REGRESSION

How do we learn from data?

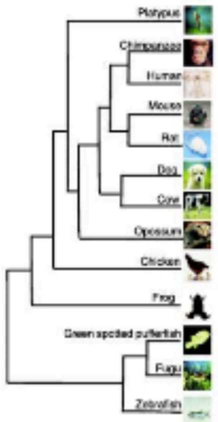
1
5
3.6
1
3
-1

Training set

Learning algorithm

Model h → Desired output **y**

Training phase

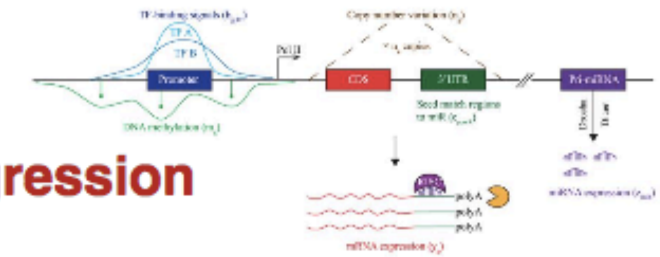


Generative Model

Maximum Likelihood Estimator

Image from http://physrev.physiology.org/content/89/3/921

Regression



Li *et al*. PLoS Comp Biol 10, e1003908 (2014)

Dynamic Bayesian Network

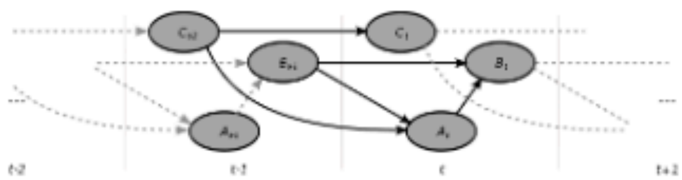
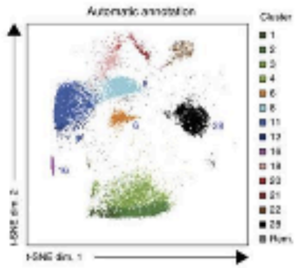


Image from https://en.wikipedia.org/wiki/Dynamic_Bayesian_network

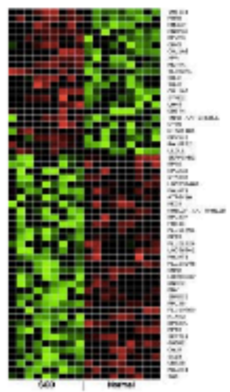
Dimensionality Reduction tSNE



Becher *et al*. Nature Immunology 15, 1181-1189 (2014)

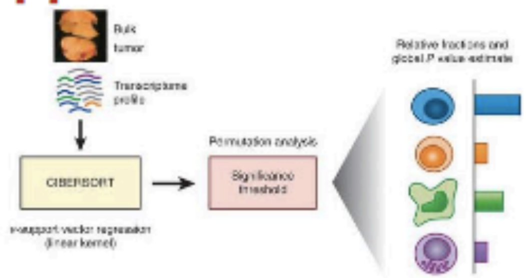
Clustering

Han *et al*. Nature Communication 8, 14238 (2017)



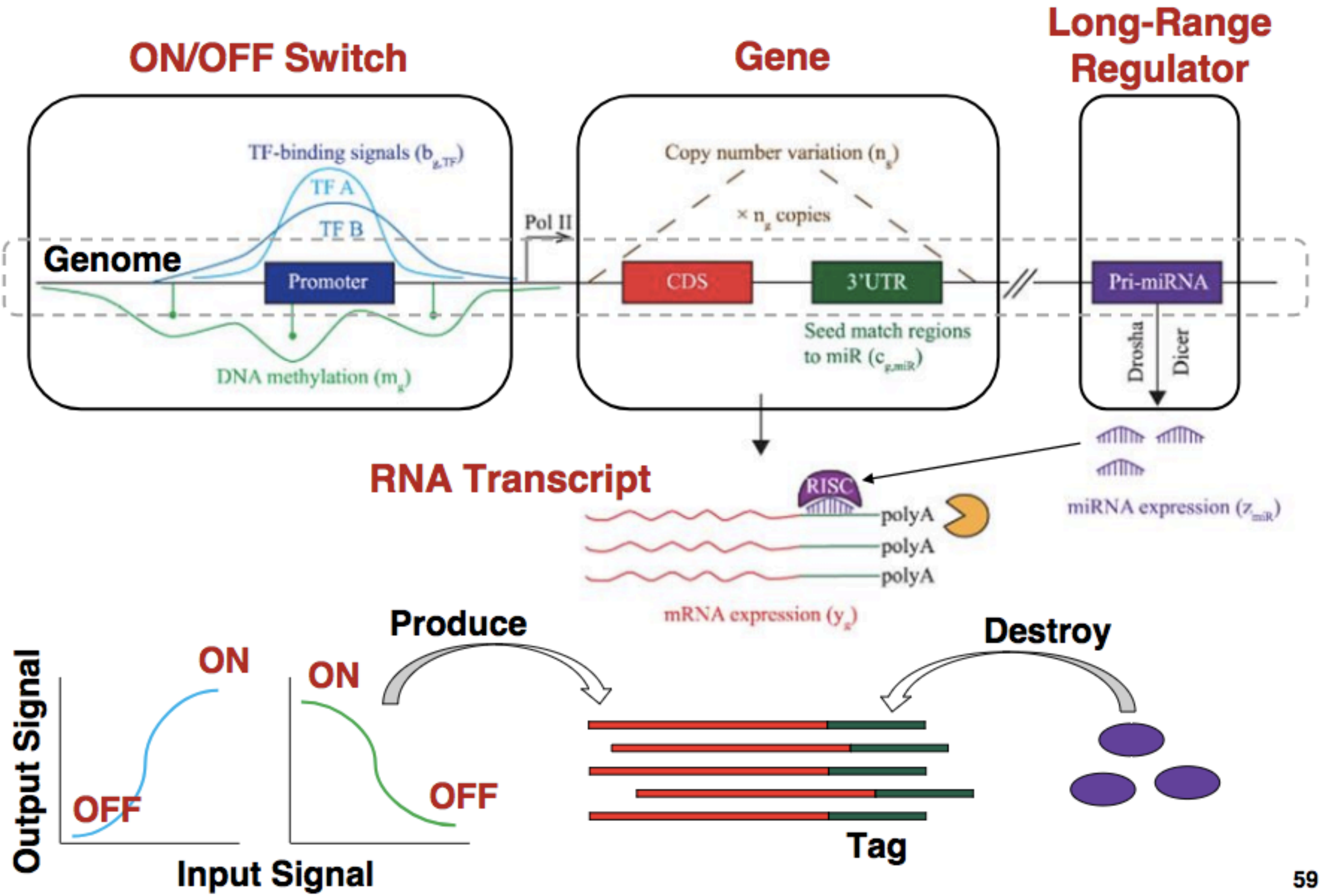
Klings *et al*. Physiological Genomics 21, 293-298 (2005)

Support Vector Machine

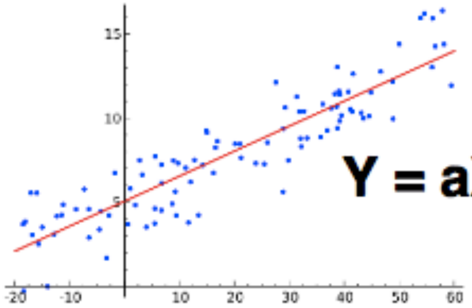


Newman *et al*. Nature Methods 12, 453-457 (2015)

