DATA221 Intro Machine Learning
02 objective function

William Trimble
Spring 2023

THE UNIVERSITY OF CHICAGO

TECHNOLOGY

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

March 27, 2022 · 7:00 AM ET

SHANNON BOND

Caught in the wild!

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

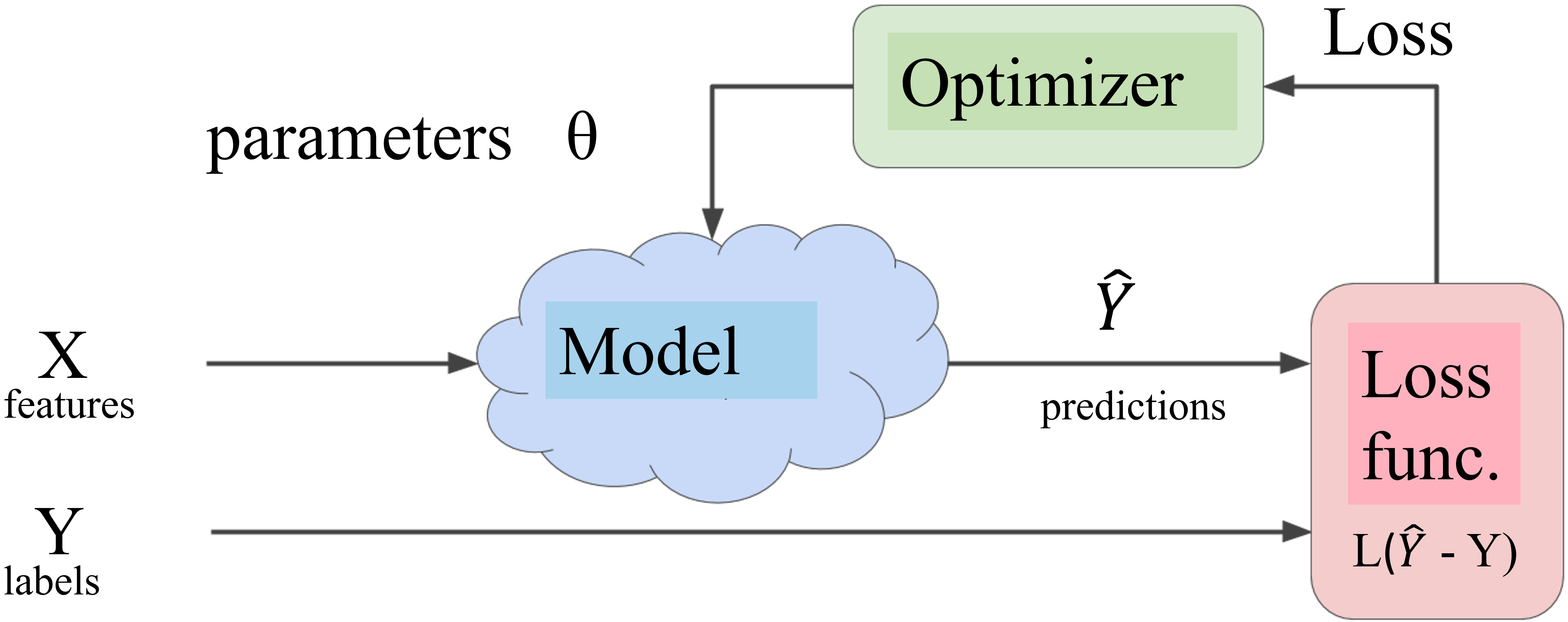
Some of the likely AI-generated faces from fake LinkedIn profiles identified by Stanford University researchers. The central positioning of the eyes is a telltale sign of a computer-created face. Click on the animation to pause.

Source: Stanford Internet Observatory

Credit: Connie Hanzhang Jin/NPR

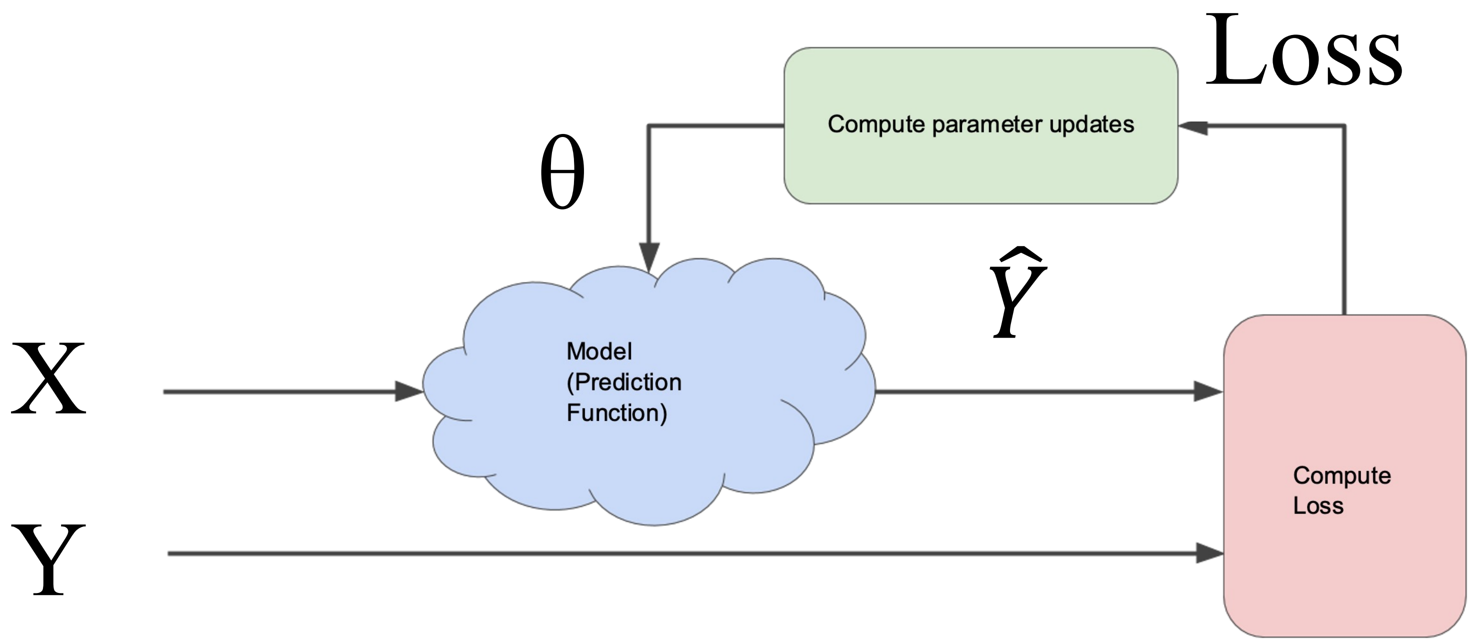
https://www.npr.org/2022/03/27/1088140809/fake-linkedin-profiles

Fairy dust picture of optimization



Argmin symmetries

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



This metafunction has some symmetries:

