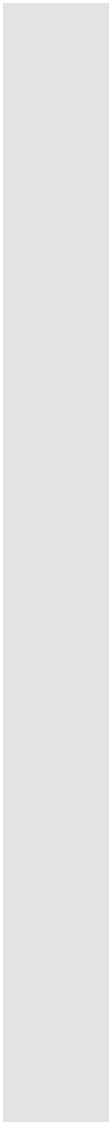


ECON7960 User Experience and A/B Test

Hong Kong Baptist
University

Topic 9: Google
Optimizer and
Bayesian



In previous
topic 8, we
cover

Google Analytics in
Ecommerce



Boosting
Algorithms



Boosting
Application

A Review on what you learn from Boosting

- Please use your phone to download an app or web <http://www.socrative.com> ("SOCRATIVE" student version) and open it, you should see

Enter the Room Name
"HUNG5085"

The quiz will start at 6:30pm before the lecture and lasted for 15 minutes.



Student Login

Room Name

HUNG5085

JOIN

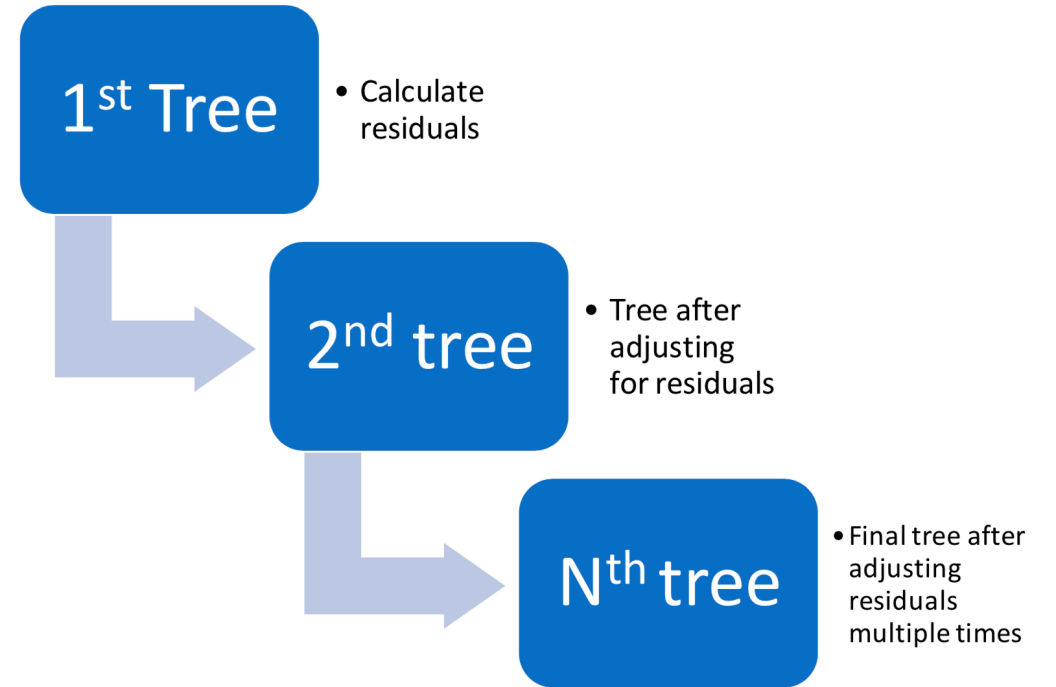
 English ▾

Boosting

Process of turning a weak learner into a strong learner

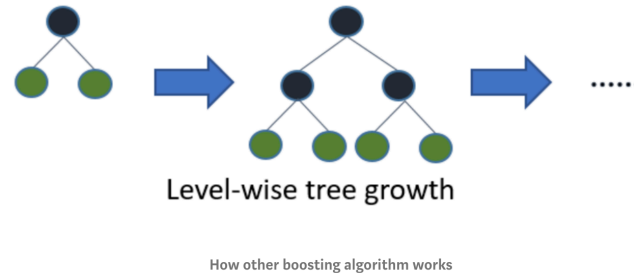
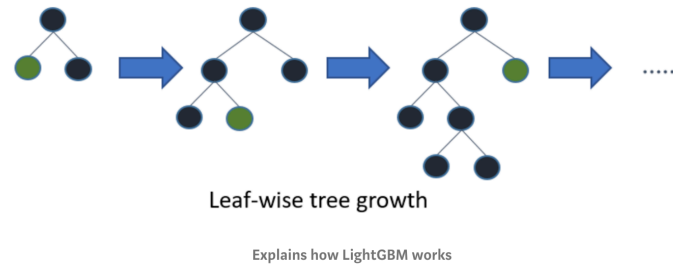
There are three algorithms of boosting Gradient Boost, Ada Boost and XG Boost based on different methodologies.

Gradient Boosting



Gradient Boosting works by sequentially adding predictors to an ensemble, each one correcting its predecessor. However, instead of tweaking the instance weights at every iteration like AdaBoost does, this method tries to fit the new predictor to the residual errors made by the previous predictor.

