

DATA221 Intro Machine Learning

02 objective function

William Trimble
Winter 2023



THE UNIVERSITY OF
CHICAGO

That smiling LinkedIn profile face might be a computer-generated fake

March 27, 2022 · 7:00 AM ET



SHANNON BOND



Some of the likely AI-generated faces from fake LinkedIn profiles identified by Stanford University researchers. The central positioning of the eyes is a telltale sign of a computer-created face. Click on the animation to pause.

Source: Stanford Internet Observatory

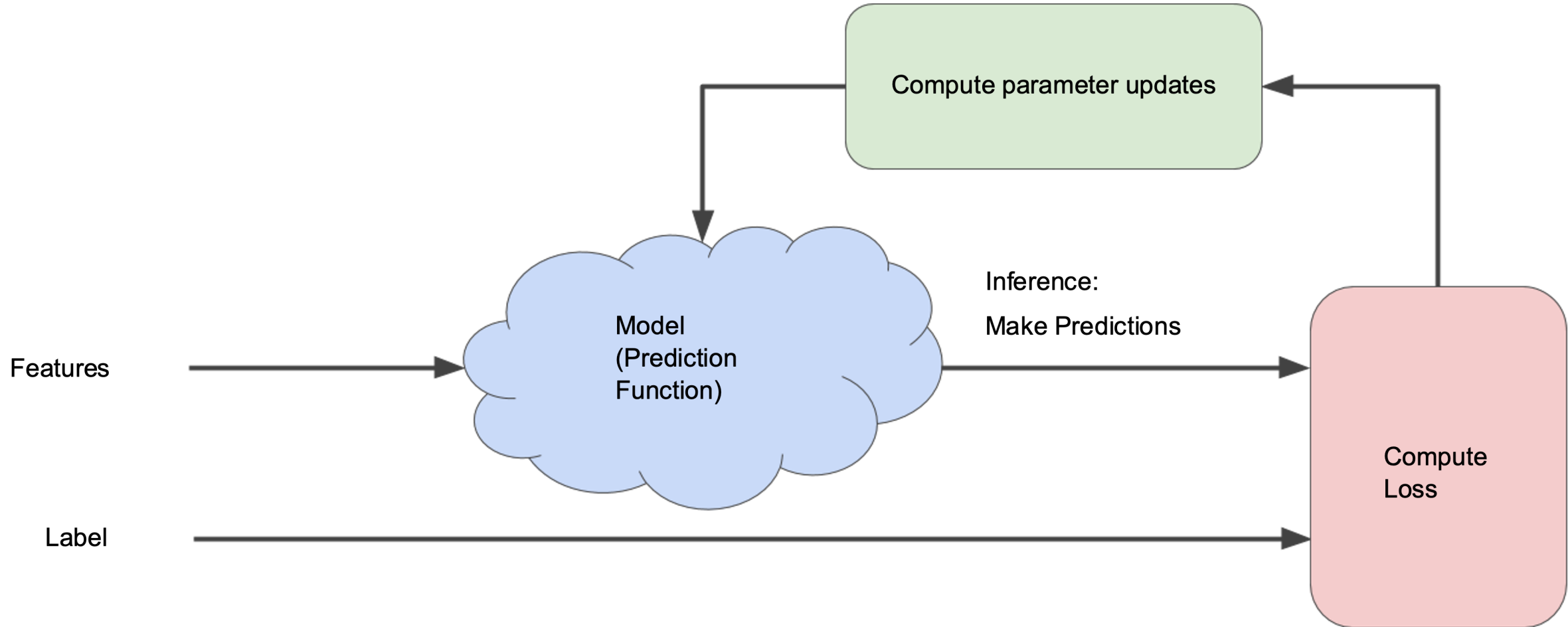
Credit: Connie Hanzhang Jin/NPR

Caught in the wild!

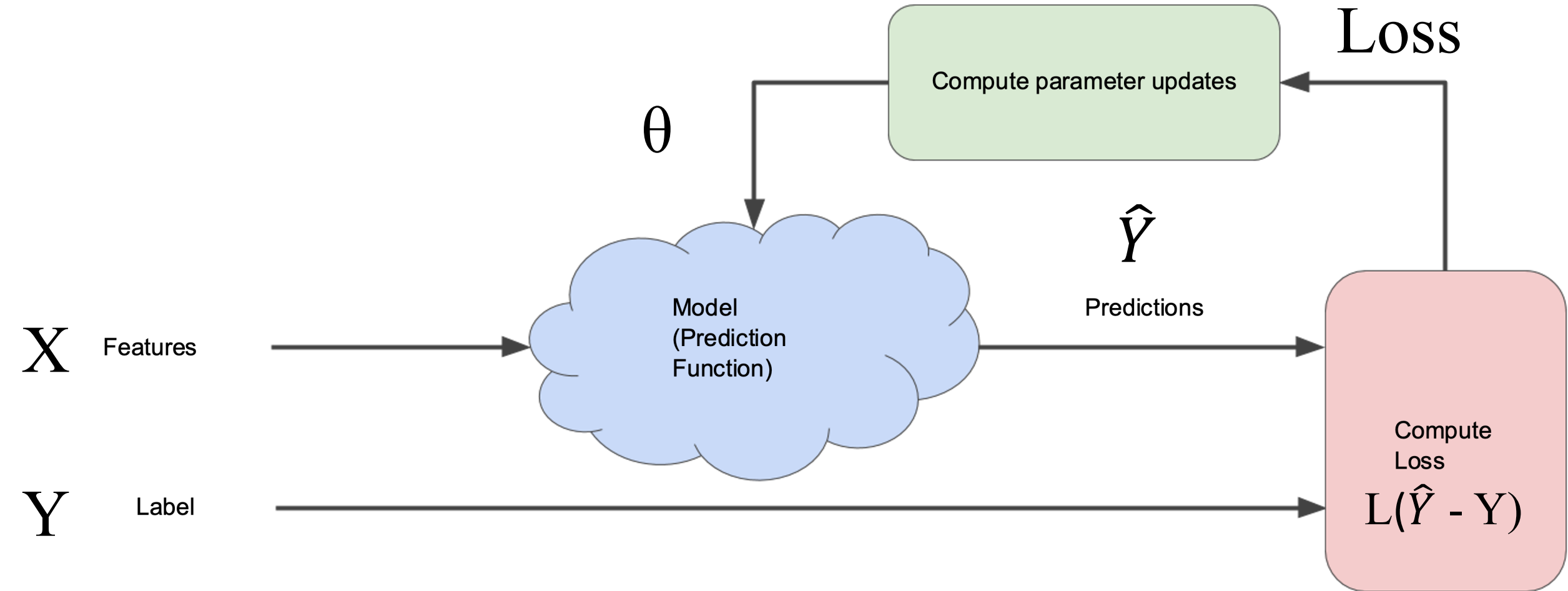
And for the noble,
purpose allowing
salespeople to send
spam with fewer
constraints!

