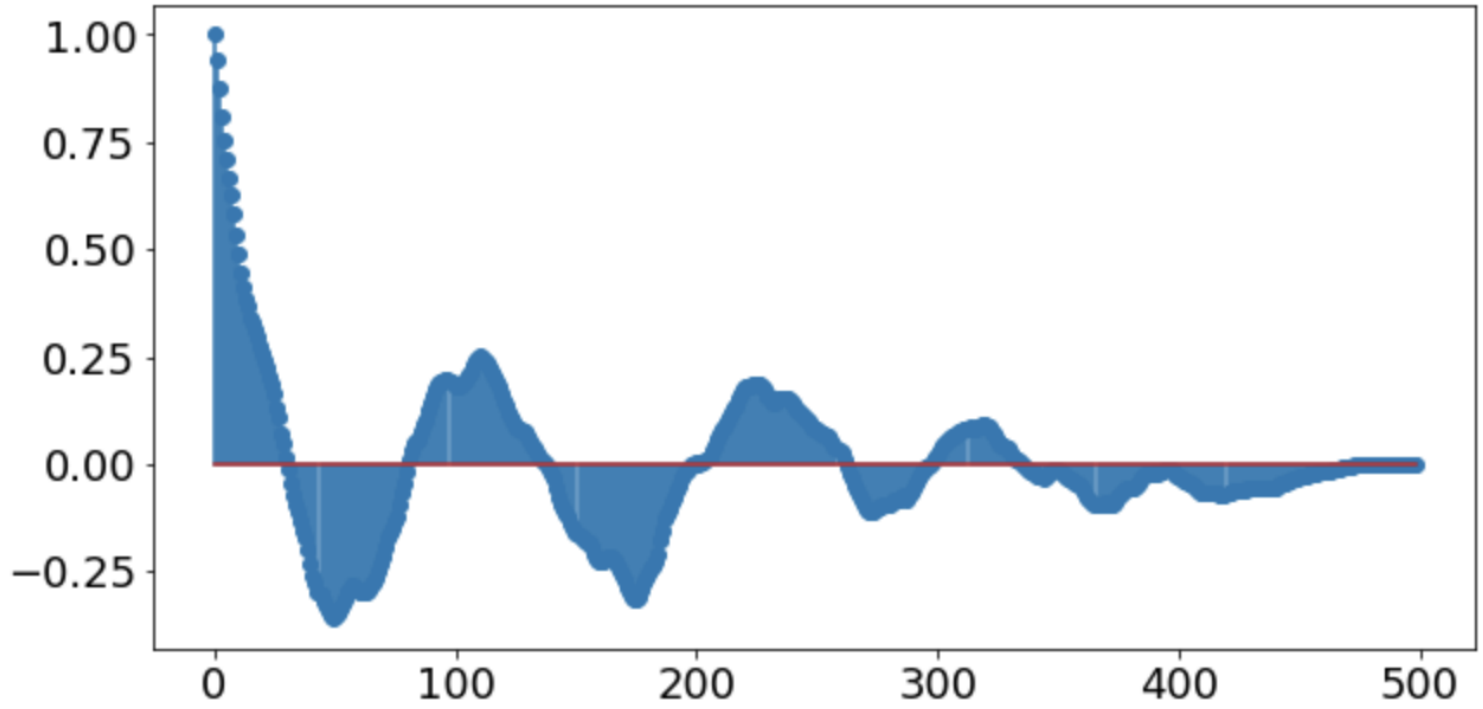
2. when have I sampled the posterior adequatly?
   has your MCMC *converged*?