LightGBM is using Gradient Boosting

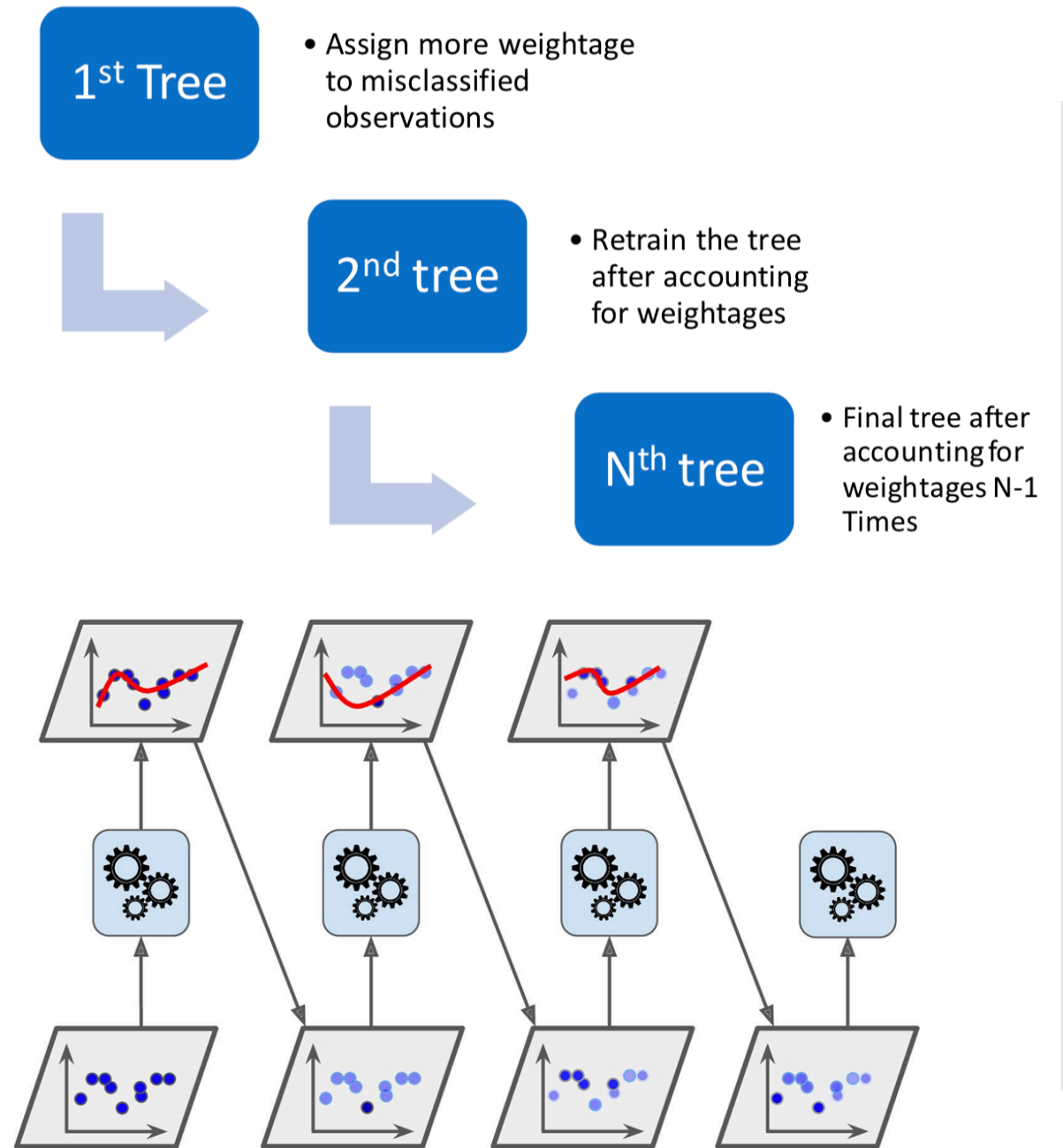


- LightGBM is a new gradient boosting tree framework, which is highly efficient and scalable and can support many different algorithms including GBDT, GBRT, GBM, and MART. LightGBM is evidenced to be several times faster than existing implementations of gradient boosting trees, due to its fully greedy tree-growth method and histogram-based memory and computation optimization.

Ada Boosting

To build an AdaBoost classifier, a first base classifier (such as a Decision Tree) is trained and used to make predictions on the training set. The relative weight of misclassified training instances is then increased. A second classifier is trained using the updated weights and again it makes predictions on the training set, weights are updated, and so on

Ada Boosting



XGBoosting

Salient features of XGBoost which make it different from other gradient boosting algorithms include:

