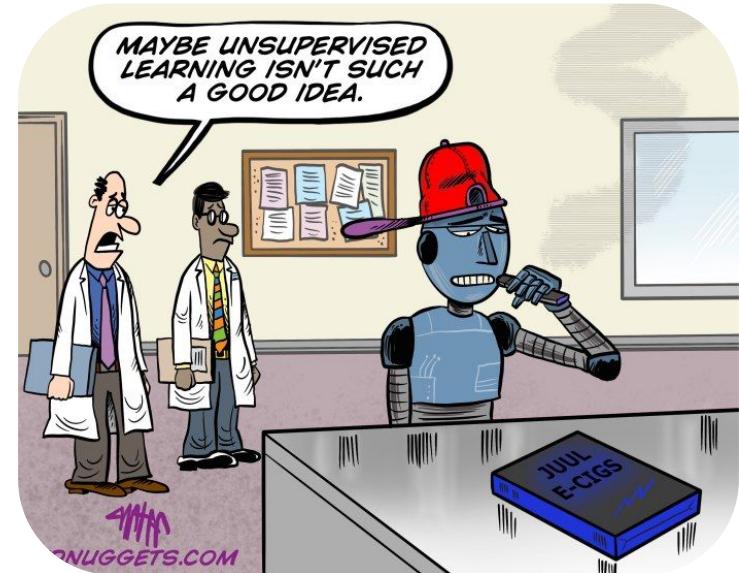




Data Science (COSE471) Spring 2021

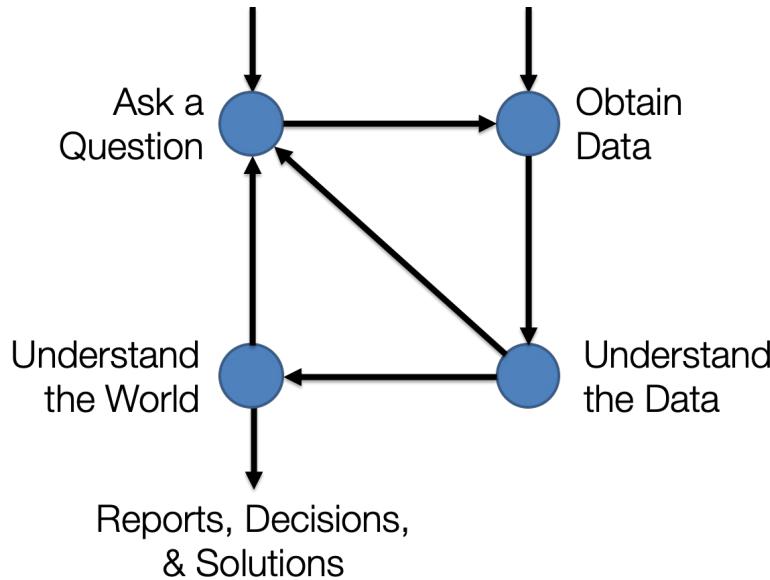
Modeling

Dept. of Computer Science and Engineering
Korea University



* This material is adapted from Berkeley CS 100 (ds100.org) and may be copyrighted by them.

Recap: Data science lifecycle



We're now moving to the fourth stage of the data science lifecycle – **Understand the World** – where we build models that try and generalize patterns in the data we collected.

What is a model?



"I love your business model!"

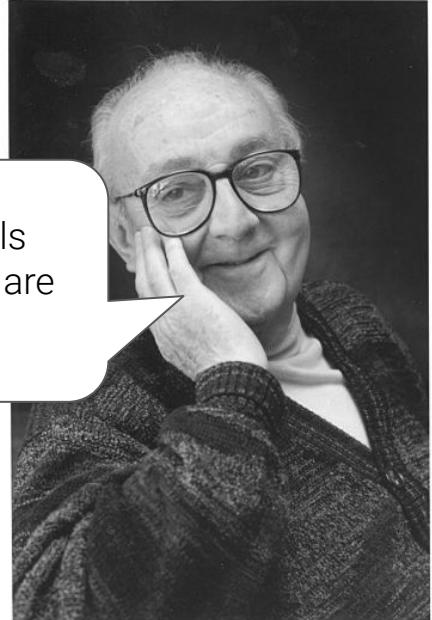
What is a model?

A model is an **idealized representation** of a system.

For example, we model the fall of an object on Earth as subject to a constant acceleration of 9.81 m/s^2 due to gravity.

- While this describes the behavior of our system, it is merely an approximation.
- It doesn't account for the effects of air resistance, local variations in gravity, etc.
- But in practice, it's accurate enough to be useful!

"Essentially, all models are wrong, but some are useful."



George Box, Statistician
(1919-2013)

Why do we build models?

To understand complex phenomena occurring in the world we live in.

- What factors play a role in the growth of COVID-19?
- How do an object's velocity and acceleration impact how far it travels?
(Physics: $d = d_0 + vt + \frac{1}{2}at^2$)

Often times, we care about creating models that are **simple** and **interpretable**, allowing us to understand what the relationships between our variables are.

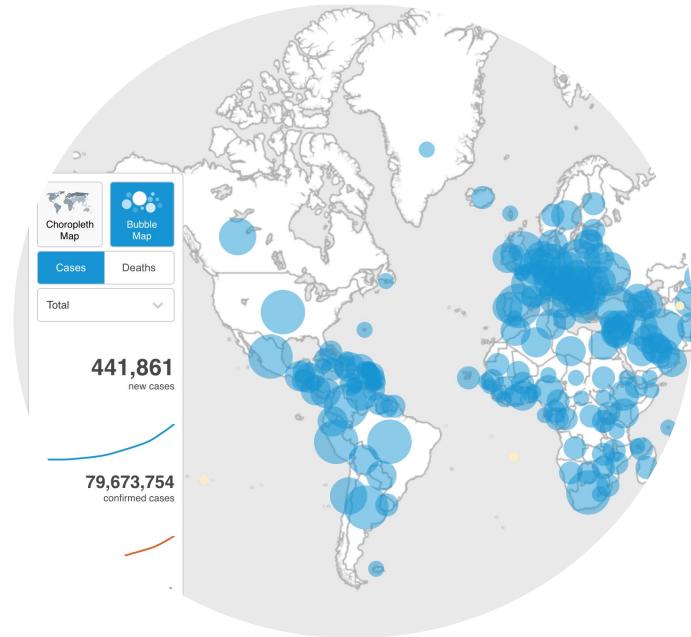
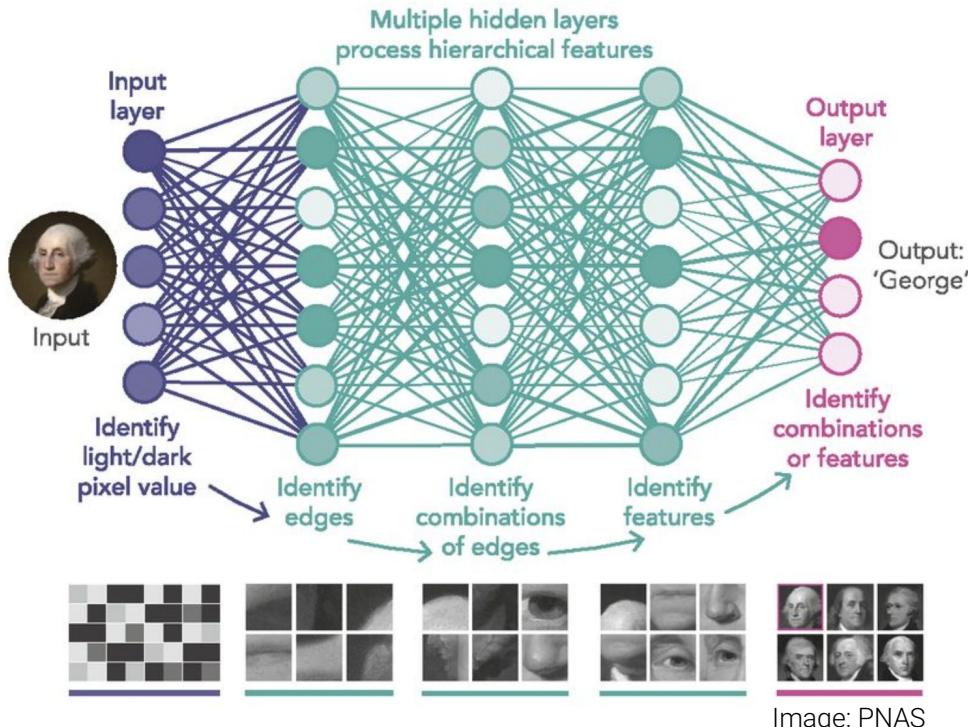


Image: WHO

Why do we build models?



To make accurate predictions about unseen data.

- Can we predict if this email is spam or not? (Project 2!)
- Can we generate a one-sentence summary of this 10-page long article?

Other times, we care more about making extremely accurate predictions, at the cost of having an uninterpretable model. These are sometimes called **black-box** models, and are common in fields like *deep learning*.

Most of the time, we try to strike a balance between interpretability and accuracy.



