Math Intro

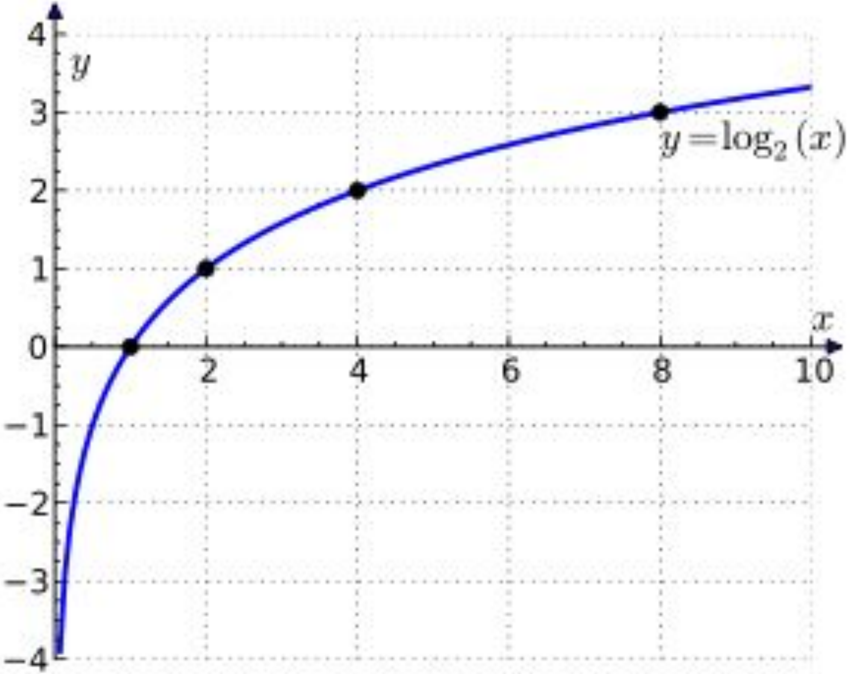
Dr. Chelsea Parlett-Pelleriti

That's really what mathematics is, it's seeing the ***connections between…different ideas****. Ideas that look different but are nevertheless* ***somehow connected****.*

- Gil Strang

Logarithms

Log rules:



Logarithms

- Generally: $(0, \infty) \Rightarrow (-\infty, \infty)$
- $(0,1) \Rightarrow$ negative
- $(1,\infty) \Rightarrow$ positive
- Opposite of Exponentiation

L_p Norms

$$\|x\|_1 = \sum_{i=0}^{n} |x_i|$$

$$\|x\|_2 = \sqrt{\sum_{i=0}^{n} x_i^2}$$

$$\|x\|_\infty = \lim_{p\to\infty} (\sum_{i=0}^{n} x_i^p)^{\frac{1}{p}}$$

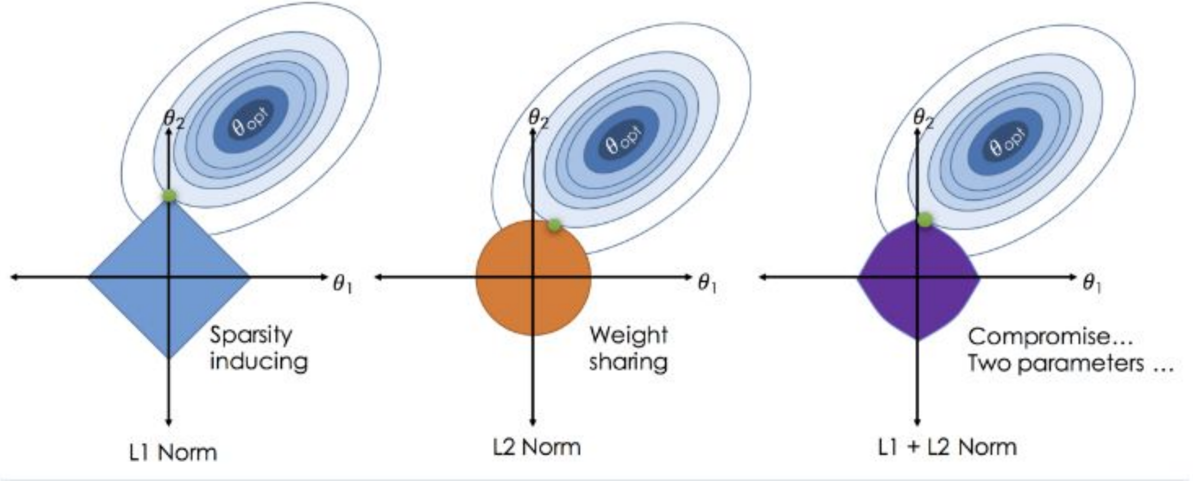
```
np.linalg.norm(x-y)
```

L_p Norms

$$\|x\|_1 = \sum_{i=0}^{n} |x_i|$$

$$\|x\|_2 = \sqrt{\sum_{i=0}^{n} x_i^2}$$

$$\|x\|_\infty = \lim_{p\to\infty} (\sum_{i=0}^{n} x_i^p)^{\frac{1}{p}}$$



L_p Norms

- Lp norms measure the **size** of an object

Lp Norms

Distance = size of vector going from **a** to **b**

Transpose

$$x \qquad x^T \text{ or } x'$$

$$\begin{bmatrix} a & b & c \end{bmatrix} \qquad \begin{bmatrix} a \\ b \\ c \end{bmatrix}$$

Dot Product

$$\begin{bmatrix} a & b \end{bmatrix} \cdot \begin{bmatrix} x \\ y \end{bmatrix} = ax + by$$

Dot Product

- Dot products are the sum of all the element wise multiplications of two vectors

Linear Combinations

Example:

Course Grade:

Attendance/Participation (in-class activities, quizzes)

15 %

Challenge Activities*:

5%

Programming Assignments:

40 %

Exam 1:

10 %

Exam 2:

10 %

Final:

20 %

Linear Combinations

Generally:

$$\mathbf{w}^{\mathbf{T}}\mathbf{x} = \sum_{i=0}^{n} w_i * x_i = \mathbf{w} \cdot \mathbf{x}$$

Linear Combinations

- Linear Combinations take a weight vector **w** and dot it with our variable vector **x**
- This gives us a new composite value that uses weights (**w**) to combine the variables in **x**
- Linear Combinations you might know:
  - Linear Regression
  - Principal Components

Matrices and Vectors

- Data as a Matrix/Vector (it’s an excel spreadsheet)
- Matrix Algebra

$$\begin{bmatrix} 1 & 0 & 1 \\ 2 & 3 & 1 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} 1 & 2 & 3 \\ 1 & 3 & 1 \\ 0 & 1 & 0 \end{bmatrix} = \begin{bmatrix} \quad & \quad & \quad \\ & & \\ & & \end{bmatrix}$$

Matrices and Vectors

- Vectors are arrays of numbers that can represent a point, or a line from the origin
- Matrices are collections of vectors and have both rows and columns
  - Data frames are like matrices
  - Correlation matrices

One Hot Encoding

$$\begin{bmatrix} apple \\ carrot \\ celery \\ carrot \end{bmatrix} \rightarrow \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

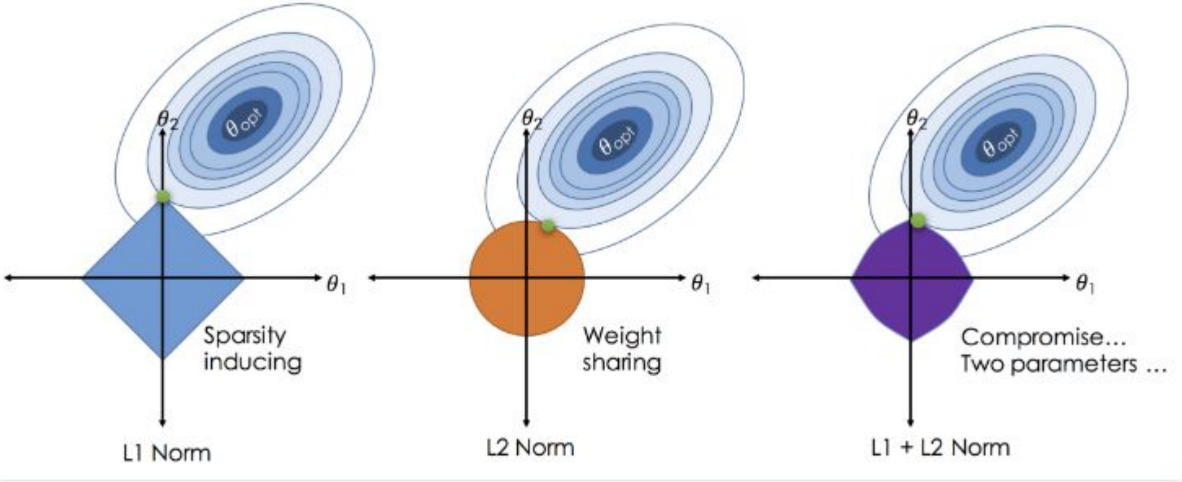
One Hot Encoding

- One Hot Encoding is like dummy variables
- To convert to One Hot Encoding you create a vector with length **n**
    - **n** is the number of different elements you can have (e.g. words, categories)
- If a variable is the nth option, it has a 1 for the nth element of the vector and 0’s everywhere else
- One Hot Encoding is *sparse*

