Lecture 3

Markets, Mechanisms and Machines

David Evans and Denis Nekipelov

Goal of statistical learning theory

- Explain
  - and predict
    - using data

Compare with the goals of science discussed during last class!

Unsupervised learning

- Recover patterns/dependencies without prior knowledge about model

Unsupervised learning

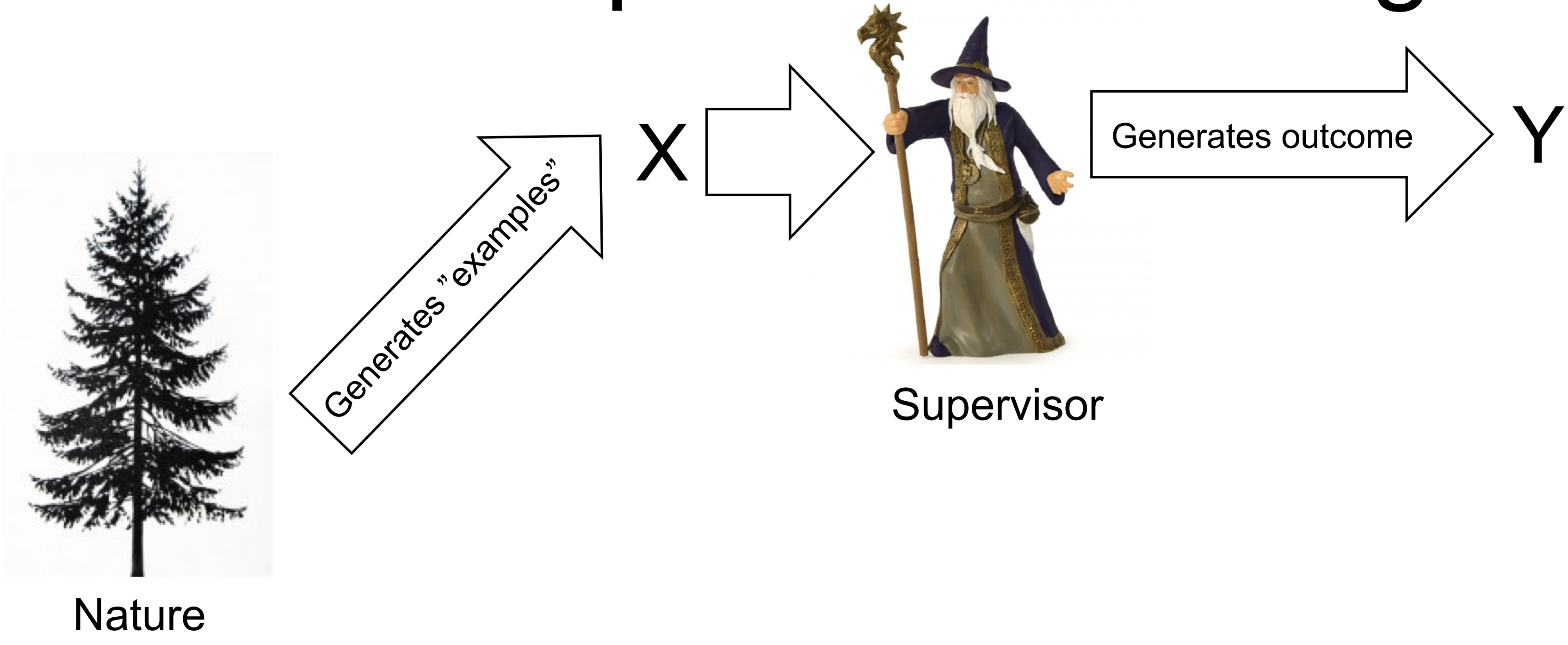
- Recover patterns/dependencies without prior knowledge about model

Unsupervised learning

- Recover patterns/dependencies without prior knowledge about model

- Do not discuss in this class but can be initial stage of data analysis

Supervised learning



Nature

Generates ”examples”

X

Supervisor

Generates outcome

Y

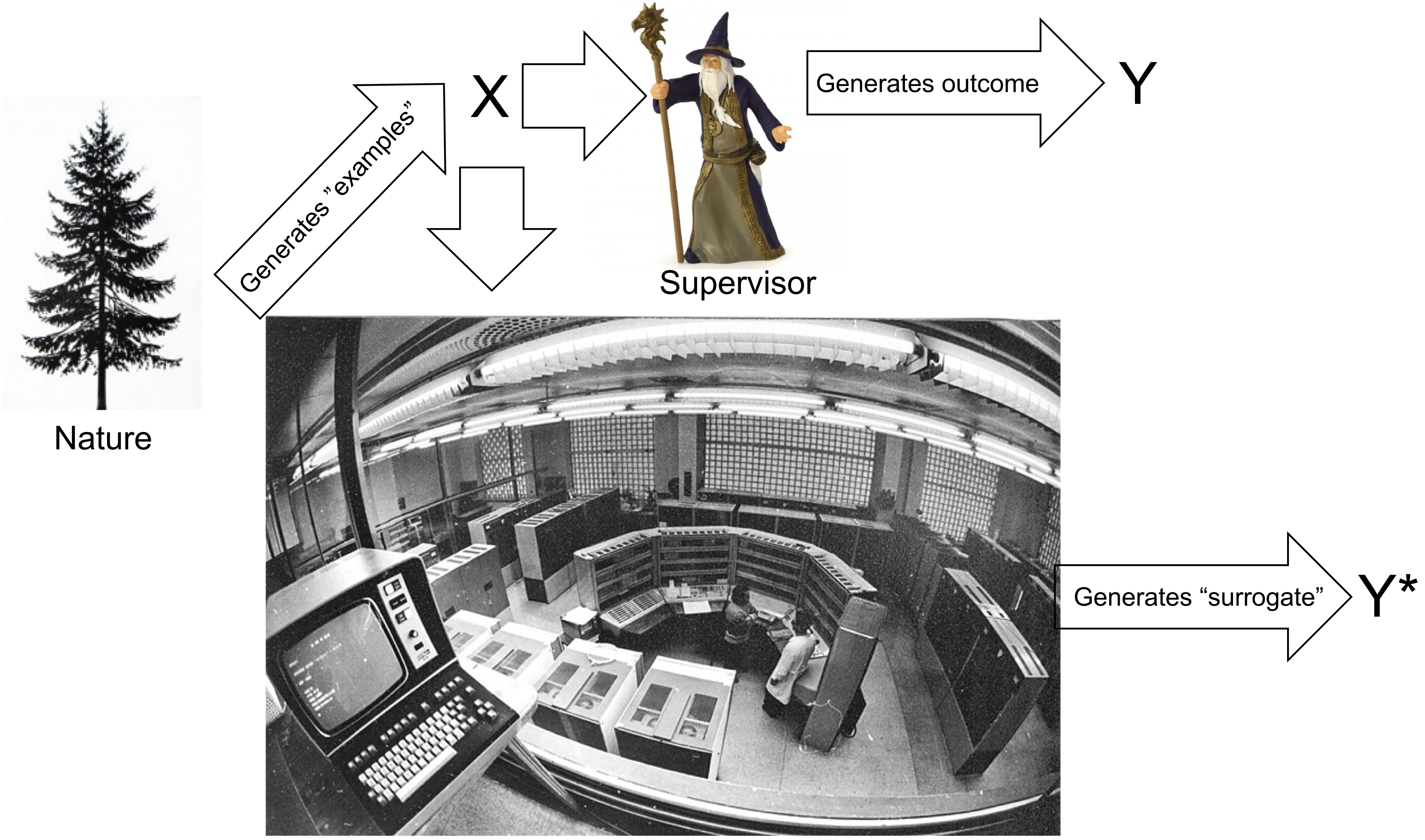
Supervised learning

- This is a model of how observed data are produced
  - Nature produces X via a particular principle (e.g. using probability distribution)