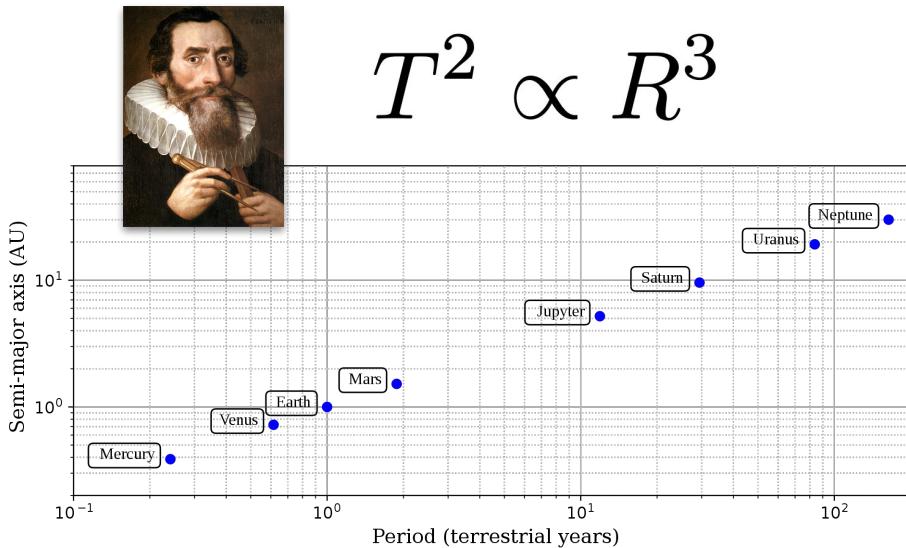
Image: SeeFood
Technologies, Inc.

From HBO's *Silicon Valley* – hot dog or not hot dog? Behind this app is indeed a model.

Physical (or mechanistic) models

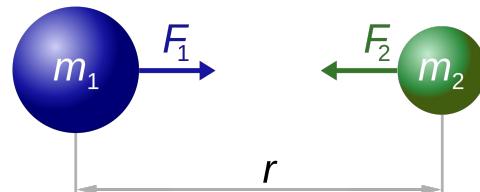
Some models, such as the aforementioned model of the acceleration due to gravity on Earth, are laws that govern how the world works. We call these physical models.

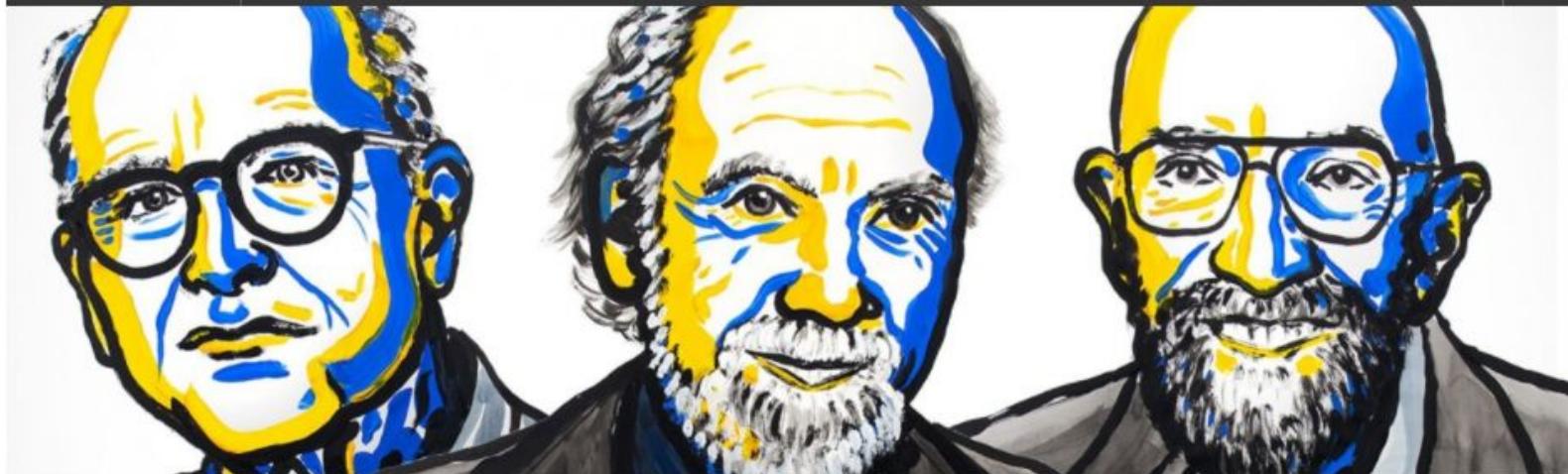
Kepler's Third Law of Planetary Motion (1619)



Newton's Laws: motion and gravitation (1687)

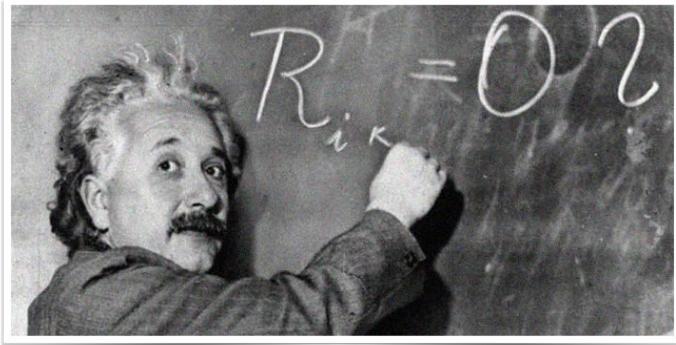
$$\mathbf{F} = m\mathbf{a}$$
$$F = G \frac{m_1 m_2}{r^2}$$





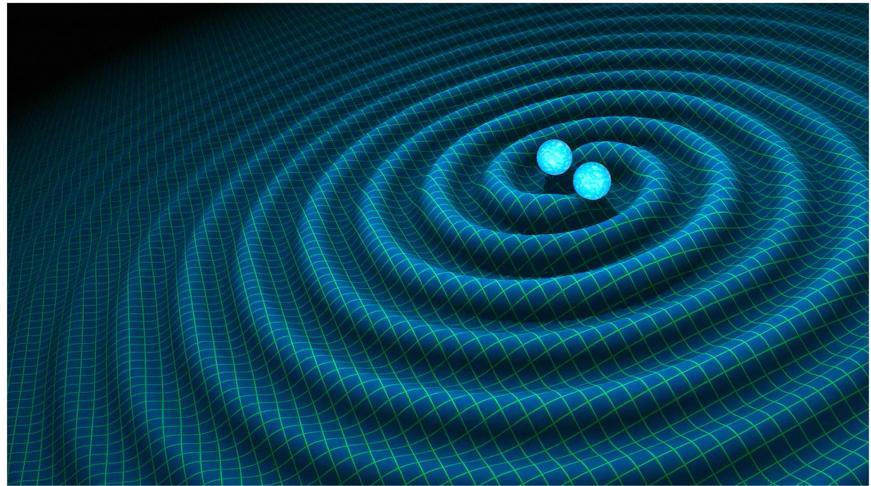
2017 NOBEL PRIZE IN PHYSICS AWARDED FOR DISCOVERY
OF GRAVITATIONAL WAVES

A long time ago in a galaxy far, far away...



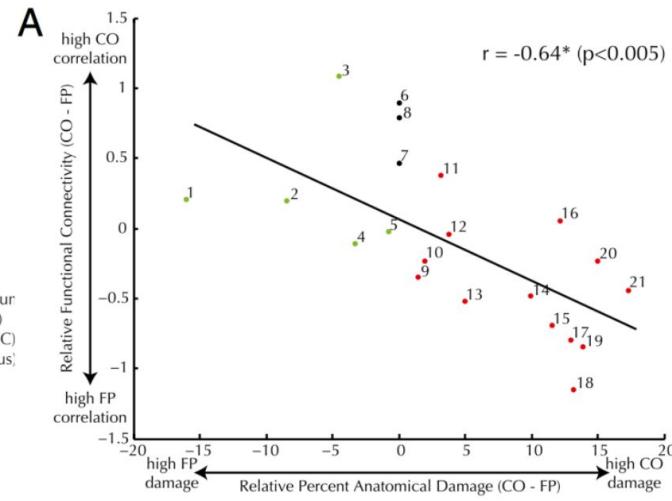
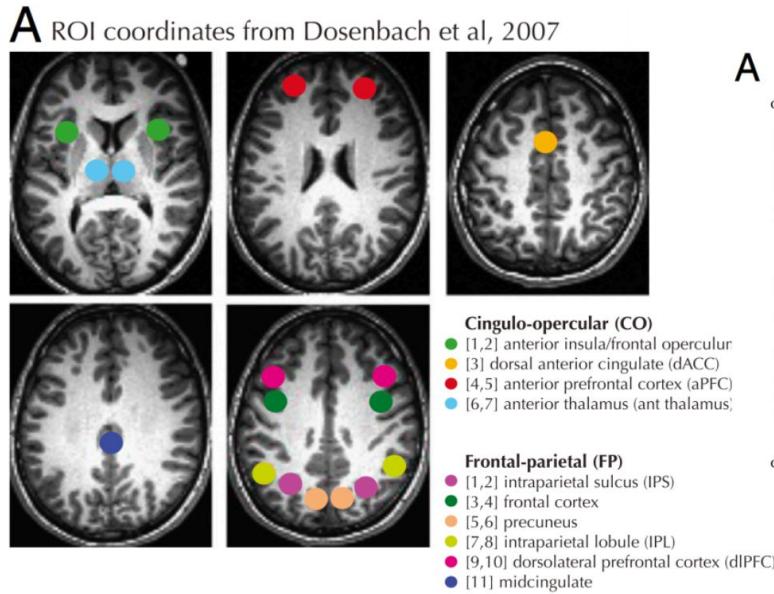
$$R_{\mu\nu} - \frac{1}{2}R g_{\mu\nu} + \Lambda g_{\mu\nu} = \frac{8\pi G}{c^4} T_{\mu\nu}$$

Einstein's Field Equations of General
Relativity
Annalen der Physik, 1916



Statistical models

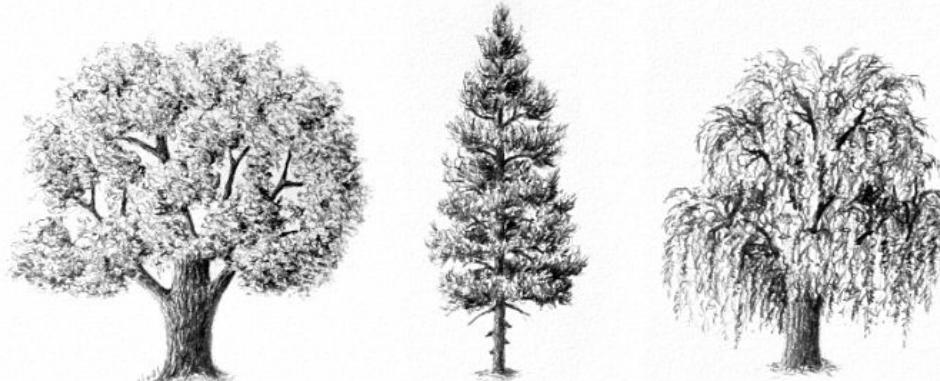
Other times, we don't have such a precise understanding of some natural relationship. In such cases, we collect data and use statistical tools to learn more about the relationships between variables.



