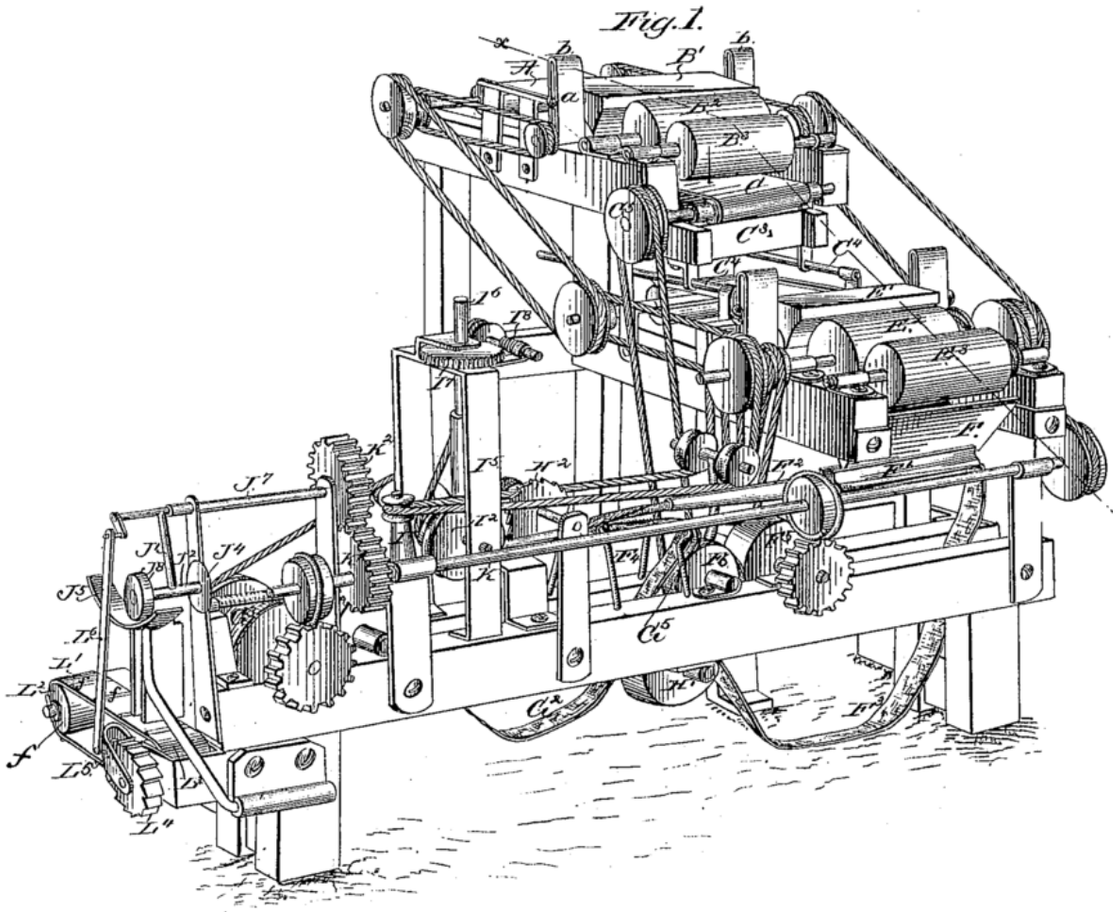
MACHINE LEARNING IN THE LSST ERA

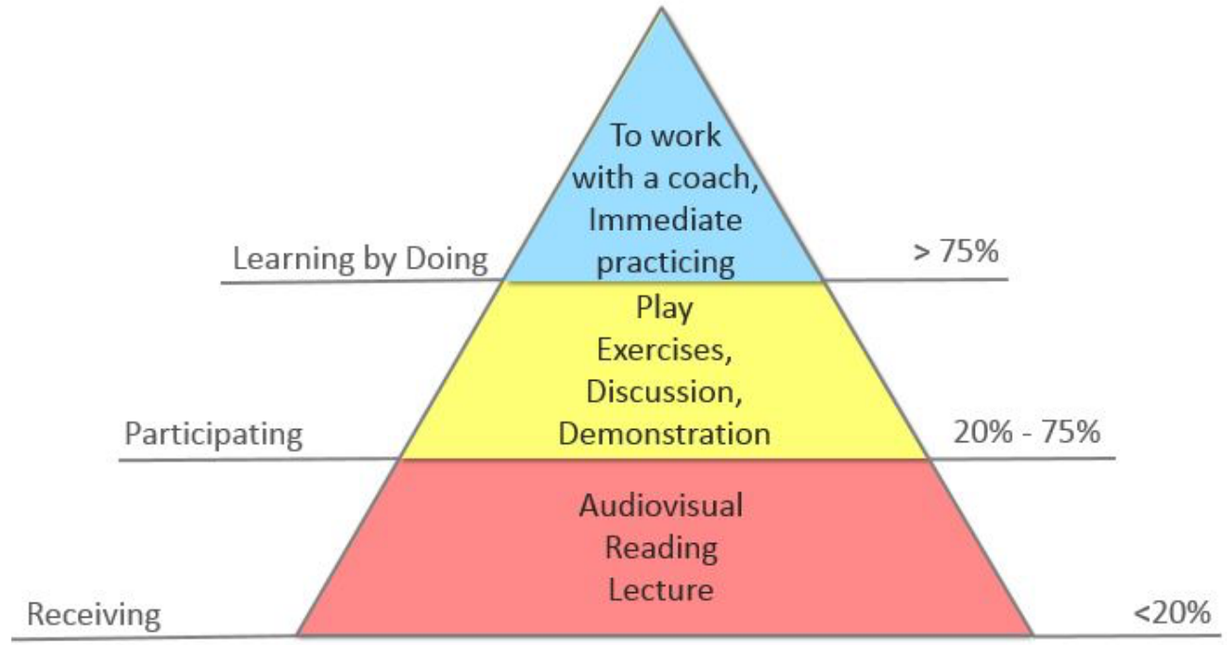
David Kirkby, UC Irvine

LSST Dark Energy School
SLAC, February 2017

MACHINE LEARNING

?

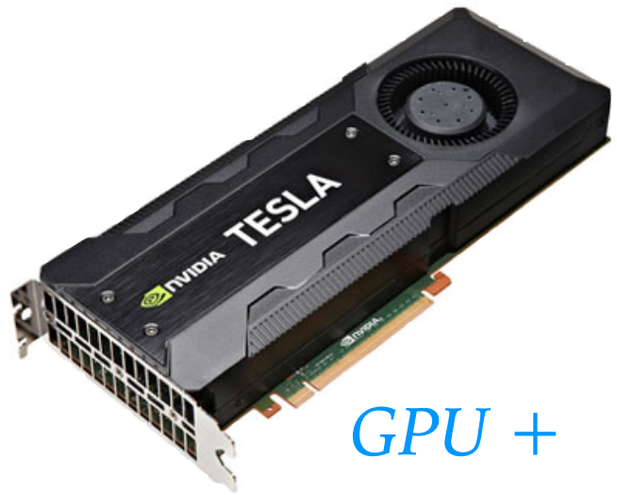




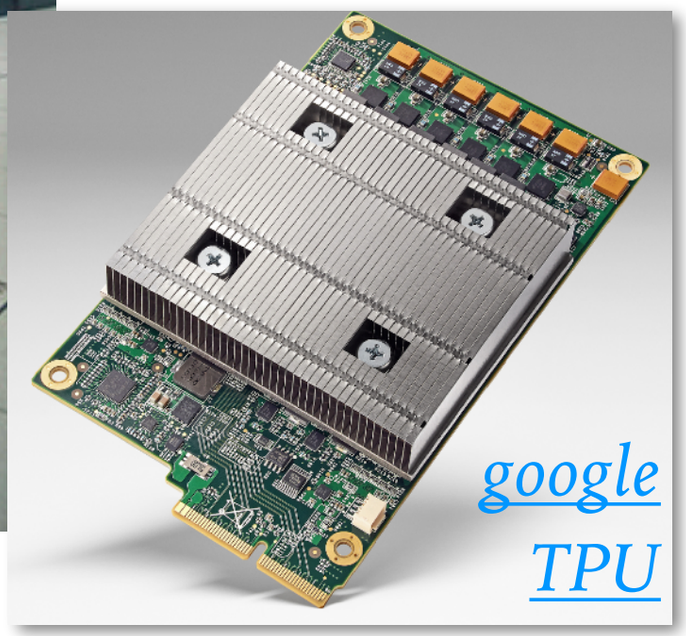
Retention of Learning

MACHINE LEARNING

```
    Returns
    -------
    centroid : float ndarray with shape (k, n_features)
        Centroids found at the last iteration of k-means.

    label : integer ndarray with shape (n_samples,)
        label[i] is the code or index of the centroid the
        i'th observation is closest to.

    inertia : float
        The final value of the inertia criterion (sum of squared distances to
        the closest centroid for all observations in the training set).

    best_n_iter: int
        Number of iterations corresponding to the best results.
        Returned only if `return_n_iter` is set to True.

    """
    if n_init <= 0:
        raise ValueError("Invalid number of initializations."
                         " n_init=%d must be bigger than zero." % n_init)
    random_state = check_random_state(random_state)

    if max_iter <= 0:
        raise ValueError('Number of iterations should be a positive number,'
                         ' got %d instead' % max_iter)

    best_inertia = np.infty
    X = as_float_array(X, copy=copy_x)
    tol = _tolerance(X, tol)

    # If the distances are precomputed every job will create a matrix of shape
    # (n_clusters, n_samples). To stop KMeans from eating up memory we only
    # activate this if the created matrix is guaranteed to be under 100MB. 12
    # million entries consume a little under 100MB if they are of type double.
    if precompute_distances == 'auto':
        n_samples = X.shape[0]
        precompute_distances = (n_clusters * n_samples) < 12e6
    elif isinstance(precompute_distances, bool):
        pass
    else:
        raise ValueError("precompute_distances should be 'auto' or True/False"
                         ", but a value of %r was passed" %
                         precompute_distances)

    # subtract of mean of x for more accurate distance computations
    if not sp.issparse(X) or hasattr(init, '__array__'):
        X_mean = X.mean(axis=0)
    if not sp.issparse(X):
        # The copy was already done above
        X -= X_mean

    if hasattr(init, '__array__'):
        init = check_array(init, dtype=X.dtype.type, copy=True)
        _validate_center_shape(X, n_clusters, init)

        init -= X_mean
        if n_init != 1:
            warnings.warn(
                'Explicit initial center position passed: '
                'performing only one init in k-means instead of n_init=%d'
                % n_init, RuntimeWarning, stacklevel=2)
            n_init = 1

    # precompute squared norms of data points
    x_squared_norms = row_norms(X, squared=True)

    best_labels, best_inertia, best_centers = None, None, None
    if n_clusters == 1:
        # elkan doesn't make sense for a single cluster, full will produce
        # the right result.
        algorithm = "full"
```



GPU + cuda/opencl



google TPU

MACHINE LEARNING

e.g.

Suggest a missing word in a sentence.

Identify a specific person in a photo.

Drive a car automatically.

Predict the sky brightness for tomorrow night's observing.

Estimate a galaxy's redshift from its LSST magnitudes.

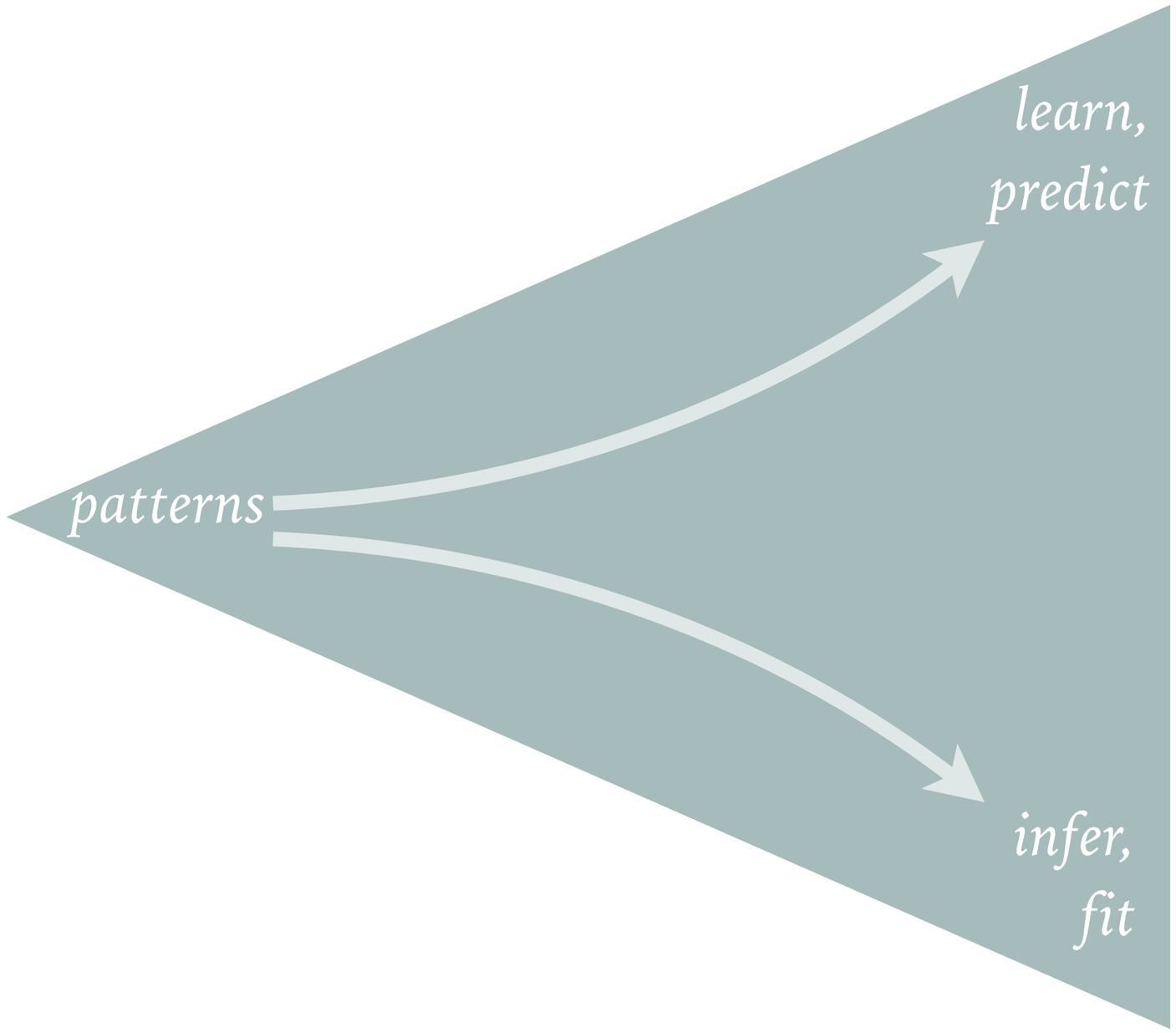
Describe the relationship between supernovae distance and redshift.