  - The supervisor **exists**

    - X and Y are related
    - There is a mapping (random or deterministic) that takes X as an input and outputs Y

Learning machine



Nature

Generates ”examples”

X

Supervisor

Generates outcome

Y

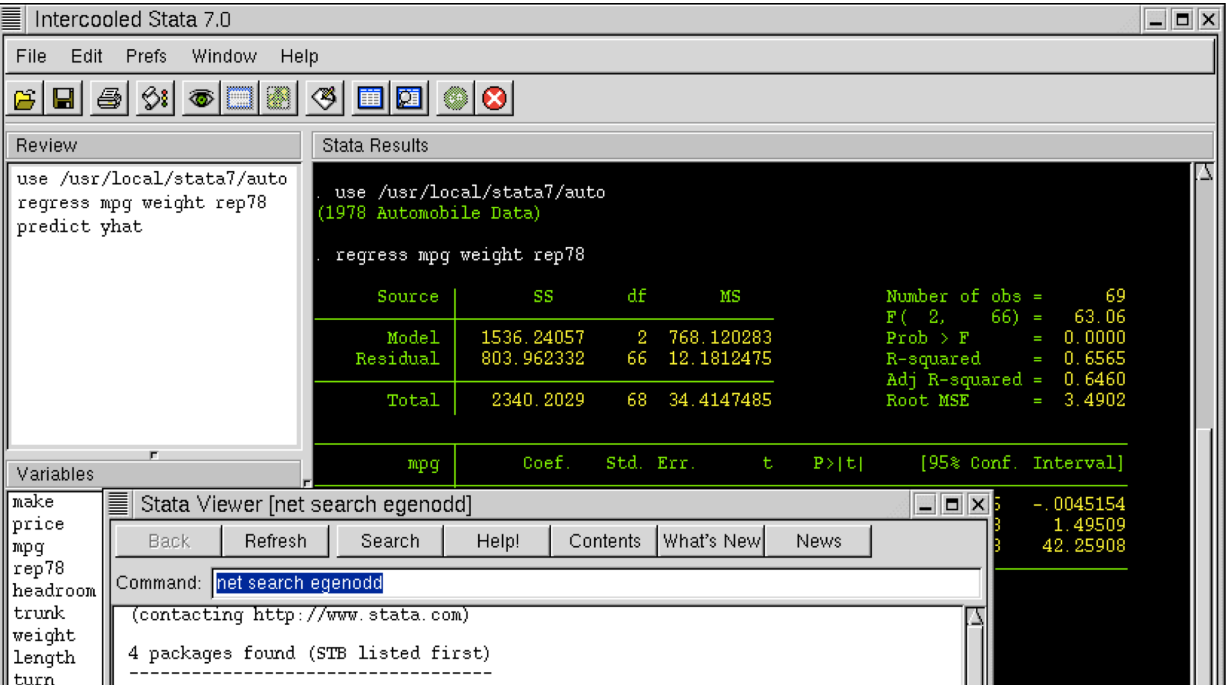
Generates “surrogate”

Y*

Learning machine

Learning machine

- Mapping (function or algorithm)
  - Uses past examples $\{X_t, Y_t\}_{t=1,\ldots,T}$ given by Nature and outputs of Supervisor
  - Tries to produce outcomes Y* close to ones produced by Supervisor for new examples given by Nature
  - May use all T data points at once



```
use /usr/local/stata7/auto
regress mpg weight rep78
predict yhat
```

```
. use /usr/local/stata7/auto
(1978 Automobile Data)

. regress mpg weight rep78

      Source |       SS       df       MS               Number of obs =      69
-------------+------------------------------            F(  2,    66) =   63.06
       Model |  1536.24057     2  768.120283            Prob > F      =  0.0000
    Residual |  803.962332    66  12.1812475            R-squared     =  0.6565
-------------+------------------------------            Adj R-squared =  0.6460
       Total |   2340.2029    68  34.4147485            Root MSE      =  3.4902

------------------------------------------------------------------------------
         mpg |      Coef.   Std. Err.       t     P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
                                                                5   -.0045154
                                                                3     1.49509
                                                                3    42.25908
------------------------------------------------------------------------------
```

```
make
price
mpg
rep78
headroom
trunk
weight
length
turn
```

Command: net search egenodd

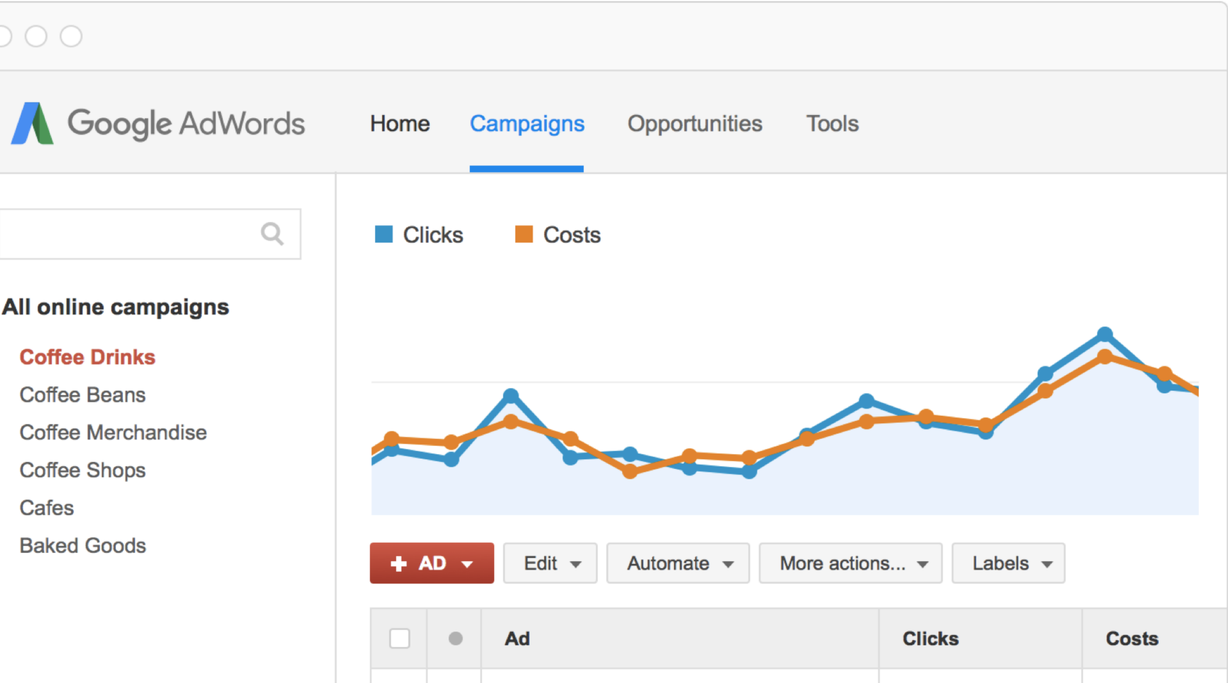
```
(contacting http://www.stata.com)

4 packages found (STB listed first)
-----------------------------------
```