What is a tree

- Show a picture to a three-year-old and ask if there is a tree
 - Probably get the correct answer
- Ask a twenty-year-old what the definition of a tree is
 - Probably get an inconclusive answer

(확고한 결정·결과에 이를 정도로) 결정적이 아닌, 결론에 이르지 못하는



Learning from data

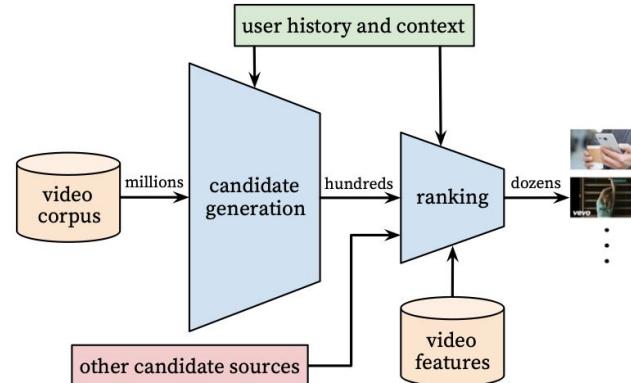
- We learned what a tree is from '**data**'
 - Not by studying the mathematical definition of trees
 - But by looking at trees
- Learning from data is used when
 - We have no analytic solution
 - But have data to construct an empirical solution
- The premise covers a lot of territory.

Example: Recommendation systems

- Good recommender systems are important
 - 20% sales are from recommendation (Amazon.com)
 - 10% improvement = 1 million dollar prize (Netflix)
 - Helping 1 billion users discover personalized content (Youtube.com)



Covington et al., “Deep Neural Networks for YouTube Recommendations,” RecSys, 2016.

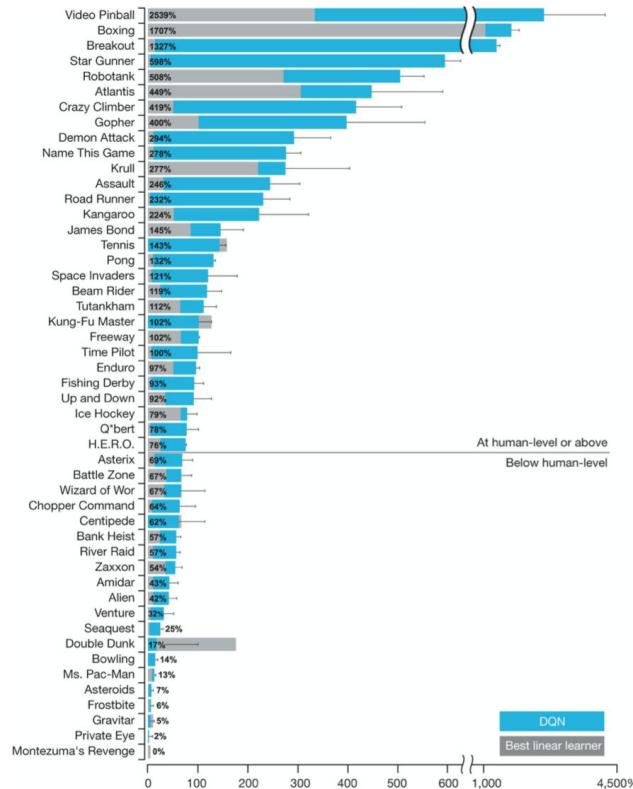
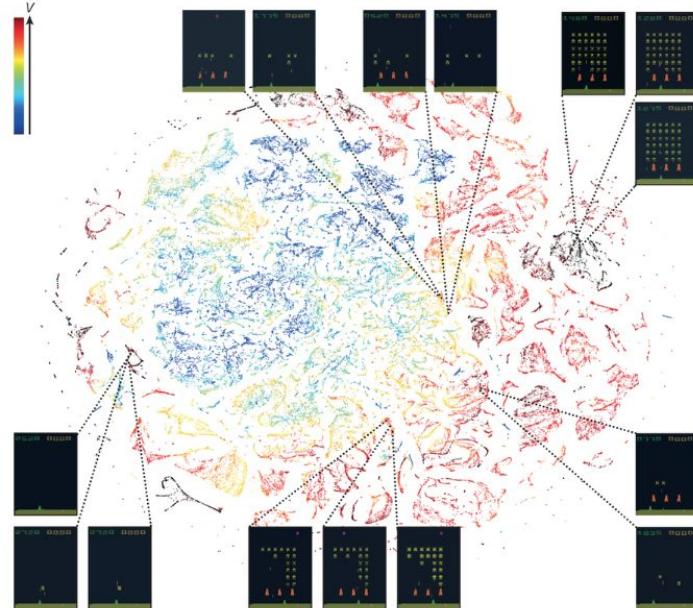


Why Learning from Data

- The main difficulty: an analytic solution is hard to get
 - The criteria users choose to view video clips: quite complex
 - Modeling those explicitly: not easy
- However, historical data reveal a lot about how people choose videos
- The power of learning from data
 - The entire process can be automated without any need for analyzing video content or viewer taste

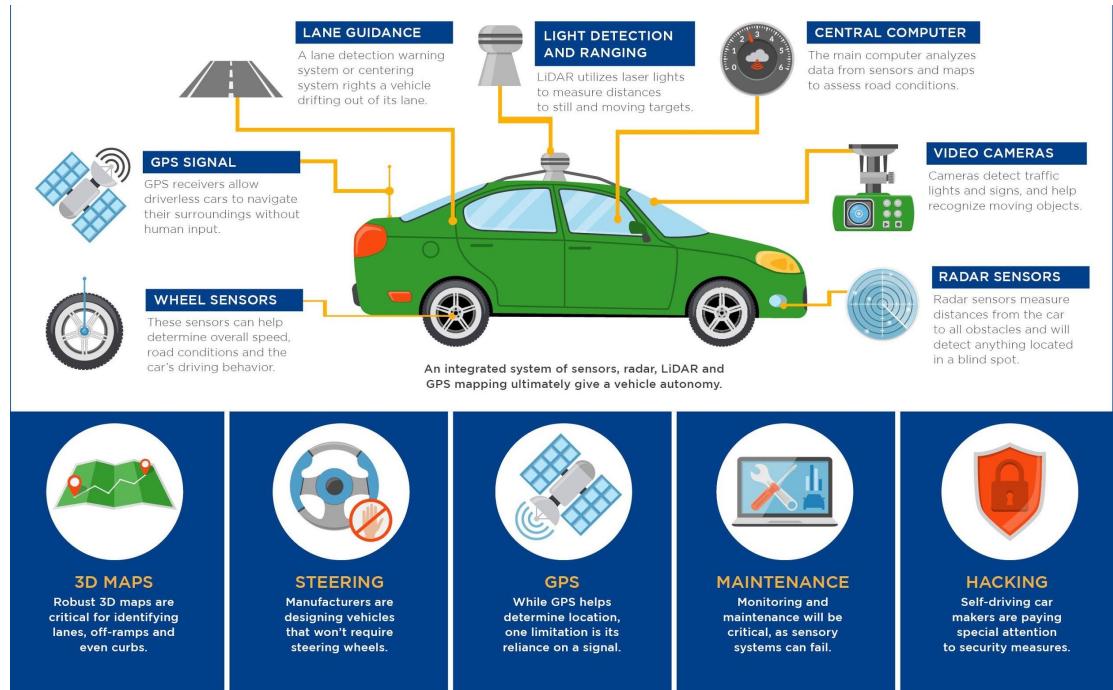
Example: human-level control (game play)

- Deep reinforcement learning
 - By Google (2015)



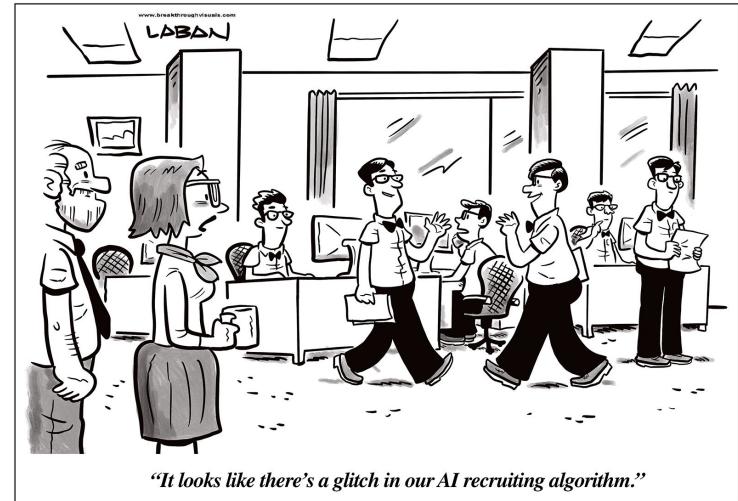
Example: Autonomous Cars

- Waymo ONE (2019)



The essence of learning from data

- We have data
- A pattern exists therein
- We cannot pin it down analytically



Case Study: Tips

A simple model – the constant model

Suppose you want to build a model to predict some numerical quantity of a population:

- The percentage tip given at restaurants.
- The GPA of students at Korea University.