MCMC convergence




```
acorr(sampler.chain[0,:,0])
pl.show()
print ("Fig 11: Chain Autocorrelation")
```

```
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:10: UserWarning: In Matplotlib 3.3 individual
  # Remove the CWD from sys.path while we load stuff.
```




```
Fig 11: Chain Autocorrelation
```

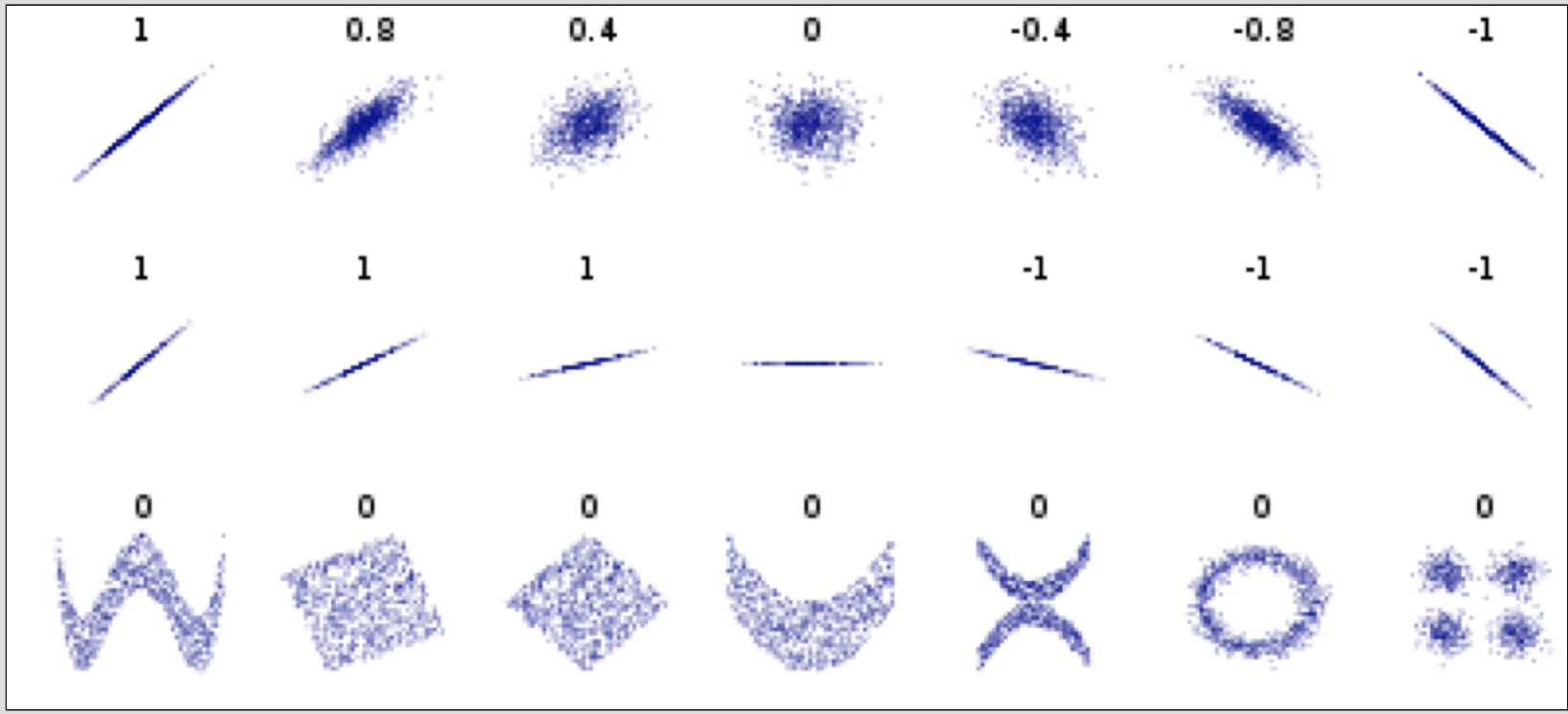
$$r_{xy} = \frac{1}{n-1} \sum_{i=1}^{N} \left(\frac{x_i - \bar{x}}{s_x} \right) \left(\frac{y_i - \bar{y}}{s_y} \right)$$

$$s_x = \sqrt{\frac{1}{n-1} \sum_{i=1}^{N} (x_i - \bar{x})^2}$$

Pearson's test: tests *linear* correlation

```
1 import scipy as sp
2 print("Pearson's correlation {:.2}, approximate p-value {:.2}".format(
3     *sp.stats.pearsonr(x, y)))
```