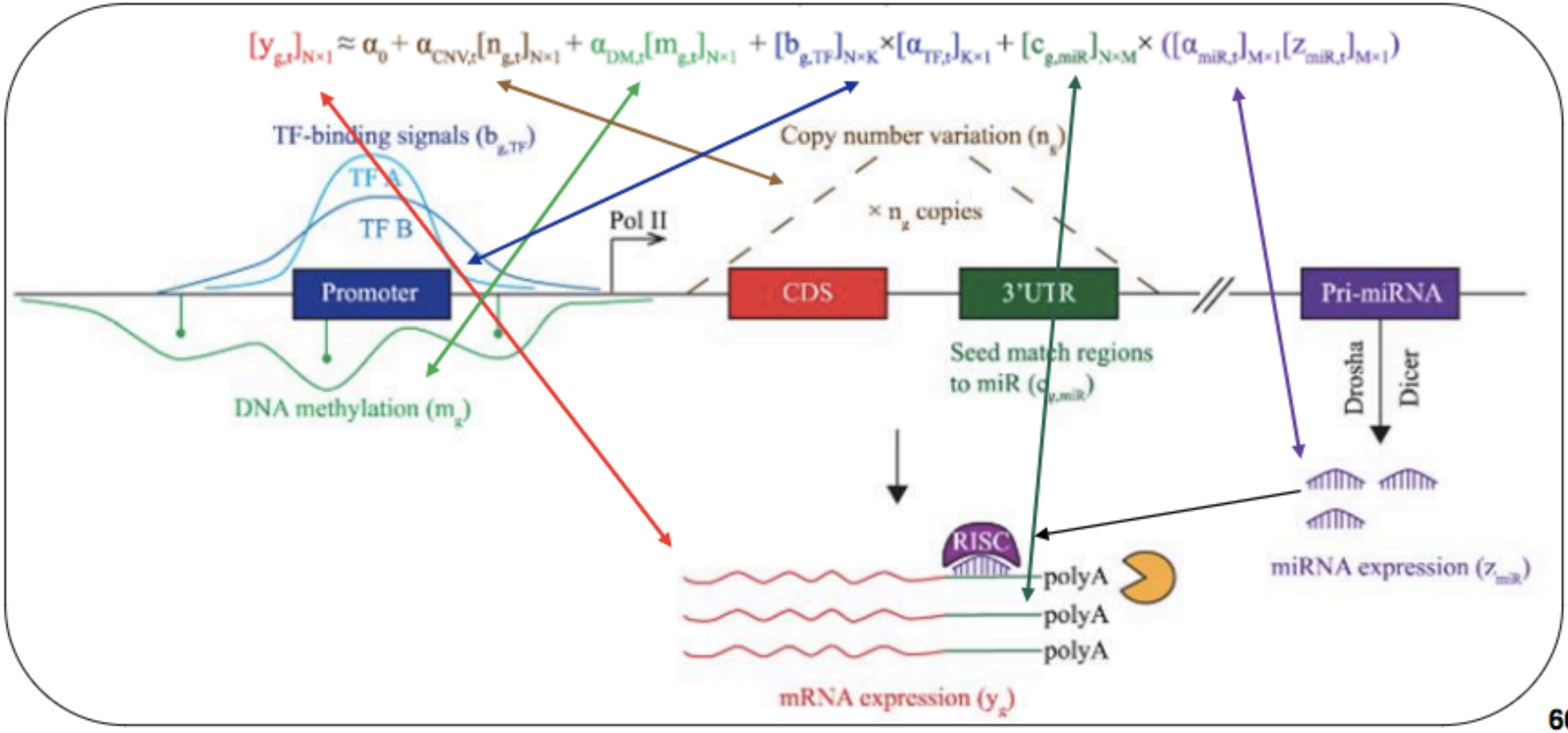
Regulation Of Gene Expression



Regression Model For Gene Expression I



$Y = aX + b \Rightarrow Y = a_0 + a_1X_1 + a_2X_2 + \ldots + a_nX_n$



Adapted from Li *et al.* PLoS Comp Biol 10, e1003908 (2014)

Let's look at an easier example

Let’s look at an easier example

https://soclaimon.wordpress.com/2015/07/24/%E0%B9%82%E0%B8%A1%E0%B9%80%E0%B8%94%E0%B8%A5%E0%B8%99%E0%B9%89%E0%B8%B33%E0%B8%A2%E0%B8%B8%E0%B8%84%E0%B8%A1%E0%B8%B2%E0%B8%A3%E0%B9%8C%E0%B8%84-%E0%B8%9B%E0%B8%B9/

Predicting amount of rainfall

https://esan108.com/%E0%B8%9E
%E0%B8%A3%E0%B8%B0%E0%B9%82%E0%B8%84%E0%B8%81%E0%B8%B4%E0%B8%99%E0%B8%AD
%E0%B8%B0%E0%B9%84%E0%B8%A3-%E0%B8%AB
%E0%B8%A1%E0%B8%B2%E0%B8%A2%E0%B8%96%E0%B8%B6%E0%B8%87.html

Predicting amount of rainfall

Cloth	Corn	Grass	Water	Beer	Rainfall
4	6	3	10	0	**76950**
5	1	0	0	7	**30234**
6	0	3	5	7	**123456**
5	0	3	12	0	**89301**
4	3	0	6	7	**?**

We assume the input features have some correlation with the amount of rainfall.

Can we create a model that predict the amount of rainfall?

What is the output?

What is the input (features)?

Predicting the amount of rainfall

- The correlation can be positive or negative

Predicting the amount of rainfall

Cloth	Corn	Grass	Water	Beer	Rainfall
4	6	3	10	0	**76950**
5	1	0	0	7	**30234**
6	0	3	5	7	**123456**
5	0	3	12	0	**89301**
4	3	0	6	7	**?**

Can we create a model that predict the amount of rainfall?

What is the output?

What is the input (features)?

Predicting the amount of rainfall

Cloth	Corn	Grass	Water	Beer	Rainfall
4	6	3	10	0	**76950**
5	1	0	0	7	**30234**
6	0	3	5	7	**123456**
5	0	3	12	0	**89301**
4	3	0	6	7	**?**

- $h_{\theta}(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_3 + \theta_4 x_4 + \theta_5 x_5$
- Where θs are the parameter of the model
- Xs are values in the table

(Linear) Regression

- $h_{\boldsymbol{\theta}}(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_3 + \theta_4 x_4 + \theta_5 x_5$
- θs are the parameter (or weights)
- We can rewrite