A simple model – the constant model

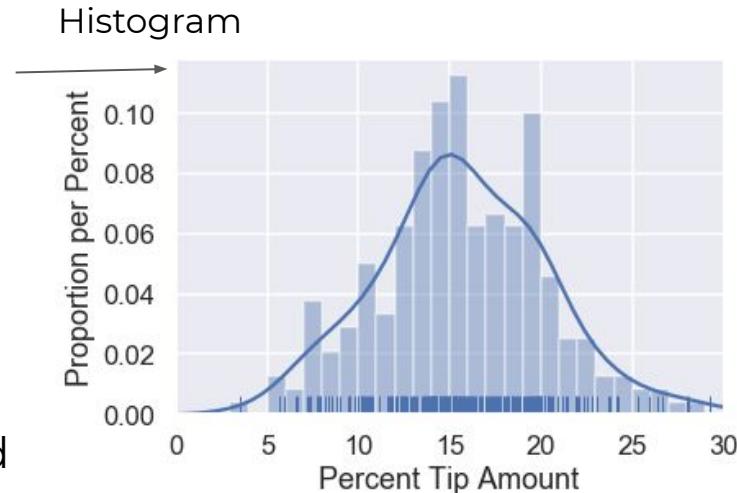
One choice of model would be to ignore any relationships between variables, and predict the same number for each individual – i.e., predicting a constant.

- We call this a summary statistic because it summarizes the data in our sample.
- For instance, tips given at restaurants likely depend on the total bill price, the time of day, how generous the customers are feeling, etc.
 - Ignoring these factors is a simplifying assumption!



Example – Tips dataset

- For instance, suppose a server **collected data about the tips** that they received over time while working at some restaurant.
- We want to pick a constant that “best models” the tips we’ve seen. It seems like most tips are somewhere around 12-20%.
 - 15% seems like a better guess than 25%.
 - But is 15% a better guess than 14%? Hard to tell.
 - We need a precise formulation of all of this.



Notation

현재 이해가 잘 안되는 부분

y represents our **true observations** (e.g. the actual observed tip %s).

y_i represents the i th observation in particular (e.g. y_4 ; the 4th observed tip).

In general, we represent our collected data as y_1, y_2, \dots, y_n

\hat{y} represents the **predicted observations** given by our model (e.g. the predicted tip %s).

\hat{y}_i represents the i th prediction in particular (e.g. \hat{y}_9 is the predicted tip for the 9th data point).

θ represents the **parameter(s) of our model**. This is what we are trying to **estimate!**

Parameters are what define our model. We make this more clear in the next slide.

모델을 통해서 만들고 싶어하는 것

$\hat{\theta}$ represents the **fitted, or optimal, parameter(s)** that we solve for. It is our goal to find this!

We want to find $\hat{\theta}$ to make the best possible model.

모델을 통해서 만든 것 : 학습을 통해서 만든 것

The Constant Model

The constant model can be stated as follows:

$$\hat{y} = \theta$$

- **Parameters are what define our model.** Parameters tell us what the relationships between our input and output variables are.
 - Note: **not all models have parameters.** KDEs are non-parametric models!
- Our model only has one parameter. Here, the only thing that defines our model is the single number we will predict, regardless of the input.
- Our goal is to find the best possible value of our parameter, which we denote with $\hat{\theta}$.
 - We know that $\hat{\theta}$ for the tips dataset is closer to 15% than it is to 25%.

Prediction vs. Estimation

These terms are often used somewhat interchangeably, but there is a subtle difference between them.

Estimation is the task of using data to determine model parameters.

Prediction is the task of using a model to predict outputs for unseen data. Once we have estimates for our model's parameters, we can use our model for prediction.

Loss functions

Loss: The cost of making predictions

We need some metric of how “good” or “bad” our predictions are. This is what loss functions provide us with. **Loss functions quantify how bad a prediction is for a single observation.**

- If our prediction is **close** to the actual value, we want **low loss**.
- If our prediction is **far** from the actual value, we want **high loss**.

A natural choice of loss function is **actual - predicted**, or $y_i - \hat{y}_i$. We call this the **error** for a single prediction.

- But, this treats “negative” predictions and “positive” predictions differently.
 - Predicting 16 when the true value is 15 should be penalized the same as predicting 14.
- This leads to two natural loss functions.

Squared loss

The most common loss function you'll see is the **squared loss**, also known as L2 loss.

$$L_2(y, \hat{y}) = (y - \hat{y})^2$$

- For a single data point in general, this is $(y_i - \hat{y}_i)^2$.
- For our constant model, since $\hat{y} = \theta$, this is $(y_i - \theta)^2$.

If our prediction is equal to the actual observation, our **loss is 0**.

Low loss means a good fit!

Absolute loss

Another common loss function is the **absolute loss**, also known as L1 loss.

$$L_1(y, \hat{y}) = |y - \hat{y}|$$

- For our constant model, for a single point, this is $|y_i - \theta|$

There are benefits and drawbacks to both of the above loss functions. We will examine those shortly. **These are also not the only possible loss functions; we will see more later.**

Loss functions and Empirical risk

We care about how bad our model's predictions are for our entire data set, not just for one point. A natural measure, then, is of the **average loss across all points**. Assuming n points:

$$\frac{1}{n} \sum_{i=1}^n L(y_i, \hat{y}_i)$$

Other names for **average loss** include **empirical risk** and an **objective function**.

The average loss of a model tells us how well it fits the given data. If our model has a low average loss across our dataset, that means it is good at making predictions. As such, **we want to find the parameter(s) that minimize average loss**, in order to make our model as good at making predictions as it can be.

MSE and MAE

둘다 써도 상관은 없다.

근데 이걸 사용해서 모델을 만들었을 때, parameter의 성격이 다르다

If we choose squared loss as our loss function, then average squared loss is typically referred to as **mean squared error (MSE)**, and is of the following form:

weight의 절대 값이 작게 하는 결과값을 반환

$$MSE(y, \hat{y}) = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2$$

If we choose absolute loss as our loss function, then average absolute loss is typically referred to as **mean absolute error (MAE)**, and is of the following form:

D demension의 모든 값이
아닌 필요한 값만 사용한다
sparse한 solution

$$MAE(y, \hat{y}) = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i|$$

These definitions hold true, regardless of our model. We want to **minimize** these quantities.

Exploring MSE

여기서는 상수인데,
theta가 dimension이 있을 수 있다.

Average loss is typically written as a function of θ , since θ defines what our model is (and hence what our predictions are). For example, with squared loss and the constant model, our average loss (and hence, the function we want to minimize) is

$$R(\theta) = \frac{1}{n} \sum_{i=1}^n (y_i - \theta)^2$$

Risk : 가지고 있는 모든 데이터에 대한 합 $\hat{\theta}$

Mathematically, our goal of finding the optimal $\hat{\theta}$ is stated as:

argmin means "the argument that minimizes the following function."

$$\hat{\theta} = \arg \min_{\theta} \frac{1}{n} \sum_{i=1}^n (y_i - \theta)^2$$

We won't use this notation again in this lecture, but it will come up again in the future.

Exploring MSE

When our model is the constant model, and we choose to use L2 loss, again, our average loss looks like:

$$R(\theta) = \frac{1}{n} \sum_{i=1}^n (y_i - \theta)^2$$

Let's examine a toy example. Suppose we have 5 observations, [20, 21, 22, 29, 33].

$$L_2(20, \theta) = (20 - \theta)^2$$

The loss for the first observation (y_1).

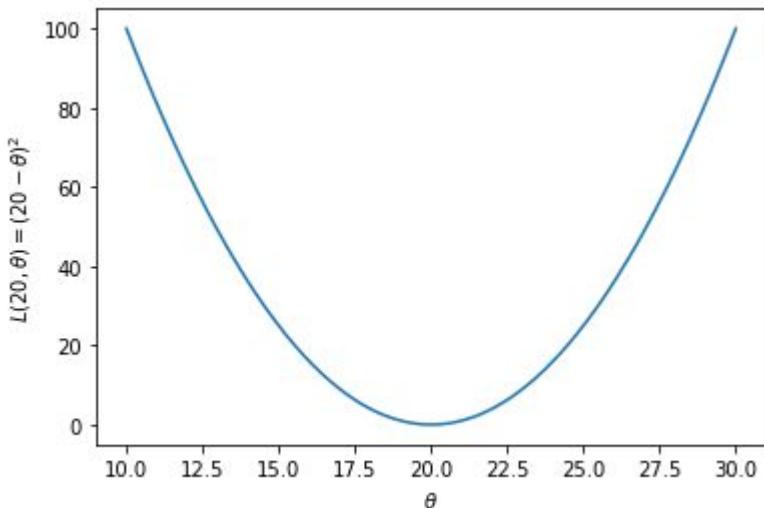
$$R(\theta) = \frac{1}{5} ((20 - \theta)^2 + (21 - \theta)^2 + (22 - \theta)^2 + (29 - \theta)^2 + (33 - \theta)^2)$$

The average loss across all observations (the MSE).