Sparsity

$$\begin{bmatrix} apple \\ carrot \\ celery \\ carrot \end{bmatrix} \rightarrow \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

Sparsity

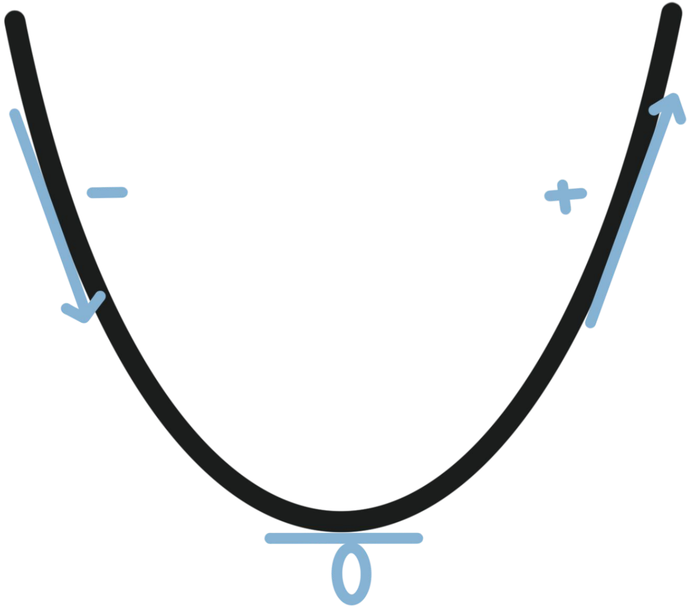


θ_2
θ_{opt}
θ_1
Sparsity inducing
L1 Norm
θ_2
θ_{opt}
θ_1
Weight sharing
L2 Norm
θ_2
θ_{opt}
θ_1
Compromise...
Two parameters ...
L1 + L2 Norm

Sparsity

- Sparsity refers to objects like matrices, tensors, or vectors, that have a lot of 0's
- The opposite of sparse is dense

Derivatives



Derivatives tell you the “instantaneous rate of change”
As you change a variable, how does the output change?

Derivatives

- Derivatives tell us the rate of change of a function
- Derivatives are **0** are **minima**, **maxima**, and **saddle points**
- Derivatives are **positive** when the function is **increasing**
- Derivatives are **negative** when the function is **decreasing**

The Chain-Rule

$$f(x) = cos(x)$$

$$g(x) = x^2$$

$$f(g(x)) = cos(x^2)$$

If we want to know how changing **x** affects **f(g(x))** we *first* need to think about how changing **x** affects **g(x)** *then* how changing **g(x)** affects **f(g(x))**

$$\frac{\partial f(g(x))}{\partial x} = \frac{\partial f}{\partial g}\frac{\partial g}{\partial x}$$

The Chain Rule

- To calculate the derivative of composite functions, we need to think about how changing the variable changes the inner part, then how changing the inner part changes the outer part

Partial Derivatives

$$f(x, y) = x^2 + xy + y^2$$

$$\frac{\partial f}{\partial y} = x + 2y \qquad \frac{\partial f}{\partial x} = 2x + y$$

For a function of with multiple variables, how does the *function change* when you change a *single* variable

Partial Derivatives

- Partial Derivatives tell us how a multivariable function changes with respect to a *single* variable (holding all other variables constant)

Gradient

$$f(x, y) = x^2 + xy + y^2$$

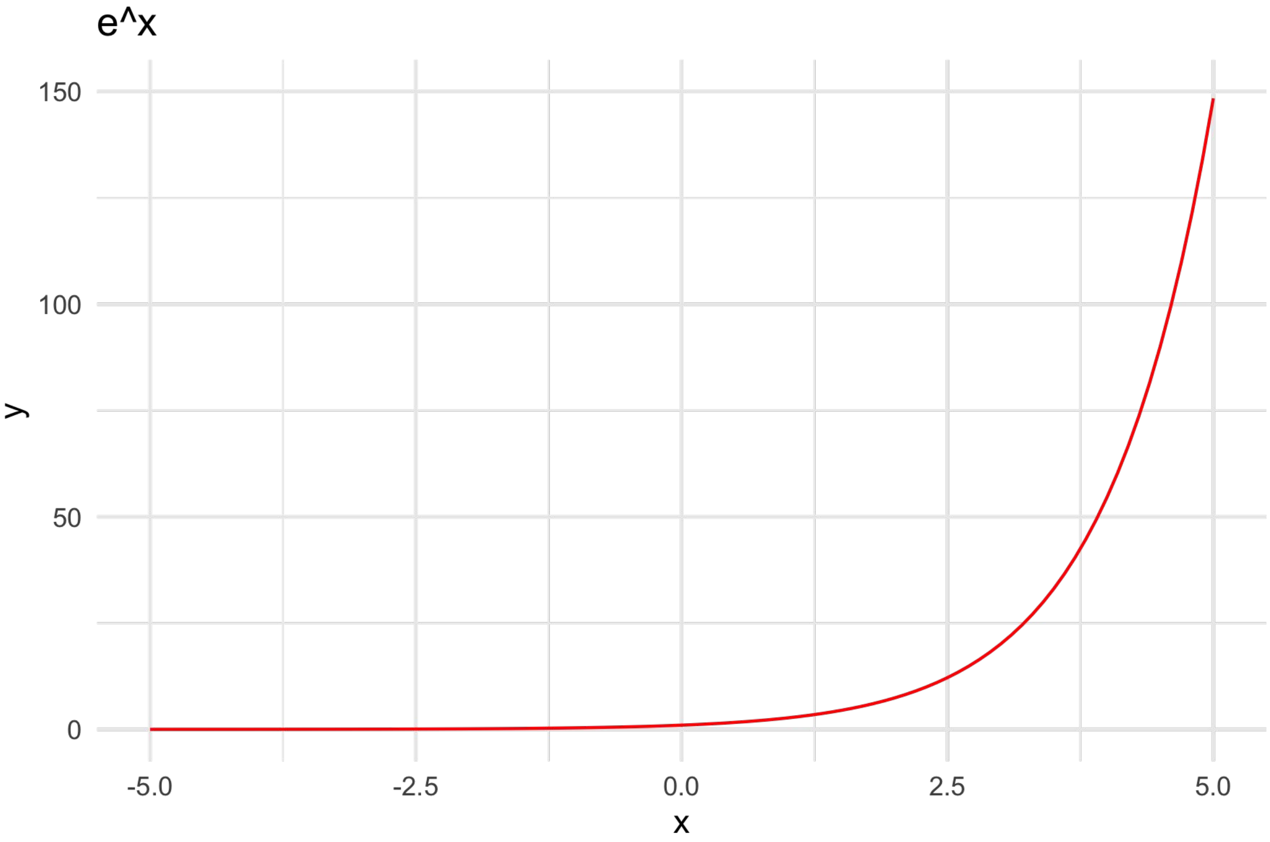
$$\frac{\partial f}{\partial y} = x + 2y \qquad \frac{\partial f}{\partial x} = 2x + y$$

$$\boxed{\nabla f(x, y) = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix}}$$