ACTIVITY: DEFINE YOUR TERMS

- What is the relationship between machine learning and statistics?
- What is the difference between a data scientist and a data engineer?
- What is "deep" about deep learning?
- Does your research focus more on data or models?

MODELS

Generalization



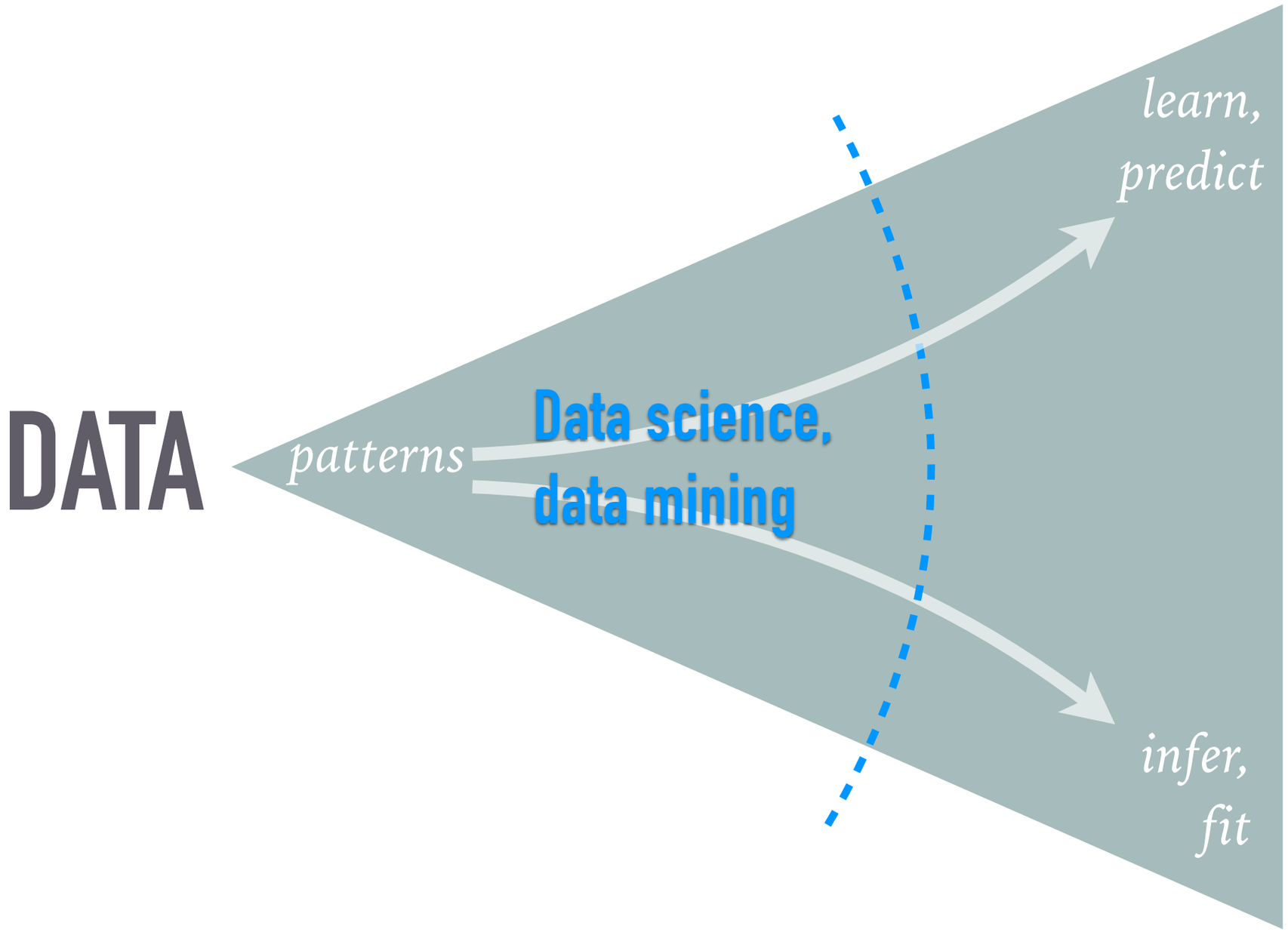
DATA

Quantification

PARAMETERS, ERRORS

MODELS

Generalization



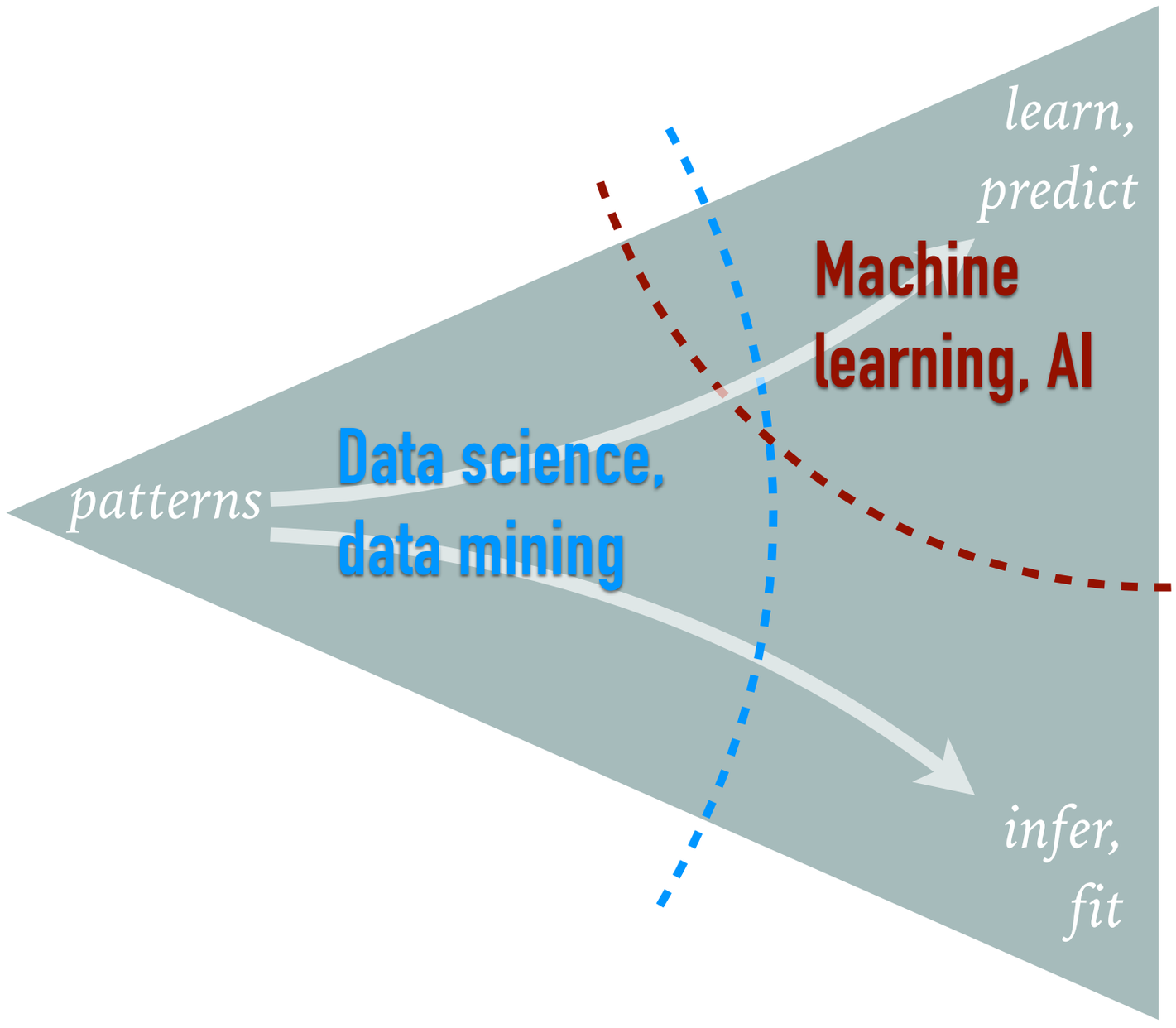
DATA

Quantification

PARAMETERS, ERRORS

MODELS

Generalization



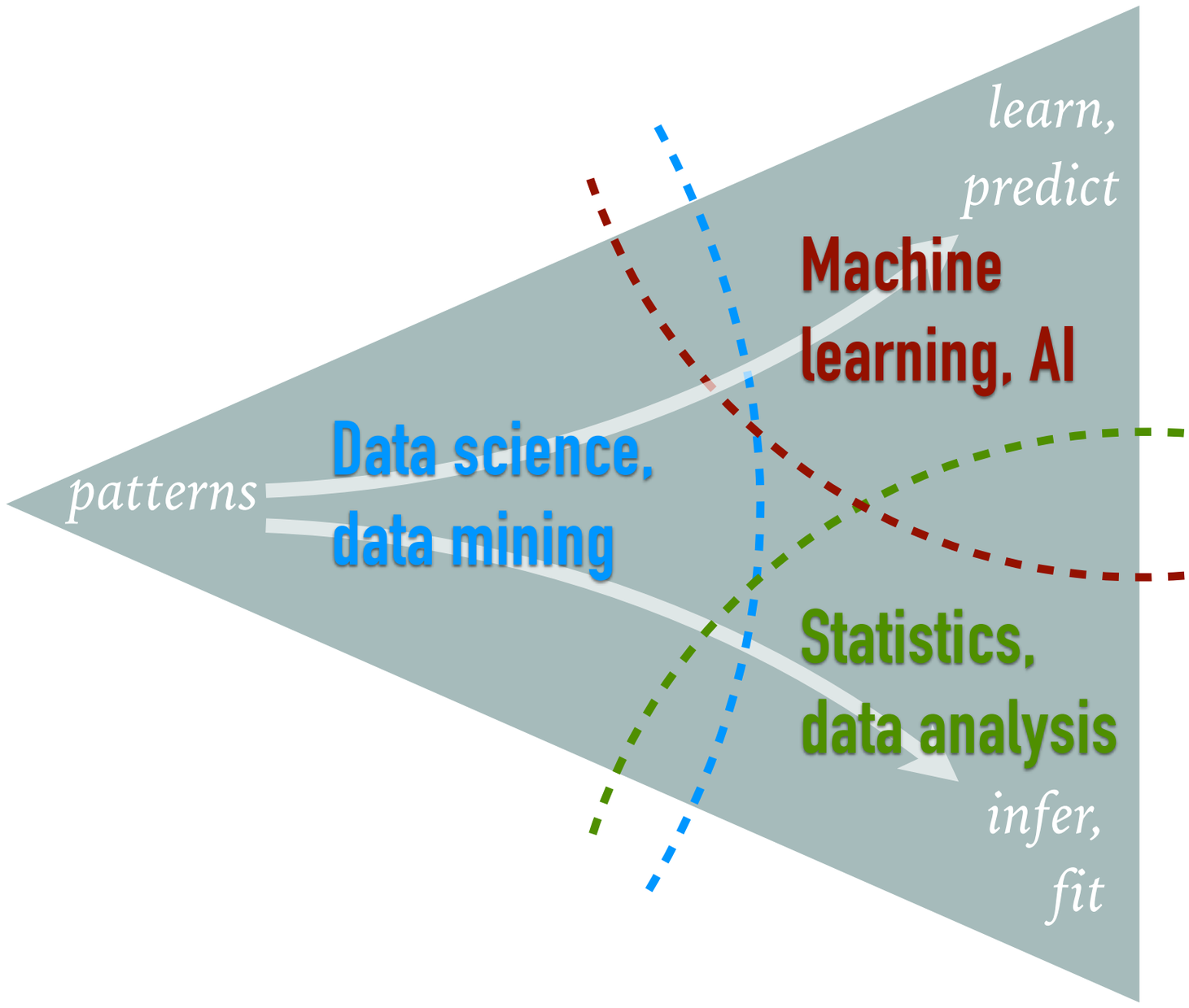
DATA

Quantification

PARAMETERS, ERRORS

MODELS

Generalization



DATA

patterns

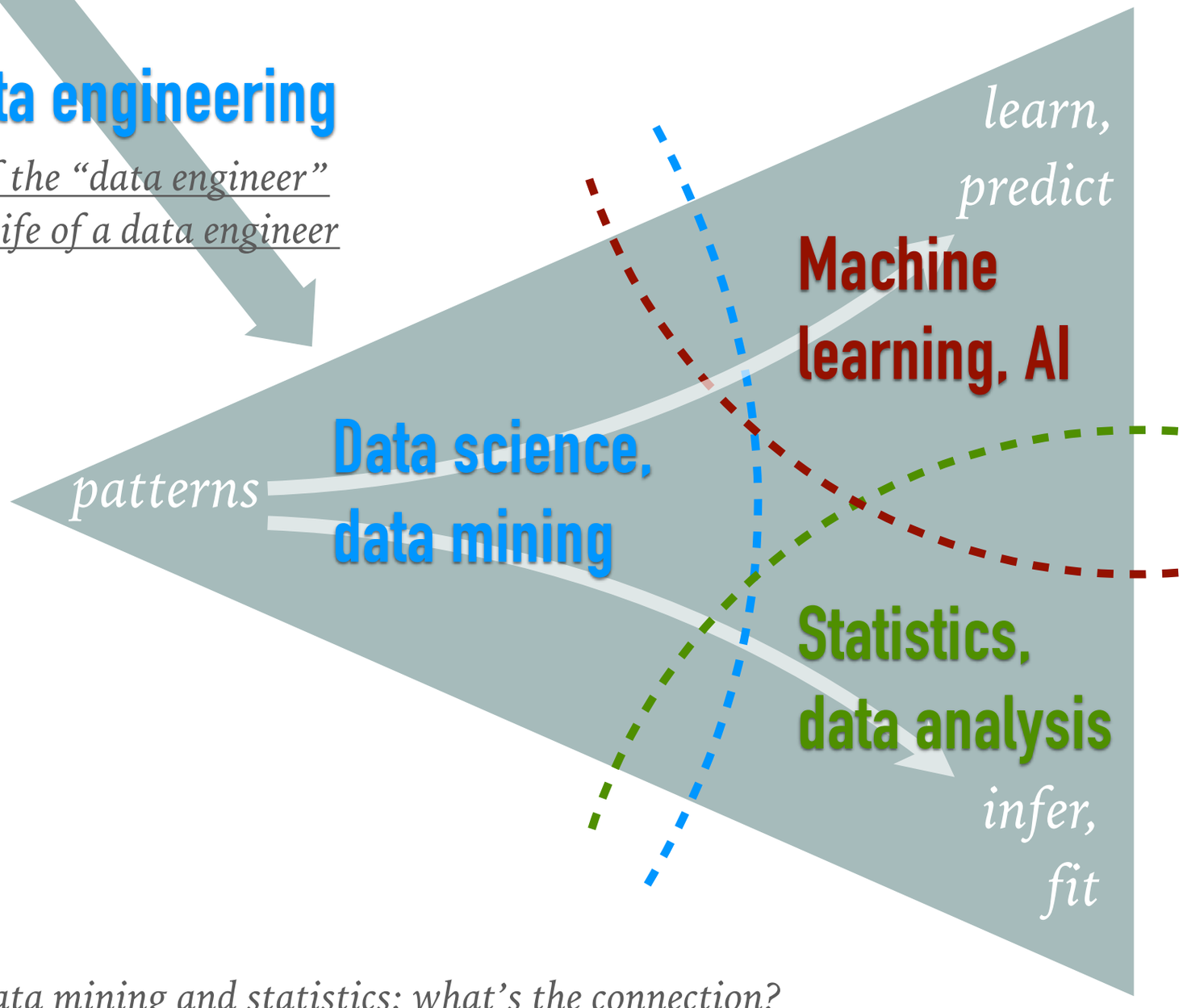
Quantification

PARAMETERS, ERRORS

Software engineering

Data engineering

The rise of the "data engineer"
A day in the life of a data engineer



DATA

MODELS

Generalization

Quantification

PARAMETERS,
ERRORS

Data mining and statistics: what's the connection?

