Feature Selection with LASSO

Dec 02 2024

Giảng viên: TS. Lưu Phúc Lợi
Luu.p.loi@googlemail.com

Content of Lecture 2

1. Norms and Vector Representation
2. Method of Lagrange Multipliers
3. Recap of simple linear regression
4. Feature Selection with LASSO
5. Practice

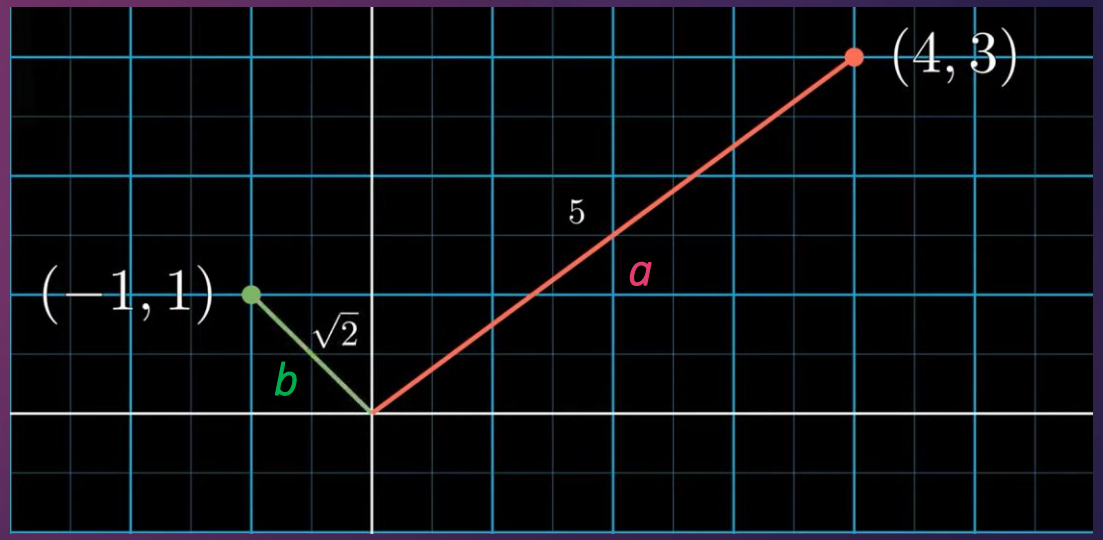
Norms and Vector Representation

Norms

- Vectors are sets of numbers representing features.
- Norms in machine learning quantify the size of vectors.
- Example:
  - L_2 norm of vector a is
    $$\|a\| = \sqrt[2]{4^2 + 3^2} = \sqrt[2]{25} = 5$$
  - L_2 norm of vector b is
    $$\|b\| = \sqrt[2]{(-1)^2 + 1^2} = \sqrt{2}$$
- The L_1 norm, L_2 norm, L_3 norm … L_∞ norm form different geometric shapes as their order increases.

$$\text{features: } X = \{X_1, X_2\}$$

$$a = \begin{bmatrix} x_1 = 4 \\ x_2 = 3 \end{bmatrix} \rightarrow a = \begin{bmatrix} 4 \\ 3 \end{bmatrix}$$



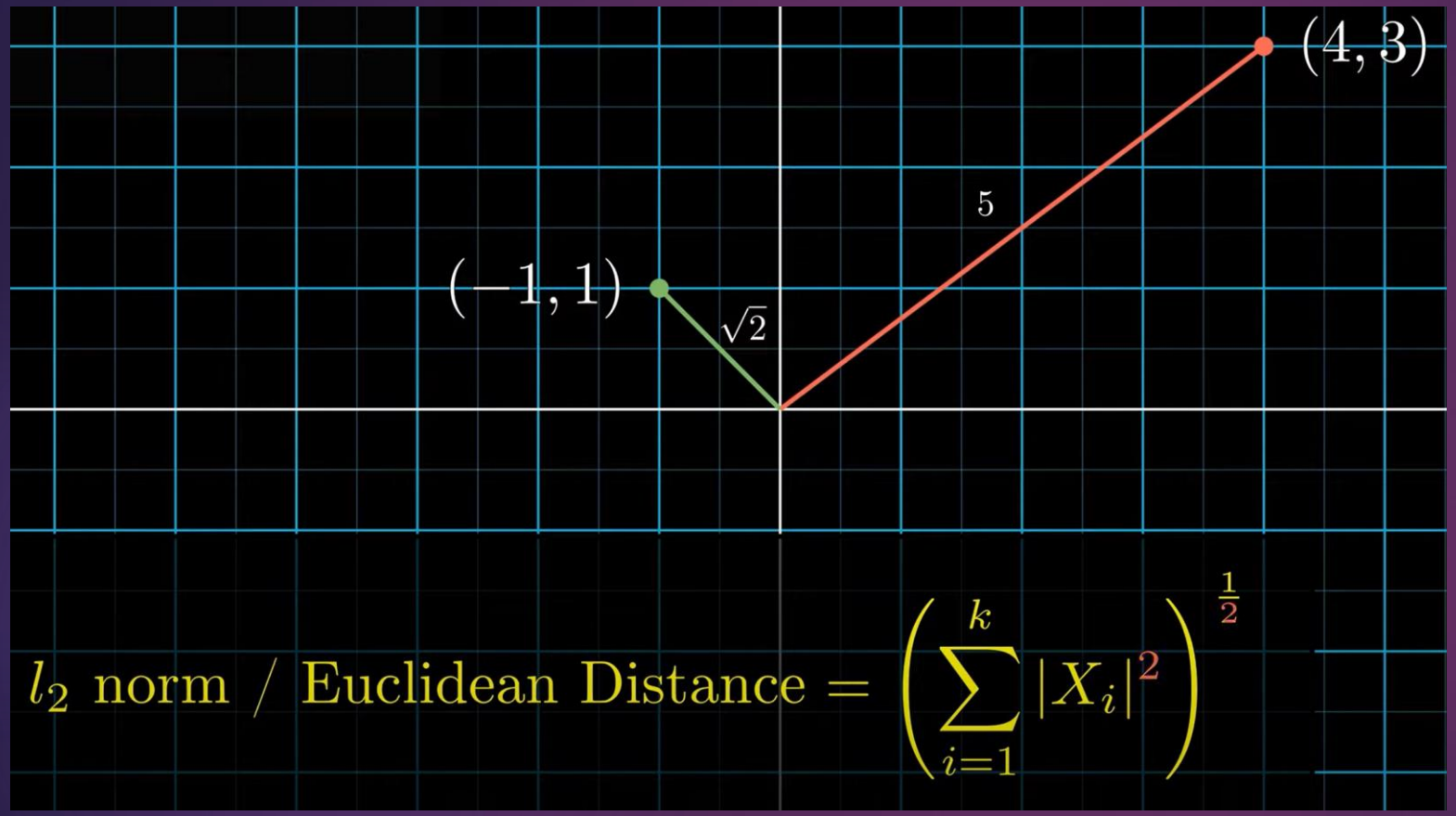
Norms: formal definition

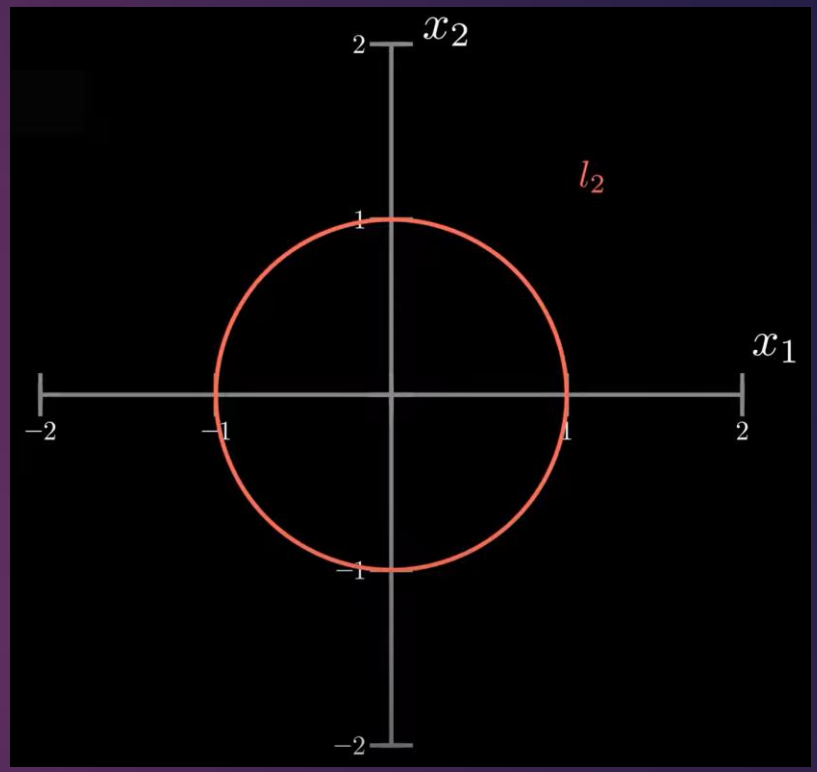
A pair of objects (V,d) *where* V *is a vector space and* d *is a norm* $d: V \times V = \|x\| \rightarrow \mathbb{R}$.

1. $\|x\| \geq 0$
2. $\|x\| = 0 \implies x = 0$
3. $\|\lambda x\| = |\lambda| \|x\| \qquad \lambda \in \mathbb{R}$
4. $\|x-z\| \leq \|x-y\| + \|y-z\|$

https://www.youtube.com/watch?v=NKuLYRui-NU

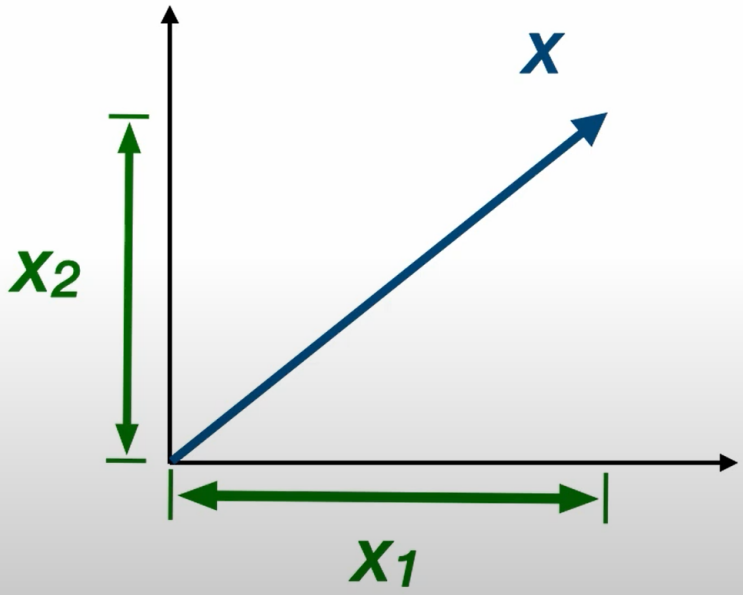
L_2 norm


(4, 3)
5
(−1, 1)
√2
l_2 norm / Euclidean Distance = $\left(\sum_{i=1}^{k} |X_i|^2\right)^{\frac{1}{2}}$