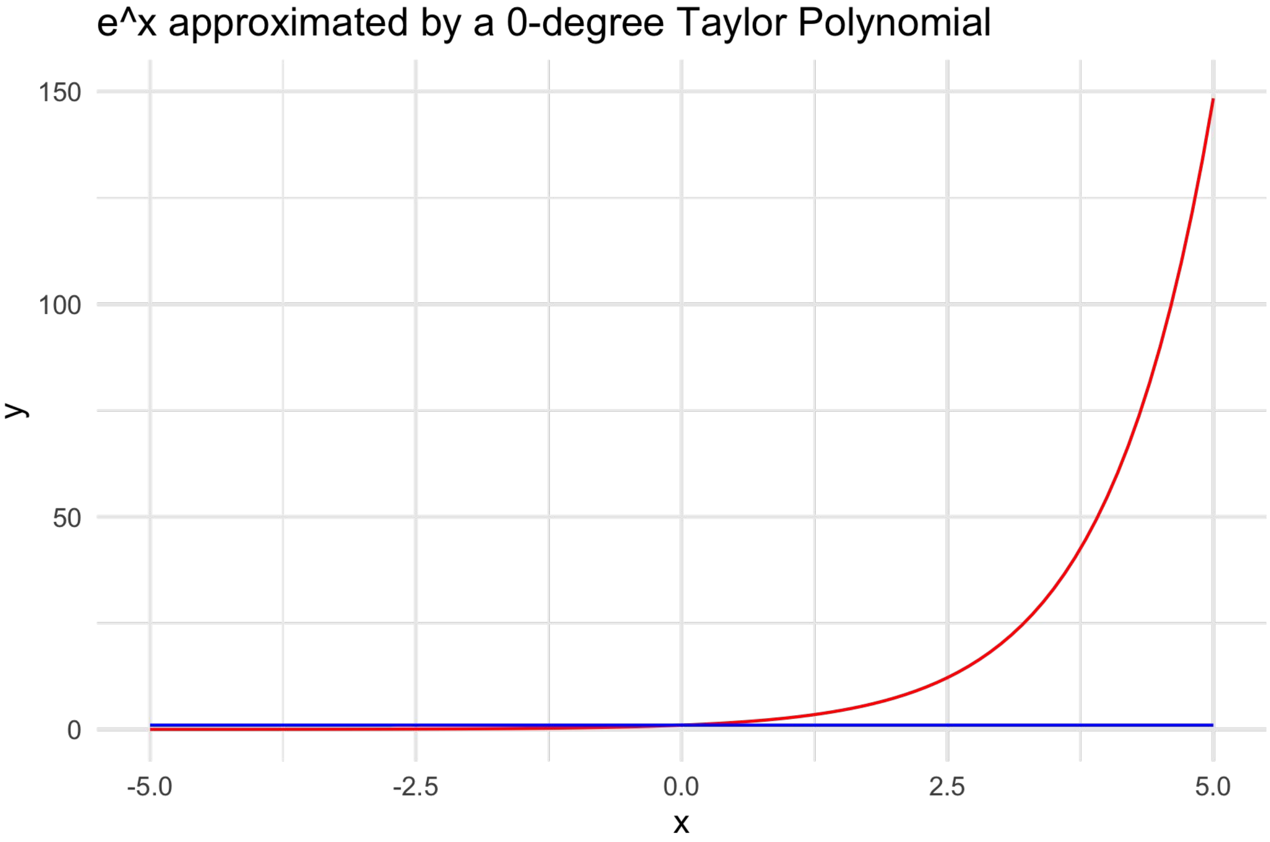
Gradient

- A Gradient is a vector of partial derivatives
- it tell us how a multivariable function changes with respect to a *each* variable (holding all other variables constant)

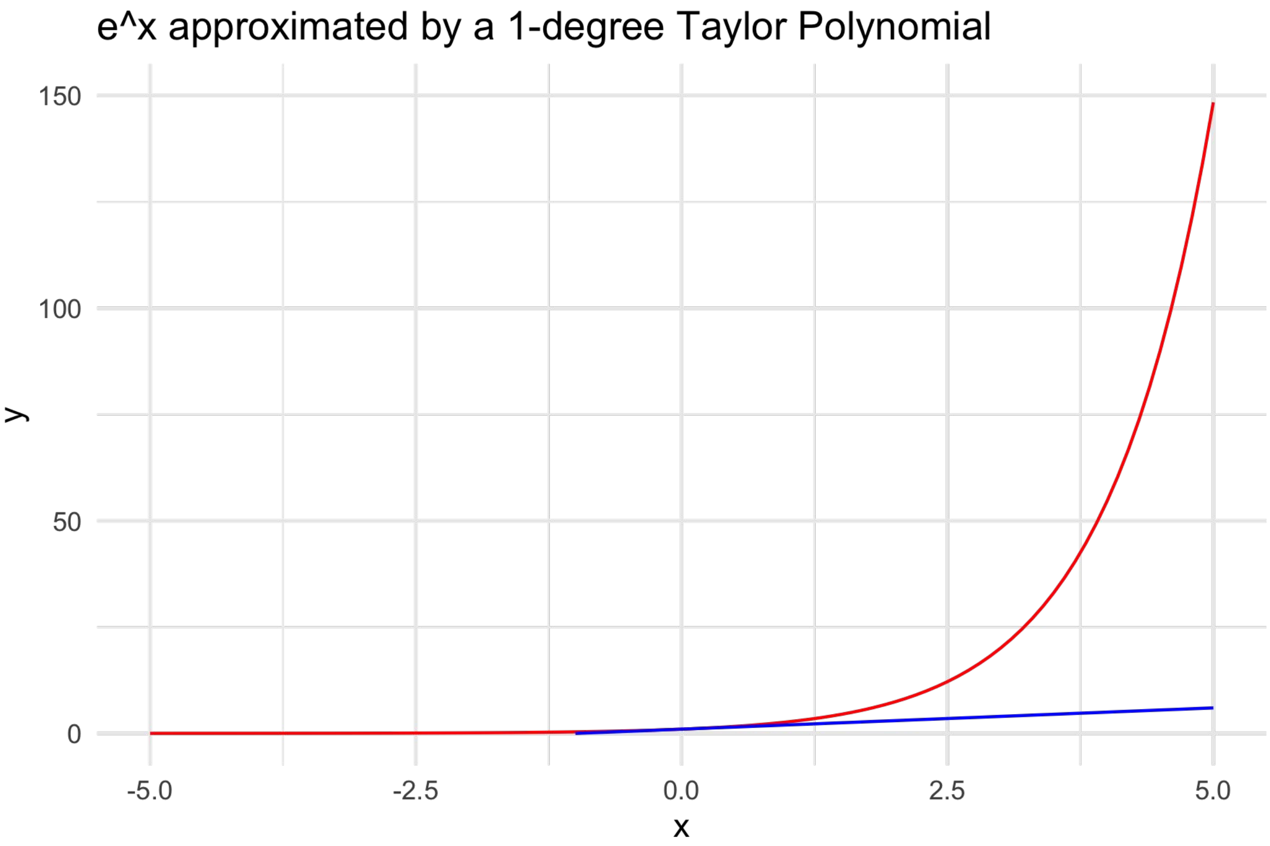
Taylor Series (visually)



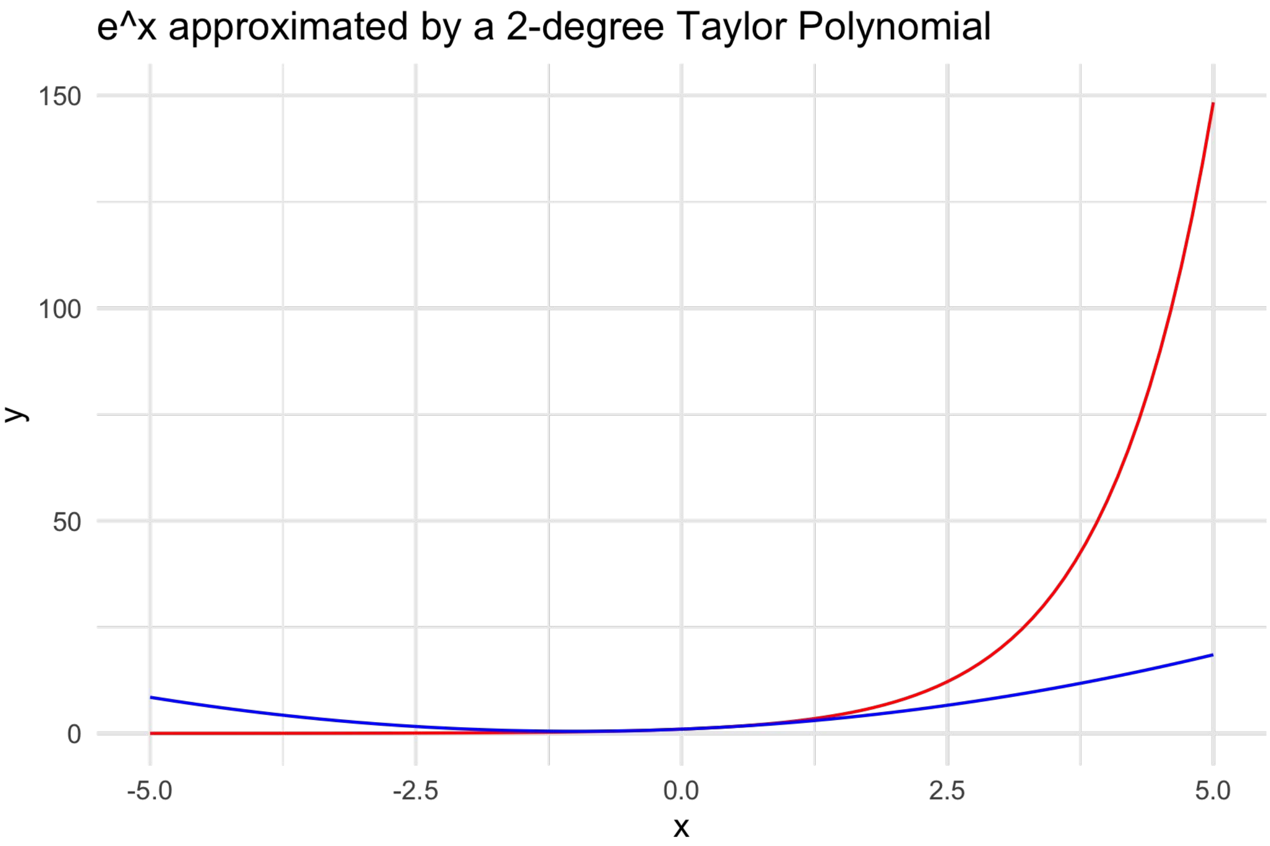
Taylor Series (visually)