Assume x_0 is always 0

$$h_\theta(x) = \Sigma_{i=0}^{n} \theta_i x_i = \theta^T \mathbf{x}$$

- Notation: vectors are bolded
- Notation: vectors are column vectors

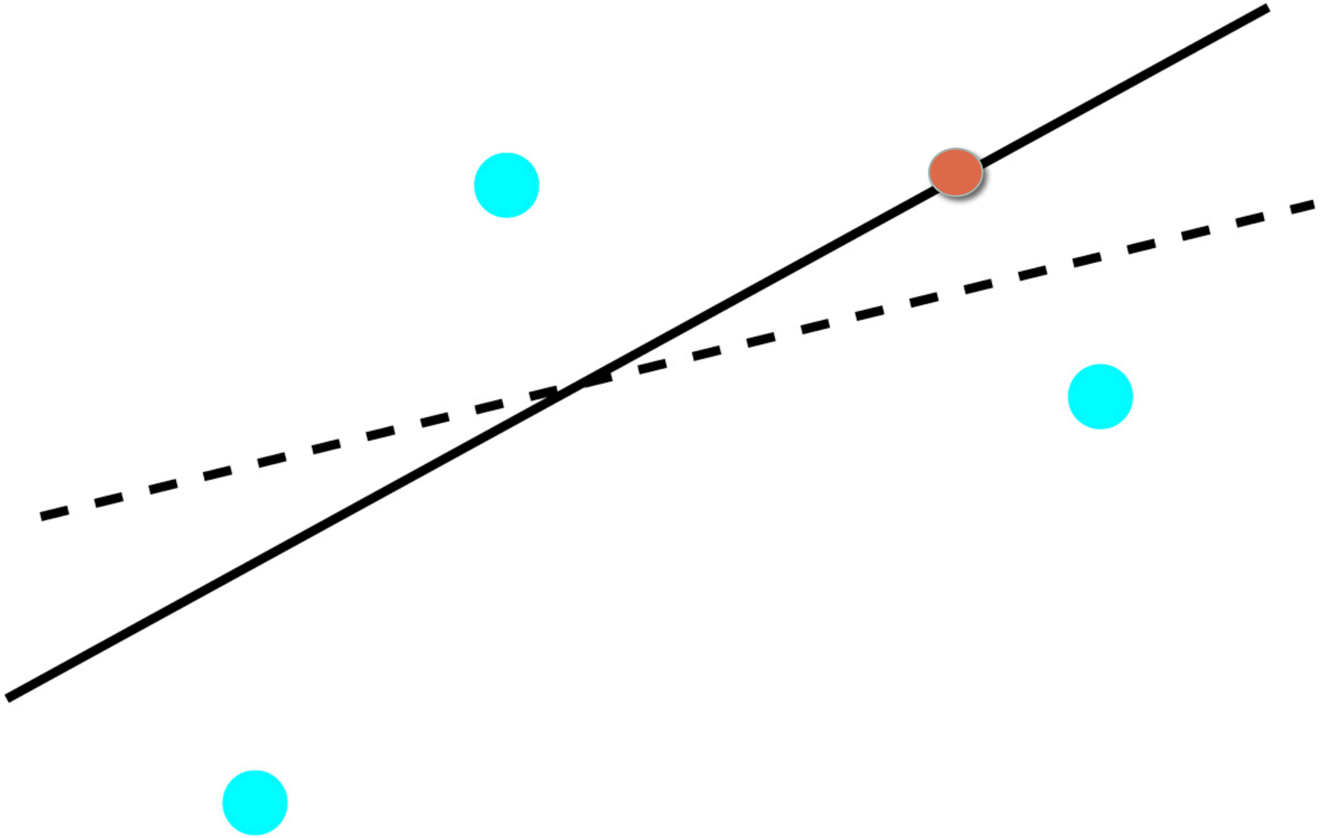
Picking θ

- Random until you get the best performance?

- How to quantify best performance?

$$h_\theta(x) = \Sigma_{i=0}^{n} \theta_i x_i = \theta^T \mathbf{x}$$

$$h_\theta(x) = \Sigma_{i=0}^{n} \theta_i x_i = \theta^T \mathbf{x}$$



https://chem.libretexts.org/Textbook_Maps/Analytical_Chemistry_Textbook_Maps/Map%3A_Analytical_Chemistry_2.0_(Harvey)/05_Standardizing_Analytical_Methods/5.4%3A_Linear_Regression_and_Calibration_Curves

Cost function (Loss function)

- Let's use the mean square error (MSE)

$$J(\theta) = \frac{1}{m} \Sigma_{i=1}^{m} (y_i - \theta^T \mathbf{x_i})^2$$

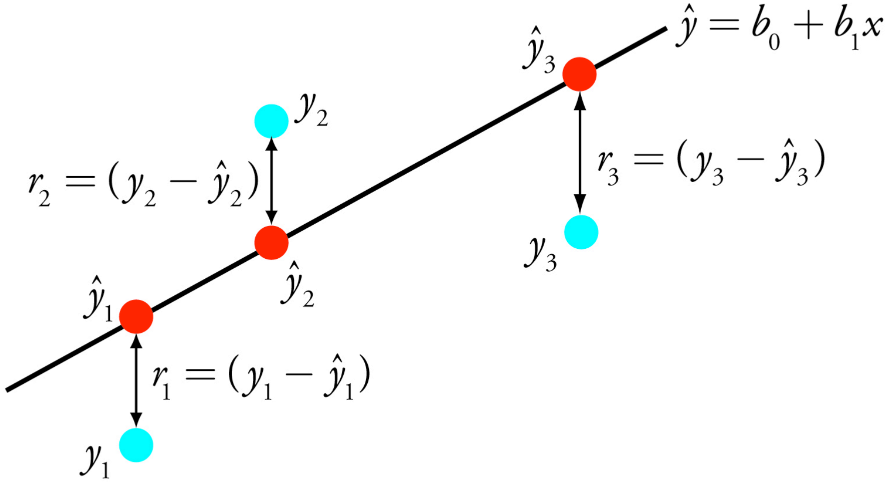
We want to pick **θ** that minimize the loss

Cost function (Loss function)

- Let's use the mean square error (MSE)

$$J(\theta) = \frac{1}{m}\Sigma_{i=1}^{m}(y_i - \theta^T \mathbf{x_i})^2$$

We want to pick **θ** that minimize the loss



Cost function (Loss function)

- Let's use the mean square error (MSE)

$$J(\theta) = \frac{1}{m}\Sigma_{i=1}^{m}(y_i - \theta^T \mathbf{x_i})^2$$

We want to pick **θ** that minimize the loss

$$\frac{m}{2}J(\theta) = \boxed{\frac{1}{2}\Sigma_{i=1}^{m}(y_i - \theta^T \mathbf{x_i})^2}$$

Picking θ

- Random until you get the best performance?
  - Can we do better than random chance?

- How to quantify best performance?