Pearson's correlation 0.94, approximate p-value 2.9e-49



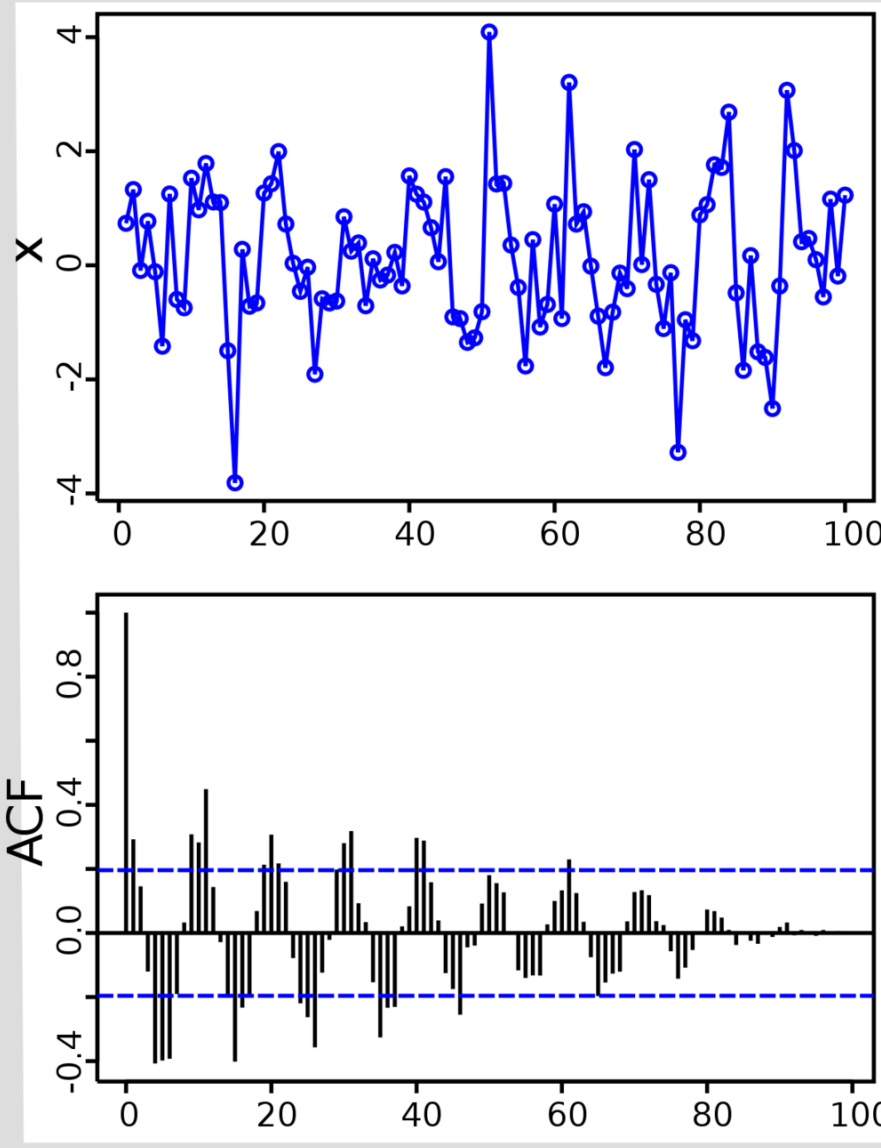
correlation

$$r_{xy} = \frac{1}{n-1} \sum_{i=1}^{N} \left(\frac{x_i - \bar{x}}{s_x} \right) \left(\frac{y_i - \bar{y}}{s_y} \right)$$