Exploring MSE

$$L_2(20, \theta) = (20 - \theta)^2$$

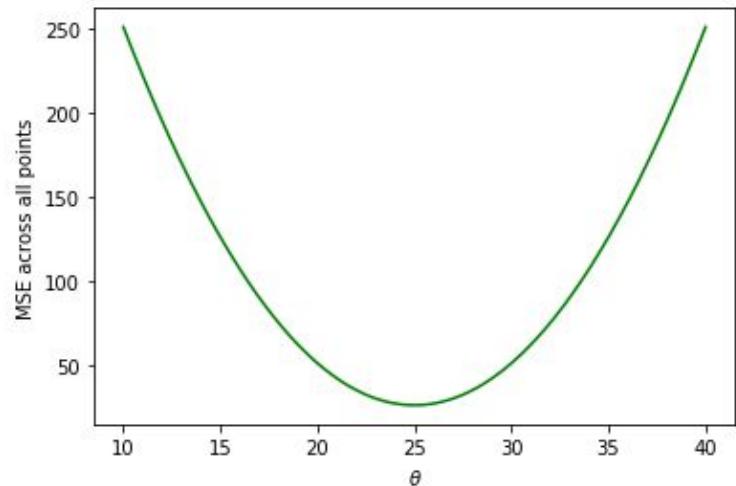
The loss for the first observation (y_1).



A parabola, minimized at theta = 20.

$$R(\theta) = \frac{1}{5} ((20 - \theta)^2 + (21 - \theta)^2 + (22 - \theta)^2 + (29 - \theta)^2 + (33 - \theta)^2)$$

The average loss across all observations (the MSE).



Also a parabola! Minimized at theta = 25.

Minimizing Mean Squared Error (MSE)

Minimizing MSE

We saw with the toy example of [20, 21, 22, 29, 33] that the value that minimizes the MSE of the constant model was 25, which was the **mean of our observations**.

We can try other examples if we want to, and we'll end up with the same result. Let's instead pivot to proving this rigorously, using mathematics. There are two ways we'll go about doing this:

- Using calculus.
- Using a neat algebraic trick.

For both derivations, the slides contain the key ideas, but the lecture videos will contain a step-by-step walkthrough.

MSE minimization using calculus

One way to minimize a function is by using calculus: we can take the derivative, set it equal to 0, and solve for the optimizing value.

- The derivative of the sum of several pieces is equal to the sum of the derivative of said pieces.
- The derivative of the loss for a single point is $\frac{d}{d\theta}(y_i - \theta)^2 = 2(y_i - \theta)(-1) = -2(y_i - \theta)$.

Then:

$$\begin{aligned} R(\theta) &= \frac{1}{n} \sum_{i=1}^n (y_i - \theta)^2 \\ \implies \frac{d}{d\theta} R(\theta) &= \frac{1}{n} \sum_{i=1}^n \frac{d}{d\theta} (y_i - \theta)^2 = \frac{1}{n} \sum_{i=1}^n (-2)(y_i - \theta) = \frac{-2}{n} \sum_{i=1}^n (y_i - \theta) \end{aligned}$$

MSE minimization using calculus

$$R(\theta) = \frac{1}{n} \sum_{i=1}^n (y_i - \theta)^2$$
$$\implies \frac{d}{d\theta} R(\theta) = \frac{1}{n} \sum_{i=1}^n \frac{d}{d\theta} (y_i - \theta)^2 = \frac{1}{n} \sum_{i=1}^n (-2)(y_i - \theta) = \frac{-2}{n} \sum_{i=1}^n (y_i - \theta)$$

Setting this term to 0, we have:

$$0 = \frac{-2}{n} \sum_{i=1}^n (y_i - \theta)$$

we can separate sums

$$0 = \sum_{i=1}^n (y_i - \theta) = \sum_{i=1}^n y_i - \sum_{i=1}^n \theta = \sum_{i=1}^n y_i - n\theta$$
$$n\theta = \sum_{i=1}^n y_i$$

$c + c + \dots + c = n * c$

$$\implies \hat{\theta} = \frac{\sum_{i=1}^n y_i}{n} = \mathbf{mean}(y)$$

Thus, with squared loss and the constant model, the sample mean minimizes MSE.

MSE minimization using calculus

$$\implies \hat{\theta} = \frac{\sum_{i=1}^n y_i}{n} = \text{mean}(y)$$

We're not done yet! To be thorough, we need to perform the second derivative test, to guarantee that the point we found is truly a **minimum** (rather than a maximum or saddle point). We hope that the second derivative of our objective function is positive, indicating our function is convex opening upwards.

$$\frac{d}{d\theta} R(\theta) = \frac{-2}{n} \sum_{i=1}^n (y_i - \theta)$$

$$\frac{d^2}{d\theta^2} R(\theta) = \frac{-2}{n} \sum_{i=1}^n (0 - 1) = \frac{2}{n} \sum_{i=1}^n 1 = 2$$

Fortunately, it is, so the sample mean truly is the minimizer we were looking for.

We will interpret what this means shortly.

MSE minimization using an algebraic trick

It turns out that in this case, there's another rather elegant way of performing the same minimization algebraically, but without using calculus.

- In this proof, you will need to use the fact that the **sum of deviations from the mean is 0** (in other words, that $\sum_{i=1}^n (y_i - \bar{y}) = 0$).

$$\begin{aligned}\sum_{i=1}^n (y_i - \bar{y}) &= \sum_{i=1}^n y_i - \sum_{i=1}^n \bar{y} \\ &= \sum_{i=1}^n y_i - n\bar{y} = \sum_{i=1}^n y_i - n \cdot \frac{1}{n} \sum_{i=1}^n y_i = \sum_{i=1}^n y_i - \sum_{i=1}^n y_i \\ &= 0\end{aligned}$$

- Our proof will also use the definition of the variance of a sample. As a refresher:

$$\sigma_y^2 = \frac{1}{n} \sum_{i=1}^n (y_i - \bar{y})^2$$

Equal to the MSE of the sample mean!

MSE minimization using an algebraic trick

$$\begin{aligned} R(\theta) &= \frac{1}{n} \sum_{i=1}^n (y_i - \theta)^2 \\ &= \frac{1}{n} \sum_{i=1}^n [(y_i - \bar{y}) + (\bar{y} - \theta)]^2 \\ &= \frac{1}{n} \sum_{i=1}^n [(y_i - \bar{y})^2 + 2(y_i - \bar{y})(\bar{y} - \theta) + (\bar{y} - \theta)^2] \\ &= \frac{1}{n} \left[\sum_{i=1}^n (y_i - \bar{y})^2 + 2(\bar{y} - \theta) \sum_{i=1}^n (y_i - \bar{y}) + n(\bar{y} - \theta)^2 \right] \\ &= \frac{1}{n} \sum_{i=1}^n (y_i - \bar{y})^2 + \frac{2}{n} (\bar{y} - \theta) \cdot 0 + (\bar{y} - \theta)^2 \\ &= \sigma_y^2 + (\bar{y} - \theta)^2 \end{aligned}$$

variance of sample!

from the previous slide

This proof relies on an algebraic trick. We can write the difference $\mathbf{a} - \mathbf{b}$ as $(\mathbf{a} - \mathbf{c}) + (\mathbf{c} - \mathbf{b})$, where a, b, and c are any numbers.

Using that fact, we can write $y_i - \theta = (y_i - \bar{y}) + (\bar{y} - \theta)$, where $\bar{y} = \frac{1}{n} \sum_{i=1}^n y_i$, our sample mean.

Also note: going from line 3 to 4, we distribute the sum to the individual terms. This is a property of sums you should become familiar with!

Minimization using an algebraic trick

In the previous slide, we showed that $R(\theta) = \sigma_y^2 + (\bar{y} - \theta)^2$.

- Since variance can't be negative, the first term is greater than or equal to 0.
 - Of note, **the first term doesn't involve θ at all.** Changing our model won't change this value, so for the purposes of determining $\hat{\theta}$, we can ignore it.
- The second term is being squared, and so also must be greater than or equal to 0.
 - This term does involve θ , and so picking the right value of θ will minimize our average loss.
 - We need to pick the θ that sets the second term to 0.
 - This is achieved when $\theta = \bar{y}$. In other words:

$$\hat{\theta} = \bar{y} = \mathbf{mean}(y)$$

Mean minimizes MSE for the constant model

As we determined a variety of ways, for the constant model with squared loss, the mean of the dataset is the optimal model.

$$\hat{\theta} = \mathbf{mean}(y) = \bar{y}$$

- This holds true **regardless of the dataset** we use, but it's only true for **this combination of model and loss**.
- If we choose any other constant other than the sample mean, the empirical risk will not be as small as possible, and so our model is “worse” (for this loss).

This is not all that surprising! **It provides some formal reasoning as to why we use means so commonly as summary statistics.** It is the best, in some sense.

Note, we now write $\hat{\theta}$ instead of θ . This is because we are referring to the **optimal parameter**, not just any arbitrary θ .

Minimum value of MSE is the sample variance

It's worth noting that when we substitute $\theta = \bar{y}$ back into our average loss, we obtain a familiar result:

$$R(\bar{y}) = \frac{1}{n} \sum_{i=1}^n (y_i - \bar{y})^2 = \sigma_y^2$$

That is, the **minimum value** that mean squared error can take on (again, for the constant model) **is the sample variance**.

Put another way, the following statement is true whenever $\theta \neq \bar{y}$:

$$\frac{1}{n} \sum_{i=1}^n (y_i - \bar{y})^2 < \frac{1}{n} \sum_{i=1}^n (y_i - \theta)^2$$

Minimizing Mean Absolute Error (MAE)

Exploring MAE

When we use absolute (or L1) loss, we call the average loss **mean absolute error**. For the constant model, our MAE looks like:

$$R(\theta) = \frac{1}{n} \sum_{i=1}^n |y_i - \theta|$$

Let's again re-visit our toy example of 5 observations, [20, 21, 22, 29, 33].

$$L_1(20, \theta) = |20 - \theta|$$

The loss for the first observation (y_1).