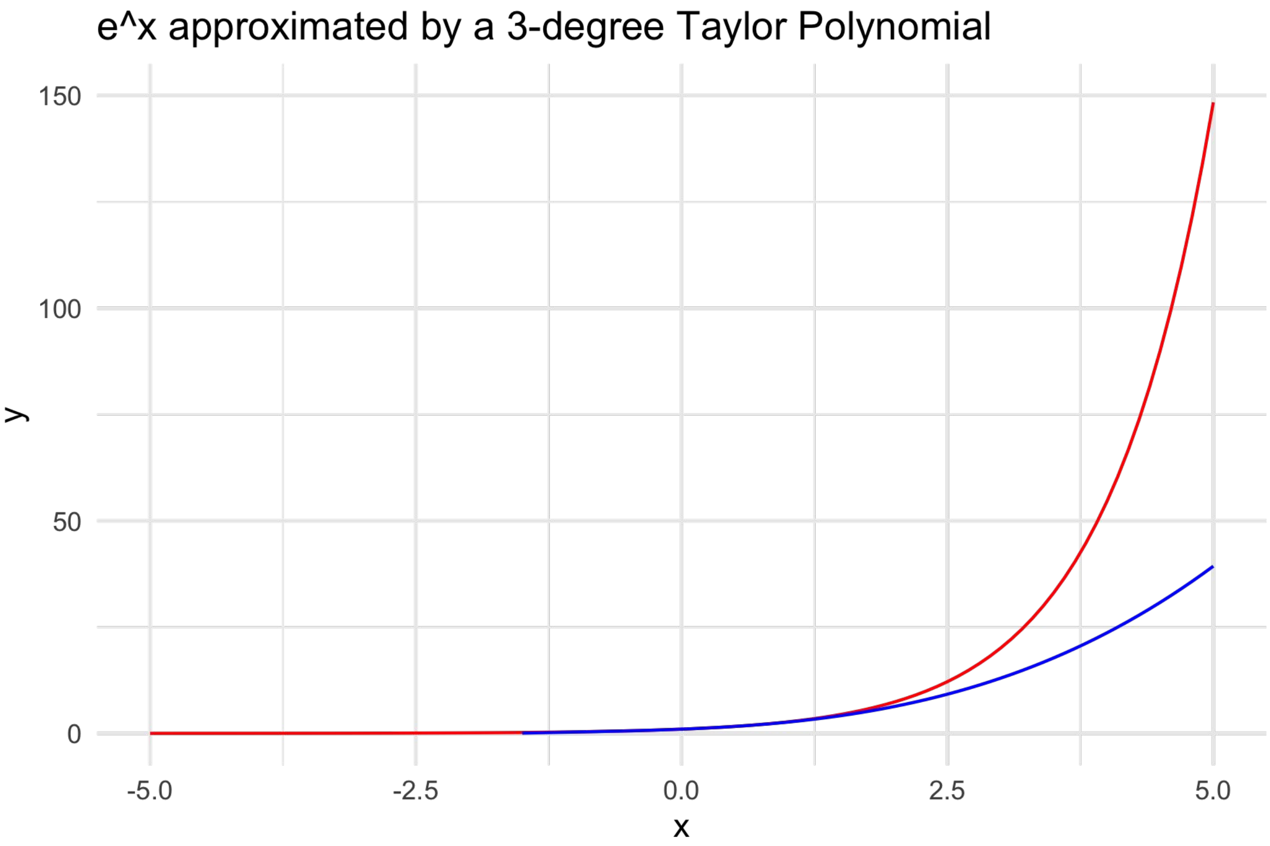
Taylor Series (visually)



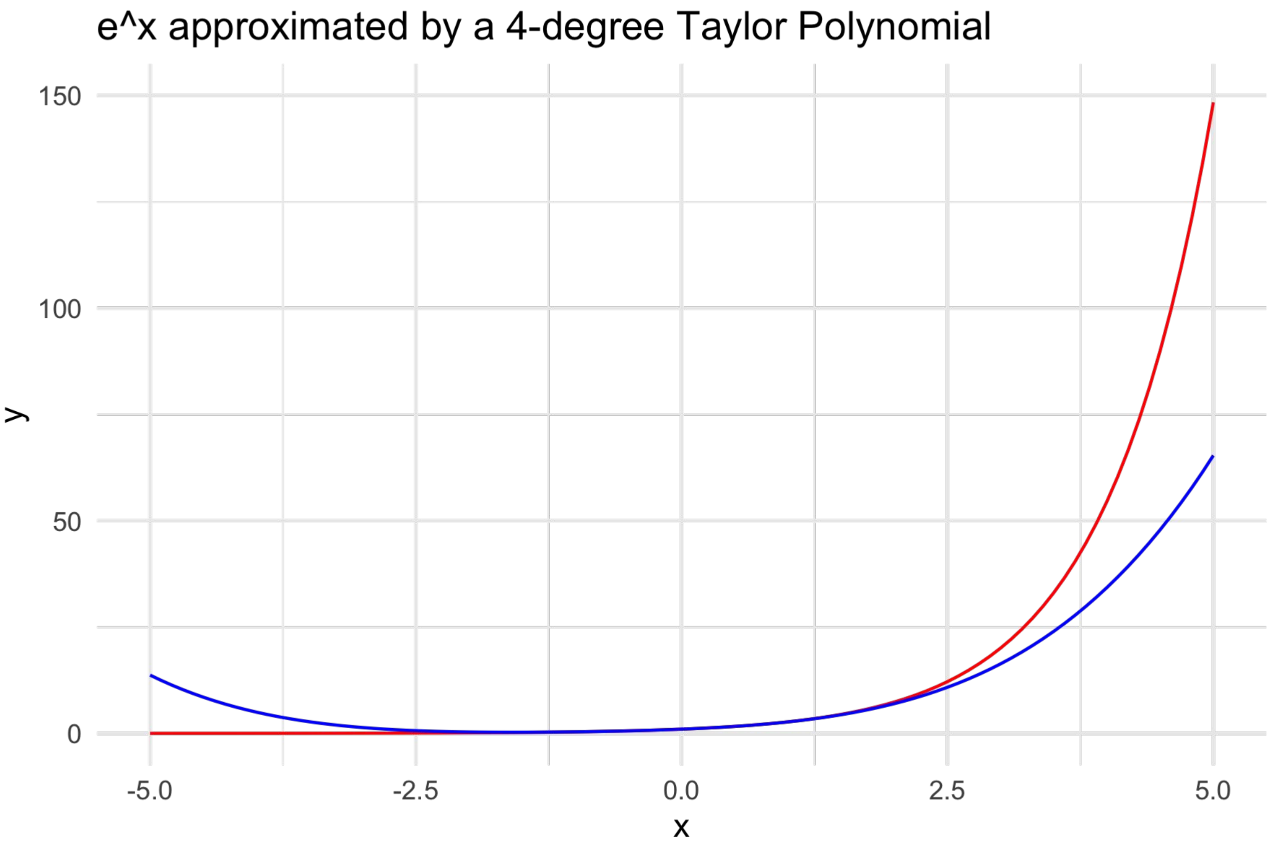
Taylor Series (visually)



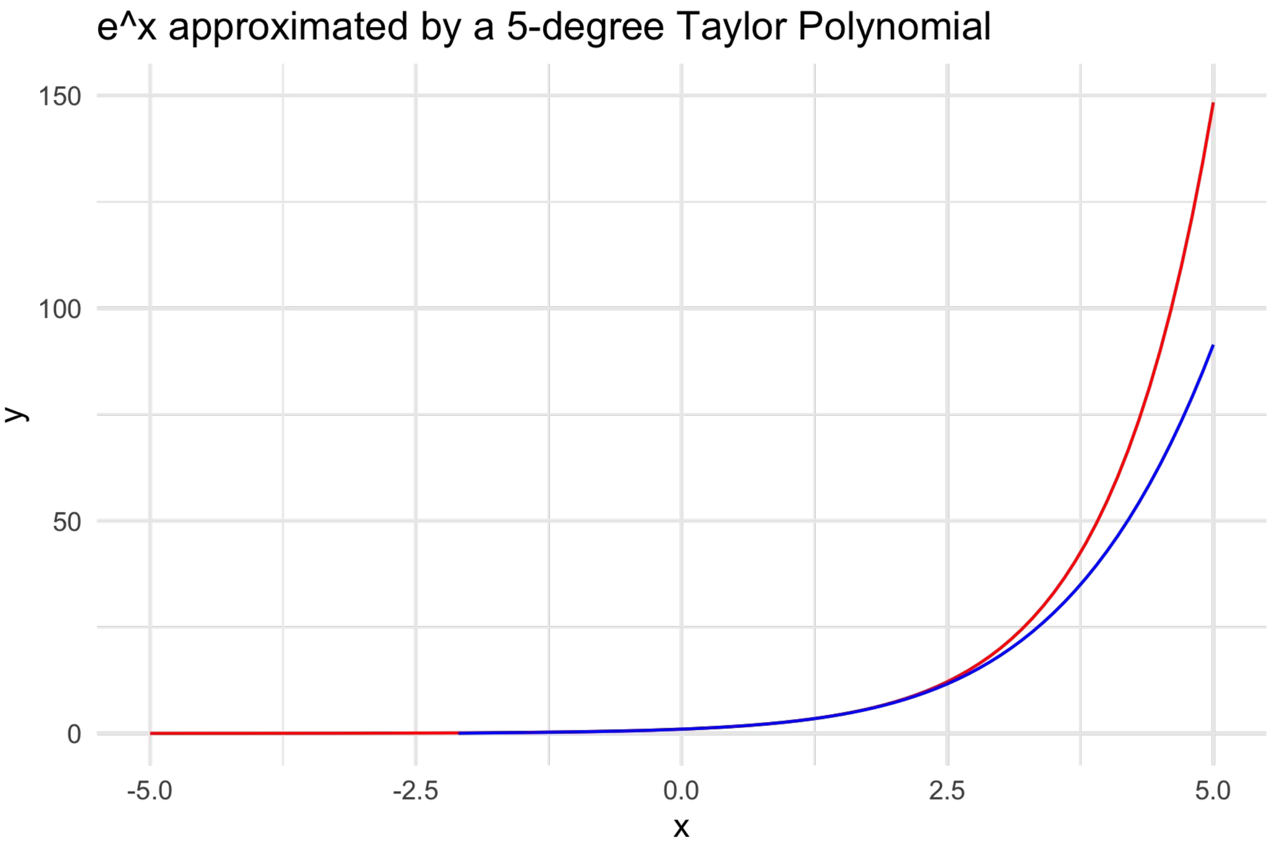
Taylor Series (visually)