$$s_x = \sqrt{\frac{1}{n-1} \sum_{i=1}^{N} (x_i - \bar{x})^2}$$

Pearson's test

autocorrelation

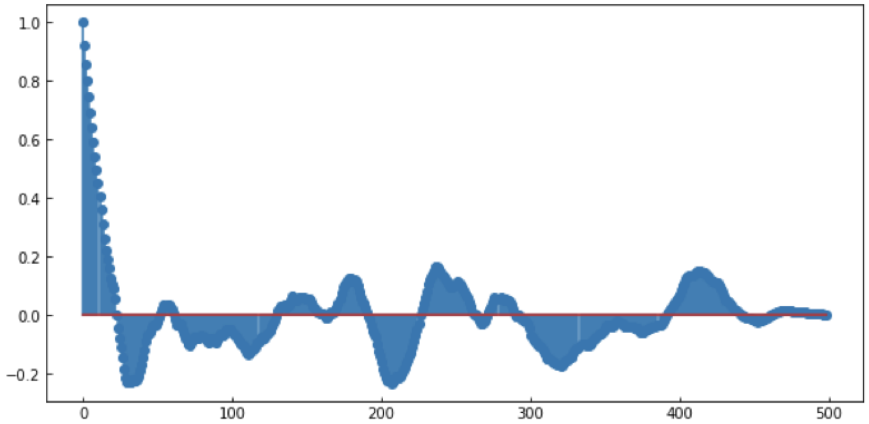


$$ACF(lag) = \frac{1}{n-1} \sum_{i=1}^{N} \left(\frac{x_i - \bar{x}}{s_x} \right) \left(\frac{x_j - \bar{x}}{s_x} \right)$$

$$i - j = \text{lag}$$

correlation

MCMC convergence



Goal: sample the posterior distribution

Questions:

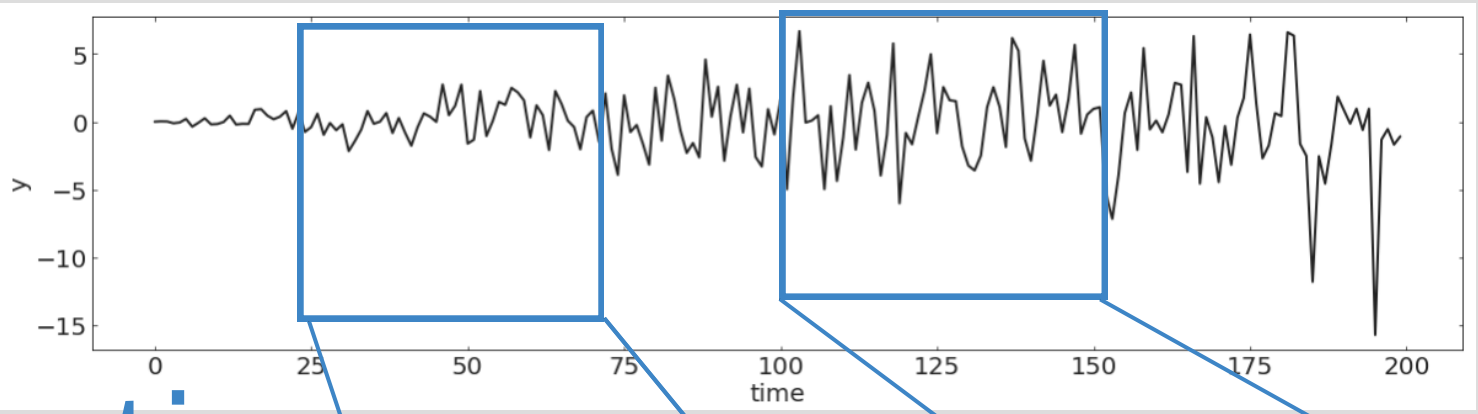
1. how do I choose the next point?
2. when have I sampled the posterior adequatly?
   has your MCMC *converged*?

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)

Stochastic process

A random variable indexed by time.