Batch learning

Typical task in Econometrics and Statistics

Learning machine

- Mapping (function or algorithm)
  - Uses past examples $\{X_t, Y_t\}_{t=1,\ldots,T}$ given by Nature and outputs of Supervisor
  - Tries to produce outcomes Y* close to ones produced by Supervisor for new examples given by Nature
  - Can update with each new example



Online learning

Typical setup for online user behavior prediction and ad targeting

Designing the learning machine

- Convenient model of Supervisor: random variable
  - Supervisor draws Y from conditional distribution F(Y|X)

- Convenient model of Nature: random variable
  - Nature randomly draws examples X from F(X)
  - Note: many methods can work when Nature is **adversarial** (generates worst possible examples for learning)

- Goal of the learning machine: learn information about F(Y|X) assuming $\{X_t, Y_t\}_{t=1,\ldots,T}$ is sampled from the joint distribution

Designing the learning machine

Can pursue two goals:

1. **Predict** outcomes produced by supervisor
   - Construct surrogate Y*(X) that produces closest possible values observed outcomes Y
   - This is relatively easy (e.g. in the context of regression)
2. **Explain** outcomes produced by supervisor
   - Find out which mapping Y(X) was used by supervisor and construct Y*(X) as close as possible to it
   - This can be very difficult

Designing the learning machine

- What does it mean to “construct mapping” Y*(X)?
  - Formally, learning machine can implement some fixed set of functions (design choice)
  - During the learning process it chooses an appropriate function from this set
  - **The learning process is a process of choosing an appropriate function from a given set of functions**

Designing the learning machine

To construct learning machine we need:

1. Select the set of possible candidates for $Y^*(X)$: $Y^*_1(X), Y^*_2(X), \ldots, Y^*_N(X)$
   - E.g. possible candidates can be all linear functions $Y^*(X)=a+bX$ where *a* and *b* are arbitrary real numbers
2. Define loss function to evaluate performance of the learning machine relative to the output from the Supervisor
   - E.g. we already introduced a quadratic loss function $l(Y,Y^*(X))=(Y-Y^*(X))^2$

Designing the learning machine

3. Determine which function from you selected set of candidates minimizes the expected loss over the distribution of examples provided by Nature and the distribution of the outcomes produced by the Supervisor

$$R(Y^*(X))=E[l(Y,Y^*(X))]$$

- This is called ”the risk function”
- The learning machine solves the problem of risk minimization

Loss functions

- How do we choose the loss function

  - Cannot be arbitrary: the choice of loss function is linked to the specifics of the problem
  - Poor choice of loss function can hinder performance of the learning machine
  - Convexity of the loss function is the key property in most cases