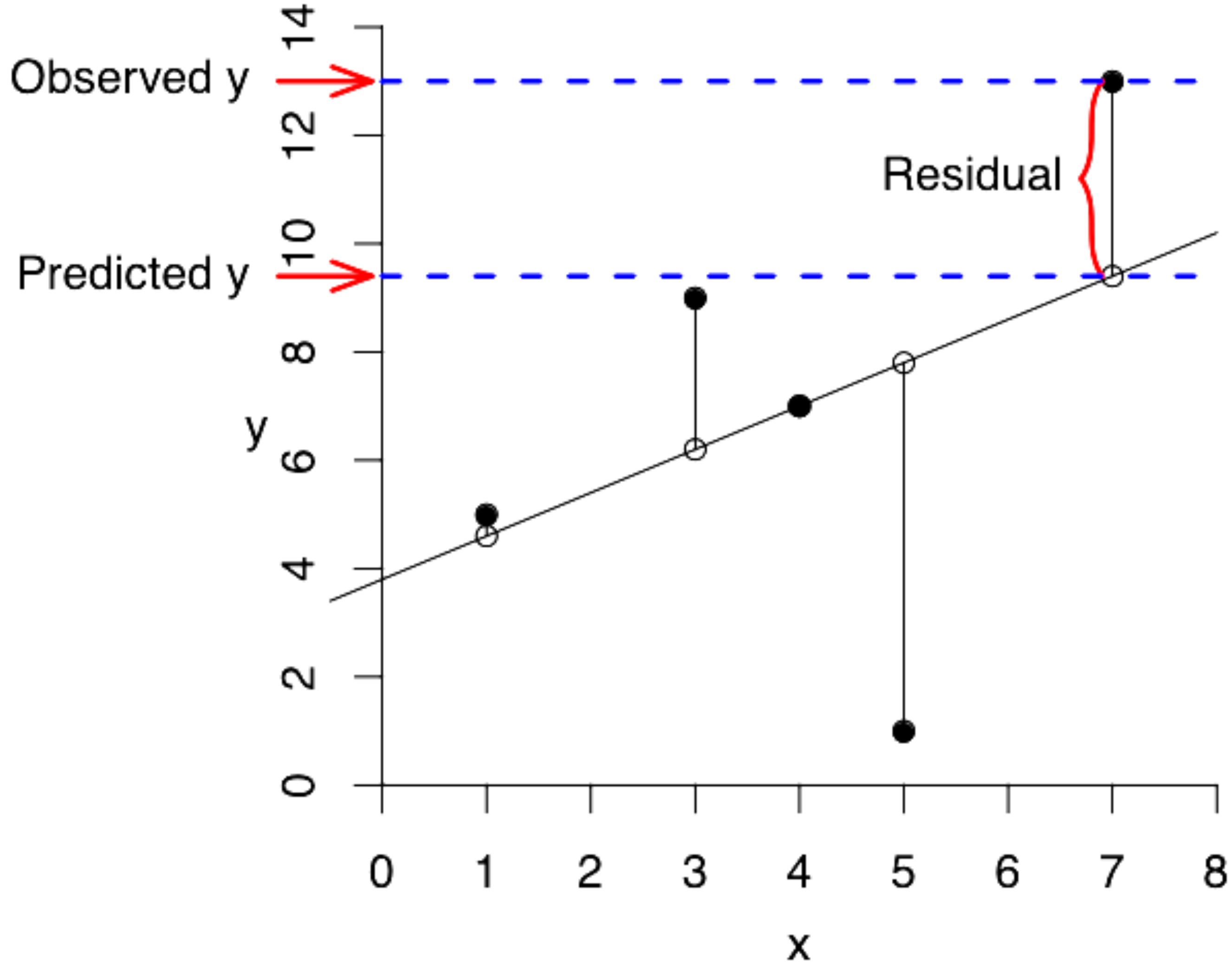
$L+c$ has same argmin as L

cL has same argmin as L

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

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

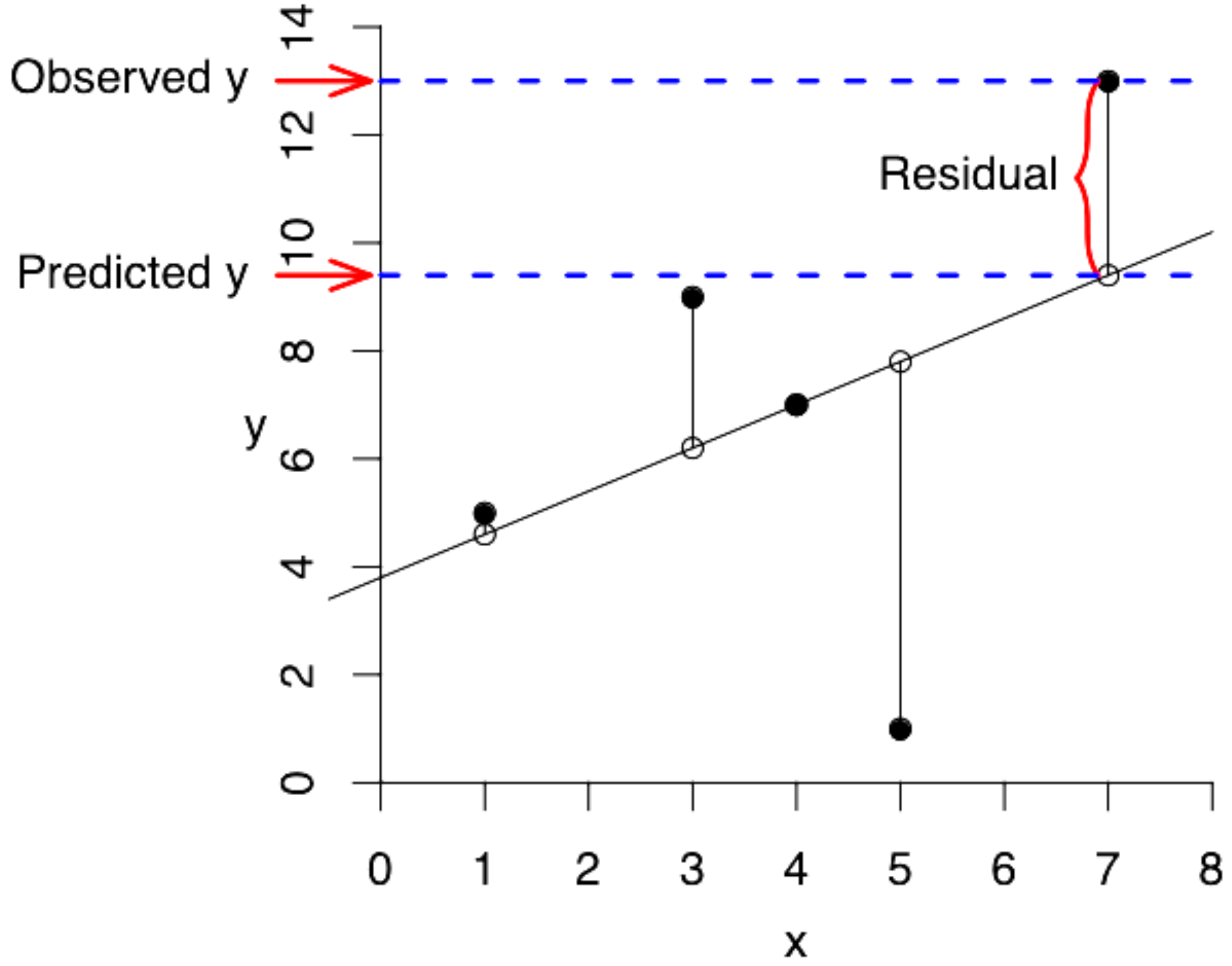
“Regression” = prediction of values



Summed squared error

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

"Regression" = prediction of values



Summed squared error

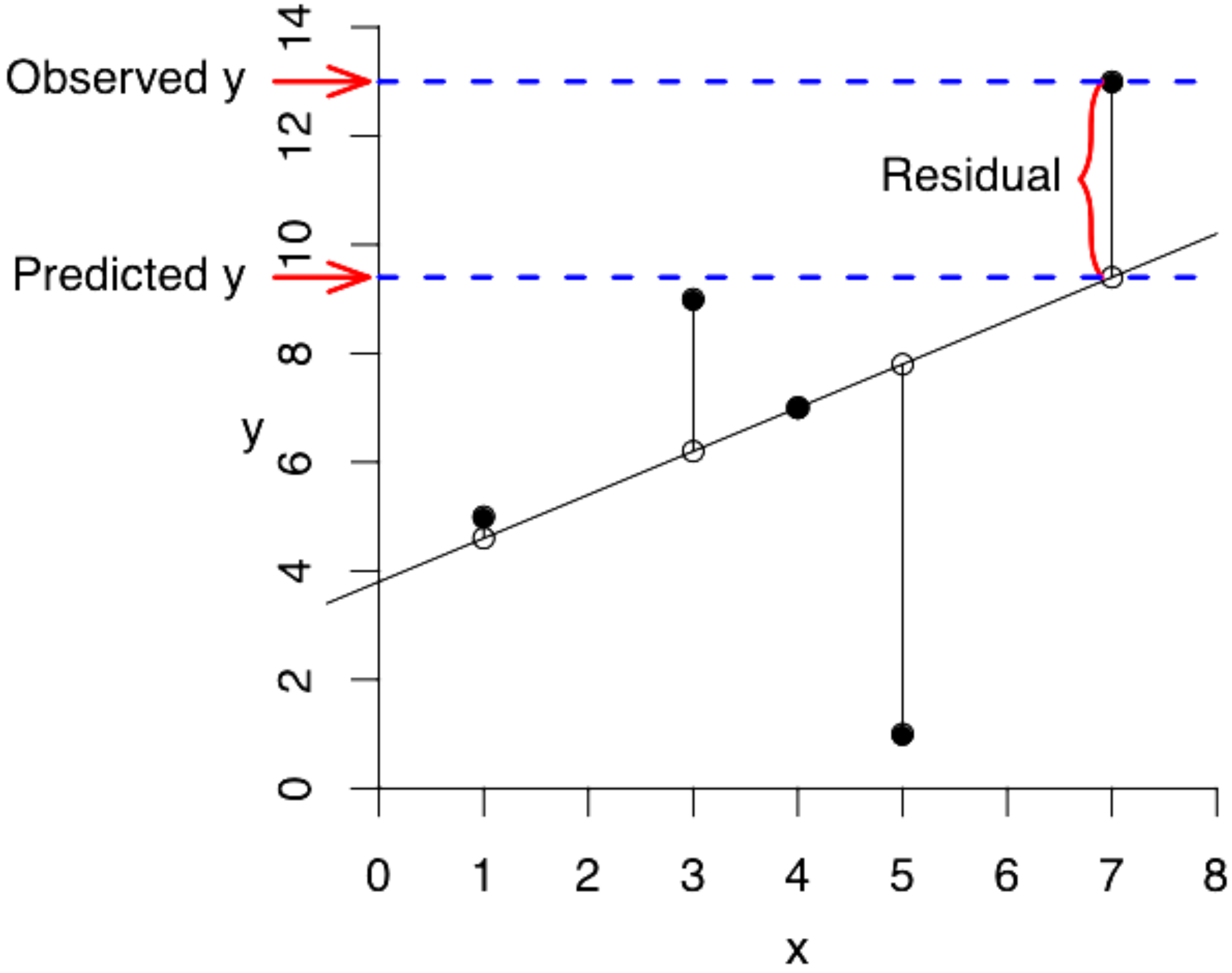
$$SSE = \Sigma\ (\hat{y}_i - y_i\ )^2$$

Root mean square

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

WHY?

“Regression” = prediction of values



Summed squared error

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

Root mean square

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

Summed absolute error

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