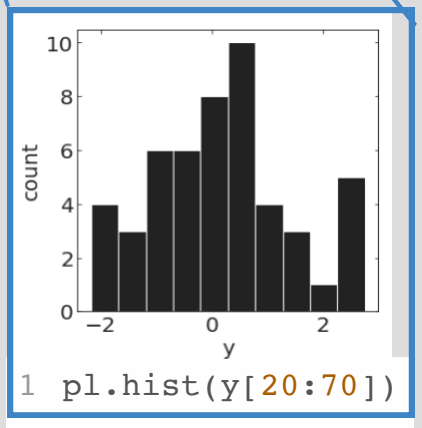
```
1 pl.figure(figsize=(20,5))
2 N = 200
3 np.random.seed(100)
4 y = np.random.randn(N)
5 t = np.linspace(0, N, N, endpoint=False)
6 pl.plot(t, y, lw=2)
7 pl.xlabel("time")
8 pl.ylabel("y");
```

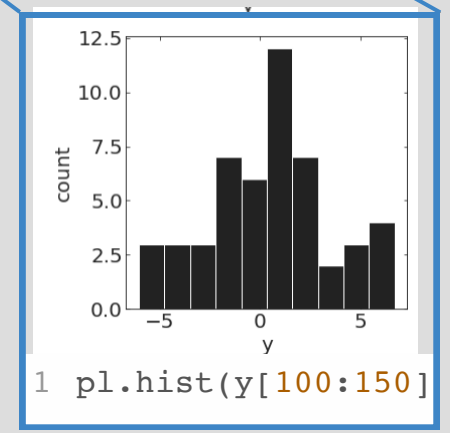


Discrete stochastic process

For any subset of points in time the dependent variable follows the a probability distribution