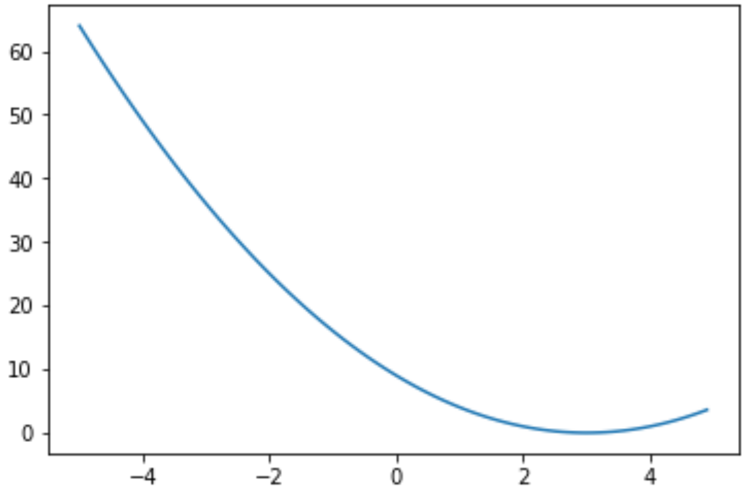
$$\frac{m}{2}J(\theta) = \frac{1}{2}\Sigma_{i=1}^{m}(y_i - \theta^T\mathbf{x_i})^2$$

Minimizing a function

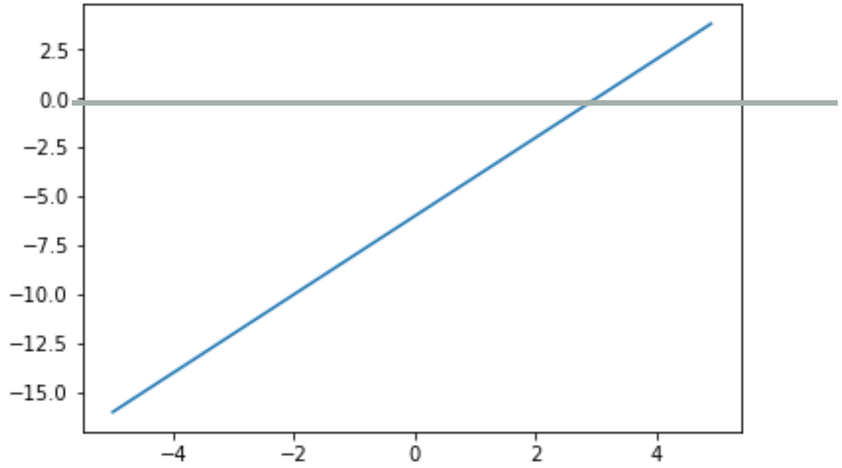
- You have a function
  - $y = (x - a)^2$
- You want to minimize Y with respect to x
  - $dy/dx = 2x - 2a$
  - Take the derivative and set the derivative to 0
    - (And maybe check if it's a minima, maxima or saddle point)

- We can also go with an iterative approach

Gradient descent



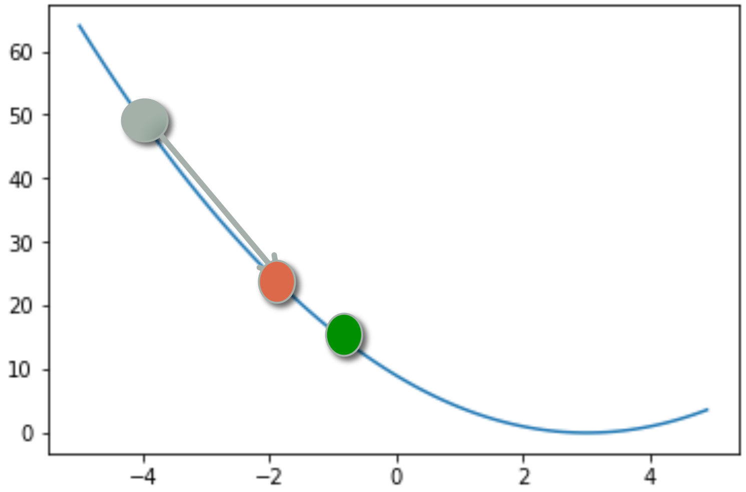
y



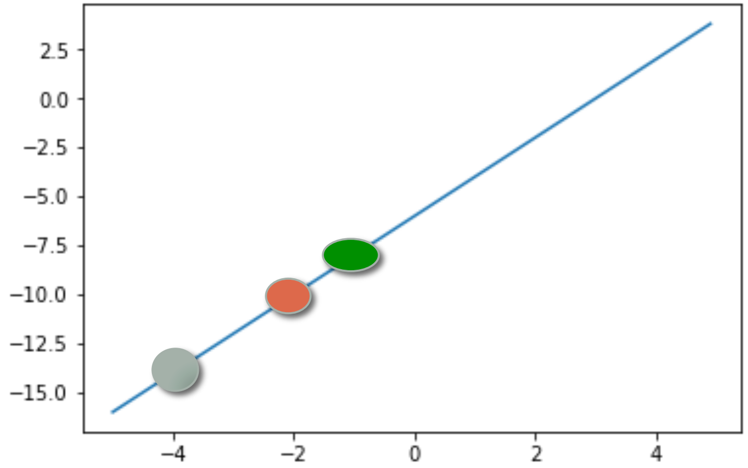
dy/dx

First what does dy/dx means?

Gradient descent



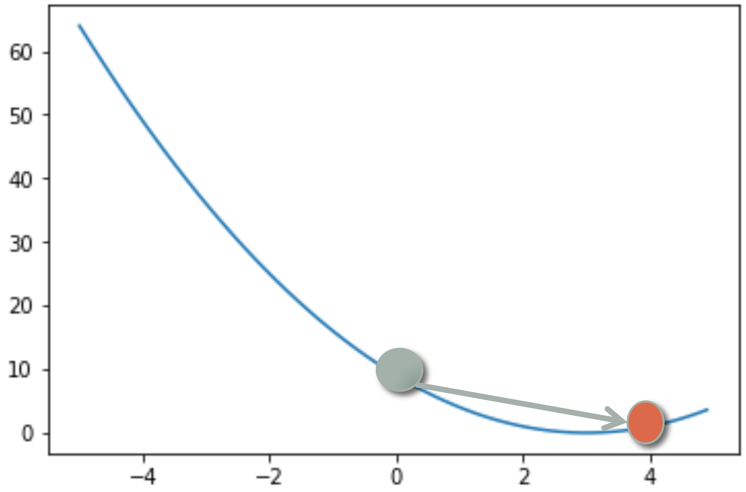
y



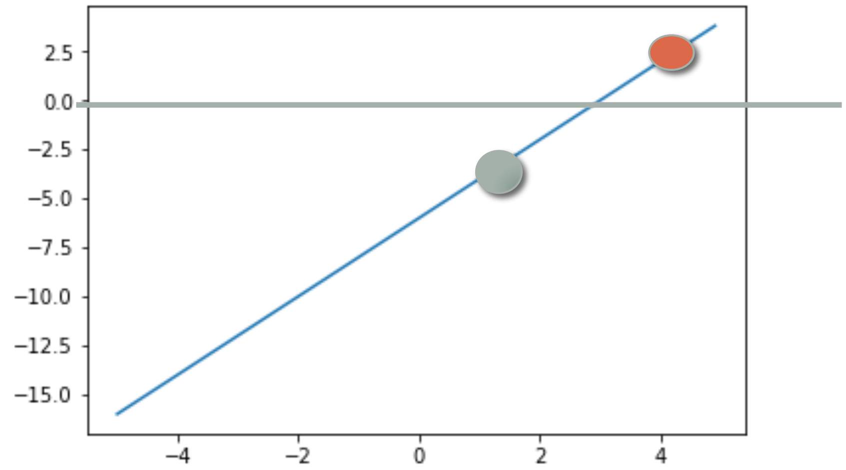
dy/dx

Move along the negative direction of the gradient
The bigger the gradient the bigger step you move