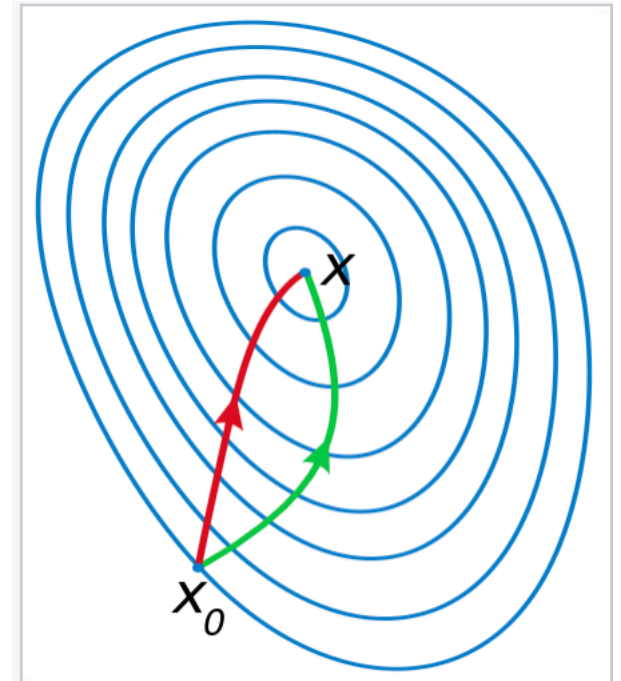
- Clever penalization of trees
- A proportional shrinking of leaf nodes
- **Newton Boosting**
- Extra randomization parameter

Almost similar on Gradient Boost

- XG-boost used a more regularized model formalization to control over-fitting, which gives it better performance.
- For model, it might be more suitable to be called as regularized gradient boosting.

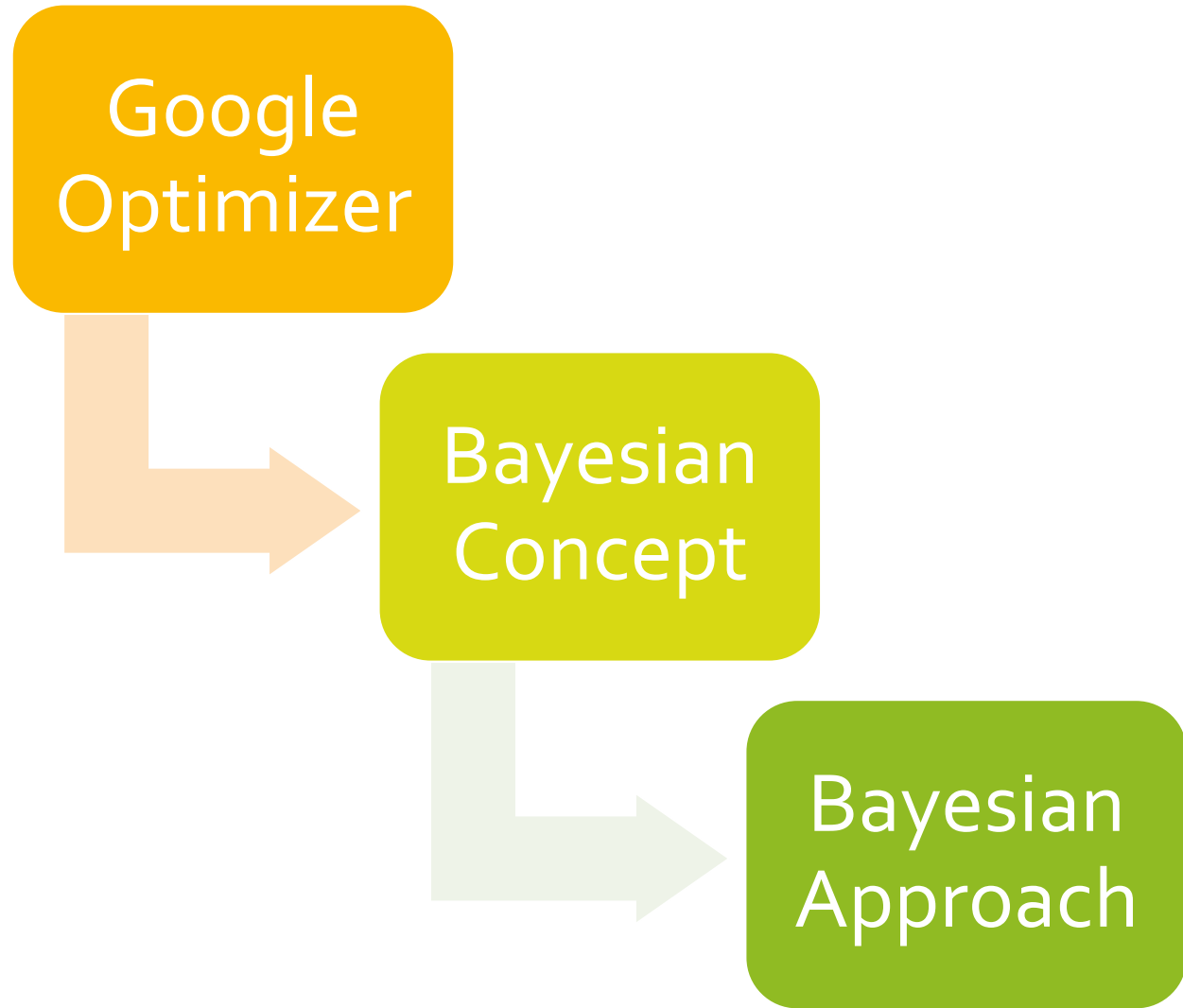
Regularization

- The cost function we are trying to optimize (MSE in regression etc) also contains a penalty term for number of variables. In a way, we want to minimize the number of variables in final model along with the MSE or accuracy. This helps in avoiding overfitting
- XG-Boost contains regularization terms in the cost function.



