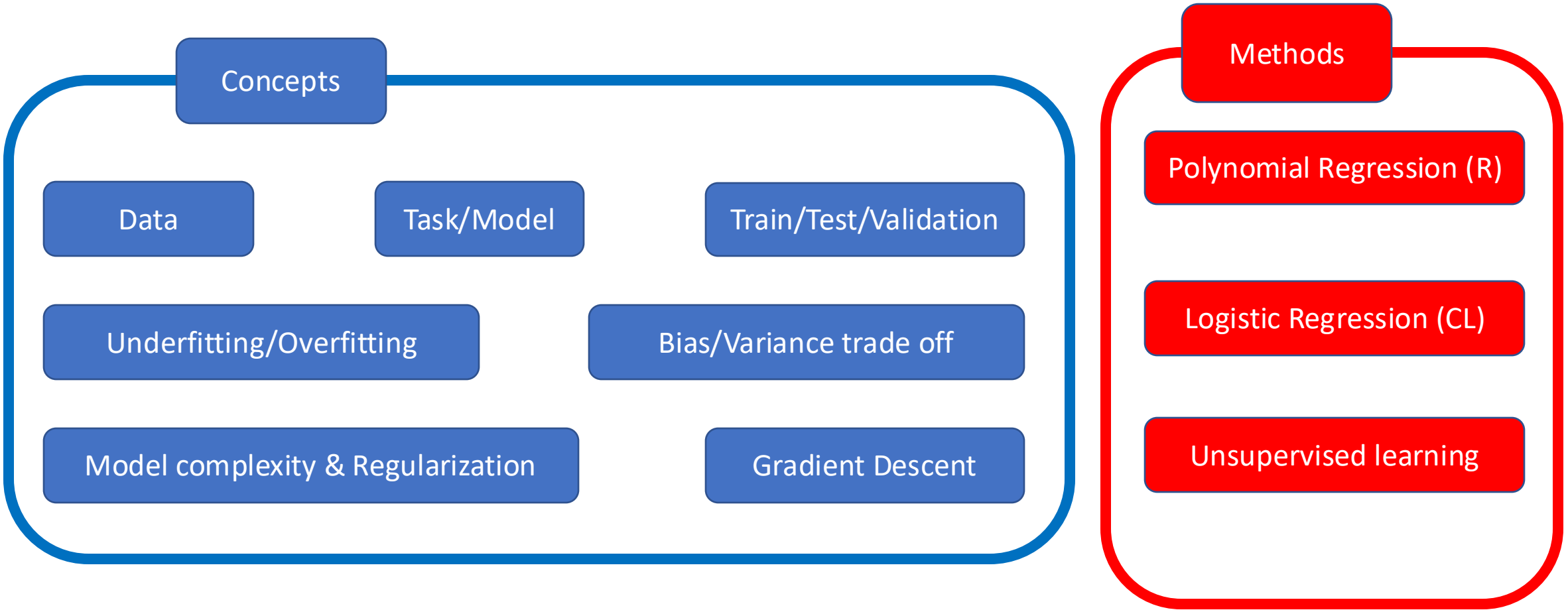
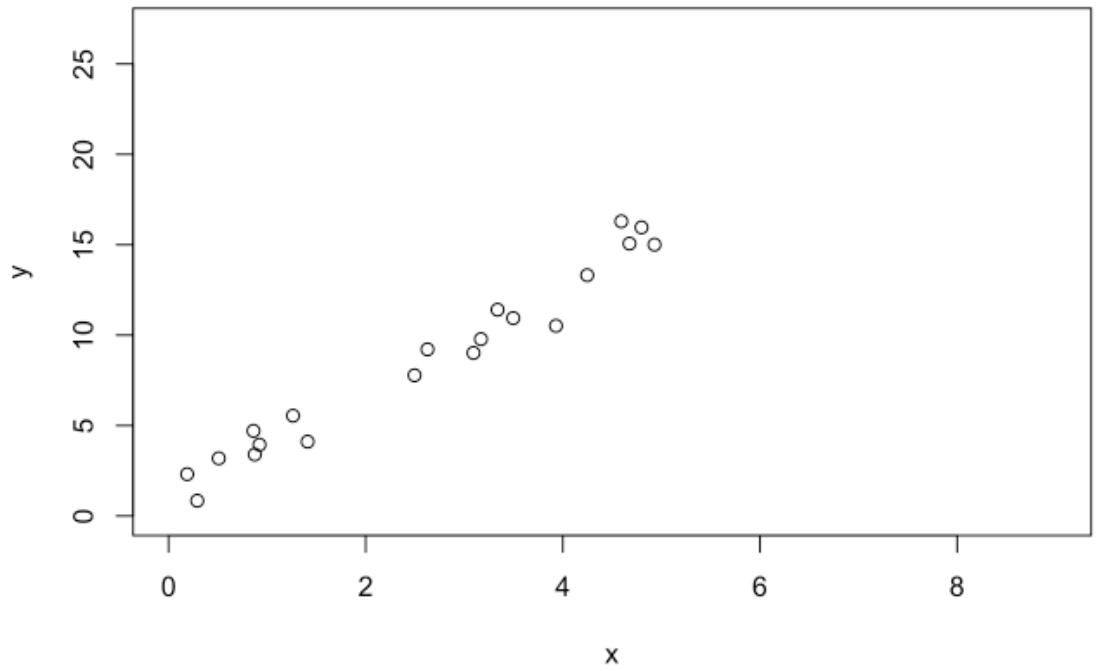
Summarizing…



- Linear regression: closed form solution
- Logistic regression: gradient descent (derivation).
- Other regressions (based on noise type). Bias/Variance: derivation
- Regularization (ridge e lasso, elastic nets)

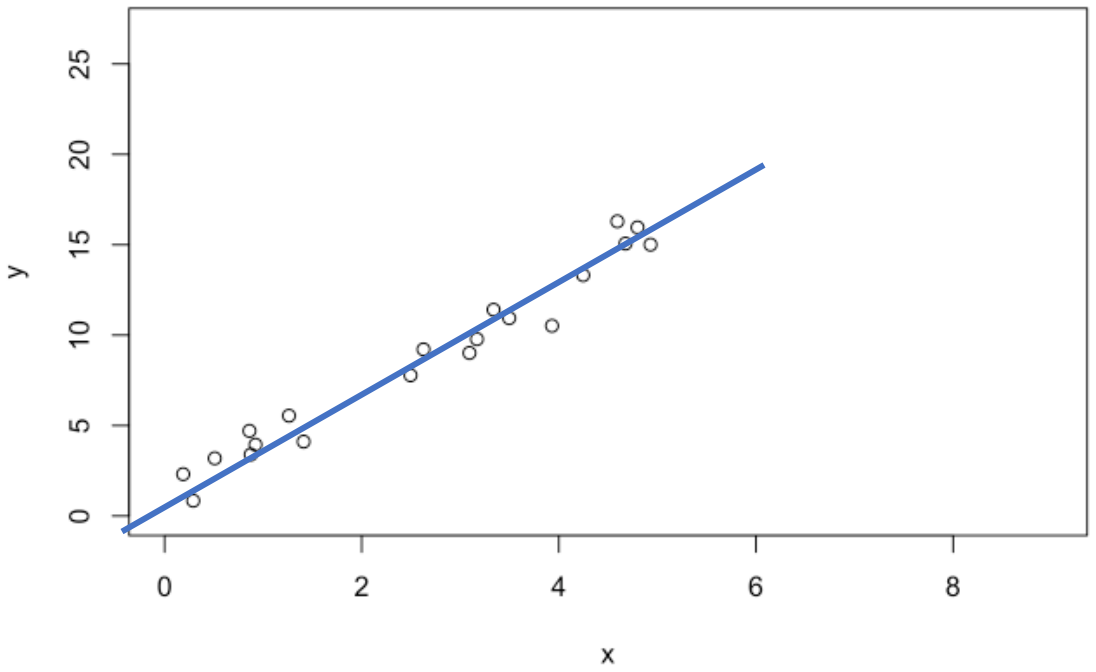
Linear regression: closed form solution

Linear regression: recap



$$x \in \mathbb{R}^d, y \in \mathbb{R}$$

Linear regression: recap



$$y = w_0 + w_1 x_1 + w_2 x_2 + \cdots$$

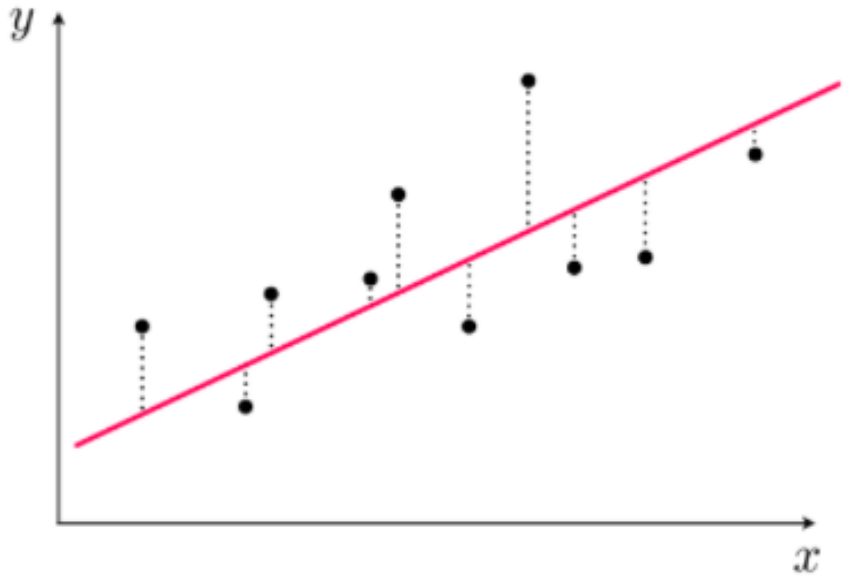
Model hypothesis

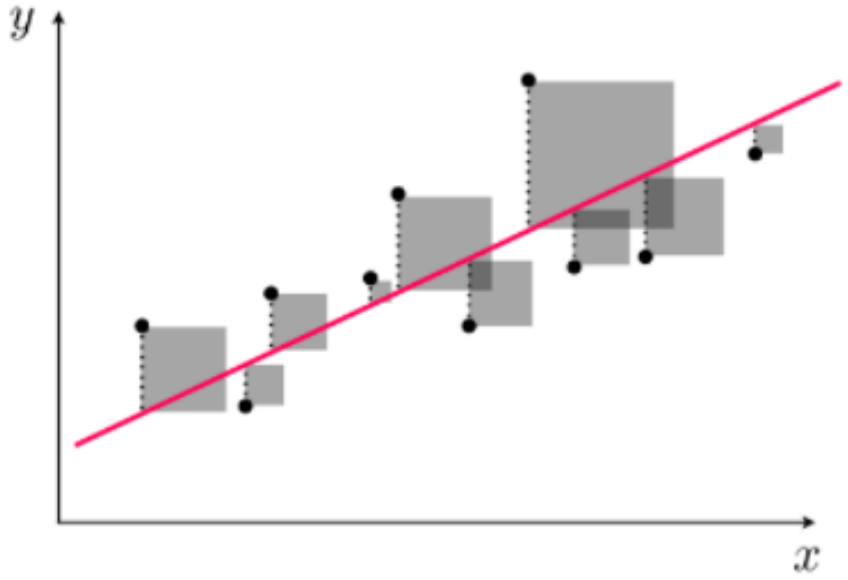
$$y = wx + b$$

$$y = \sum_{i=1}^{d} w_i x_i \ + b$$

Even if our model is perfect there is noise and the fit is not perfect: how do we measure the error we make?

Measuring errors: square cost





$$\mathcal{E}(w, x, y) = \frac{1}{N} \sum_{i=1}^{N} (y_i - \sum_j w_j x_j^i)^2$$

How do we find a solution?

How do I write the RHS in a compact form?

More compactly

$$w, x \in \mathbb{R}^d, \qquad X \in \mathbb{R}^{d \times N}, \quad y \in \mathbb{R}^N$$

$$\mathcal{E}(w, X, y) = \frac{1}{N} \|y - X^T w\|_2^2$$

We want to minimize the error we make choosing the best w

Optimization problem: least mean squares

$$w^* = \arg\min_w \frac{1}{N} \|y - X^T w\|_2^2$$

How do we minimize?

To minimize the loss we take the gradient: how do we calculate?

$$\nabla_w \|y - X^T w\|_2^2$$

Matrix Calculus

Let $y, x \in \mathbb{R}$ be scalars, $\mathbf{y} \in \mathbb{R}^M$ and $\mathbf{x} \in \mathbb{R}^P$ be vectors, and $\mathbf{Y} \in \mathbb{R}^{M \times N}$ and $\mathbf{X} \in \mathbb{R}^{P \times Q}$ be matrices

Denominator \ *Numerator*	scalar	vector	matrix
Types of Derivatives			
scalar	$\frac{\partial y}{\partial x}$	$\frac{\partial \mathbf{y}}{\partial x}$	$\frac{\partial \mathbf{Y}}{\partial x}$
vector	$\frac{\partial y}{\partial \mathbf{x}}$	$\frac{\partial \mathbf{y}}{\partial \mathbf{x}}$	$\frac{\partial \mathbf{Y}}{\partial \mathbf{x}}$
matrix	$\frac{\partial y}{\partial \mathbf{X}}$	$\frac{\partial \mathbf{y}}{\partial \mathbf{X}}$	$\frac{\partial \mathbf{Y}}{\partial \mathbf{X}}$

Matrix Calculus

Let $y, x \in \mathbb{R}$ be scalars, $\mathbf{y} \in \mathbb{R}^M$ and $\mathbf{x} \in \mathbb{R}^P$ be vectors, and $\mathbf{Y} \in \mathbb{R}^{M \times N}$ and $\mathbf{X} \in \mathbb{R}^{P \times Q}$ be matrices

Denominator \ *Numerator*	scalar	vector	matrix
Types of Derivatives			
scalar	$\frac{\partial y}{\partial x}$	$\frac{\partial \mathbf{y}}{\partial x}$	$\frac{\partial \mathbf{Y}}{\partial x}$
vector	$\frac{\partial y}{\partial \mathbf{x}}$	$\frac{\partial \mathbf{y}}{\partial \mathbf{x}}$	$\frac{\partial \mathbf{Y}}{\partial \mathbf{x}}$
matrix	$\frac{\partial y}{\partial \mathbf{X}}$	$\frac{\partial \mathbf{y}}{\partial \mathbf{X}}$	$\frac{\partial \mathbf{Y}}{\partial \mathbf{X}}$

Types of Derivatives	**scalar**
scalar	$\frac{\partial y}{\partial x} = \left[\frac{\partial y}{\partial x}\right]$
vector	$\frac{\partial y}{\partial \mathbf{x}} = \begin{bmatrix} \frac{\partial y}{\partial x_1} \\ \frac{\partial y}{\partial x_2} \\ \vdots \\ \frac{\partial y}{\partial x_P} \end{bmatrix}$
matrix	$\frac{\partial y}{\partial \mathbf{X}} = \begin{bmatrix} \frac{\partial y}{\partial X_{11}} & \frac{\partial y}{\partial X_{12}} & \cdots & \frac{\partial y}{\partial X_{1Q}} \\ \frac{\partial y}{\partial X_{21}} & \frac{\partial y}{\partial X_{22}} & \cdots & \frac{\partial y}{\partial X_{2Q}} \\ \vdots & & & \vdots \\ \frac{\partial y}{\partial X_{P1}} & \frac{\partial y}{\partial X_{P2}} & \cdots & \frac{\partial y}{\partial X_{PQ}} \end{bmatrix}$

$$\frac{\partial \mathbf{x}^T \mathbf{a}}{\partial \mathbf{x}} = \frac{\partial \mathbf{a}^T \mathbf{x}}{\partial \mathbf{x}} = \mathbf{a}$$

$$\frac{\partial \mathbf{a}^T \mathbf{X} \mathbf{b}}{\partial \mathbf{X}} = \mathbf{a}\mathbf{b}^T$$

$$\frac{\partial \mathbf{x}' \mathbf{A} \mathbf{x}}{\partial \mathbf{x}} = (\mathbf{A} + \mathbf{A}')\,\mathbf{x}$$

The Matrix Cookbook

[http://matrixcookbook.com]

Kaare Brandt Petersen
Michael Syskind Pedersen

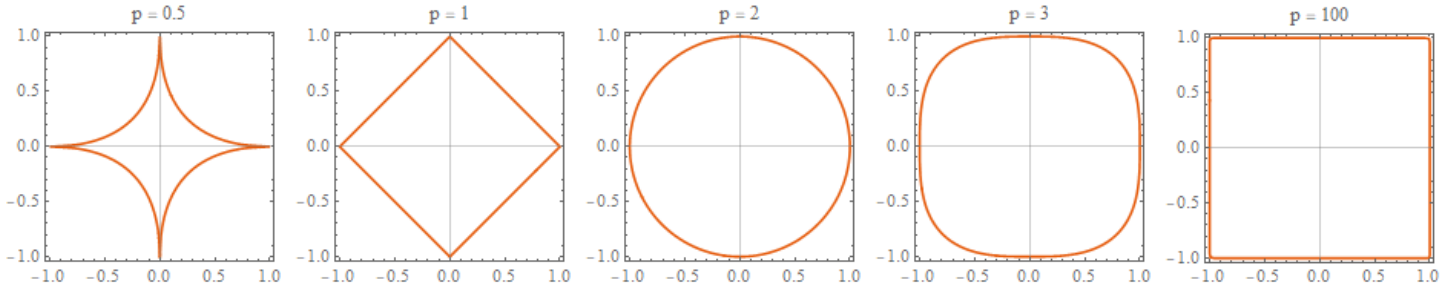
First Order

$$\begin{aligned}
\frac{\partial}{\partial \mathbf{X}}\mathrm{Tr}(\mathbf{X}) &= \mathbf{I} \\
\frac{\partial}{\partial \mathbf{X}}\mathrm{Tr}(\mathbf{XA}) &= \mathbf{A}^T \\
\frac{\partial}{\partial \mathbf{X}}\mathrm{Tr}(\mathbf{AXB}) &= \mathbf{A}^T\mathbf{B}^T \\
\frac{\partial}{\partial \mathbf{X}}\mathrm{Tr}(\mathbf{AX}^T\mathbf{B}) &= \mathbf{BA} \\
\frac{\partial}{\partial \mathbf{X}}\mathrm{Tr}(\mathbf{X}^T\mathbf{A}) &= \mathbf{A} \\
\frac{\partial}{\partial \mathbf{X}}\mathrm{Tr}(\mathbf{AX}^T) &= \mathbf{A} \\
\frac{\partial}{\partial \mathbf{X}}\mathrm{Tr}(\mathbf{A}\otimes\mathbf{X}) &= \mathrm{Tr}(\mathbf{A})\mathbf{I}
\end{aligned}$$

2.5.2 Second Order

$$\begin{aligned}
\frac{\partial}{\partial \mathbf{X}}\mathrm{Tr}(\mathbf{X}^2) &= 2\mathbf{X}^T \\
\frac{\partial}{\partial \mathbf{X}}\mathrm{Tr}(\mathbf{X}^2\mathbf{B}) &= (\mathbf{XB}+\mathbf{BX})^T \\
\frac{\partial}{\partial \mathbf{X}}\mathrm{Tr}(\mathbf{X}^T\mathbf{BX}) &= \mathbf{BX}+\mathbf{B}^T\mathbf{X} \\
\frac{\partial}{\partial \mathbf{X}}\mathrm{Tr}(\mathbf{BXX}^T) &= \mathbf{BX}+\mathbf{B}^T\mathbf{X} \\
\frac{\partial}{\partial \mathbf{X}}\mathrm{Tr}(\mathbf{XX}^T\mathbf{B}) &= \mathbf{BX}+\mathbf{B}^T\mathbf{X} \\
\frac{\partial}{\partial \mathbf{X}}\mathrm{Tr}(\mathbf{XBX}^T) &= \mathbf{XB}^T+\mathbf{XB} \\
\frac{\partial}{\partial \mathbf{X}}\mathrm{Tr}(\mathbf{BX}^T\mathbf{X}) &= \mathbf{XB}^T+\mathbf{XB} \\
\frac{\partial}{\partial \mathbf{X}}\mathrm{Tr}(\mathbf{X}^T\mathbf{XB}) &= \mathbf{XB}^T+\mathbf{XB} \\
\frac{\partial}{\partial \mathbf{X}}\mathrm{Tr}(\mathbf{AXBX}) &= \mathbf{A}^T\mathbf{X}^T\mathbf{B}^T+\mathbf{B}^T\mathbf{X}^T\mathbf{A}^T \\
\frac{\partial}{\partial \mathbf{X}}\mathrm{Tr}(\mathbf{X}^T\mathbf{X}) &= \frac{\partial}{\partial \mathbf{X}}\mathrm{Tr}(\mathbf{XX}^T) = 2\mathbf{X} \\
\frac{\partial}{\partial \mathbf{X}}\mathrm{Tr}(\mathbf{B}^T\mathbf{X}^T\mathbf{CXB}) &= \mathbf{C}^T\mathbf{XBB}^T+\mathbf{CXBB}^T \\
\frac{\partial}{\partial \mathbf{X}}\mathrm{Tr}\left[\mathbf{X}^T\mathbf{BXC}\right] &= \mathbf{BXC}+\mathbf{B}^T\mathbf{XC}^T \\
\frac{\partial}{\partial \mathbf{X}}\mathrm{Tr}(\mathbf{AXBX}^T\mathbf{C}) &= \mathbf{A}^T\mathbf{C}^T\mathbf{XB}^T+\mathbf{CAXB} \\
\frac{\partial}{\partial \mathbf{X}}\mathrm{Tr}\Big[(\mathbf{AXB}+\mathbf{C})(\mathbf{AXB}+\mathbf{C})^T\Big] &= 2\mathbf{A}^T(\mathbf{AXB}+\mathbf{C})\mathbf{B}^T \\
\frac{\partial}{\partial \mathbf{X}}\mathrm{Tr}(\mathbf{X}\otimes\mathbf{X}) &= \frac{\partial}{\partial \mathbf{X}}\mathrm{Tr}(\mathbf{X})\mathrm{Tr}(\mathbf{X}) = 2\mathrm{Tr}(\mathbf{X})\mathbf{I}
\end{aligned}$$

Lp norms and their gradient



$$x \in \mathbb{R}^d, \quad \|x\|_p = \left(\sum_{i=1}^{d} x_i^p \right)^{1/p}$$

What is its gradient?

Lp norms and their gradient

$$
\begin{aligned}
\frac{\partial}{\partial x_j}||\mathbf{x}||_p &= \frac{\partial}{\partial x_j}\left(\sum_{i=1}^{n}|x_i|^p\right)^{1/p} \\
&= \frac{1}{p}\left(\sum_{i=1}^{n}|x_i|^p\right)^{(1/p)-1}\frac{\partial}{\partial x_j}\left(\sum_{i=1}^{n}|x_i|^p\right) \\
&= \frac{1}{p}\left(\sum_{i=1}^{n}|x_i|^p\right)^{\frac{1-p}{p}}\sum_{i=1}^{n}p|x_i|^{p-1}\frac{\partial}{\partial x_j}|x_i| \\
&= \left[\left(\sum_{i=1}^{n}|x_i|^p\right)^{\frac{1}{p}}\right]^{1-p}\sum_{i=1}^{n}|x_i|^{p-1}\delta_{ij}\frac{x_i}{|x_i|} \\
&= ||\mathbf{x}||_p^{1-p}\cdot|x_j|^{p-1}\frac{x_j}{|x_j|} \\
&= \frac{x_j|x_j|^{p-2}}{||\mathbf{x}||_p^{p-1}}
\end{aligned}
$$

Closed form solution of regression

$$\nabla_w \|y - X^T w\|_2^2 = \nabla_w [(y - X^T w)^T (y - X^T w)]$$

$$= \nabla_w (y^T y - y^T X^T w - w^T X y + w^T X X^T w)$$

$$= \nabla_w (-2 w^T X y + w^T X X^T w)$$

$$= -2(Xy - X X^T w) = -2X(y - X^T w)$$

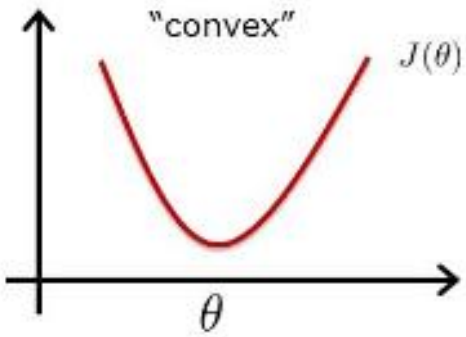
$$\frac{\partial \mathbf{x}^T \mathbf{a}}{\partial \mathbf{x}} = \frac{\partial \mathbf{a}^T \mathbf{x}}{\partial \mathbf{x}} = \mathbf{a}$$

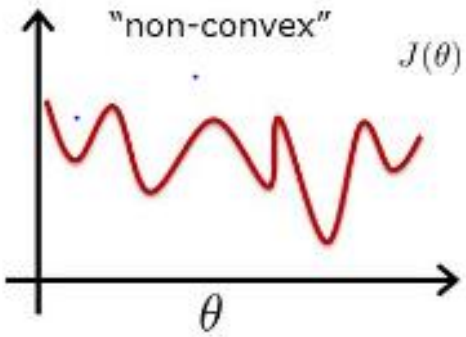
$$\frac{\partial \mathbf{x}' \mathbf{A} \mathbf{x}}{\partial \mathbf{x}} = (\mathbf{A} + \mathbf{A}')\,\mathbf{x}$$

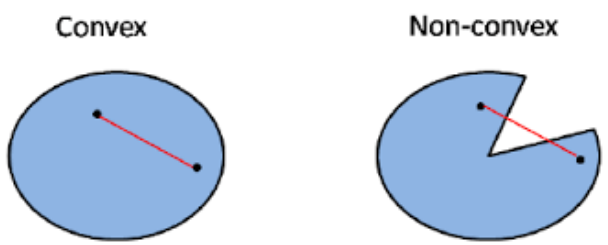
$$2X(y - X^T w) = 0 \longrightarrow \boxed{w = (X X^T)^{-1} X y}$$

Is the solution unique?