<https://www.npr.org/2022/03/27/1088140809/fake-linkedin-profiles>

Fairy dust picture of optimization



Fairy dust picture of optimization



Argmin symmetries

$$\hat{\theta} = \operatorname{argmin}_{\theta} L(\theta, X, Y)$$

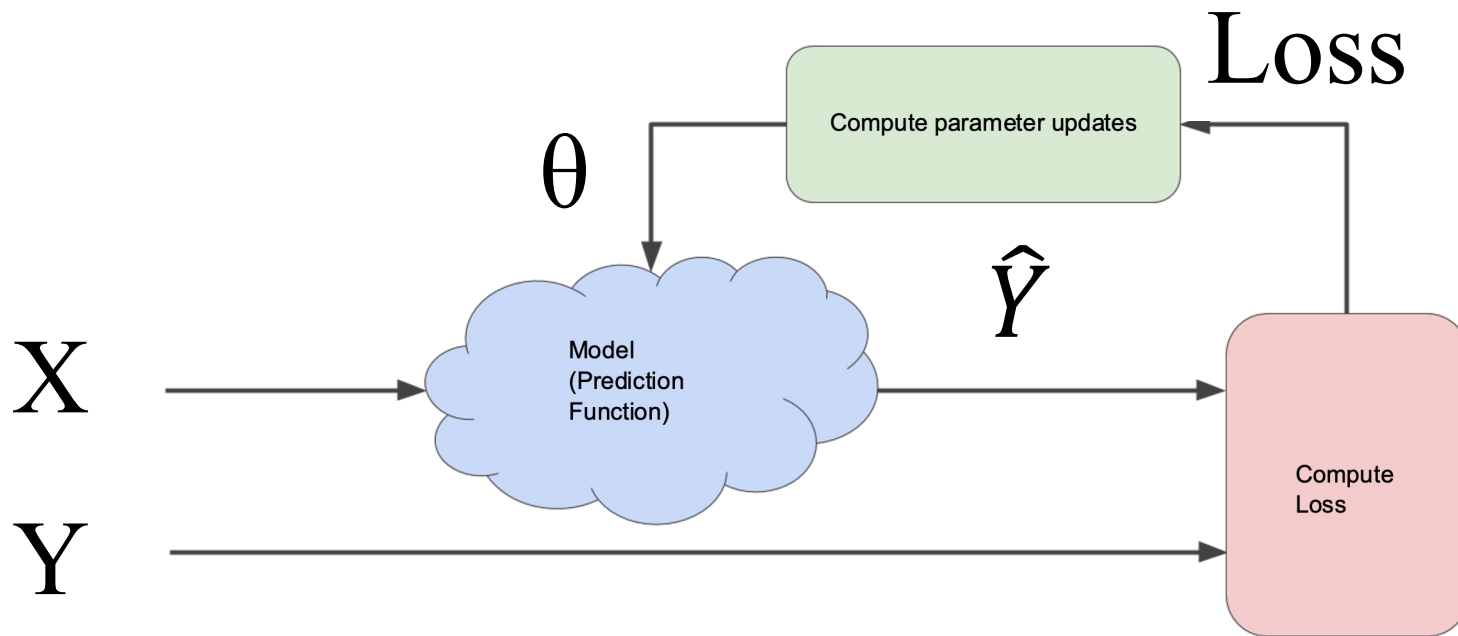
This metafunction has some symmetries:

$L+c$ has same argmin as L

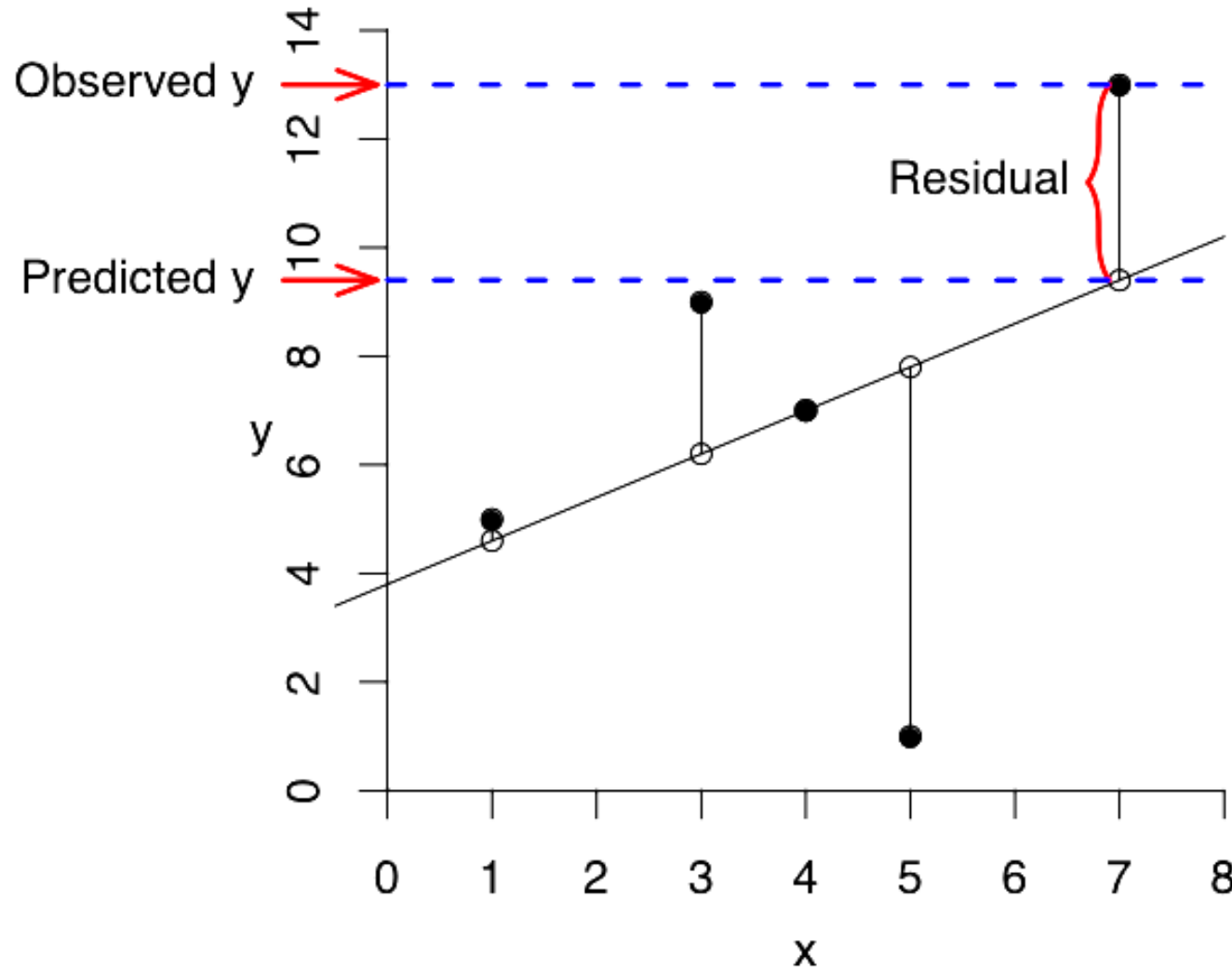
cL has same argmin as L

L^2 , $\operatorname{abs}(L)$, $L^{1/2}$ same argmin

$\log(L)$ has the same argmin if L isn't outside its domain



“Regression” = prediction of values



Summed squared error

$$SSE = \sum (\hat{y}_i - y_i)^2$$

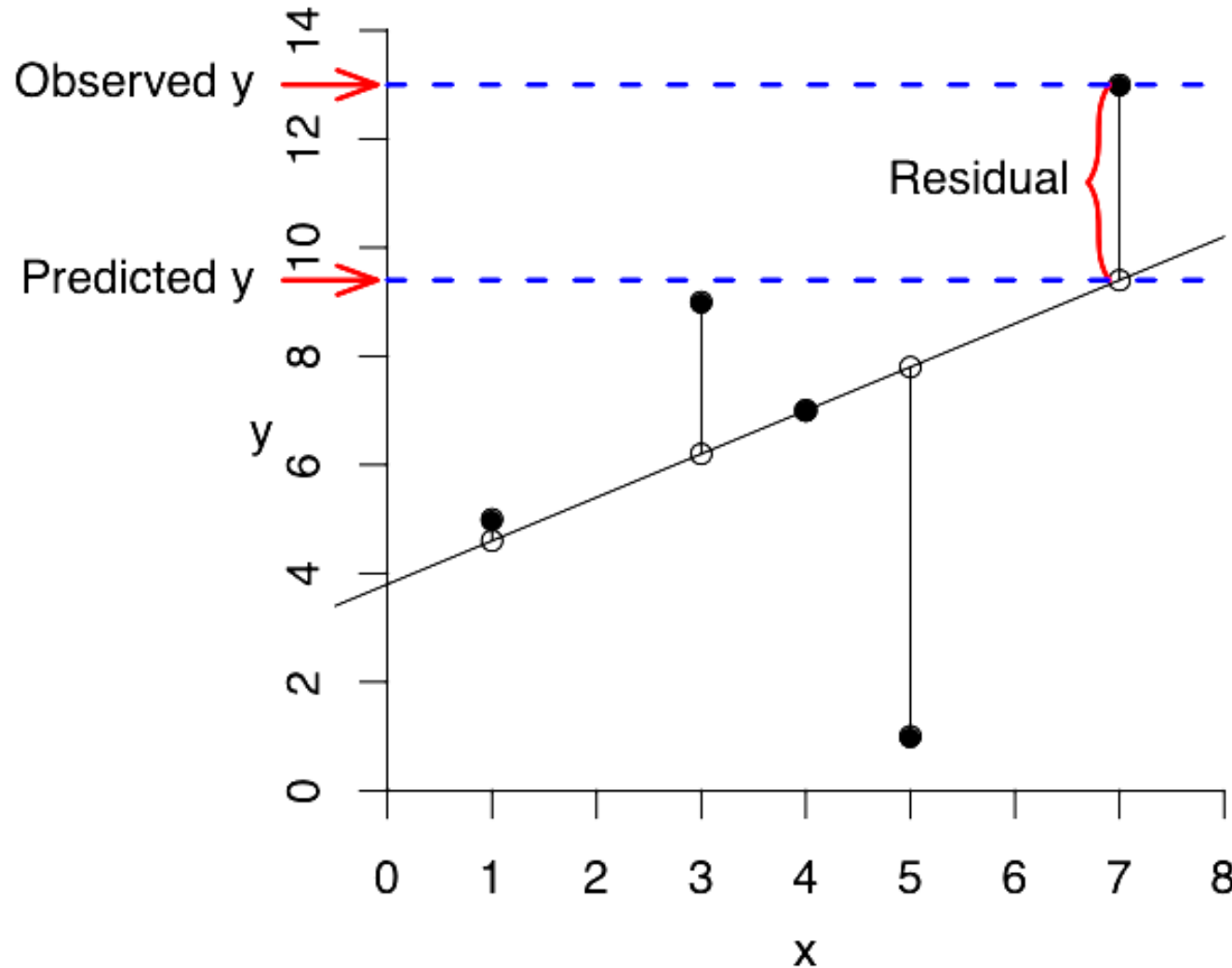
Root mean square

$$RMS = \sqrt{\frac{\sum (\hat{y}_i - y_i)^2}{n}}$$

Summed absolute error

$$SAE = \sum |\hat{y}_i - y_i|$$

“Regression” = prediction of values



Summed squared error
$$SSE = \sum (\hat{y}_i - y_i)^2$$

Summed absolute error
$$SAE = \sum |\hat{y}_i - y_i|$$

I don't have to calculate these because of the argmin symmetries.

These are monotonic functions of SSE and SAE

Mean squared error

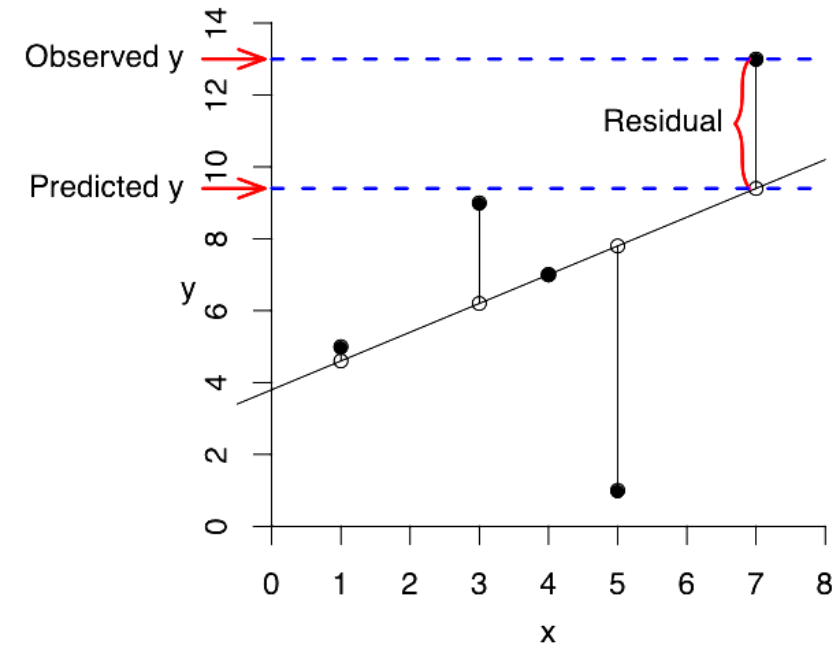
$$\text{MSE} = \frac{\sum (\hat{y}_i - y_i)^2}{n}$$

Root mean square

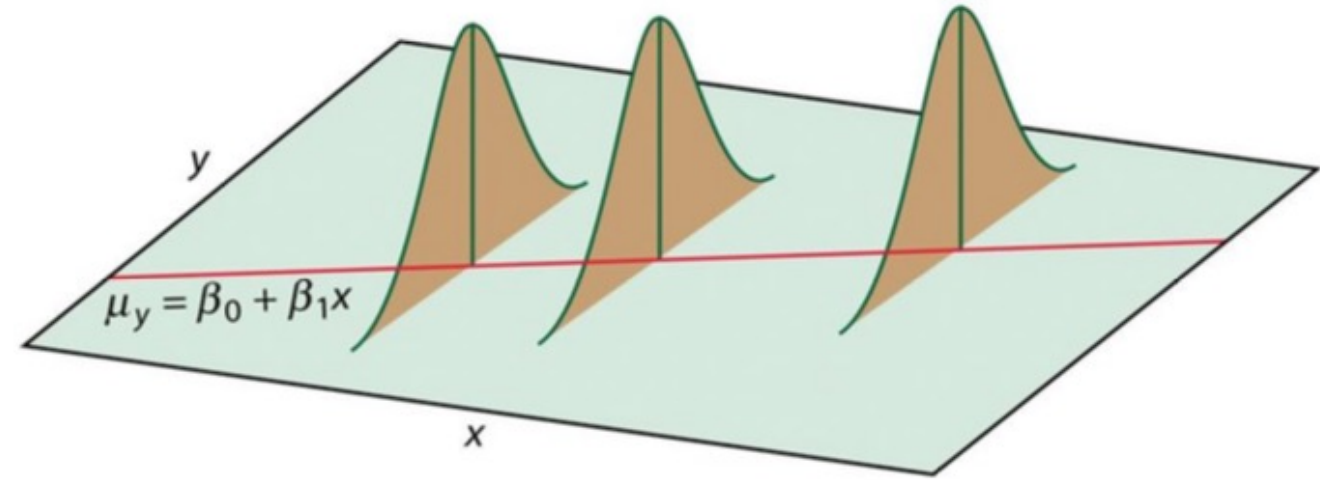
$$\text{RMS} = \sqrt{\frac{\sum (\hat{y}_i - y_i)^2}{n}}$$