A comparison of **gradient descent** (green) and Newton's method (red) for minimizing a function (with small step sizes). Newton's method uses **curvature** information (i.e. the second derivative) to take a more direct route.

In topic 9, we
will do



Bayesian Approach

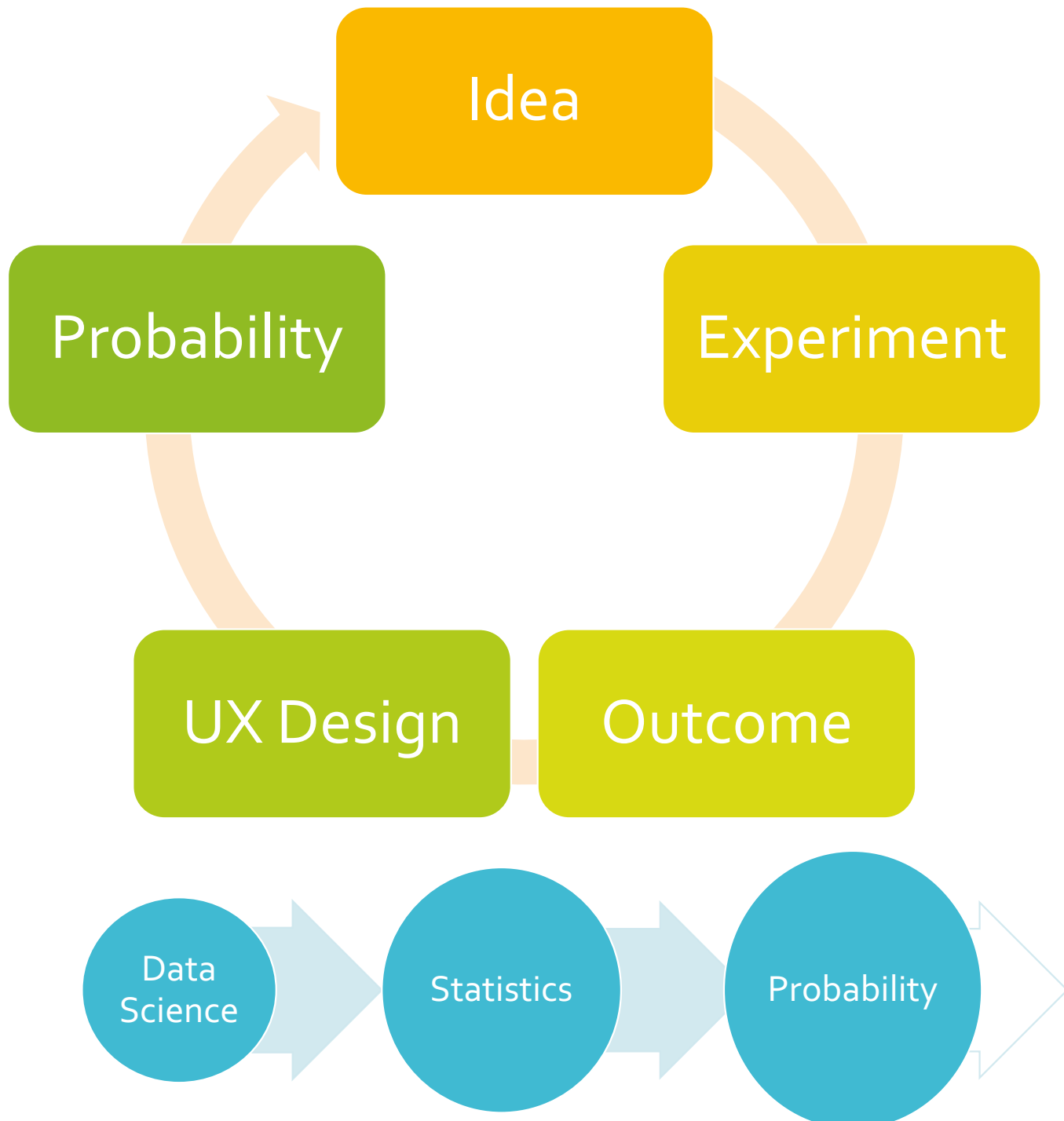
Understanding Bayes Theorem

Solve Statistical Problems with Bayesian Estimation

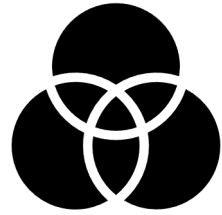
Bayesian Method in Predictive Analysis

Probabilistic Approach against Frequentist Approach

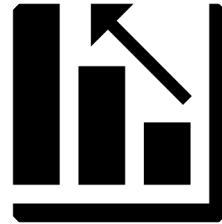
recognize most phenomena are difficult to grasp because one generally has to deal with incomplete and/or noisy data, we are intrinsically limited by our evolution-sculpted primate brain, or any other sound reason you could add. As a consequence, we use a modelling approach that explicitly takes uncertainty into account.