x_2
x_1
l_2
2
1
−1
−2
−2
−1
1
2

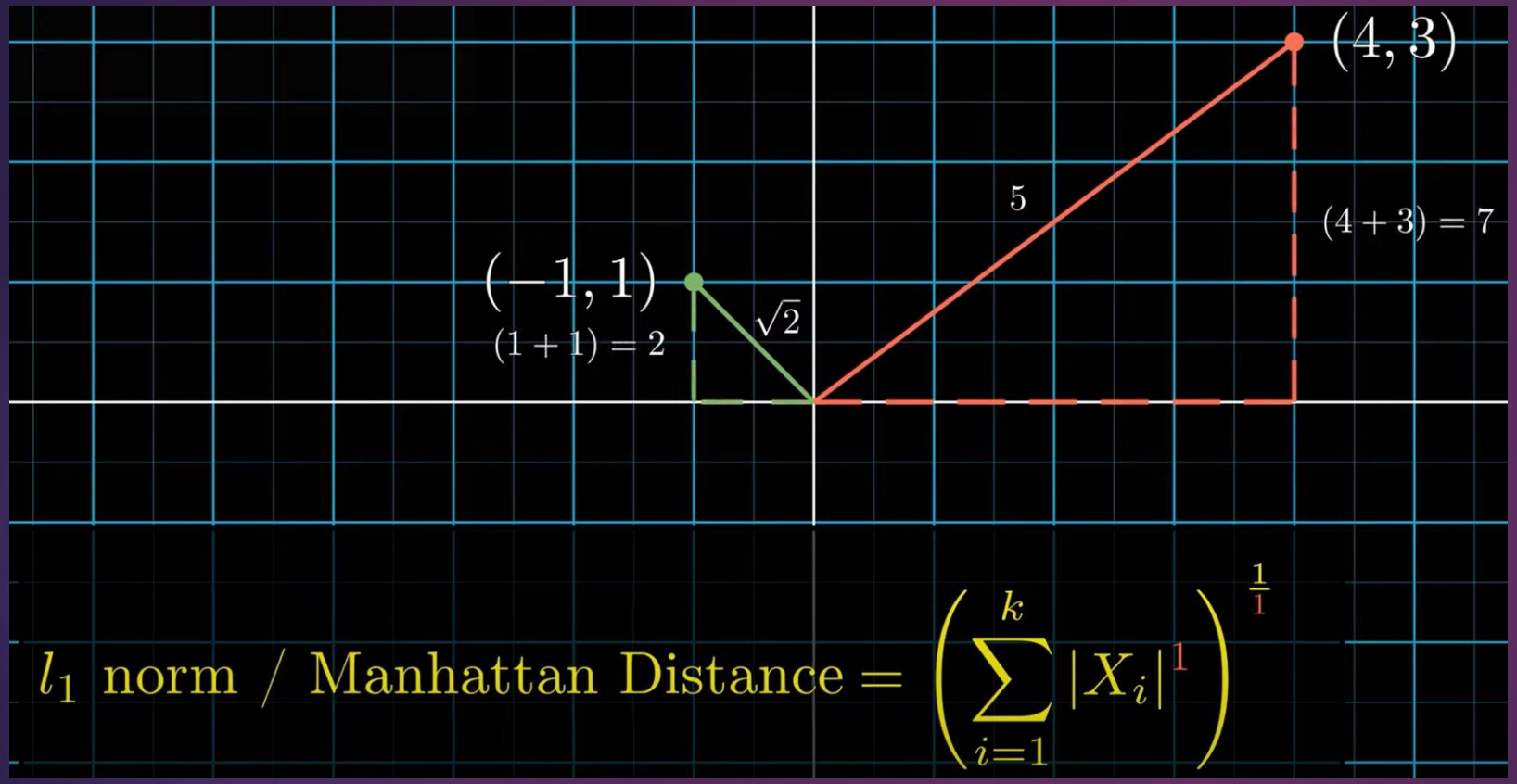
L_2 norm

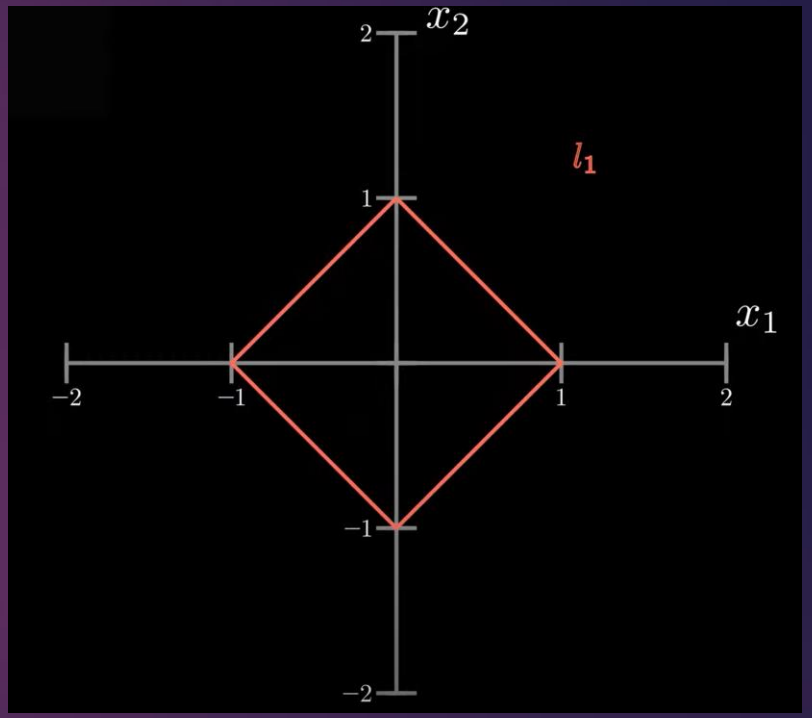
L_2 is defined as $\|x\|_2 = \sqrt{\sum_{i=1}^{n} |x_i|^2}$



$$\|x\|_2 = \sqrt{|x_1|^2 + |x_2|^2}$$

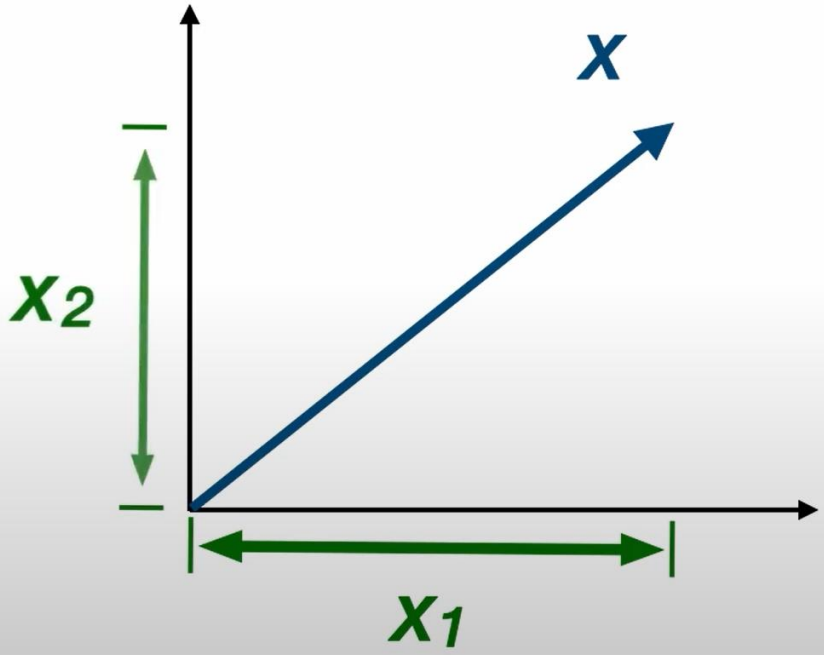
L_1 norm


(4, 3)
5
(4 + 3) = 7
(−1, 1)
(1 + 1) = 2
√2
l1 norm / Manhattan Distance = $\left(\sum_{i=1}^{k} |X_i|^1\right)^{\frac{1}{1}}$


x2
x1
l1
2
1
−1
−2
−2
−1
1
2

L_1 norm

L_1, written as $\|x\|_1$, is defined as $\|x\|_1 = \sum_{i=1}^{n} |x_i|$

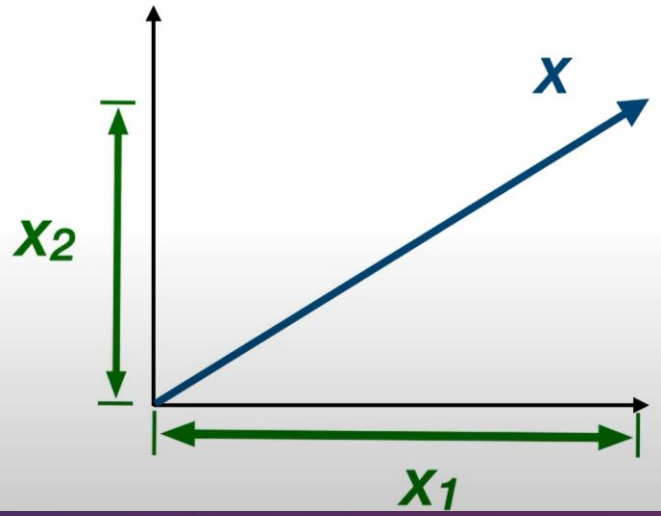


$\|x\|_1 = |x_1| + |x_2|$

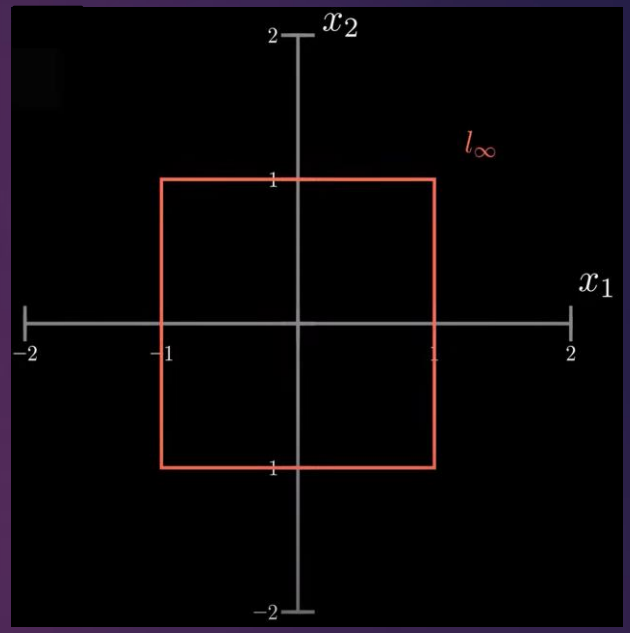
https://www.youtube.com/watch?v=NKuLYRui-NU