Important examples of risk minimization

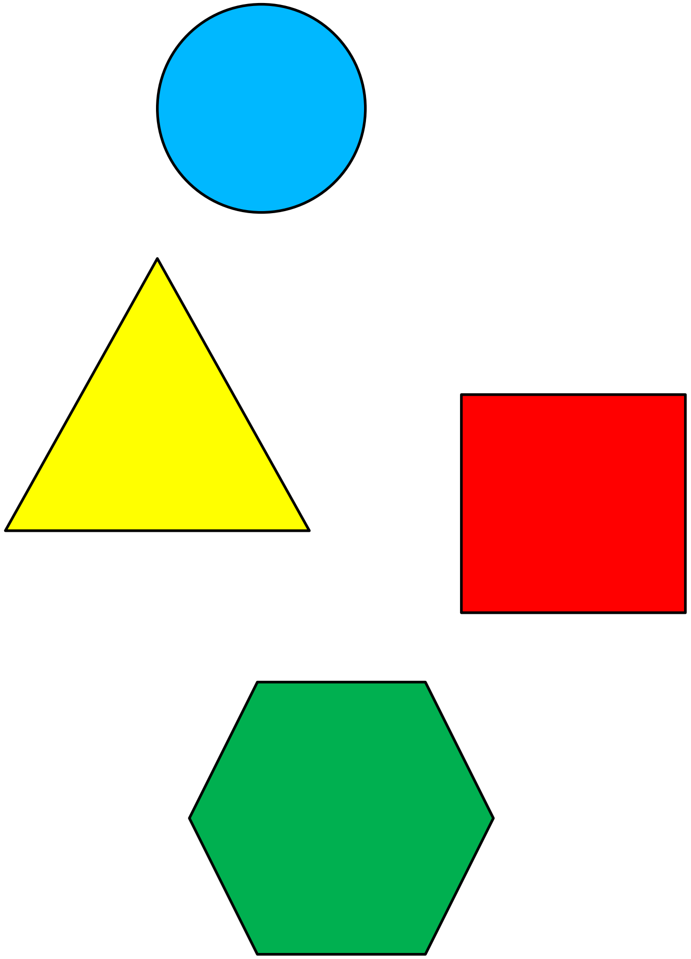
1. Regression

2. Pattern recognition

3. Density estimation

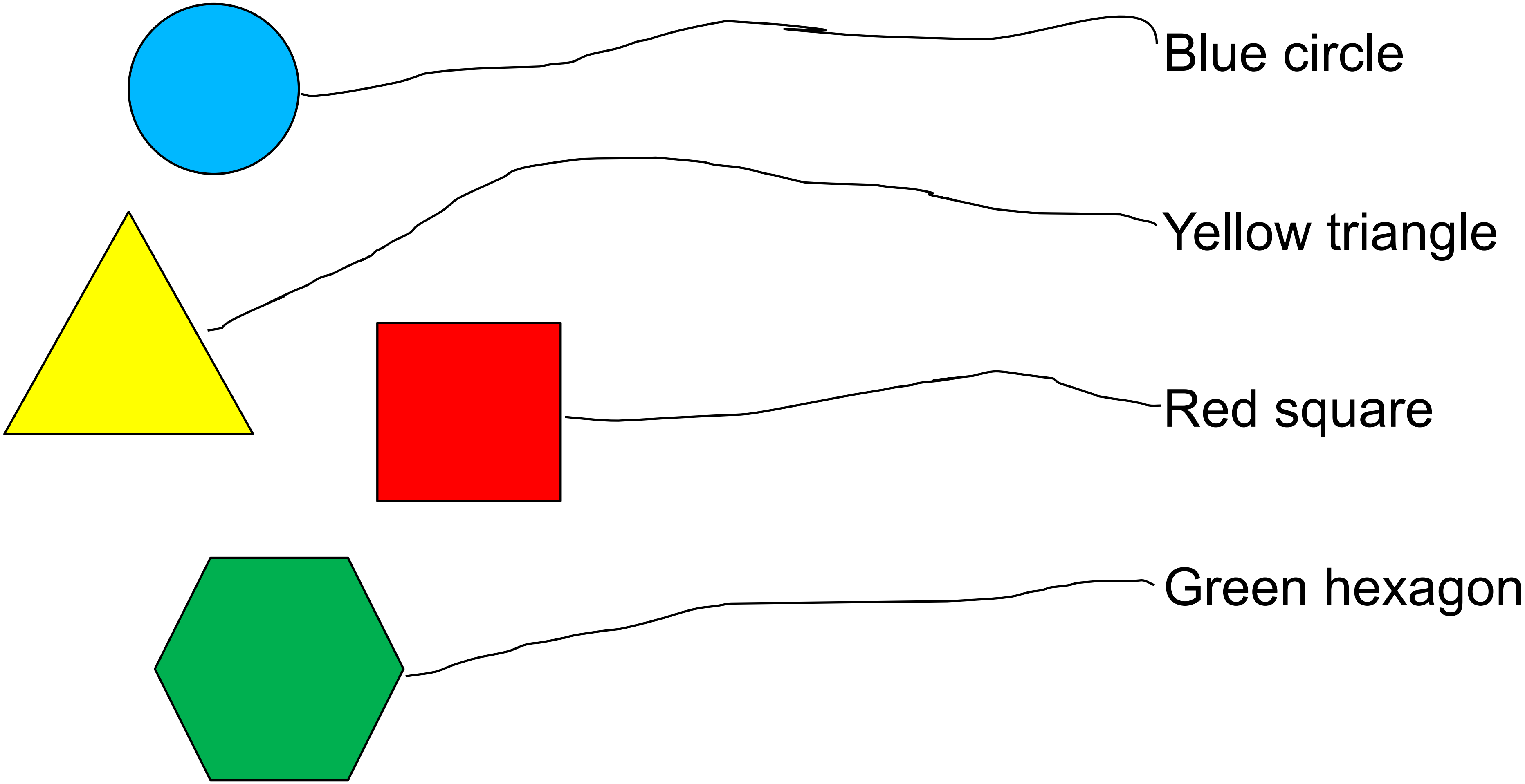
Pattern recognition

Nature generates examples with specific features



Pattern recognition

Supervisor assigns them to fixed set of classes



Blue circle

Yellow triangle

Red square

Green hexagon

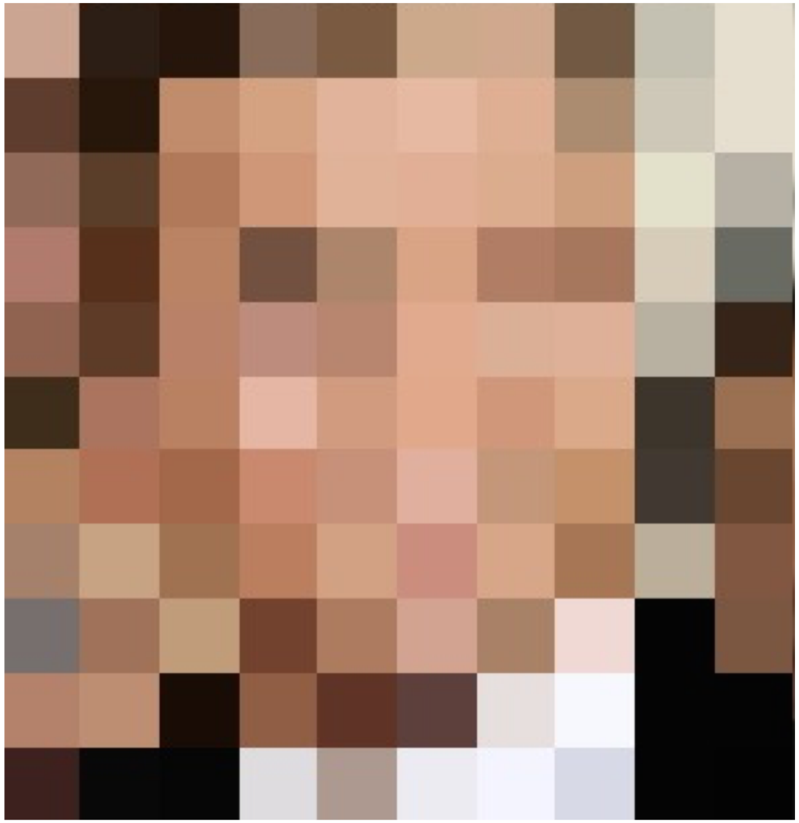
Pattern recognition

Learning machine

- searches for appropriate functions that take features of objects as input and output their labels
- Simplest label system is binary
  - Labels are of two types
  - Pioneered by F. Rosenblatt in 1950’s: perceptron
    - Simple linear separator
    - At the core of neural networks and many modern ML techniques (deep learning, GANs)

Pattern recognition

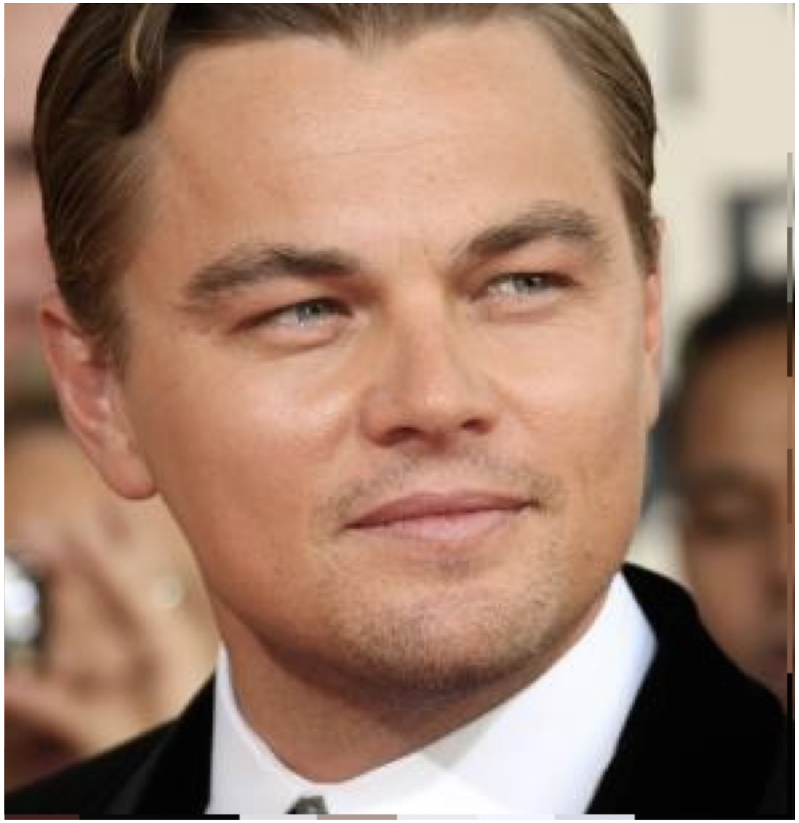
- In image recognition features are typically colors of pixels



- Ryan Gosling or Madonna?