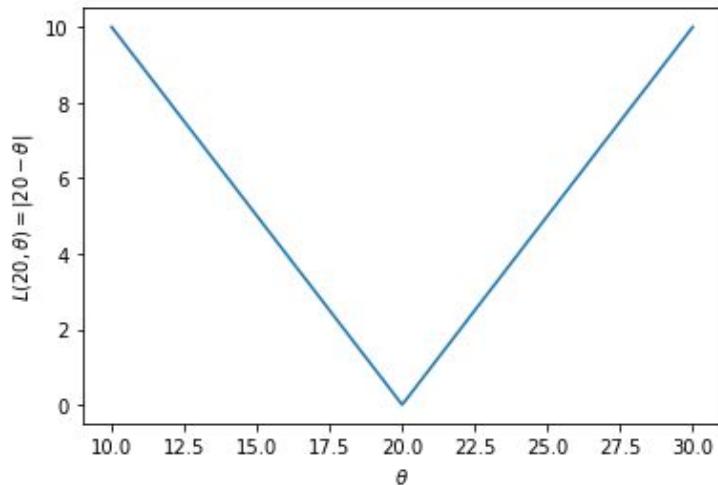
$$R(\theta) = \frac{1}{5}(|20 - \theta| + |21 - \theta| + |22 - \theta| + |29 - \theta| + |33 - \theta|)$$

The average loss across all observations (the MAE).

Exploring MAE

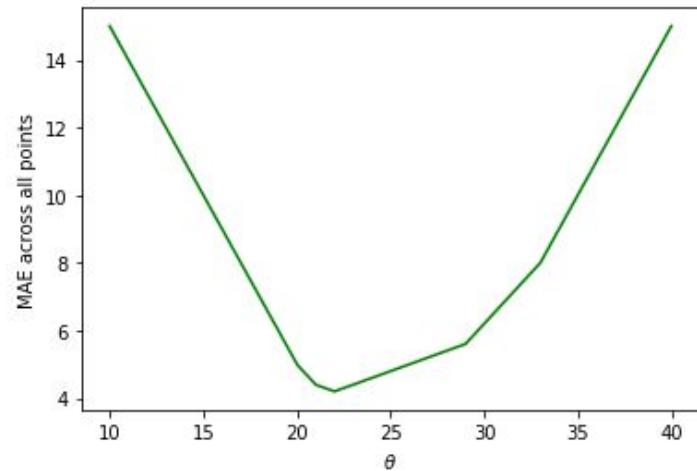
$$L_1(20, \theta) = |20 - \theta|$$

The loss for the first observation (y_1).



$$R(\theta) = \frac{1}{5}(|20 - \theta| + |21 - \theta| + |22 - \theta| + |29 - \theta| + |33 - \theta|)$$

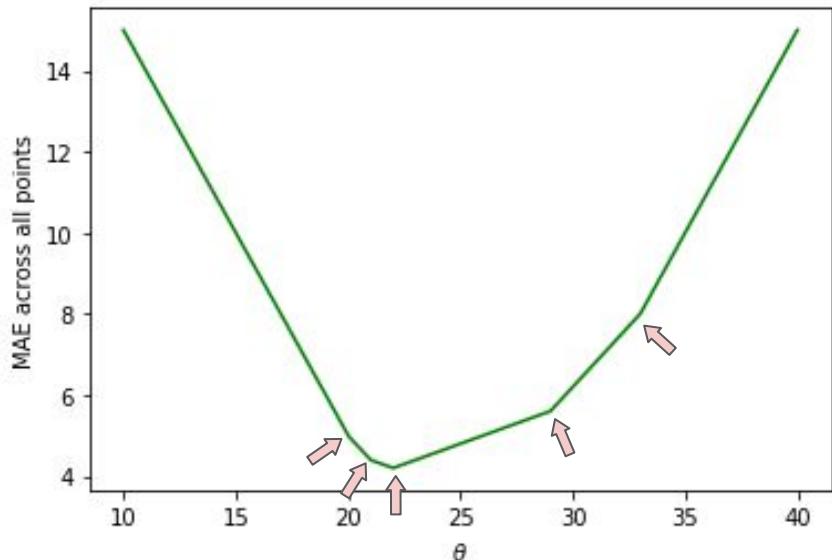
The average loss across all observations (the MAE).



An absolute value curve, centered at theta = 20.

Some weird shape.... minimized near theta = 22?

Exploring MAE



The bends, or “kinks,” all appear at our observations! (20, 21, 22, 29, 33)

The shape of the MAE with the constant model seems to be jagged. This is because it is the (weighted) sum of several absolute value curves, which results in a **piecewise linear function**.

It also doesn't seem to be immediately clear what the optimal choice of theta should be. It's somewhere in the “middle” of our points, but it's **clearly not 25**, which was the minimizing value for the MSE.

Let's once again resort to calculus!

MAE minimization using calculus

Once again, we can use calculus to determine the $\hat{\theta}$ optimal .

The first step is to determine the derivative of our loss function for a single point. Absolute value functions can be written as two piecewise linear functions:

$$|y_i - \theta| = \begin{cases} y_i - \theta & \text{if } \theta \leq y_i \\ \theta - y_i & \text{if } \theta > y_i \end{cases}$$

The derivative of our loss for a single point, then, is also a piecewise linear function:

$$\frac{d}{d\theta} |y_i - \theta| = \begin{cases} -1 & \text{if } \theta < y_i \\ 1 & \text{if } \theta > y_i \end{cases}$$

Note: The derivative of the absolute value when the argument is 0 (i.e. when $y_i = \theta$) is technically undefined. We ignore this case in our derivation, since thankfully, it doesn't change our result.

MAE minimization using calculus

$$\frac{d}{d\theta} |y_i - \theta| = \begin{cases} -1 & \text{if } \theta < y_i \\ 1 & \text{if } \theta > y_i \end{cases}$$

From here, we again use the fact that the derivative of a sum is a sum of derivatives:

$$\begin{aligned}\frac{d}{d\theta} R(\theta) &= \frac{1}{n} \sum_{i=1}^n \frac{d}{d\theta} |y_i - \theta| \\ &= \frac{1}{n} \left[\sum_{\theta < y_i} (-1) + \sum_{\theta > y_i} 1 \right]\end{aligned}$$

Add -1 for each time an observation y_i is greater than our choice of theta.

Add 1 for each time an observation y_i is less than our choice of theta.

MAE minimization using calculus

Setting this derivative equal to 0:

$$0 = \frac{1}{n} \left[\sum_{\theta < y_i} (-1) + \sum_{\theta > y_i} 1 \right]$$

$$0 = - \sum_{\theta < y_i} 1 + \sum_{\theta > y_i} 1$$

$$\sum_{\theta < y_i} 1 = \sum_{\theta > y_i} 1$$

The last line is telling us that in order for our MAE to be minimized, we need to choose a theta such that **the number of observations less than theta** needs to be equal to **the number of observations greater than theta**.

MAE minimization using calculus

$$\sum_{\theta < y_i} 1 = \sum_{\theta > y_i} 1$$

In order for our MAE to be minimized, we need to choose a theta such that the number of observations less than theta needs to be equal to the number of observations greater than theta. In other words, theta needs to be such that there are **an equal number of points to the left and right.**

This is the definition of the median! For example, in our toy dataset, the point below in **red (22)** is the median of our observations. It is the value in the “middle.”

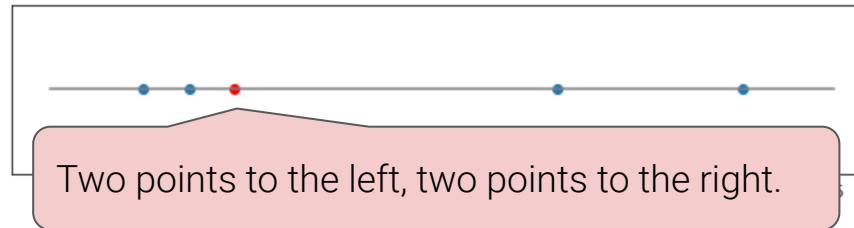


MAE minimization using calculus

$$\sum_{\theta < y_i} 1 = \sum_{\theta > y_i} 1$$

In order for our MAE to be minimized, we need to choose a theta such that the number of observations less than theta needs to be equal to the number of observations greater than theta. In other words, theta needs to be such that there are **an equal number of points to the left and right.**

This is the definition of the median! For example, in our toy dataset, the point below in **red (22)** is the median of our observations. It is the value in the “middle.”