L_∞ norm

L_∞, written as $\|x\|_\infty$, is defined as $\|x\|_\infty = \max_i(|x_i|)$



$\|x\|_\infty = |x_1|$

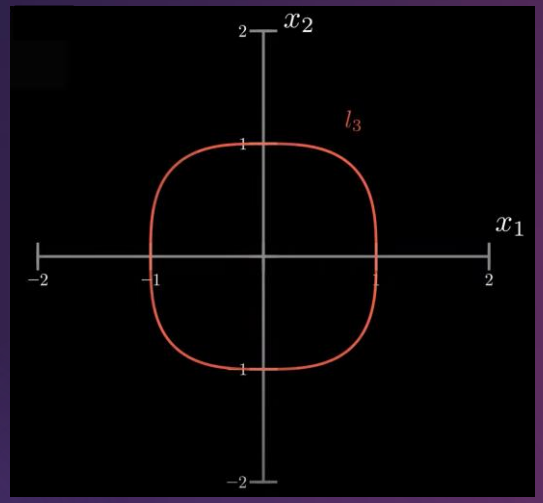


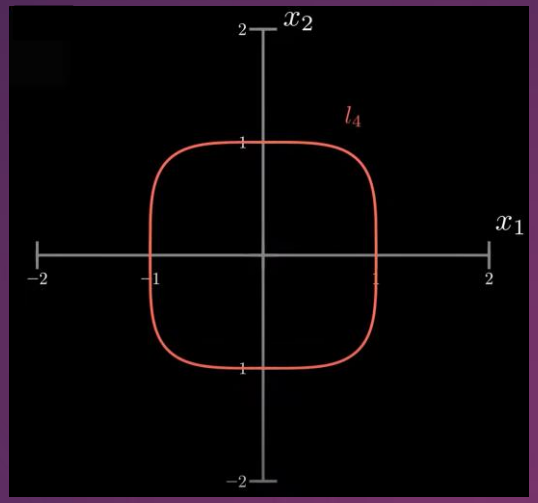
https://www.youtube.com/watch?v=NKuLYRui-NU

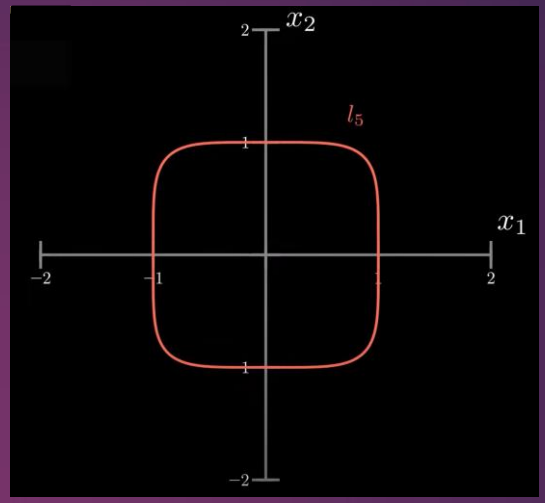
L_n norm

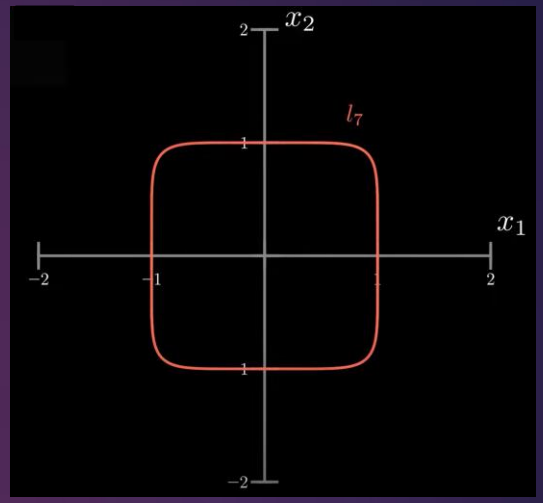
$$l_n \text{ norm / Generalized Norm} = \left(\sum_{i=1}^{k} |X_i|^n \right)^{\frac{1}{n}}$$

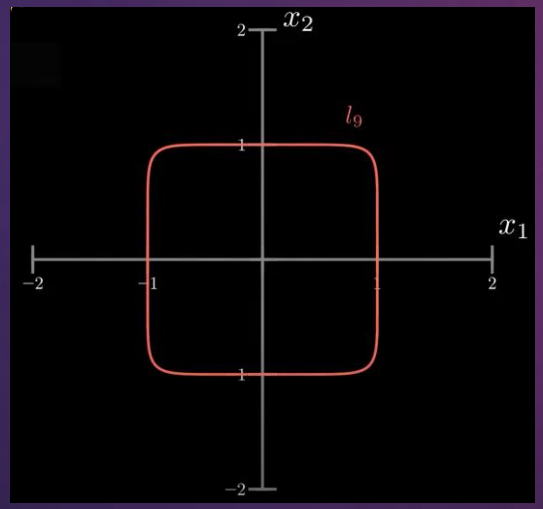
Norms

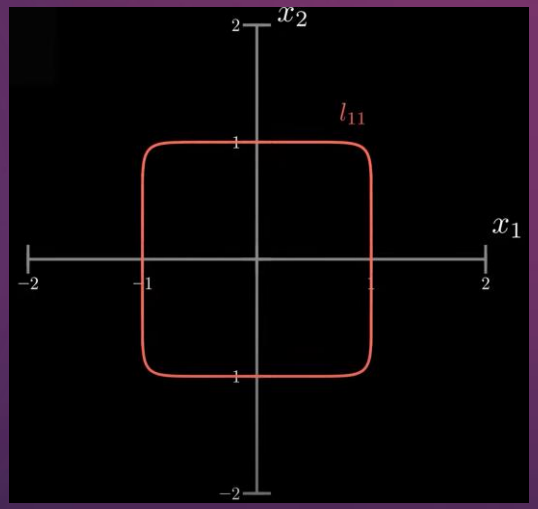

x_2
l_3
x_1

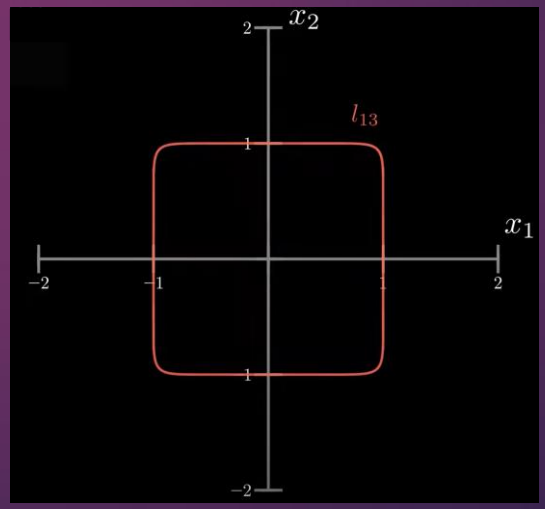

x_2
l_4
x_1

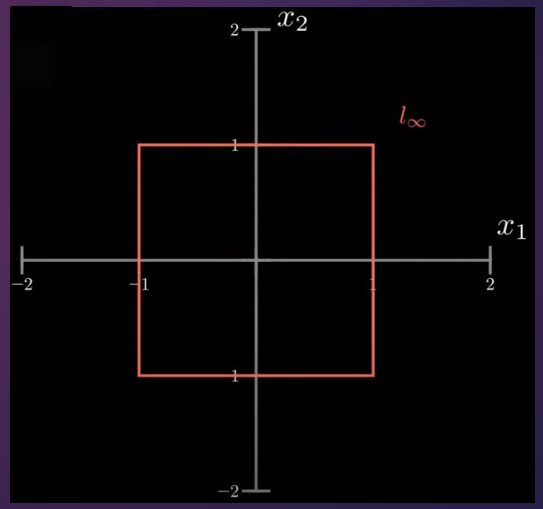

x_2
l_5
x_1


x_2
l_7
x_1


x_2
l_9
x_1


x_2
l_{11}
x_1


x_2
l_{13}
x_1


x_2
l_∞
x_1

Method of Lagrange Multipliers

Method of Lagrange Multipliers

We want to find the minimum and maximum value of a function, $f(x, y, z)$ subject to the constraint $g(x, y, z) = k$

1. Solve the following system of equations.

$$\nabla f(x, y, z) = \lambda \nabla g(x, y, z)$$
$$g(x, y, z) = k$$

2. Plug in all solutions, (x, y, z), from the first step into $f(x, y, z)$ and identify the minimum and maximum values, provided they exist and $\nabla g \neq \vec{0}$ at the point.

The constant, λ, is called the **Lagrange Multiplier**.

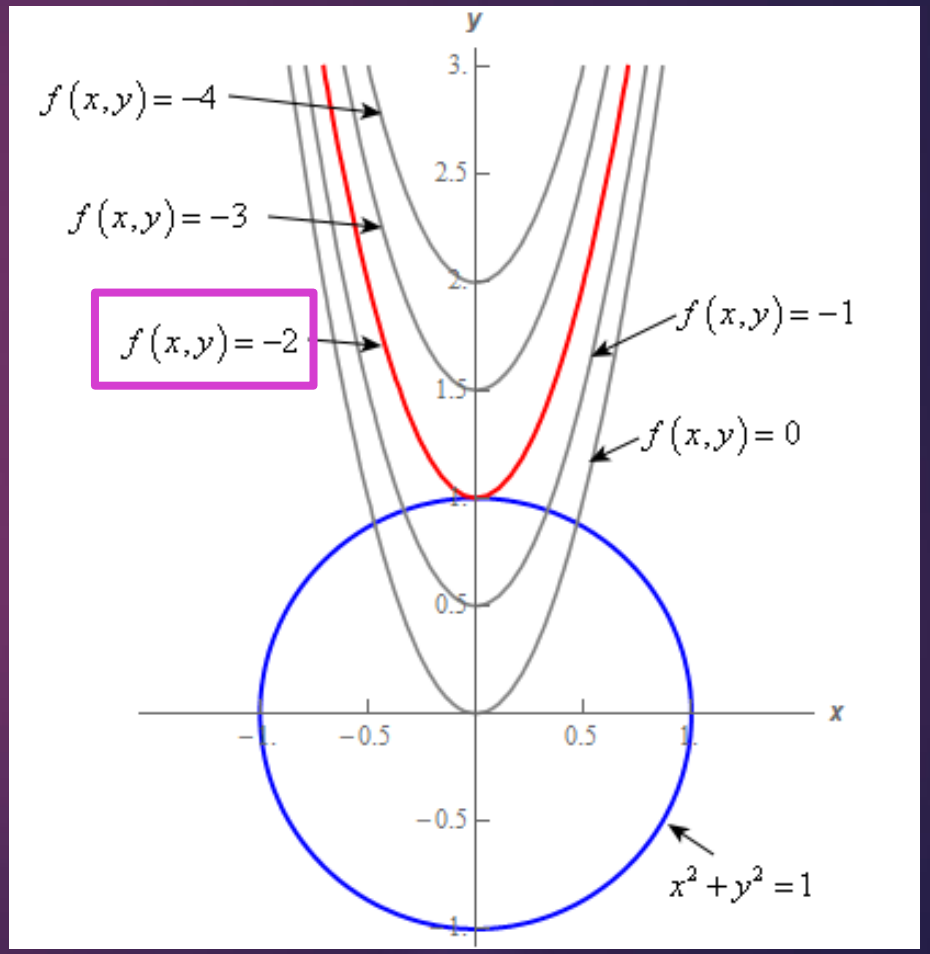
https://tutorial.math.lamar.edu/classes/calciii/lagrangemultipliers.aspx

Method of Lagrange Multipliers, example 1

We want to find the minimum value of a function $f(x,y) = 8x^2 - 2y$ subject to the constraint $x^2 + y^2 = 1$

Who will show that?