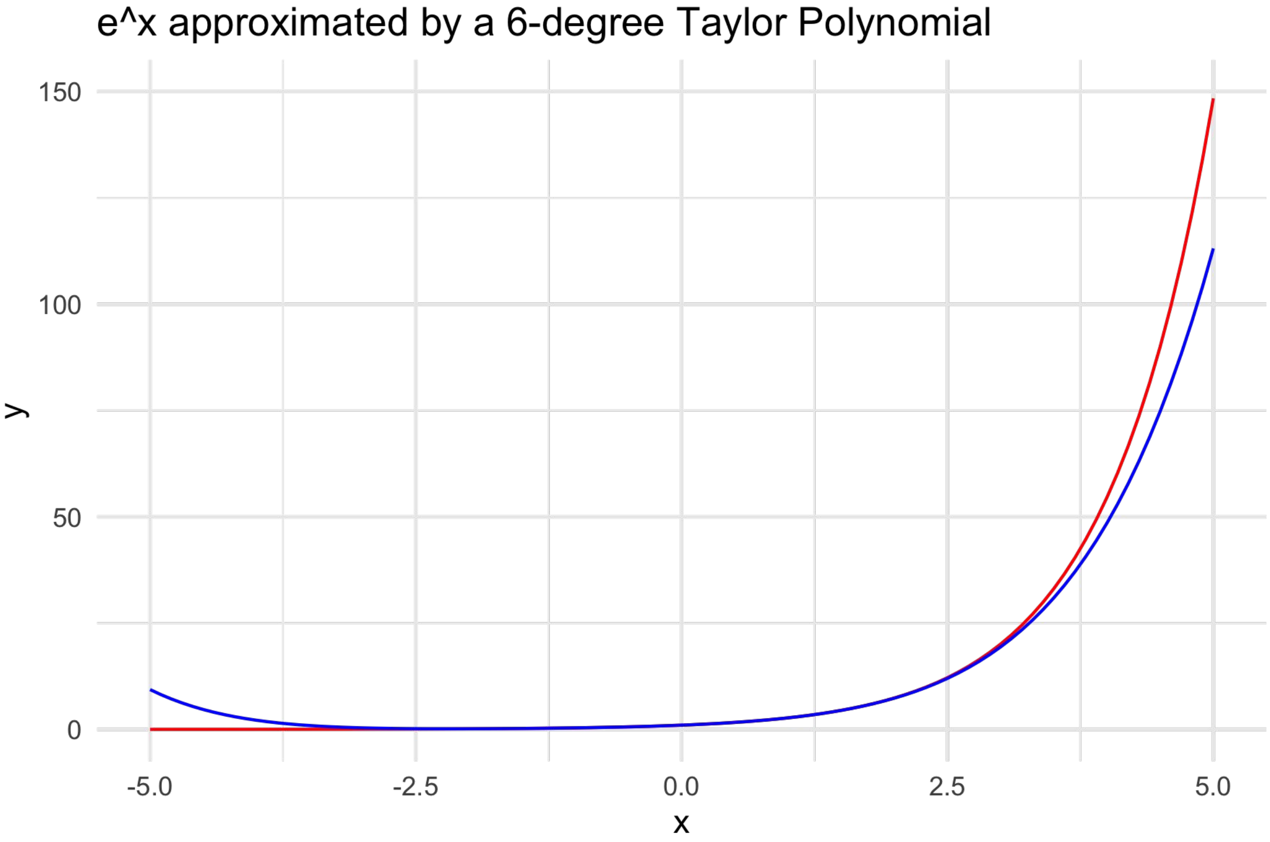
Taylor Series (visually)



Taylor Series (visually)



Taylor Series (visually)



Taylor Series

- Is there a simpler function that approximates f(x)?

  Yes!

$$\sum_{n=0}^{\infty} \frac{f^n(a)}{n!}(x-a)^n = f(a)+\frac{f'(a)}{1}(x-a)+\frac{f''(a)}{2!}(x-a)^2+\ldots+\frac{f^\infty(a)}{\infty!}(x-a)^\infty$$

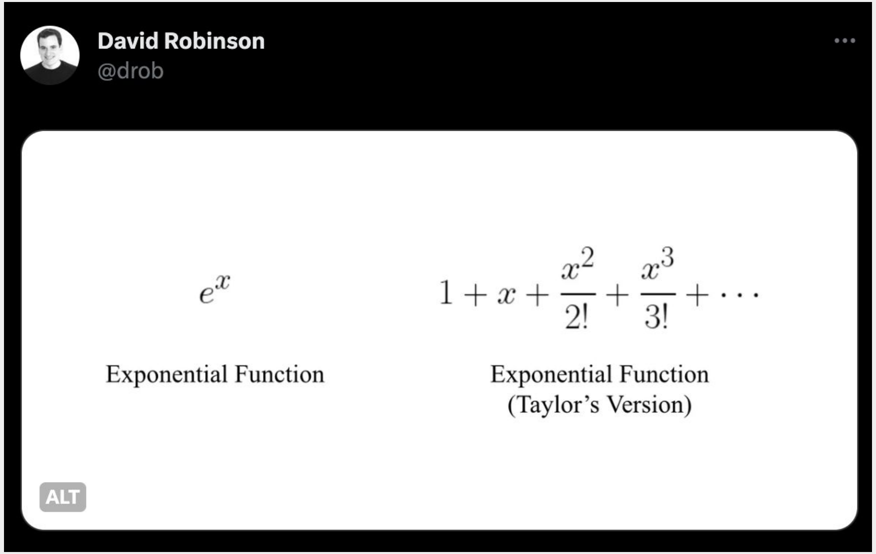
Taylor Series

$$\sum_{n=0}^{\infty} \frac{f^n(a)}{n!}(x-a)^n = f(a) + \frac{f'(a)}{1}(x-a) + \frac{f''(a)}{2!}(x-a)^2 + \ldots + \frac{f^{\infty}(a)}{\infty!}(x-a)^{\infty}$$

$$\sum_{n=0}^{\infty} \frac{f^n(a)}{n!}(x-a)^n = \boxed{\underbrace{f(a) + \frac{f'(a)}{1}(x-a) + \frac{f''(a)}{2!}(x-a)^2}_{\text{use just this for approximation of f(x))}}} \cancel{+ \ldots + \frac{f^{\infty}(a)}{\infty!}(x-a)^{\infty}}$$

Taylor Series

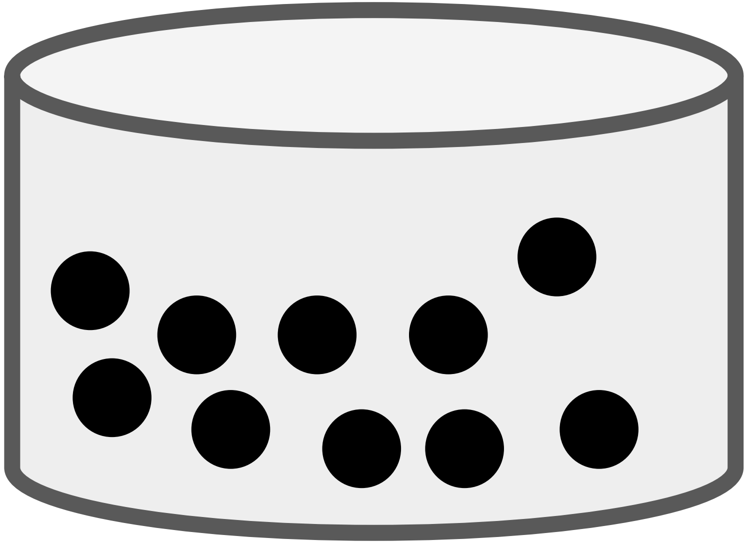
- Taylor Series are ways to **re-write a function** using its derivatives
- A Taylor Series is the sum of an **infinite** number of terms
- We can use just a few of these terms if we want an **approximation** of the function