Glossary

Machine learning	Statistics
network, graphs	model
weights	parameters
learning	fitting
generalization	test set performance
supervised learning	regression/classification
unsupervised learning	density estimation, clustering
large grant = \$1,000,000	large grant= \$50,000
nice place to have a meeting: Snowbird, Utah, French Alps	nice place to have a meeting: Las Vegas in August
python	*R*
conference talk	*journal article*

http://statweb.stanford.edu/~tibs/stat315a/glossary.pdf

MACHINE LEARNING

=

DATA + MODELS

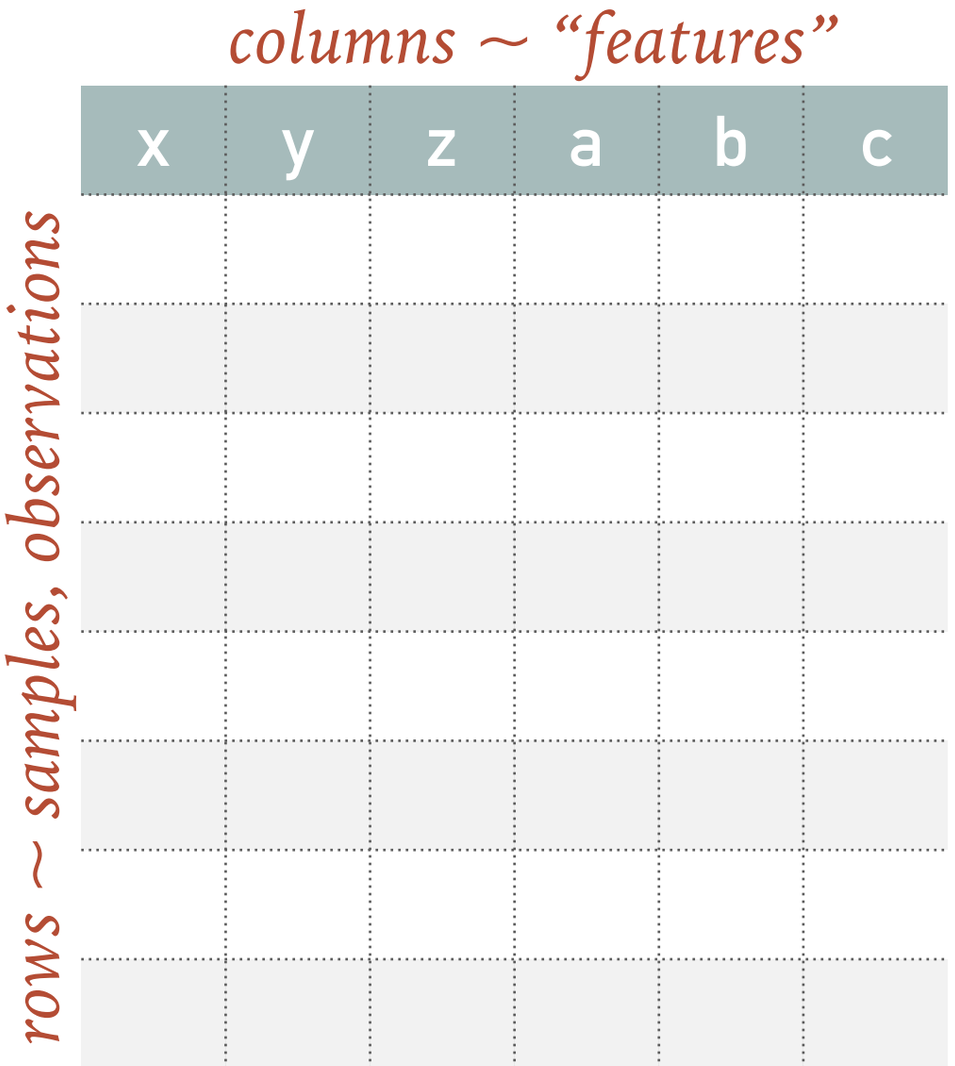
ROADMAP

- ✓ Introduction
- ➤ Data & models
- ➤ Statistical context of ML
- ➤ Types of learning, problems, solutions
- ➤ The bleeding edge of ML

DATA + MODELS

Data are a finite set of measurements:

- e.g., spreadsheet, FITS table, …
- features: numeric / categorical?
- samples: ordered? independent? identically distributed? (“i.i.d.”)
- measurement errors?
- binned / un-binned?
- similarity measure on samples?



ACTIVITY: DESCRIBE YOUR DATA

- Pick one of these machine learning problems:
  - *Predict the sky brightness for tomorrow night's observing.*
  - *Estimate a galaxy's redshift from its LSST magnitudes.*
- Describe the rows (samples) and columns (features) of the data you might use to solve this problem.

DATA + MODELS

Models specify the probabilities of possible measurements:

- explicit: probability density function.
- implicit: algorithm to generate random outcomes (forward / generative model).

- usually wrong (except “Toy MC”)
- observables (latent variables):
  - integrability: required to calculate normalized probabilities.
- parameters (hyper-parameters):
  - differentiability: required to find most probable (uphill) direction.
- variance - bias tradeoffs (regularization).

WHAT IS SPECIAL ABOUT ML IN ASTRONOMY?

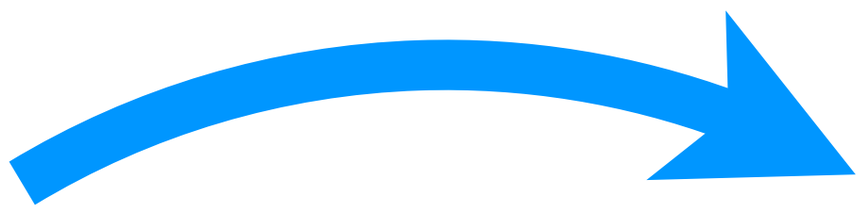
- We are data producers, not data consumers:
  - Experiment / survey design.
  - Optimization of statistical errors.
  - Control of systematic errors.
- Our data measures physical processes.
  - Measurements often reduce to counting photons, etc, with known a-priori errors.
  - Dimensions & units.

WHAT IS SPECIAL ABOUT ML IN ASTRONOMY?

- Our models are usually traceable to an underlying physical theory:
  - Models constrained by theory and previous observations.
  - Parameter values often intrinsically interesting.
- A parameter error estimate is just as important as its value:
  - Prefer methods that handle input data errors (weights) and provide output parameter error estimates.

(ΛCDM is not a "model" in the ML sense unless augmented with assumptions about measurement errors, e.g., via a χ^2 function.)

A model is trained on, fit to, or inferred from data.



DATA + MODELS

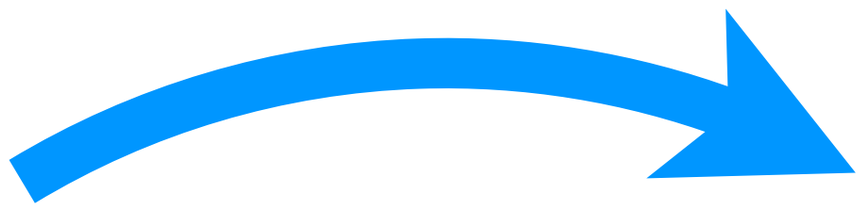
Data is a realization of some model

(but probably not the one you are using).

HOW TO BUILD A MODEL?

Exploratory data analysis & visualization:

- single feature: histogram of PDF, CDF.
- two features: scatter plot.
- multiple features:
  - pair-wise corner plot.
  - local embedding (tSNE, …).



DATA + MODELS

THE LANGUAGE OF ML IS STATISTICS
(NOT PYTHON!)

- Key ideas:
  - Bayesian reasoning.
  - Model naturalness (Occam's razor).

- Given some data:
  - Infer probabilities assuming a model.
  - Compare alternative models.

ACTIVITY: BAYESIAN REASONING

English or not english?

➤ Write down your best guess: YES/NO.

ACTIVITY: BAYESIAN REASONING

English or not english?

- Write down your best guess: YES/NO.
- Think about the probability that the answer is YES. Write down a number.

ACTIVITY: BAYESIAN REASONING

English or not english?

- Write down your best guess: YES/NO.
- Think about the probability that the answer is YES. Write down a number.
- Discuss your reasoning with your neighbor, then update your answer.

ACTIVITY: BAYESIAN REASONING

- What do we know?
  - DATA = "wearing an ENGLAND t-shirt"
- What question are we asking?
  - P(english | DATA) ?
- What do we need to assume?
  - MODEL =

```
if english:
   prob[DATA] = 1/3
else:
   prob[DATA] = 1/5
```

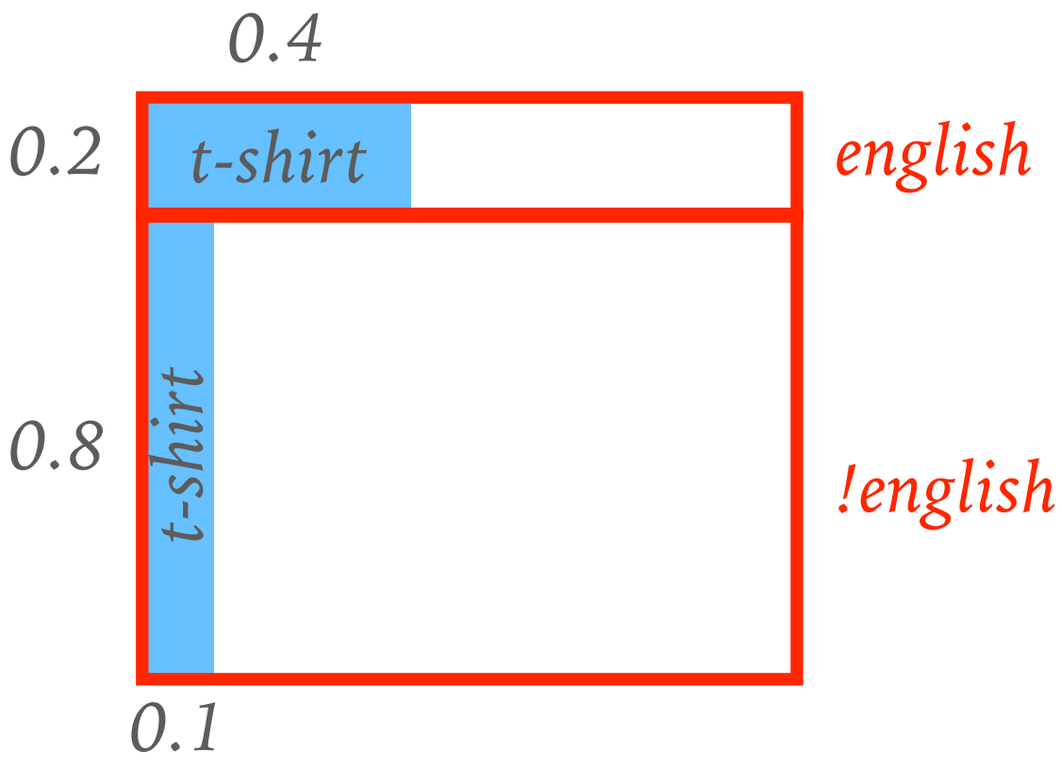
  - PRIOR = "20% of astronomers are english"

ACTIVITY: BAYESIAN REASONING

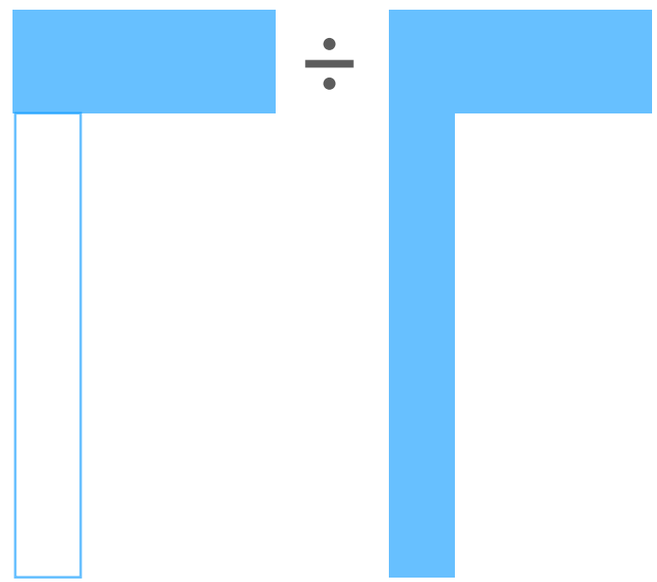
- PRIOR = "20% of astronomers are english"
- MODEL =

```
if english:
  prob[DATA] = 0.4
else:
  prob[DATA] = 0.1
```

- DATA = "wearing an ENGLAND tshirt"