- The minimum value of $f(x,y)$ is -2 which occurs at (0,1)



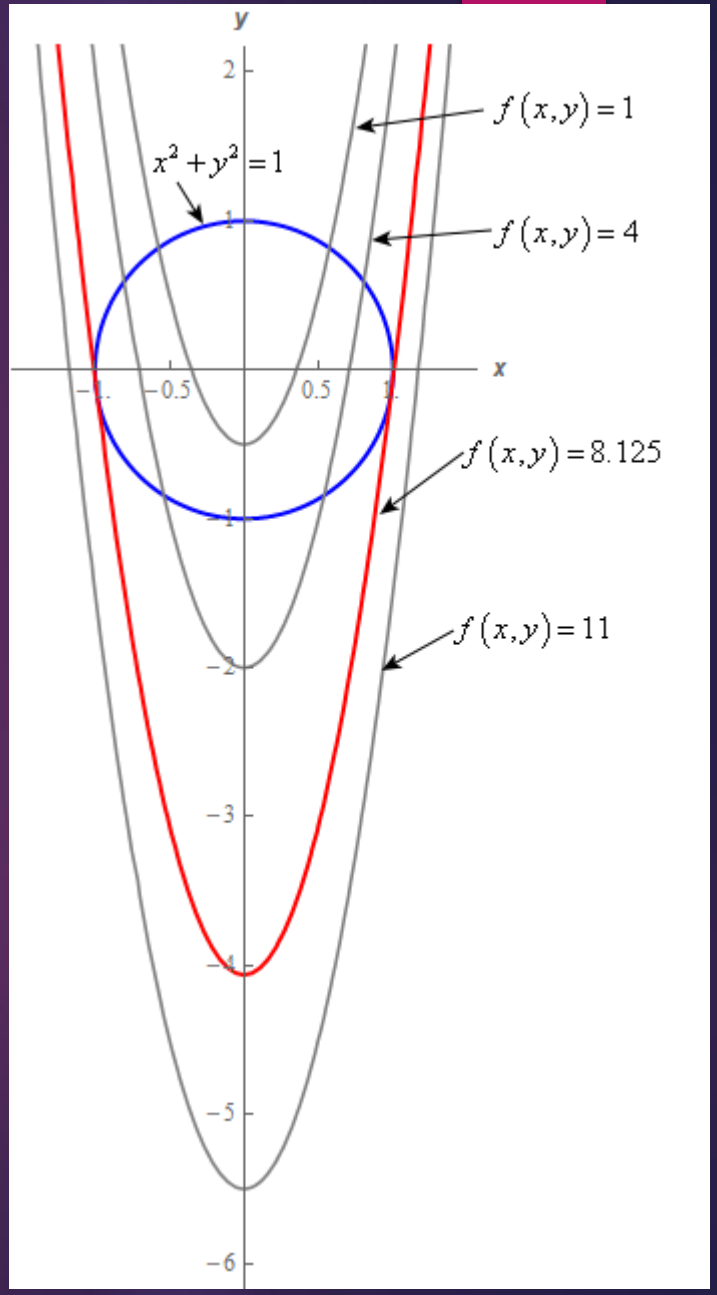
https://tutorial.math.lamar.edu/classes/calciii/lagrangemultipliers.aspx

Example 1

We want to find the maximum value of a function $f(x,y) = 8x^2 - 2y$ subject to the constraint $x^2 + y^2 = 1$

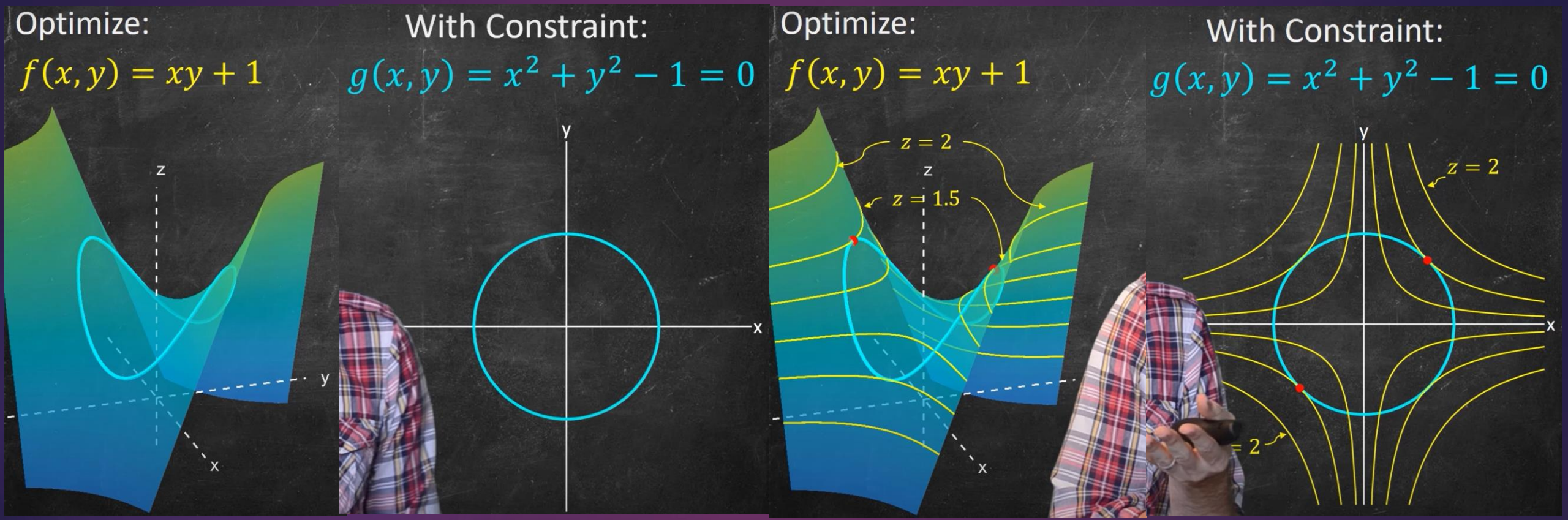
Who will show that?

- The maximum value of $f(x,y)$ is 8.125 which occurs at $\left(-\frac{3\sqrt{7}}{8}, -\frac{1}{8}\right)$ and $\left(\frac{3\sqrt{7}}{8}, -\frac{1}{8}\right)$



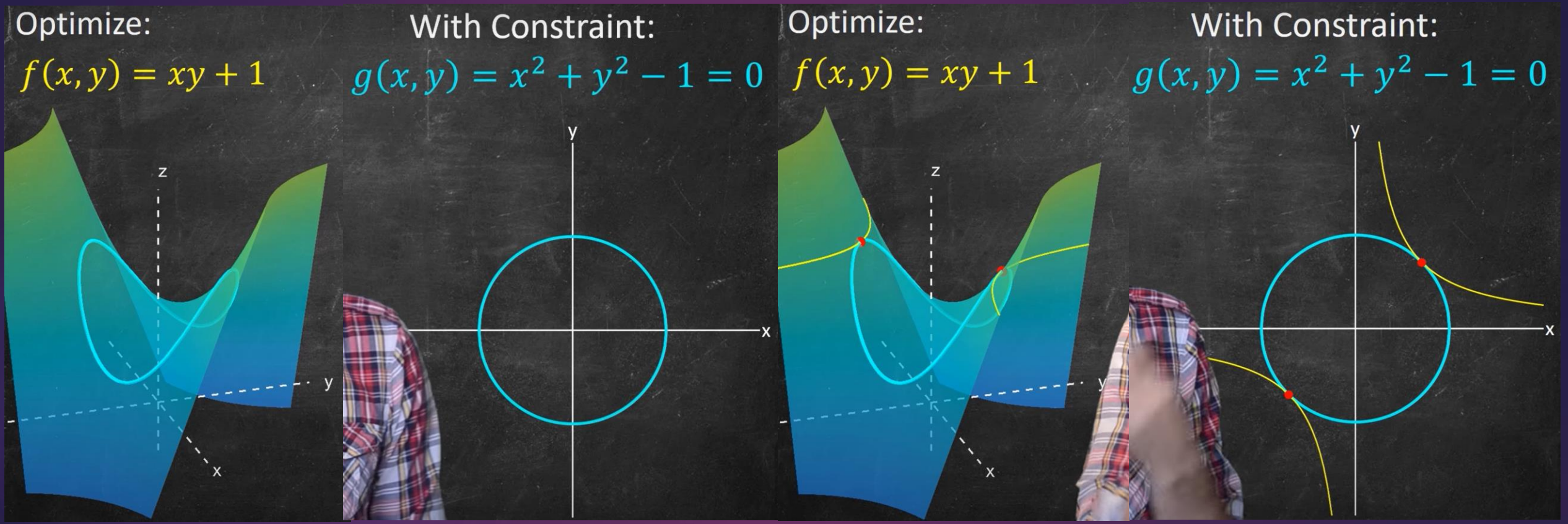
https://tutorial.math.lamar.edu/classes/calciii/lagrangemultipliers.aspx