Why Frequentist Approach is not Appropriate



Apply the “known” model to
analyse the likelihood of
some event



Test whether the model
predict correctly based the
“test” sample



Use to predict the event
assuming in the long run and
under the “same” conditions

Frequentist Approach

Where

$P(X)$ = Frequentist Probability

$n(X)$ = Number of trial where the event X occurs

$N(T)$ = Total number of trials

$$P(X) = \lim_{n \rightarrow \infty} \left(\frac{n(X)}{n(T)} \right)$$

Draw conclusions from sample data by emphasizing the frequency of the data

It relies heavily on the fixed parameters

Repeating the process over and over again

Bayesian Approach

- Uses a prior belief to define a prior probability distribution on the possible values of the unknown parameters
- It takes unknown factors into consideration while making guesses based on previous observations to draw conclusion

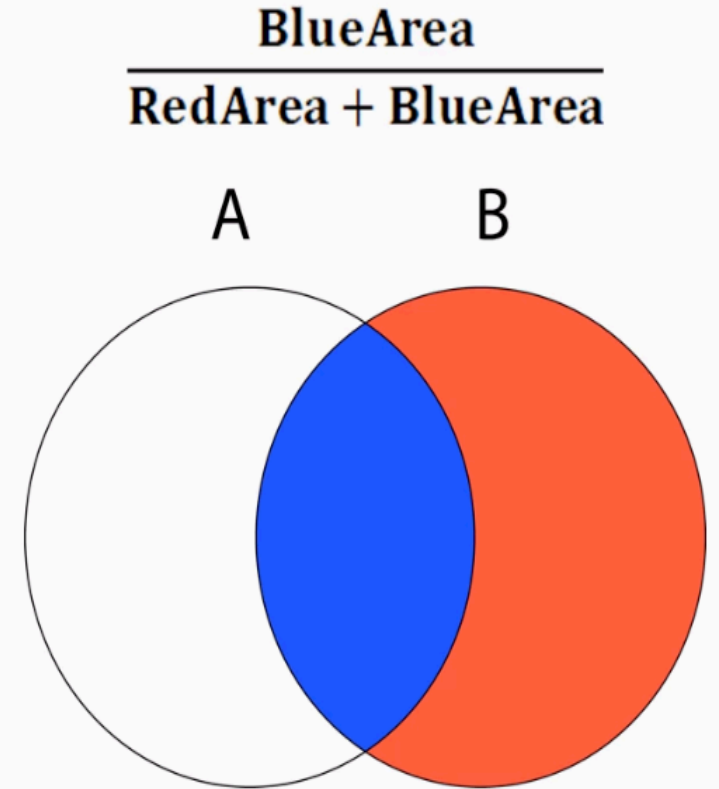
$$P(\theta|d) = \frac{P(d|\theta)*P(\theta)}{P(d)}$$

Conditional Probability is the Probability of event A given happening of event B

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

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

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

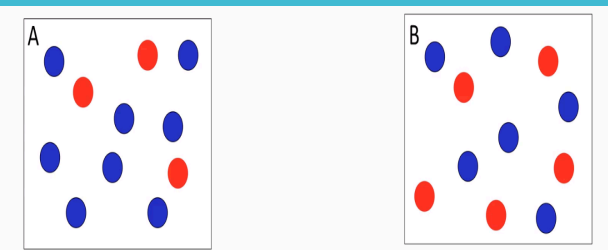


Bayes Approach

Two groups of customers with persona A and persona B:

RED (converted)

BLUE (not converted)



- Using experiment outcomes “e” to predict thing (group A or B) is always a Bayesian concept, based on conditional probability, example: In finance, rate the risk of lending money and in medical field, determine the accuracy of medial test

Likelihood

How probable is the evidence given that our hypothesis is true?

Prior

How probable was our hypothesis before observing the evidence?

$$P(H | e) = \frac{P(e | H)P(H)}{P(e)}$$

Posterior

How probable is our hypothesis given the observed evidence?
(Not directly computable)

Marginal

How Probable is the new evidence under all possible hypothesis?
 $P(e) = P(e|H)P(H)$

A Bayesian Problem

- Consider an example. Your company have two group of customers, YOUNG and ELDERLY. YOUNG contains 75 NOT LIKE YOUR PRODUCT and 25 LIKE YOUR PRODUCT, while ELDERLY contains 50 LIKE YOUR PRODUCT and 50 DISLIKE YOUR PRODUCT.
- They randomly visit your web site and you know they from one of these groups.
- Assume out of 10, only 1 convert.
- What are the probabilities for the hypotheses " H_1 : this sample is from YOUNG " and " H_2 :this sample is from ELDERLY," respectively?
- Before they randomly draw come to your site, both belief that these two groups are equally likely. After you observe the result with 10% convert, one needs to update the probability for both hypotheses according to Bayes' formula.
- Consider hypothesis , the probability from YOUTH group.