I don't have to calculate these because of the argmin symmetries.
These are monotonic functions of SSE and SAE

Mean squared error

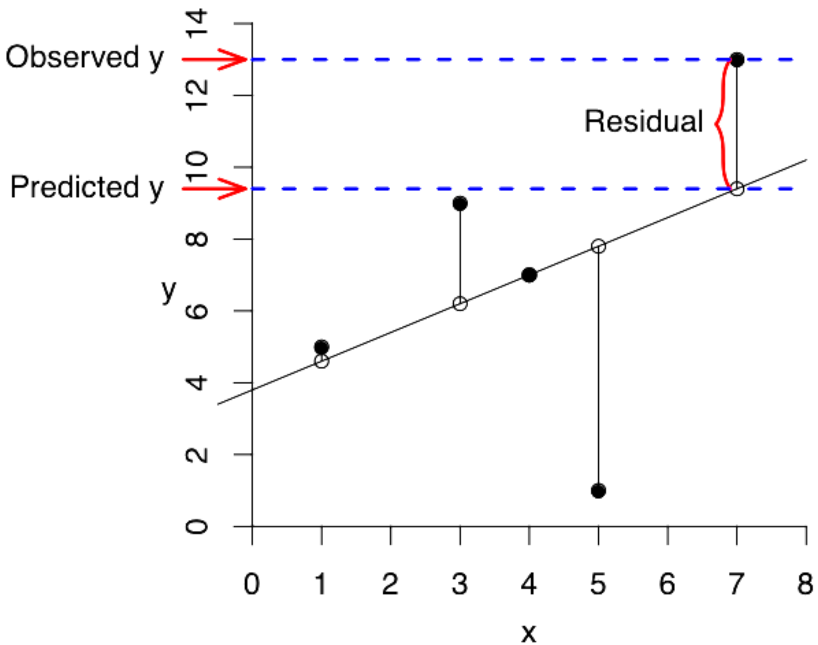
$$MSE = \frac{\Sigma\,(\hat{y}_i - yi\,)^2}{n}$$

Root mean square

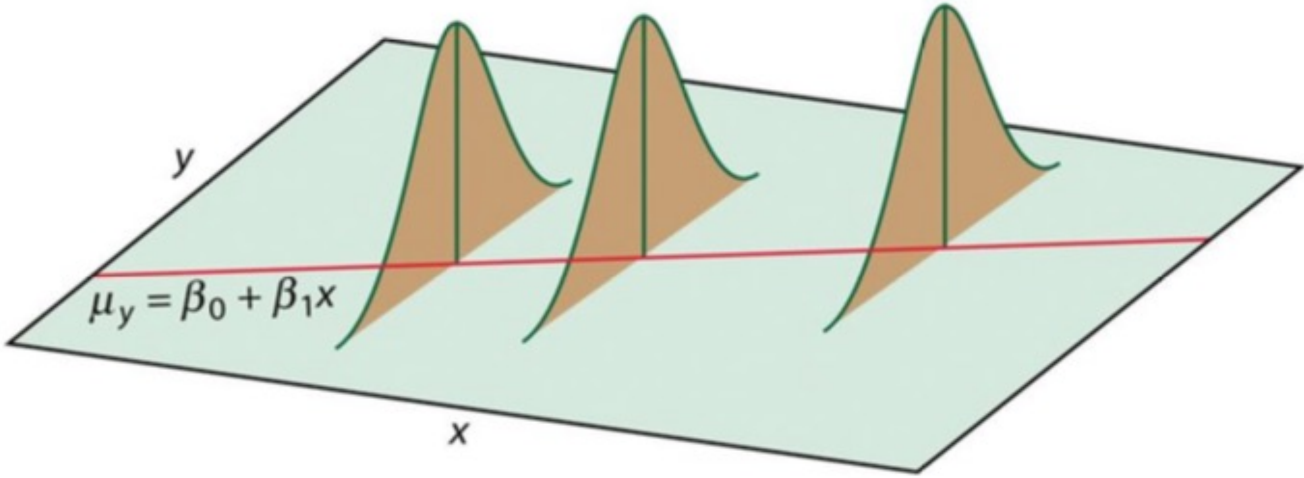
$$RMS = \sqrt{\frac{\Sigma\,(\hat{y}_i - yi\,)^2}{n}}$$

Mean absolute error

$$MAE = \frac{\Sigma\,|\hat{y}_i - yi|}{n}$$



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

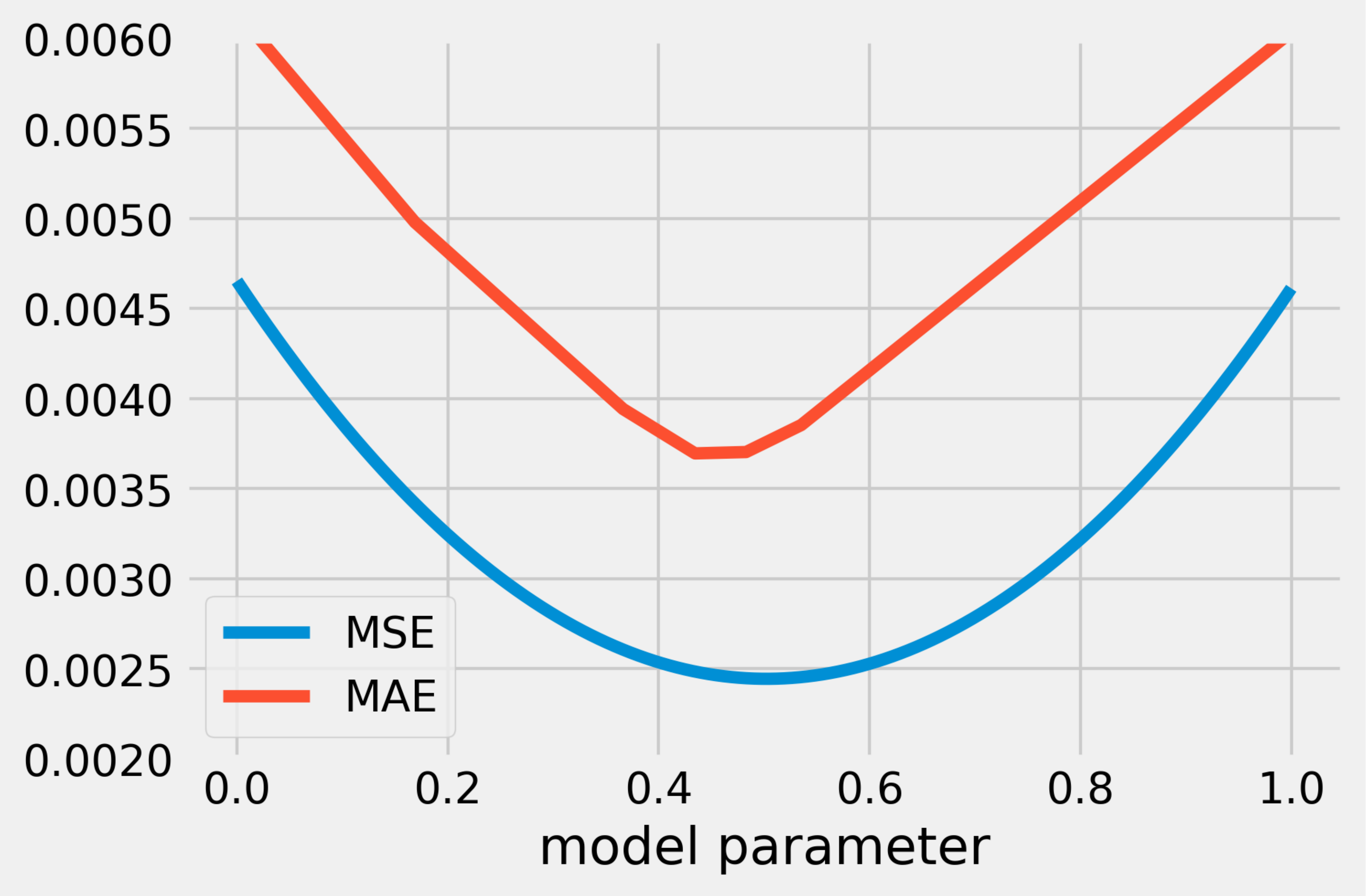
- When the objective function looks like (is proportional to) a log-probability function, optimizing it looks a lot like maximum likelihood / maximum a posteriori estimation.
- Sum squared errors.. are the log of additive normal error distributions
- Certain regularization terms , like w^Tw, solve the 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