Example 2



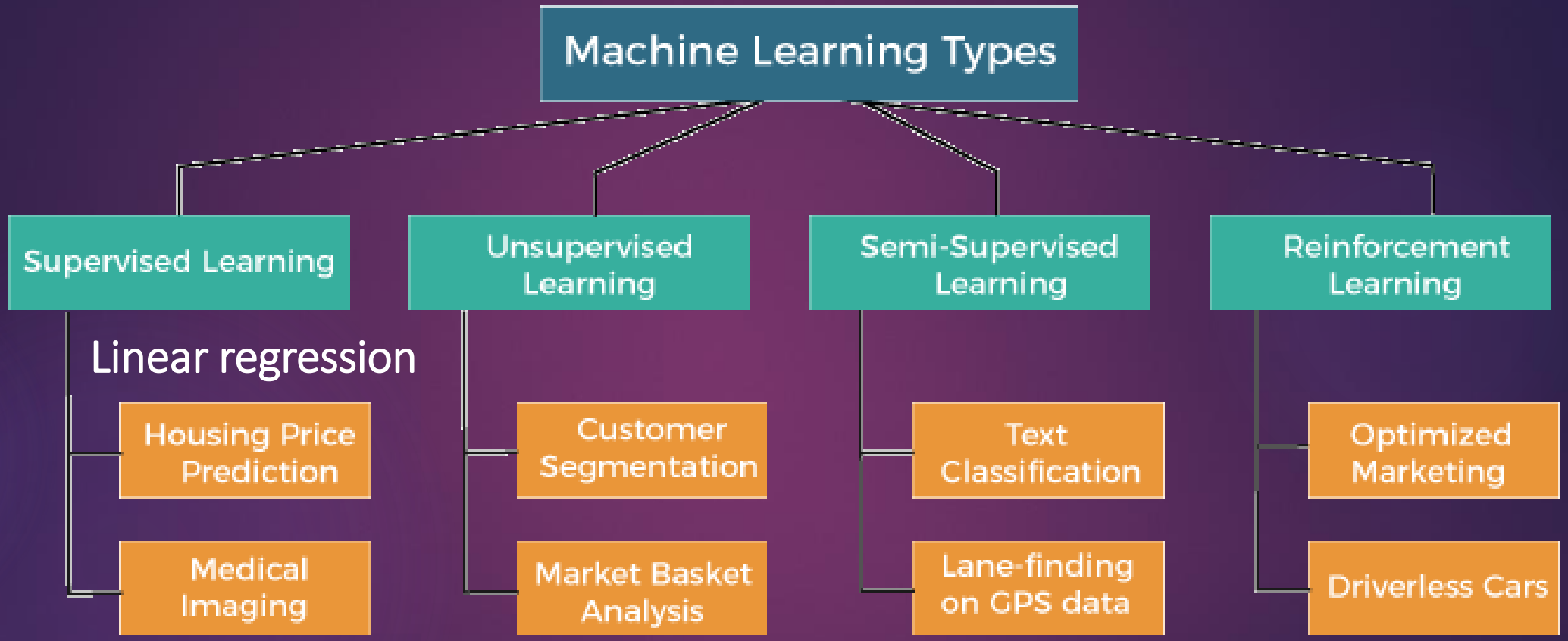
https://www.youtube.com/watch?v=8mjcnxGMwFo

Example 2



https://www.youtube.com/watch?v=8mjcnxGMwFo

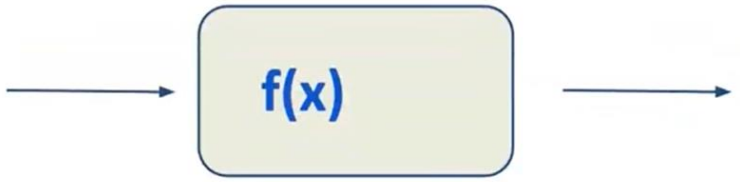
Recap of simple linear regression


Machine Learning Types
Supervised Learning
Unsupervised Learning
Semi-Supervised Learning
Reinforcement Learning
Linear regression
Housing Price Prediction
Medical Imaging
Customer Segmentation
Market Basket Analysis
Text Classification
Lane-finding on GPS data
Optimized Marketing
Driverless Cars

Example (predicting Gestational Age):

Urine/Blood Concentrations after 1st Trimester.

MicroPlastics	PFAS	Lead	....
2	1	3	....
1	2	2	....
3	2	1	....
1	1	1	....
....	....	....	....



Gestational Age

Yes/No
32
37
40
40
....

This model can raise the alarm during early stage of the pregnancy that a premature birth may occur. This information can help doctors to identify environmental factors that may be the cause.

How do we actually find f(x) ? Regression

The Goal of Regression

Given the data with n samples as $\{x_1, x_2, ..., x_n\}$ and its corresponding labels $\{y_1, y_2, y_3, ...\}$. We can find the best function $f(x)$ that gives us the best prediction if we solve

$$\min_f \underbrace{\frac{1}{n}\sum_i^n \underbrace{(f(x_i) - y_i)}_{\text{error}}^2}_{\text{The average prediction error}}. \tag{1}$$

1. Given x_i The difference between your prediction $f(x_i)$ and the true label y_i is the prediction error.
2. Given n samples, this function is simply the average prediction error^2
3. Solving this will give us the best function $f(x)$ yielding the minimum prediction error.
4. This is the most commonly used strategy to identify the prediction function.
5. The only problem is that there are ∞ possible functions, how do we pick?

Finding the best $f(x)$

To find the best $f(x)$ in the regression problem

$$\min_f \frac{1}{n}\sum_{i=1}^{n}(f(x_i)-y_i)^2 \quad \text{where} \quad f:\mathbb{R}^d \to \mathbb{R}, \quad x_i \in \mathbb{R}^d, \quad y_i \in \mathbb{R}.$$

We simply assume that the function is a linear combination of a bunch of other functions $\phi_1, \phi_2, ...$

$$f(x) = w_1\phi_1(x) + w_2\phi_2(x) + w_3\phi_3(x) + ... + w_m\phi_m(x) \quad \text{where} \quad w_1, w_2, ... \in \mathbb{R} \quad \text{and} \quad \phi_i : \mathbb{R}^d \to \mathbb{R}.$$

1. The set of $\phi_1(x), \phi_2(x), ...$ functions we "intentionally choose" to reconstruct $f(x)$ are called the basis functions.
2. The set of weights, $w_1, w_2, w_3, ...$, are the proportions of the basis functions combined to form $f(x)$.
3. In regression, we **decide ahead of time** the basis function $\phi_1(x), \phi_2(x), \phi_3(x), ...$ we want to use
4. Depending on the set of basis functions you decide use, we end up with a different algorithm.
5. And then find the best proportion of weights w to combine the basis function that gives us the best $f(x)$.

We next rewrite the function $f(x)$

Since we make the initial assumption of how the function will look like, i.e.,

$$\underbrace{f(x) = w_1\phi_1(x) + w_2\phi_2(x) + w_3\phi_3(x) + ... + w_m\phi_m(x)}_{\textbf{This is too long}} \quad \text{where} \quad w_1, w_2, ... \in \mathbb{R} \quad \text{and} \quad \phi_i : \mathbb{R}^d \to \mathbb{R}.$$

The equation could be very long, we simplify the notation by converting the equation into vector/vector multiplication where

$$f(x) = \underbrace{\begin{bmatrix} w_1 & w_2 & ... \end{bmatrix}}_{w} \underbrace{\begin{bmatrix} \phi_1(x) \\ \phi_2(x) \\ .. \end{bmatrix}}_{\phi(x)} = w^\top \phi(x).$$

It is very important here to distinguish the difference between $w_i, \phi_i(x)$ and w, ϕ.

1. $w_i \in \mathbb{R}$ and $\phi_i(x) \in \mathbb{R}$ **(with a subscript)** are the individual basis functions and its associated weight.
2. $w \in \mathbb{R}^m, \phi(x) \in \mathbb{R}^m$ **(without a subscript)**, are the collection of weights and basis functions.
3. It is important to pay attention to the dimensional difference between $w_i, w, \phi_i(x), \phi(x)$.
4. The vector $\phi(x)$ is called the feature map.

We next rewrite the Minimization problem

Given $f(x) = w^\top \phi(x)$, we can now rewrite the minimiation problem

$$\underbrace{\min_f \frac{1}{n}\sum_{i=1}^{n}(f(x_i) - y_i)^2}_{\text{The generalized Regression Formulation}} \longrightarrow \underbrace{\min_w \frac{1}{n}\sum_{i=1}^{n}(w^\top \phi(x) - y_i)^2}_{\text{The Kernelized Regression Formulation}}$$

With the original generalized regression formulation

- We had to find the best f to minimize the problem $\min_f$, *that was impossible.*

Now with the kernelized regression formulation

- We simply need to find the vector w that minimizes the expression, *this can be done via gradient descen*

To finally solve this problem, there is only 1 question remaining

How do we go about picking the basis functions $\phi_1(x), \phi_2(x), \ldots$

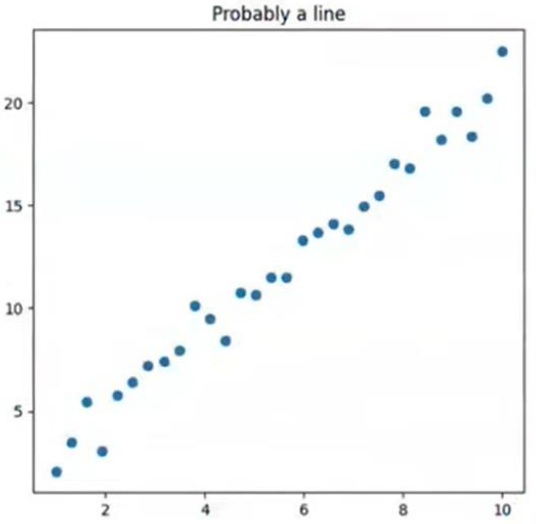
How do we pick basis functions? Educated Guess

If the data is simple, we can plot it out as shown on the right.
An equation of a line will probably work for figure 1.

- For this case, we assume that $\phi_1(x) = x$ and $\phi_2(x) = 1$, giving us

$$\begin{aligned} f(x) &= w_1 \quad \phi_1(x) \quad + \quad w_2 \quad \phi_2(x) \\ &= w_1 \quad (x) \quad + \quad w_2 \quad (1) \quad = \quad \underbrace{w_1 x + w_2}_{\text{equation of a line}} \end{aligned}$$

- If we use $\phi_1(x) = x$ and $\phi_2(x) = 1$ to find the best $f(x)$, this is called a linear regression.

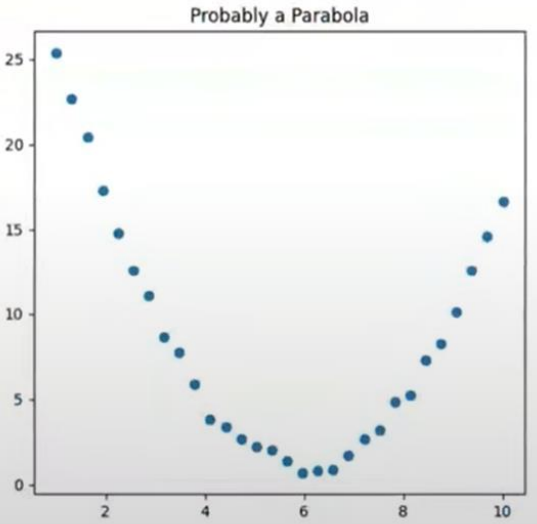


An equation of a parabola will probably work for figure 2.

- For this case, we assume that $\phi_1(x) = x^2$, $\phi_2(x) = x$, and $\phi_3(x) = 1$, giving us

$$\begin{aligned} f(x) &= w_1 \quad \phi_1(x) \quad + \quad w_2 \quad \phi_2(x) \quad + \quad w_2 \quad \phi_3(x) \\ &= w_1 \quad (x^2) \quad + \quad w_2 \quad (x) \quad + \quad w_2 \quad (1) \quad = \quad \underbrace{w_1 x^2 + w_2 x + w_3}_{\text{equation of a parabola}} \end{aligned}$$

- Using these basis functions to find the best $f(x)$ is called a Polynomial regression.