e.g. $p(x_{t1} \ldots x_{tn}) \sim N(\mu, \sigma)$



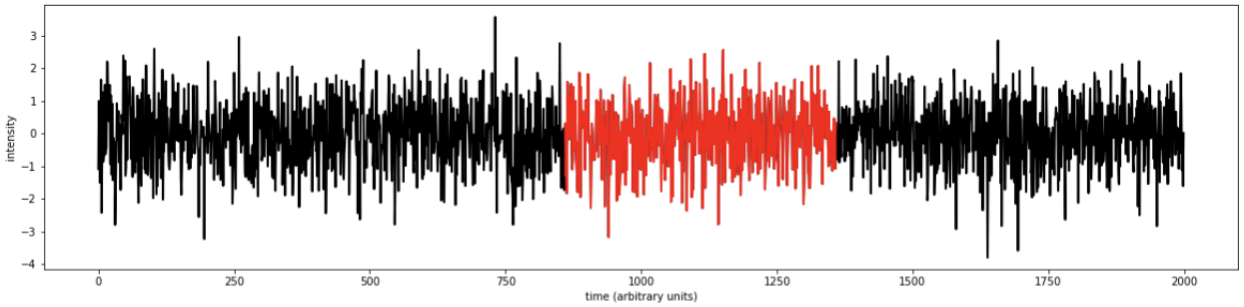


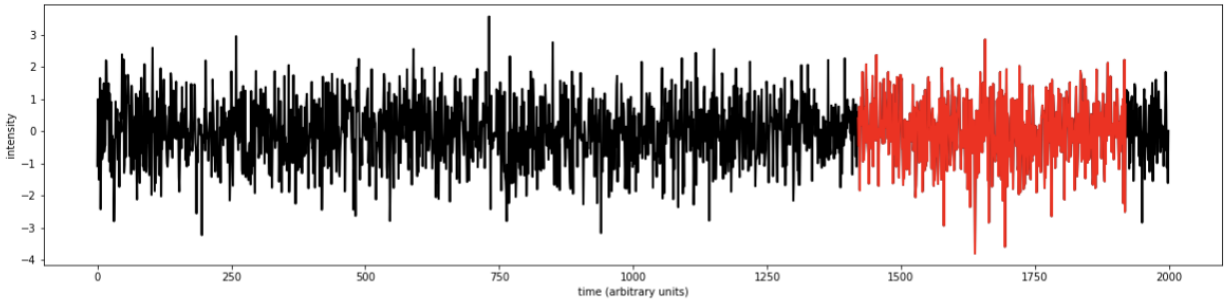
strictly stationary process

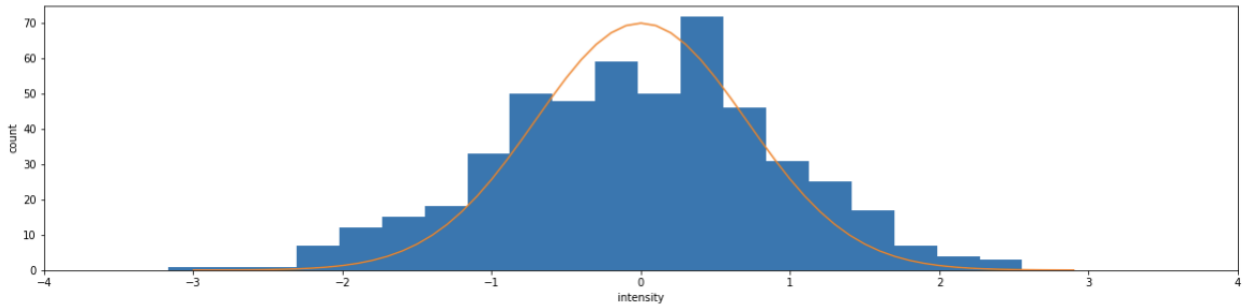
A time series is ***strictly stationary*** if for any *i* and *Δt*

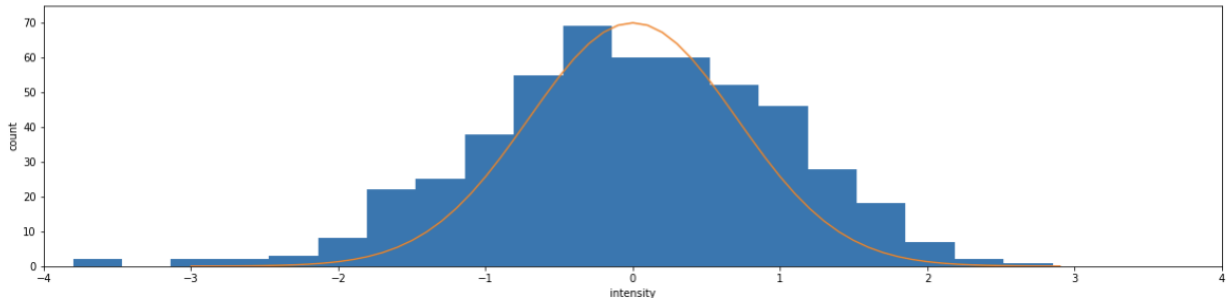
$$p(x_i \ldots . x_{n+i}) \sim p(x_i + \Delta x \ldots . x_{n+i} + \Delta x)$$

https://github.com/fedhere/MLTSA_FBianco/blob/master/CodeExamples/StationaryTSAnimation.ipynb