Contrast L2 and L1



Discontinuities in the slope of mean absolute error.

Not analytic, not smooth…

Derivative won't vanish anywhere

Threshold effect when

$$\frac{\partial L}{\partial \theta} < \alpha$$

Contrast L2 and L1

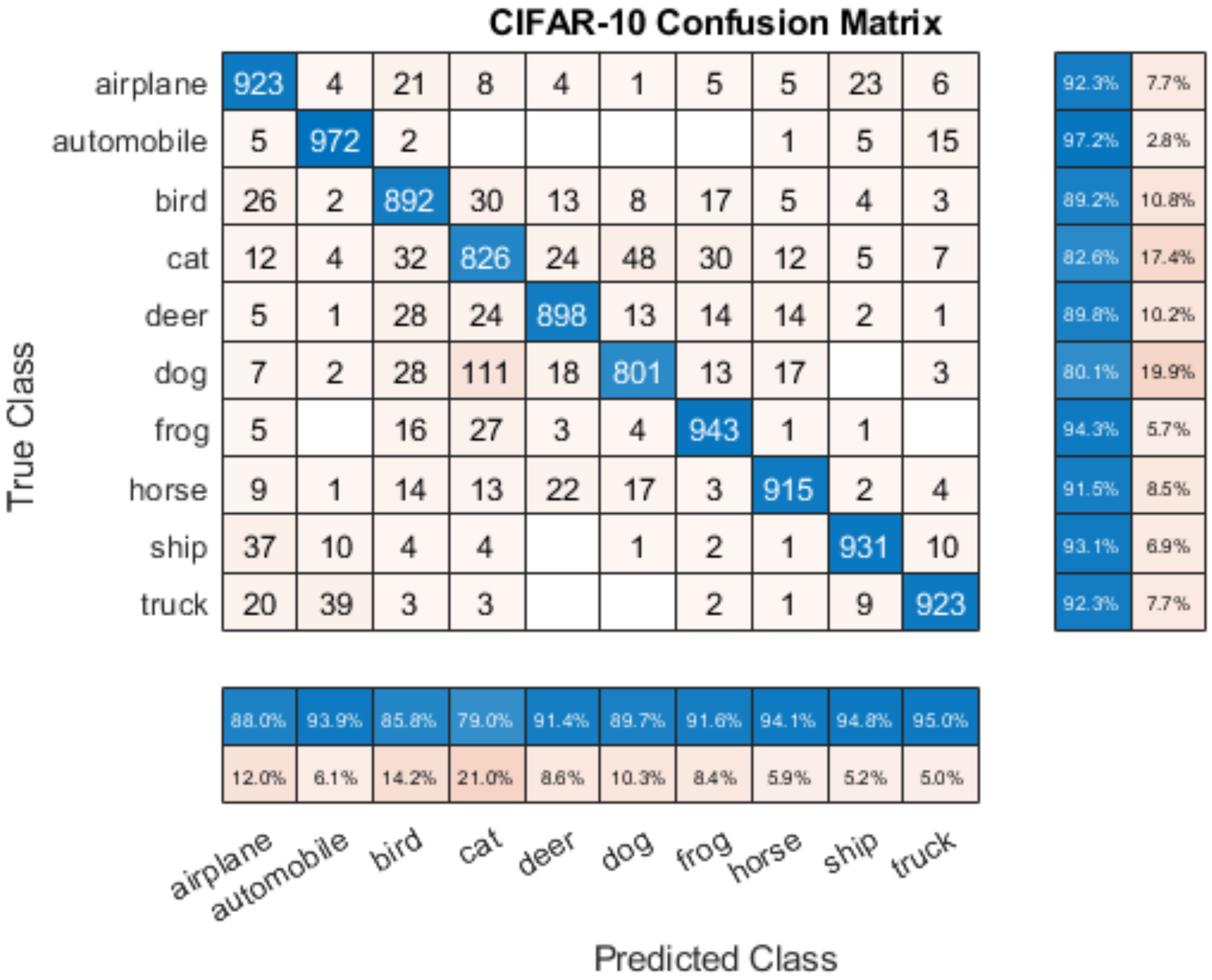
- L2 in the residuals gives the same solution as optimizing additive normally distributed errors.

- 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



Expected loss

For unknown state of nature P(ωj | 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(ωj | x)

- R(α $_{\mathrm{i}}$ | **x**) = $\sum_j \lambda(\alpha_i|\,\omega_j)$P(ωj | **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 | likely 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.

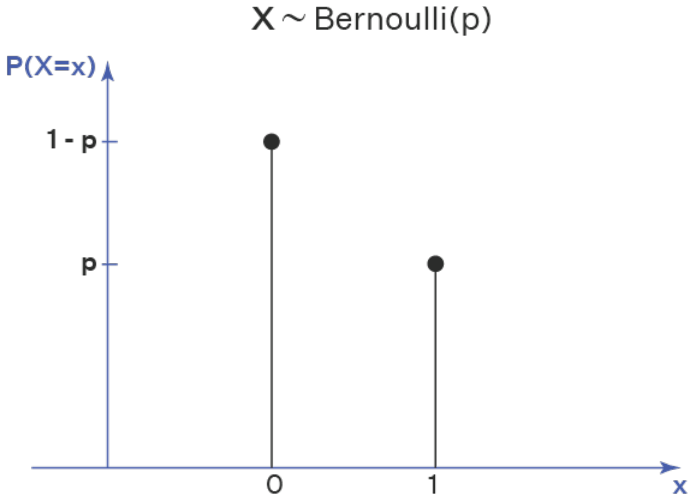
$$\lambda_{ij} = 1 - \delta(i, j)$$

This counts the number of errors.

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

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?