Entropy

$$H(p) = -\sum_{i}^{N} p(x_i) * log(p(x_i))$$

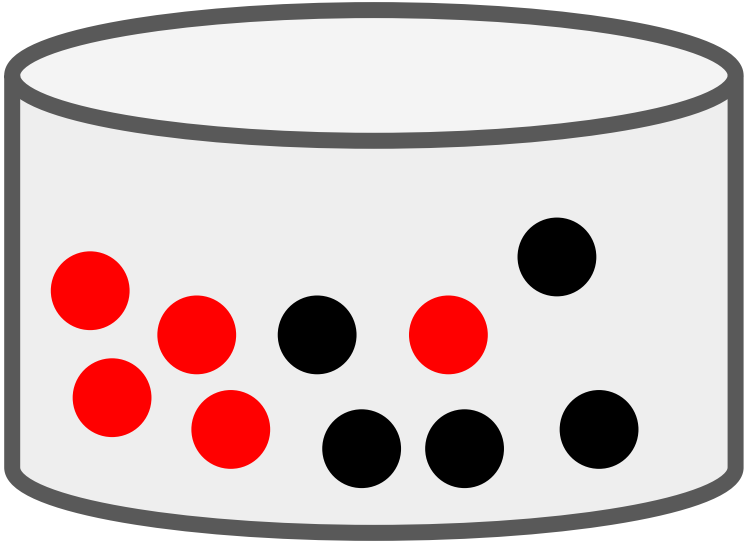
Measure of surprise



Entropy

$$H(p) = -\sum_{i}^{N} p(x_i) * log(p(x_i))$$

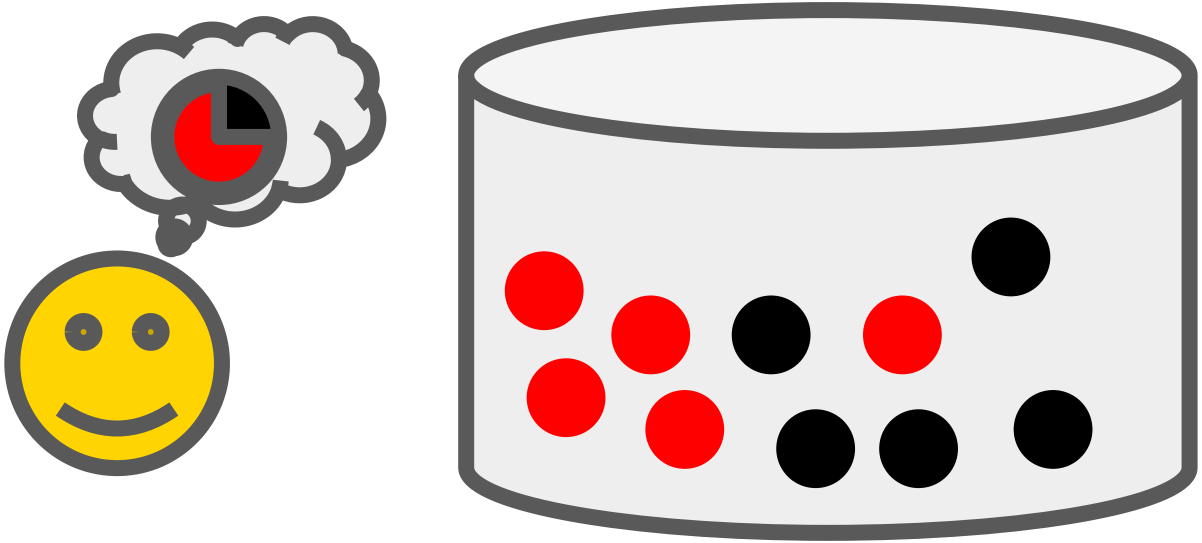
Measure of surprise



Cross-Entropy

$$H(p, q) = -\sum_{i}^{N} p(x_i) * log(q(x_i))$$

Measure of surprise for events with probability p, if you believe the events have probability q



KL Divergence

$$D_{KL}(p||q) = \underbrace{H(p,q)}_{\text{cross entropy}} - \overbrace{H(p)}^{\text{entropy}}$$

$$D_{KL}(p||q) = \sum_{i}^{N} p(x_i) * (log(p(x_i)) - log(q(x_i)))$$

A similarity metric between two distributions **p** and **q**.

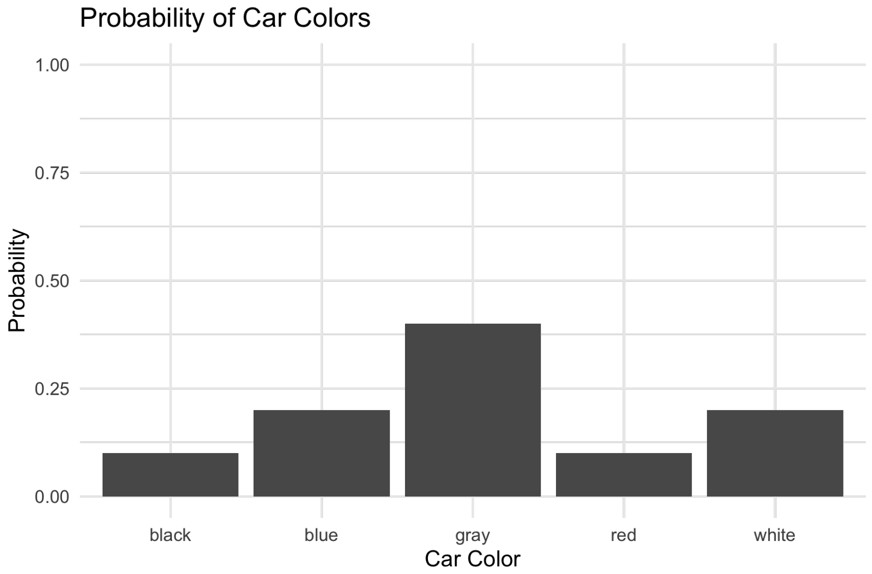
How much information is lost when we use **q** to approximate **p**.

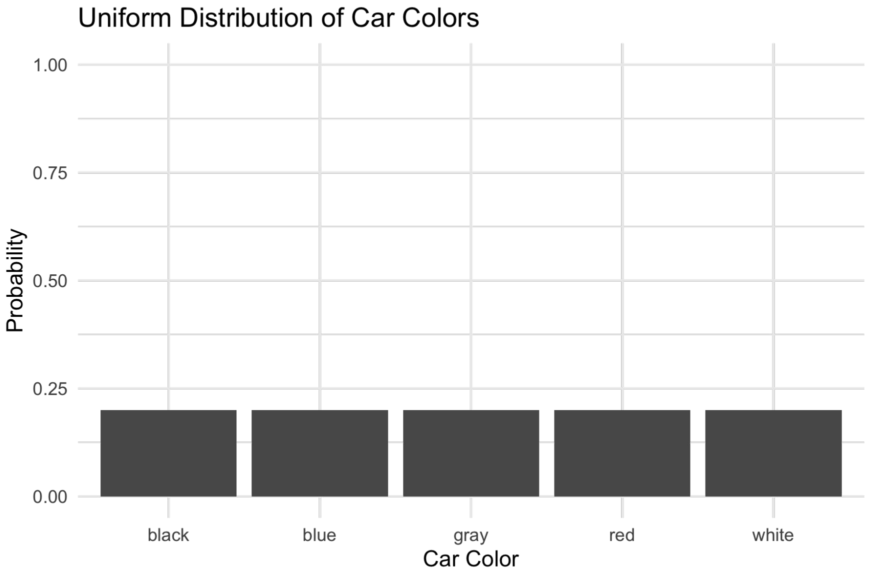
How much more surprised I expect to be if I have incorrect information.

KL Divergence

$$D_{KL}(p||q) = \sum_{i}^{N} p(x_i) * (log(p(x_i)) - log(q(x_i)))$$

	black	blue	gray	red	white
p(x)	0.1	0.2	0.4	0.1	0.2
q(x)	0.2	0.2	0.2	0.2	0.2





KL Divergence

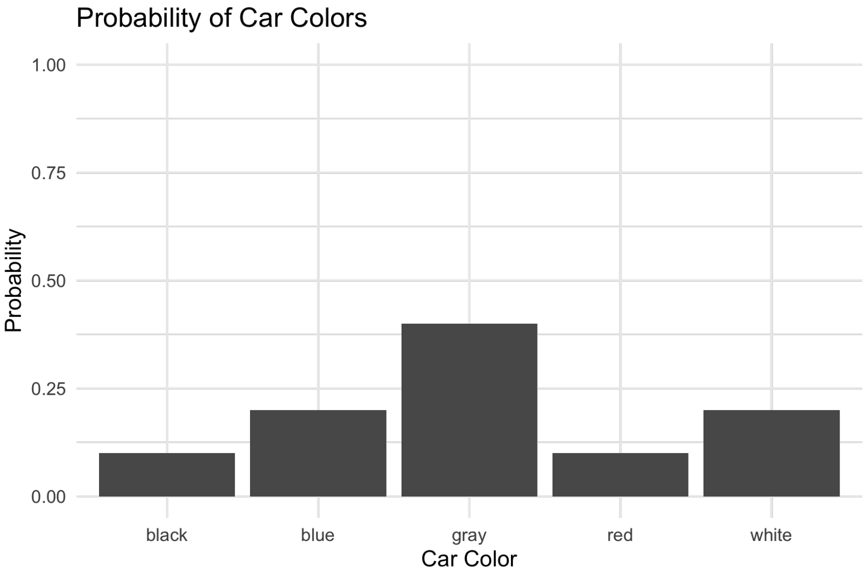
$$D_{KL}(p||q) = \sum_{i}^{N} p(x_i) * (log(p(x_i)) - log(q(x_i)))$$