P(english | DATA) =



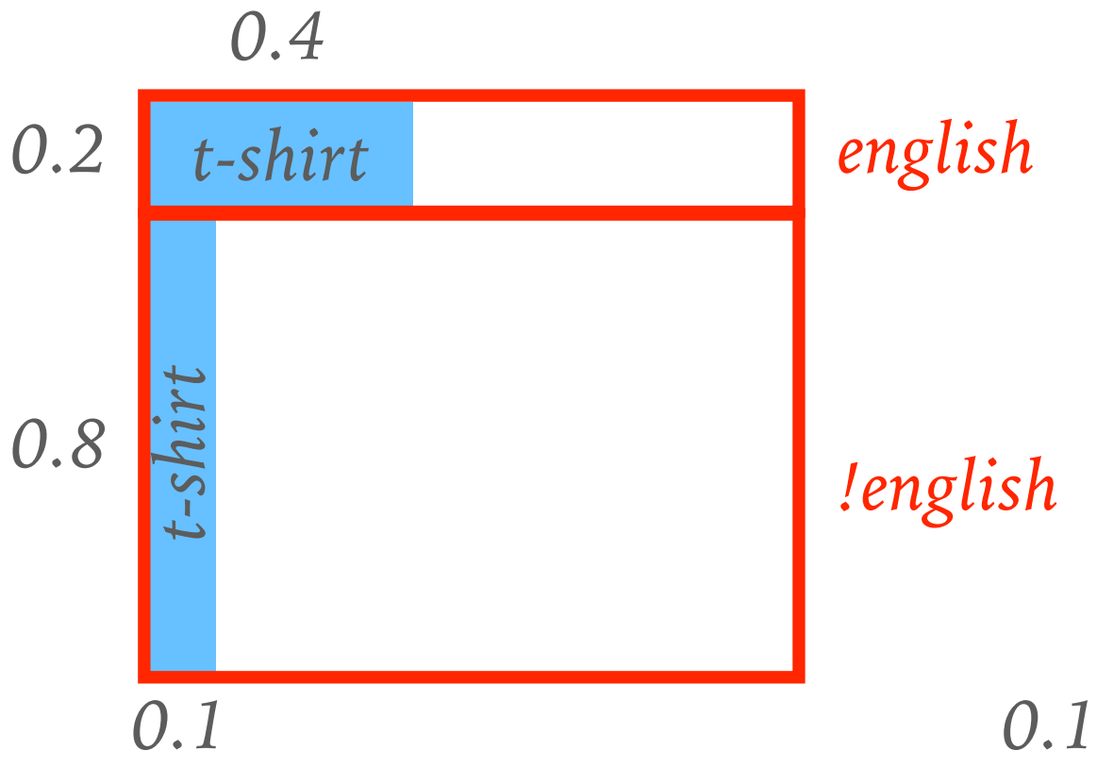
ACTIVITY: BAYESIAN REASONING

$$P(\text{english} \mid \text{tshirt}) = \frac{P(\text{tshirt} \mid \text{english})\, P(\text{english})}{P(\text{tshirt})}$$

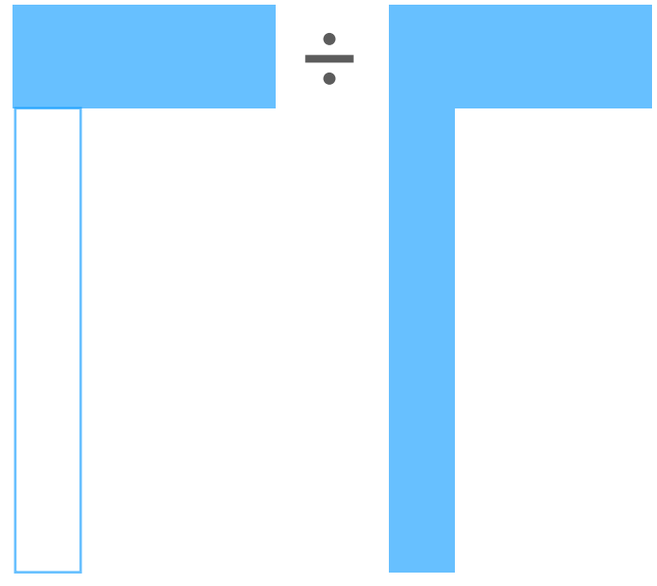
$P(\text{english} \mid \text{tshirt})$: 0.5 — $P(\text{tshirt} \mid \text{english})$: 0.4 — $P(\text{english})$: 0.2 — $P(\text{tshirt})$: 0.16

$$P(\text{tshirt}) = P(\text{tshirt} \mid \text{english})\, P(\text{english}) + P(\text{tshirt} \mid \text{!english})\, P(\text{!english})$$

0.16 — 0.4 — 0.2 — 0.1 — 0.8



P(english | DATA) =



BAYES' THEOREM

http://setosa.io/ev/conditional-probability/

- The theorem has two ingredients:
  - The definition of conditional probability for outcomes.
  - Unified treatment of observables (data) and parameters (model) as outcomes.

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

P(A) = 0.200 or 20.0%

P(B) = 0.200 or 20.0%

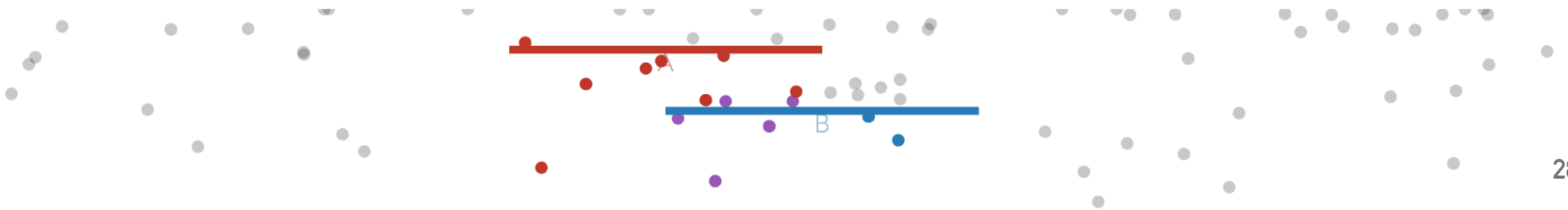
P(A n B) = 0.100 or 10.0%

P(B|A) = 0.500 or 50.0%

If we have a ball and we know it hit the red shelf, there's a 50.0% chance it also hit the blue shelf.

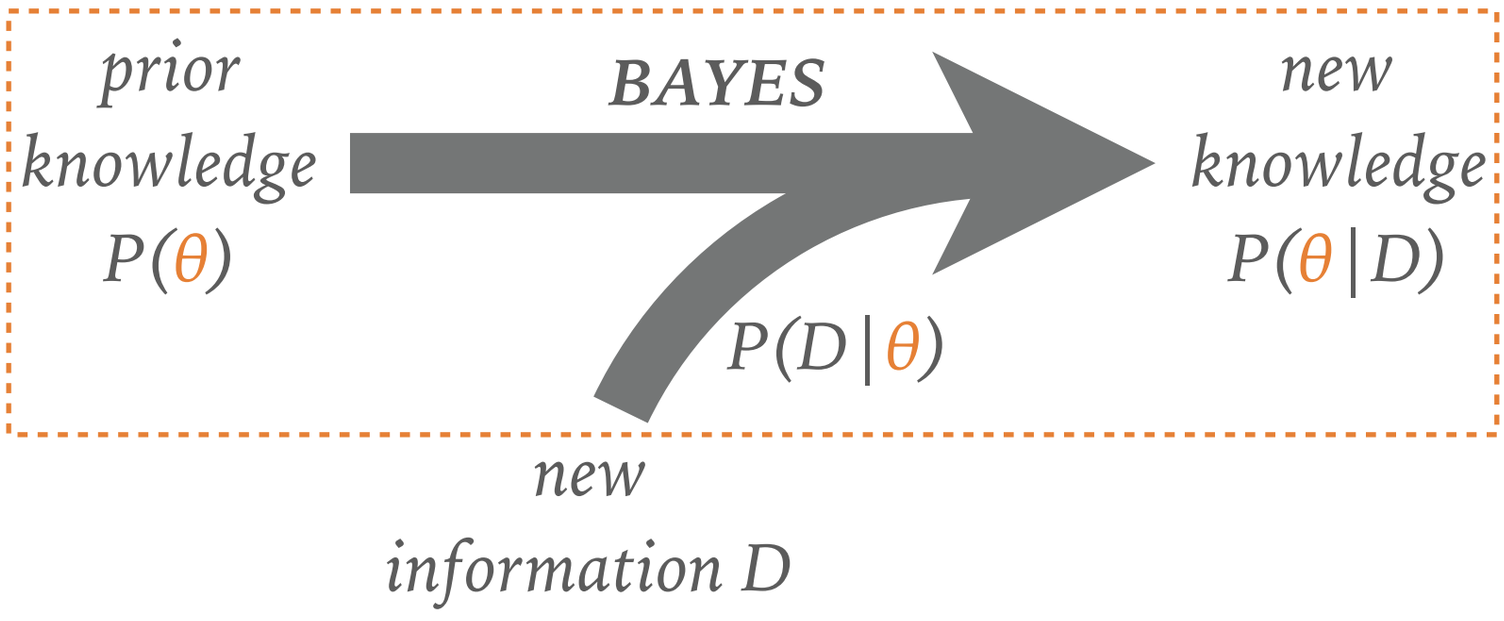
P(A|B) = 0.500 or 50.0%

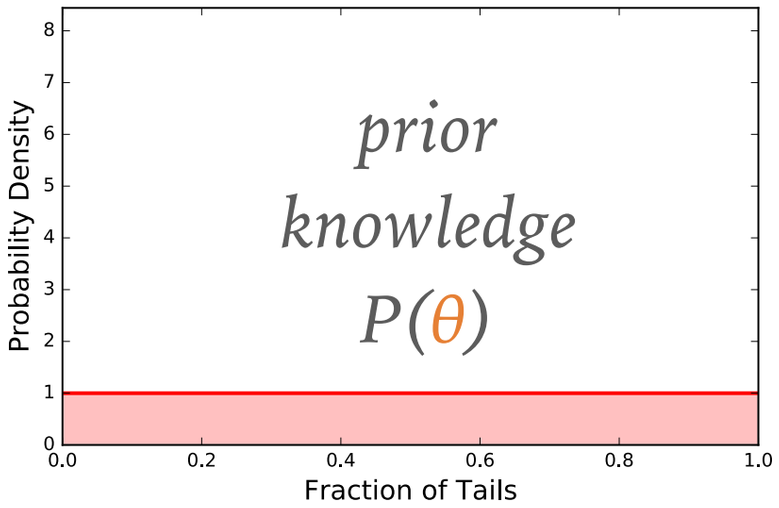
If we have a ball and we know it hit the blue shelf, there's a 50.0% chance it also hit the red shelf.



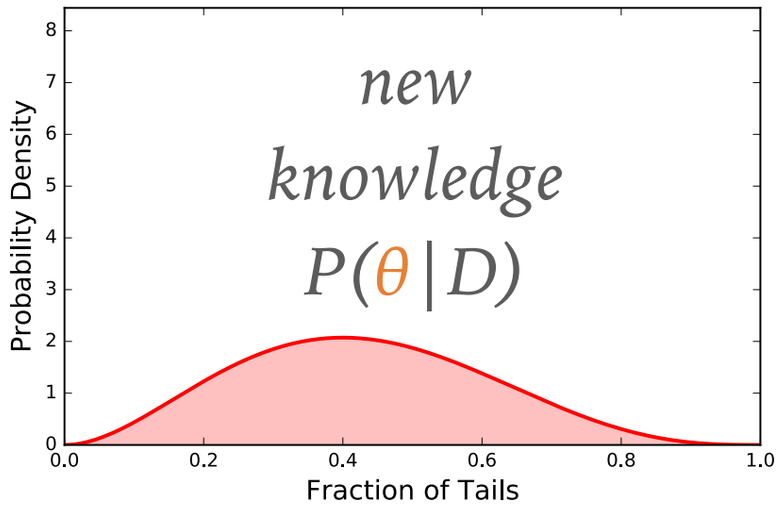
HOW DO WE USE BAYES' THEOREM?

- To update our knowledge based on new information.
  - Must specify (1) a model and (2) priors!