Convex vs not convex Loss

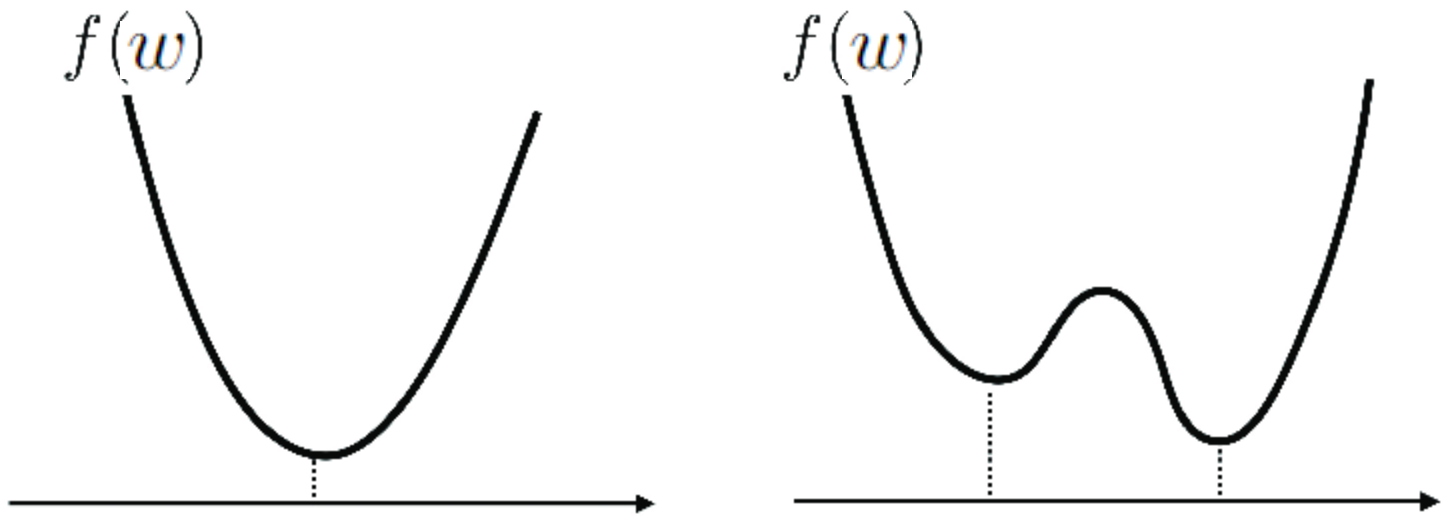




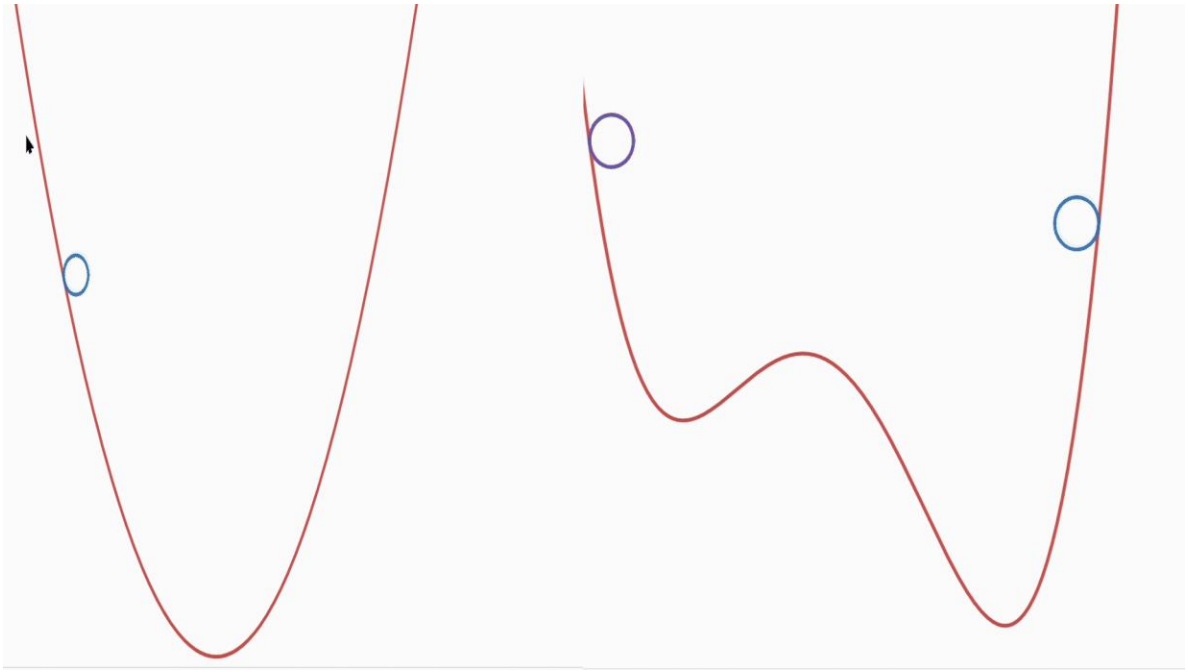


Why is it important?

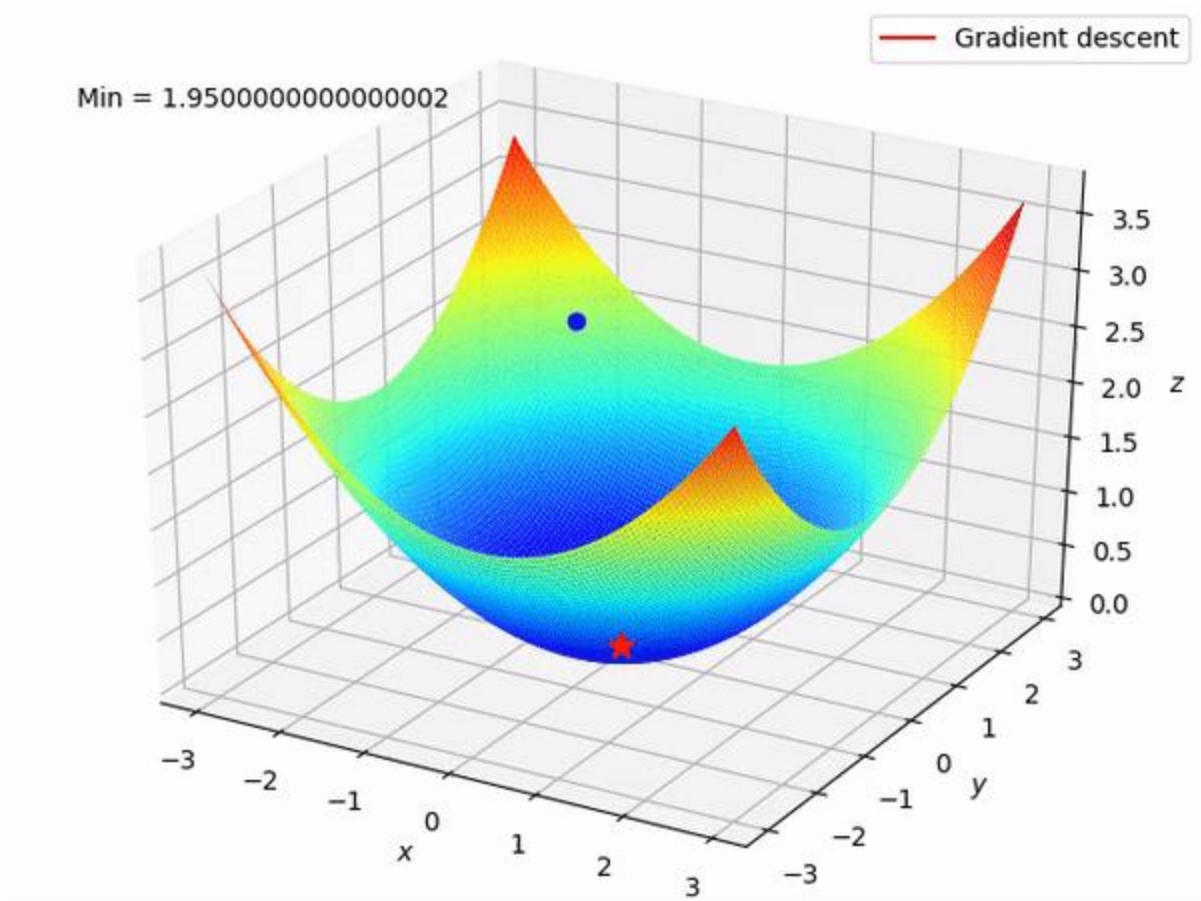
Convex: global= local minima



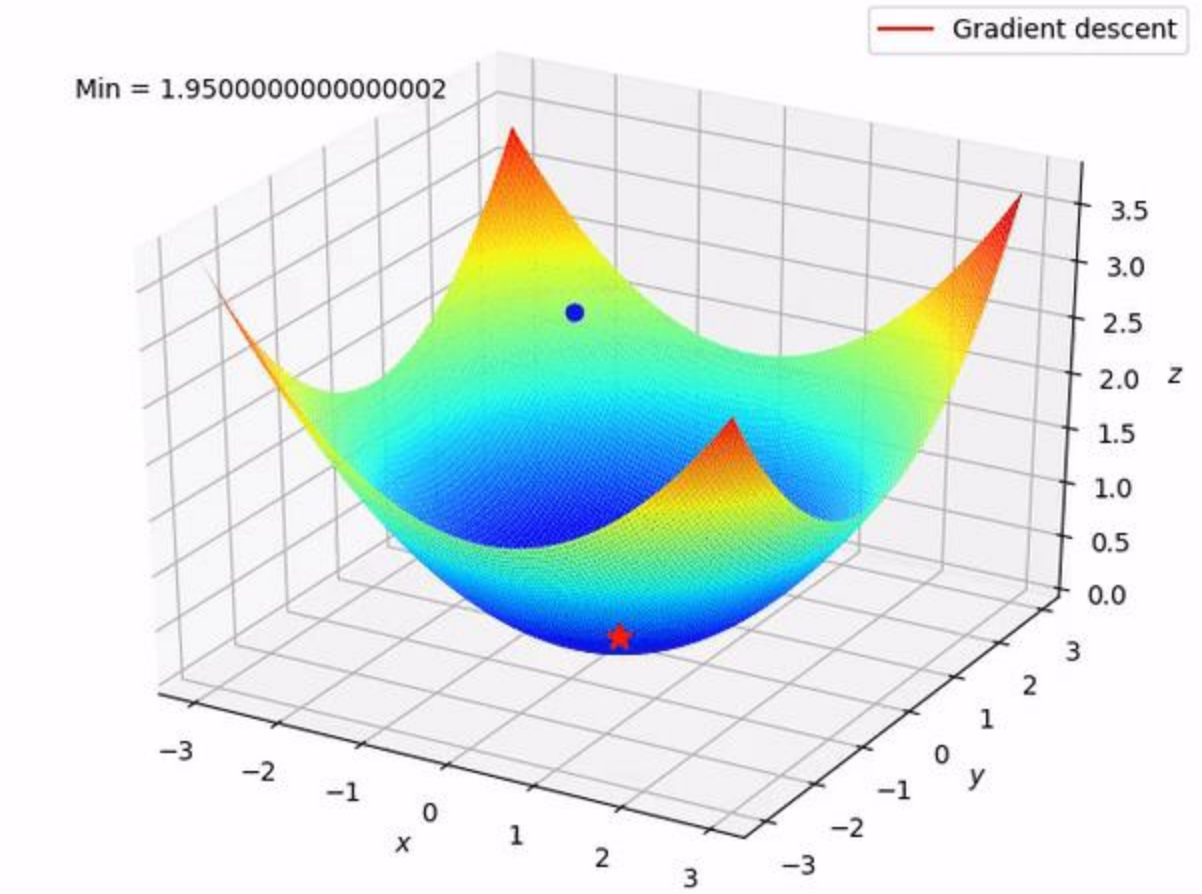
Convexity: global= local minima



Convex: global= local minima



Is our way of counting errors convex in w?
How do you test if a function is convex in one dimension?



We test if the Hessian matrix is positive definite

$$\mathbf{H}_f = \begin{bmatrix} \dfrac{\partial^2 f}{\partial x_1^2} & \dfrac{\partial^2 f}{\partial x_1\,\partial x_2} & \cdots & \dfrac{\partial^2 f}{\partial x_1\,\partial x_n} \\ \dfrac{\partial^2 f}{\partial x_2\,\partial x_1} & \dfrac{\partial^2 f}{\partial x_2^2} & \cdots & \dfrac{\partial^2 f}{\partial x_2\,\partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \dfrac{\partial^2 f}{\partial x_n\,\partial x_1} & \dfrac{\partial^2 f}{\partial x_n\,\partial x_2} & \cdots & \dfrac{\partial^2 f}{\partial x_n^2} \end{bmatrix}$$

$$\langle x, Ax \rangle \geq 0$$

Hessian of LSM

$$\mathcal{L}(w) = \|y - X^T w\|_2^2$$

$$\nabla_w \|y - X^T w\|_2^2 = -2X(y - X^T w)$$

Hessian of LSM

$$\nabla_w \nabla_w \|y - X^T w\|_2^2 = -\nabla_w 2X(y - X^T w) = 2XX^T$$

Why is positive definite? $\langle x, Ax \rangle \geq 0$

Geometric interpretation of the solution and model complexity

$$\hat{y} = X^T w$$

$$w* = (XX^T)^{-1} Xy$$

$$\hat{y}* = X^T (XX^T)^{-1} Xy$$

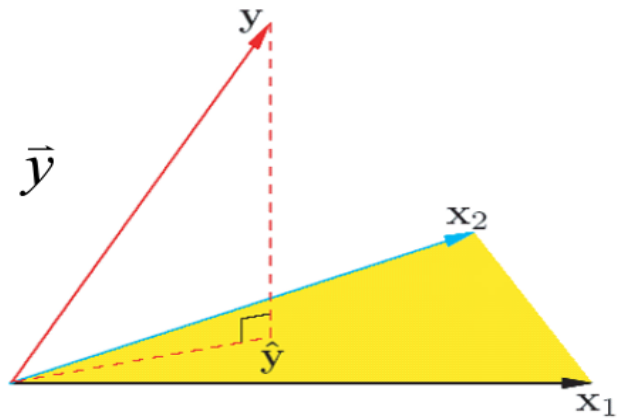
$$y - \hat{y}* = (I - X^T (XX^T)^{-1} X) y$$

$$X(y - \hat{y}*) = X(I - X^T (XX^T)^{-1} X) y = 0$$

Geometric interpretation

The difference of the estimated and real y is orthogonal to the span of X

$\hat{\vec{y}}$ is the orthogonal projection of $\vec{y}$ into the space spanned by the columns of X



Why we do care? This is a *limitation of the expressive power* of a linear model. Was evident from the beginning, why?

- The predictions on the training data are:

$$\hat{\vec{y}} = X\theta^* = X\left(X^T X\right)^{-1} X^T \vec{y}$$

- Note that

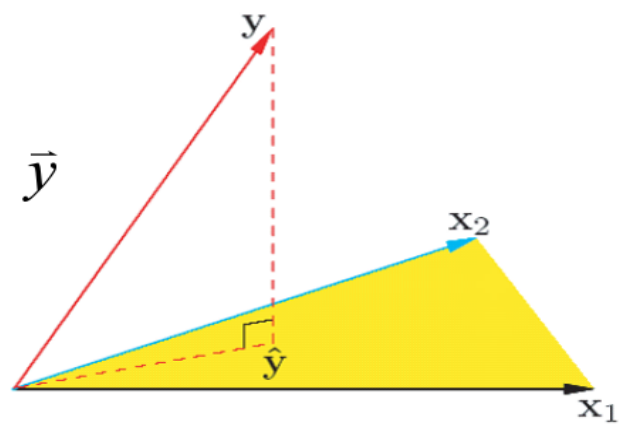
$$\hat{\vec{y}} - \vec{y} = \left(X\left(X^T X\right)^{-1} X^T - I\right)\vec{y}$$

and

$$\begin{aligned} X^T\left(\hat{\vec{y}} - \vec{y}\right) &= X^T\left(X\left(X^T X\right)^{-1} X^T - I\right)\vec{y} \\ &= \left(X^T X\left(X^T X\right)^{-1} X^T - X^T\right)\vec{y} \\ &= 0 \text{ !!} \end{aligned}$$

$\hat{\vec{y}}$ is the orthogonal projection of $\vec{y}$ into the space spanned by the columns of X

$$\vec{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} \qquad \mathbf{X} = \begin{bmatrix} -- & \mathbf{x}_1 & -- \\ -- & \mathbf{x}_2 & -- \\ \vdots & \vdots & \vdots \\ -- & \mathbf{x}_n & -- \end{bmatrix}$$



Other limitations

$$\begin{aligned}\nabla_w \|y - X^T w\|_2^2 &= \nabla_w [(y - X^T w)^T (y - X^T w)] \\ &= 2\nabla_w [(y - X^T w)](y - X^T w)] \\ &= -2X(y - X^T w)\end{aligned}$$

$$2X(y - X^T w) = 0 \quad \longrightarrow \quad w = (XX^T)^{-1} Xy$$

What could go possibly wrong?

N<d: not enough data to estimate the parameters

$$X \in \mathbb{R}^{d \times N} \quad XX^T \in \mathbb{R}^{d \times d}$$

XX^T is not invertible (why: too small rank, not bijective)

SOLUTION: regularization

Proof that for a “slim” A we have problems with AA^T

$$rk(A) = dim(img(A))$$

$$rk(A) = rk(AA^T)$$

- Define $\langle x, y \rangle = x^T A^T A y$
- Is positive definite
- If $x \in Ker(A^T A)$ then $x \in Ker(A)$
- However in general $Ker(A) \subset Ker(BA)$
- Then $Ker(A^T A) = Ker(A)$
- $A \to A^T$

What other things could possibly go wrong?

Columns are linearly dependent
(Data are highly correlated)

What about the offset?

$$\mathcal{L}(w) = \|y - X^T w - b\|_2^2$$

$$X \to \begin{pmatrix} X \\ 1^T \end{pmatrix}$$

$$w \to \begin{pmatrix} w \\ b \end{pmatrix}$$

$$\|y - X^T w\|_2^2$$

Dealing with the offset

$$XX^T w = Xy \rightarrow \begin{pmatrix} X \\ 1^T \end{pmatrix} \begin{pmatrix} X \\ 1^T \end{pmatrix}^T \begin{pmatrix} w \\ b \end{pmatrix} = \begin{pmatrix} X \\ 1^T \end{pmatrix} y$$

$$\begin{pmatrix} XX^T & X1 \\ 1^T X^T & 1^T 1 \end{pmatrix} \begin{pmatrix} w \\ b \end{pmatrix} = \begin{pmatrix} Xy \\ 1^T y \end{pmatrix}$$

$$w* = (XX^T)^{-1}(Xy - X1b)$$

$$b = (1^T X^T w * -1^T y)/d$$

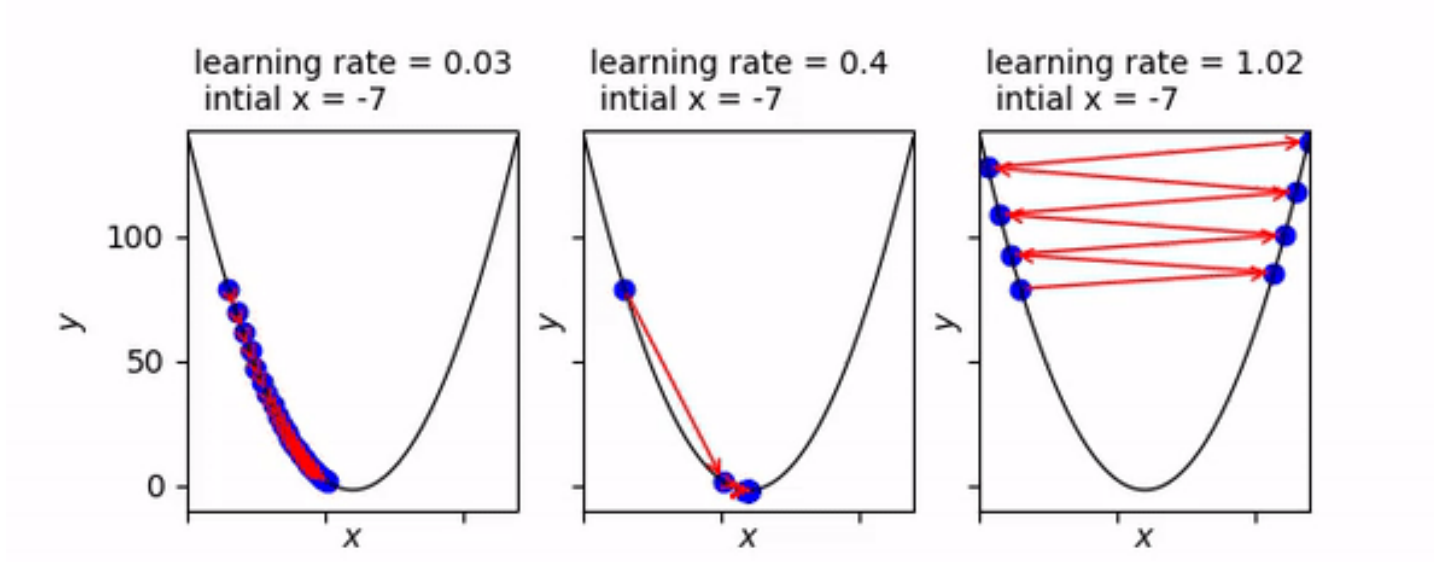
Gradient descent approach

$$w_{t+1} = w_t - \gamma \frac{d\mathcal{L}(w)}{dw}$$

$$w_{t+1} = w_t + \gamma \, 2X(y - X^T w)$$

Which gamma should I use?

Importance of step size



Which step size should we use?

Taylor expansion

$$f(x) = \sum_{k=0}^{\infty} \frac{1}{k!} f^{(k)}(c)\,(x-c)^k$$

$$f(y) = f(x) + \langle \nabla f(x), y - x \rangle + \frac{1}{2} \langle y - x, \nabla^2 f(x)(y - x) \rangle + \cdots$$

Which step size should we use?

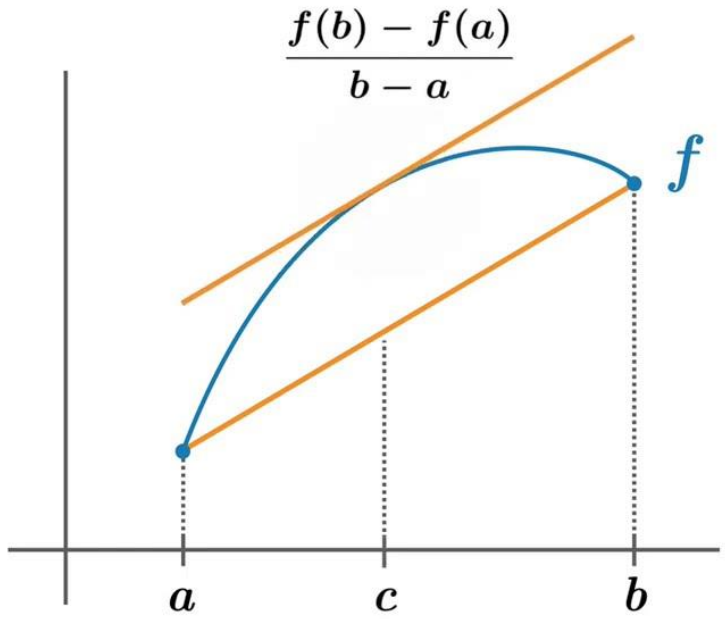
The Mean Value Theorem

Suppose f is continuous on $[a, b]$ and differentiable on (a, b). Then there exists some argument c such that

$$a < c < b$$

and

$$f'(c) = \frac{f(b) - f(a)}{b - a}$$



$$\nabla f(y) - \nabla f(x) = \nabla^2 f(z)(y - x)$$

Lipschitz continuity

$$\|\nabla f(y) - \nabla f(x)\| \leq L\|y - x\|$$

Which step size should we use?

$$f(y) = f(x) + \langle \nabla f(x), y - x \rangle + \frac{1}{2} \langle y - x, \nabla^2 f(x)(y - x) \rangle + \cdots$$

$$\nabla f(y) - \nabla f(x) = \nabla^2 f(z)(y - x)$$

Take the norm use Cauchy Schwarz and use Lipschitz

$$\|\nabla f(y) - \nabla f(x)\| \leq L \|y - x\|$$

$$f(y) \leq f(x) + \langle \nabla f(x), y - x \rangle + \frac{L}{2} \|y - x\|^2$$

Which step size should we use?