Pattern recognition

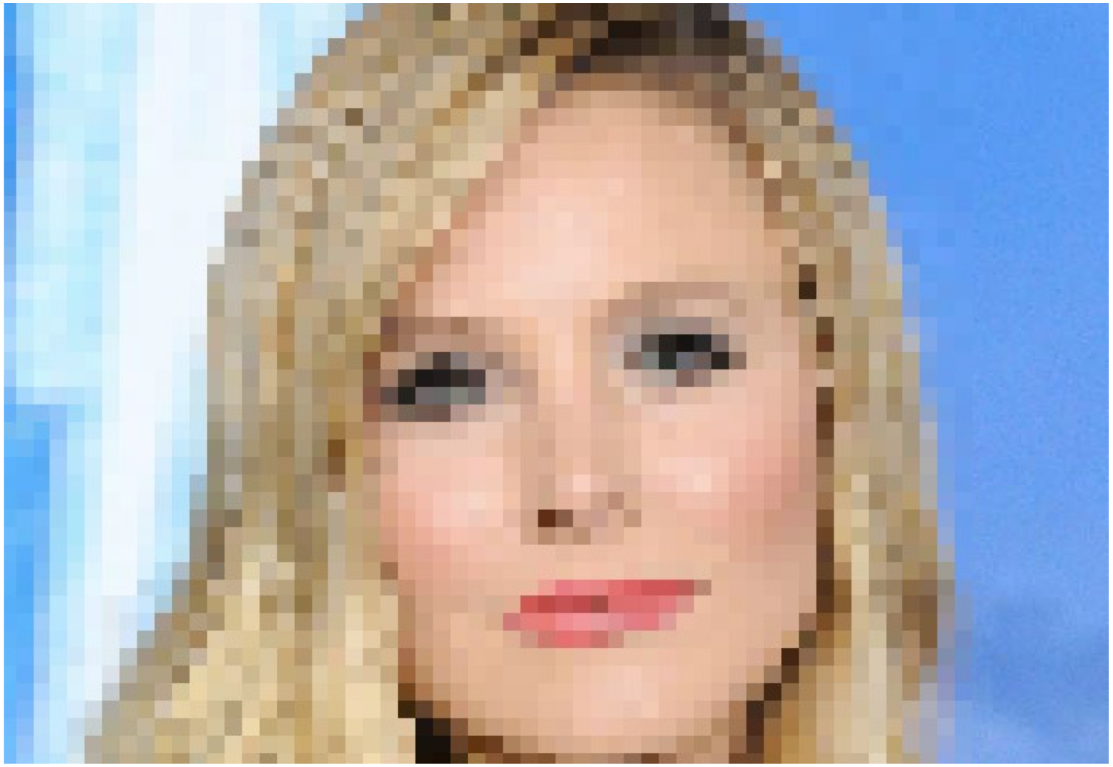
- Of course, the model needs to be **correctly specified** (we need to know all possible labels)



- Ryan Gosling or Madonna?

Pattern recognition

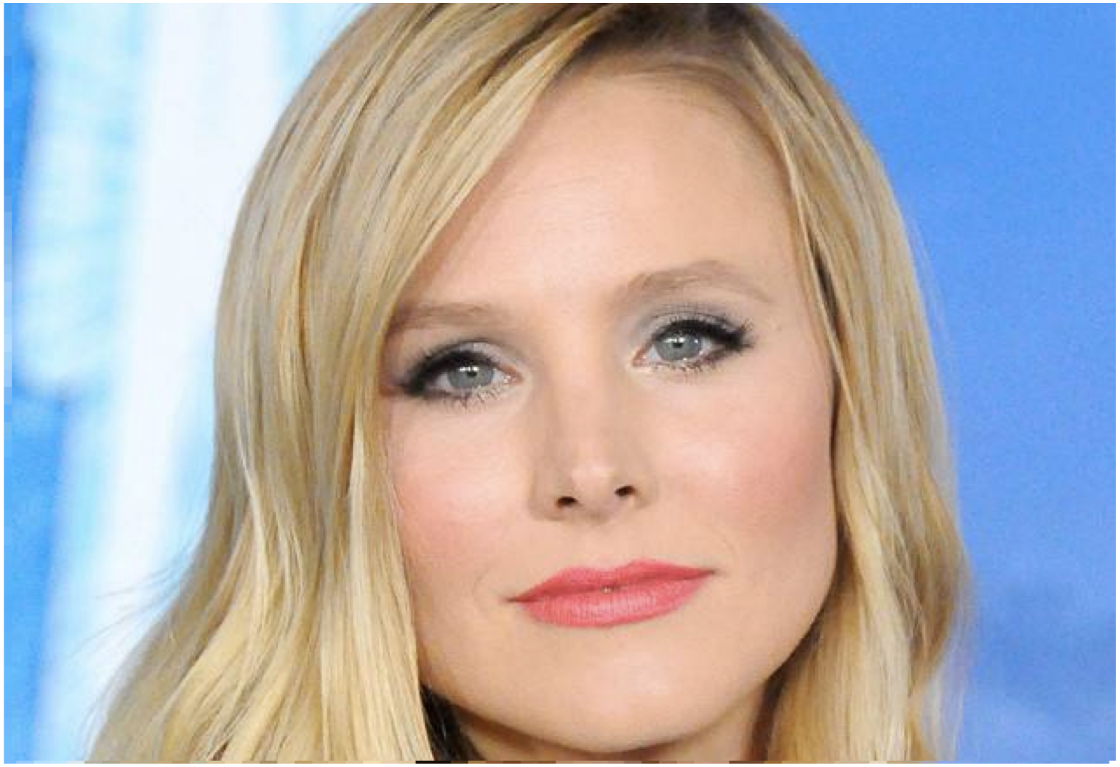
- With well-selected features and correctly specified label set can learn with few samples



- Ryan Gosling or Kristen Bell?

Pattern recognition

- With well-selected features and correctly specified label set can learn with few samples



- Ryan Gosling or Kristen Bell?