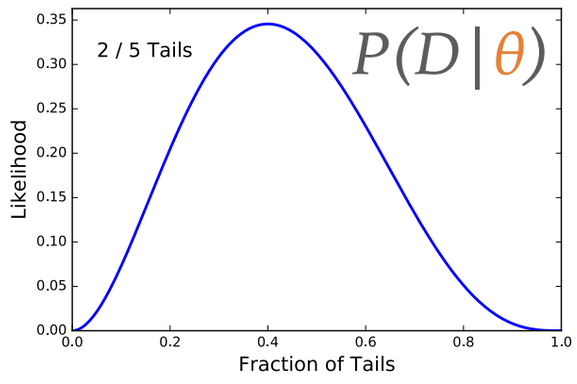




BAYES

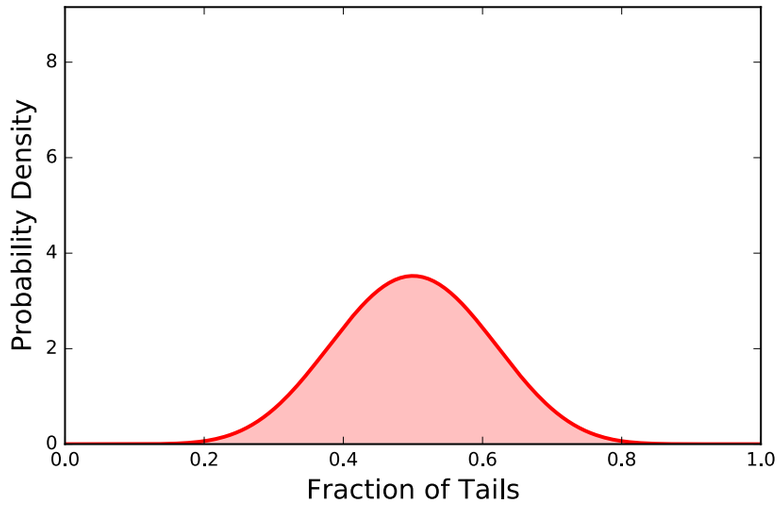


prior dependent = Bayesian

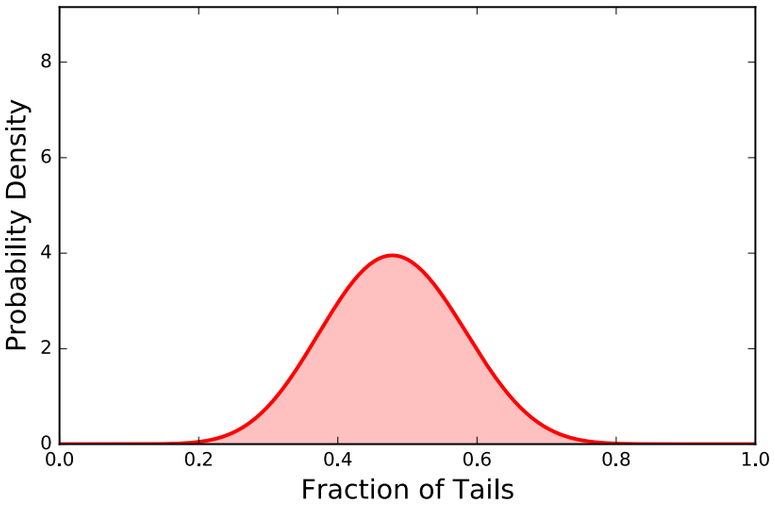


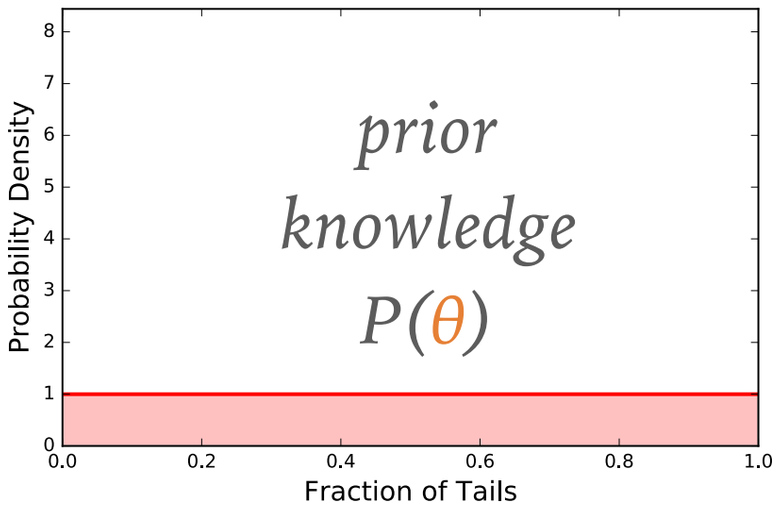
model dependent

D = *observed 2 tails from 5 coin tosses*

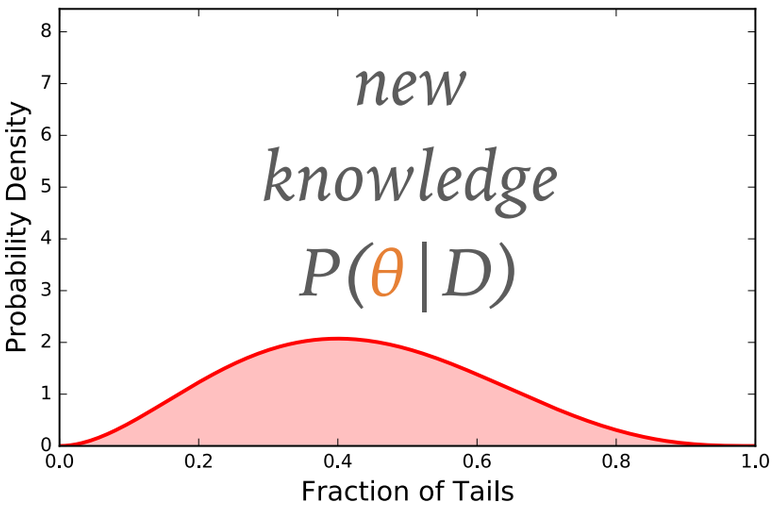


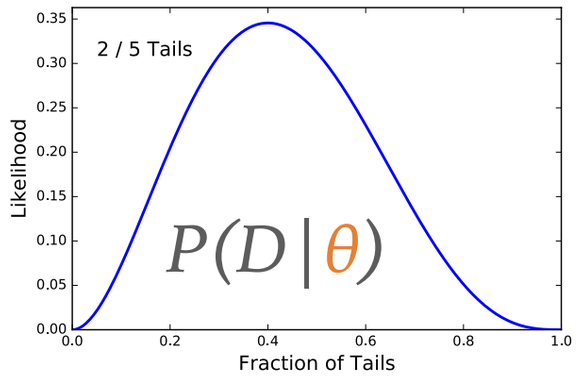
Posterior will depend strongly on prior with insufficient data!



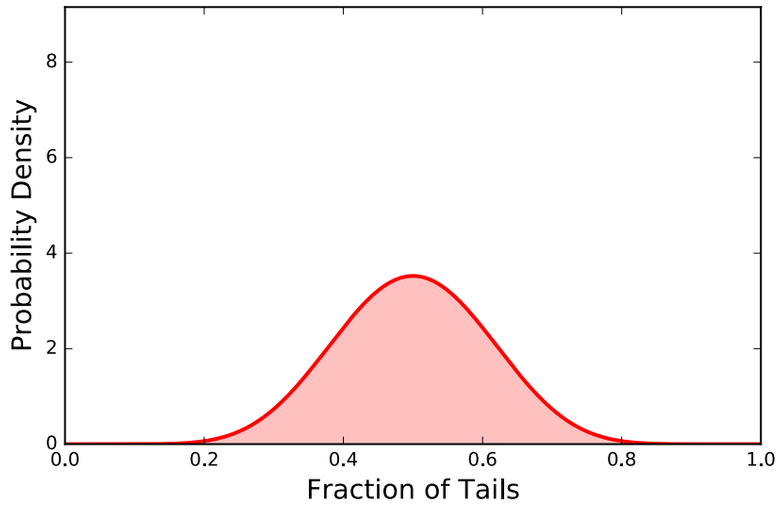


BAYES

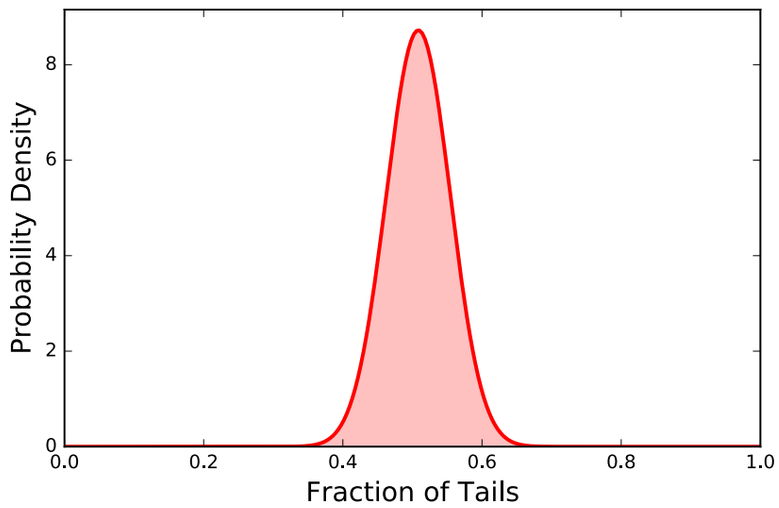


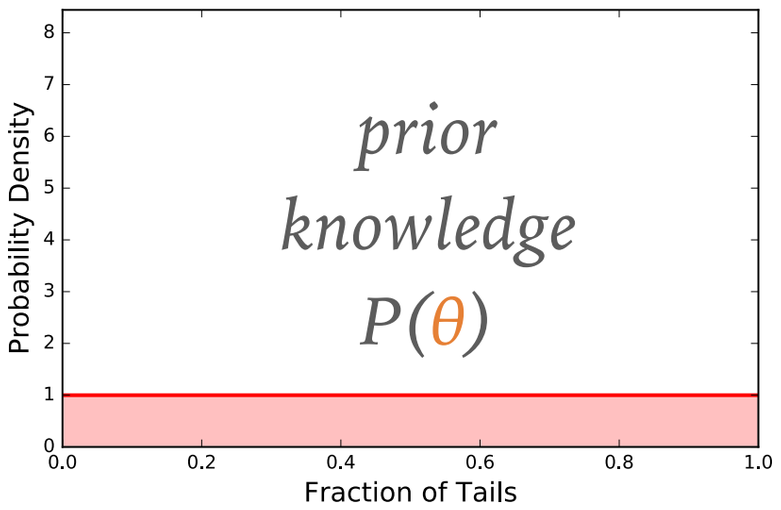


D = *observed 2 tails from 5 coin tosses*

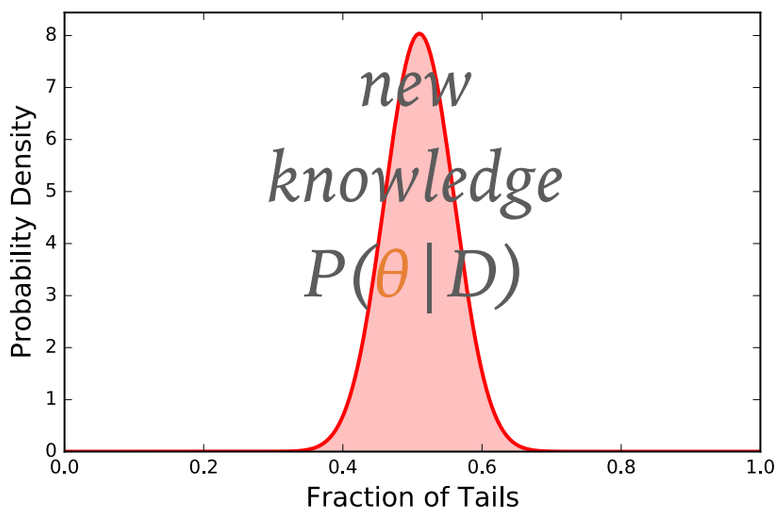


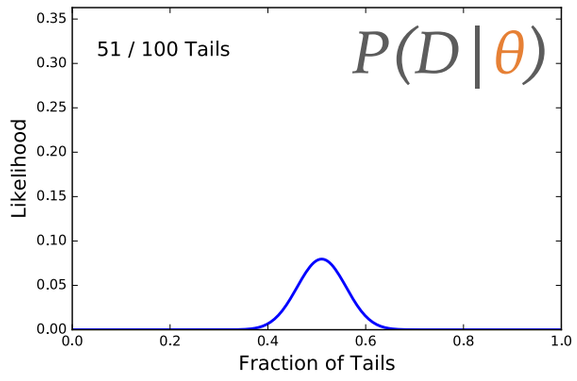
Add more/better data for robust inference.





BAYES

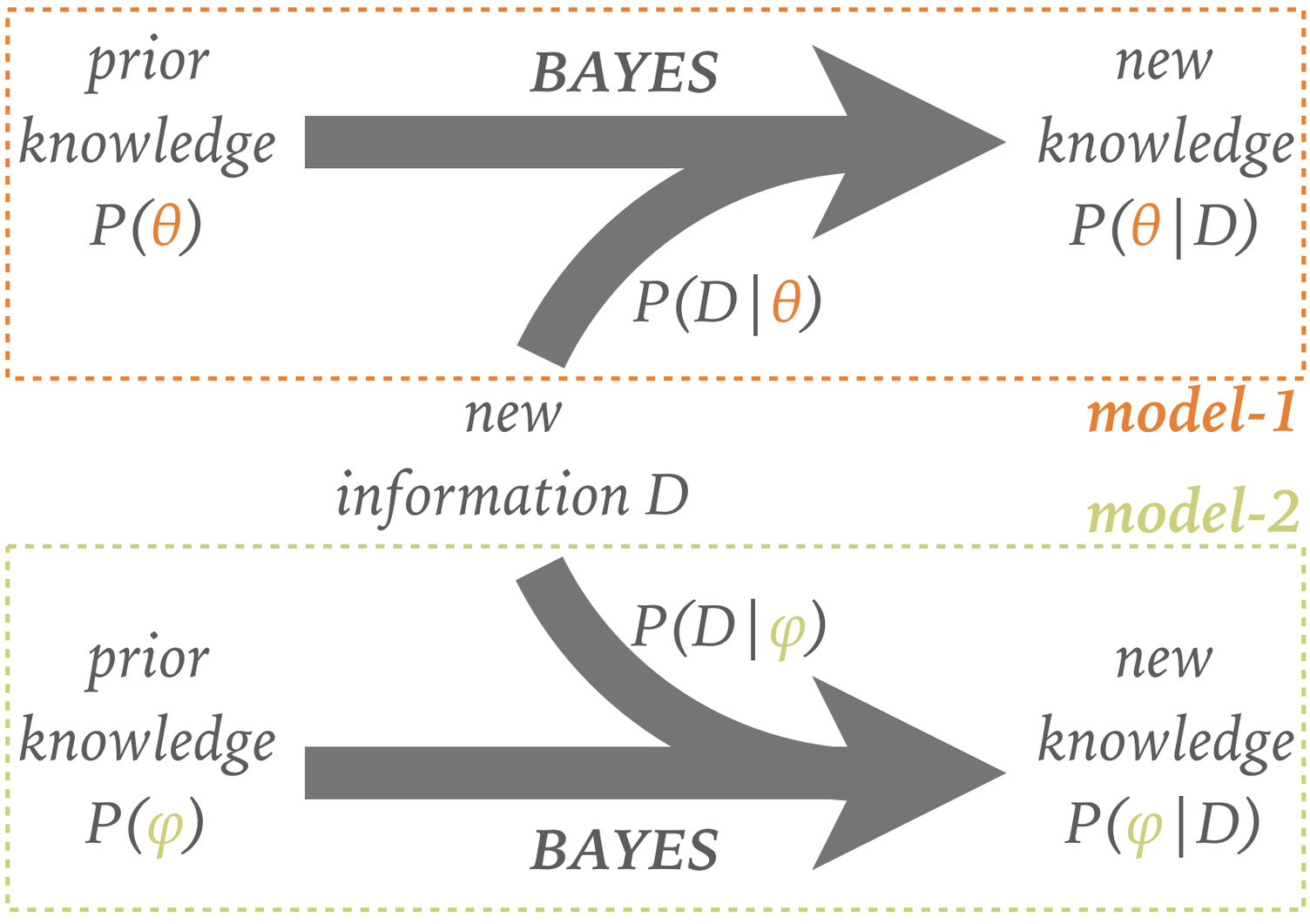




D = observed 51 tails from 100 coin tosses

HOW DO WE USE BAYES' THEOREM?

- To update our knowledge based on new information.
- To compare alternative models that explain the same data.