Let y be the one step of gradient descent from x, i.e., $y = x - \alpha \nabla f(x)$. Since f is Lipschitz gradient function with constant L, we have

$$\begin{aligned} f(y) - f(x) &\leq \langle \nabla f(x), y - x \rangle + \frac{L}{2} \|y - x\|^2 \\ &= -\alpha \|\nabla f(x)\|^2 + \frac{L}{2} \alpha^2 \|\nabla f(x)\|^2. \end{aligned}$$

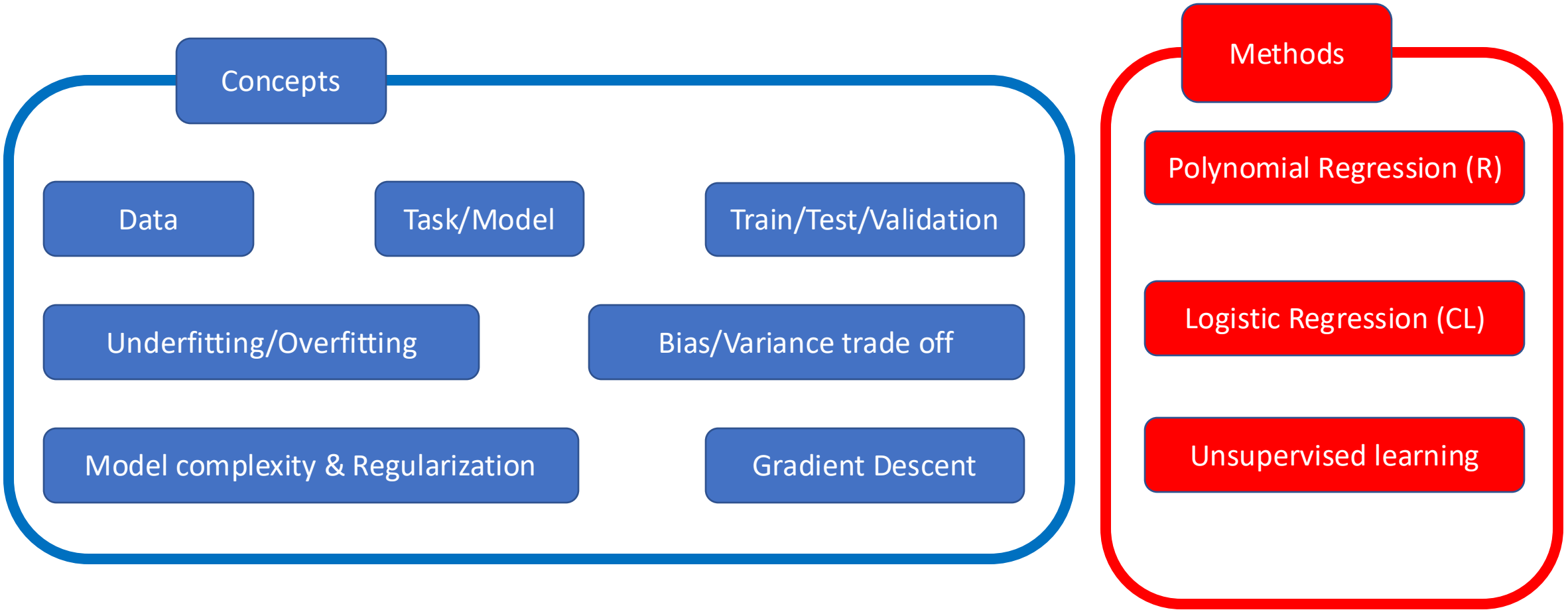
To guarantee the sufficient decreasing, we need

$$\frac{L}{2} \alpha^2 - \alpha \leq 0 \Rightarrow \alpha \leq \frac{2}{L}.$$

Summarizing

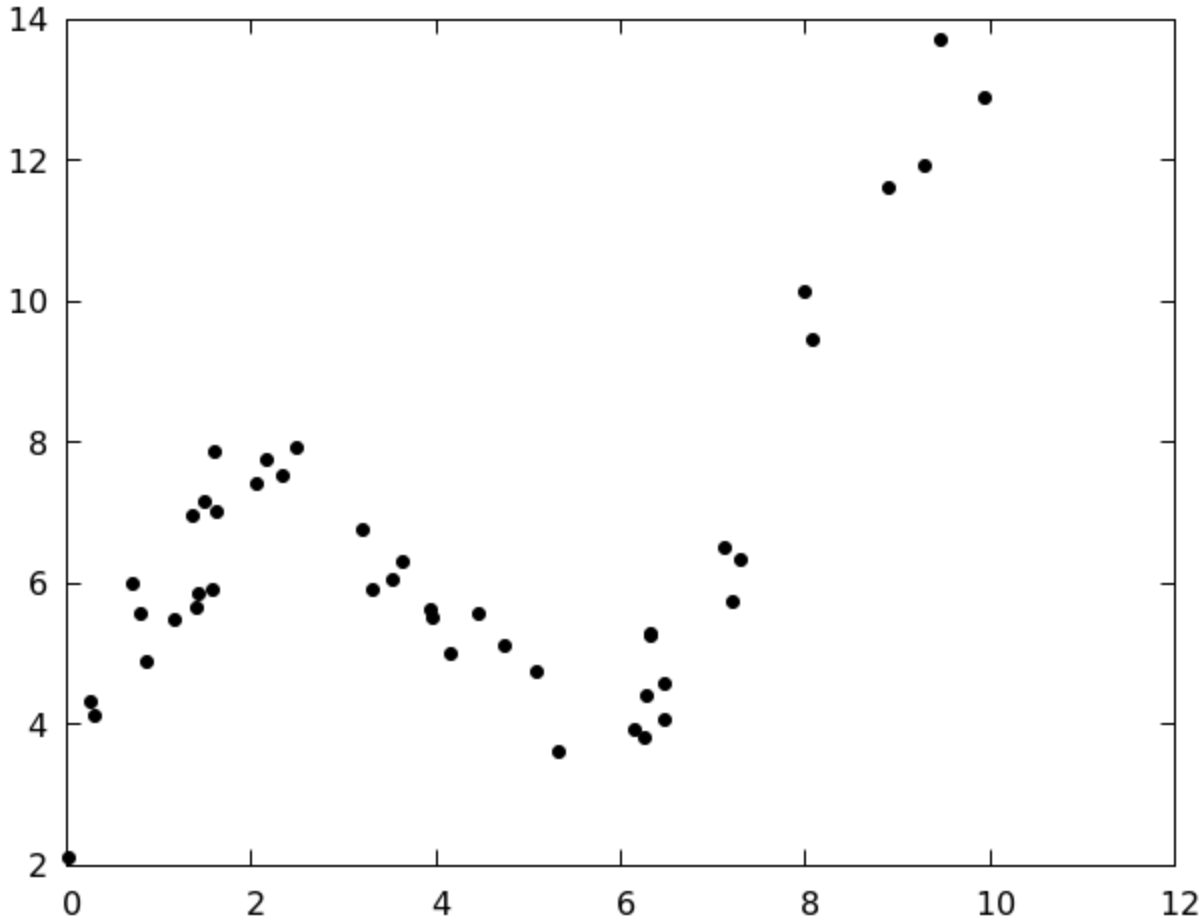
- Closed form solution of linear regression $w = (XX^T)^{-1}Xy$
- Interpretation and problems of the solution
- Convexity and uniqueness of the solution $2XX^T$
- Offset
$$w* = (XX^T)^{-1}(Xy - X1b)$$
$$b = (1^T X^T w * -1^T y)/d$$
- Stepsize $\alpha \leq \frac{2}{L}$

Summarizing…



Applying linear regression…

$$w = (XX^T)^{-1}Xy$$

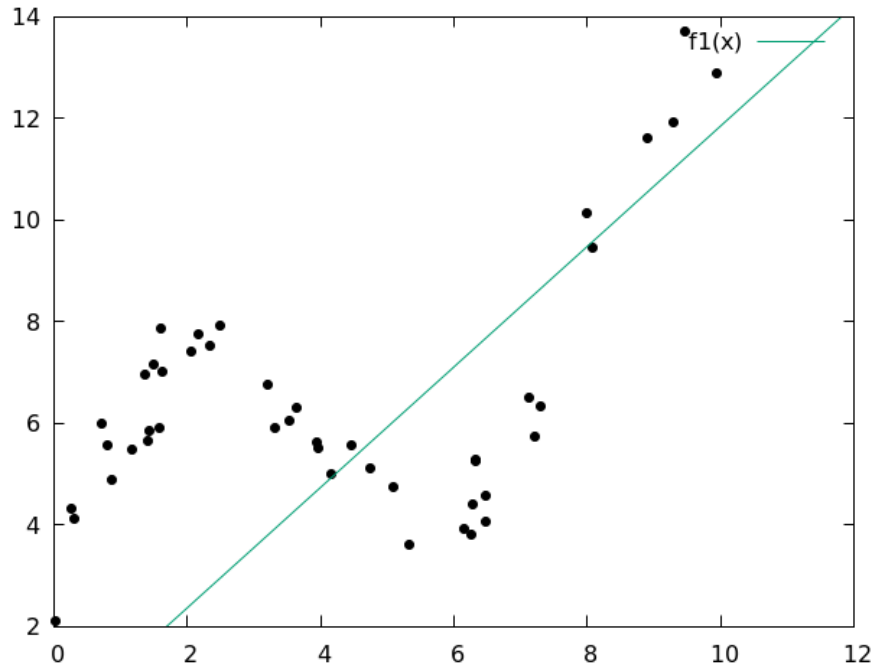


Applying linear regression…

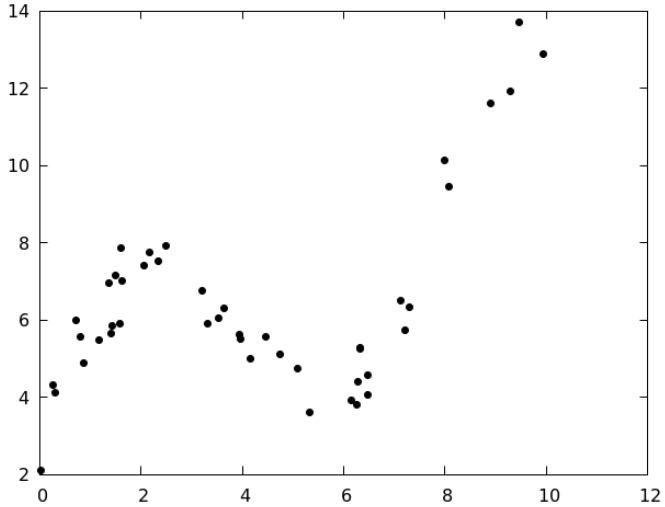
$X = (6.1, 0.2, 10.1, \ldots, 4, 5, 1.3)$

$y = (4.1, 3.9, 13.6, \ldots, 5.6, 5.3)$

$$w = (XX^T)^{-1}Xy$$



How can I do better with the same model?



GOING POLYNOMIAL!

Applying ~~linear~~ polynomial regression…

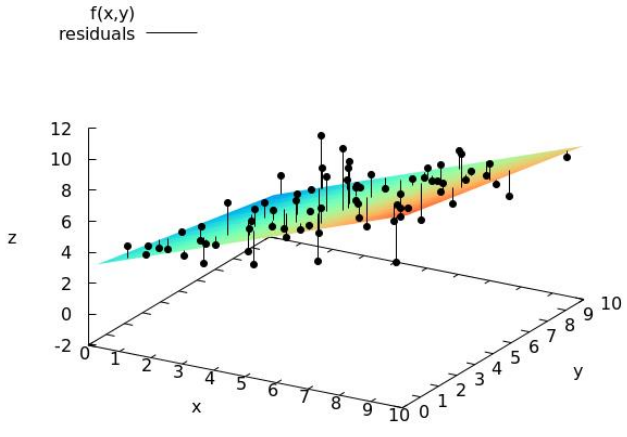
$$y = w_0 x^0 + w_1 x^1 + w_2 x^2 + \cdots$$

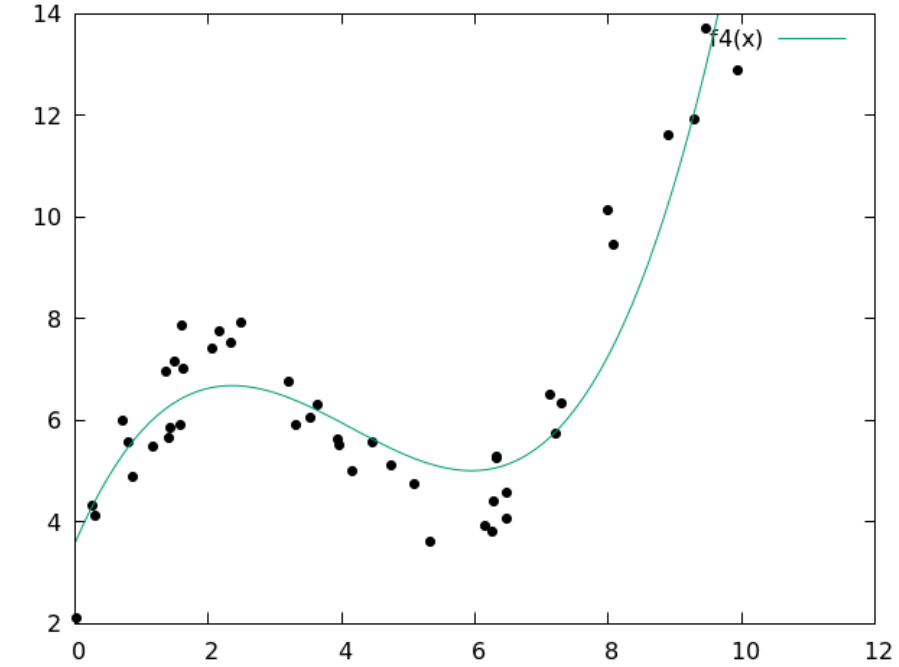
Note that this is the bias term (b)

$$X = \begin{pmatrix} 6.1 & \dots & 1.3 \\ 37.21 & \dots & 1.69 \\ 1 & \dots & 1 \end{pmatrix}$$

$$w = (XX^T)^{-1}Xy$$

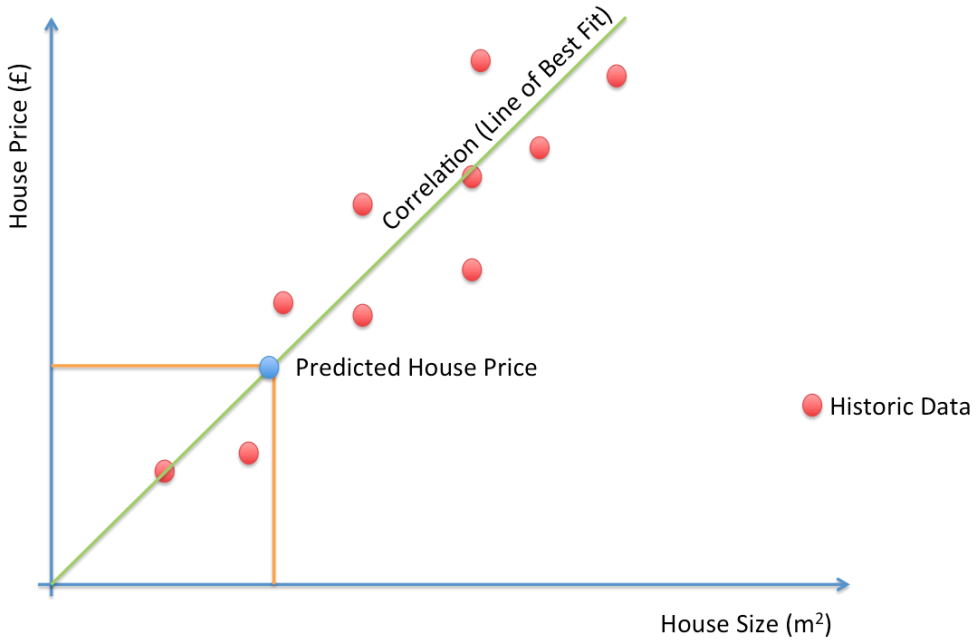
$$y = (4.1, 3.9, 13.6, \dots, 5.6, 5.3)$$

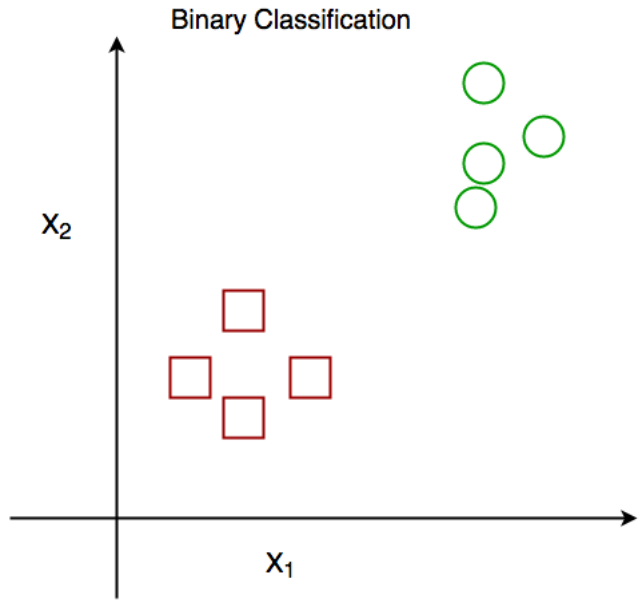




Logistic regression: classification, gradient descent (derivation).

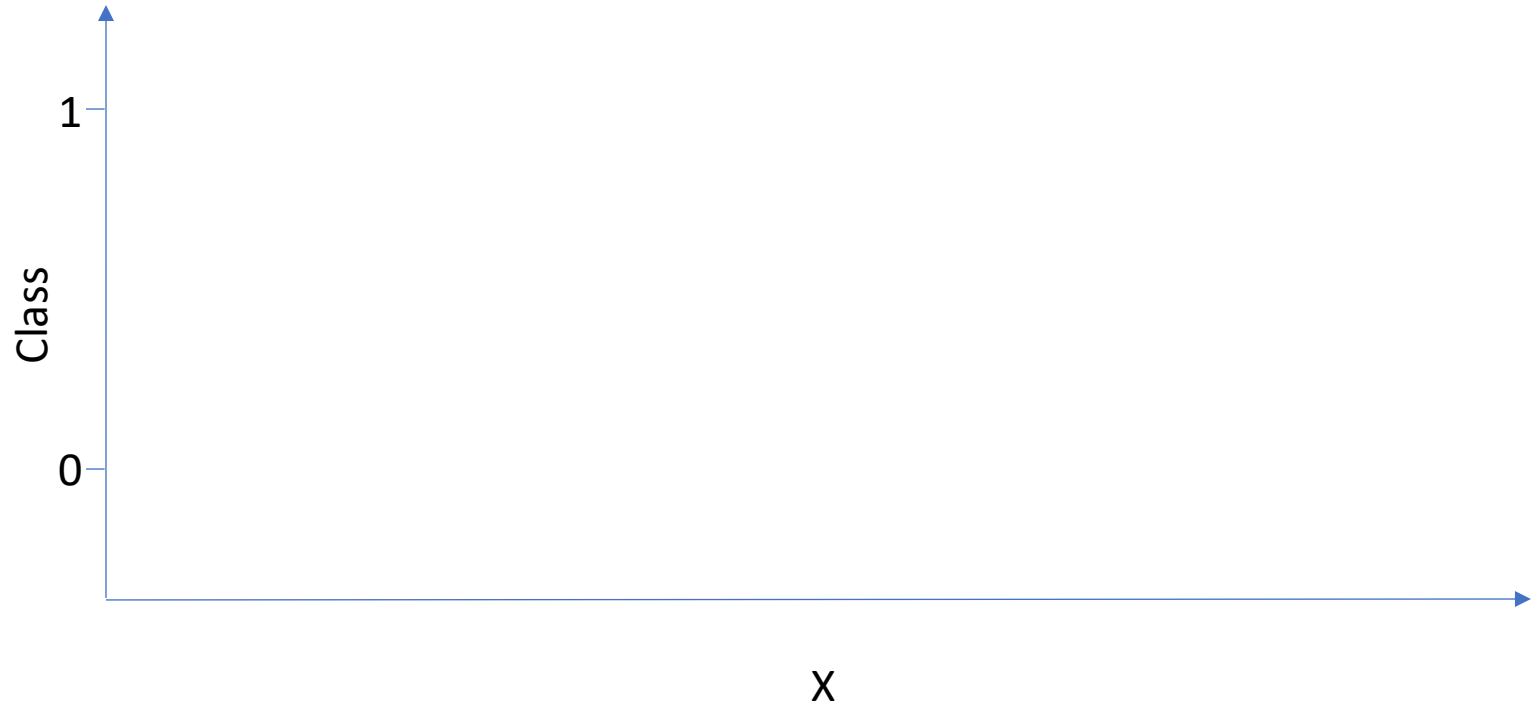
Regression vs classification, a different task





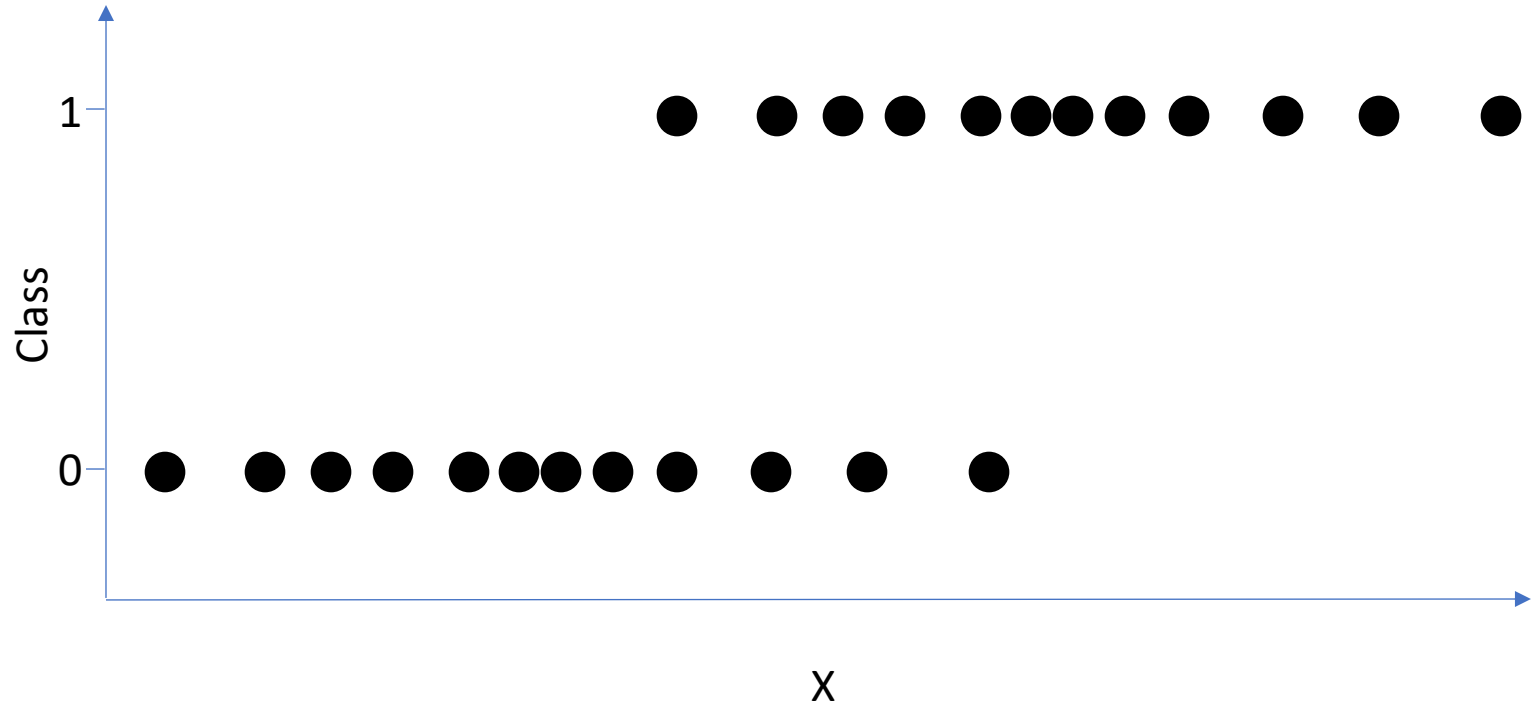
Can we use the same methods? Applying linear regression to a binary classification problem.