Calculation:

This result also makes sense intuitively. The probability for drawing non convert from YOUTH is twice as high as for the convert event happening with ELDERLY. Therefore, having drawn 9 non-convert visitor out of 10, the hypothesis YOUNG has with an updated probability higher than the updated probability for hypothesis ELDERLY.

- Probability of experiment outcome D: p (1 convert, 9 not convert)
- Prior : $p(\text{YOUTH}) = 0.5$
- Likelihood : $p(D | \text{YOUTH}) = 10 \times (0.25) \times (0.75)^9 = .188$
- What is $p(D)$?
 - $= p(D | \text{YOUTH}) p(\text{YOUTH}) + p(D | \text{ELDERLY}) p(\text{ELDERLY})$
 - $= 10 \times (0.25) \times (0.75)^9 \times .5 + 10 \times (.5) \times (0.5)^9 \times 0.5$
 - $= .094 + .005$
 - $= .099$
- This gives for the updated probability of web visitor are
 - $p(\text{YOUTH} | D) = \{p(D | \text{YOUTH}) \times p(\text{YOUTH})\} / p(D)$
 $= .5 \times .188 / .099$
 $= .95$
- Concluded that this batch of visitors has this observation outcome of conversion that are not half YOUNG and half ELDERLY. It is 95% of youth.

PyMC3: A Python package

Theano is a Python library that was originally developed for deep learning and allows us to define, optimize, and evaluate mathematical expressions involving multidimensional arrays efficiently. PyMC3 uses Theano because some of the sampling methods, such as NUTS, need gradients to be computed, and Theano knows how to compute gradients. Theano compiles Python code to C code, and hence PyMC3.

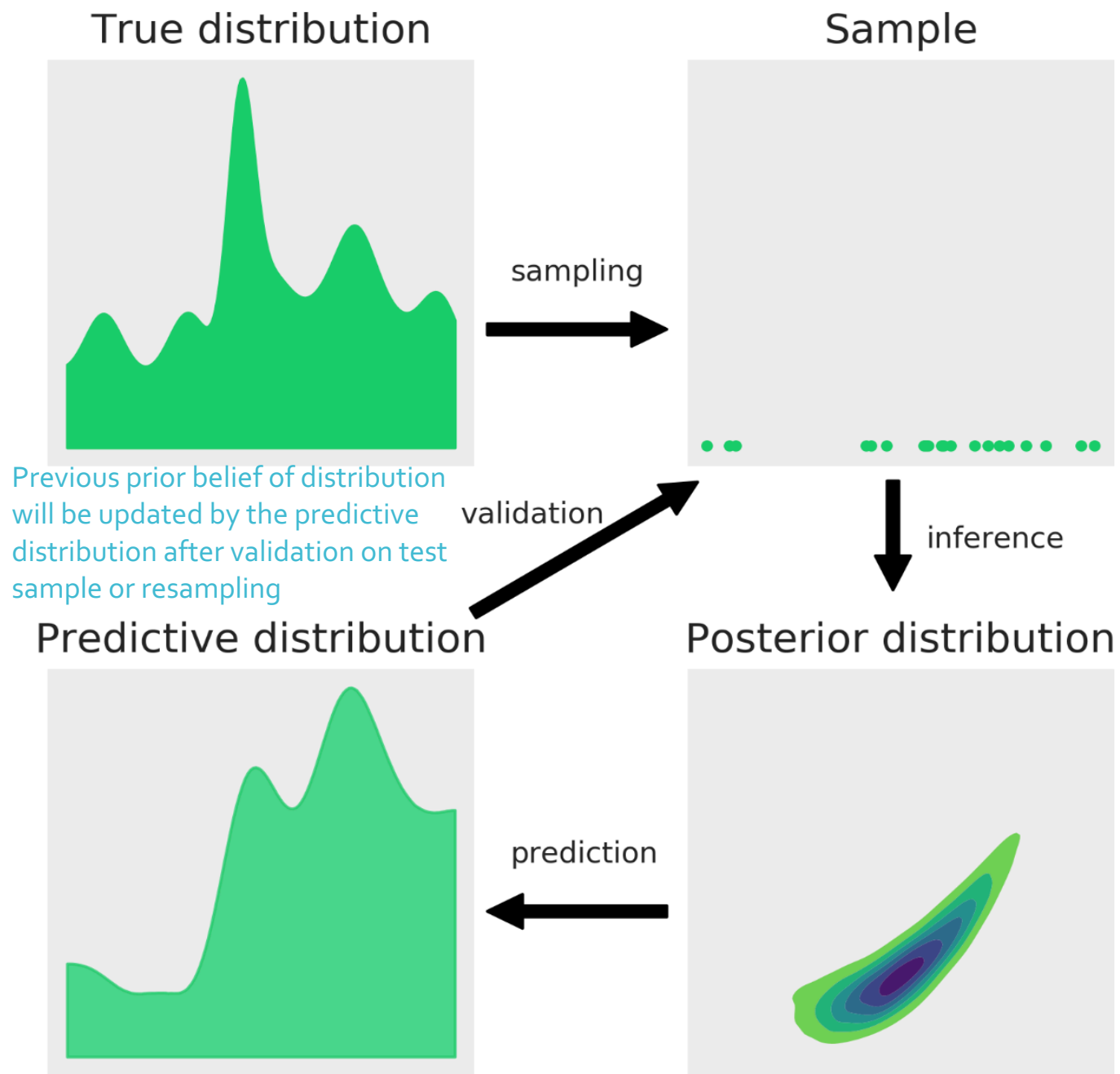
Python package for Bayesian statistical modelling