BAYESIAN MODEL COMPARISON

- We normally use Bayes' rule for the (posterior) probability of data D given specified parameters θ and model M:

$$P(\theta \mid D, M) = \frac{P(D \mid \theta, M)\, P(\theta, M)}{P(D, M)}$$

- In order to turn this into a statement about the model without specifying the parameters, we need to marginalize (integrate) them out:

$$P(M \mid D) = \frac{P(D \mid M)\, P(M)}{P(D)}$$

(I am skipping many lines of probability calculus here)

BAYESIAN MODEL COMPARISON

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

- The denominator P(D) can only be evaluated if you can fully specify all possible models!
- Generally cannot make statements about the absolute (posterior) probability of a single model.
- However, P(D) cancels in probability ratios:

$$\frac{P(M_1|D)}{P(M_2|D)} = \frac{P(D|M_1)\,P(M_1)}{P(D|M_2)\,P(M_2)}$$

Model priors

Odds ratio *Bayes factor*

BAYESIAN MODEL COMPARISON

- How is the "naturalness" of a model taken into account?
  - Model priors.
  - Occam factor.

$$\frac{P(M_1|D)}{P(M_2|D)} = \frac{P(D|M_1)\,P(M_1)}{P(D|M_2)\,P(M_2)}$$

Model priors

Bayes factor

$$\frac{P(D|M_1)}{P(D|M_2)} \propto \frac{(\textit{fraction of } M_1 \textit{ param. space favored by } D)}{(\textit{fraction of } M_2 \textit{ param. space favored by } D)}$$

Bayes factor — *Occam factor*

OCCAM FACTOR

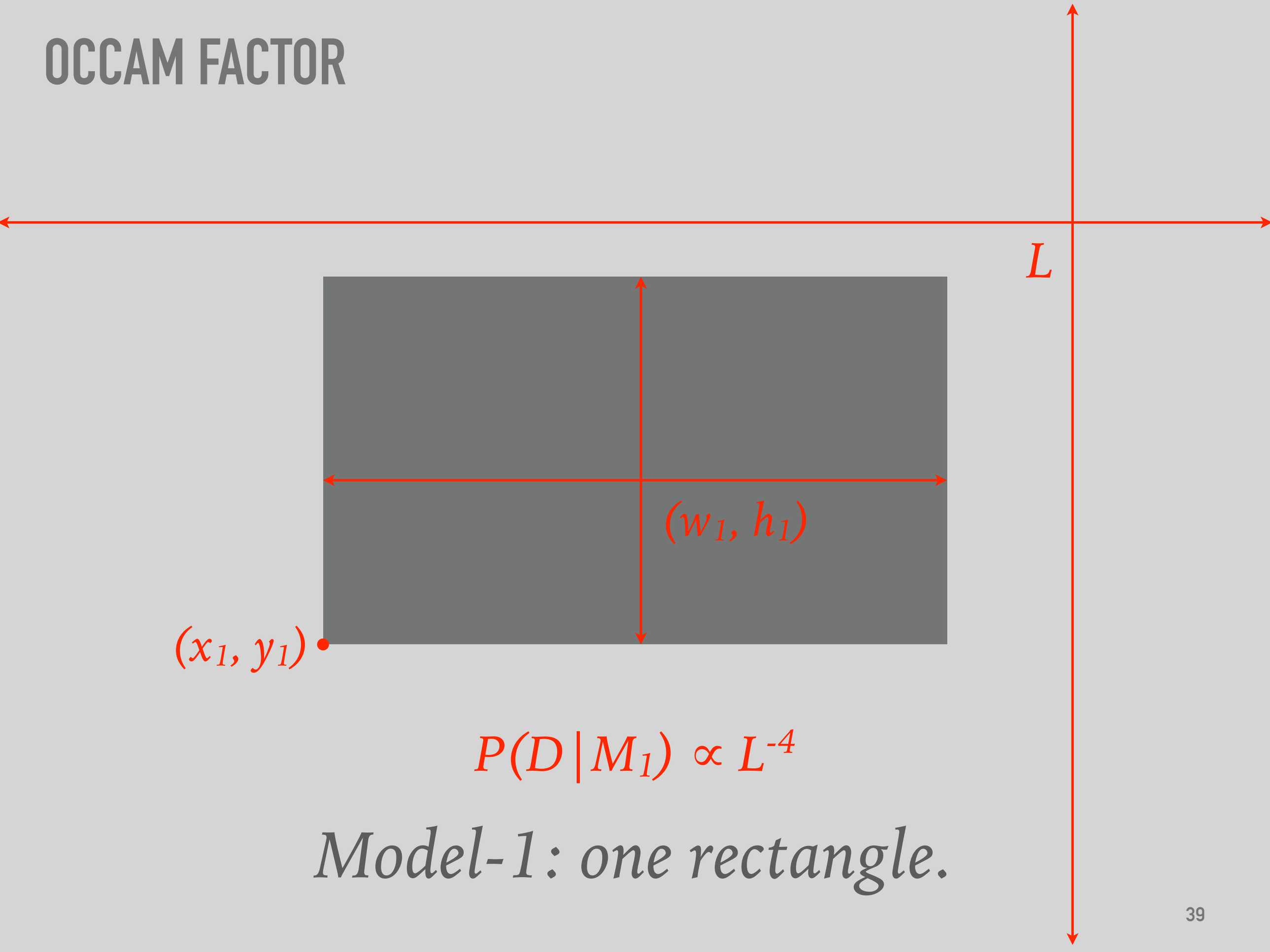
How many large galaxies?

OCCAM FACTOR

- *Is it possible there are two rectangles?*
- *Why are two rectangles an unnatural model?*

How many rectangles?

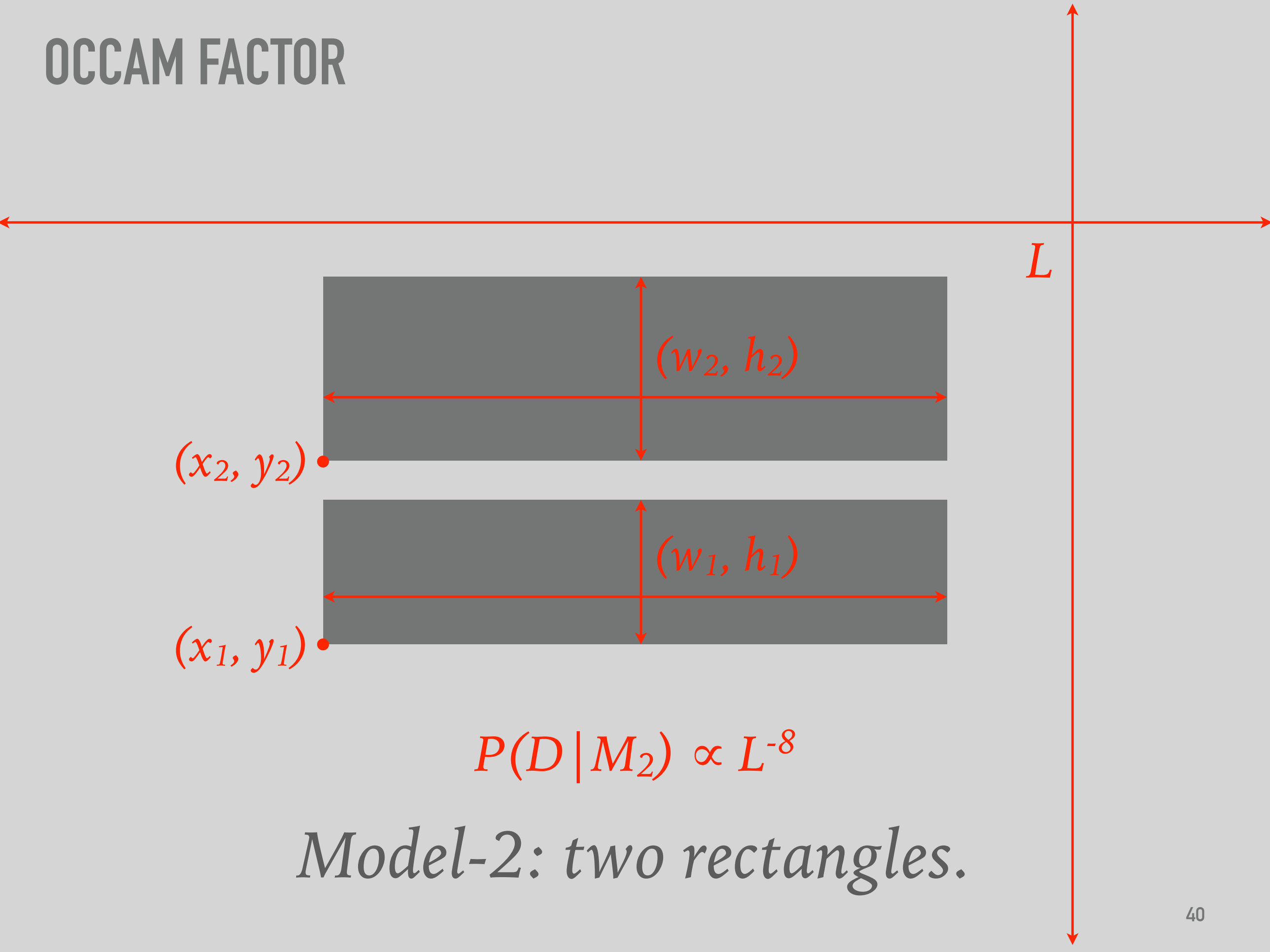
OCCAM FACTOR



$$P(D|M_1) \propto L^{-4}$$

Model-1: one rectangle.

OCCAM FACTOR



$$P(D|M_2) \propto L^{-8}$$

Model-2: two rectangles.

OCCAM FACTOR

- Two rectangles are possible but extremely unlikely, even if we believe that one vs two rectangles are equally likely a-priori!

$$P(D|M_2) \,/\, P(D|M_1) \propto L^{-4} \ll 1$$ *Occam factor*

How many rectangles?

TYPES OF LEARNING

- Supervised
- Un-supervised
- Reinforcement
  - Currently hot topic in ML community
  - Video games, GO (pong example)
  - LSST observing strategy, cadence?

Unsupervised

x	y	z	a	b	c

Reinforcement

x	y	z	a	b	c

✔ ✔ ✔

Supervised

x	y	z	a	b	c

train

				?	

test

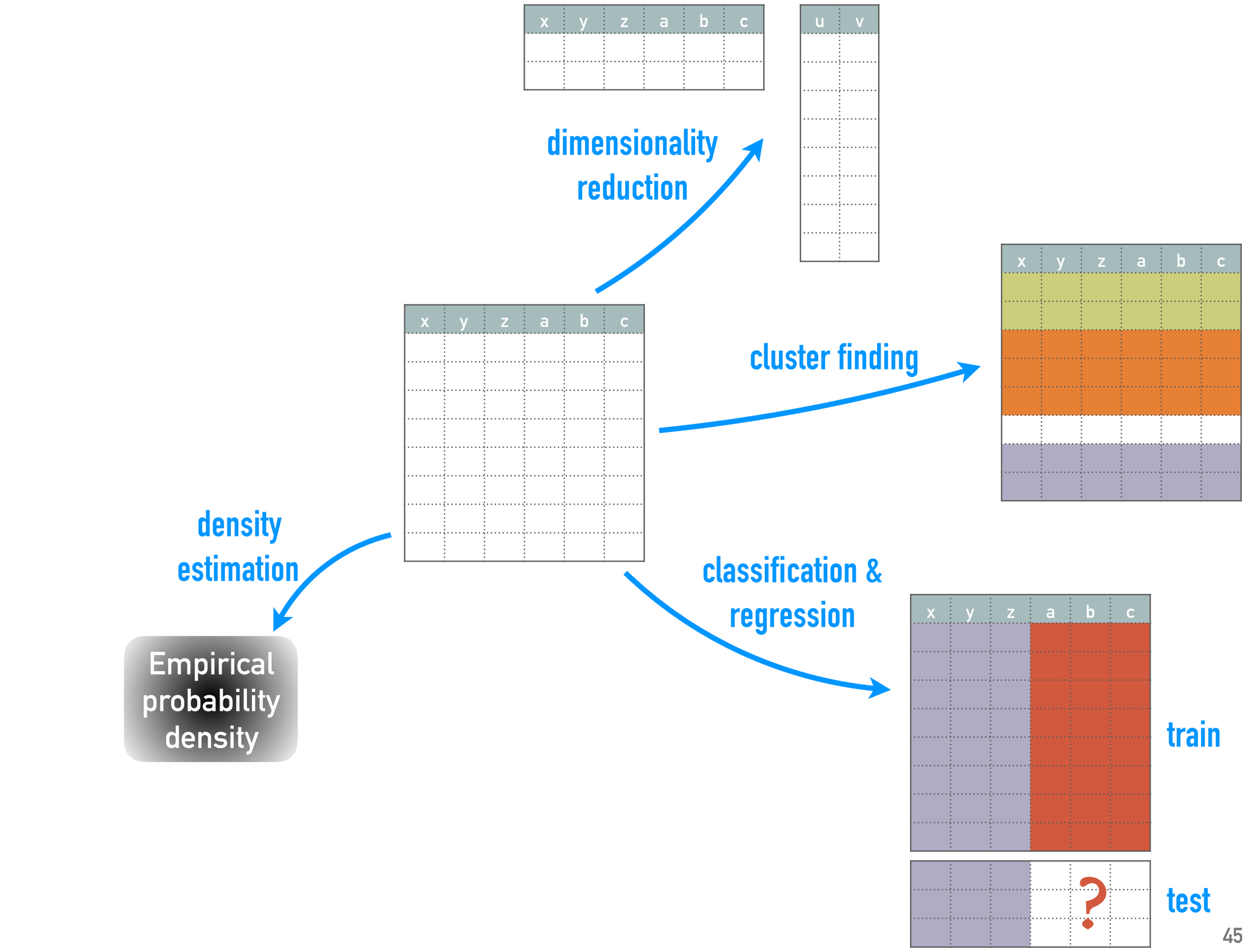
TYPES OF PROBLEM

- Classification: discrete target
- Regression: continuous target

Supervised

- Cluster finding.
- Density estimation.
- Dimensionality reduction.