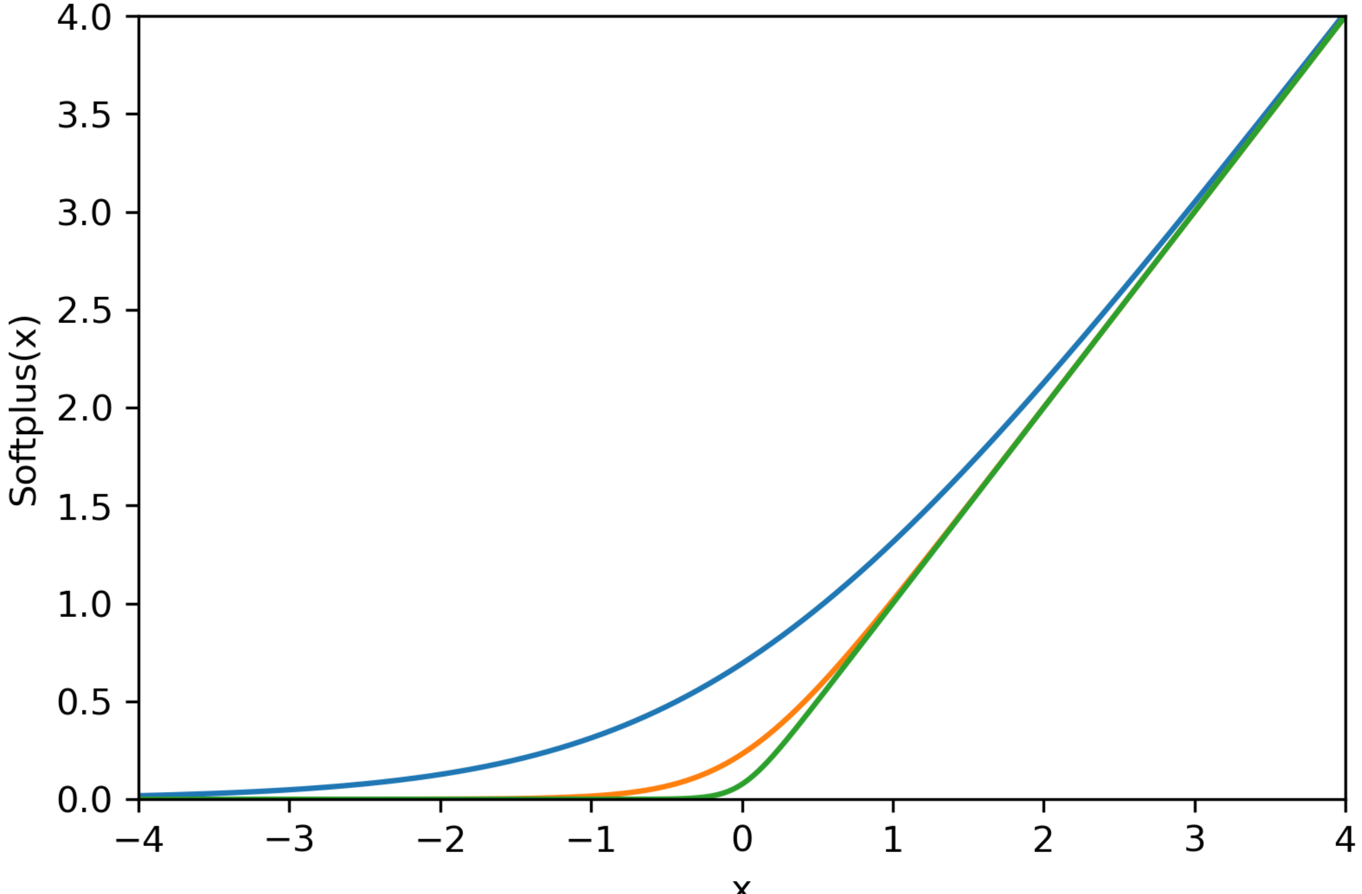
Loss (min(max(0, x), 1))   is not a good choice.

$\text{Loss}_{\text{CONSTRAINED}} = \text{Loss} + \text{HUGE}\,(x > 1) + \text{HUGE}\,(x < 0)$

also not a good choice

Softplus

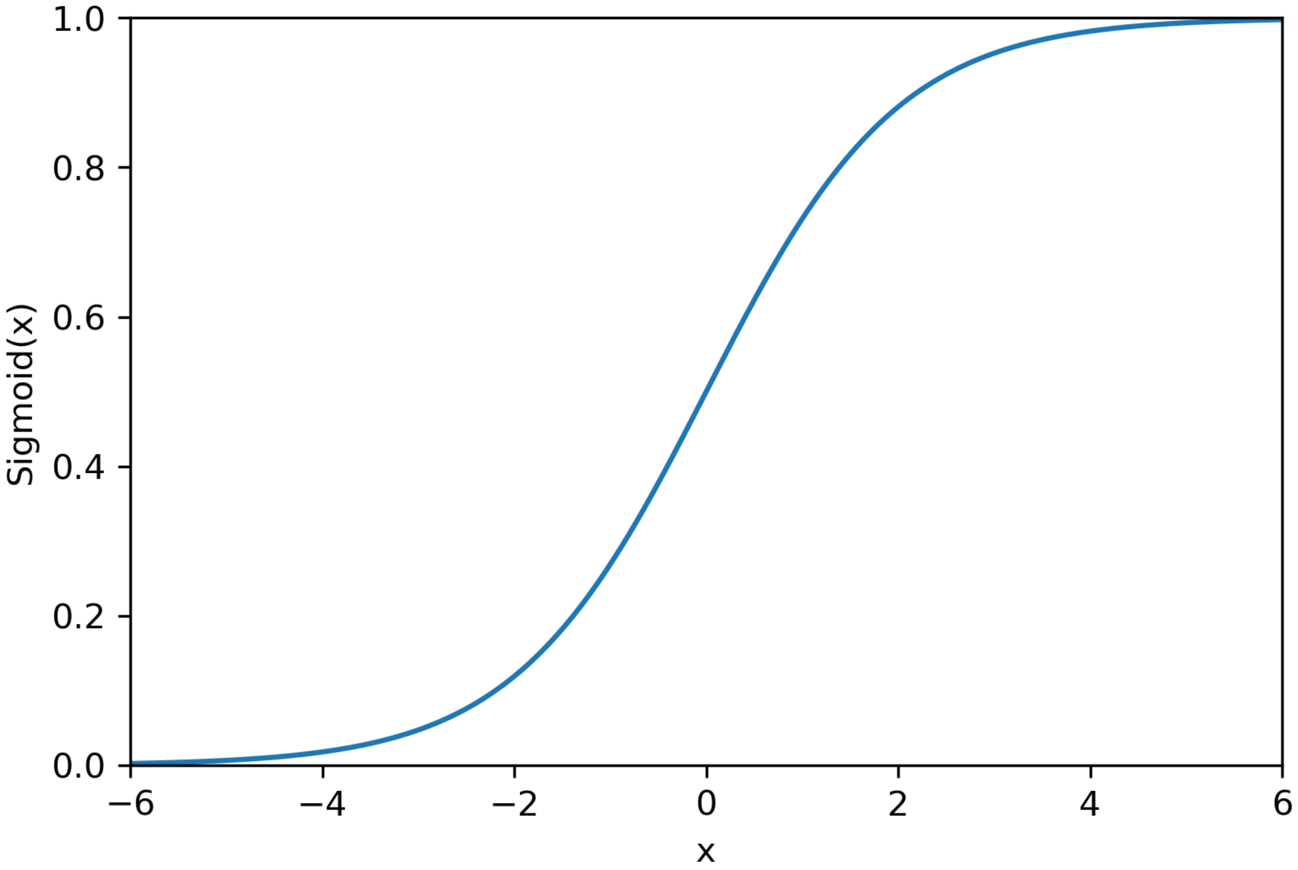
$$softplus(x) = \log(\,1 + \exp(x))$$



- gentler form of max(0, x)