Gradient descent



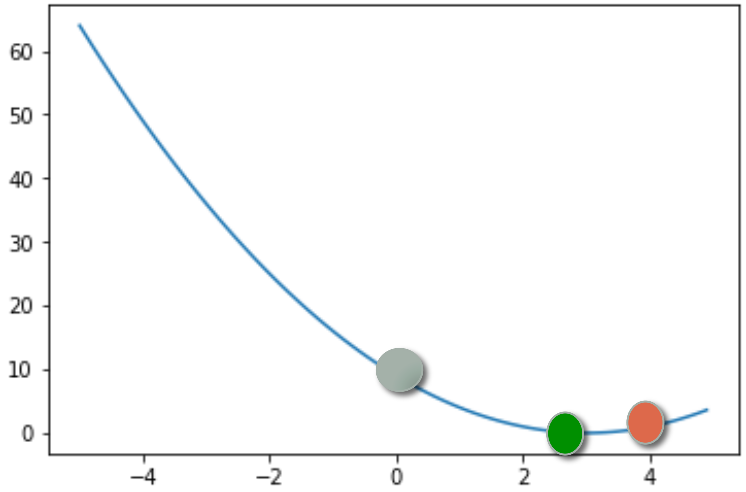
y



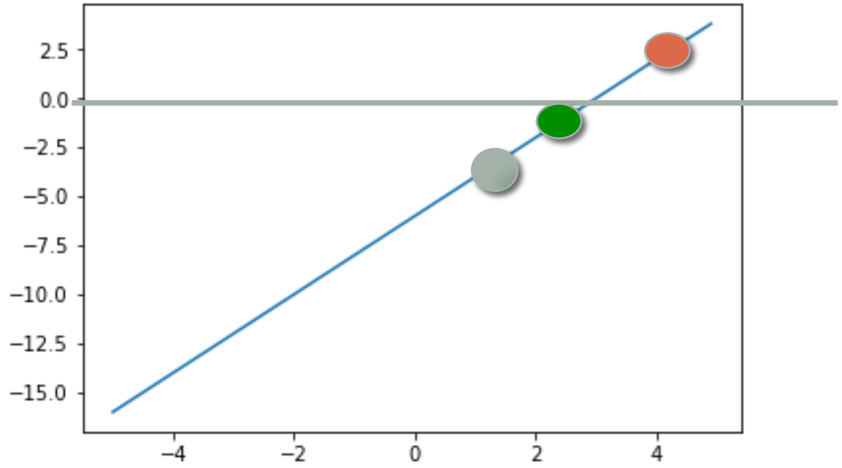
dy/dx

What happens when you overstep?

Gradient descent



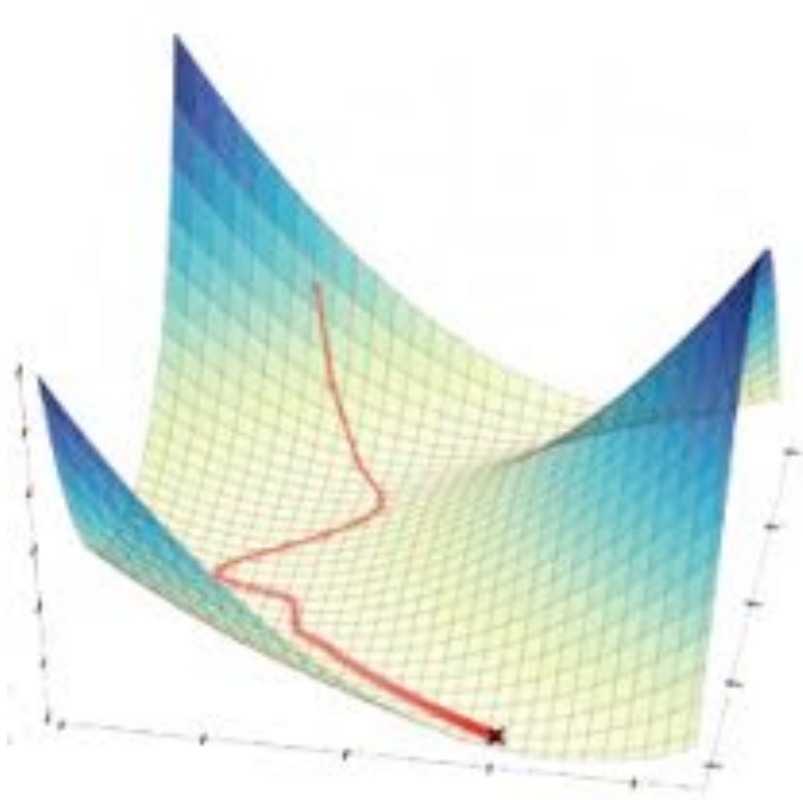
y



dy/dx

If you over step you can move back

Gradient descent in 3d



http://dsdeepdive.blogspot.com/2016/03/optimizations-of-gradient-descent.html

Formal definition

- $y = f(x)$

- Pick a starting point x_0

- Moves along $-dy/dx$
- $x_{n+1} = x_n - r * dy/dx$
- Repeat till convergence
- r is the learning rate

  Big r means you might overstep

  Small r and you need to take more steps

Picking θ

- Random until you get the best performance?
  - Can we do better than random chance?
    - Gradient descent (a better guess!)

- How to quantify best performance?

$$\frac{m}{2}J(\theta) = \frac{1}{2}\Sigma_{i=1}^{m}(y_i - \theta^T \mathbf{x_i})^2$$

LMS regression with gradient descent

$$J(\theta) = \frac{1}{2}\Sigma_{i=1}^{m}(y_i - \theta^T \mathbf{x_i})^2$$

$$\frac{\partial J}{\partial \theta_j} = -\Sigma_{i=1}^{m}(y_i - \theta^T \mathbf{x}_i)x_i^{(j)}$$

LMS regression with gradient descent

$$\frac{\partial J}{\partial \theta_j} = -\Sigma_{i=1}^{m}(y_i - \theta^T \mathbf{x}_i)x_i^{(j)}$$

$$\theta_j \Leftarrow \theta_j + r\Sigma_{i=1}^{m}(y_i - \theta^T \mathbf{x}_i)x_i^{(j)}$$

Interpretation?

Batch updates vs mini-batch

$$\theta_j \Leftarrow \theta_j - r\Sigma_{i=1}^{m}(y_i - \theta^T \mathbf{x}_i)x_i^{(j)}$$

- Batch updates (considering the whole training data) estimate the Loss function precisely
  - Can takes a long time if m is large
- Updates with a subset of m
  - We now have an estimate of the loss function
    - This can lead to a wrong direction, but we get faster updates
  - Called Stochastic Gradient Descent (SGD) or incremental gradient descent