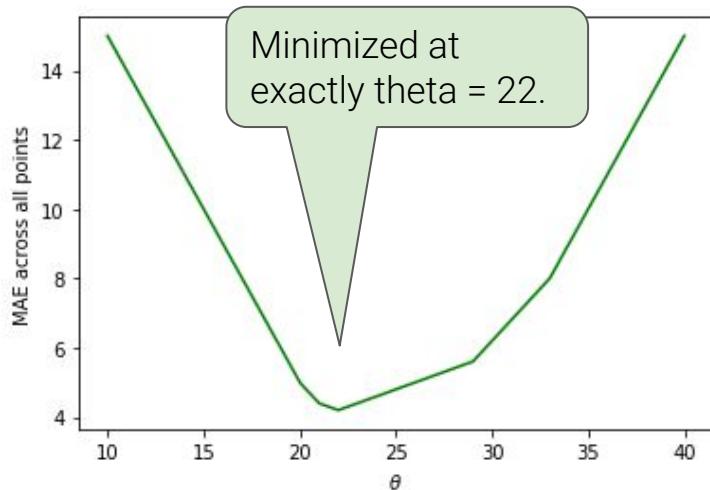


Median minimizes MAE for the constant model

We've now shown that the median minimizes MAE for the constant model.

$$\hat{\theta} = \text{median}(y)$$

This is consistent with what we saw earlier, when plotting the MAE for our toy dataset:



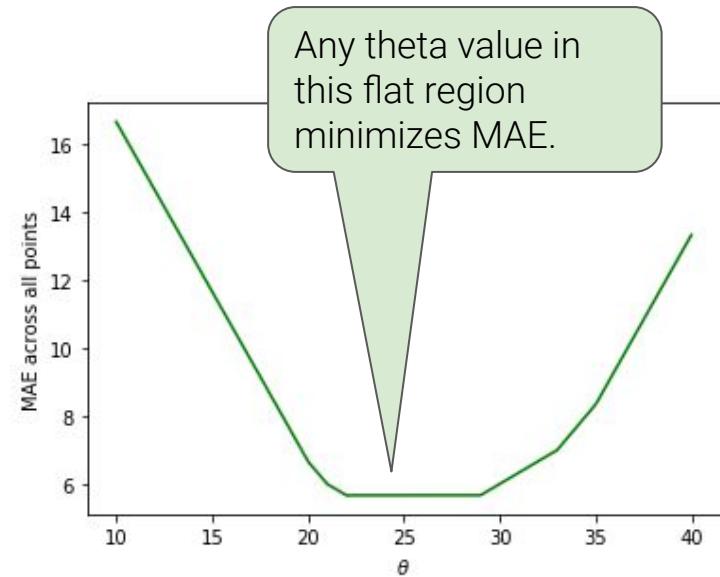
Important note: In general, the mean and median of a dataset are not the same. Therefore, using MSE and MAE give us **different optimal theta values!**

A key takeaway here is that our choice of loss function determines the optimal parameters for our model.

Median minimizes MAE for the constant model

Our toy dataset only had 5 observations. What if it had an even number of observations? Let's say our toy dataset is now [20, 21, 22, 29, 33, 35]. The 35 is new.

- **There's no longer a unique solution!**
- Any value in the range [22, 29] minimizes MAE.
- This reflects the fact that there are an even number of observations, and any number in that range has the same number of points to the left and right.
- (When there are an even number of data points, we typically set the median to be the mean of the two middle ones. Here, that'd be 25.5)

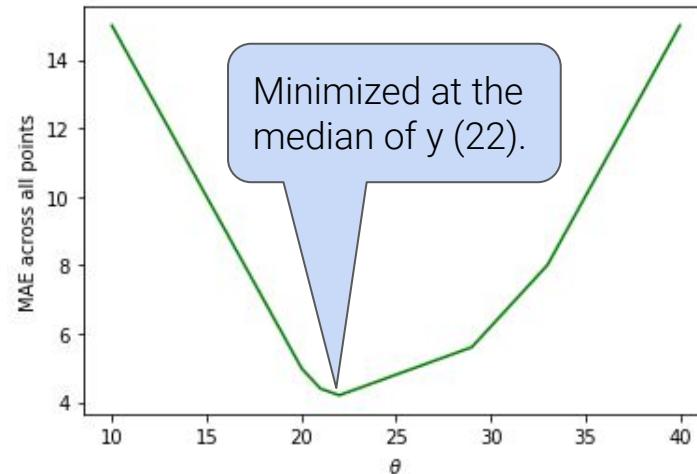
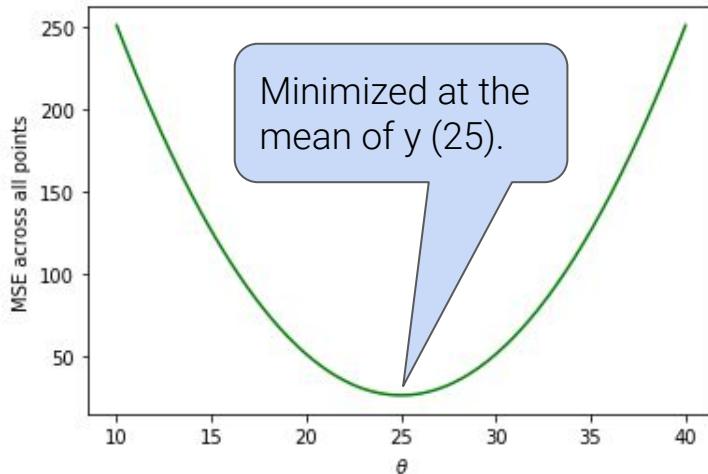


MSE vs. MAE

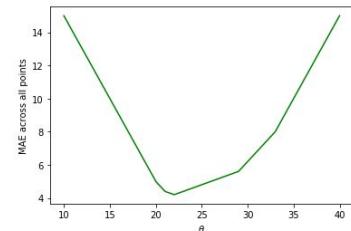
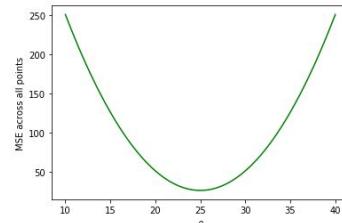
MSE vs. MAE for toy data

Below, we present the plot of the **loss surface** for our toy dataset, using L2 loss (left) and L1 loss (right).

- A **loss surface** is a plot of the loss encountered for each possible value of θ .
- If our model had 2 parameters, this plot would be 3 dimensional.



MSE vs. MAE



What else is different about squared loss (MSE) and absolute loss (MAE)?

Mean squared error (optimal parameter for the constant model is the [sample mean](#))

- **Very smooth.** Easy to minimize using numerical methods (coming later in the course).
- **Very sensitive to outliers,** e.g. if we added 1000 to our largest observation, the optimal theta would become 225 instead of 25.

Mean absolute error (optimal parameter for the constant model is the [sample median](#))

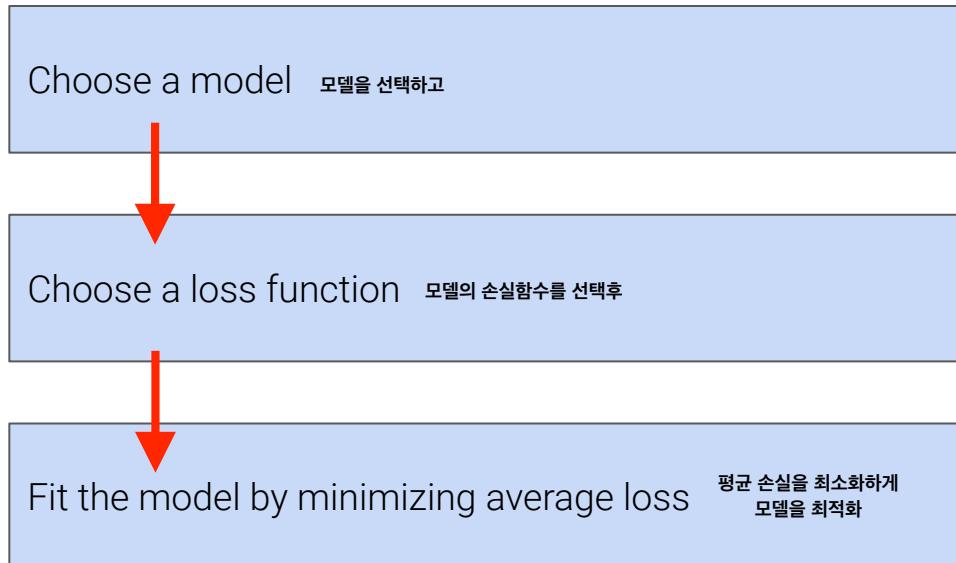
- **Not as smooth** – at each of the “kinks,” it’s not differentiable. Harder to minimize.
- **Robust to outliers!** E.g, adding 1000 to our largest observation doesn’t change the median.

It’s not clear that one is “better” than the other. In practice, **we get to choose our loss function!**

Summary

The modeling process

We've implicitly introduced this three-step process, which we will use constantly throughout the rest of the course.



The modeling process

We've implicitly introduced this three-step process, which we will use constantly throughout the rest of the course.

Choose a model

Choose a loss function

Fit the model by minimizing average loss

In this lecture, we focused exclusively on the **constant model**, which has a single **parameter**.

Parameters define our model. They tell us the relationship between the variables involved in our model. (Not all models have parameters, though!)

In the coming lectures, we will look at more sophisticated models.

The modeling process

We've implicitly introduced this three-step process, which we will use constantly throughout the rest of the course.

Choose a model

Choose a loss function

Fit the model by minimizing average loss

We introduced two loss functions here: L2 (**squared**) loss and L1 (**absolute**) loss. There also exist others.

Both have their benefits and drawbacks. **We get to choose** which loss function we use, for any modeling task.

The modeling process

We've implicitly introduced this three-step process, which we will use constantly throughout the rest of the course.

Choose a model

Choose a loss function

Fit the model by minimizing average loss

Lastly, we choose the **optimal parameters** by determining the parameters that **minimize average loss** across our entire dataset. **Different loss functions lead to different optimal parameters.**

This process is called **fitting the model to the data**. We did it by hand here, but in the future we will rely on computerized techniques.

What's next...

- **Changing the model.**
 - Next, we'll introduce the simple linear regression model.
 - We'll also look at multiple regression, logistic regression, decision trees, and random forests, all of which are different types of models.
- **Changing the loss function.**
 - L2 loss (and, hence, mean squared error) will appear a lot.
 - But we'll also introduce new loss functions, like cross-entropy loss.
- **Changing how we fit the model to the data.**
 - We did this largely by hand in this lecture.
 - But shortly, we'll run into combinations of models and loss functions for which the optimal parameters can't be determined by hand.
 - As such, we'll learn about techniques like gradient descent.