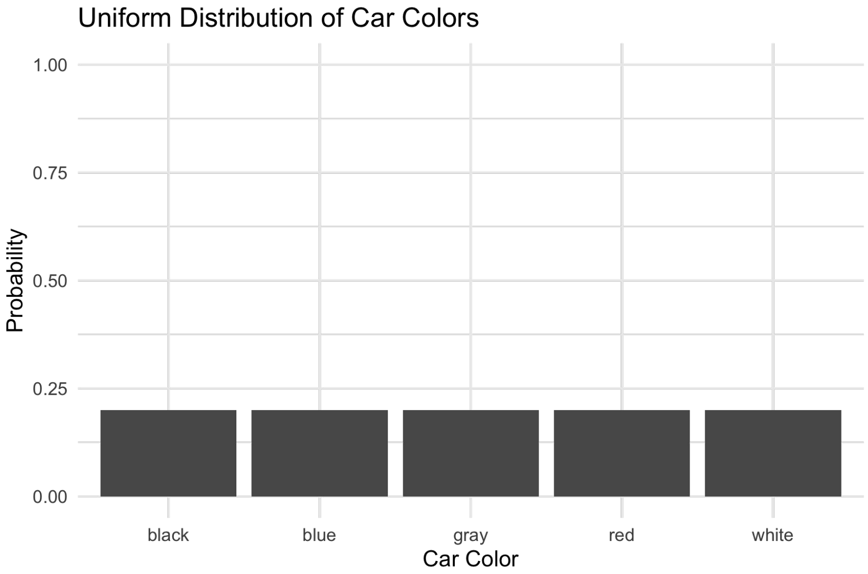
	black	blue	gray	red	white
p(x)	0.1	0.2	0.4	0.1	0.2
q(x)	0.2	0.2	0.2	0.2	0.2

KL Divergence

$$D_{KL}(p||q) = \sum_{i}^{N} p(x_i) * (log(p(x_i)) - log(q(x_i)))$$

	black	blue	gray	red	white
p(x)	0.1	0.2	0.4	0.1	0.2
q(x)	0.2	0.2	0.2	0.2	0.2





Entropy, Cross-Entropy, KL Divergence

- **Entropy** is a measure of chaos, or surprise. The more homogenous a group the lower entropy there is (remember Decision Trees?)
- **Cross-Entropy** can be thought of as a loss function measuring how close our predicted probabilities (0-1) are to the actual event (0 or 1)
- KL Divergences is a **non-symmetric** similarity metric that measures the similarity between two distributions

Lagrangians and Constrained Optimization

Optimize $f(x, y)$ subject to $g(x, y) = c$

1

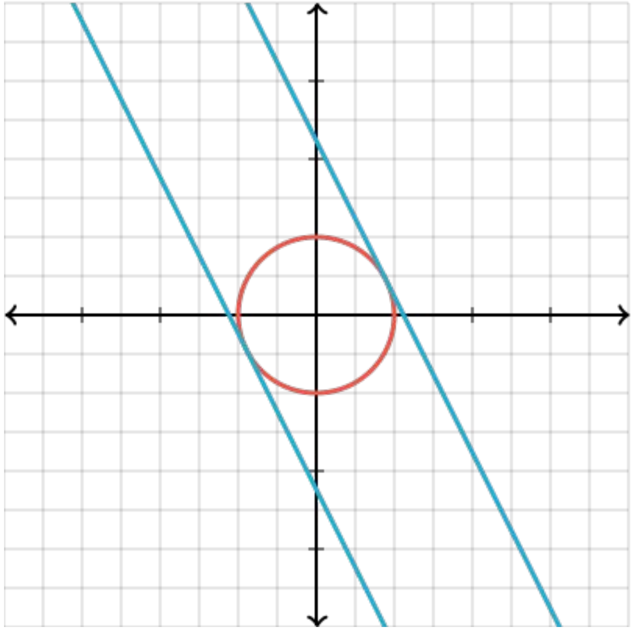


Image from Khan Academy's Lagrange Multiplier and Constrained Optimization Page: https://www.khanacademy.org/math/multivariable-calculus/applications-of-multivariable-derivatives/constrained-optimization/a/lagrange-multipliers-single-constraint

Lagrangians and Constrained Optimization

Optimize $f(x, y)$ subject to $g(x, y) = c$

2

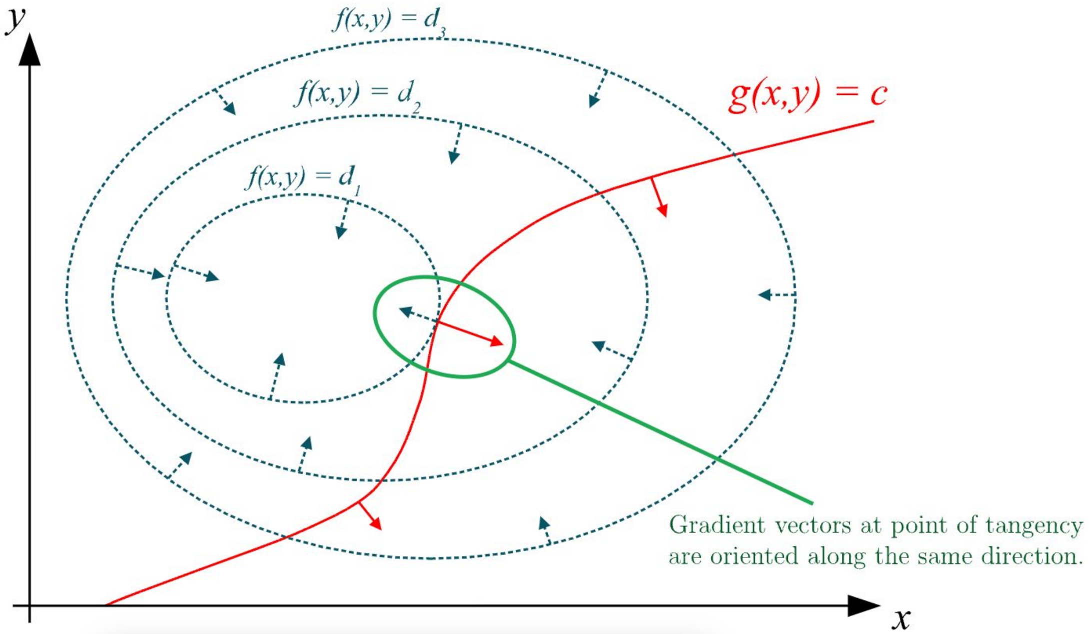


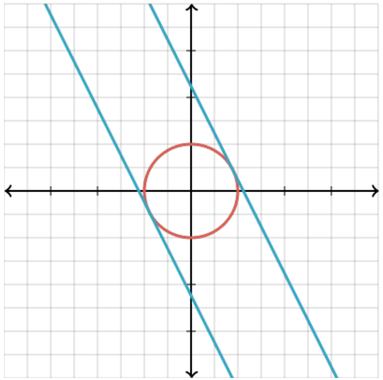
Image from Khan Academy's Lagrange Multiplier and Constrained Optimization Page: https://www.khanacademy.org/math/multivariable-calculus/applications-of-multivariable-derivatives/constrained-optimization/a/lagrange-multipliers-single-constraint

Lagrangians and Constrained Optimization

Optimize $f(x, y)$ subject to $g(x, y) = c$

3 $$\nabla f(x_0, y_0) = \lambda \nabla g(x_0, y_0)$$

Lagrangians and Constrained Optimization



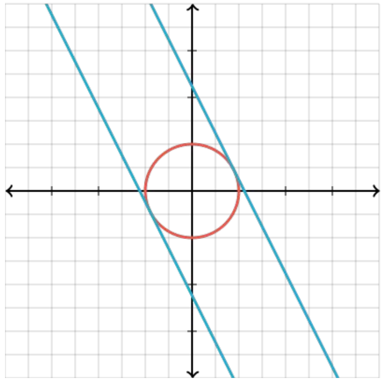
Optimize $f(x, y)$ subject to $g(x, y) = c$

3 $\nabla f(x_0, y_0) = \lambda \nabla g(x_0, y_0)$

$$f(x, y) = 2x + y \qquad g(x, y) = x^2 + y^2 = 1$$

$$\nabla f(x, y) = \begin{bmatrix} \frac{\partial}{\partial x} 2x + y \\ \frac{\partial}{\partial y} 2x + y \end{bmatrix} = \begin{bmatrix} 2 \\ 1 \end{bmatrix} \qquad \nabla g(x, y) = \begin{bmatrix} \frac{\partial}{\partial x} x^2 + y^2 \\ \frac{\partial}{\partial y} x^2 + y^2 \end{bmatrix} = \begin{bmatrix} 2x \\ 2y \end{bmatrix}$$

Lagrangians and Constrained Optimization



Optimize $f(x, y)$ subject to $g(x, y) = c$

3 $\nabla f(x_0, y_0) = \lambda \nabla g(x_0, y_0)$

$f(x, y) = 2x + y$

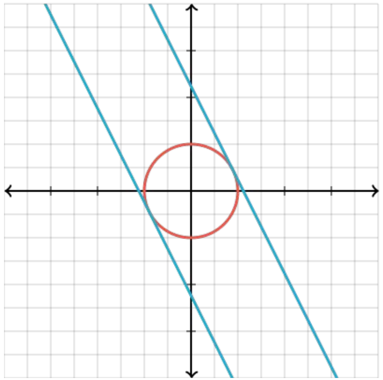
$g(x, y) = x^2 + y^2 = 1$

$$\nabla f(x, y) = \begin{bmatrix} \frac{\partial}{\partial x} 2x + y \\ \frac{\partial}{\partial y} 2x + y \end{bmatrix} = \begin{bmatrix} 2 \\ 1 \end{bmatrix}$$

$$\nabla g(x, y) = \begin{bmatrix} \frac{\partial}{\partial x} x^2 + y^2 \\ \frac{\partial}{\partial y} x^2 + y^2 \end{bmatrix} = \begin{bmatrix} 2x \\ 2y \end{bmatrix}$$