Mean absolute error

$$\text{MAE} = \frac{\sum |\hat{y}_i - y_i|}{n}$$



Consequences of sum-squared-error



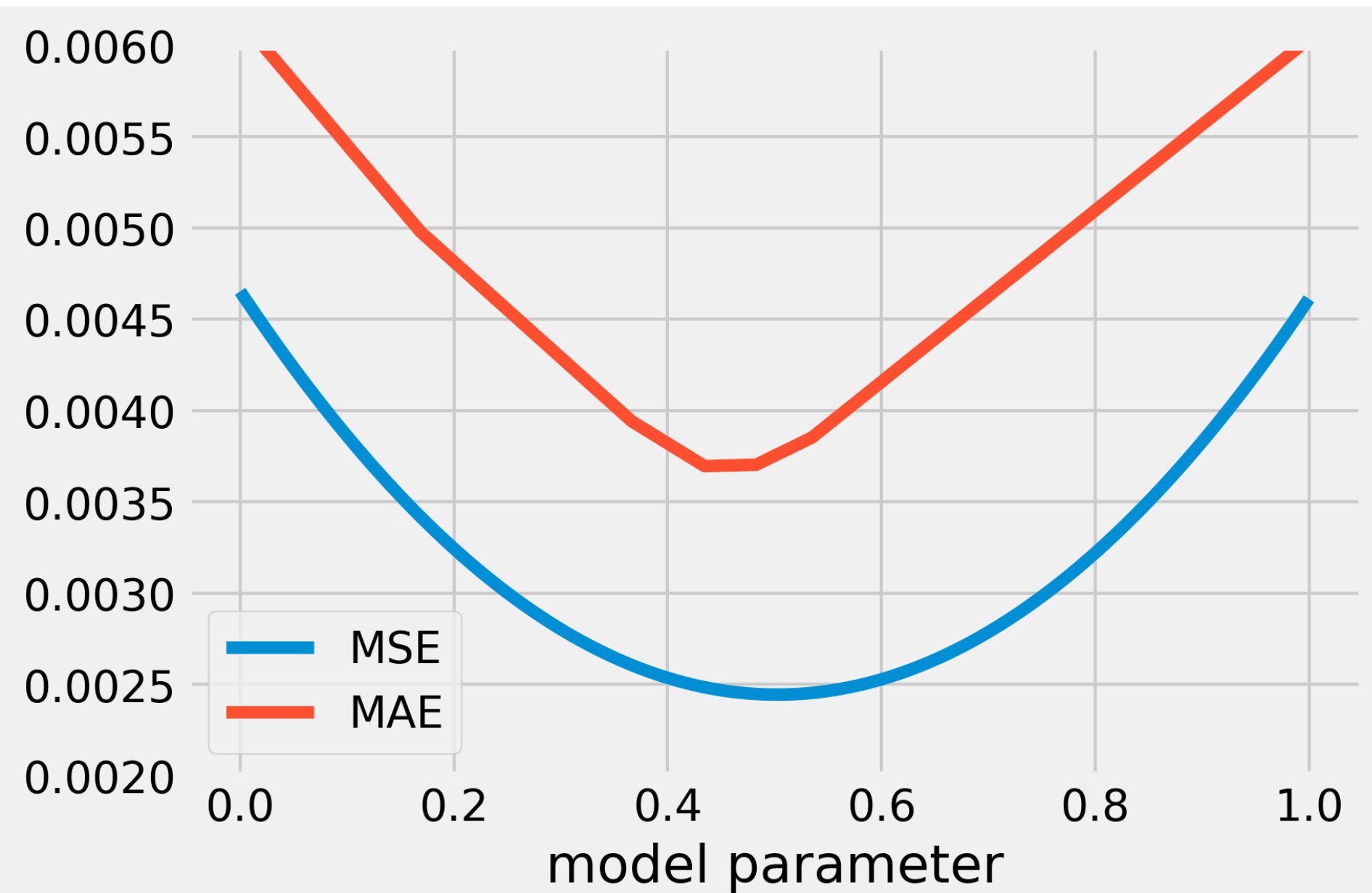
- Minimizing sum-squared error solves the problem of model + additive normally-distributed noise in y where **the noise level at each point is the same.**
- This is often not reasonable; each point often should not get the same weight. **Examples?**
- Weighted (per-datapoint) sums of sum-squared error relax this requirement if you have a theoretical (or empirical) reason to estimate them differently. (Standard error of the mean, anyone?)

Loss function sometimes implies probability

- When the objective function looks like (is proportional to) a log-probability function, optimizing it looks a lot like maximum likelihood / maximum a posteriori estimation.
- Sum squared errors.. are the log of additive normal error distributions
- Certain regularization terms , like $w^T w$, solve the problem with as if the total probability distribution includes normal priors on the components of w . (!!!)
- Nice if it's bounded below
- Discontinuities are frowned upon.

Objective function options : choices

- The choice of function here sets the balance between small errors and large errors.
- Mean Squared Error / sum squared error “L2 loss function”
 - penalizes large errors much more than small ones
- Mean absolute error “L1 loss function”
 - discontinuous derivative; derivative does not vanish anywhere; all the points pull on the fit whether above or below the fit.
- Constraints on parameters, parameter domain...
- Cross entropy or expected $-\log(p)$
 - Useful for completely blind density fitting



Discontinuities in the slope of mean absolute error are hard on optimization.

Contrast L2 and L1

- L2 in the residuals gives the same solution as optimizing additive normally distributed errors.
- L1 in the residuals give the same solution as optimizing Laplace-distributed errors.
- But we can add terms to the loss function that don't correspond to probability distributions.. these will steer the solution around..

Loss functions for categorical data:

- “Accuracy” -- number of correct assignments on the test set
- Makes sense to assign penalties to each wrong answer. They can be all the same or they can be different

CIFAR-10 Confusion Matrix

True Class	airplane	923	4	21	8	4	1	5	5	23	6	92.3%	7.7%
	automobile	5	972	2					1	5	15	97.2%	2.8%
	bird	26	2	892	30	13	8	17	5	4	3	89.2%	10.8%
	cat	12	4	32	826	24	48	30	12	5	7	82.6%	17.4%
	deer	5	1	28	24	898	13	14	14	2	1	89.8%	10.2%
	dog	7	2	28	111	18	801	13	17		3	80.1%	19.9%
	frog	5		16	27	3	4	943	1	1		94.3%	5.7%
	horse	9	1	14	13	22	17	3	915	2	4	91.5%	8.5%
	ship	37	10	4	4		1	2	1	931	10	93.1%	6.9%
	truck	20	39	3	3			2	1	9	923	92.3%	7.7%

88.0%	93.9%	85.8%	79.0%	91.4%	89.7%	91.6%	94.1%	94.8%	95.0%
12.0%	6.1%	14.2%	21.0%	8.6%	10.3%	8.4%	5.9%	5.2%	5.0%

airplane automobile bird cat deer dog frog horse ship truck

Predicted Class

Expected loss

For unknown state of nature $P(\omega_j | \mathbf{x})$, and action α the risk associated with action α_i is the weighted sum of the loss function $\lambda(\alpha_i | \omega_j)$ over all the possible states of nature $P(\omega_j | \mathbf{x})$

- $R(\alpha_i | \mathbf{x}) = \sum_j \lambda(\alpha_i | \omega_j) P(\omega_j | \mathbf{x})$

This means I need to define a loss $\lambda(\alpha_i | \omega_j)$ for every element of the confusion matrix.