Unsupervised


x y z a b c
u v
dimensionality reduction
x y z a b c
cluster finding
x y z a b c
density estimation
Empirical probability density
classification & regression
x y z a b c
train
?
test

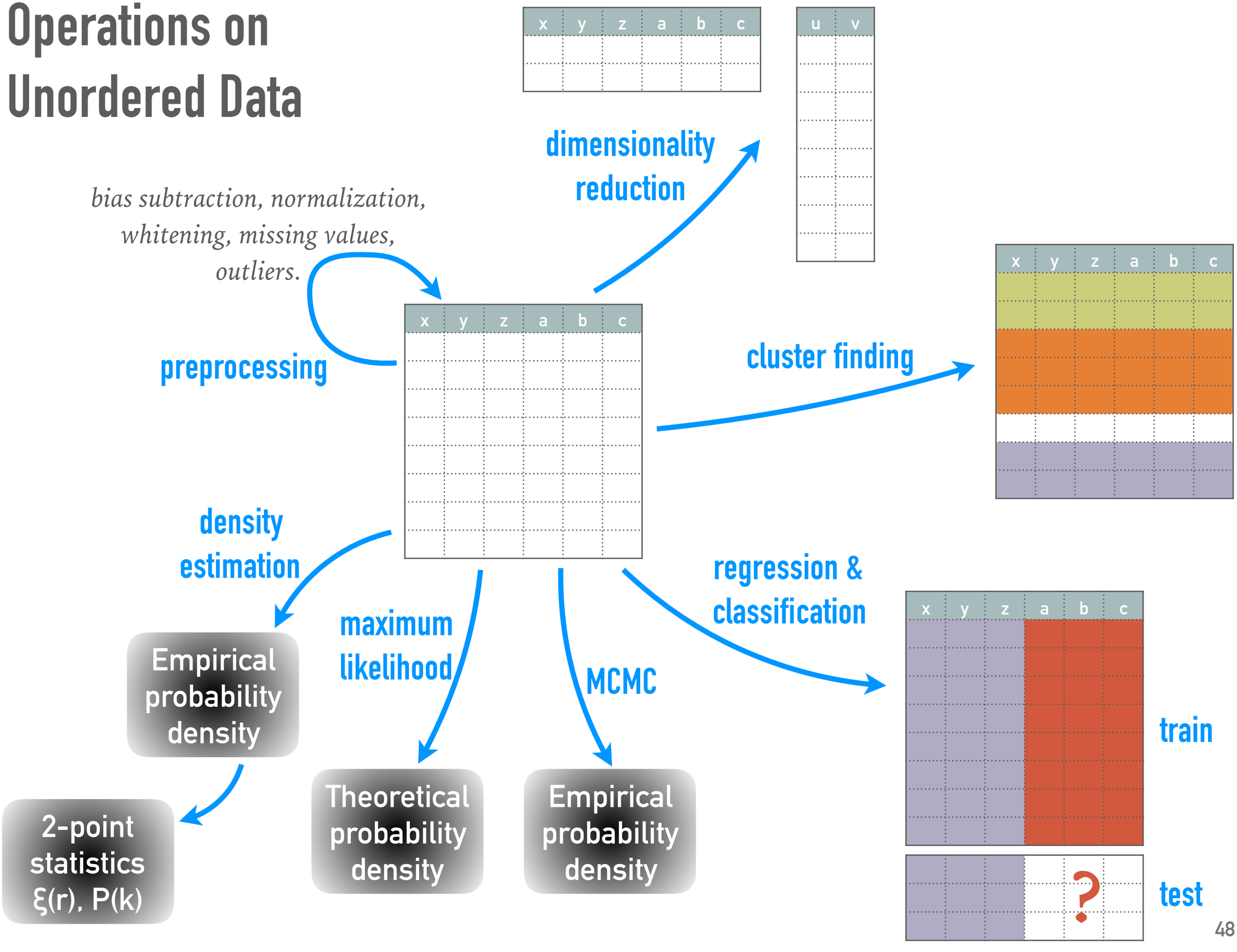
TYPES OF SOLUTION

- Fundamental problem: evidence P(D) is difficult to evaluate.
- Exact solution:
  - enumerate all possible outcomes (do it when you can!)
- Approximate solution:
  - Analytic / Deterministic:
    - maximum likelihood (best-fit parameters).
    - Laplace's approximation (parabolic errors on best-fit params).
    - variational inference (exact results for an approx. posterior).
  - Sampling:
    - Markov-chain MC (approx. results for an exact posterior).

ACTIVITY: UNDERSTAND THE PROBLEM

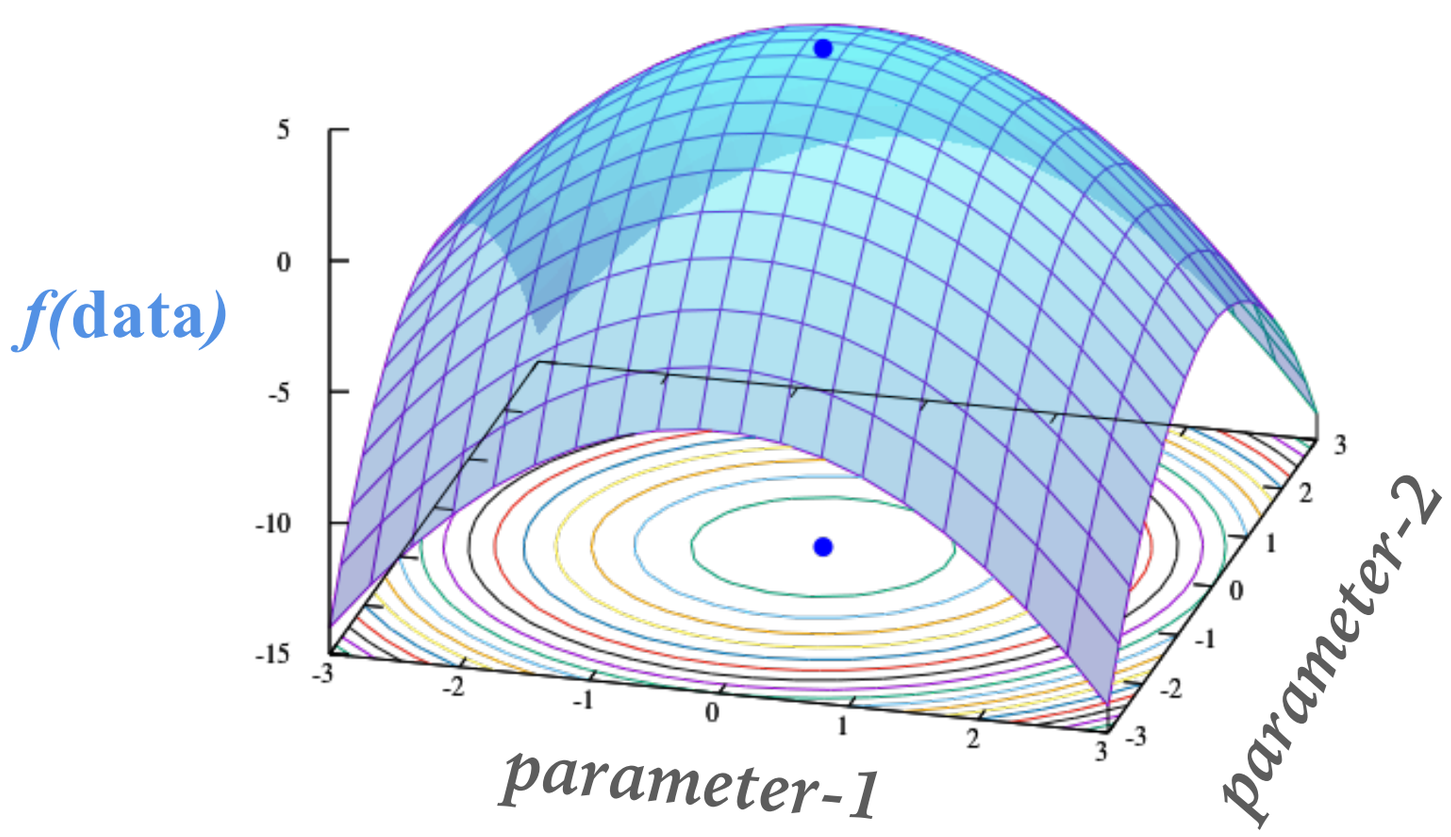
- Pick one of these machine learning problems:
  - *Predict the sky brightness for tomorrow night's observing.*
  - *Estimate a galaxy's redshift from its LSST magnitudes.*
- Is this a supervised or unsupervised learning problem?
- What approach would you try first?

Operations on Unordered Data



MACHINE LEARNING ~ OPTIMIZATION

- In practice, “learning” is accomplished via optimization:
  - minimize some function ***f*(data)** with respect to some parameters.
  - ***f*(data)** represents goodness-of-fit / loss / cost / regret / …

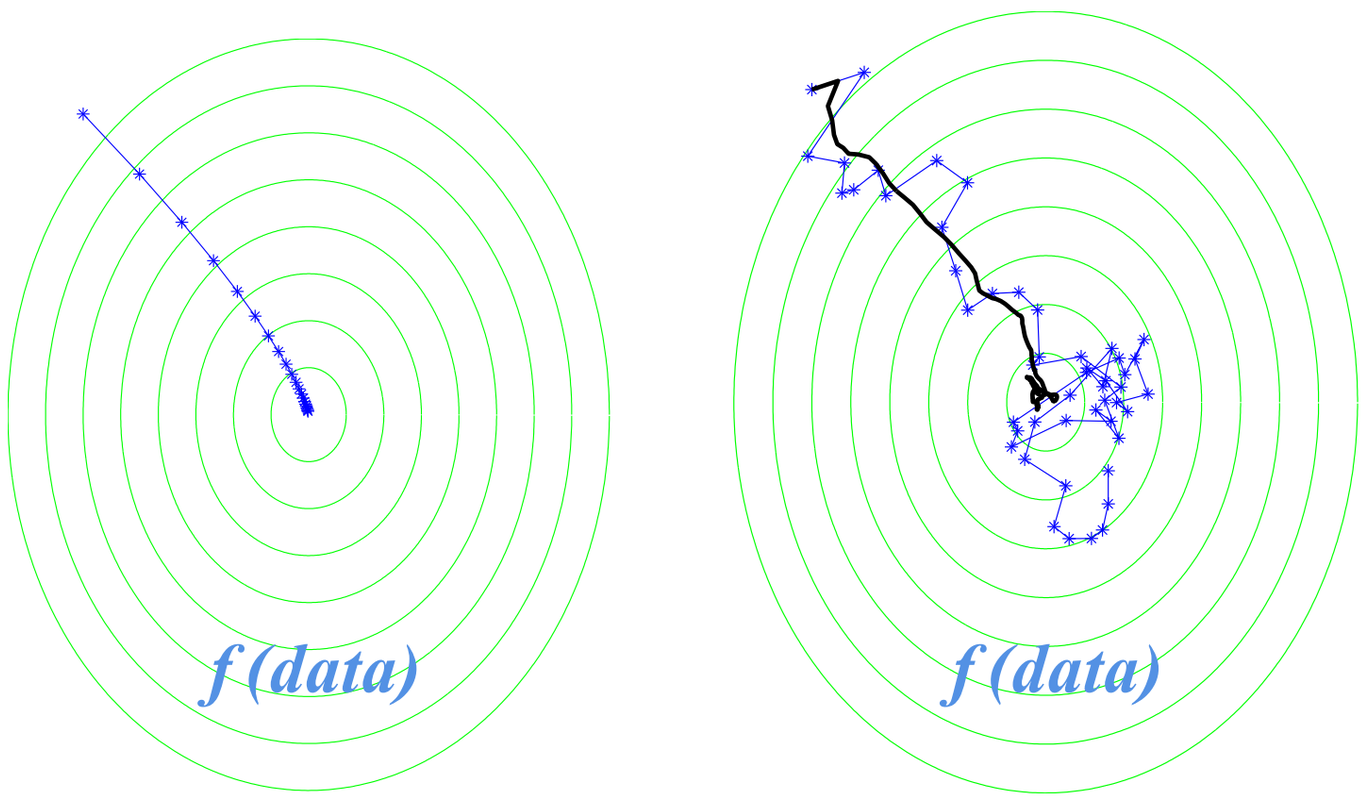


MACHINE LEARNING ~ OPTIMIZATION

- Convex / concave functions have a special role.
  - Guaranteed to find global min / max.
  - Jensen's inequality: $f(\langle X \rangle) \leq \langle f(X) \rangle$
  - discrete parameters: set submodular ~ convex/concave.
    - can sometimes coerce ***f*(data)** to be convex / concave. (regularization, hinge loss)
- Most interesting functions ***f*(data)** are not convex.
  - Non-convex optimization is formally NP-hard.
  - However, many real-world problems are tractable in practice.

MACHINE LEARNING ~ OPTIMIZATION

- Stochastic optimization is probably most important algorithm of modern machine learning.
  - use a random subset of data for each iteration.
  - provable convergence of noisy estimator to F(data) !



RECURRING THEMES OF MACHINE LEARNING

- Neighbors
- Kernels
- Ensembles
- Regularization
- Entropy

THE MACHINE-LEARNING ZOO

MODELS

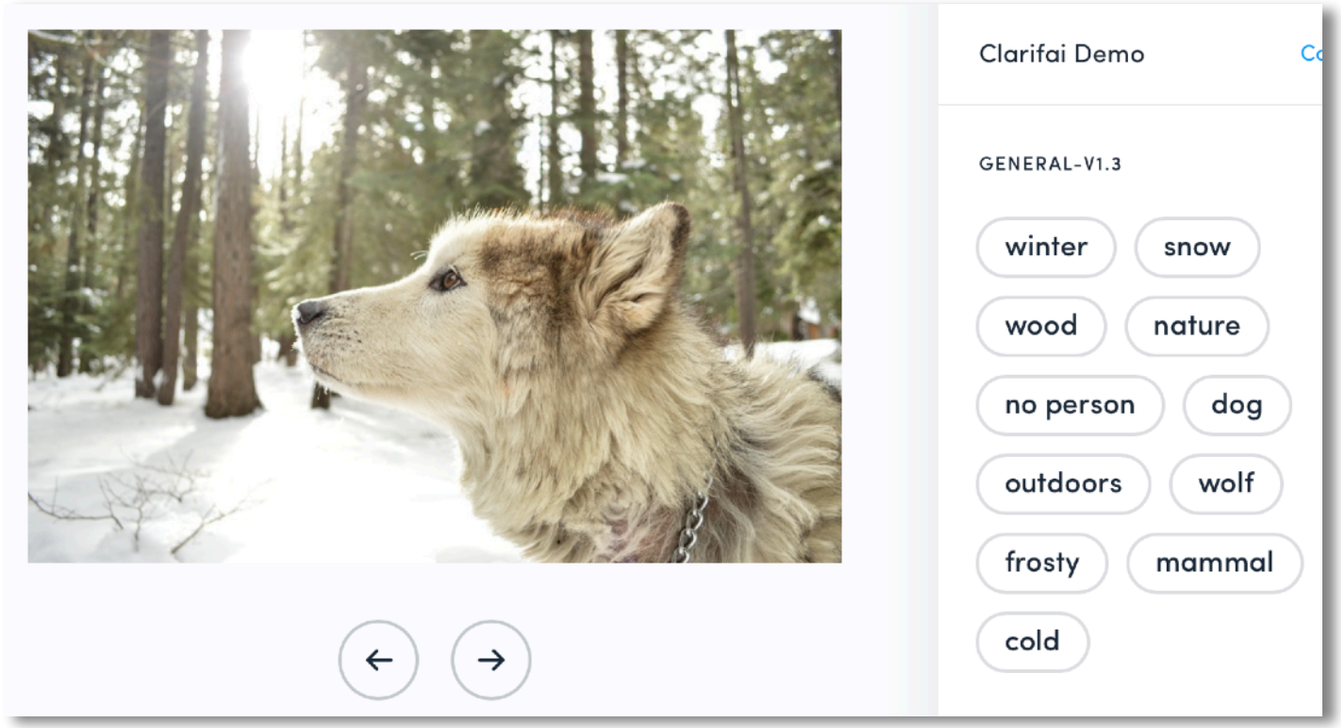
Science *Online Services* *Financial Services*