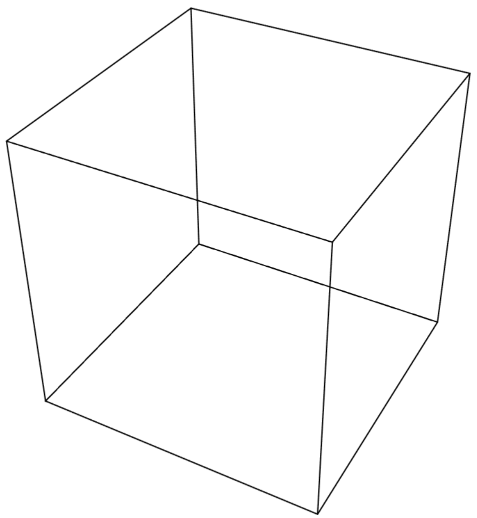
Sigmoid function

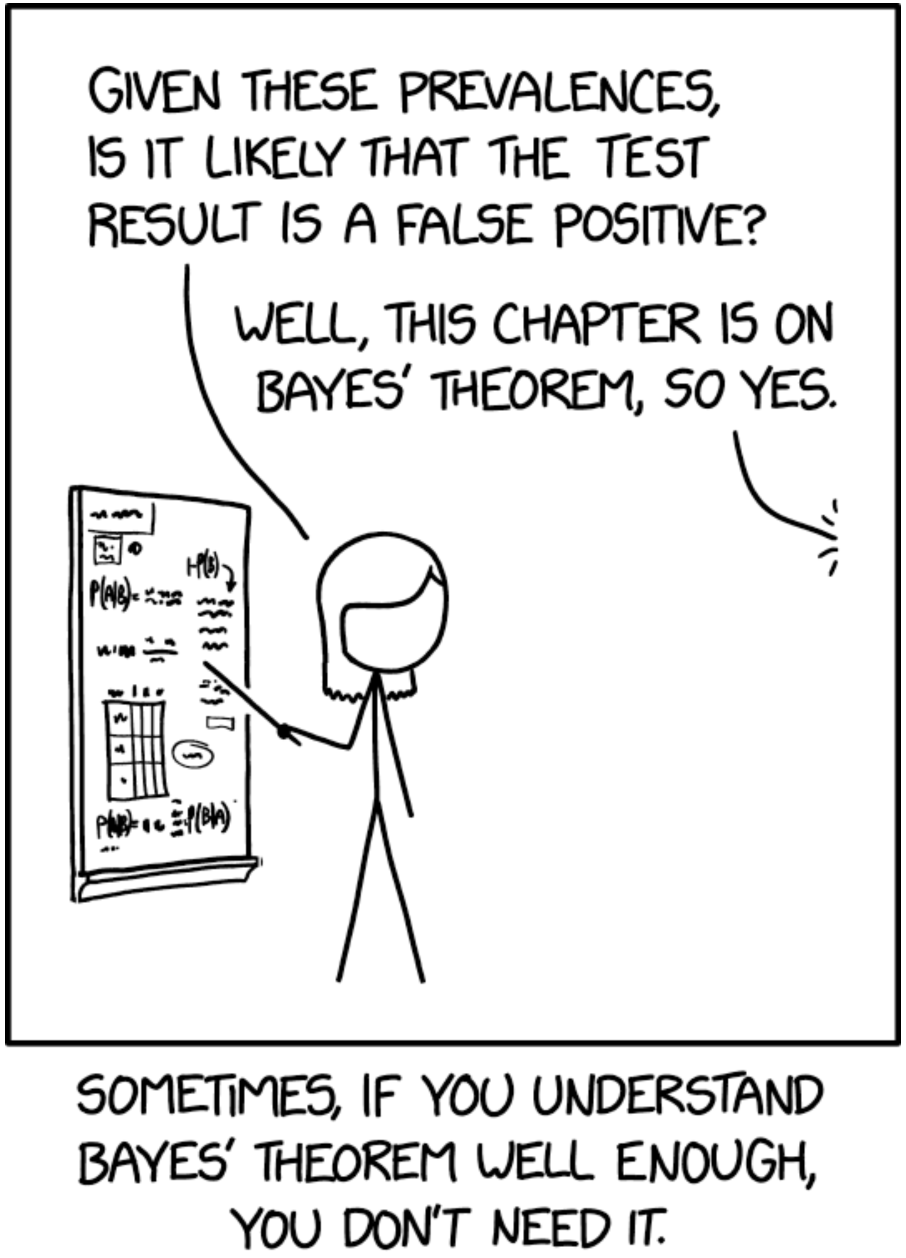
$$sigmoid(x) = \frac{\exp(x)}{1 + \exp(x)}$$



- gentler form of 0.5* sign(x) + 0.5
- continuous mapping from $\mathbb{R}$ to (0,1)

$\mathbb{R}^3$ -> 1^3

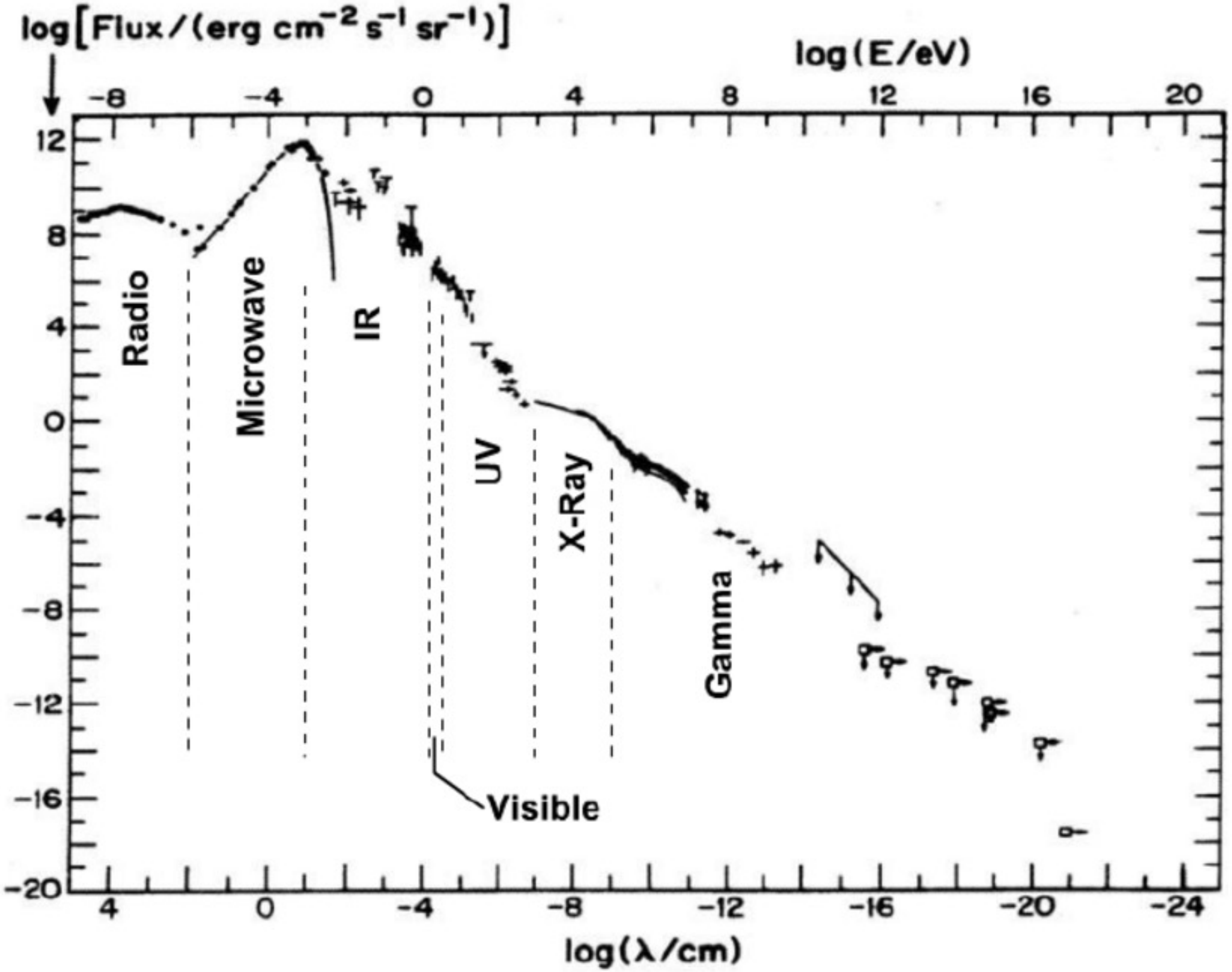




In favorable cases, we can use Bayes’ theorem to estimate our parameters.

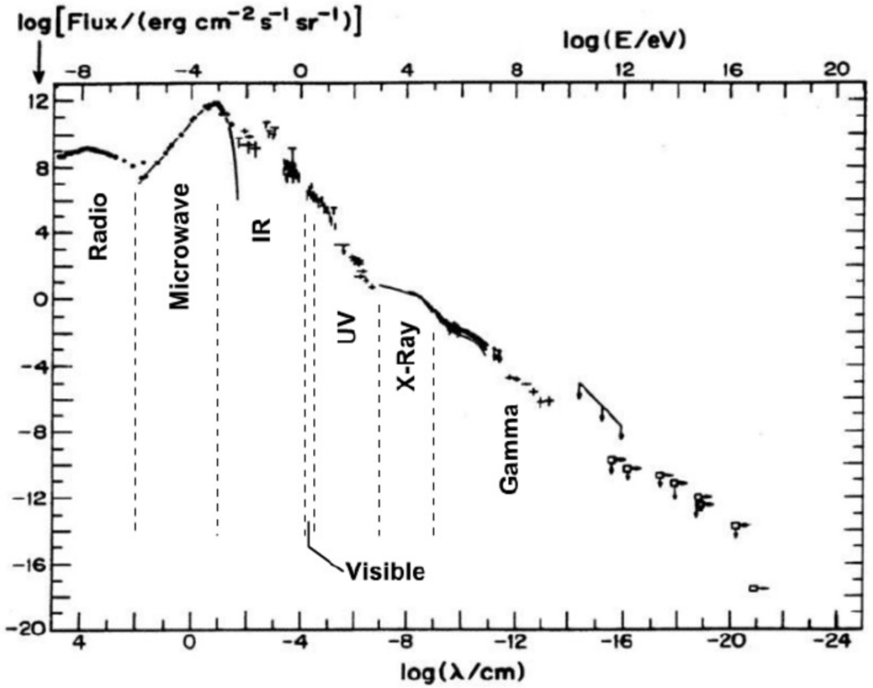
In less favorable cases, we can do a randomized search for possible parameter values.