Markov Chain Monte Carlo and variation fitting algorithms

Relies on Theano for automatic differentiation

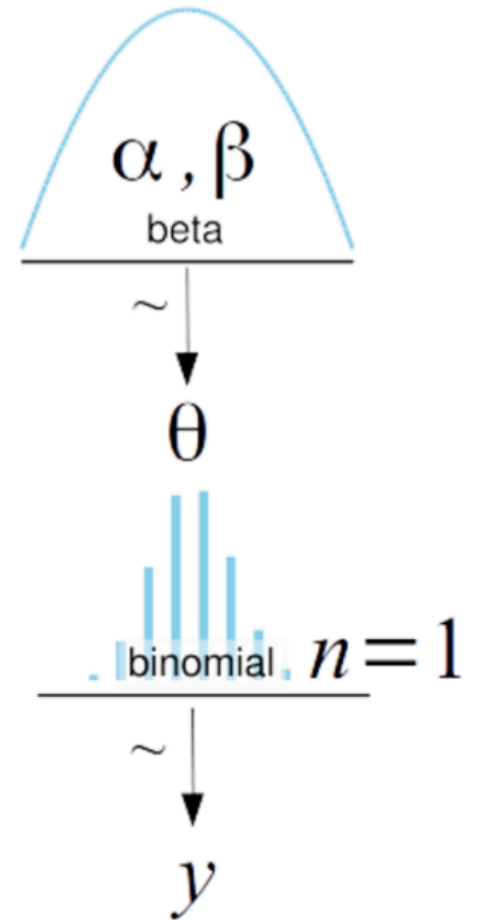
Used for data creation, model definition, model fitting, and posterior analysis

Bayesian Approach



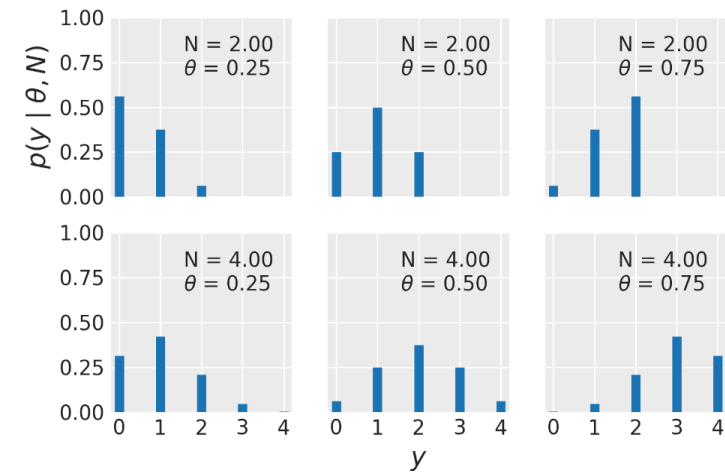
The experiment is the $\langle y, N \rangle$, the outcome is y_o {H,T}, how would you understand θ (fairness of the coin) with sample information.

- One does not know the exact θ , given a prior belief that is based on a beta distribution. α and β are hyperparameters, a representation of your belief of θ .
- Given the prior distribution of θ , one based on information to deduce the posterior distribution of θ
- Inference
 - A commonly-used device to summarize the spread of a posterior distribution is to use a **Highest-Posterior Density (HPD)** interval that for assessment
 - An HPD is the shortest interval containing a given portion of the probability density. One of the most commonly-used is the 95% HPD, often accompanied by the 50% HPD.
 - 95% HPD for some analysis is (.2, .5) mean that according to our data and model, we think the parameter in question is between 2 and 5 with a probability of 0.95.