No matter what basis you pick, you can always write the function as $f(x) = \phi(x)^\top w = w^\top \phi(x)$. This is because

$$f(x) = w_1\phi_1(x) + w_2\phi_2(x) + \ldots = \begin{bmatrix} \phi_1(x) & \phi_2(x) & \ldots \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \\ \ldots \end{bmatrix} = \phi(x)^\top w.$$

Obtaining $\phi(x_i)$

We previously learned how to minimize the objective

$$\underbrace{\min_w \frac{1}{n}\sum_{i=1}^{n}(w^\top x_i - y_i)^2}_{\text{Today, we solved this minimization problem}} \xrightarrow{\text{but now we have}} \underbrace{\min_w \frac{1}{n}\sum_{i=1}^{n}(w^\top \phi(x_i) - y_i)^2}_{\text{This is actually the same problem}}$$

- If we have the data $\{x_1, x_2, ..., x_n\}$, we just need to calculate $\{\phi(x_1), \phi(x_2), ..., \phi(x_n)\}$.
- For example, let's say we assume we have 3 data points $x_1 = 1, x_2 = 2, x_3 = 3$ corresponding to 3 labels $y_1 = 2, y_2 = 4, y_3 = 6$.
- Also assume we are using **linear regression with** $\phi_1(x) = x$ and $\phi_2(x) = 1$, then

$$\underbrace{\begin{array}{c} x \\ \hline 1 \\ 2 \\ 3 \end{array}}_{\textbf{Original data}} \quad \text{and} \quad \underbrace{\begin{array}{c} y \\ \hline 2 \\ 4 \\ 6 \end{array}}_{\textbf{Original label}} \quad \text{then} \quad \underbrace{\begin{bmatrix} \phi(x_1 = 1) \\ \phi(x_2 = 2) \\ \phi(x_3 = 3) \end{bmatrix} = \begin{bmatrix} \phi_1(x_1) & \phi_2(x_1) \\ \phi_1(x_2) & \phi_2(x_2) \\ \phi_1(x_3) & \phi_2(x_3) \end{bmatrix} = \begin{bmatrix} x_1 & 1 \\ x_2 & 1 \\ x_3 & 1 \end{bmatrix}}_{\textbf{converting the data into its feature map}} = \underbrace{\begin{bmatrix} 1 & 1 \\ 2 & 1 \\ 3 & 1 \end{bmatrix}}_{\text{we now know } \phi(x_i)\ \forall\ i}$$

For this example, we would then minimize

$$\min_w \frac{1}{n}\left((w^\top \phi(x_1) - y_1)^2 + (w^\top \phi(x_2) - y_2)^2 + (w^\top \phi(x_3) - y_3)^2\right)$$

$$\min_w \frac{1}{n}\left(\left(\begin{bmatrix} w_1 & w_2 \end{bmatrix}\begin{bmatrix} 1 \\ 1 \end{bmatrix} - 2\right)^2 + \left(\begin{bmatrix} w_1 & w_2 \end{bmatrix}\begin{bmatrix} 2 \\ 1 \end{bmatrix} - 4\right)^2 + \left(\begin{bmatrix} w_1 & w_2 \end{bmatrix}\begin{bmatrix} 3 \\ 1 \end{bmatrix} - 6\right)^2\right)$$

After you find the $w = \begin{bmatrix} w_1 & w_2 \end{bmatrix}^\top$ that minimizes the means squared error, you found $f(x)$ since

$$f(x) = w_1 x + w_2$$

Solving Regression

Here are the steps to solve Regression

$$\min_w \mathcal{L} = \min_w \frac{1}{n}\sum_i^n (w^\top \phi(x_i) - y_i)^2$$

1. Pick the basis functions in the feature map
   - Linear Regression : $\phi(x) = \begin{bmatrix} x & 1 \end{bmatrix}^\top$
   - Polynomial Regression 2nd Order : $\phi(x) = \begin{bmatrix} x^2 & x & 1 \end{bmatrix}^\top$
2. Calculate the feature map for each sample, plug each x_i into $\phi(x_i) = \begin{bmatrix} x_i & 1 \end{bmatrix}^\top$
3. Find the derivative

$$\frac{d\mathcal{L}}{dw} = \frac{2}{n}\sum_i^n \underbrace{(w^\top \phi(x_i) - y_i)}_{\text{scalar value}} \underbrace{\phi(x_i)}_{\text{a vector}} \xrightarrow{\text{note that}} \text{the derivative is the summation of vectors}$$

4. Use gradient descent to identify the best w vector $w_{\text{next}} = w_{\text{now}} + \eta \frac{d\mathcal{L}}{dw}(w_{\text{now}})$, where $w = \begin{bmatrix} w_1 \\ w_2 \end{bmatrix}$
5. We have obtained the prediction function

$$\underbrace{f(x) = w_1 x + w_2}_{\text{Linear Function}} \quad \text{and} \quad \underbrace{f(x) = w_1 x^2 + w_2 x + w_3}_{\text{Polynomial Function}}$$

Reporting your error

- Remember you are trying to minimize the objective

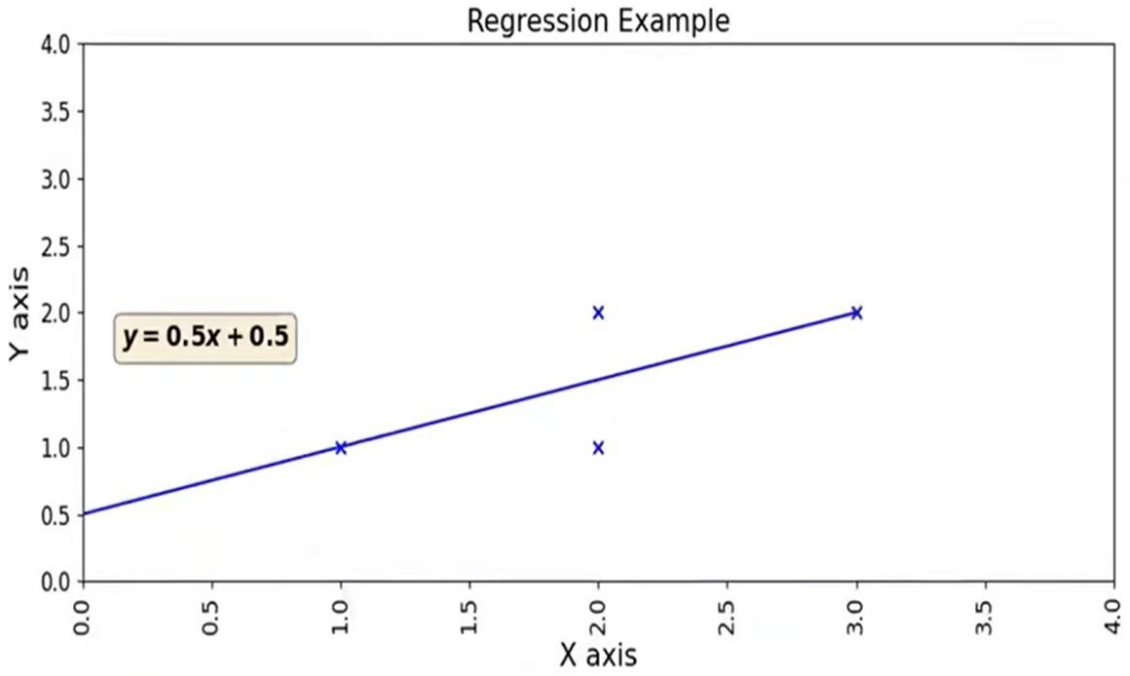
$$\min_{w} \underbrace{\frac{1}{n}\sum_{i}^{n}(w^{\top}\phi(x_i) - y_i)^2}_{\text{The mean squared error, MSE}}$$

- When you are asked to report the error of your prediction, you simply plug your final w value in and calculate your avera
  This is the number you would report.

Exercise 1. Write the code that solves a simple linear regression objective with **Gradient Descent**

x	y
1	1
2	1
2	2
3	2

You should be able to get the equation
$y = 0.5\ x + 0.5$.



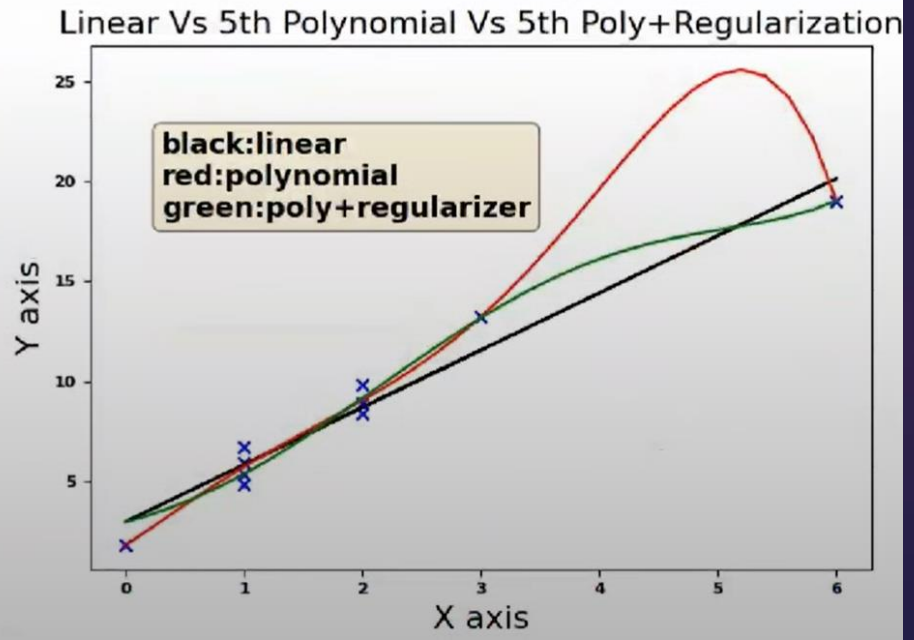
Feature Selection with LASSO

So how does Regularization work?

- When we perform regression, we aim to minimize the average square of the prediction error.
- Regularization works by adding an addition **constraint term** into the minimization problem

$$\min_f \frac{1}{n}\sum_i (f(x_i) - y_i)^2 \xrightarrow[\textbf{extra term}]{\textbf{add an}} \min_f \underbrace{\frac{1}{n}\sum_i (f(x_i) - y_i)^2 + \textbf{Constraint}}_{\textbf{We now minimize this whole thing}}$$

- As a result of adding an extra **Constraint term**.
  - We can continue to use very complicated models
  - While not increasing much variation (shown in figure)
  - Black is a simple linear model.
  - Red is a complex 5th-order polynomial model.
  - Green is also a complex 5th order with constraint.
  - The constraint created a compromise!
- Regularization: Use complicated models without overfitting.