Appendix: Learning paradigms

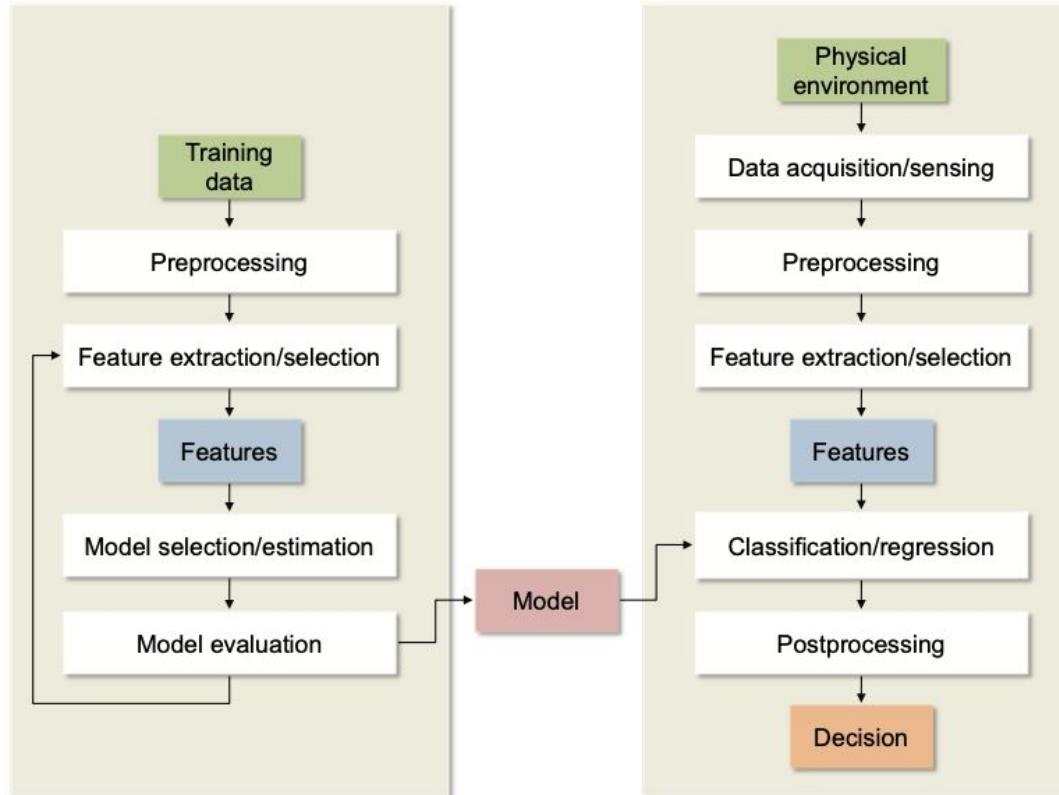
Learning paradigms

- Basic premise of learning from data
 - Use of observations to uncover an underlying process
 - Very broad and difficult to fit into a single framework
- Different learning paradigms have arisen
 - Supervised learning
 - Unsupervised learning
 - Reinforcement learning

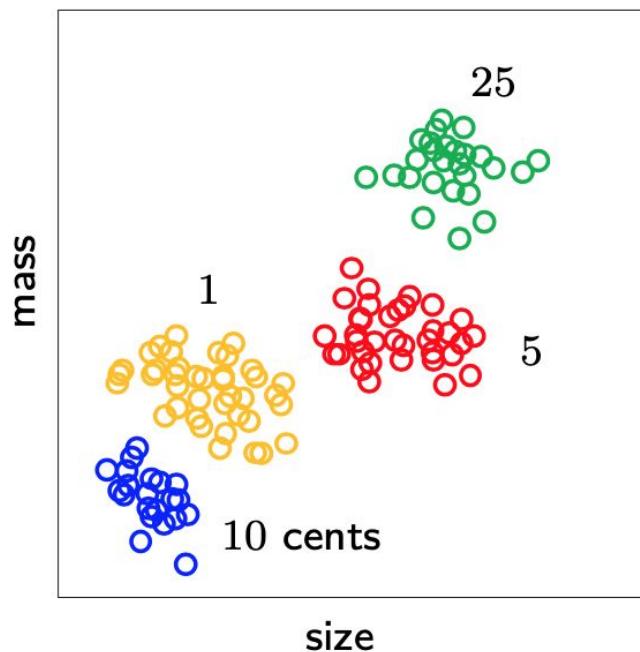
Supervised learning

- The most studied and most utilized type of learning
 - Our main focus for a while
- Supervised learning setting
 - Training data contains explicit examples of what the observations should be for given inputs
- Learning is ‘supervised’ in that
 - Some ‘supervisor’ has taken the trouble to look at each input and determine the correct output
 - The correct ‘label’ is available for each training sample
- Most well-known approaches: classification and clustering

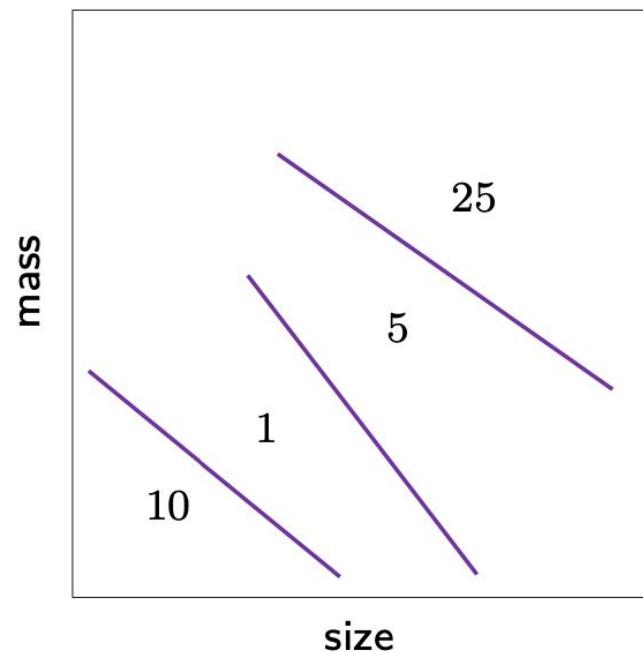
Typical supervised learning procedure



Example from vending machines: coin recognition



(a) coin data

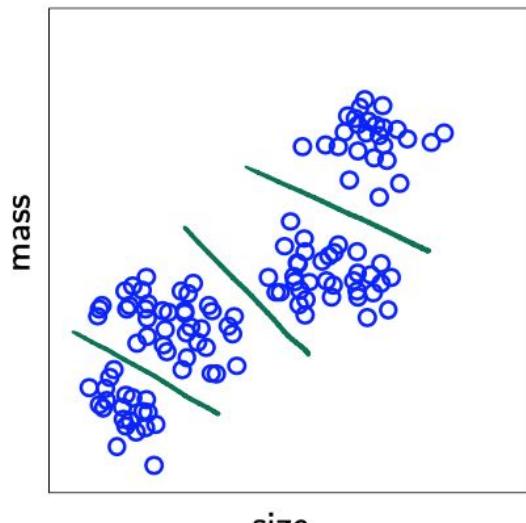


(b) learned classifier

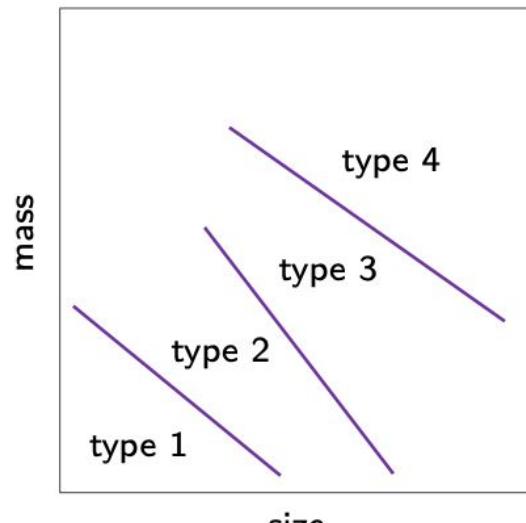
Unsupervised learning

- the training data do not contain any output information at all
 - instead of (input, correct output), we get (input, ?)
 - that is, we are just given input examples
- how could we possibly learn anything from mere inputs?
- approaches to unsupervised learning
 - clustering (k-means, mixture models, hierarchical)
 - hidden Markov models (HMMs)
 - feature extraction (e.g. PCA, ICA, SVD)

Example: coin clustering problem



(a) unlabeled coin data



(b) unsupervised learning

Figure 5: the decision regions in unsupervised learning may be identical to those in supervised learning, but without the labels

Unsupervised learning

- unlabeled data
 - we do not know the denomination of any coin
- however, the correct clustering is less obvious now
 - and even the number of clusters may be ambiguous
- nonetheless, this example shows that
 - we can learn something from the inputs by themselves

Unsupervised learning can be viewed as

- spontaneously finding patterns and structure in input data
 - e.g. categorize a set of books into topics
- a precursor to supervised learning
 - e.g. learning Spanish without knowing the meaning first and then taking Spanish lessons will be easier to follow
- a way to create higher-level representation of the data
 - e.g. automated feature extraction

Reinforcement Learning

- when training data contain no correct output for each input
 - no longer in a supervised learning setting
- e.g. a toddler learning not to touch a hot cup of tea
 - training examples do not say what to do
 - nevertheless, she uses the examples to reinforce better actions
 - eventually she learns what she should do in similar situations

- this characterizes reinforcement learning
 - the training example does not contain the target output
 - but instead contains some possible output together with a measure of how good that output is
- compare how a training example looks:
 - supervised learning:
 - (input, **correct** output)
 - reinforcement learning:
 - (input, **some** output, reward for this output)

- reinforcement learning is especially useful for learning a game