Other loss functions

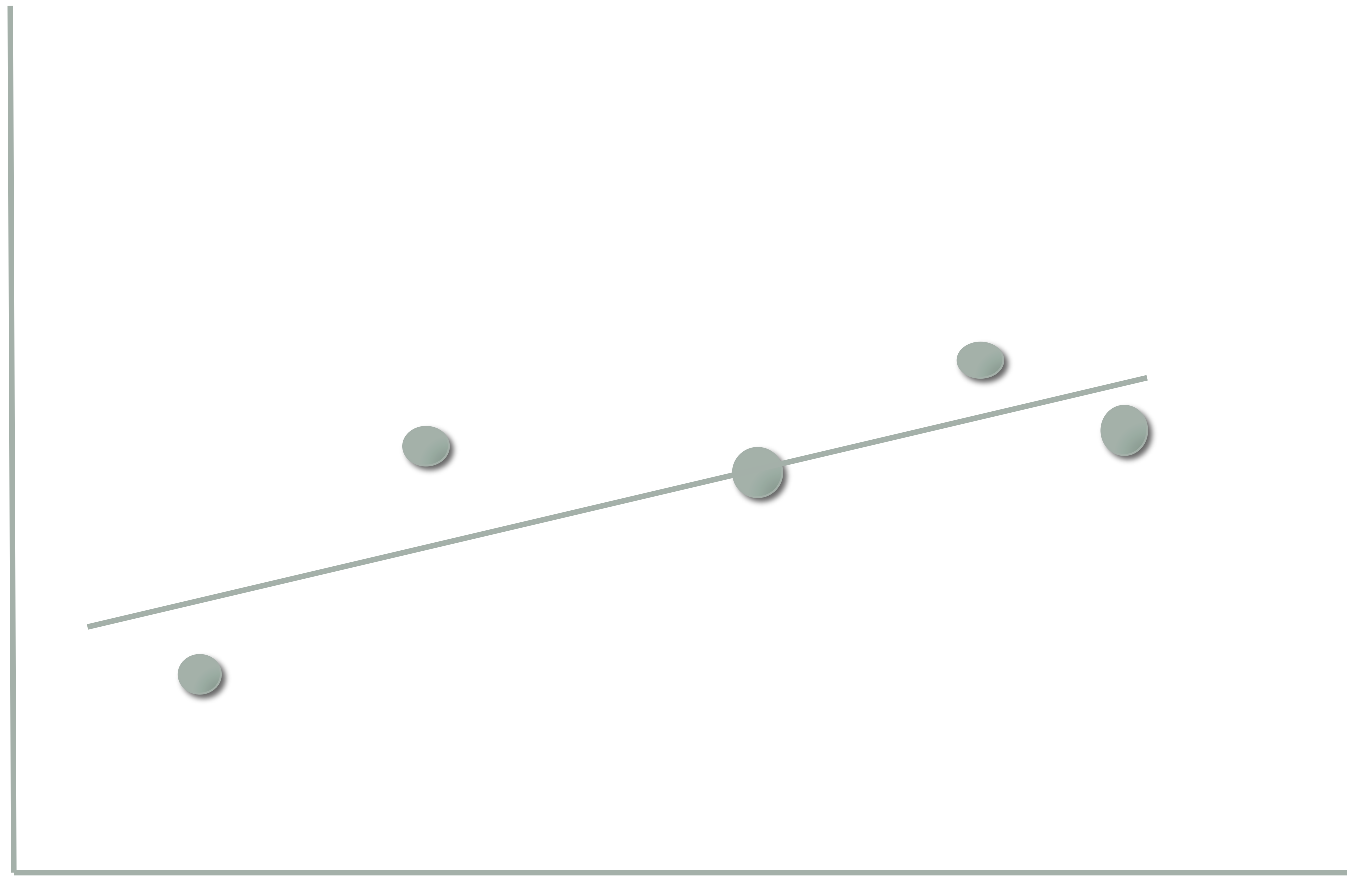
- MSE

$$J(\theta) = \frac{1}{m}\Sigma_{i=1}^{m}(y_i - \theta^T \mathbf{x_i})^2$$

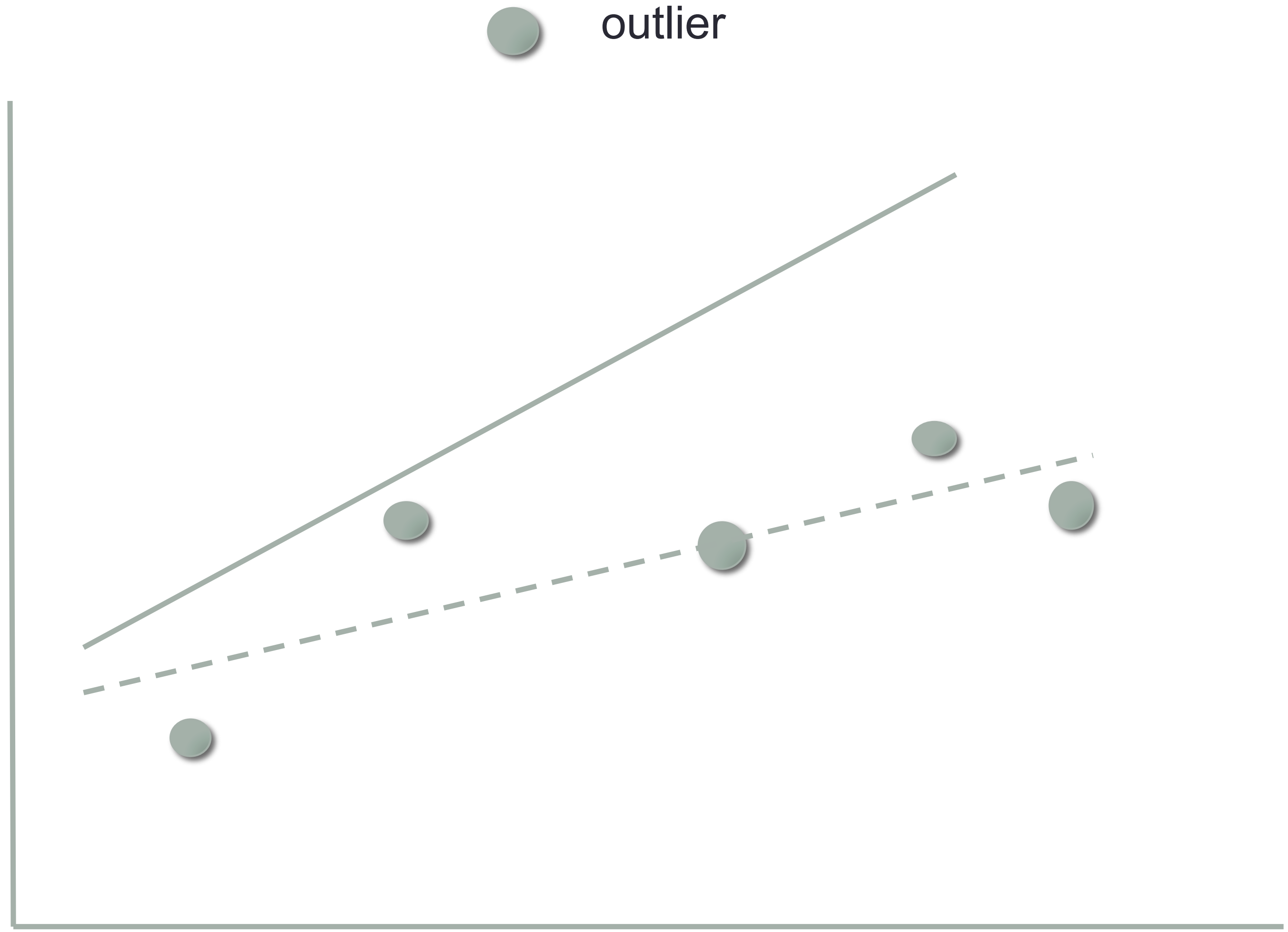
  - Also called L2 loss
- L1 loss

$$\frac{1}{m}\Sigma_{i=1}^{m}|y_i - \theta^T \mathbf{x_i}|$$

L2 vs L1 loss



L2 vs L1 loss



Outlier frequently happens in the real world

Norms (p-norm or Lp-norm)