Pattern recognition

- Loss function is binary: loss=1 if example is mislabeled and 0 otherwise
- Note: very different from regression loss
  - Once you learn that specific set of features correspond to a specific label, there is no need to further improve accuracy
  - With regression loss, accuracy monotonically increases as the number of examples grows
- This explains faster learning rates for pattern recognition problems
  - Some can be learned **exactly** with fixed number of examples

Density estimation

- The task of not just **predicting** what the Supervisor does but **understanding** it
- This is the true task of causal inference
- This what makes it the hardest one
- It also shows clearly why we need a model to do causal inference

Density estimation

- Consider simplified setting where examples from Nature are trivial (e.g. all exactly the same)
- Supervisor does not use input from Nature and just produces an outcome using random variable Y
- The mapping used by Supervisor is then simply **the distribution** of Y

Density estimation

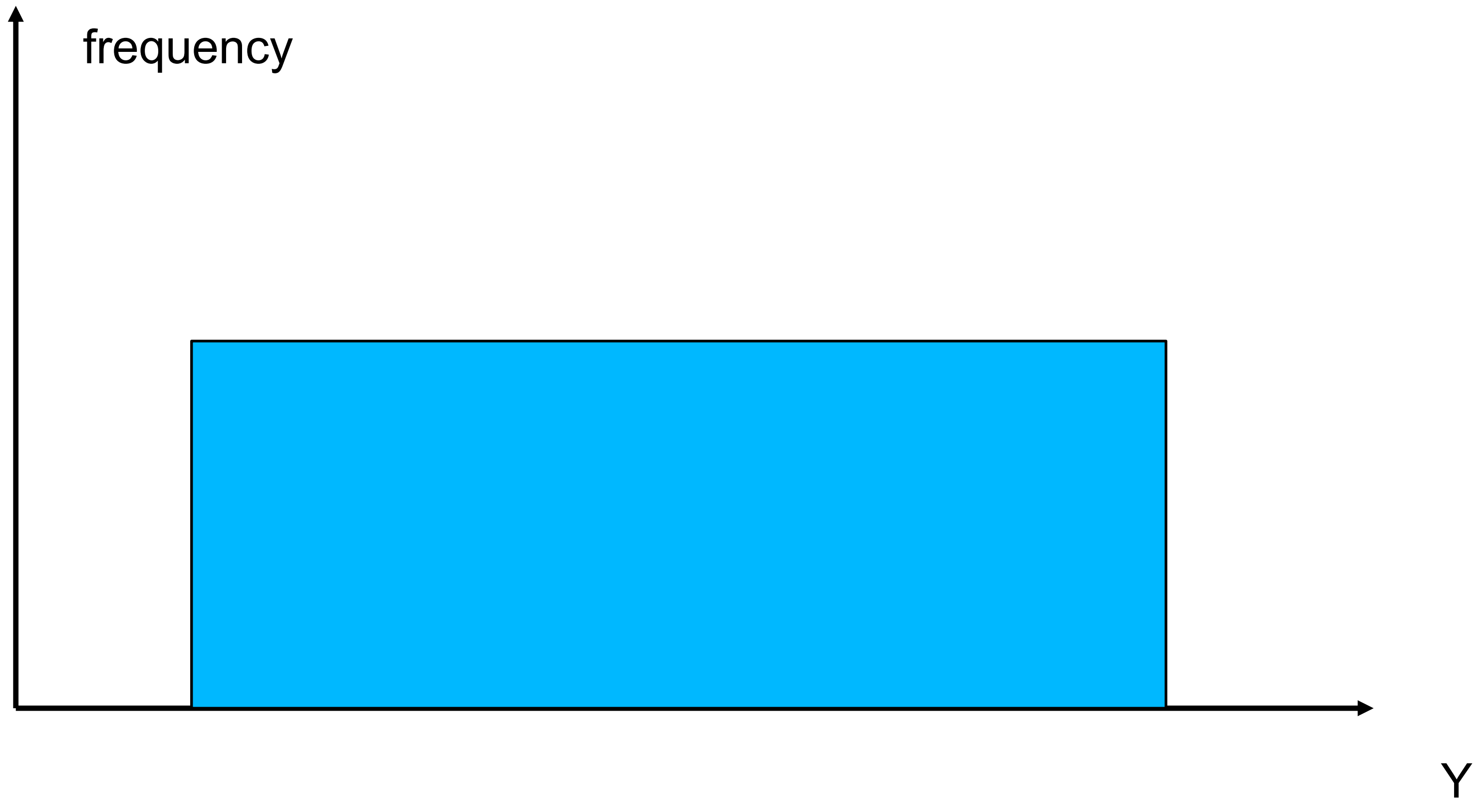
- If we can estimate that distribution, we fully learn what supervisor does

- If we recover this distribution correctly we recover the true causal model for the outcome

- To further simplify analysis, assume that Y is a single-dimensional continuous random variable
  - I.e. it has a density function

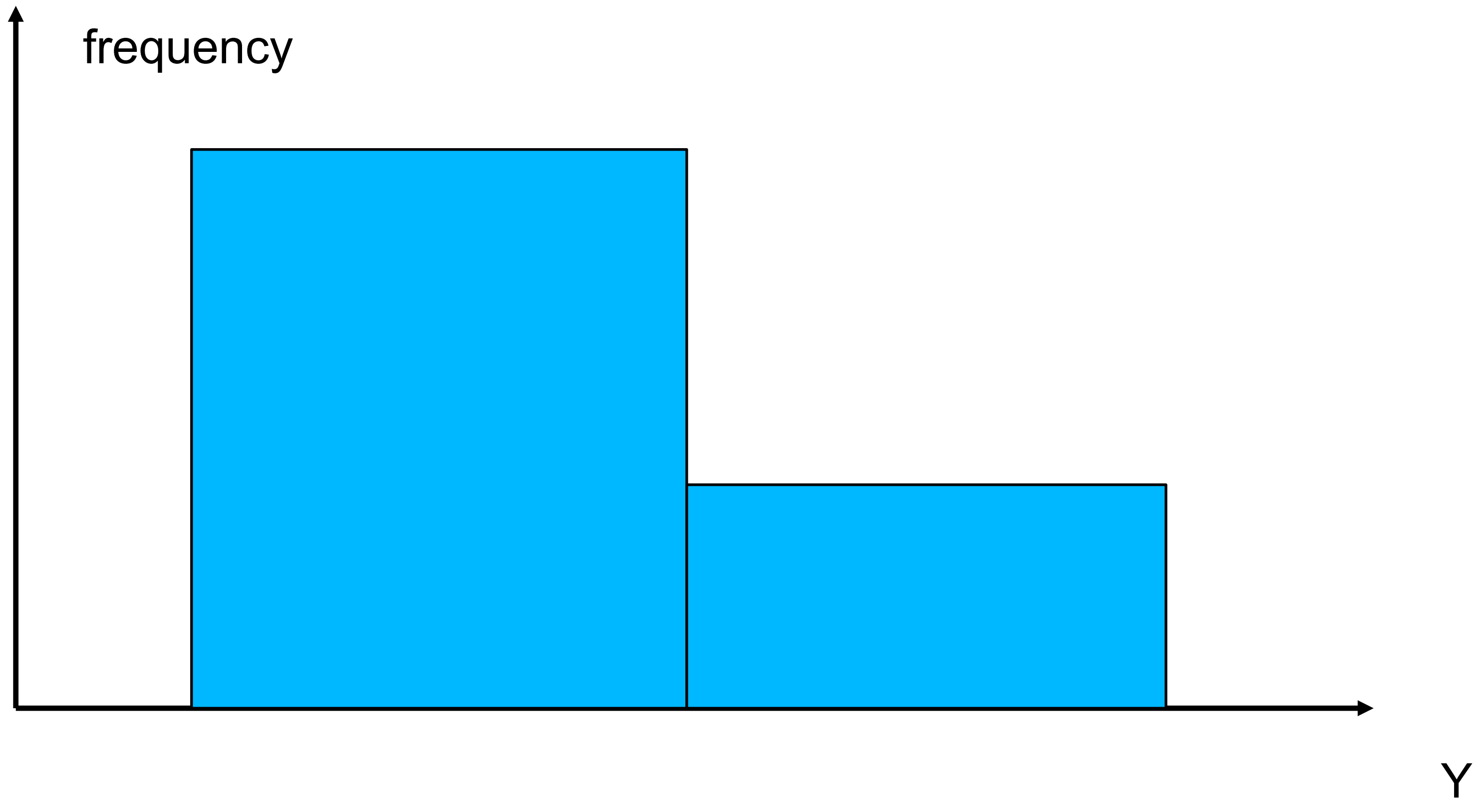
Density estimation

- Have a good estimator for density: histogram
  - Pick a bin size
  - Report frequency counts in equally sized bins