Thoughts on the loss

- Perhaps a different value for λ (gorilla | human) than for λ (dog | cat) for general-purpose images?
- At first glance, this looks like a place that we could fit our preferences for the balance between type-I and type-II errors.
- λ (no action | unexpected pedestrian walking in front of car)
- λ (take action so extreme it may cause injury | unexpected pedestrian walking in front of car)

Zero-one loss function

- The simplest loss function, called the zero-one loss function, is just zero for all of the correct decisions and one for all of the incorrect decisions.

$$\lambda_{ij} = 1 - \delta(i, j)$$

This counts the number of errors.

$$\lambda = \begin{matrix} & \begin{matrix} 0 & 1 & 1 \end{matrix} \\ \begin{matrix} 0 \\ 1 \\ 1 \end{matrix} & \begin{matrix} 1 & 0 & 1 \\ 1 & 1 & 0 \end{matrix} \end{matrix}$$

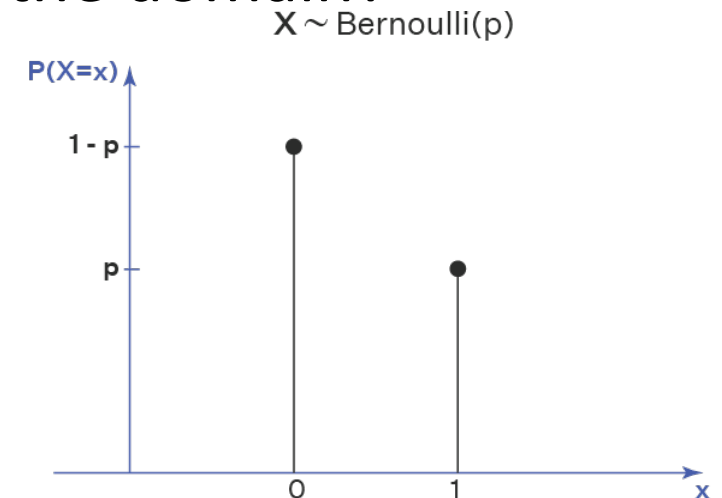
How about the domains?

- When estimating probabilities (like mixing ratios) some parameters naturally live in the parameter space of the Bernoulli (or multinomial / categorical distribution)
- That is to say, there are a lot of useful parameters out there that are between 0 and 1.
- How do we search a space without running out of the domain?

Loss ($\min(\max(0, x), 1)$) is not a good choice.

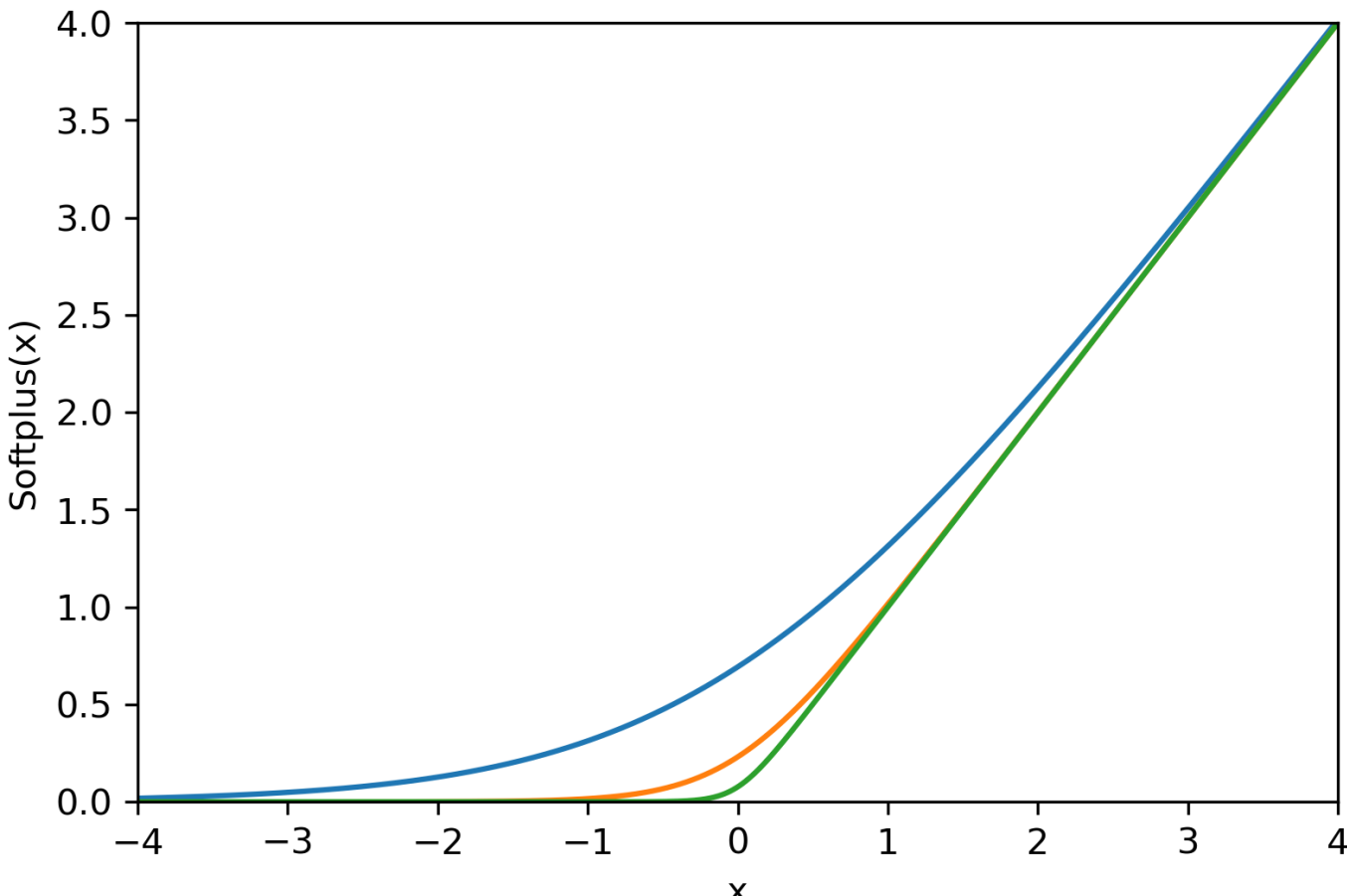
Loss_{CONSTRAINED} = Loss + HUGE ($x > 1$) + HUGE ($x < 0$)