Lagrangians and Constrained Optimization

Optimize $f(x, y)$ subject to $g(x, y) = c$



3 $\nabla f(x_0, y_0) = \lambda \nabla g(x_0, y_0)$

$f(x, y) = 2x + y$

$g(x, y) = x^2 + y^2 = 1$

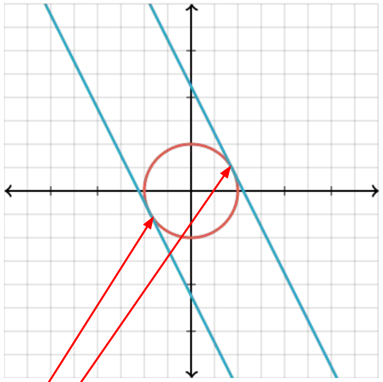
$$\nabla f(x, y) = \begin{bmatrix} \frac{\partial}{\partial x} 2x + y \\ \frac{\partial}{\partial y} 2x + y \end{bmatrix} = \begin{bmatrix} 2 \\ 1 \end{bmatrix}$$

$$\nabla g(x, y) = \begin{bmatrix} \frac{\partial}{\partial x} x^2 + y^2 \\ \frac{\partial}{\partial y} x^2 + y^2 \end{bmatrix} = \begin{bmatrix} 2x \\ 2y \end{bmatrix}$$

$$\boxed{\begin{bmatrix} 2 \\ 1 \end{bmatrix} = \lambda_0 \begin{bmatrix} 2x_0 \\ 2y_0 \end{bmatrix}}$$

Lagrangians and Constrained Optimization

Optimize $f(x, y)$ subject to $g(x, y) = c$



3 $\nabla f(x_0, y_0) = \lambda \nabla g(x_0, y_0)$

$f(x, y) = 2x + y$

$g(x, y) = x^2 + y^2 = 1$

$$\nabla f(x, y) = \begin{bmatrix} \frac{\partial}{\partial x} 2x + y \\ \frac{\partial}{\partial y} 2x + y \end{bmatrix} = \begin{bmatrix} 2 \\ 1 \end{bmatrix} \qquad \nabla g(x, y) = \begin{bmatrix} \frac{\partial}{\partial x} x^2 + y^2 \\ \frac{\partial}{\partial y} x^2 + y^2 \end{bmatrix} = \begin{bmatrix} 2x \\ 2y \end{bmatrix}$$

$$\boxed{\begin{bmatrix} 2 \\ 1 \end{bmatrix} = \lambda_0 \begin{bmatrix} 2x_0 \\ 2y_0 \end{bmatrix}}$$

Lagrangians and Constrained Optimization

Optimize $f(x, y)$ subject to $g(x, y) = c$

4

$$\mathcal{L}(\boxed{x}, \boxed{y}, \boxed{\lambda}) = f(\boxed{x}, \boxed{y}) - \boxed{\lambda}(g(\boxed{x}, \boxed{y}) - c)$$

$$\mathcal{L}(\boxed{x}, \boxed{y}, \boxed{\lambda}) = 2\boxed{x} + \boxed{y} - \boxed{\lambda}(\boxed{x}^2 + \boxed{y}^2 - 1)$$

For example

Lagrangians and Constrained Optimization

Optimize $f(x, y)$ subject to $g(x, y) = c$

4

$$\mathcal{L}(x, y, \lambda) = f(x, y) - \lambda(g(x, y) - c)$$

$$\frac{\partial \mathcal{L}}{\partial \lambda} = 0 - (g(x, y) - c) = -g(x, y) + c$$

$$\frac{\partial \mathcal{L}}{\partial \lambda} = -g(x, y) + c$$

Lagrangians and Constrained Optimization

Optimize $f(x, y)$ subject to $g(x, y) = c$

4

$$\mathcal{L}(\boxed{x}, \boxed{y}, \boxed{\lambda}) = f(\boxed{x}, \boxed{y}) - \boxed{\lambda}(g(\boxed{x}, \boxed{y}) - c)$$

$$\frac{\partial \mathcal{L}}{\partial \boxed{\lambda}} = 0 - (g(x, y) - c) = -g(x, y) + c$$

$$\frac{\partial \mathcal{L}}{\partial \boxed{\lambda}} = -g(x, y) + c = 0$$

Lagrangians and Constrained Optimization

Optimize $f(x, y)$ subject to $g(x, y) = c$

4

$$\mathcal{L}(x, y, \lambda) = f(x, y) - \lambda(g(x, y) - c)$$

$$\frac{\partial \mathcal{L}}{\partial \lambda} = 0 - (g(x, y) - c) = -g(x, y) + c$$

$$\frac{\partial \mathcal{L}}{\partial \lambda} = -g(x, y) + c = 0 \text{ when } g(x, y) = c$$

Lagrangians and Constrained Optimization

Optimize $f(x, y)$ subject to $g(x, y) = c$

4 $$\frac{\partial \mathcal{L}}{\partial \boxed{x}} f(x, y) - \lambda(g(x, y) - c)$$

$$f_x(x, y) - \lambda g_x(x, y)$$

Lagrangians and Constrained Optimization

Optimize $f(x, y)$ subject to $g(x, y) = c$

4 $$\frac{\partial \mathcal{L}}{\partial \boxed{x}} f(x, y) - \lambda(g(x, y) - c)$$

$$f_x(x, y) - \lambda g_x(x, y) = 0$$

Lagrangians and Constrained Optimization

Optimize $f(x, y)$ subject to $g(x, y) = c$

4 $$\frac{\partial \mathcal{L}}{\partial \boxed{x}} f(x, y) - \lambda(g(x, y) - c)$$

$$f_x(x, y) - \lambda g_x(x, y) = 0$$

$$f_x(x, y) = \lambda g_x(x, y)$$

Lagrangians and Constrained Optimization

Optimize $f(x, y)$ subject to $g(x, y) = c$

4 $$\frac{\partial \mathcal{L}}{\partial \boxed{y}} f(x, y) - \lambda(g(x, y) - c)$$

$$f_y(x, y) - \lambda g_y(x, y) = 0$$

$$f_y(x, y) = \lambda g_y(x, y)$$

Lagrangians and Constrained Optimization

Optimize $f(x, y)$ subject to $g(x, y) = c$

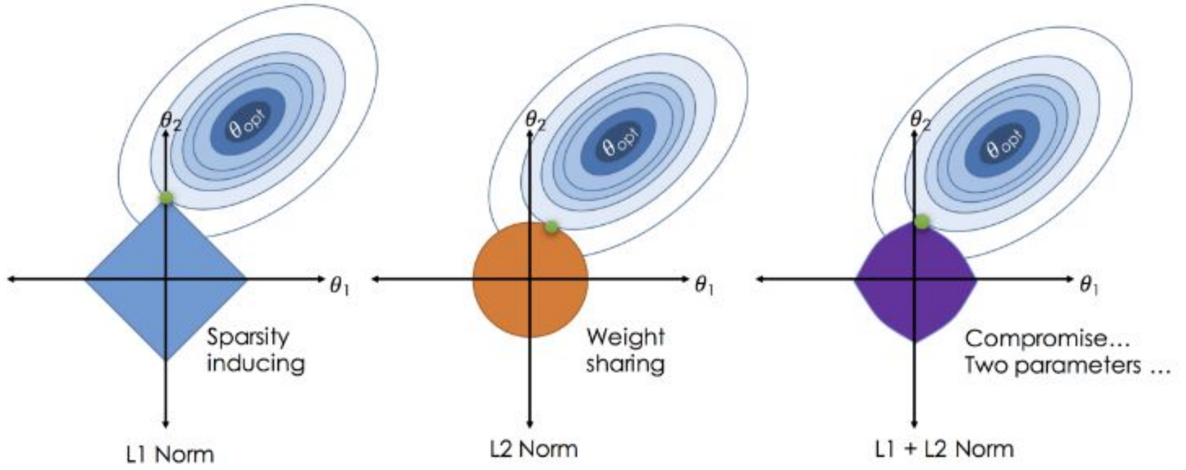
4

$$\mathcal{L}(x, y, \lambda) = f(x, y) - \lambda(g(x, y) - c)$$

$$\nabla \mathcal{L} = \begin{bmatrix} \frac{\partial \mathcal{L}}{\partial \lambda} \\ \frac{\partial \mathcal{L}}{\partial x} \\ \frac{\partial \mathcal{L}}{\partial y} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

Lagrangians and Constrained Optimization

Optimize $f(x, y)$ subject to $g(x, y) = c$



Lagrangians

- Lagrangians (and Lagrangian Multipliers) are a way to do **constrained optimization**
- Constrained Optimization is finding the min (or max) of a function within some boundary
- Because we know that the **optima** of a function (with constraints) occur at places where the two functions (function and constraint) are **tangent**, we can find where their **gradients are parallel** and that will be an optima!
- **Lagrangians** are a clever way of solving for these points by shoving them all into an equation and setting the gradient to 0
- The variable λ is called a Lagrangian Multiplier and represents the scaling factor between the two gradients
    - Can be interpreted as how much your function will change as you change your constraint