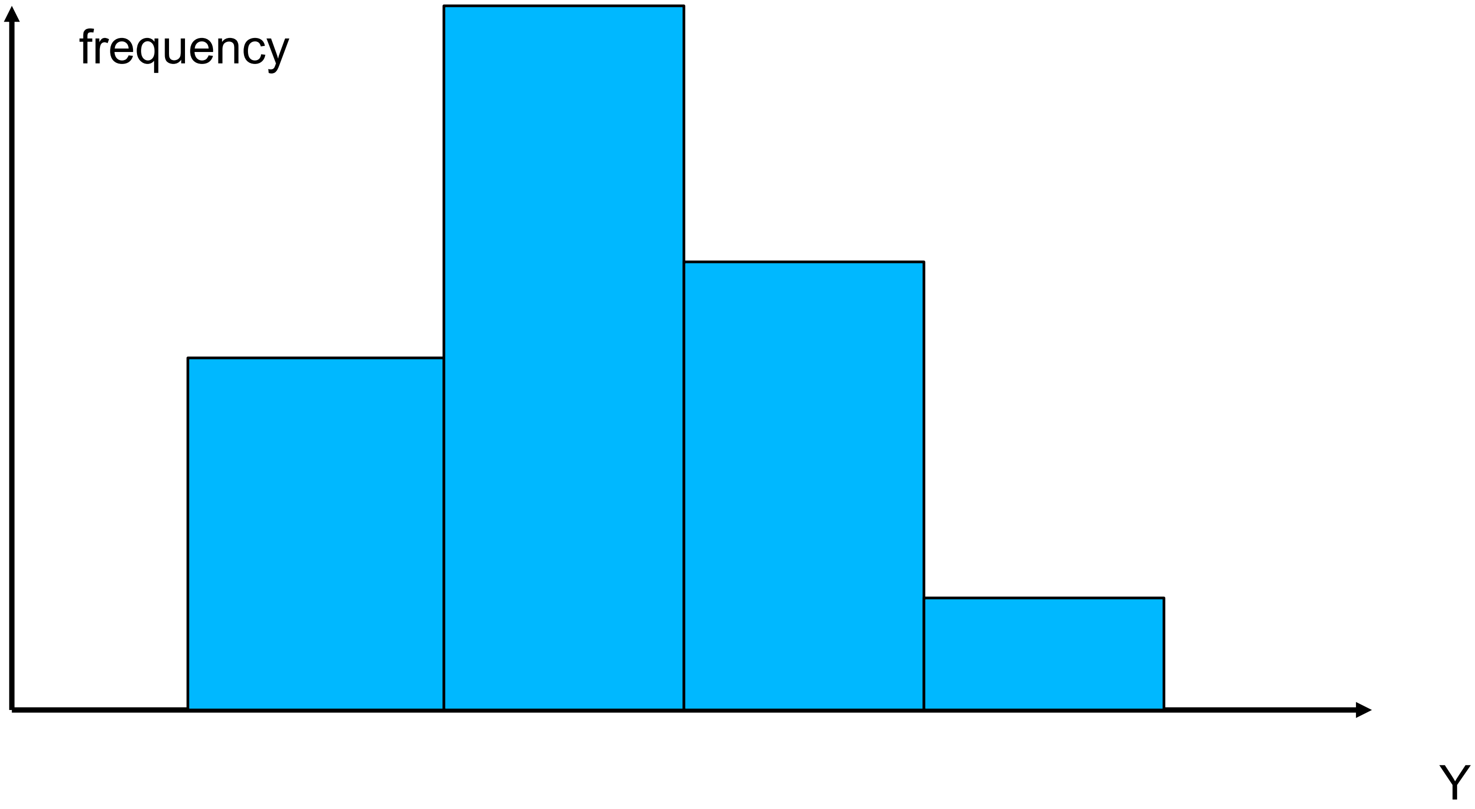
Density estimation

- Have a good estimator for density: histogram
  - Pick a bin size
  - Report frequency counts in equally sized bins