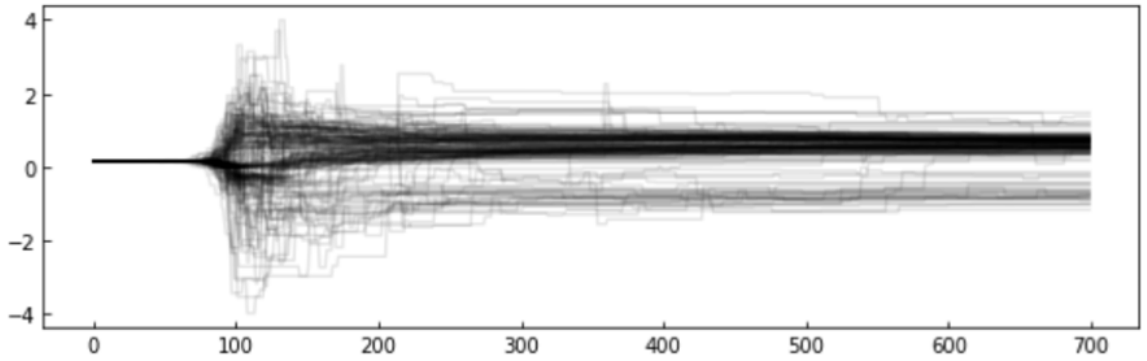








MCMC convergence



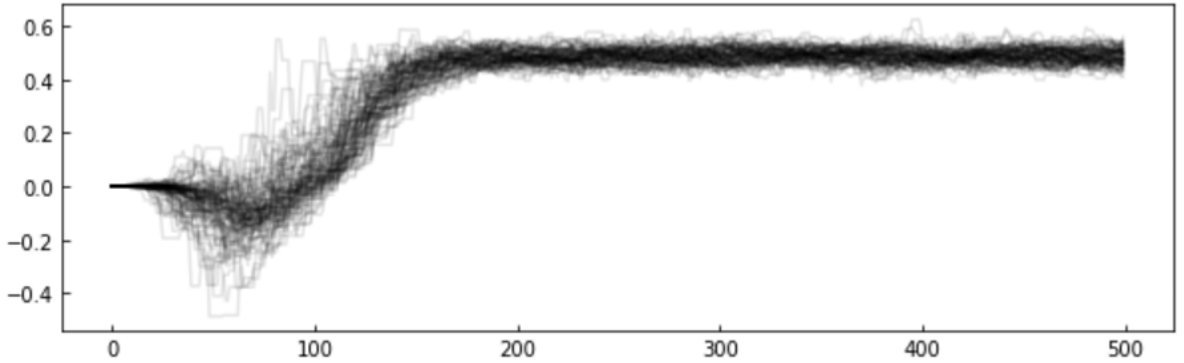
Goal: sample the posterior distribution

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



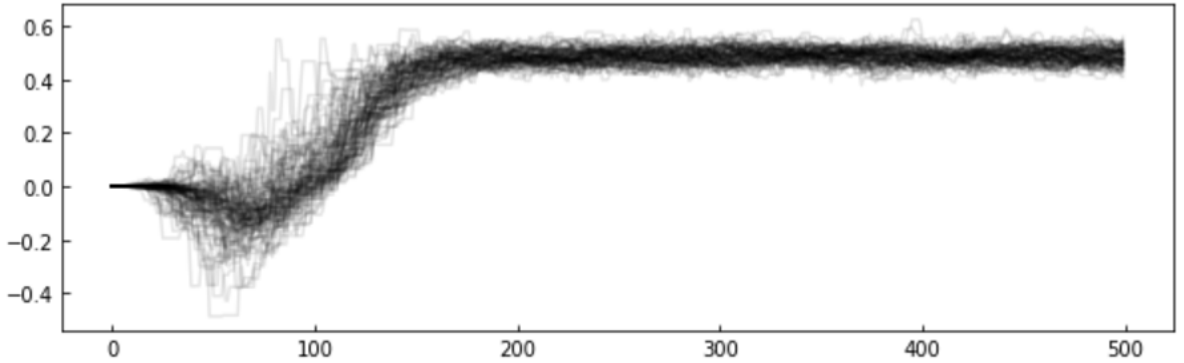
Goal: sample the posterior distribution

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).

Posterior(data, model-parameters) = Likelihood(data, model-parameters) * Prior(model-parameters)

Evidence(data)

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

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 ***α***

The choice of the proposal distribution and rule ***α*** 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

While My MCMC Gently Samples

Bayesian modeling, Computational Psychiatry, and Python

A blog by

https://twiecki.io

VP of data science at Quantopian

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 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!