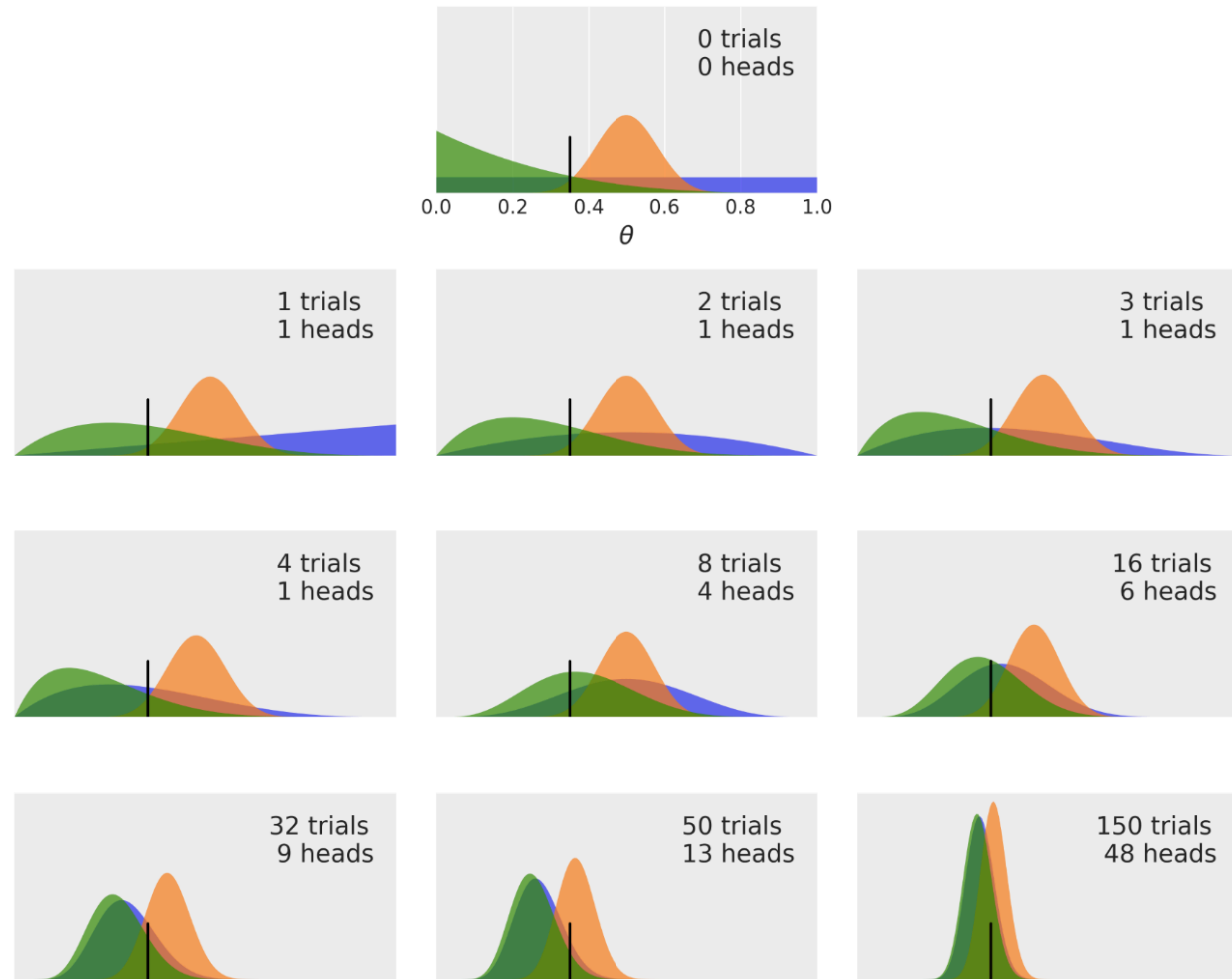


LAB2

Binomial distribution: N trial; y number of head, based on fairness of the coin



$$p(y | \theta, N) = \frac{N!}{y!(N-y)!} \theta^y (1-\theta)^{N-y}$$

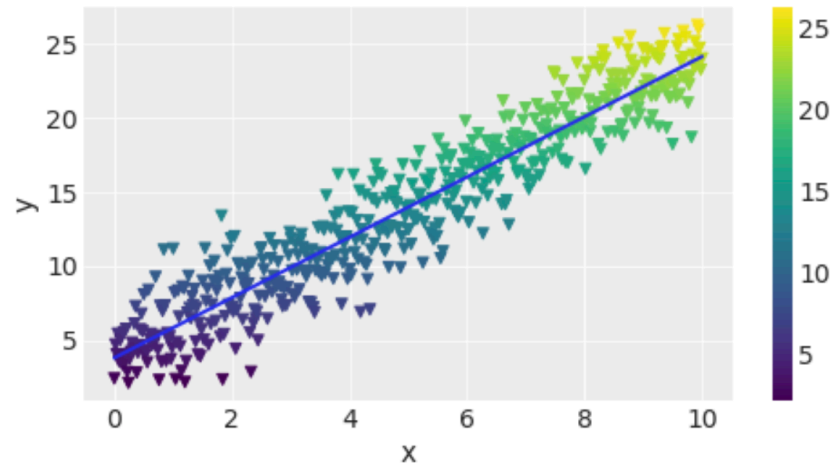


A prior distribution of the theta

$$p(\theta) = \frac{\Gamma(\alpha + \beta)}{\Gamma(\alpha)\Gamma(\beta)} \theta^{\alpha-1} (1-\theta)^{\beta-1}$$

LAB2: Bayesian Regression

Classical Regression



Bayesian Regression