1. Map the two classes onto 0-1



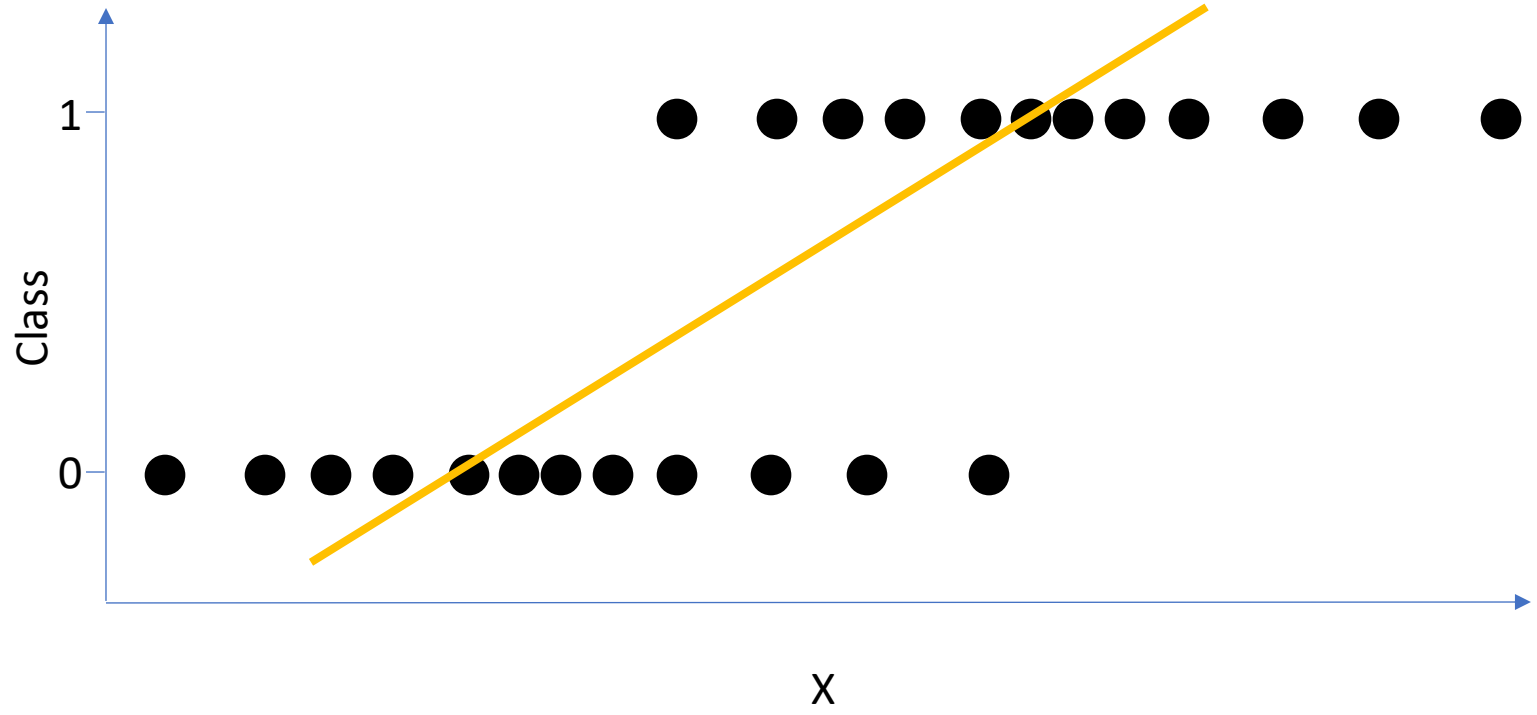
Can we use the same methods? Applying linear regression to a binary classification problem.

1. Map the two classes onto 0-1



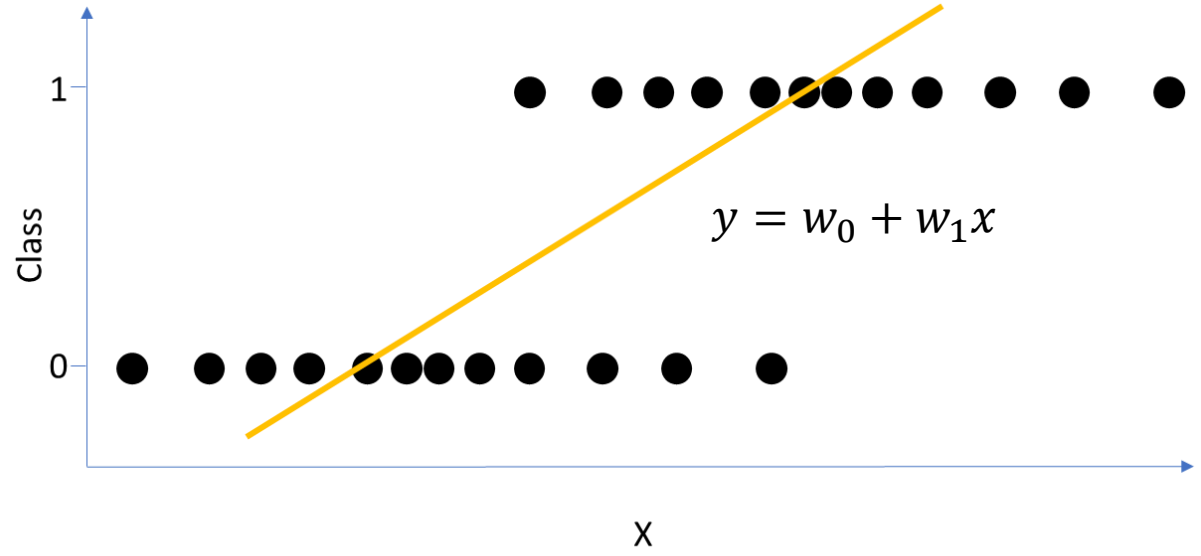
Can we use the same methods? Applying linear regression to a binary classification problem.

2. Fit the equation $y = w_0 + w_1 x$



What is not working in this approach?

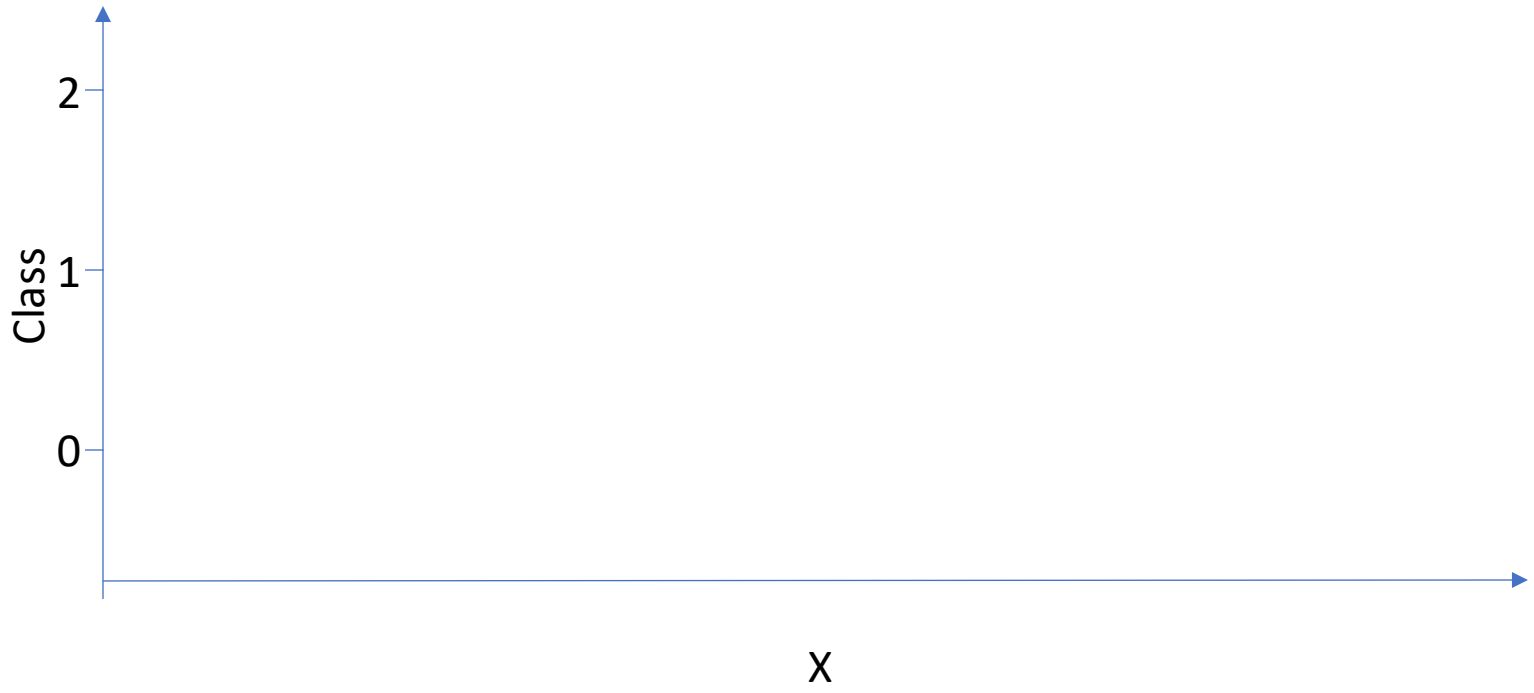
Can we use the same methods? Applying linear regression to a binary classification problem.



- What's the meaning of y?
- What happens for $y > 1$ or $y < 0$?
- And how should we interpret w_0 or w_1?
- What if I don't care about interpretability?

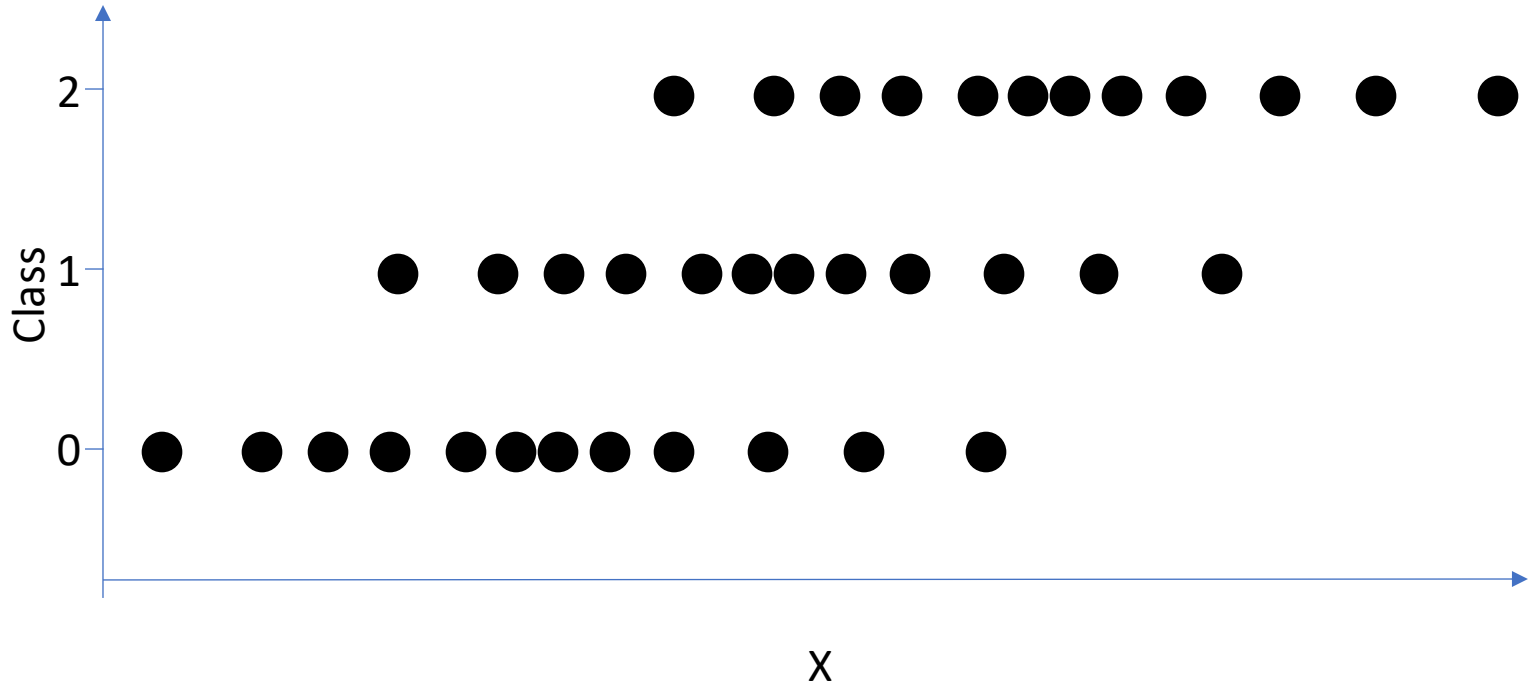
But classification task is not always binary. Let's see what happens with 3 classes?

1. Map the three classes onto 0-1-2



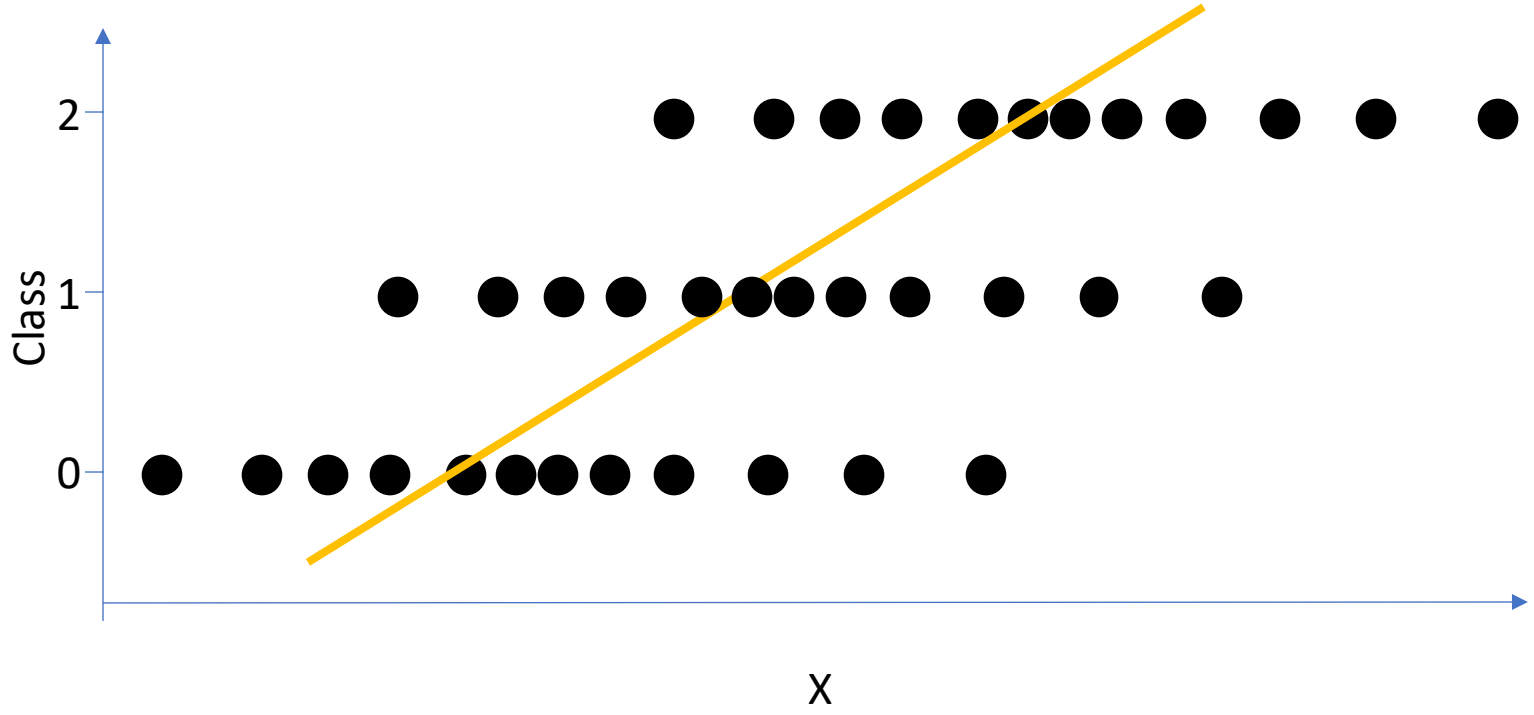
Let’s see what happens with 3 classes

1. Map the three classes onto 0-1-2

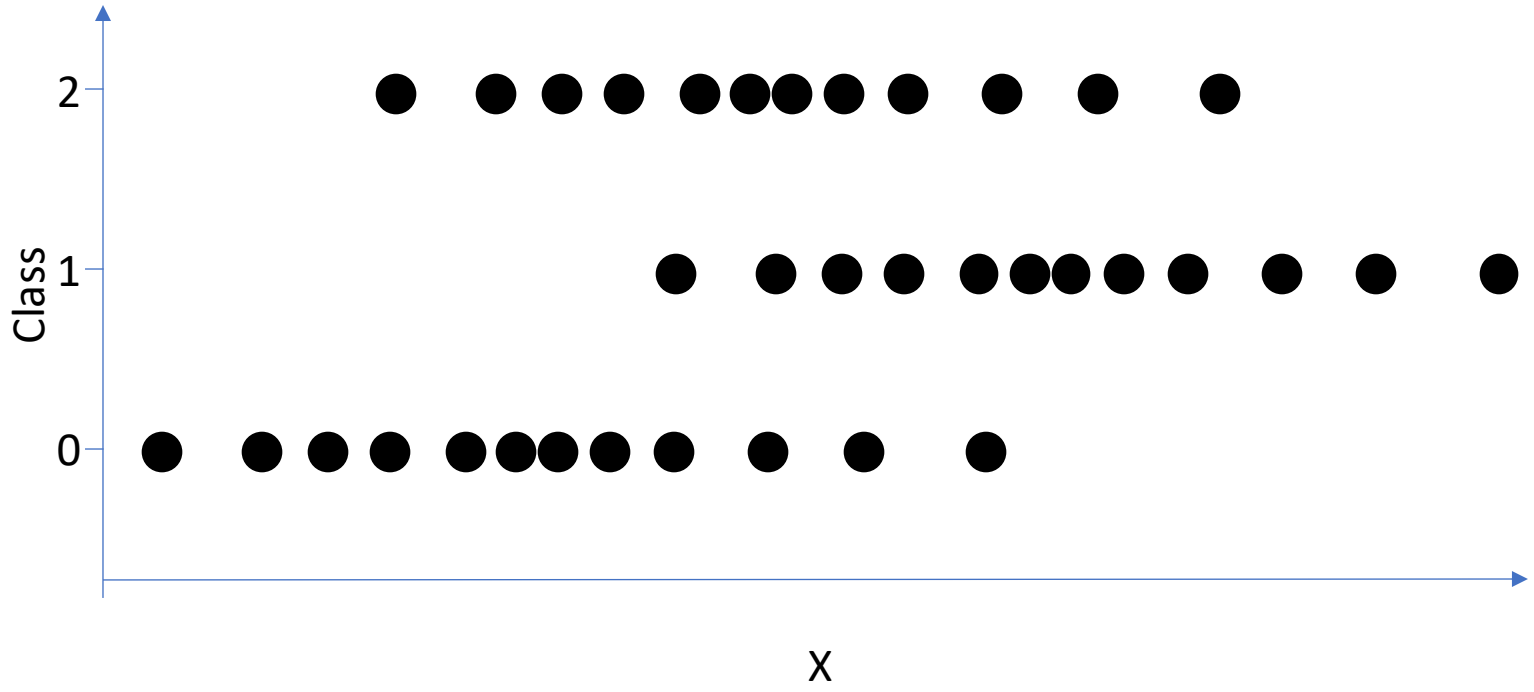


Let’s see what happens with 3 classes

2. Fit the equation $y = w_0 + w_1 x$



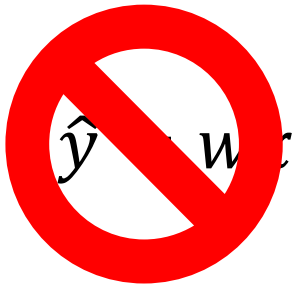
What if I change the mapping?



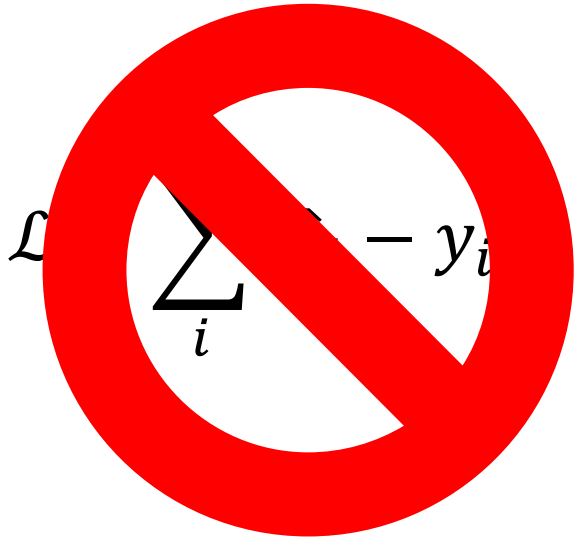
I need a loss that allows me to take this invariance into account!

We need…

- A different model for my data.



- A different loss.



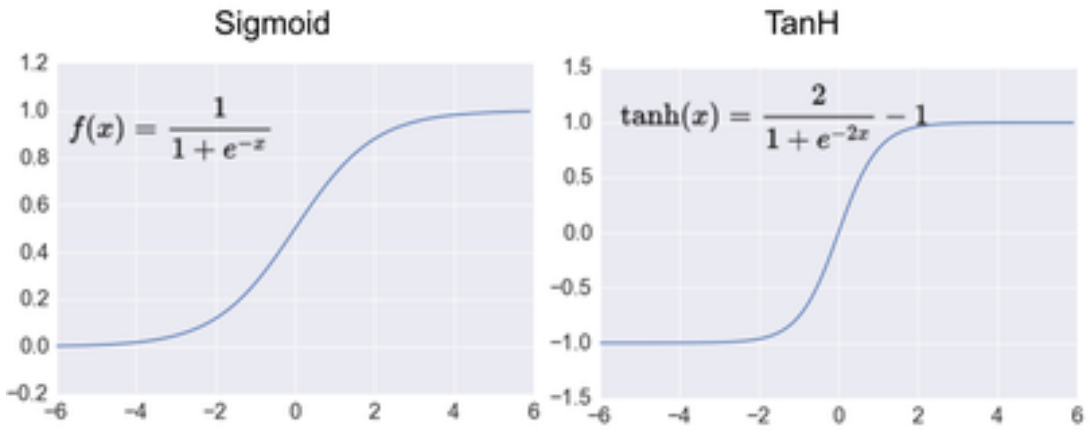
The logistic model for a binary classification:

We want to use a linear model (regression) but with (0,1) output: how?

Regression **Classification (probability)**

$$[-\infty, +\infty] \quad \rightarrow \quad [0, 1]$$

Squashing functions



Logistic Function

$$p(y|x) = \frac{1}{1 + e^{w^T x}}$$

Properties of the logistic function

If $p(y=0|x) = \frac{1}{1+e^{w^T x}}$ what function is $p(y=1|x)$?

$$p(y=1|x) = 1 - \frac{1}{1+e^{w^T x}} = \frac{e^{w^T x}}{1+e^{w^T x}}$$

What happens at $p(y=1|x) = p(y=0|x)$?

Properties of the logistic function

Classification (probability) **Regression**

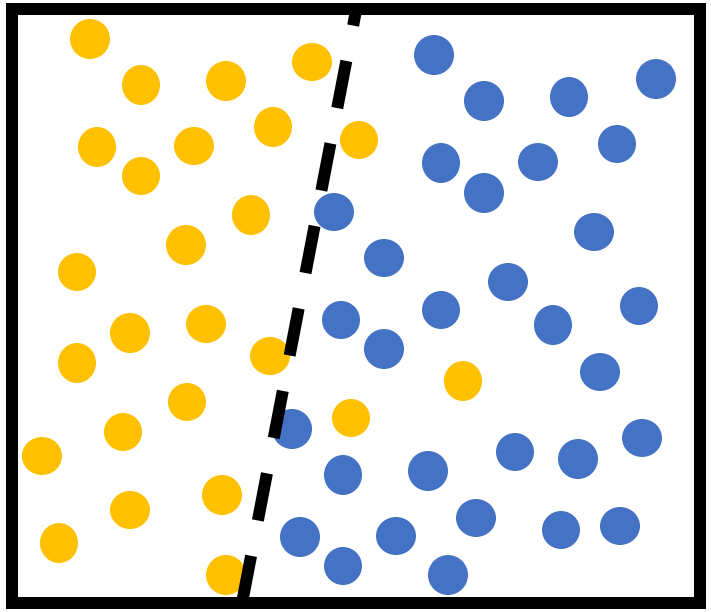
$$p(y = 0|x) = p(y = 1|x)$$

$$[0, 1] \rightarrow [-\infty, +\infty]$$

$$\frac{p(y = 1|x)}{p(y = 0|x)} = 1 = \frac{\frac{e^{w^T x}}{1 + e^{w^T x}}}{\frac{1}{1 + e^{w^T x}}} = e^{w^T x}$$

$$\log\left(\frac{p(y = 1|x)}{p(y = 0|x)}\right) = w^T x = 0$$

Log of the odds, or logit



Direct interpretation of the coefficients

$$\log\left(\frac{p(y=1|x)}{p(y=0|x)}\right) = w^T x = 0$$

Log of the odds, or logit

Keeping all the other quantities fixed, changing x_i by one unit, changes the logit by w_i

How do we write an associated loss?