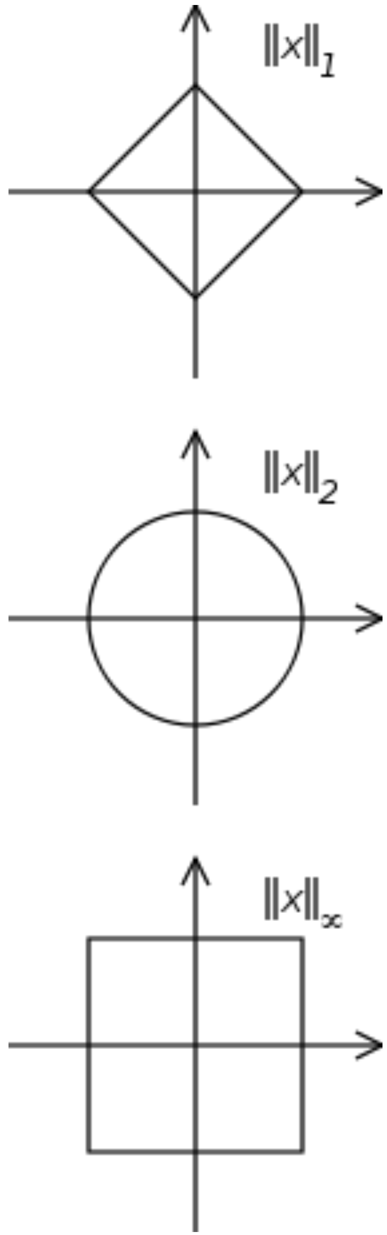
- For any real number p > 1

$$||\mathbf{x}||_p = (|x_1|^p + |x_2|^p + \ldots + |x_n|^p)^{\frac{1}{p}}$$

- For p = ∞

$$\|x\|_\infty = \max\{|x_1|, |x_2|, \ldots, |x_n|\}$$

- We'll see more of p-norms when we get to neural networks



https://en.wikipedia.org/wiki/Lp_space

Minimizing a function

- You have a function
  - $y = (x - a)^2$
- You want to minimize Y with respect to x
  - $dy/dx = 2x - 2a$
  - Take the derivative and set the derivative to 0
    - (And maybe check if it's a minima, maxima or saddle point)
- We can also go with an iterative approach (Gradient descent)

LMS regression with matrix derivatives

- First let's definite what's a derivative of a matrix
- For a function $f : \mathbb{R}^{m\times n} \mapsto \mathbb{R}$
- The derivative wrt to A is

$$\nabla_A f(A) = \begin{bmatrix} \frac{\partial f}{\partial A_{11}} & \cdots & \frac{\partial f}{\partial A_{1n}} \\ \vdots & \ddots & \vdots \\ \frac{\partial f}{\partial A_{m1}} & \cdots & \frac{\partial f}{\partial A_{mn}} \end{bmatrix}$$

http://cs229.stanford.edu/notes/cs229-notes1.pdf

Example

$$f : \mathbb{R}^{m\times n} \mapsto \mathbb{R}$$

- Suppose

$$A = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \qquad \nabla_A f(A) = \begin{bmatrix} \frac{\partial f}{\partial A_{11}} & \cdots & \frac{\partial f}{\partial A_{1n}} \\ \vdots & \ddots & \vdots \\ \frac{\partial f}{\partial A_{m1}} & \cdots & \frac{\partial f}{\partial A_{mn}} \end{bmatrix}$$

$$f(A) = \frac{3}{2}A_{11} + 5A_{12}^2 + A_{21}A_{22}$$

$$\nabla_A f(A) = \begin{bmatrix} \frac{3}{2} & 10A_{12} \\ A_{22} & A_{21} \end{bmatrix}$$

Trace of a matrix

- trA is the sum of the diagonals of matrix A (A must be a square matrix)

$$trA = \Sigma_i^N A_{ii}$$

- Trace of a real number? (1x1 matrix)

Trace properties

- tr (a) = a
- trA = trA^T
- tr(A+B) = trA+trB
- tr(aA) = atr(A)

$$\nabla_A trAB = B^T$$
$$\nabla_{A^T} f(A) = (\nabla_A f(A))^T$$
$$\nabla_A trABA^TC = CAB + C^TAB^T$$
$$\nabla_{A^T} trABA^TC = B^TA^TC^T + BA^TC$$

LMS regression with matrix derivatives

$$X = \begin{bmatrix} - & x_1^T & - \\ - & x_2^T & - \\ - & x_m^T & - \end{bmatrix} \qquad y = \begin{bmatrix} y_1 \\ y_2 \\ y_m \end{bmatrix}$$

$$X\theta - y = \begin{bmatrix} x_1^T\theta \\ | \\ x_m^T\theta \end{bmatrix} - \begin{bmatrix} y_1 \\ | \\ y_m \end{bmatrix}$$

LMS regression with matrix derivatives

$$X\theta - y = \begin{bmatrix} x_1^T\theta \\ | \\ x_m^T\theta \end{bmatrix} - \begin{bmatrix} y_1 \\ | \\ y_m \end{bmatrix}$$

$$\frac{1}{2}(X\theta - y)^T(X\theta - y) = \frac{1}{2}\sum_{i=1}^{m}(y_i - \theta^T x)^2$$

We want to minimize this term wrt to **θ**

LMS regression with matrix derivatives

$$\theta = (X^T X)^{-1} X^T y$$

Trace properties

1 • tr (a) = a

2 • $trA = trA^T$

3 • tr(A+B) = trA+trB

4 • tr(aA) = atr(A)

5 $$\nabla_A trAB = B^T$$

6 $$\nabla_{A^T} f(A) = (\nabla_A f(A))^T$$

7 $$\nabla_A trABA^T C = CAB + C^T AB^T$$

8 $$\nabla_{A^T} trABA^T C = B^T A^T C^T + BA^T C$$

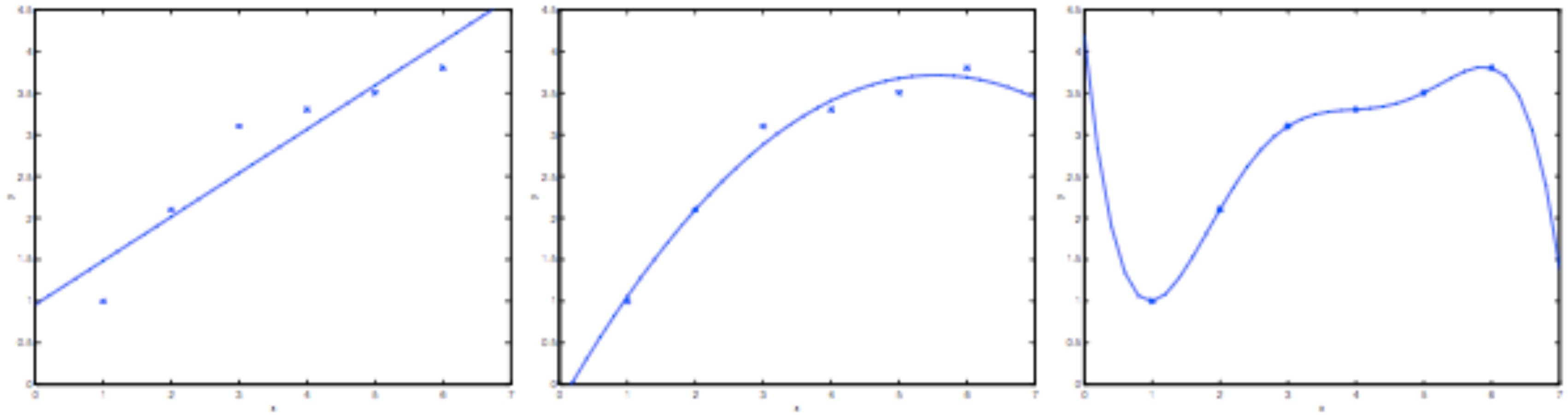
Regression with non-linear features

- If we add extra features that are non-linear
  - For example x^2

Cloth	Corn	Grass	Water	Beer	Rainfall
4	6	3	10	0	**76950**
5	1	0	0	7	**30234**
6	0	3	5	7	**123456**
5	0	3	12	0	**89301**
4	3	0	6	7	**?**