We call it Adding a constraint because we add a constraint to our objective

Here is our basic regression objective with a constraint term

$$\min_w \quad \frac{1}{2}\sum_i^N (w^\top \phi(x_i) - y_i)^2 + \text{constraint}$$

There are many different constraints. The 3 most common are ...

$$\min_w \quad \frac{1}{2}\sum_i^N (w^\top \phi(x_i) - y_i)^2 + |w|_1$$

This objective is called **Lasso**

$$\min_w \quad \frac{1}{2}\sum_i^N (w^\top \phi(x_i) - y_i)^2 + |w|_2^2$$

This objective is called **Ridge Regression**

$$\min_w \quad \frac{1}{2}\sum_i^N (w^\top \phi(x_i) - y_i)^2 + |w|_1 + |w|_2^2$$

This objective is called **Elastic Net**

Practice taking derivatives

$$\min_w \quad \frac{1}{2}\sum_i^N (w^\top \phi(x_i) - y_i)^2 + |w|_1$$

$$\min_w \quad \frac{1}{2}\sum_i^N (w^\top \phi(x_i) - y_i)^2 + |w|_2^2$$

$$\min_w \quad \frac{1}{2}\sum_i^N (w^\top \phi(x_i) - y_i)^2 + |w|_1 + |w|_2^2$$

To form a compromise you simply run Gradient descent over this objective.

Find the derivative used during gradient descent for each function.

Regularization

$$\min_w \quad \frac{1}{2}\sum_i^N (w^\top \phi(x_i) - y_i)^2 + |w|_1$$

$$\min_w \quad \frac{1}{2}\sum_i^N (w^\top \phi(x_i) - y_i)^2 + |w|_2^2$$

$$\min_w \quad \frac{1}{2}\sum_i^N (w^\top \phi(x_i) - y_i)^2 + |w|_1 + |w|_2^2$$

They are commonly **solve with gradient descent**

Luckily, you know how to take the derivative of these objective. By taking the derivative, you can slowly walk towards a good solution !!!

$$\sum_i^N (w^\top \phi(x_i) - y_i)\phi(x_i) + sign(w)$$

$$\sum_i^N (w^\top \phi(x_i) - y_i)\phi(x_i) + 2w$$

$$\sum_i^N (w^\top \phi(x_i) - y_i)\phi(x_i) + sign(w) + 2w$$

Why do we call them constraints ?

$$\min_w \quad \frac{1}{2}\sum_i^N (w^\top \phi(x_i) - y_i)^2 + |w|_1$$

$$\min_w \quad \frac{1}{2}\sum_i^N (w^\top \phi(x_i) - y_i)^2 + |w|_2^2$$

$$\min_w \quad \frac{1}{2}\sum_i^N (w^\top \phi(x_i) - y_i)^2 + |w|_1 + |w|_2^2$$

Notices that as the size of the weight w increase, the total value of the objective also increase.

Therefore, using a weight vector w that are large goes against the minimization objective.

(The constraints constrains the solution to smaller values)

This is why the constraint is also called the **penalty term**.

(It penalizes solutions with large weights values)

Adding the constraint is like adding an “and” statement saying that

I want to achieve my regression objective and I want the weights to be as small as possible.

Regularization Weights λ

$$\min_w \quad \frac{1}{2}\sum_i^N (w^\top \phi(x_i) - y_i)^2 + |w|_1$$

$$\min_w \quad \frac{1}{2}\sum_i^N (w^\top \phi(x_i) - y_i)^2 + |w|_2^2$$

$$\min_w \quad \frac{1}{2}\sum_i^N (w^\top \phi(x_i) - y_i)^2 + |w|_1 + |w|_2^2$$

$$\min_w \quad \frac{1}{2}\sum_i^N (w^\top \phi(x_i) - y_i)^2 + \lambda|w|_1$$

$$\min_w \quad \frac{1}{2}\sum_i^N (w^\top \phi(x_i) - y_i)^2 + \lambda|w|_2^2$$

$$\min_w \quad \frac{1}{2}\sum_i^N (w^\top \phi(x_i) - y_i)^2 + \lambda_1|w|_1 + \lambda_2|w|_2^2$$

We can gain further control over the compromise by adding hyperparameters in front of the constraint; λ.

In the previously shown cases, we assume λ to be 1. But they can be any value.

The larger the λ, the heavier the penalty. Therefore, when λ is large, the objective tries to find the smallest possible weight vector w.

Unfortunately, there are many ways to say the same thing

adding a constraint = regularization = adding a penalty

Constraint = penalty = regularizer

Next, let's better understand Linear Regression Weights

Given the following data on cancer predictions

Protein 1	Protein 3	Protein 3	Percentage %
2	3	4	32
3	1	7	27
1	0	0	7
6	7	9	84
...	...	...	...

Table 1: Using Protein to Predict Cancer Rate

After using linear regression, we identified the weights as

$protein1$	$protein2$	$protein3$	1
w_1	w_2	w_3	1
7	6	0.00001	0

$$\implies \quad Xw = \hat{y} \quad \implies \quad \begin{bmatrix} 2 & 3 & 4 & 1 \\ 3 & 1 & 7 & 1 \\ 1 & 0 & 0 & 1 \\ 6 & 7 & 9 & 1 \\ \dots & \dots & \dots & \dots \end{bmatrix} \begin{bmatrix} 7 \\ 6 \\ 0.00001 \\ 0 \end{bmatrix} = \begin{bmatrix} 32 \\ 27 \\ 7 \\ 84 \end{bmatrix}$$

If we look at the weights carefully, how much does each Protein impact the final percentage?

- Notice how Pretein 3's weight is so small, it doesn't impact the final prediction.
- This observation tells us that the weights of **linear regression** not only give us the function, it also tell how much each feature contributes to the final prediction.

The constraints create a compromise solution, but when do we use L1 vs L2 norm?

The L1 and L2 norm forces the weight to be small, but **L1 norm** forces w to be small in a special way.

Below are the results from

1. Linear Regression
2. Linear Regression with L2
3. Linear Regression with L1

Linear Regression with L1
w = [2.1 1.0 0]
Accuracy Score: 0.86

True Solution
w = [3 2 0]

Regular Linear Regression
w = [3.1 2.0 -0.03]
Accuracy Score: 0.999

Linear Regression with L2
w = [2.7 1.8 0.25]
Accuracy Score: 0.99

L1 pushes the weights smaller, but it makes none important weights nearly 0.

Therefore, we can **identify important factors** easier with L1 regularization, at the cost of lower accuracy.

Regular Regression gives you the most accurate result

L2 makes the weights smaller and gives you slightly less accurate result

The LASSO estimator

The lasso estimate is defined by the solution to the l_1 optimization problem

$$\text{minimize} \left(\frac{\|\mathbf{Y} - \mathbf{X}\beta\|_2^2}{n} \right) \qquad \text{subject to} \sum_{j=1}^{k} \|\beta\|_1 < t$$

where t is the upper bound for the sum of the coefficients. This optimization problem is equivalent to the parameter estimation that follows

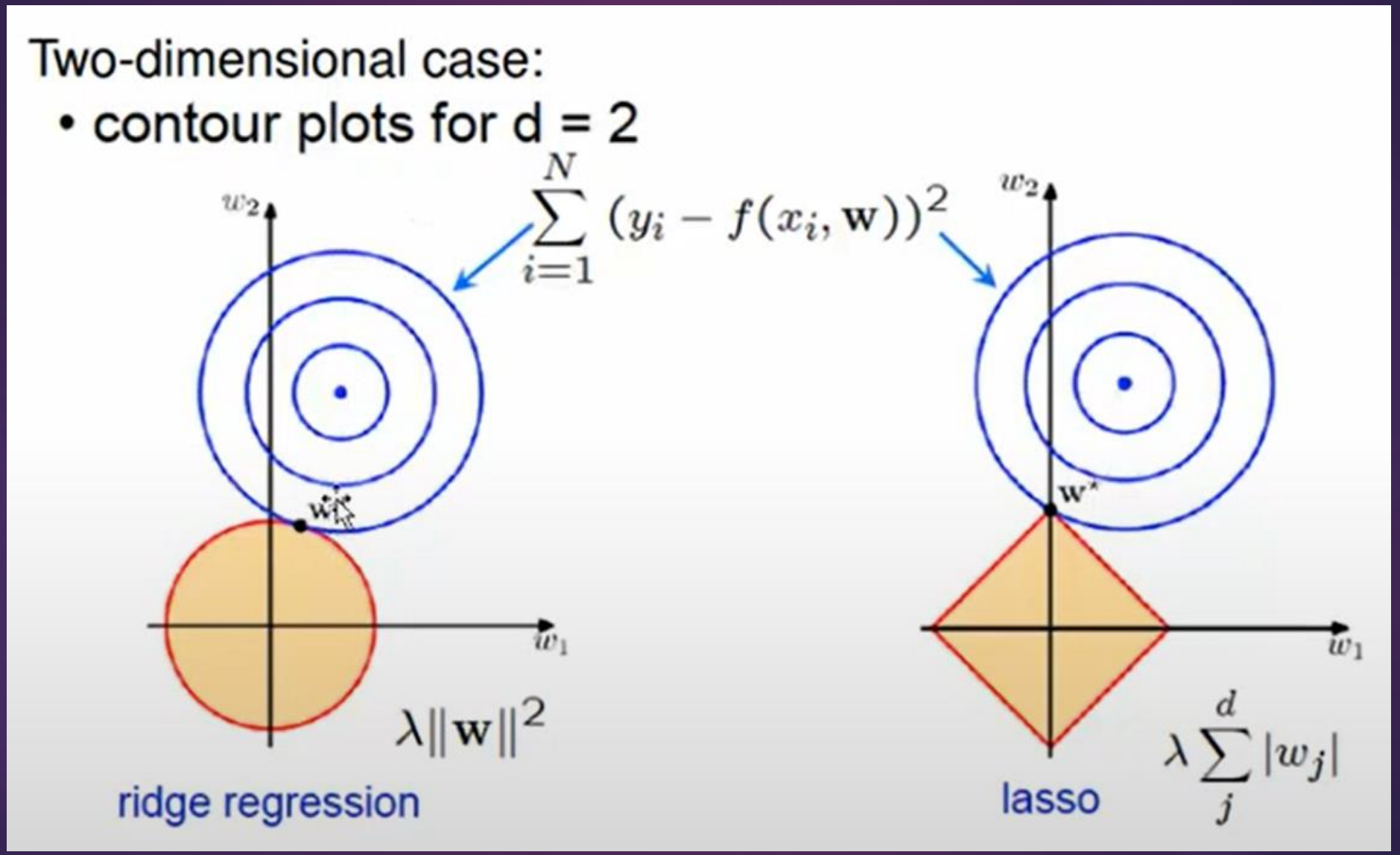
$$\hat{\beta}(\lambda) = \underset{\beta}{\text{argmin}} \left(\frac{\|\mathbf{Y} - \mathbf{X}\beta\|_2^2}{n} + \lambda\|\beta\|_1 \right)$$

where $\|\mathbf{Y} - \mathbf{X}\beta\|_2^2 = \sum_{i=0}^{n}(Y_i - (\mathbf{X}\beta)_i)^2$, $\|\beta\|_1 = \sum_{j=1}^{k}|\beta_j|$ and $\lambda \geq 0$ is the parameter that controls the strength of the penalty, the larger the value of λ, the greater the amount of shrinkage.