Two ways:

1) Derive it from a Maximum Likelihood approach.
2) Difference between true and estimated probability distributions.

Maximum Likelihood

- Likelihood: Product of event probabilities.
- For simplicity, let's take $p(y = 1|x_i) = p(x_i)$ *(*$p(y = 1|x_i)$ *is* $p(y = 1|x)$ *evaluated at the point* x_i*)*
- I can write the likelihood as:

$$\mathcal{L} = \prod_{i:y_i=1} p(x_i) \prod_{i:y_i=0} \big(1 - p(x_i)\big)$$

Using $a^0 = 1$

$$\mathcal{L} = \prod_{i} p(x_i)^{y_i} \big(1 - p(x_i)\big)^{1-y_i}$$

Maximum Likelihood

$$\mathcal{L} = \prod_i p(x_i)^{y_i} \left(1 - p(x_i)\right)^{1-y_i}$$

Taking logarithm

$$\mathcal{L} = \sum_i y_i \log p(x_i) + (1 - y_i) \log\left(1 - p(x_i)\right)$$

How would you transform a maximization problem in a minimization one?

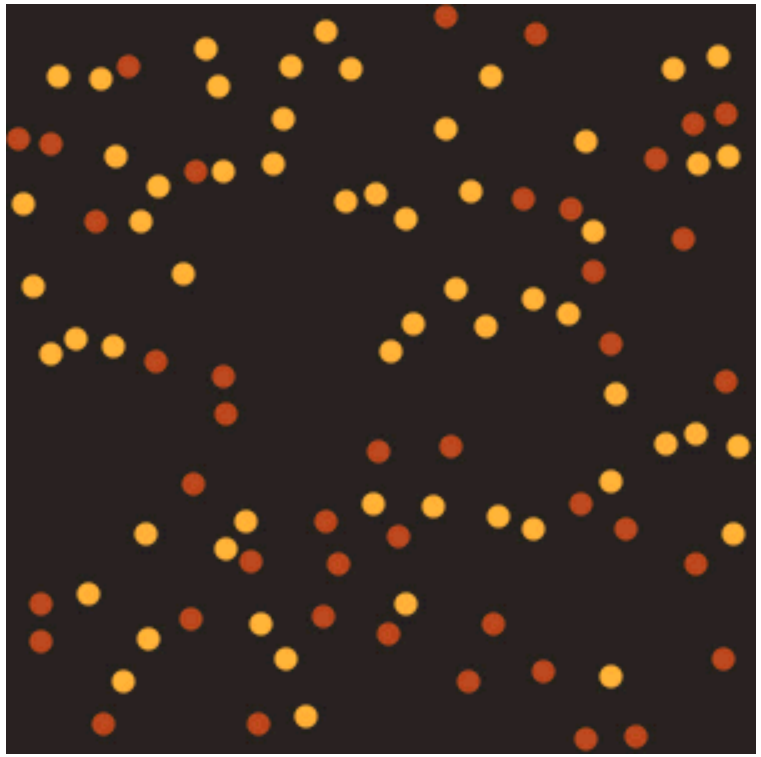
A deeper view: Entropy as information measure

Let H be our "measure".

1. H is continous at every p_i
2. If $p_i = 1$, then H is minimum with a value of 0, no uncertainty.
3. If $p_1 = p_2 = \cdots = p_n$, i.e. $p_i = \frac{1}{n}$, then H is maximum. In other words, when every outcome is equally likely, the uncertainty is greatest, and hence so is the entropy.
4. If a choice is broken down into two successive choices, the value of the original H should be the weighted sum of the value of the two new ones.

The only H satisfying the conditions above is:

$$H = -K \sum_{i=1}^{n} p_i log(pi)$$



Is the entropy of the system minimal or maximal?

Cross entropy: distance between distributions

$$H(q,p) = -\frac{1}{N}\sum_i p_i \log(q_i)$$

p is the true distribution.
q s the estimated distribution.
$H(q,p)$ is a measure of the information needed to encode p with q (if both are identical, $H(q,p) = 0$)

Binary cross entropy

$$\hat{y} = q(y = 1|x) = \frac{1}{1 + e^{-w^T x}}$$

$$1 - \hat{y} = q(y = 0|x) = 1 - q(y = 1|x)$$

$$p \in \{y, 1 - y\}, \quad q \in \{\hat{y}, 1 - \hat{y}\}$$

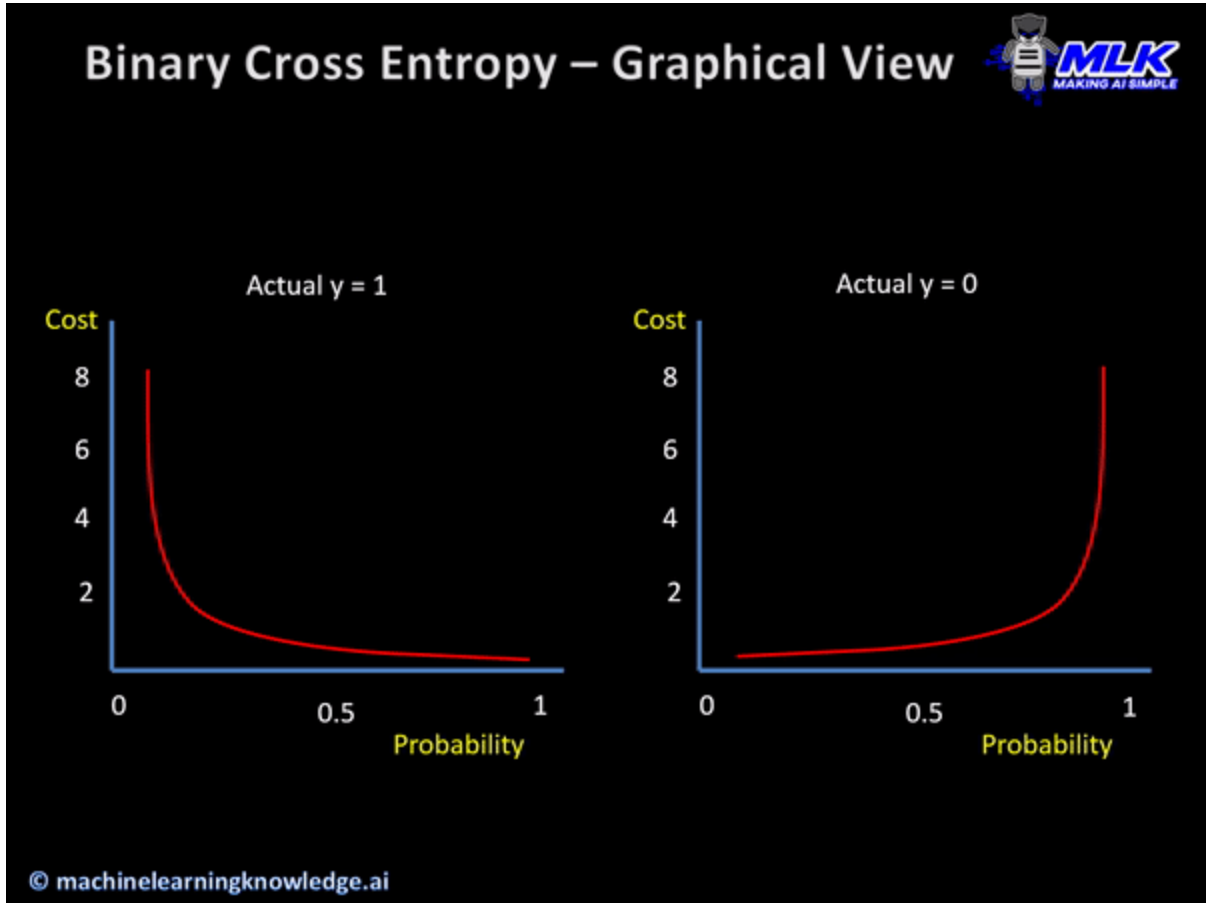
$$H(q, p) = -\frac{1}{N} \sum_i p_i \log(q_i) = \frac{1}{N} \sum_i -y_i \log(\hat{y}_i) - (1 - y_i) \log(1 - \hat{y}_i)$$

Minimizing the cross entropy is equivalent to maximize the likelihood

$$y_j \log h_{\boldsymbol{\theta}}(\boldsymbol{x}_j) = \begin{cases} 0, & \text{if} \quad y_j = 1, \quad \text{and} \quad h_{\boldsymbol{\theta}}(\boldsymbol{x}_j) = 1, \\ -\infty, & \text{if} \quad y_j = 1, \quad \text{and} \quad h_{\boldsymbol{\theta}}(\boldsymbol{x}_j) = 0, \end{cases}$$

$$(1 - y_j)(1 - \log h_{\boldsymbol{\theta}}(\boldsymbol{x}_j)) = \begin{cases} 0, & \text{if} \quad y_j = 0, \quad \text{and} \quad h_{\boldsymbol{\theta}}(\boldsymbol{x}_j) = 0, \\ -\infty, & \text{if} \quad y_j = 0, \quad \text{and} \quad h_{\boldsymbol{\theta}}(\boldsymbol{x}_j) = 1. \end{cases}$$

Why is a good Loss?



Derivative of the Loss

$$\mathcal{L} = -\frac{1}{N}\sum_i y_i \log p(x_i) + (1 - y_i) \log\big(1 - p(x_i)\big)$$

$$\mathcal{L} = -\frac{1}{N}\sum_i y_i \log \frac{p(x_i)}{1 - p(x_i)} + \log\big(1 - p(x_i)\big)$$

$$\mathcal{L} = -\frac{1}{N}\sum_i y_i (w^T x_i) - \log\left(1 + e^{w^T x_i}\right)$$

Exercise: Derive the expression for the gradient

$$p(x_i) = p(y = 1|x_i) = \frac{e^{w^T x}}{1 + e^{w^T x}}$$

$$\log \frac{p(y = 1|x)}{p(y = 0|x)} = w^T x$$

$$1 - p(x_i) = p(y = 0|x_i) = \frac{1}{1 + e^{w^T x}}$$

Derivative of the Loss

$$\mathcal{L} = -\frac{1}{N}\sum_i y_i(w^T x_i) - \log\left(1 + e^{w^T x_i}\right)$$

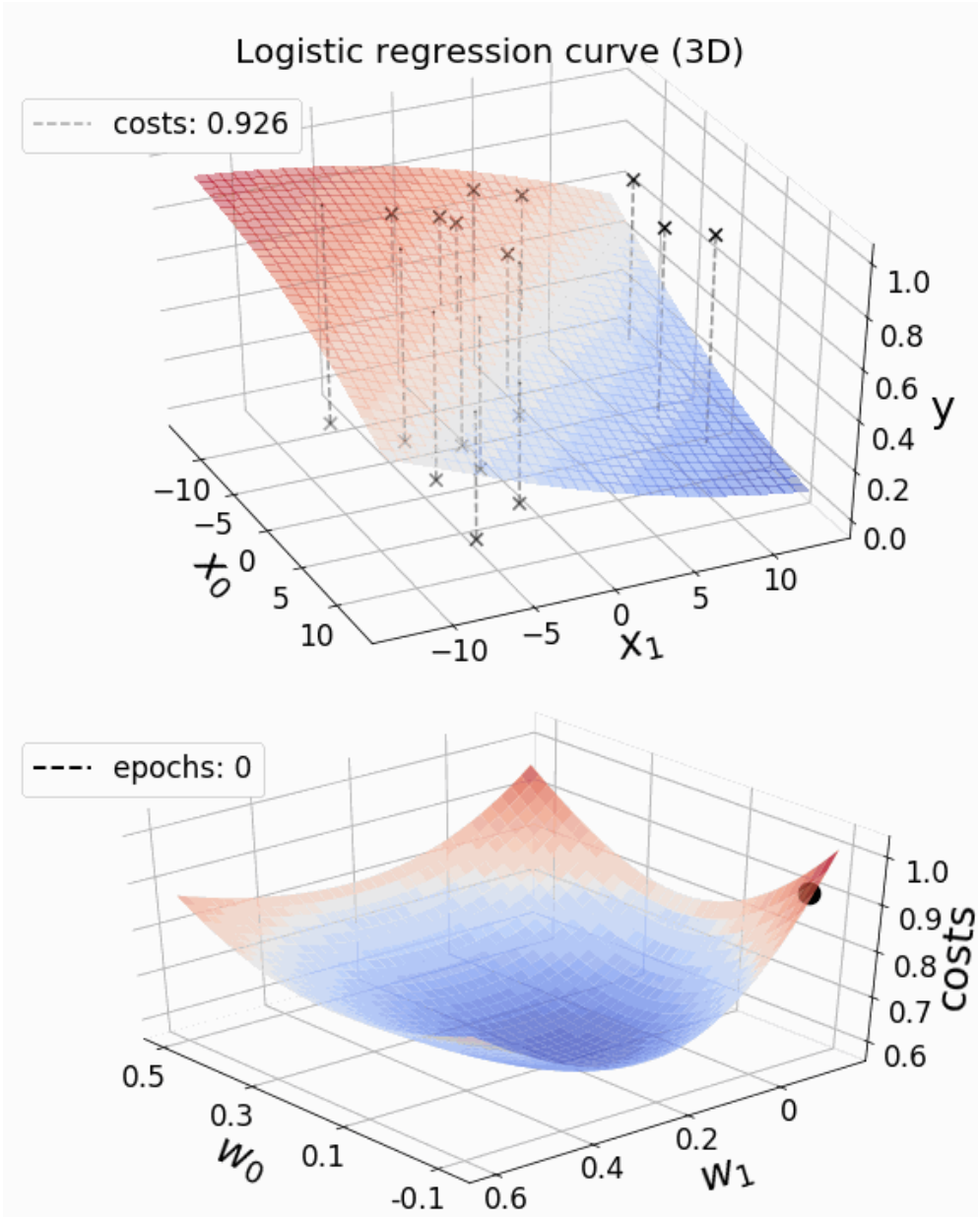
$$\frac{\partial \mathcal{L}}{\partial w} = \frac{1}{N}\sum_i -y_i x_i + \frac{x_i e^{w^T x_i}}{1 + e^{w^T x_i}}$$

$$\frac{\partial \mathcal{L}}{\partial w} = \frac{1}{N}\sum_i x_i\left(\frac{1}{1 + e^{-w^T x_i}} - y_i\right)$$

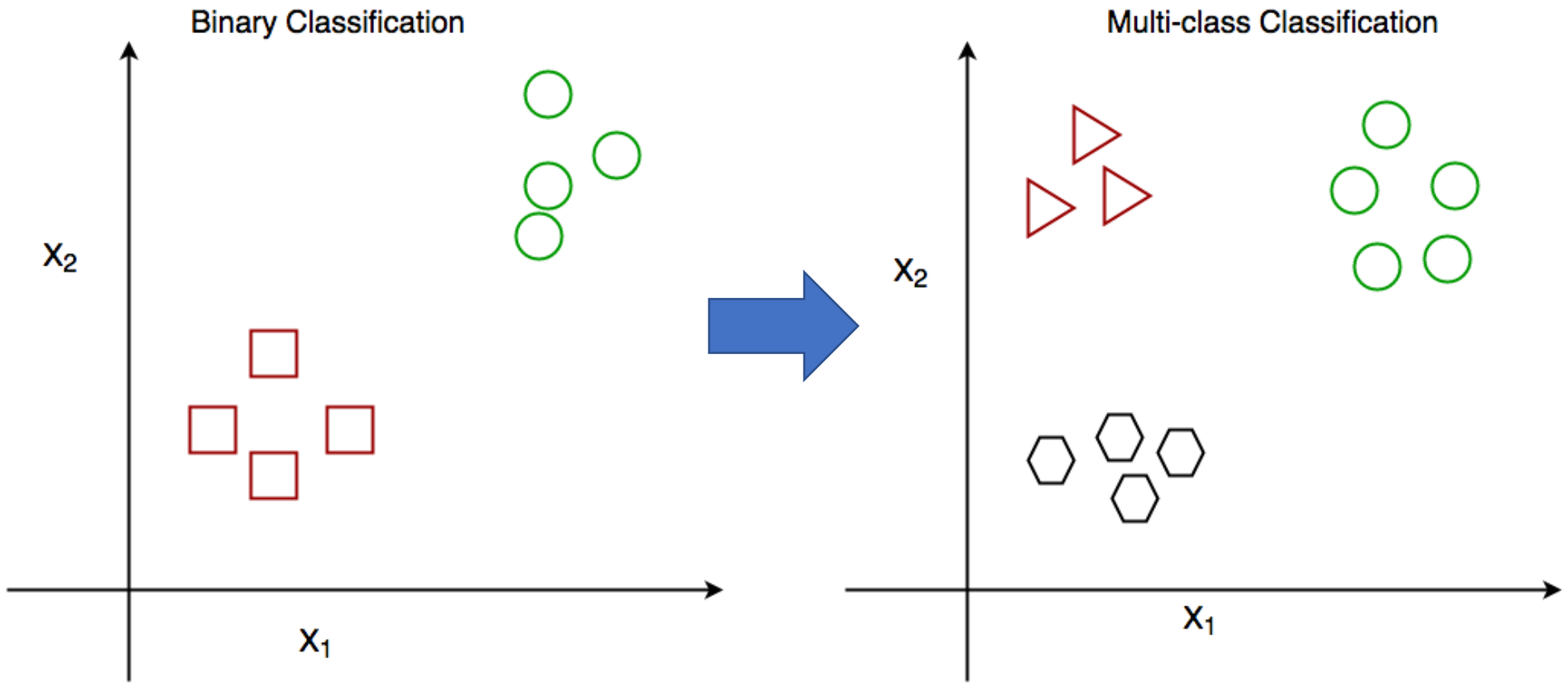
Gradient descent of the logistic regression

$$w_{t+1} = w_t - \gamma \frac{\partial \mathcal{L}}{\partial w} = w_t - \gamma \left(\frac{1}{N} \sum_i x_i \left(\frac{1}{1 + e^{-w^T x_i}} - y_i \right) \right)$$

Gradient descent on logistic loss



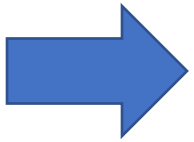
Extending the logistic model: Multinomial regression to K classes



Extending the logistic model: Multinomial regression to K classes

$$p(y=0|x) = \frac{1}{1+e^{w^T x}}$$

$$p(y=1|x) = \frac{e^{w^T x}}{1+e^{w^T x}}$$



$$p(y=1|x) = \frac{e^{w_1^T x}}{1+\sum_j^K e^{w_j^T x}}$$

$$p(y=2|x) = \frac{e^{w_2^T x}}{1+\sum_j^K e^{w_j^T x}}$$

…

$$p(y=K-1|x) = \frac{e^{w_{K-1}^T x}}{1+\sum_j^K e^{w_j^T x}}$$

$$p(y=K|x) = \frac{1}{1+\sum_j^K e^{w_j^T x}}$$

Extending the logistic model: Multinomial regression to K classes

$$\log\left(\frac{p(y=1|x)}{p(y=0|x)}\right) = w^T x \quad \Rightarrow \quad \log\left(\frac{p(y=j|x)}{p(y=K|x)}\right) = w_j^T x$$

$$\mathcal{L} = -\sum_{j=1}^{M}\sum_{i}\left(\delta_{j,y_i}\log\left(p_j(x_i)\right) + \left(1-\delta_{j,y_i}\right)\log\left(1-p_j(x_i)\right)\right)$$

What happens with this loss?

if $j = y_i$: $\delta_{j,y_i} = 1$; else $\delta_{j,y_i} = 0$

$p_j(x_i) = p(y = j|x_i)$

Few words on **stochastic gradient descent** (SGD)

Two facts about Gradient Descent method:

- Gradient descent can easily got trapped in local minima.
- Gradient descent can be slow (we need to perform sums over all the elements…) $w_{t+1} = w_t - \gamma \frac{\partial \mathcal{L}}{\partial w} = w_t - \gamma \left(\frac{1}{N} \sum_i x_i \left(\frac{1}{1 + e^{-w^T x_i}} - y_i \right) \right)$

STOCHASTIC GRADIENT DESCENT (SGD):

- Estimate the gradient in small batches of randomly selected data points and perform a step.
- The added noise reduces the chances of getting trapped.
- The minimization is fastest because many steps are done before the derivative is computed for all the points (epoch).

Previously…

Linear Regression:

- $\hat{y} = X^T w$
- $\mathcal{L} = \|\hat{y} - y\|^2 = \|X^T w - y\|^2$
- $w = (XX^T)^{-1} Xy$
- Convex solution ($\mathbb{H} = 2XX^T$) and possible problems.
- GD solution and considerations about the stepsize
- Offset
- From linear to polynomial

Logistic Regression:

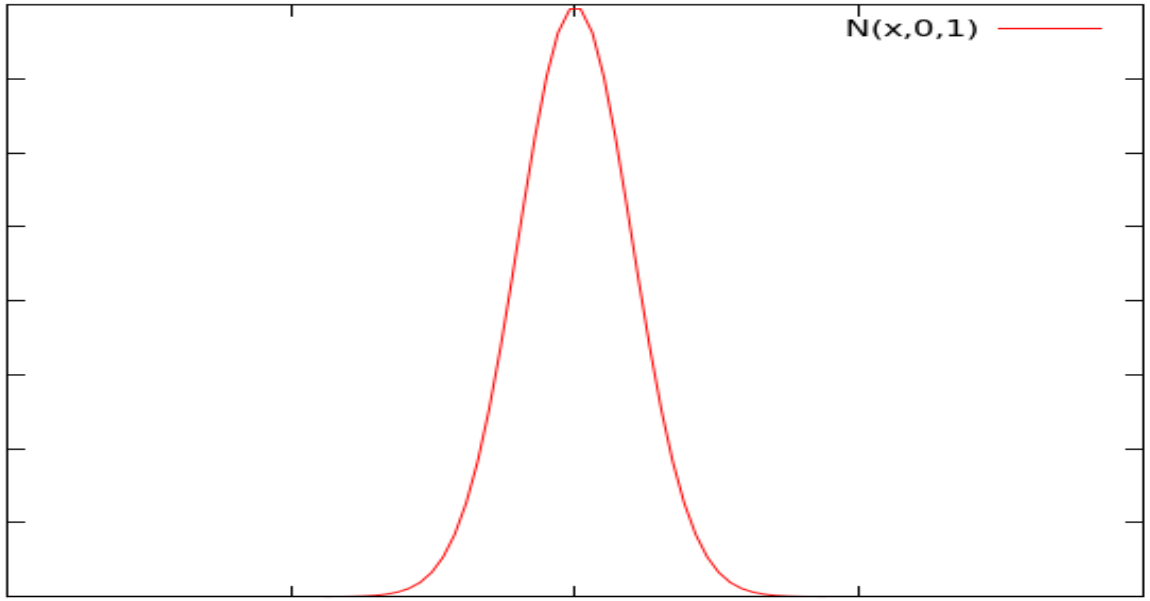
- $p(y|x) = \frac{1}{1+e^{w^T x}}$
- $\mathcal{L} = -\frac{1}{N}\sum_i y_i (w^T x_i) - \log\left(1 + e^{w^T x_i}\right)$ derived both from ML and entropy approaches.
- No closed form, numerical minimization needed (SGD).
- Extension to multinomial.