also not a good choice



Softplus

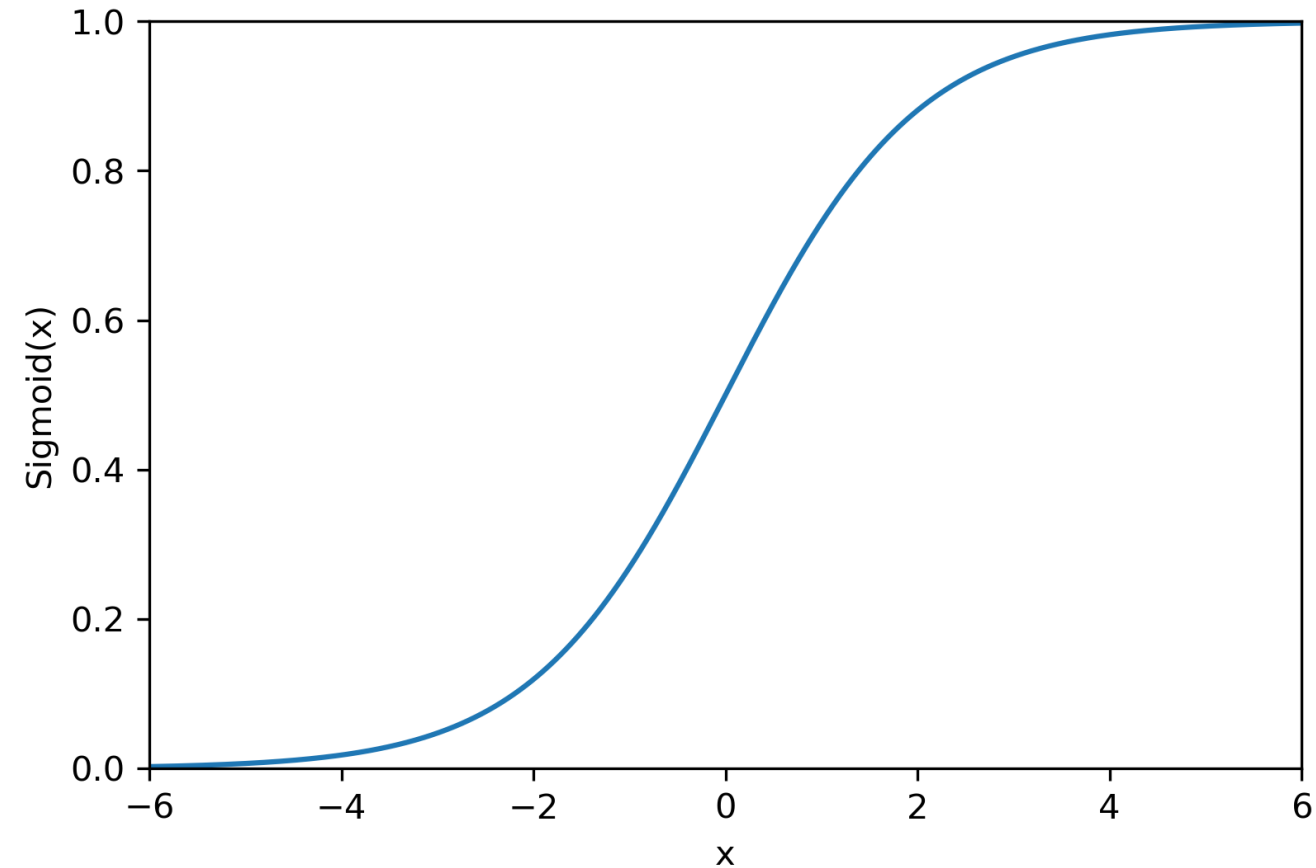
$$\text{softplus}(x) = \log(1 + \exp(x))$$



- gentler form of $\max(0, x)$

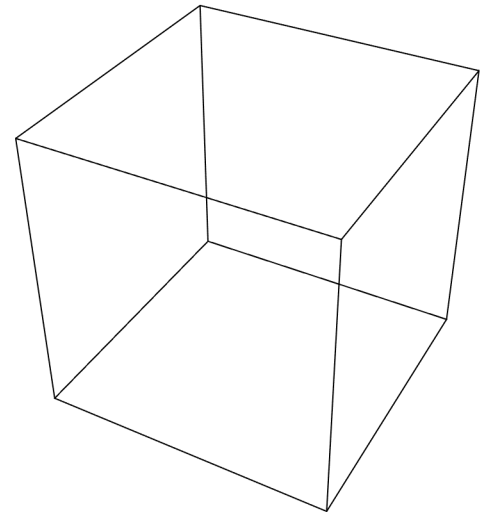
Sigmoid function

$$\textit{sigmoid}(x) = \frac{\exp(x)}{1 + \exp(x)}$$



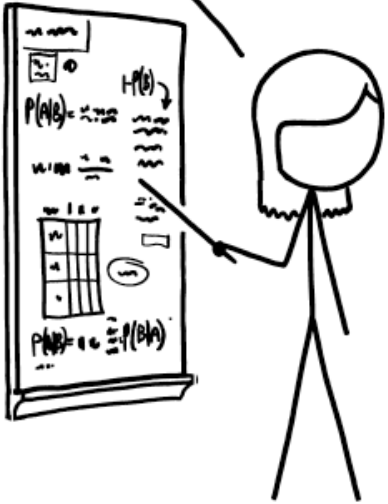
- continuous mapping from \mathbb{R} to $(0,1)$

\mathbb{R}^3



GIVEN THESE PREVALENCES,
IS IT LIKELY THAT THE TEST
RESULT IS A FALSE POSITIVE?

WELL, THIS CHAPTER IS ON
BAYES' THEOREM, SO YES.



SOMETIMES, IF YOU UNDERSTAND
BAYES' THEOREM WELL ENOUGH,
YOU DON'T NEED IT.

In favorable cases, we
can use Bayes' theorem
to estimate our
parameters.

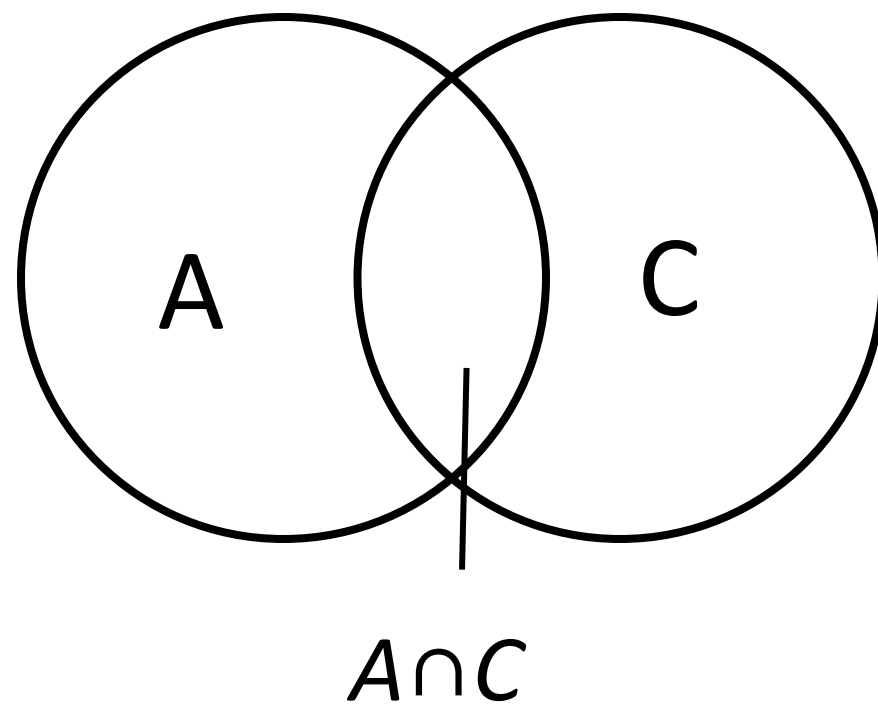
In less favorable cases,
we can do a randomized
search for possible
parameter values.