Why do we take logs?



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

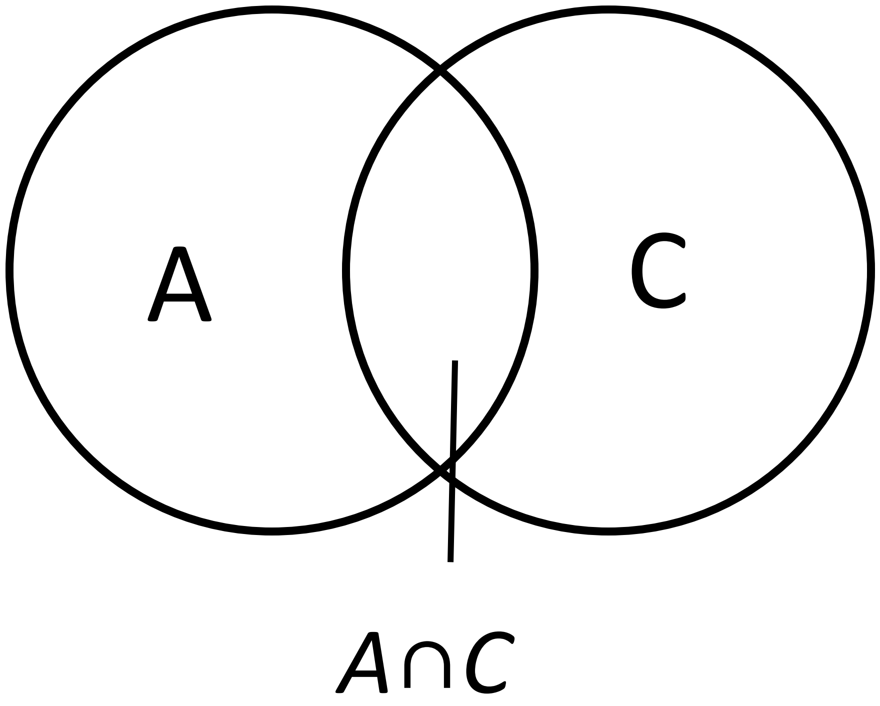


Bayesian inference

Event A: unknown state of nature

Event C: experiment

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

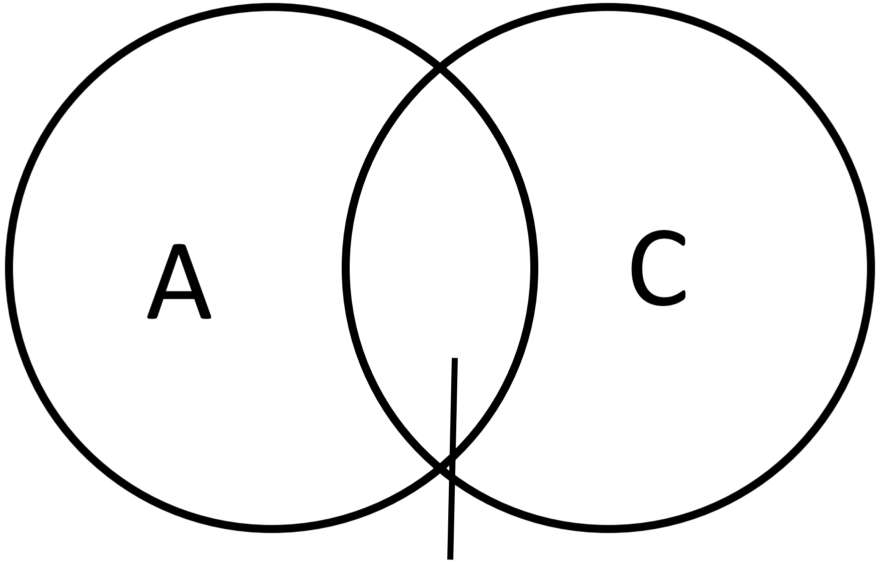


Bayesian inference

Event A: unknown state of nature

Event C: experiment

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



$A \cap C$

"Numerical decisionmaking for grownups"

Bayesian inference

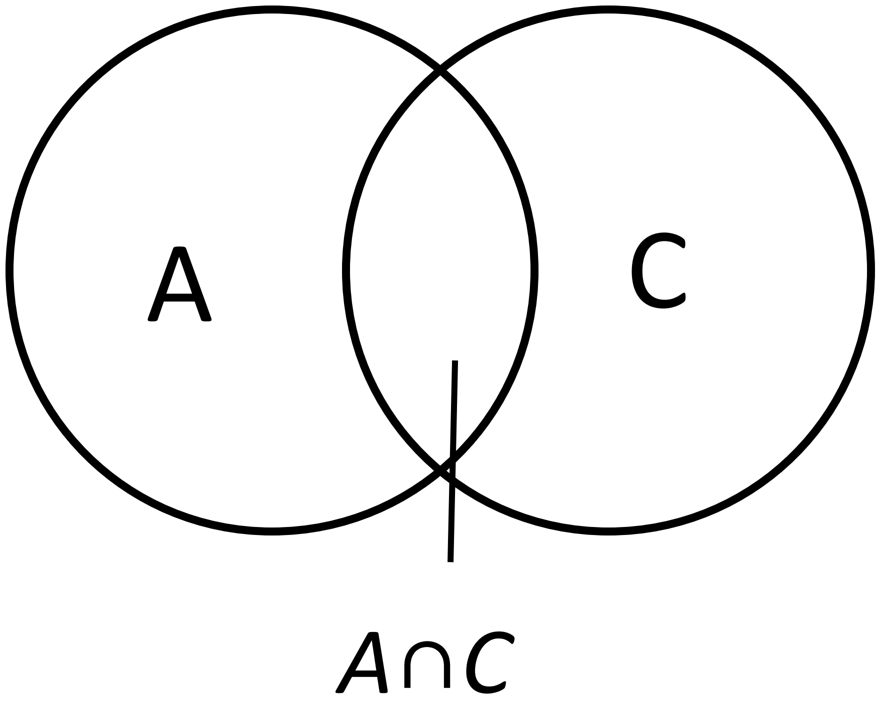
Event A: am I infected?

Event C: experiment (test result)