Towards a deeper understanding of linear regression

- In logistic regression we have a expression for the probability of an event, we want the same in for the linear regression.
- $p(y|x_i) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{1}{2}\left(\frac{y-\hat{y}}{\sigma}\right)^2} = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{1}{2}\left(\frac{y-w^T x_i}{\sigma}\right)^2}$

Towards a deeper understanding of linear regression

- In logistic regression we have a expression for the probability of an event, we want the same in for the linear regression.
- $p(y|x_i) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{1}{2}\left(\frac{y-\hat{y}}{\sigma}\right)^2} = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{1}{2}\left(\frac{y-w^T x_i}{\sigma}\right)^2}$



Towards a deeper understanding of linear regression

- Maximum Likelihood approach

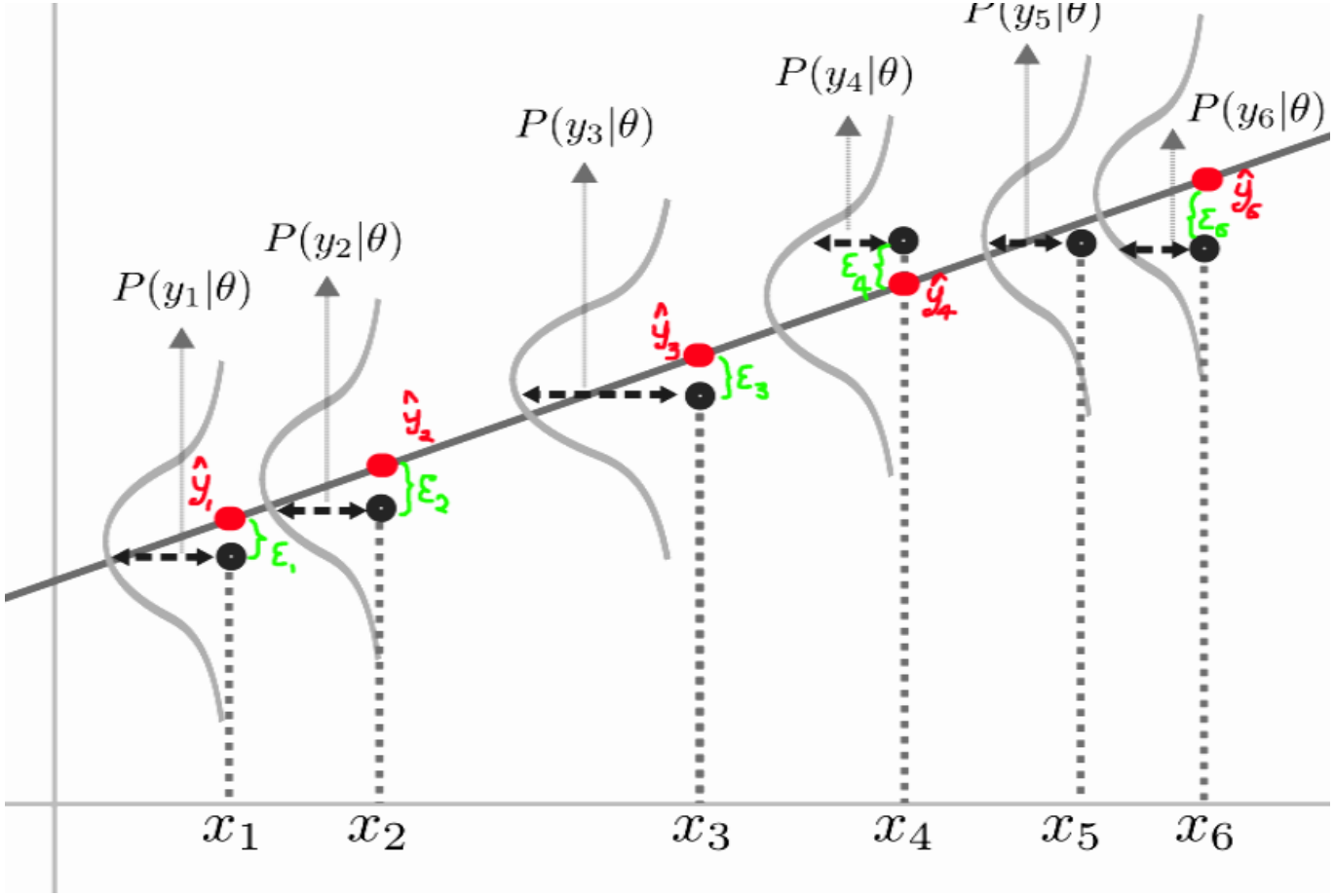
$$\mathcal{L} = \prod_i p(y|x_i) = \prod_i \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{1}{2}\left(\frac{y-w^T x_i}{\sigma}\right)^2}$$

$$\log \mathcal{L} = \sum_i \log\left(\frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{1}{2}\left(\frac{y-w^T x_i}{\sigma}\right)^2}\right) =$$

$$\sum_i \log\left(\frac{1}{\sqrt{2\pi\sigma^2}}\right) - \frac{1}{2}\left(\frac{y - w^T x_i}{\sigma}\right)^2 =$$

$$-\frac{N}{2}\log(2\pi\sigma^2) - \frac{1}{2\sigma^2}\sum_i (y - w^T x_i)^2$$

Towards a deeper understanding of linear regression



Testing significance of the weights

- Unbiased estimator of $\sigma^2 = \frac{1}{N-d-1}\sum_i (y - \hat{y})^2$
- Variance associated with our estimated weights:

$$Var(w) = (XX^T)^{-1}\sigma^2$$

- Then we can:
  - Estimate confidence intervals for w
  - Testing against the null hypothesis $w = 0$
- **Remember, this analysis is only 100% valid if the underlying hypothesis are true (the model is correct, the noise is gaussian distributed and independent of x)**

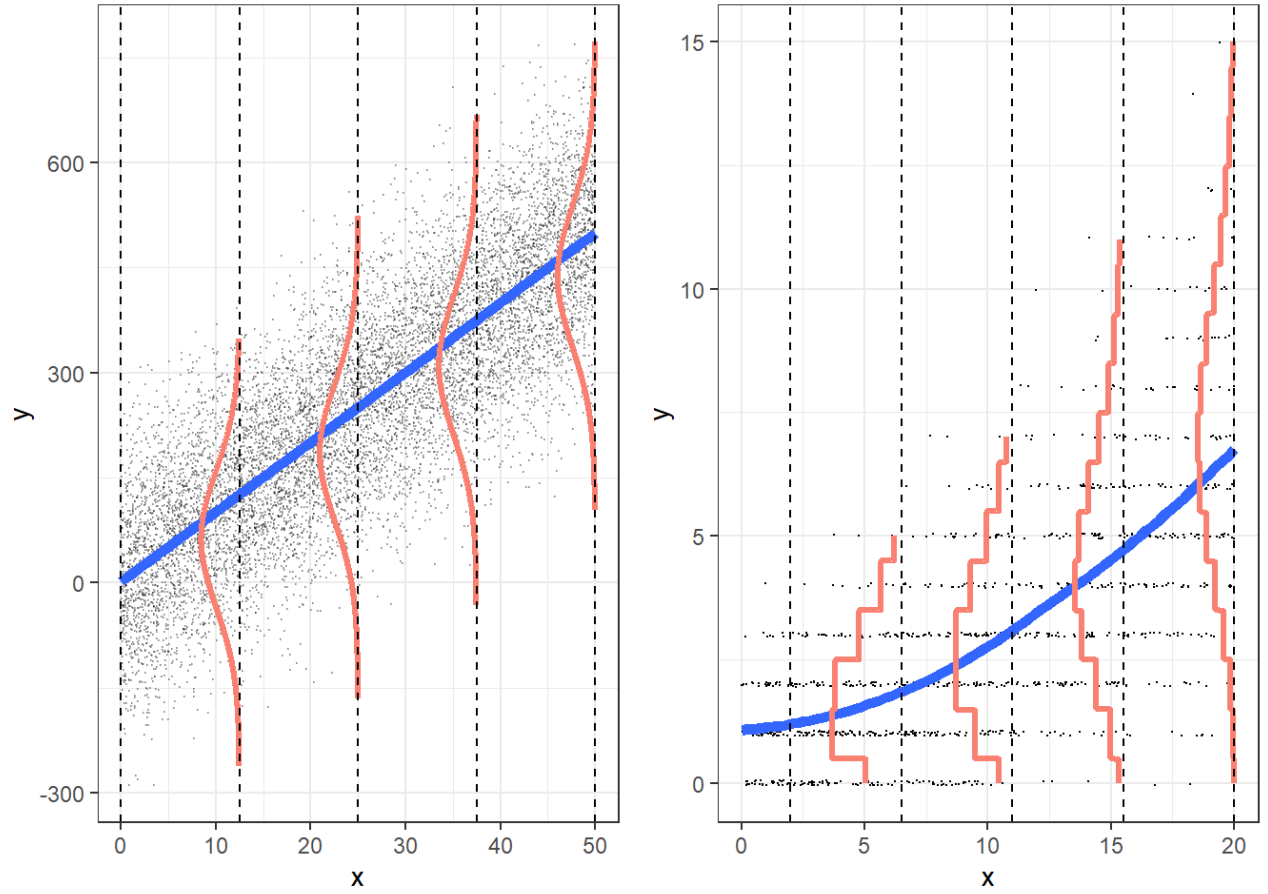
A case in which this model is not correct

- The dependent variable is a count variable taking small values (less than 100).
- The dependent variable follows a distribution whose parameters are determined by exogenous variables.
- Justified when the variable considered describes the number of occurrences of an event in a give time span eg. # of job-related accidents=f(factory charact.), ship damage=f(type, yr.con., pd.op.)

Poisson regression: The model

- $p(y|x) = \frac{e^{-\lambda}\lambda^{y}}{y!}$ This is the Poisson distribution
- $\mathbb{E}(y) = \lambda$ (Easy to show)
- Where is the dependence on x?
- $\log \lambda_i = w^T x_i$

Poisson regression vs linear regression



Poisson regression: Maximum Likelihood derivation

- $\log \mathcal{L} = \sum_i \log\left(\frac{e^{-\lambda_i}{\lambda_i}^{y_i}}{y_i!}\right) = \sum_i(-\lambda_i + y_i \log \lambda_i - \log y_i!) =$
  $\sum_i \left(-e^{w^T x_i} + y_i w^T x_i - \log y_i!\right)$
- $\frac{\partial \log \mathcal{L}}{\partial w} = \sum_i \left(-x_i e^{w^T x_i} + y_i x_i\right) = \sum_i \left(x_i \left(y_i - e^{w^T x_i}\right)\right)$

Gradient descent of the Poisson regression

$$w_{t+1} = w_t - \gamma \frac{\partial \mathcal{L}}{\partial w} = w_t - \gamma \left(\sum_i \left(x_i \left(e^{w^T x_i} - y_i \right) \right) \right)$$

Interpretation of the coefficients

- Using $\log \lambda_i = w^T x_i$ and $\mathbb{E}(y) = \lambda$ we can arrive to:

$$w = \frac{\partial \log \mathbb{E}(y)}{\partial \log x_i}$$

It means that w can be interpreted as an elasticity (a measure of the sensitivity of the dependent variable y to changes in the variables x)

Generalized linear models

Generalized linear models and link functions

$$p(y|x) = f(w^T x)$$

- We want to match the domain of the link function to the range of the distribution function's mean.
- We need a new definition of the loss

The one-parameter exponential family

- The probability density function should be in the form:

$$p(y; w) \propto e^{[a(y)b(w)+c(w)+d(y)]}$$

- And the domain of $p(y; w)$ should be independent of w
- $b(w)$is also known as the *canonical link*. It can be shown that, if we set $b(w) = w^T x_i$ then

$$\mathbb{E}(y) = -\frac{c'(w)}{b'(w)} \,\&\, Var(y) = \frac{b''(w)c'(w) - b'(w)c''(w)}{[b'(w)]^3}$$

Is the case of Poisson regression?

- $p(y) = \frac{e^{-\lambda}\lambda^{y}}{y!}$
- $\log p(y) = -\lambda + y\log\lambda - \log y!$ $\quad p(y;w) = e^{[a(y)b(w)+c(w)+d(y)]}$
- $a(y) = y$
- $b(w) = \log\lambda$
- $c(w) = -\lambda$
- $d(y) = -\log y!$

$$\mathbb{E}(y) = -\frac{c'(w)}{b'(w)} = -\frac{-1}{1/\lambda} = \lambda$$

$$\mathrm{Var}(y) = \frac{b''(w)c'(w) - b'(w)c''(w)}{[b'(w)]^3} = \frac{\left(-1/\lambda^2\right)(-1) - \left(1/\lambda\right)(0)}{\left[1/\lambda\right]^3} = \lambda$$

Poisson regression

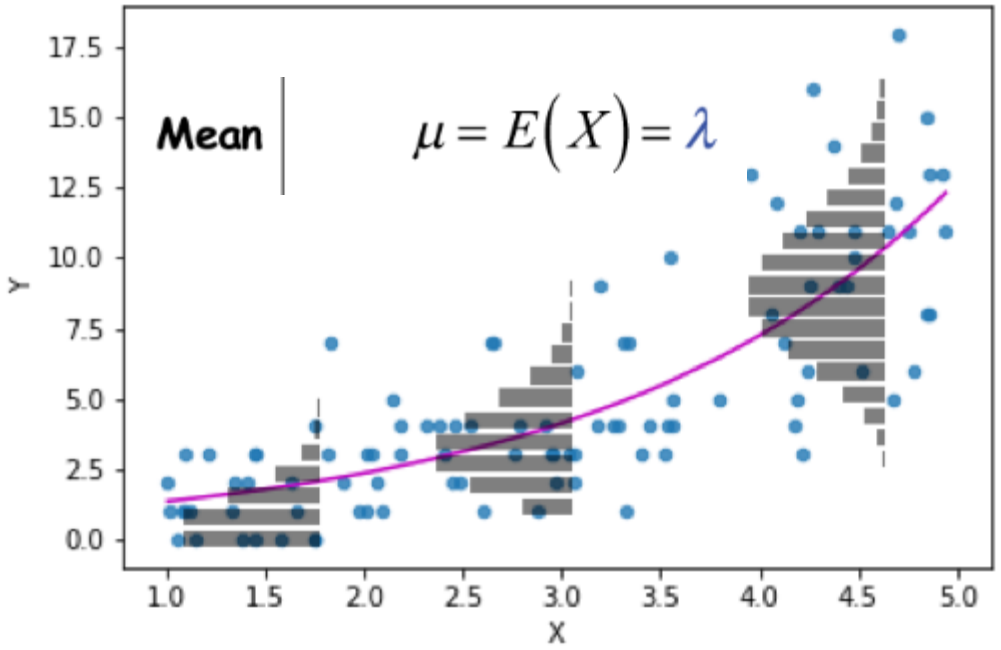
Link function Linear predictor

$$\ln \lambda_i = b_0 + b_1 x_i$$

$$y_i \sim \text{Poisson}(\lambda_i)$$

Probability distribution

Poisson regression



Poisson regression illustrated

What's the range of the mean?

Is the case of Gaussian probability?

- $p(y) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{1}{2}\left(\frac{y-\hat{y}}{\sigma}\right)^2}$; assuming $\sigma = 1 \rightarrow p(y) = \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}(y-\hat{y})^2}$
- $\log p(y) = -\frac{1}{2}(y-\hat{y})^2$
- $\log p(y) = -\frac{1}{2}(y^2 - 2y\hat{y} + \hat{y}^2)$
- $\mathrm{a}(y) = y$
- $b(w) = \hat{y}$
- $c(w) = -\frac{1}{2}\hat{y}^2$
- $d(y) = -\frac{1}{2}y^2$

$$p(y;w) = e^{[a(y)b(w)+c(w)+d(y)]}$$

$$\mathbb{E}(y) = -\frac{c'(w)}{b'(w)} = -\frac{-\hat{y}}{1} = \hat{y}$$

$$\mathrm{Var}(y) = \frac{b''(w)c'(w) - b'(w)c''(w)}{[b'(w)]^3} = \frac{(0)(-\hat{y}) - (1)(-1)}{[1]^3} = 1$$?!

Study of the binomial distribution

$$p(y;w) = e^{[a(y)b(w)+c(w)+d(y)]}$$

- $p(Y = y) = \binom{n}{y} p^y (1-p)^{(n-y)}$

  n trials; p probability of success
- $\log p(y) = y \log p + (n-y)\log(1-p) + \log \binom{n}{y}$
- $\log p(y) = y \log \frac{p}{1-p} + n \log(1-p) + \log \binom{n}{y}$
- $a(y) = y$
- $b(w) = \log \frac{p}{1-p}$
- $c(w) = n \log(1-p)$
- $d(y) = \log \binom{n}{y}$

Observe that the link function is the logit equation for the logistic regression

The Logistic regression is a GLM with the binomial distribution

A generalized Loss for GLMs

- Use the sum of *deviances* as Loss.
- The deviance is a way of measuring the lack of agreement between the model prediction and the ground truth:
- $\mathcal{D}(y, \hat{y}) = 2\big(\log\big(p(y|w_s)\big) - \log\big(p(y|w_o)\big)\big)$
- w_o are the parameters optimized in our model.
- w_s are the parameters optimized in a theoretical model that perfectly fits the data (*saturated* model).

A generalized Loss for GLMs

- $\mathcal{D}(y, \hat{y}) = 2\big(\log\big(p(y|w_s)\big) - \log\big(p(y|w_o)\big)\big)$
- $\mathcal{L} = \sum_i \mathcal{D}(y_i, \hat{y}_i) = 2\sum_i\big(\log\big(p(y_i|w_s)\big) - \log\big(p(y_i|w_o)\big)\big)$
- $p(y_i|w_s) \rightarrow \hat{y}_i = y_i$
- Gaussian(assuming $\sigma = 1 \rightarrow p(y) = \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}(y-\hat{y})^2}$) :

$$\mathcal{D}(y, \hat{y}) = 2\left(\log\left(\frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}(\hat{y}-\hat{y})^2}\right) - \log\left(\frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}(y-\hat{y})^2}\right)\right) = (y-\hat{y})^2$$

- Poisson($p(y) = \frac{e^{-\lambda}\lambda^y}{y!}$) (Do it as exercise)

Other GLM

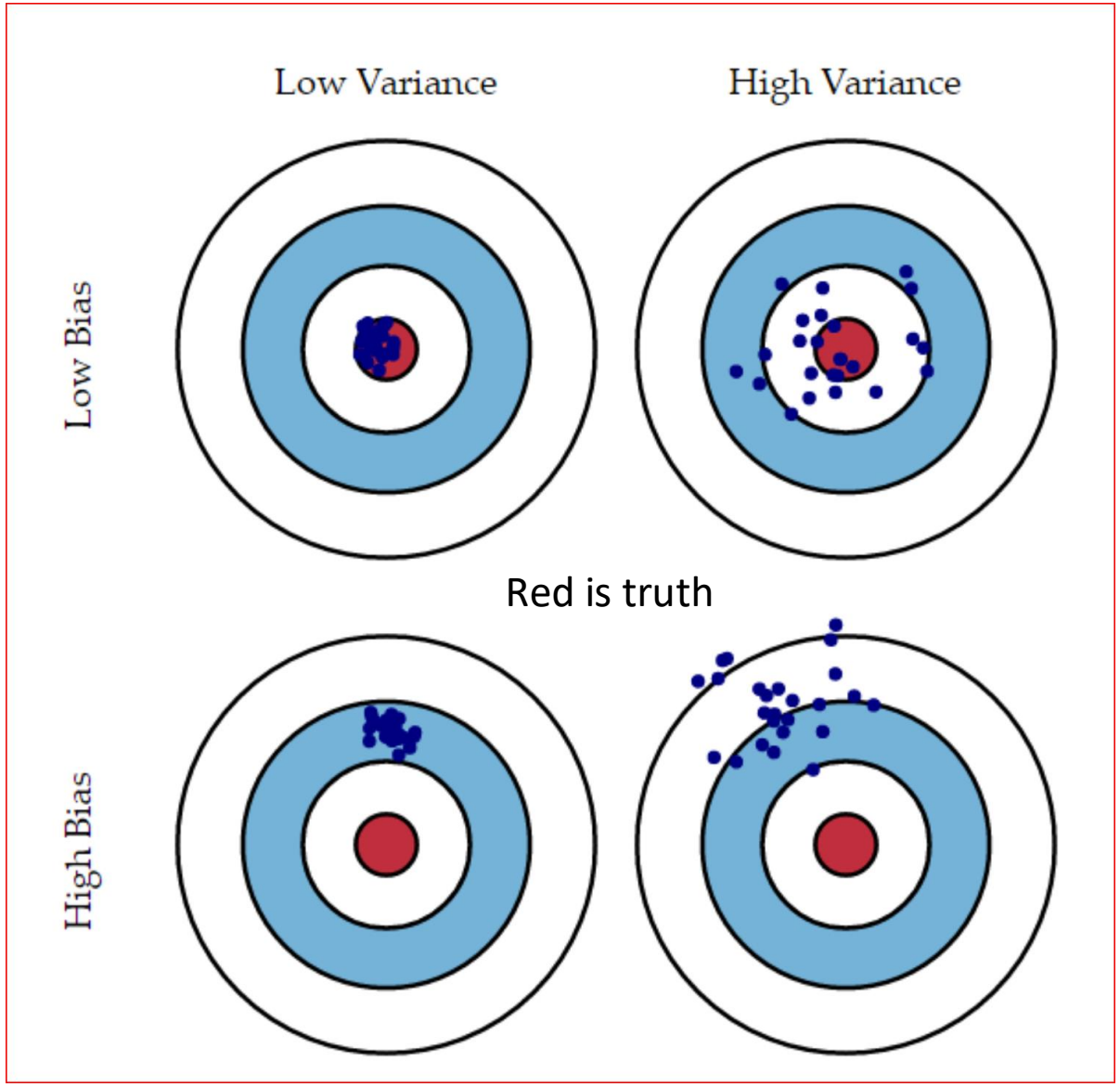
Generalized linear model. (2022, October 4). In Wikipedia. https://en.wikipedia.org/wiki/Generalized_linear_model

Common distributions with typical uses and canonical link functions

Distribution	Support of distribution	Typical uses	Link name	Link function, $\mathbf{X}\boldsymbol{\beta} = g(\mu)$	Mean function
Normal	real: $(-\infty, +\infty)$	Linear-response data	Identity	$\mathbf{X}\boldsymbol{\beta} = \mu$	$\mu = \mathbf{X}\boldsymbol{\beta}$
Exponential Gamma	real: $(0, +\infty)$	Exponential-response data, scale parameters	Negative inverse	$\mathbf{X}\boldsymbol{\beta} = -\mu^{-1}$	$\mu = -(\mathbf{X}\boldsymbol{\beta})^{-1}$
Inverse Gaussian	real: $(0, +\infty)$		Inverse squared	$\mathbf{X}\boldsymbol{\beta} = \mu^{-2}$	$\mu = (\mathbf{X}\boldsymbol{\beta})^{-1/2}$
Poisson	integer: $0, 1, 2, \ldots$	count of occurrences in fixed amount of time/space	Log	$\mathbf{X}\boldsymbol{\beta} = \ln(\mu)$	$\mu = \exp(\mathbf{X}\boldsymbol{\beta})$
Bernoulli	integer: $\{0, 1\}$	outcome of single yes/no occurrence	Logit	$\mathbf{X}\boldsymbol{\beta} = \ln\left(\frac{\mu}{1-\mu}\right)$	$\mu = \frac{\exp(\mathbf{X}\boldsymbol{\beta})}{1+\exp(\mathbf{X}\boldsymbol{\beta})} = \frac{1}{1+\exp(-\mathbf{X}\boldsymbol{\beta})}$
Binomial	integer: $0, 1, \ldots, N$	count of # of "yes" occurrences out of N yes/no occurrences		$\mathbf{X}\boldsymbol{\beta} = \ln\left(\frac{\mu}{n-\mu}\right)$	
Categorical	integer: $[0, K)$ K-vector of integer: $[0, 1]$, where exactly one element in the vector has the value 1	outcome of single K-way occurrence		$\mathbf{X}\boldsymbol{\beta} = \ln\left(\frac{\mu}{1-\mu}\right)$	
Multinomial	K-vector of integer: $[0, N]$	count of occurrences of different types (1 .. K) out of N total K-way occurrences			

Bias/Variance: derivation

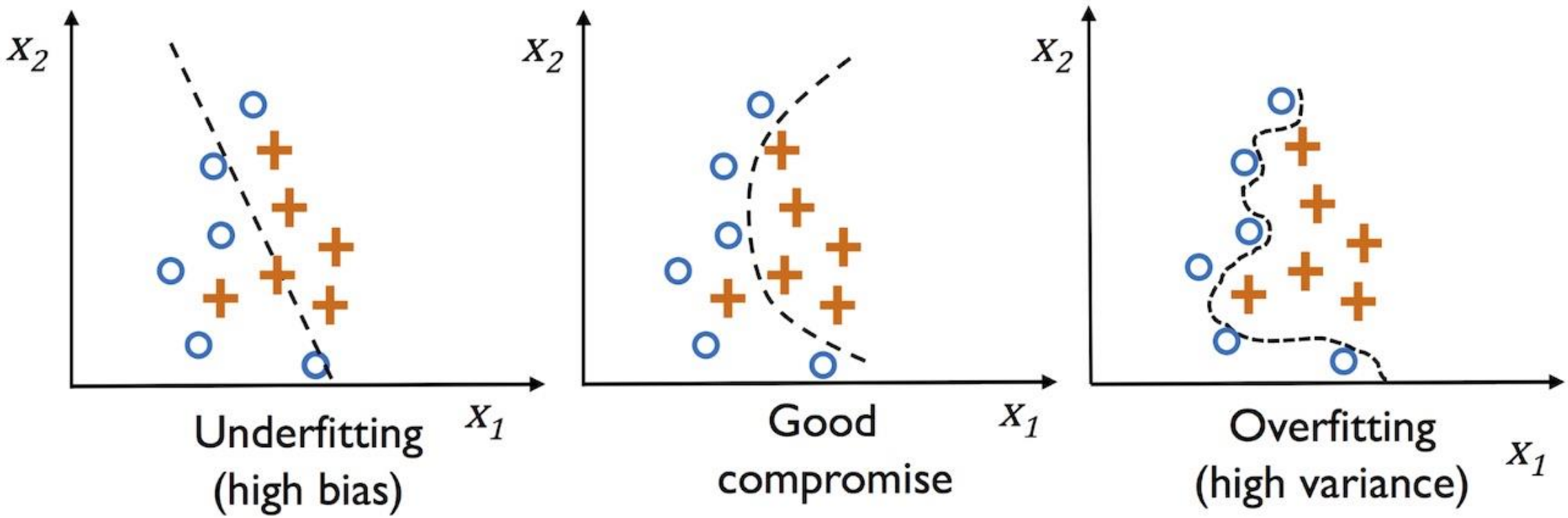
Bias and variance

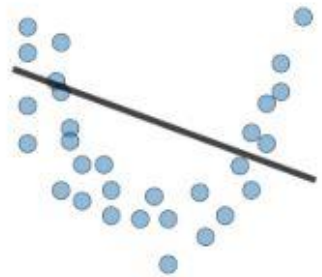
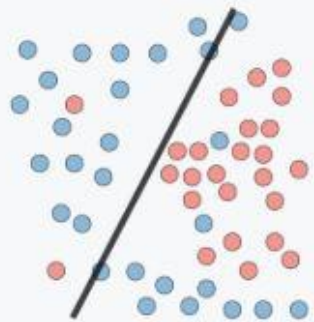
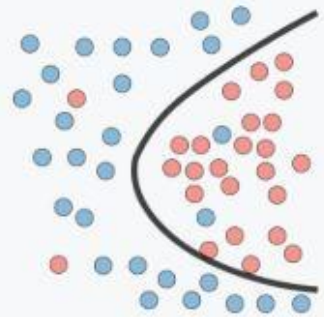
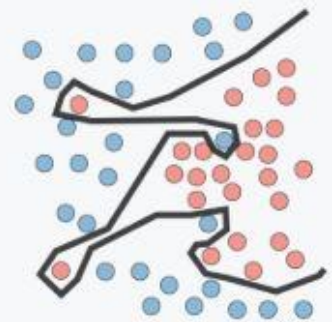
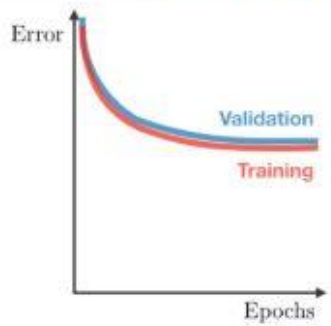
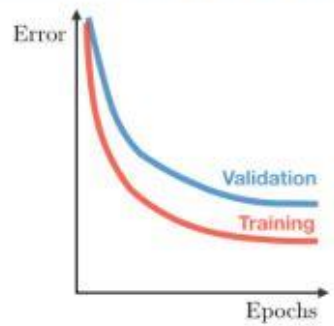
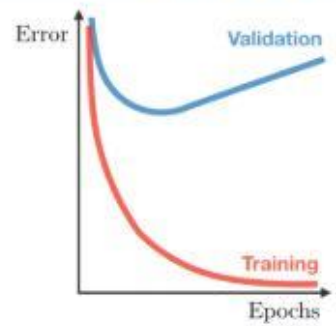


- The *bias* error is an error from erroneous assumptions in the learning algorithm. High bias can cause an algorithm to miss the relevant relations between features and target outputs (underfitting).