Bayesian inference

Event A: unknown state of nature

Event C: experiment

$$P(A \mid C) = P(C \mid A) P(A) / P(C)$$

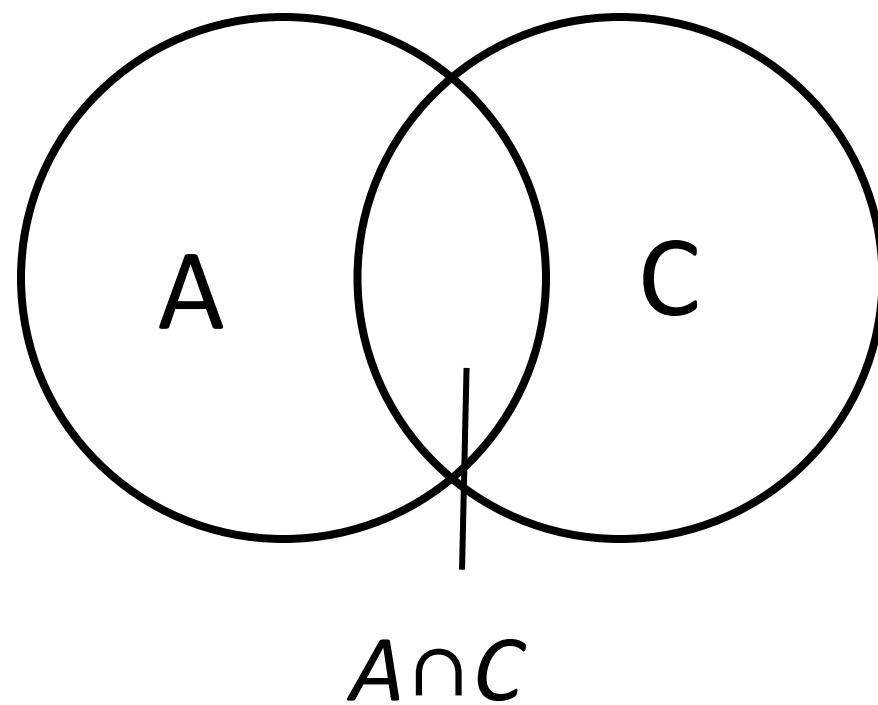


Bayesian inference

Event A: unknown state of nature

Event C: experiment

$$P(A \mid C) = P(C \mid A) P(A) / P(C)$$



“Numerical decisionmaking
for grownups”

Bayesian inference

Event A: am I infected?

Event C: experiment (test result)

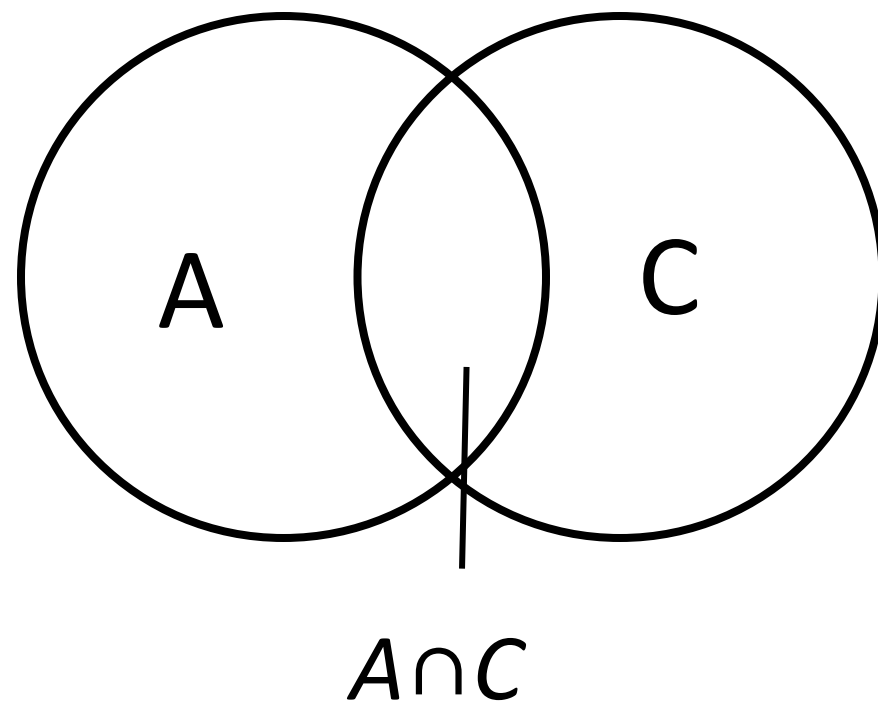
$$P(A \mid C) = P(C \mid A) P(A) / P(C)$$