The relation between λ and the upper bound t is a reverse relationship. Indeed as t becomes infinity, the problem becomes an ordinary least squares and λ becomes 0. Viceversa as t becomes 0, all coefficients shrink to 0 and λ goes to infinity.

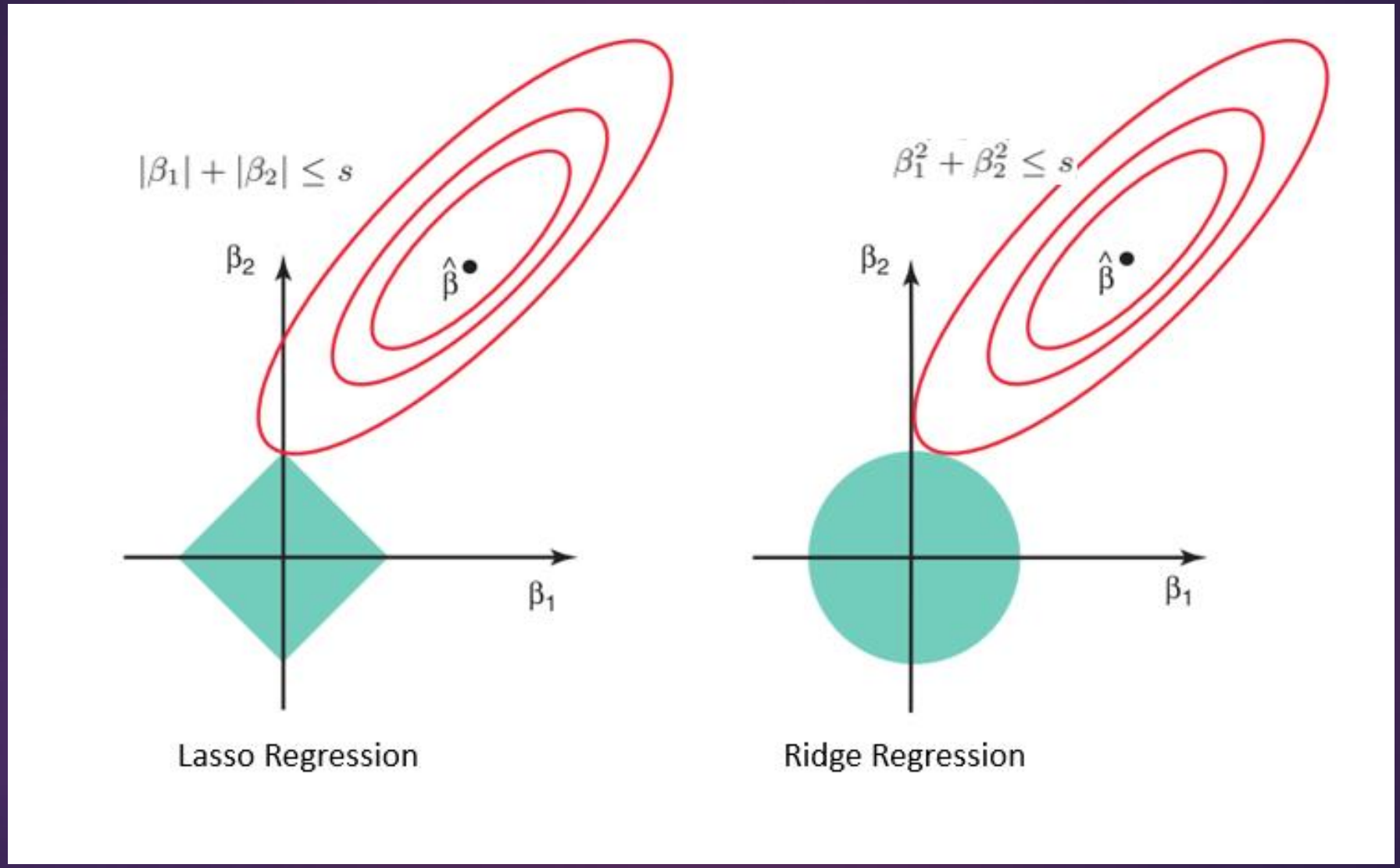
https://vu-business-analytics.github.io/internship-office/papers/paper-fonti.pdf

The LASSO estimator for feature selection



https://www.youtube.com/watch?v=9LNpiiKCQUo&t=195s

The LASSO estimator for feature selection



https://www.datacamp.com/tutorial/tutorial-lasso-ridge-regression

Hyperparameters and Identifying Regularization Weights λ with Train/Validation/Test method

$$\min_w \quad \frac{1}{2}\sum_i^N (w^\top \phi(x_i) - y_i)^2 + \lambda |w|_1$$

$$\min_w \quad \frac{1}{2}\sum_i^N (w^\top \phi(x_i) - y_i)^2 + \lambda |w|_2^2$$

$$\min_w \quad \frac{1}{2}\sum_i^N (w^\top \phi(x_i) - y_i)^2 + \lambda_1 |w|_1 + \lambda_2 |w|_2^2$$

These values are called **hyperparameters**.

They are things you need to set before training the algorithm.

We previously discussed model selection: **Hyperparameter** are also model choices

During the learning of the function stage, we split the data into

Train / Validate / Test

- We make a list of hyperparameter settings
  - Different polynomial orders
  - Test out different λ values

- Under each possible permutation of hyperparameter setting, we perform regression
- We use validation data to see which hyperparameter setting the performed the best and pick them
- We finally use the chosen hyperparameter on the test dataset and report the final accuracy result.

Put Everything We Have Learned So Far Together.

Given data X, label Y, starting w_0, and feature map $\phi(x)$ as

$$\text{data} = \begin{bmatrix} X & Y \\ 0 & 0 \\ 1 & 1 \\ 2 & 1 \\ 3 & 2 \end{bmatrix} \quad , \quad w_0 = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \quad , \quad \phi(x) = \begin{bmatrix} x^2 \\ x \\ 1 \end{bmatrix} \quad , \quad \lambda = 0.2 \quad , \quad \eta = 0.01$$

1. Given the data above, identify the derivative for

$$\min_w \frac{1}{n} \sum_i^n (f(x_i) - y_i)^2 + |w|_1.$$

2. Put the derivative in Matrix/Vector form (Very Important you can do this).
3. Write the Python code to run 1 gradient descent step with the Lasso objective (L1-norm) and feature map of $\phi(x)$.
4. Add a for loop and minimize the LASSO objective.
5. Plot out the Gradient Descent steps as they lower the objective.

Why are Constraints so important (The world peace objective)

Let's say you tell the computer to create world peace by minimizing war

$$\min_f \quad war(f(x))$$

Gradient Descent does **not care** how you achieve the final objective.

So the computer algorithm might tell you that the best solution to minimize war is

To kill all humans.

Gradient Descent does not have the same values as us, it must have constraints

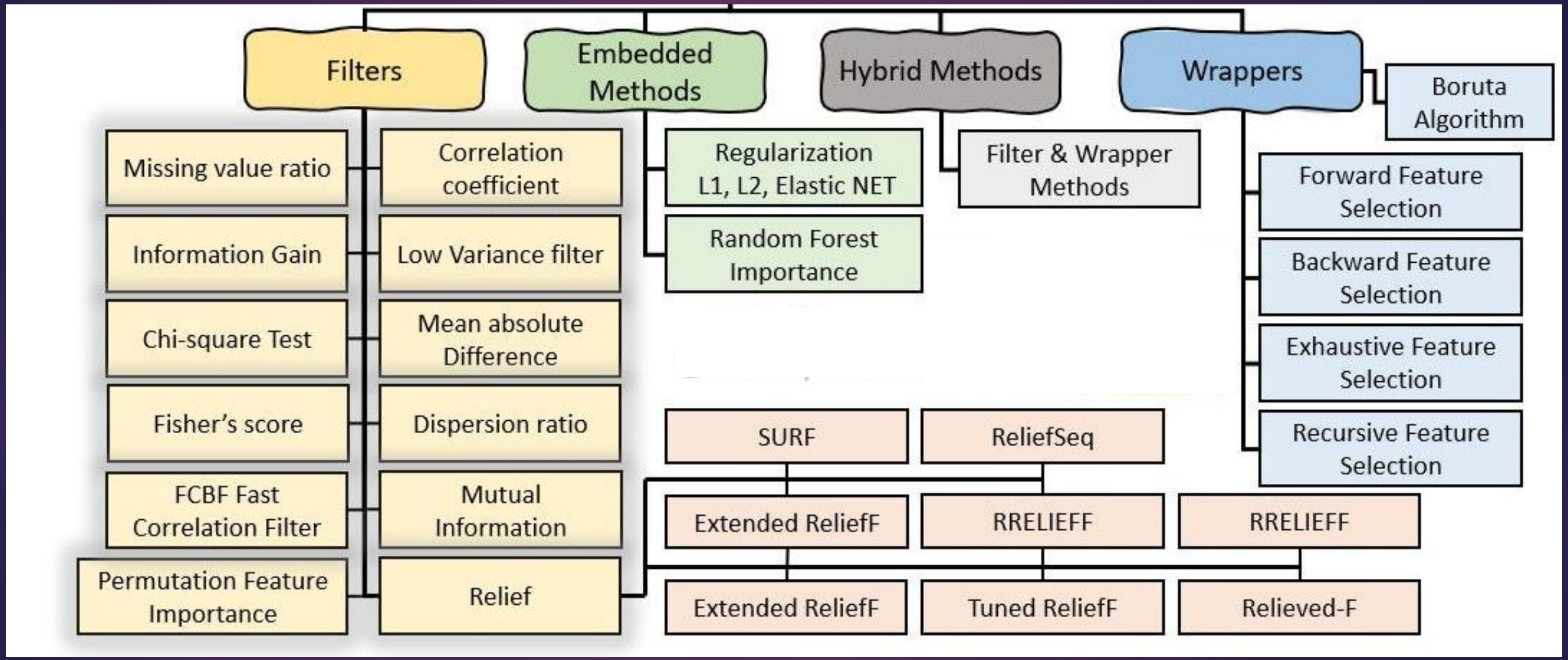
$$\min_f \quad war(f(x)) + \text{no killing humans}$$

Constraints cannot guarantee our safety, because the best solution might then be

Practice on Lasso

- https://www.datacamp.com/tutorial/tutorial-lasso-ridge-regression
- https://medium.com/@agrawalsam1997/feature-selection-using-lasso-regression-10f49c973f08
- https://www.blog.trainindata.com/lasso-feature-selection-with-python/

How do we carry out feature selection?
Supervised Feature Selection



https://medium.com/analytics-vidhya/feature-selection-extended-overview-b58f1d524c1c

Xin chân thành cảm ơn!

LUU PHUC LOI, PHD

ZALO: 0901802182

LUU.P.LOI@GOOGLEMAIL.COM