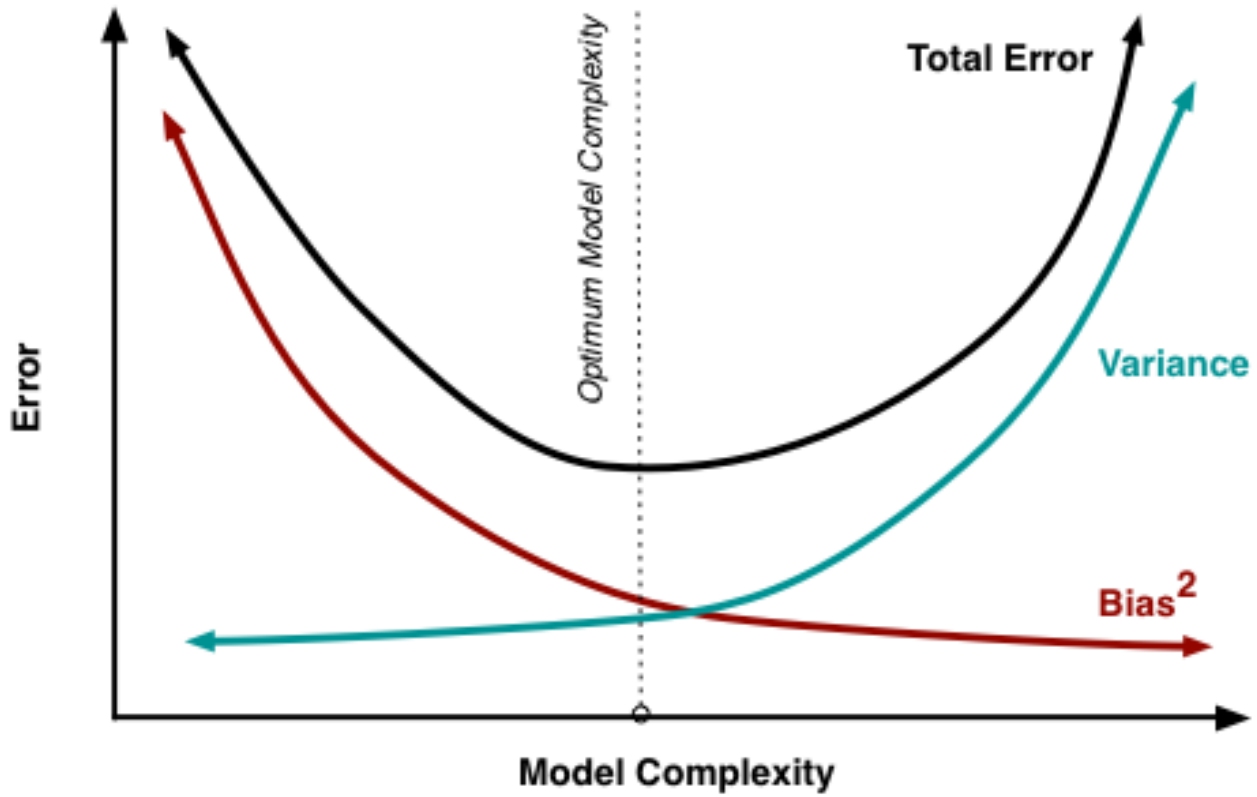
- The *variance* is an error from sensitivity to small fluctuations in the training set. High variance may result from an algorithm modeling the random noise in the training data (overfitting).

Bias and variance tradeoff

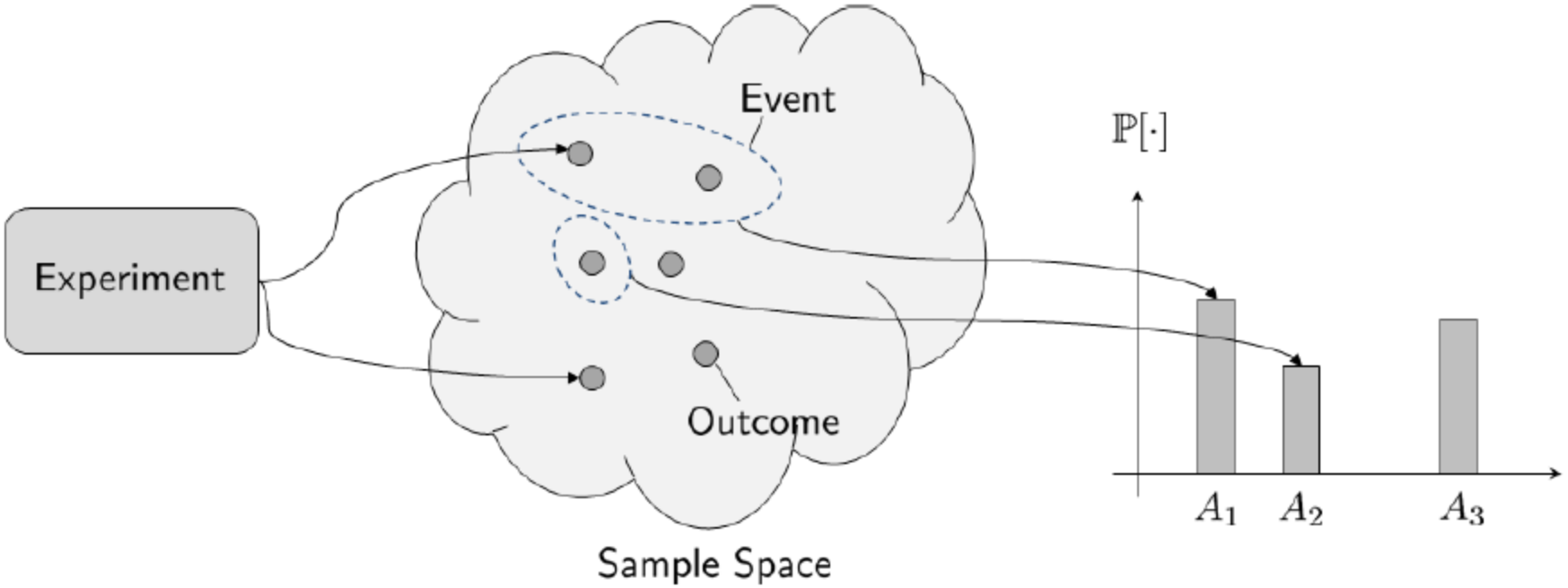


	Underfitting	Just right	Overfitting
Symptoms	• High training error • Training error close to test error • High bias	• Training error slightly lower than test error	• Very low training error • Training error much lower than test error • High variance
Regression illustration			
Classification illustration			
Deep learning illustration			
Possible remedies	• Complexify model • Add more features • Train longer		• Perform regularization • Get more data

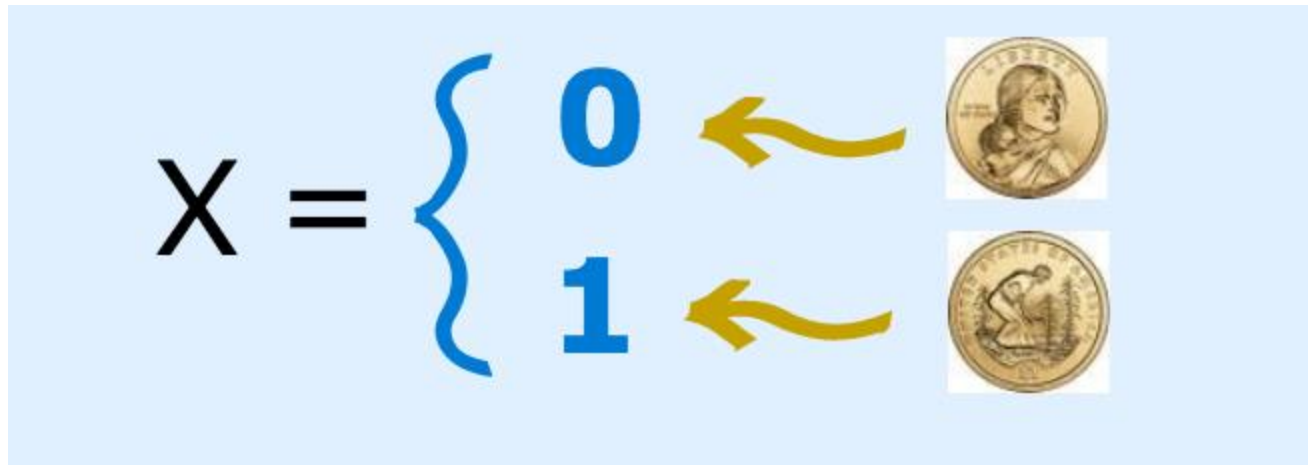
Bias and variance tradeoff



Random variables: recap



Random variable: A random variable is a rule/function that assigns a numerical value to each outcome in a sample space.


X =
0
1

Expectation of a random variable

The **expectation** of X is defined as

$$\mathbb{E}[X] = \int x \, p_X(x) dx$$

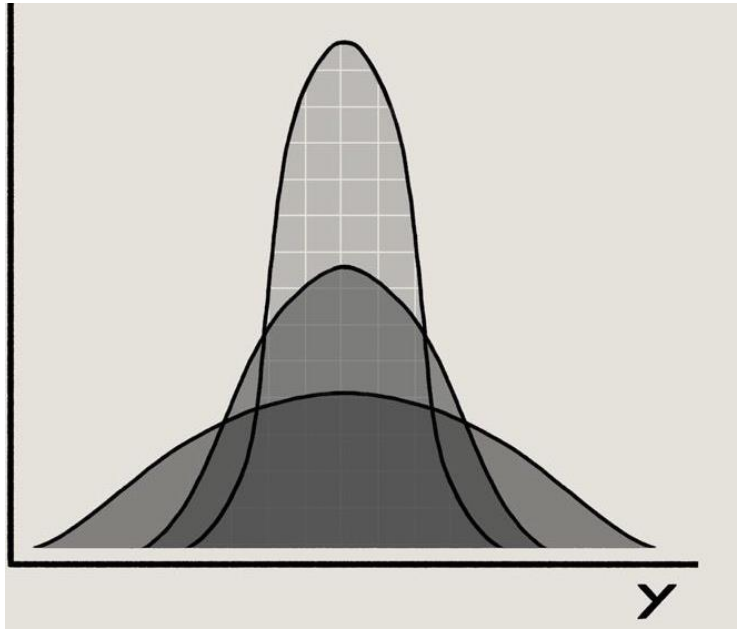
For any function g, the expectation $\mathbb{E}[g(X)]$ is defined as

$$\mathbb{E}[g(X)] = \int g(x) \, p_X(x) dx.$$

For example, if $g(x) = x^2$ then $\mathbb{E}[g(X)] = \mathbb{E}[X^2]$ is the **second moment** of X. The **variance** of X is defined as $\mathrm{Var}[X] = \mathbb{E}[(X - \mathbb{E}[X])^2] = \mathbb{E}[X^2] - \mathbb{E}[X]^2$.

Intro: variance of random variable

$$
\begin{aligned}
E[(Z-\bar{Z})^2] &= E[Z^2 - 2Z\bar{Z} + \bar{Z}^2] \\
&= E[Z^2] - 2E[Z]\bar{Z}] + \bar{Z}^2 \\
&= E[Z^2] - 2\bar{Z}\bar{Z} + \bar{Z}^2 \\
&= E[Z^2] - \bar{Z}^2.
\end{aligned}
$$



Bias/Variance: derivation

- We can write the relationship between predictor variables X and the response Y as variables:

$$Y = f(X) + \epsilon \qquad \epsilon \sim \mathcal{N}(0, \sigma_\epsilon)$$

- Then the expected value of the quadratic error can be written as :

$$\mathrm{SE}(x) = E\left[(Y - \hat{f}(x))^2\right]$$

- After some transformations you get:

$$\mathrm{SE}(x) = \underbrace{\left(E[\hat{f}(x)] - f(x)\right)^2}_{\text{Bias}^2} + \underbrace{E\left[\left(\hat{f}(x) - E[\hat{f}(x)]\right)^2\right]}_{\text{Variance}} + \underbrace{\sigma_e^2}_{\substack{\text{irreducibile} \\ \text{error}}}$$

Bias/Variance: derivation

- Expected value of the square of a random variable X:

$$\mathrm{E}[X^2] = \mathrm{Var}[X] + \mathrm{E}[X]^2$$

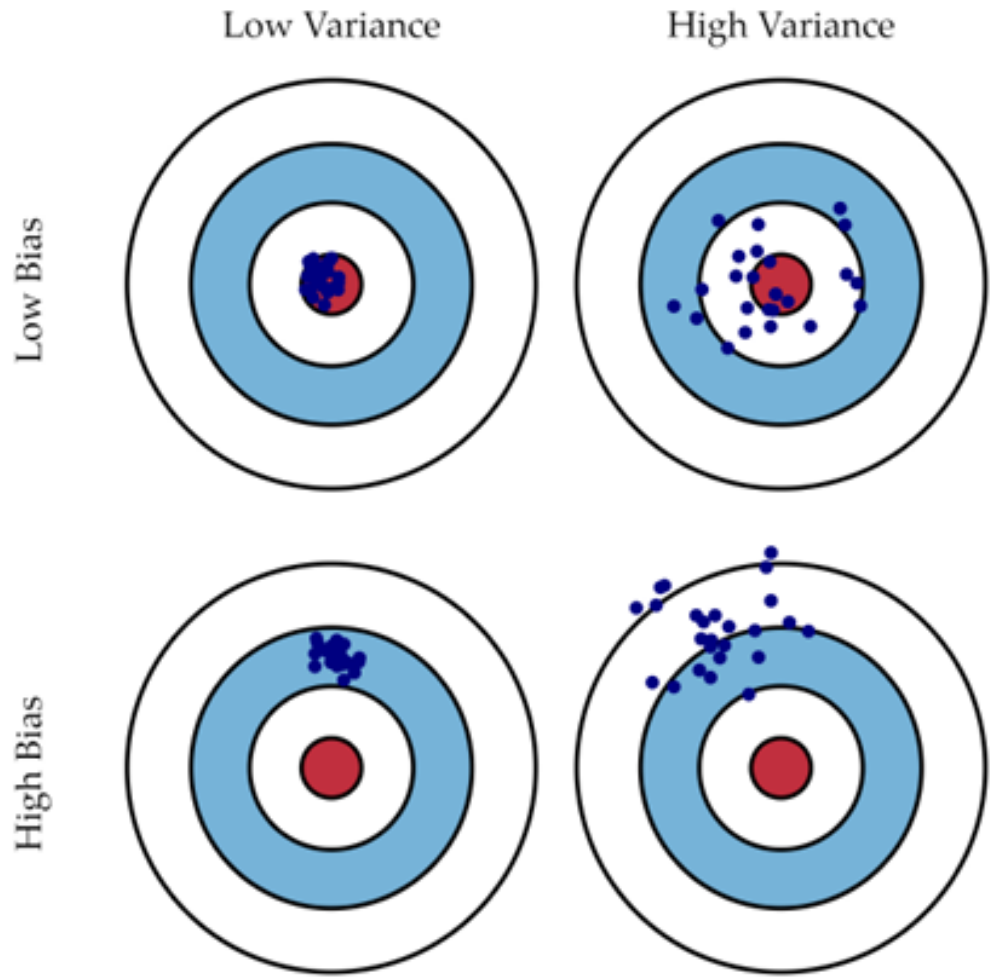
$$\mathrm{E}[y] = \mathrm{E}[f + \epsilon] = \mathrm{E}[f] = f$$

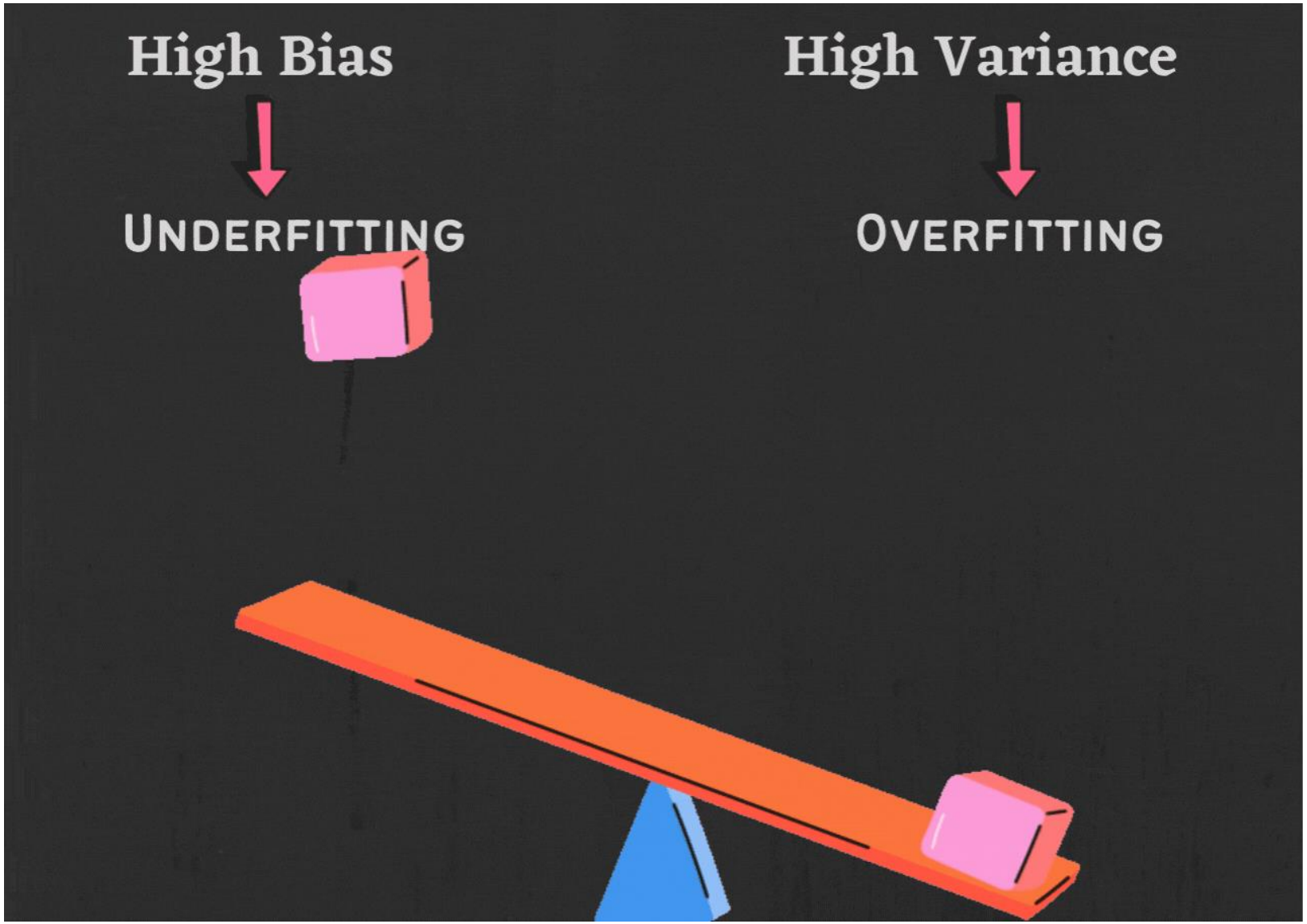
- With this:

$$\begin{aligned}
\mathrm{E}\left[(y - \hat{f})^2\right] &= \mathrm{E}[y^2 + \hat{f}^2 - 2y\hat{f}] = \mathrm{E}[y^2] + \mathrm{E}[\hat{f}^2] - \mathrm{E}[2y\hat{f}] \\
&= \mathrm{Var}[y] + \mathrm{E}[y]^2 + \mathrm{Var}[\hat{f}] + \mathrm{E}[\hat{f}]^2 - 2f\mathrm{E}[\hat{f}] \\
&= \mathrm{Var}[y] + \mathrm{Var}[\hat{f}] + (f - \mathrm{E}[\hat{f}])^2 \\
&= \mathrm{Var}[y] + \mathrm{Var}[\hat{f}] + \mathrm{E}[f - \hat{f}]^2 \\
&= \sigma^2 + \mathrm{Var}[\hat{f}] + \mathrm{Bias}[\hat{f}]^2.
\end{aligned}$$

$$\mathrm{Bias}\big[\hat{f}(x)\big] = \mathrm{E}\big[\hat{f}(x) - f(x)\big]$$

$$\mathrm{Var}\big[\hat{f}(x)\big] = \mathrm{E}\Big[\big(\hat{f}(x) - \mathrm{E}[\hat{f}(x)]\big)^2\Big]$$

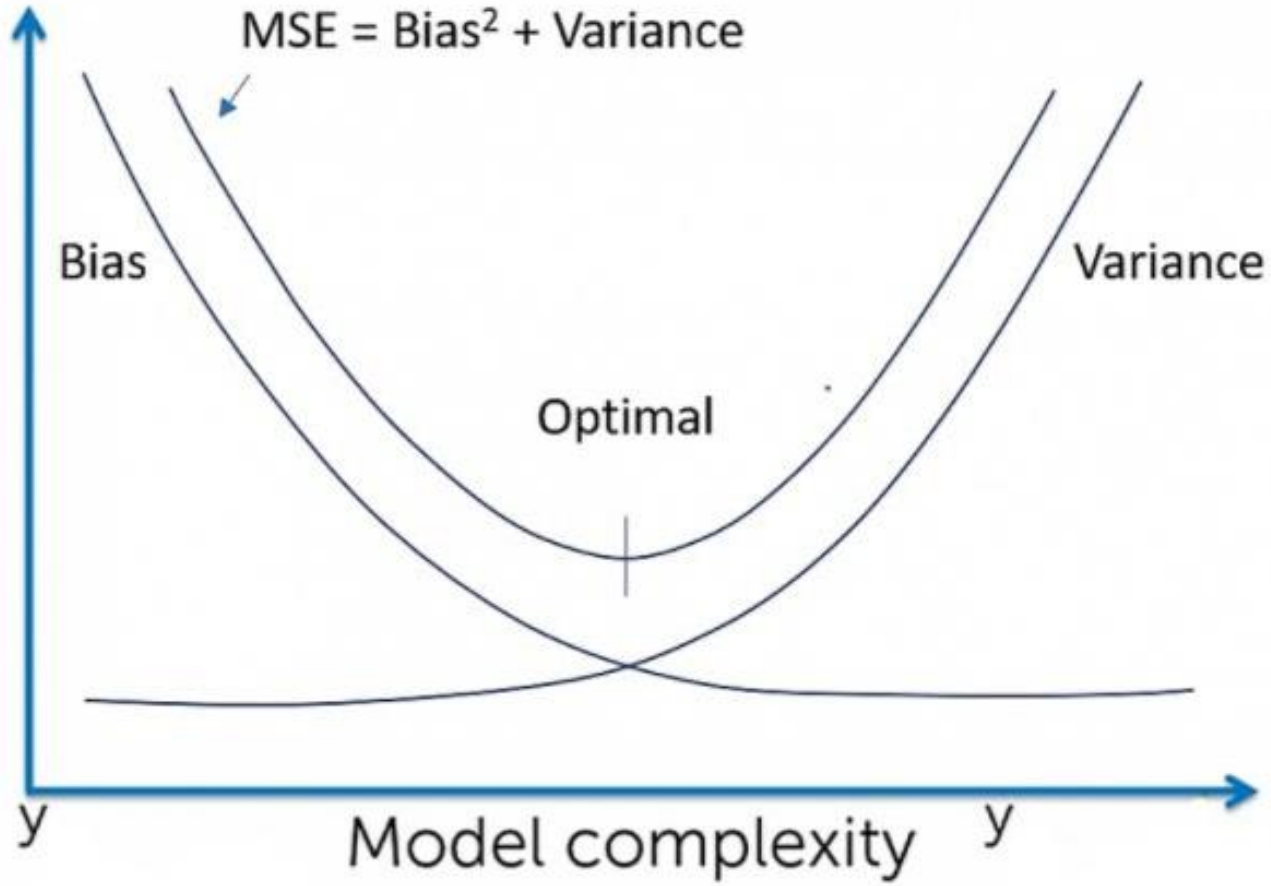


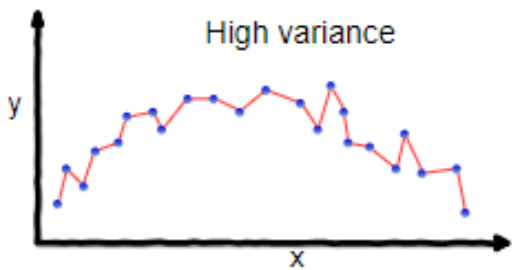

High Bias
High Variance
UNDERFITTING
OVERFITTING

Bias/Variance: again

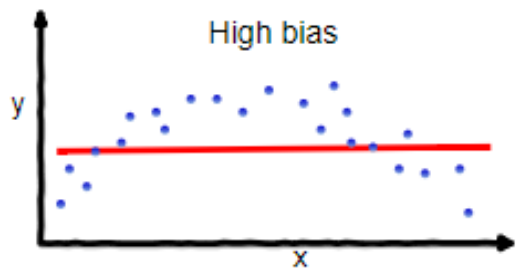
- **Bias** is the difference between the average prediction of our model and the correct value which we are trying to predict. Model with high bias pays very little attention to the training data and oversimplifies the model. It always leads to high error on training and test data.