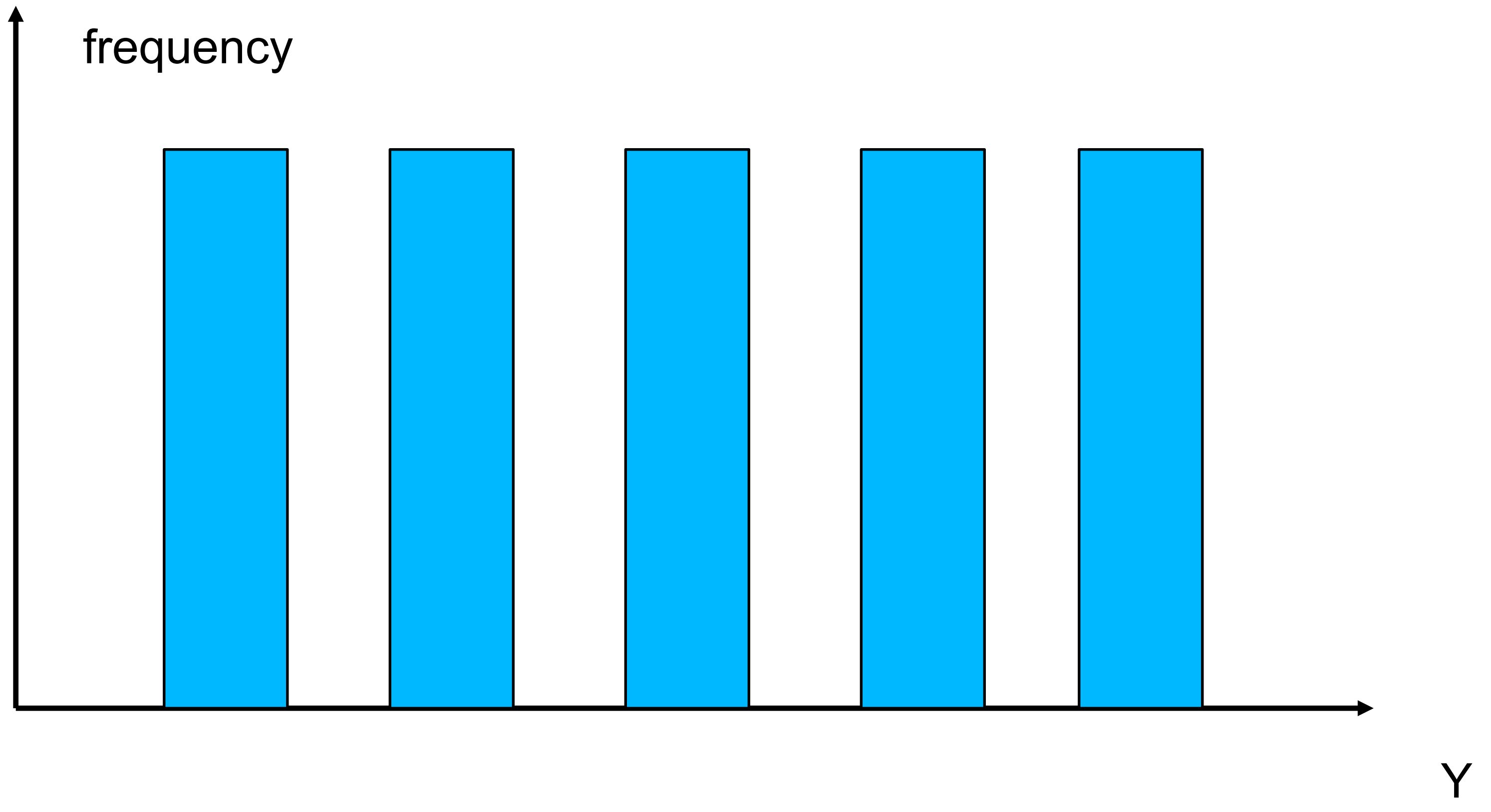
Density estimation

- Have a good estimator for density: histogram
  - Pick a bin size
  - Report frequency counts in equally sized bins



Density estimation

- Have a good estimator for density: histogram
  - Pick a bin size
  - Report frequency counts in equally sized bins



Density estimation

- How do we pick the bin size?

- Which of the histograms plotted before most accurately characterizes the distribution of the data?

Density estimation

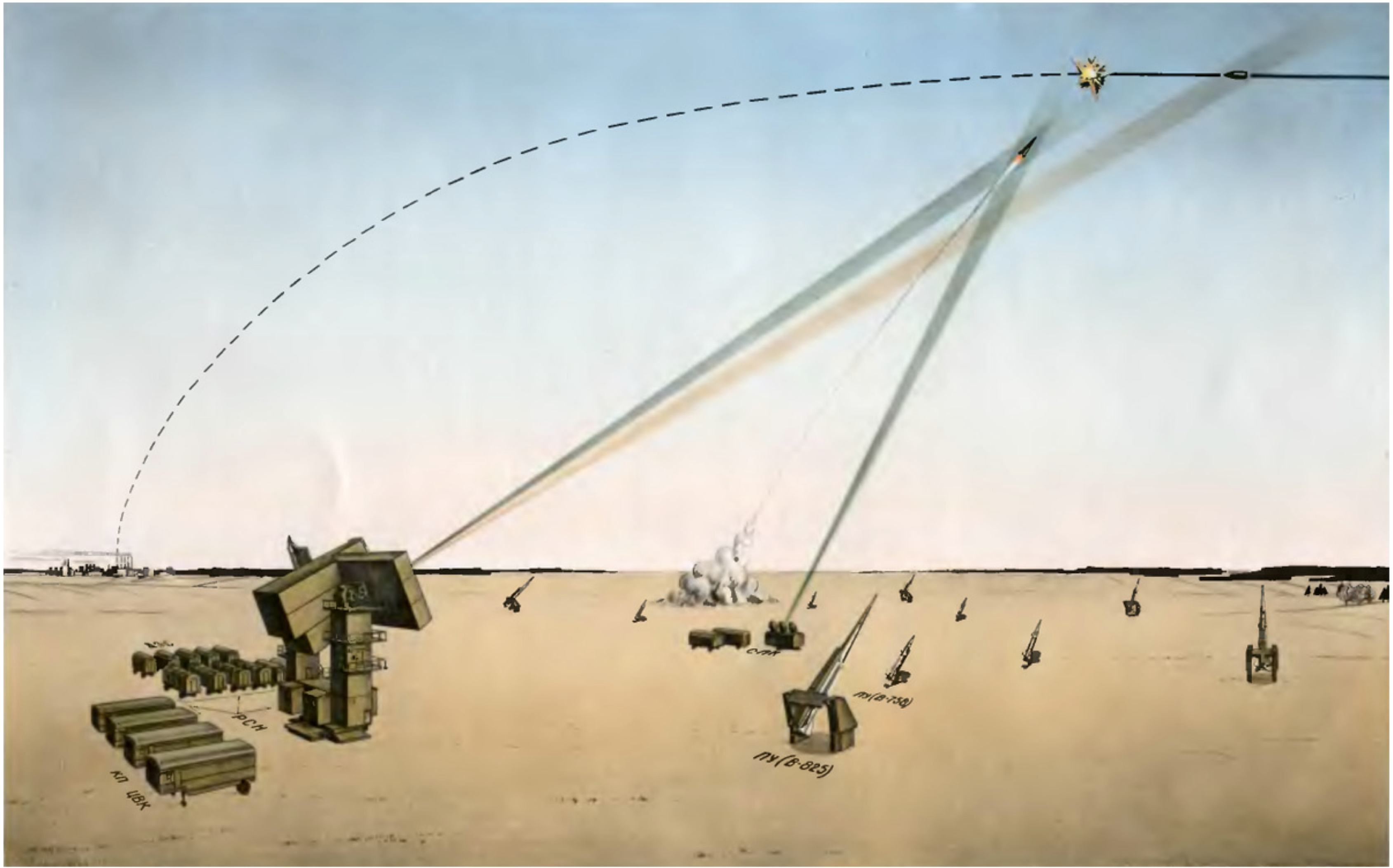
- The most accurate answer makes the least sense
- This is the reflection of the fact that density estimation without any additional conditions is an **ill-posed inverse problem**
  - Small changes in the data inputs (e.g. a single observation with a large values of Y) can lead to large changes in estimated density

Density estimation

- Ill-posed inverse problem is typical if causal inference is attempted without having a model
- In case of estimating density the model means requiring some additional properties of the density, such as existence of its first derivative
- Technically this means that when we choose bin size by looking at the mean-squared error we add penalty for large changes of estimated density
- This augmentation of the mean-squared error is called regularization

Regularization

- Regularization comes from idea of Tikhonov
  - Predicting motion of objects under external control



Regularization

- Regularization comes from idea of Tikhonov
  - Predicting motion of objects under external control
  - Impossible without any extra information: operator of aircraft or drone is the Supervisor
  - There are constraints in how objects can move (Newton's laws, atmospheric conditions)
  - These constraints can be used to form penalty in the prediction problem
- In case of density estimation constraints force estimated function to satisfy particular properties: smoothness, monotonicity, etc.