DATA

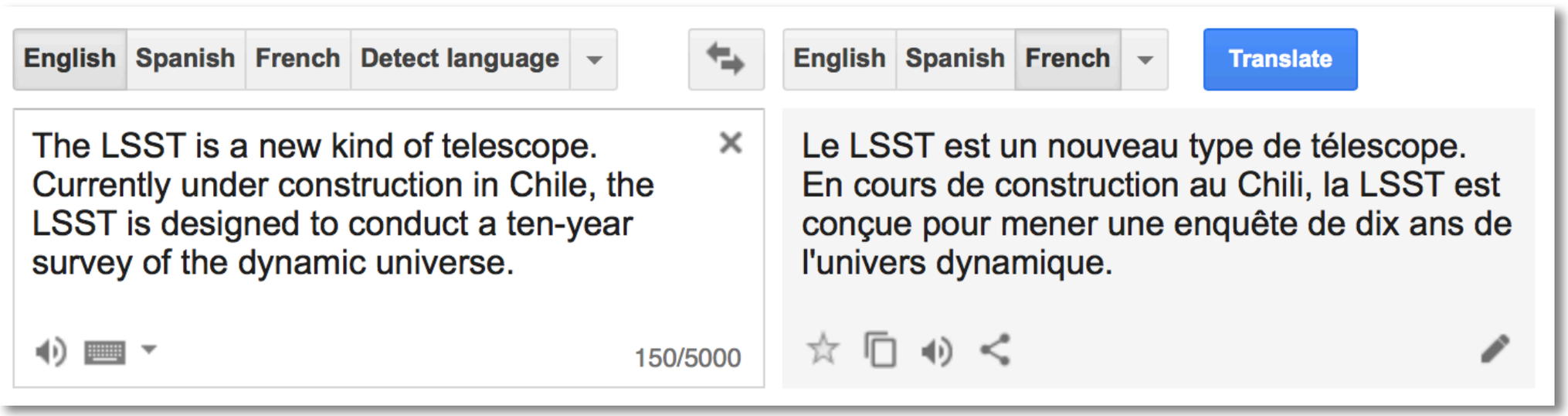
PEOPLE

http://insightdatascience.com

BLEEDING EDGE: DEEP LEARNING



Convolutional neural networks for image classification

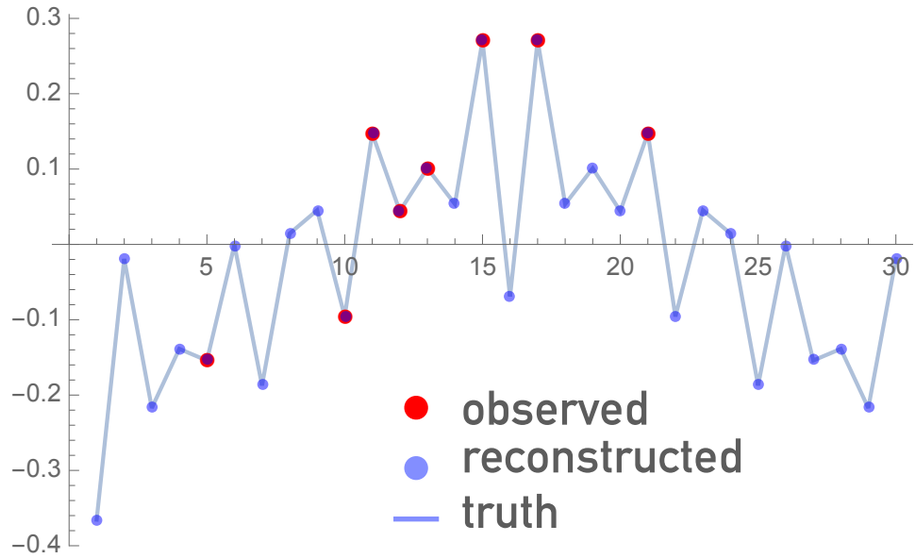


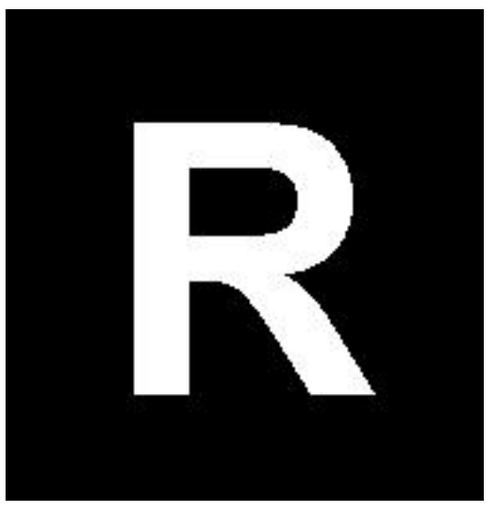
Recurrent neural networks for natural language semantics and translation

BLEEDING EDGE: COMPRESSIVE SENSING

- Reconstruct sparse information from dense data.
- Exploits incoherence of measurement & sparse representations to beat Nyquist limit!

An introduction to compressive sensing



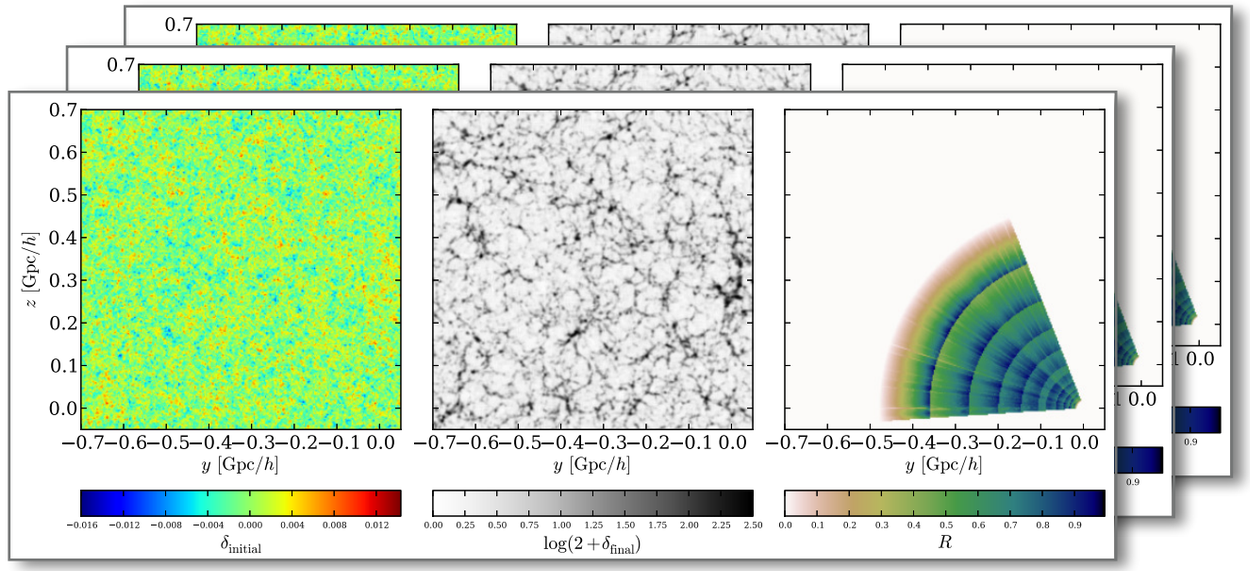


1 x 64Kpix

1300 x 1pix
(1/50)

Compressive Imaging: A New Single-Pixel Camera

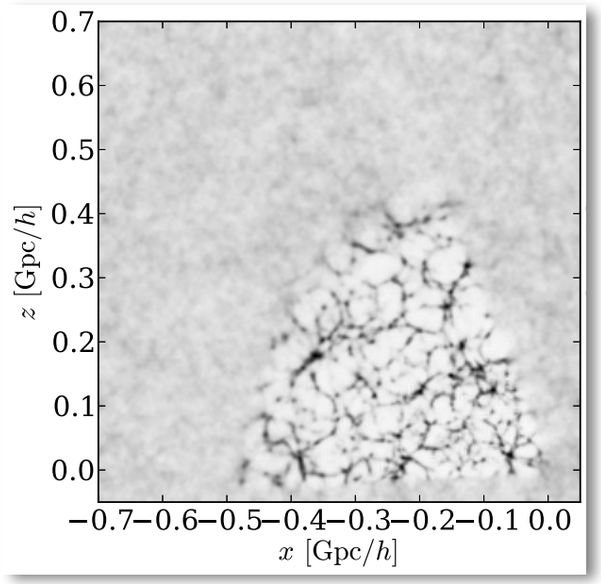
BLEEDING EDGE: HAMILTONIAN MONTE CARLO



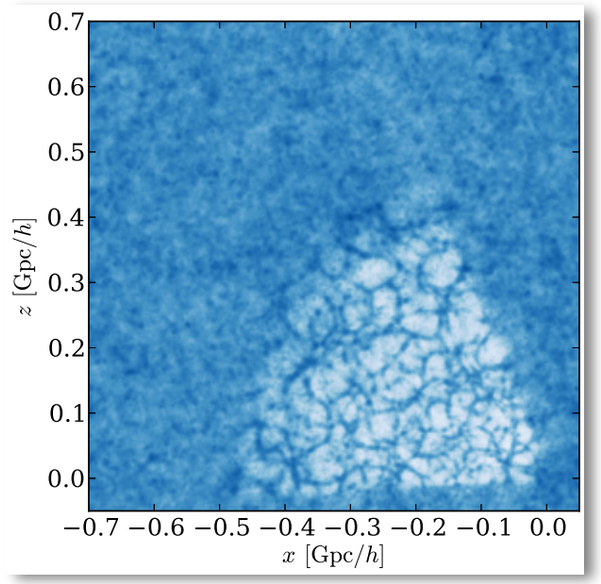
Jasche et al, "Past and present cosmic structure in the SDSS DR7 main sample"

arxiv:1409.6308

mean density



rms density

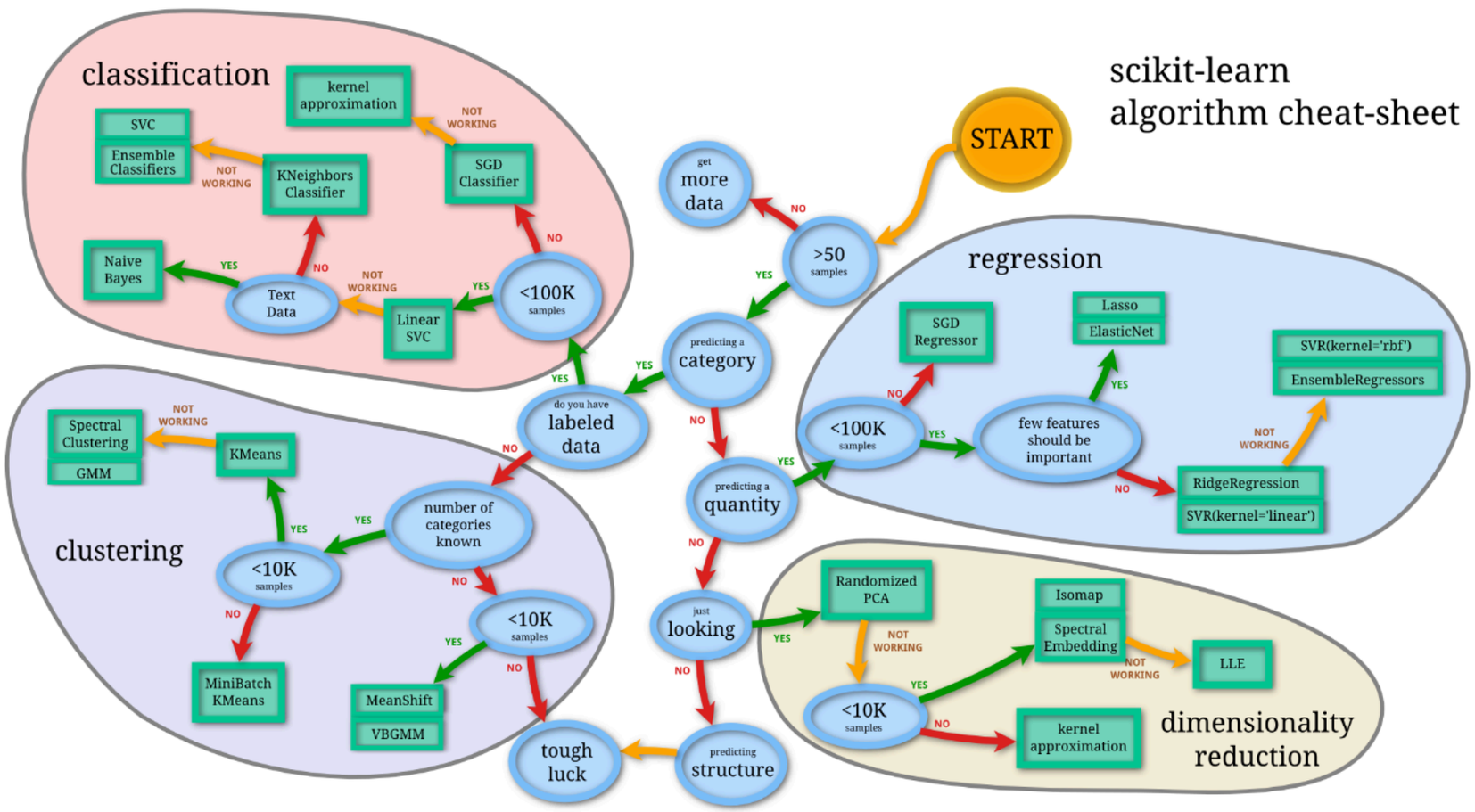


LESSONS FROM THE BLEEDING EDGE

- We need to develop & share high-quality building blocks:
  - standard data sets.
  - state of the art pre-trained solutions to low-level tasks.
- Be bold.
- Be persistent.

ONLINE RESOURCES

- LSSTC Data Science Fellows Program
  - Session 1
  - Session 2 [Intro | EM | MCMC]
- scikit-learn
- tensorflow



RECOMMENDED READING