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

The thing I want to know

The thing the FDA wants know



Bayesian inference

Event A: unknown state of nature

Event C: experiment

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

Life's persistent question

Term that depends on experiment

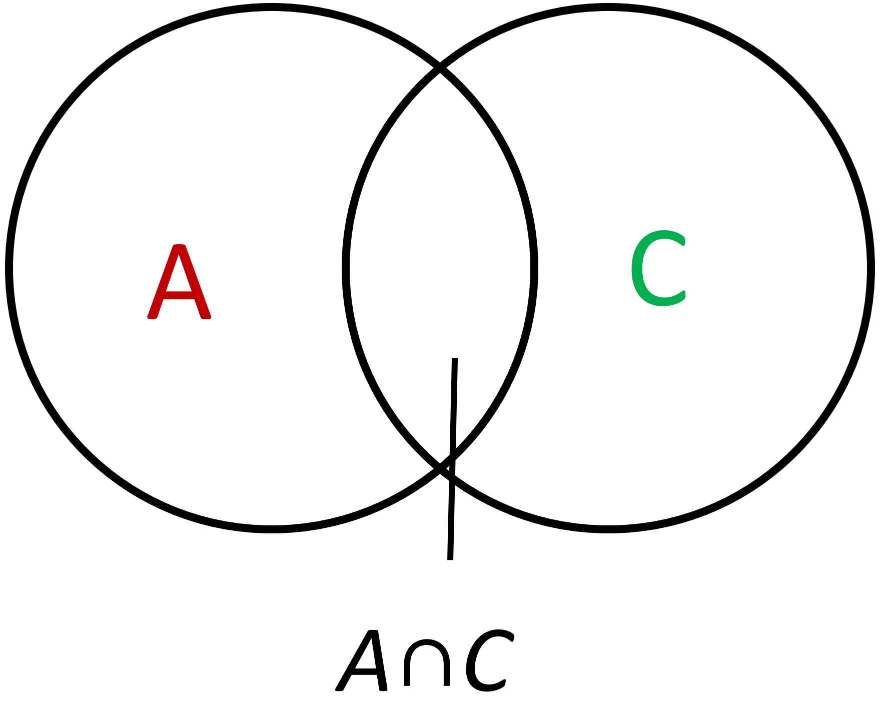
likelihood

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

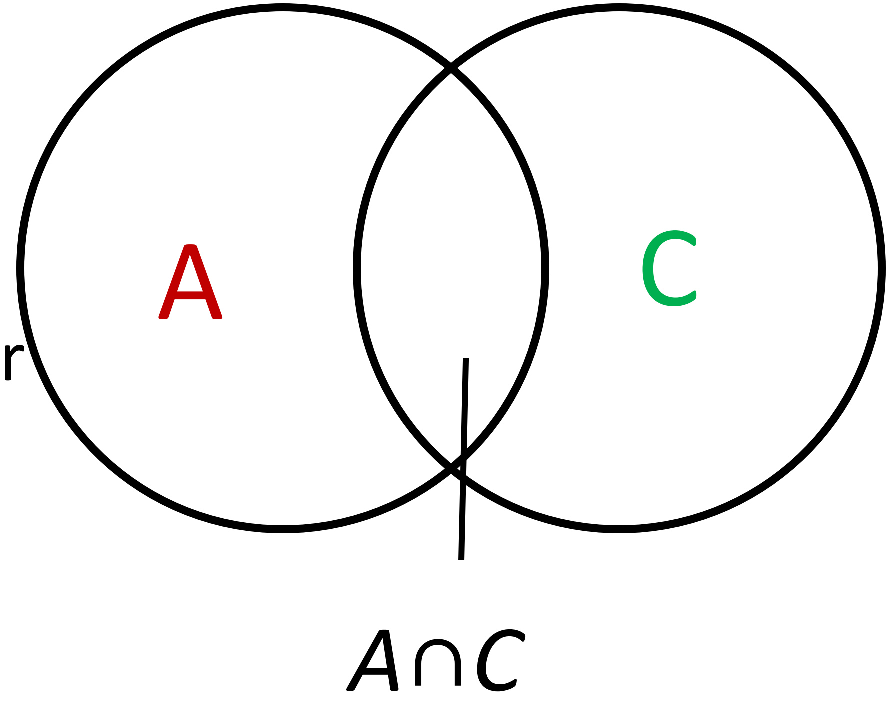
Doesn't matter for us

posterior

likelihood

prior

evidence



Log (Bayesian inference)

Event A: unknown parameter

Event C: data

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

Desired parameter distribution	The likelihood *likelihood*	Prior distribution on the parameter	depends only on the data
posterior	*likelihood*	*prior*	*evidence*

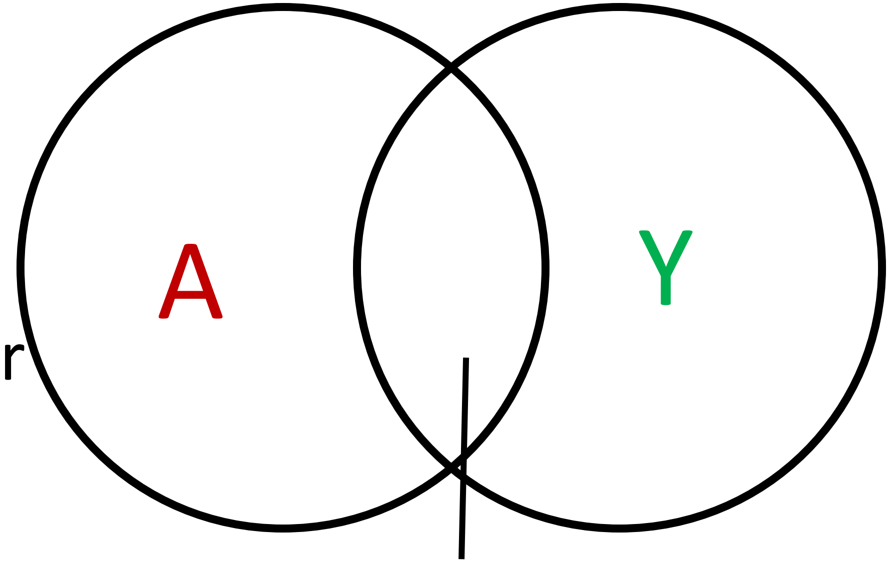
Bayesian inference

θ: value of parameter

Y: data

$$P(\theta \mid Y) = P(Y \mid \theta)\, P(\theta) \,/\, P(Y)$$

$P(\theta \mid Y)$	$P(Y \mid \theta)$	$P(\theta)$	$P(Y)$
Distribution to help me find the parameter value	This is the likelihood	Prior distribution for parameter	Doesn’t matter for us
posterior	*likelihood*	*prior*	*evidence*



A∩Y

Bayesian inference for sequential

θ: value of parameter

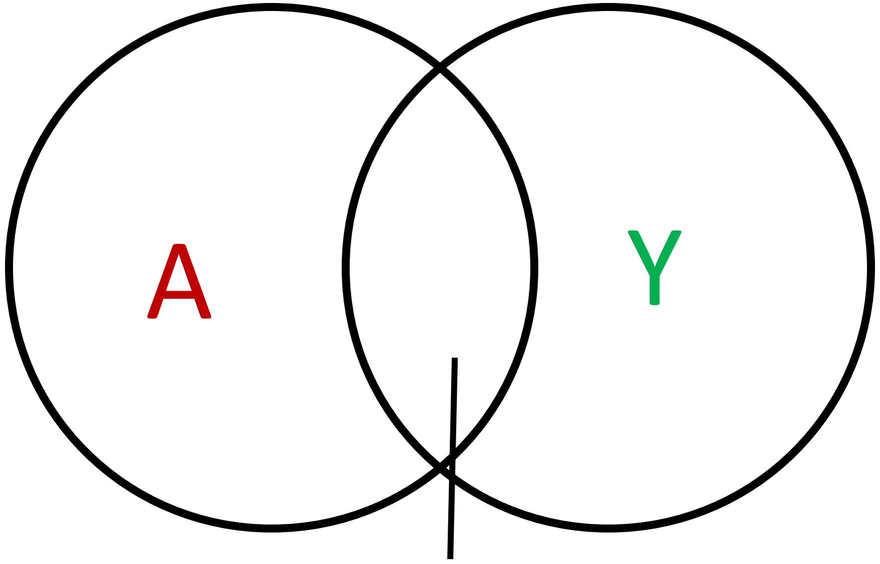
Y: data

$$P(\theta \mid \{Y_i\}) = \prod P_i(Y|\theta)\, P(\theta)$$

Posterior density

Product of likelihoods of independent observations

One factor of the prior on the parameter



$A \cap Y$

posterior

likelihood

prior

Sequential Bayesian inference for continuous parameters: MacKay's example

θ: λ value of parameter

Y: $\{y_i\}$ data

$$P(\lambda \mid \{y_i\}) = \Pi \frac{1}{\lambda} e^{-y/\lambda} \frac{1}{\lambda}$$

Posterior density
(a function of λ)

Product of likelihoods of independent observations
(a function of λ)

One factor of the prior
(a function of λ)

posterior

likelihood

prior

Typical setup for ML interpreted as inference

λ : value of parameter

Y: $\{y_i\}$ data

$$\log P(\lambda \mid \{y_i\}) = \Sigma LL(y,\lambda) + \mathit{regularization\ term}(\lambda)$$

Posterior density

Sum of log-likelihoods of independent observations

One factor of the prior
(a function of λ)

posterior

likelihood

prior

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.
- Graphs for projects must be much, much better; expect to adjust the font size on everything that has writing.
- Upload 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.
- Unevaluated code can't be marked.

Office Hours

- WT: 1st week Friday 2-3:30
- 2nd week Wednesday 12:30-2:00
- Friday 12:30-2:00

Poll?