The thing
I want to
know



The thing
the FDA
wants
know



Bayesian inference

Event A: unknown state of nature

Event C: experiment

$$P(A \mid C) = P(C \mid A) P(A) / P(C)$$



Life's
persistent
question



Term that
depends on
experiment

likelihood

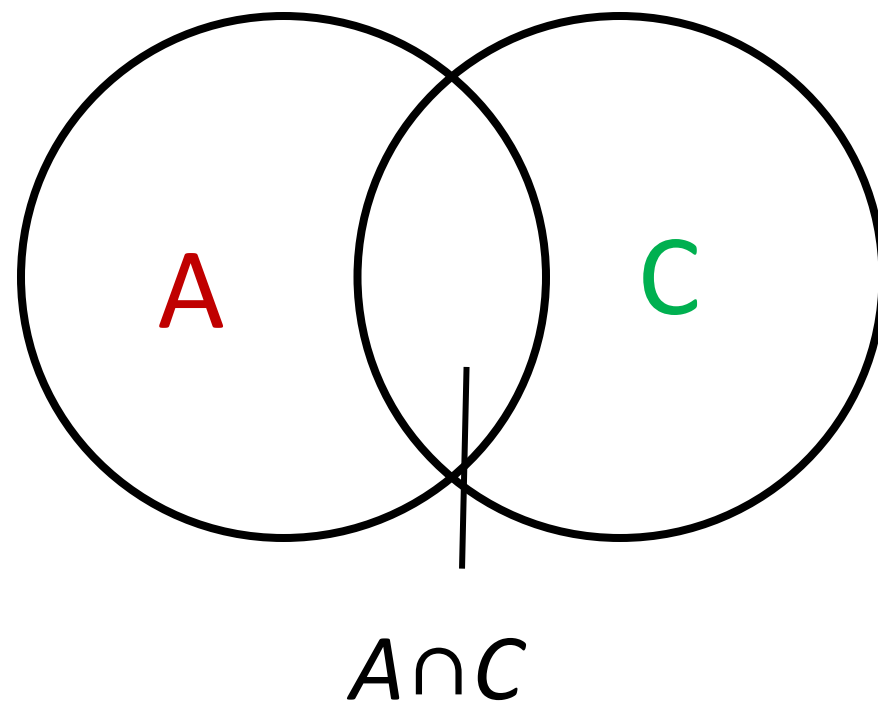


Term with
knowledge
and bias
about A

prior



Probability
for event
that is no
longer
uncertain
evidence



Bayesian inference

Event A: value of parameter

Event C: data

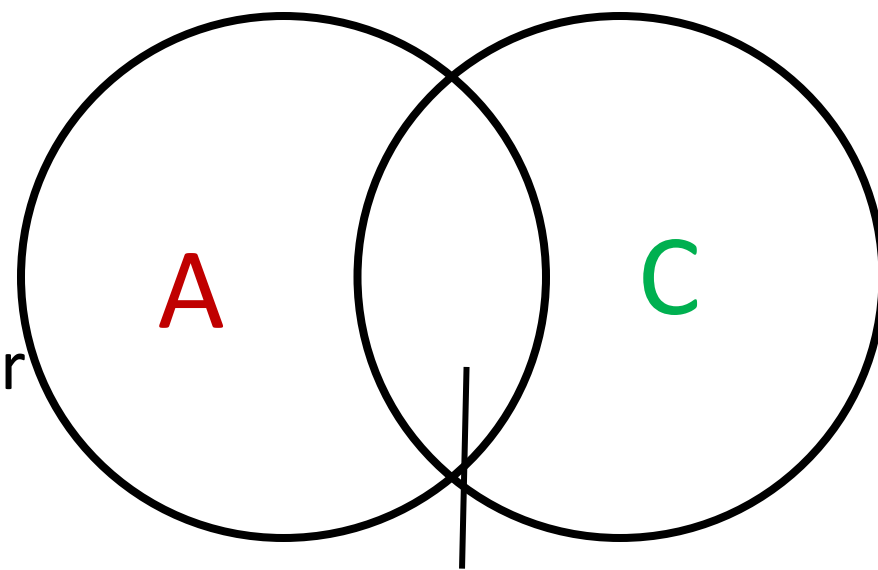
$$P(A \mid C) = P(C \mid A) P(A) / P(C)$$

Desired
distribution/
parameter
value

This is the

Prior
distribution
for parameter

Doesn't
matter for
us



posterior

likelihood

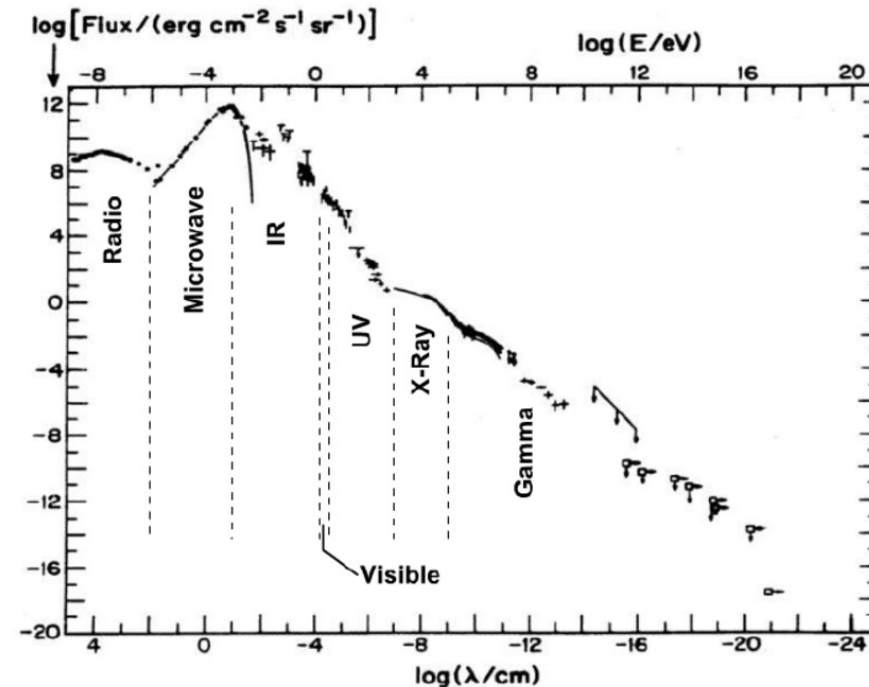
prior

evidence

$A \cap C$

Why do we take logs?

- Mathematical convenience; turns multiplication into addition
- Dynamic range; floating points only store 15 digits (Anyone tried to calculate 500 Choose 498 on a calculator?)
- Numerical precision: probabilities in high-dimensional space run into underflow problems
- log-transform is monotonic, so the location of the optimum is unperturbed



Bayesian inference

Event A: unknown parameter

Event C: data

$$\log P(A \mid C) = \log P(C \mid A) + \log P(A) - \log P(C)$$

Desired
parameter
distribution

The likelihood
likelihood

Prior distribution
on the parameter

depends
only on the
data

posterior

likelihood

prior

evidence

Bayesian inference

θ : value of parameter

Y : data

$$P(\theta | Y) = P(Y | \theta) P(\theta) / P(Y)$$

↑
Distribution
to help me
find the
parameter
value

posterior

↑
This is the
likelihood

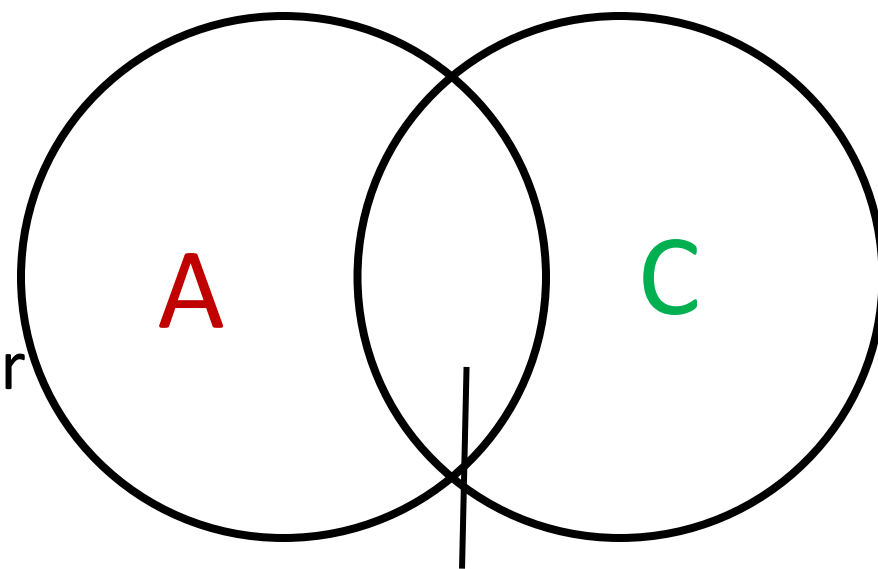
likelihood

↑
Prior
distribution
for parameter

prior

↑
Doesn't
matter for
us

evidence



$A \cap C$

Where does the prior come from?

- In some cases, there will be a correspondence between our objective function and the solution to a Bayesian inference problem.
- If we were doing this the other way, taking inference as our approach to problem-solving, we would choose prior distributions either from
 - Prior research into the values of the parameters (common in physical sciences)
 - Symmetries of the problem (geometry) that makes certain classes of solution equivalent to certain other classes of solution.
- Some of these priors don't have finite integrals... "improper priors" usually the likelihood forces the product to converge; if it doesn't, we have to take the limit of a ratio of divergent integrals.
- Sometimes, the priors are chosen because they are easy to calculate (good and bad)

Choices

- Location parameters (translational invariance) suggest uniform or “flat” priors. (Super easy ! just add dx !)
- Scale parameters that act by multiplication suggest priors proportional to $\frac{d\lambda}{\lambda}$. (But wait, doesn't that diverge? Yes. But the posterior shouldn't, unless the data fail to constrain your parameter.)
- Problems with rotational symmetry suggest priors for, for instance slope parameters, with uniform arctangents.
- Some problems (related to sampling) have prior distributions that are closely connected to the sampling process; these priors have the same form (but with different parameters) as the posteriors. These are called **conjugate priors**

Homework guidelines

- No particular tools are required (python, R, javascript..) but we're better able to help you in python and R.
- Graphs are expected to a medium-high standard. Labels, units everywhere where needed. Caption-like explanations of graphs preferred.
- Upload PDFs of answers to Canvas. Separately upload code (to a low standard) that you used to solve the homework. We probably won't look at the code.