- $h_{\theta}(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_3 + \theta_4 x_4 + \theta_5 x_5 + \theta_6 x_1^2 + \ldots$
- These can be considered as additional features
- We can now have a line that is non-linear

Overfitting Underfitting



Adding more non-linear features makes the line more curvy

(Adding more features also means more model parameters)

The curve can go directly to the outliers with enough parameters.

We call this effect overfitting

For the opposite case, having not enough parameters to model the data is called underfitting

Predicting floods

Cloth	Corn	Grass	Water	Beer	Flood?
4	6	3	10	0	**yes**
5	1	0	0	7	**yes**
6	0	3	5	7	**no**
5	0	3	12	0	**yes**
4	3	0	6	7	**?**

So far we talk about predicting an amount what if we want to do classification

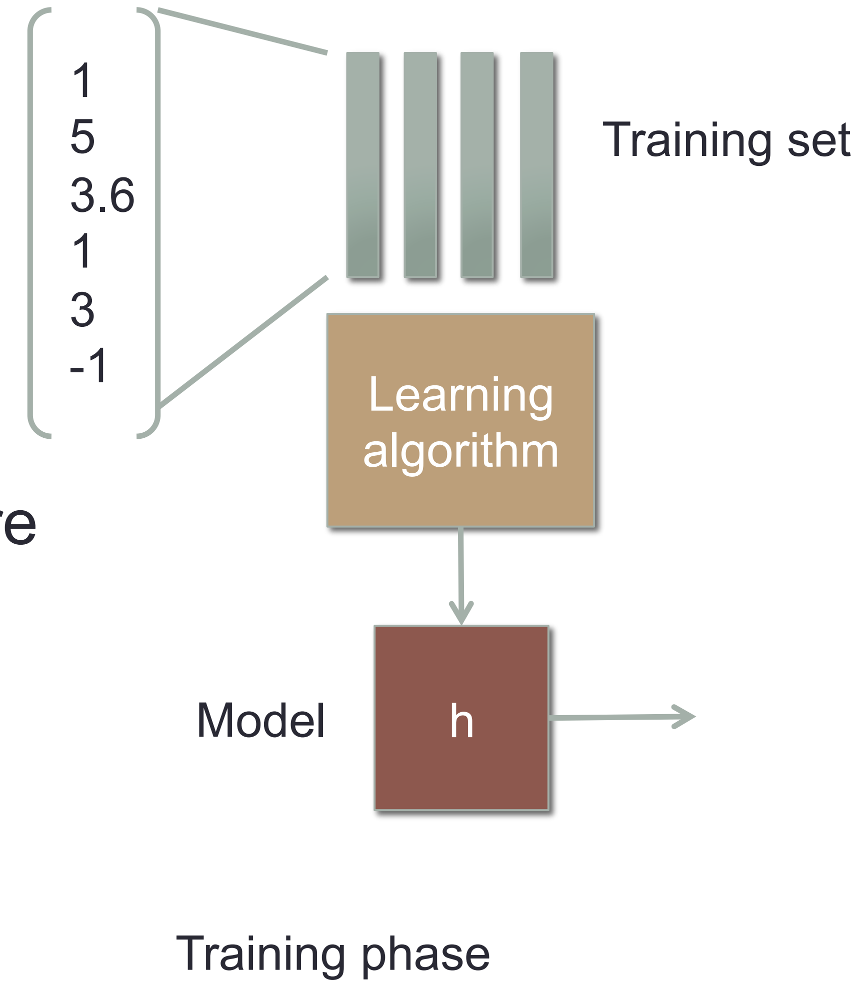
Let's start with a binary choice. Flood or no flood

Flood or no flood

- What would be the output?

- y = 0 if not flooded
- y = 1 if flooded

- Anything in between is a score for how likely it is to flood



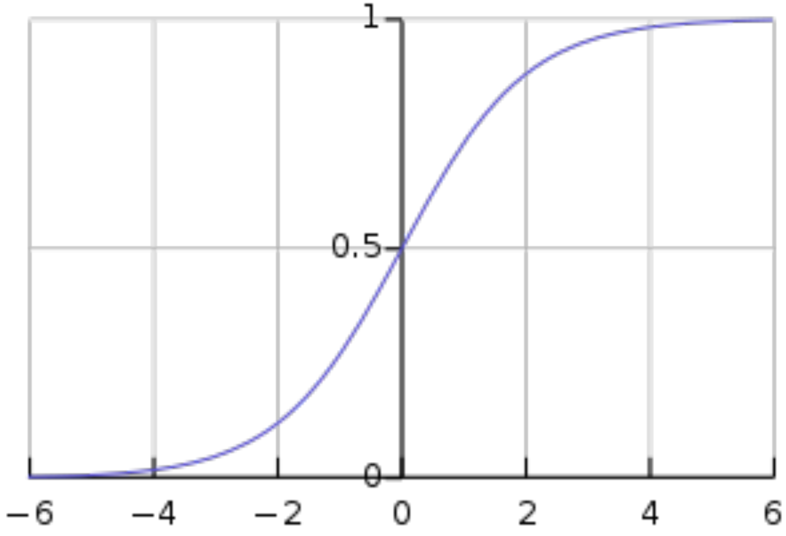
Can we use regression?

- Yes
- $h_{\theta}(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_3 + \theta_4 x_4 + \theta_5 x_5$

- But
- What does it mean when h is higher than 1?
- Can h be negative? What does it mean to have a negative flood value?

Logistic function

- Let’s force h to be between 0 and 1 somehow
- Introducing the logistic function (sigmoid function)



$$f(x) = \frac{1}{1+e^{-x}} = \frac{e^x}{1+e^x}$$

https://en.wikipedia.org/wiki/Logistic_function

Logistic Regression

- Pass $\theta^T \mathbf{x}$ through the logistic function

$$h_\theta(x) = \frac{1}{1 + e^{-\theta^T x}}$$

Loss function?

- MSE error no longer a good candidate

Logistic Regression update rule

$$\theta_j \Leftarrow \theta_j + r\Sigma_{i=1}^{m}(y_i - h_\theta(x_i))x_i^{(j)}$$

Update rule for linear regression

$$\theta_j \Leftarrow \theta_j + r\Sigma_{i=1}^{m}(y_i - \theta^T \mathbf{x}_i)x_i^{(j)}$$

Office hours

- Thursdays 16.30-18.00 at 19th floor space
- Don’t forget that Piazza also exists!

Demo - Jupyter

- http://jupyter.readthedocs.io/en/latest/install.html#
- https://www.anaconda.com/download/