- **Variance** is the variability of model prediction for a given data point or a value which tells us spread of our data. Model with high variance pays a lot of attention to training data and does not generalize on the data which it hasn't seen before. As a result, such models perform very well on training data but has high error rates on test data.

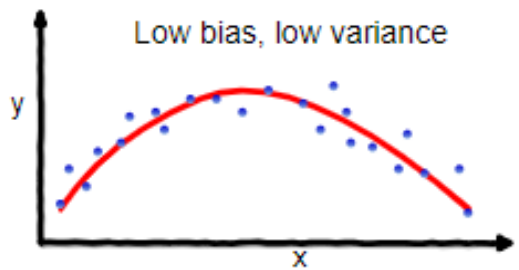




overfitting

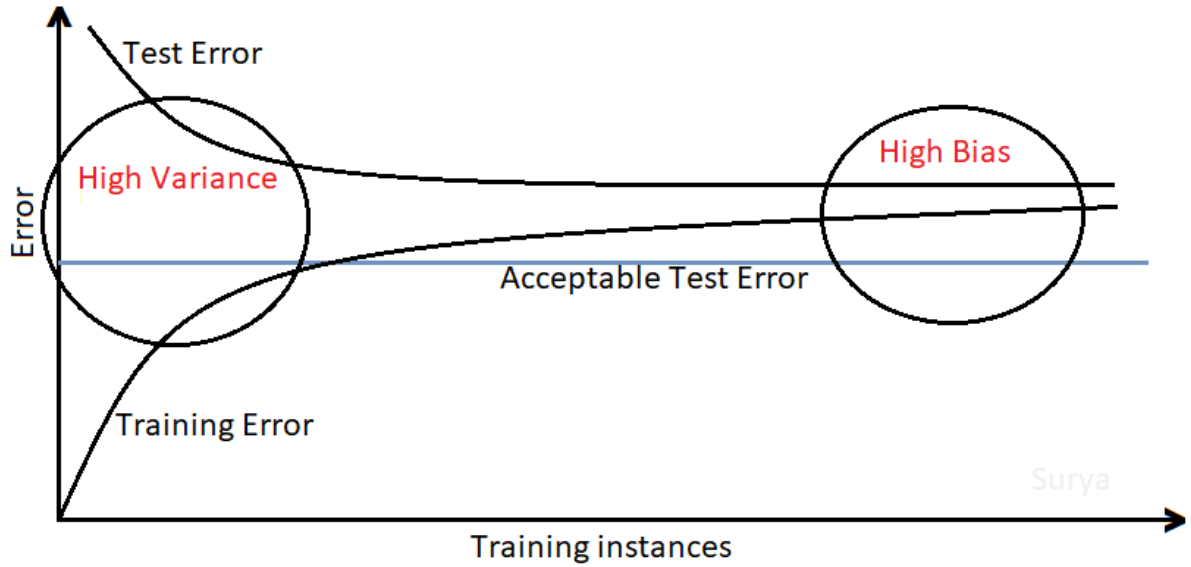


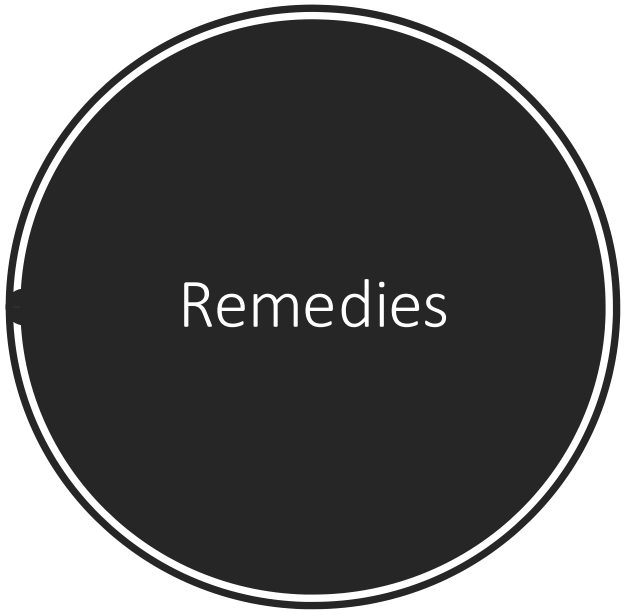
underfitting



Good balance

How do we recognize if our model has a bias or variance problem?





Regime 1 (High Variance)

In the first regime, the cause of the poor performance is high variance.

Symptoms:

1. Training error is much lower than test error
2. Training error is lower than ϵ
3. Test error is above ϵ

Remedies:

- Add more training data
- Reduce model complexity -- complex models are prone to high variance
- Bagging (will be covered later in the course)

Regime 2 (High Bias)

Unlike the first regime, the second regime indicates high bias: the model being used is not robust enough to produce an accurate prediction.

Symptoms:

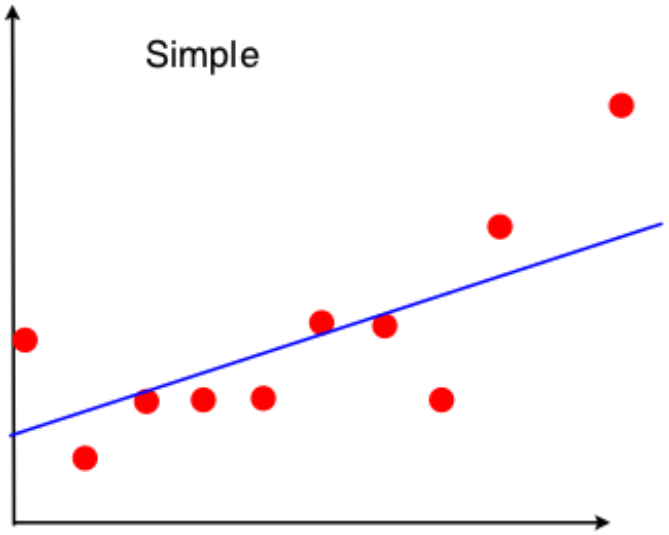
1. Training error is higher than ϵ

Remedies:

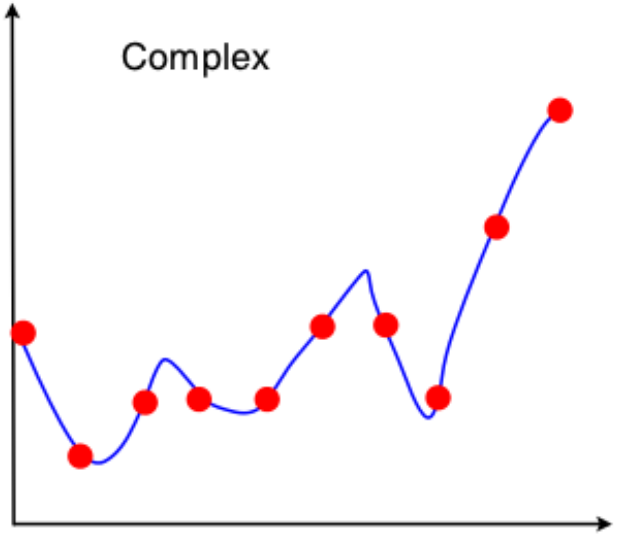
- Use more complex model (e.g. kernelize, use non-linear models)
- Add features
- Boosting (will be covered later in the course)

Regularization : ridge, lasso, elastic nets

Regularization intuition: controlling the network complexity



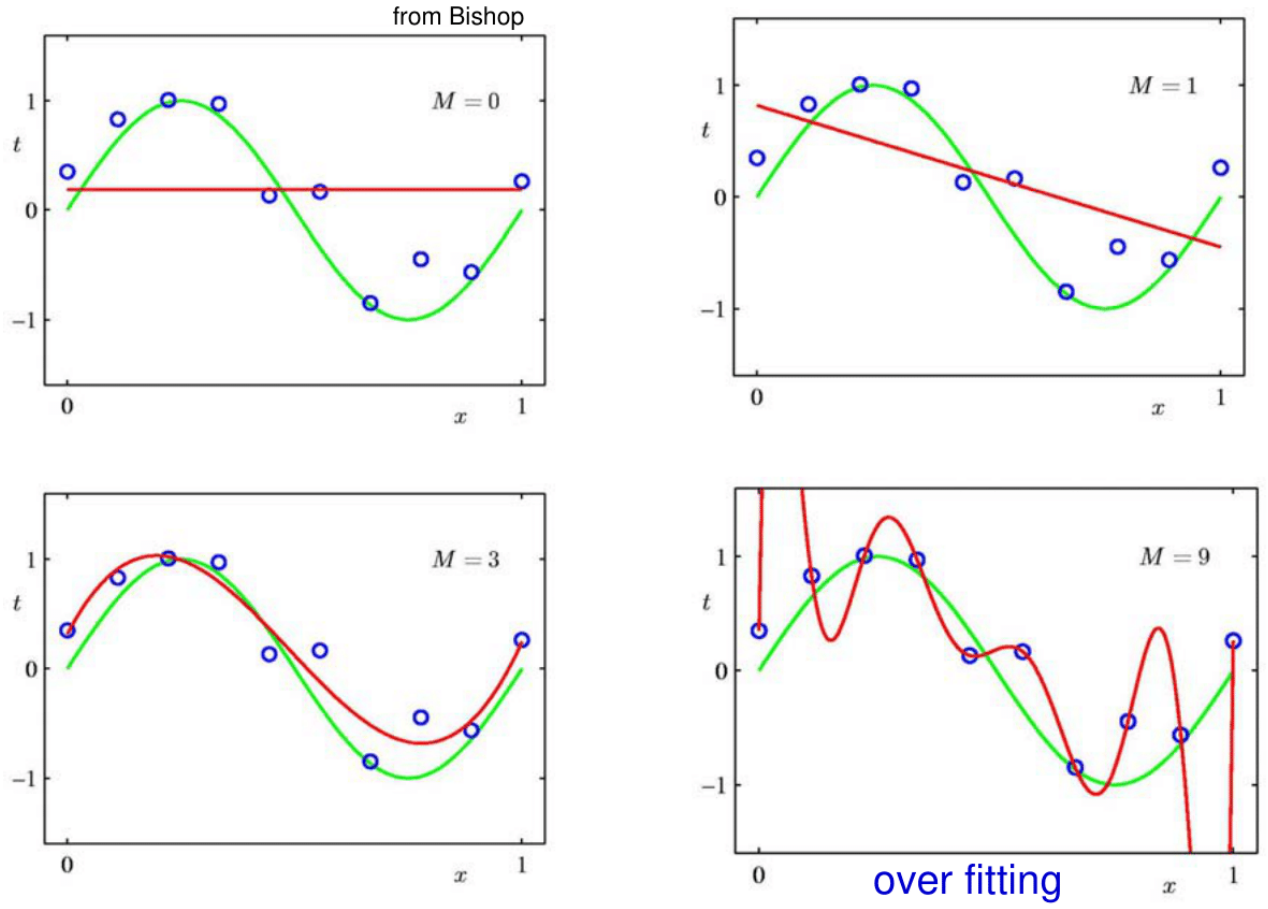
$$y = w_0 + w_1 x$$



$$y = w_0 + w_1 x + w_2 x^2 + w_3 x^3 + \cdots$$

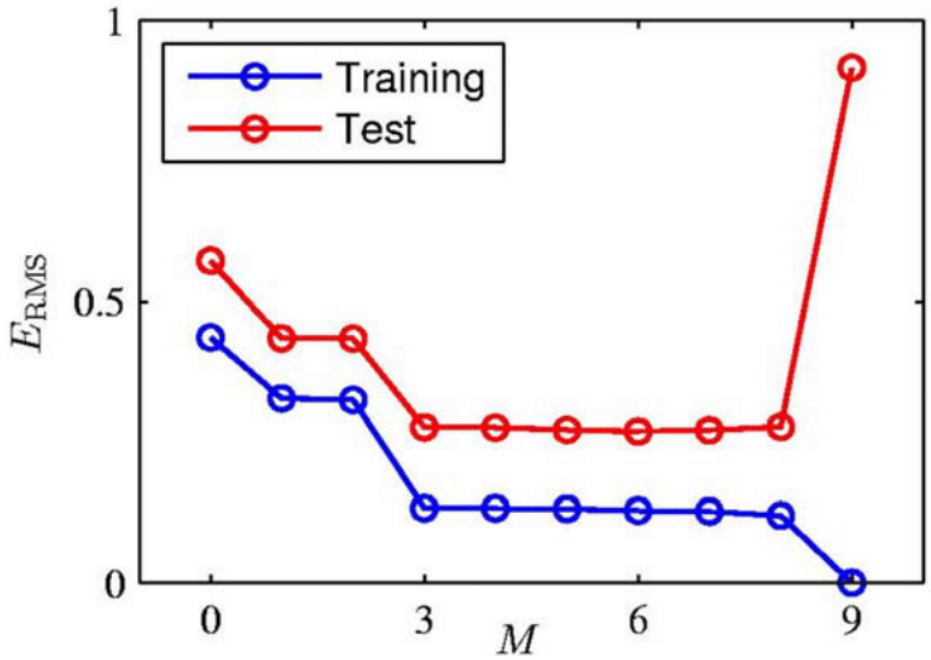
- What's the difference?
- Why should we prefer one model w.r.t. the other?

Tune Regularization for good generalization



Over-fitting

- test data: a different sample from the same true function

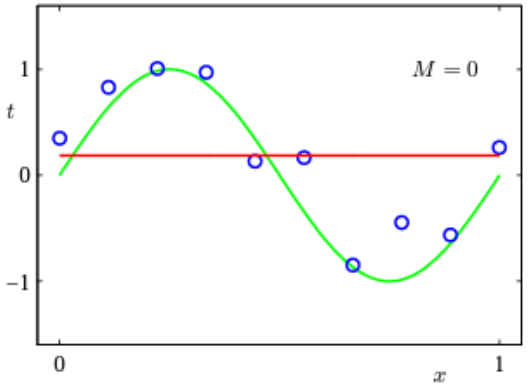


Root-Mean-Square (RMS) Error: $E_{\mathrm{RMS}} = \sqrt{2E(\mathbf{w}^{\star})/N}$

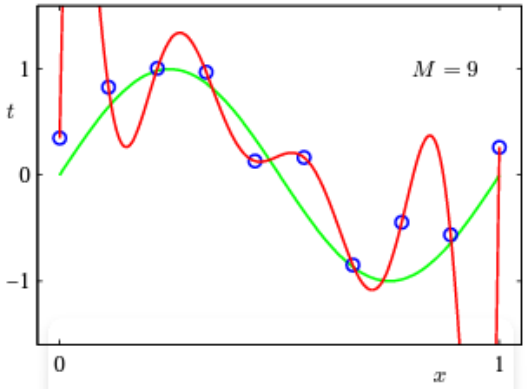
- training error goes to zero, but test error increases with M

Tune Regularization for good generalization

Underfitting : model is too simple — does not fit the data.



Overfitting : model is too complex — fits perfectly, does not generalize.



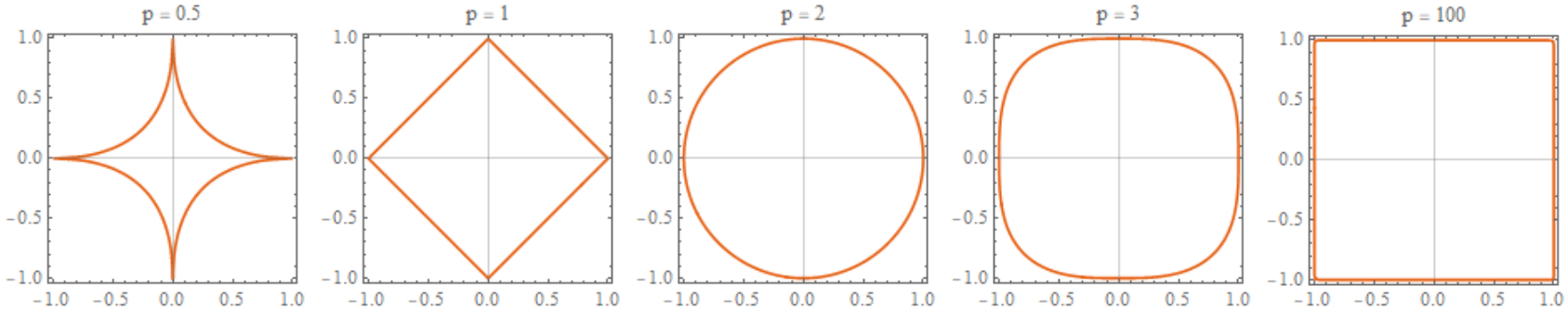
An observation

Polynomial Coefficients

	$M = 0$	$M = 1$	$M = 3$	$M = 9$
$w_0^\star$	0.19	0.82	0.31	0.35
$w_1^\star$		-1.27	7.99	232.37
$w_2^\star$			-25.43	-5321.83
$w_3^\star$			17.37	48568.31
$w_4^\star$				-231639.30
$w_5^\star$				640042.26
$w_6^\star$				-1061800.52
$w_7^\star$				1042400.18
$w_8^\star$				-557682.99
$w_9^\star$				125201.43

How can we control the model complexity?

lp norms



$$x \in \mathbb{R}^d, \quad \|x\|_p = \left(\sum_{i=1}^{d} x_i^p \right)^{1/p}$$

Minimizing the norm of the coefficients we reduce the complexity of the model

Ridge regression (regression +l2 regularization)

$$\arg\min_{w} \|y - X^T w\|_2^2 + \lambda \|w\|_2^2$$

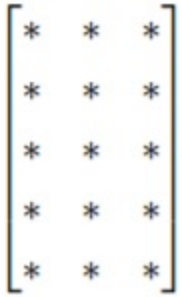
- Reconstruction term $\|y - X^T w\|_2^2$
- Complexity control $\lambda \|w\|_2^2$

Ridge optimization

$$\begin{aligned}\nabla_w[\|y - X^T w\|_2^2 + \lambda\|w\|_2^2] &= \nabla_w[(y - X^T w)^T(y - X^T w) + \lambda w^T w] \\ &= 2\nabla_w[(y - X^T w)](y - X^T w)] + 2\lambda w \\ &= 2X(y - X^T w) + 2\lambda w\end{aligned}$$

$$2X(y - X^T w + 2\lambda w) = 0 \quad \longrightarrow \quad w = (XX^T + \lambda I)^{-1} Xy$$

N<d: not enough data to estimate the parameters



$$X \in \mathbb{R}^{d \times N} \quad XX^T \in \mathbb{R}^{d \times d}$$

XX^T is not invertible!

SOLUTION: regularization

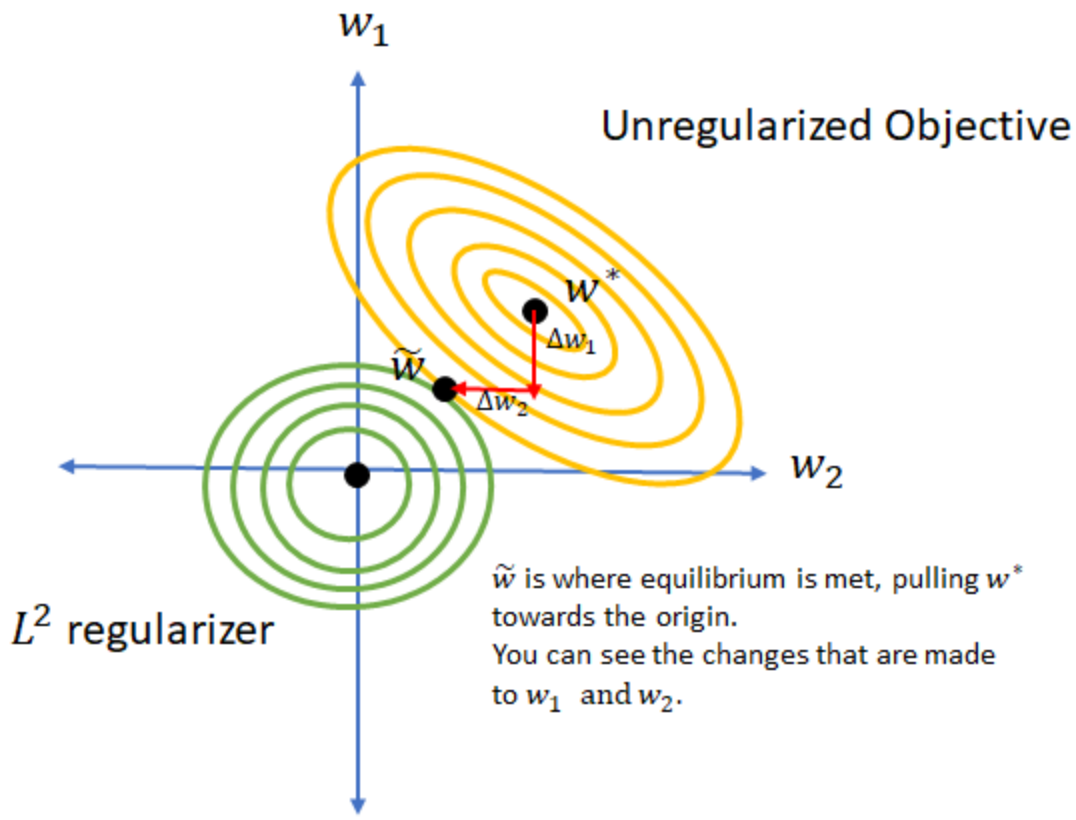
Ridge stability

$$(XX^T + \lambda I)^{-1}$$

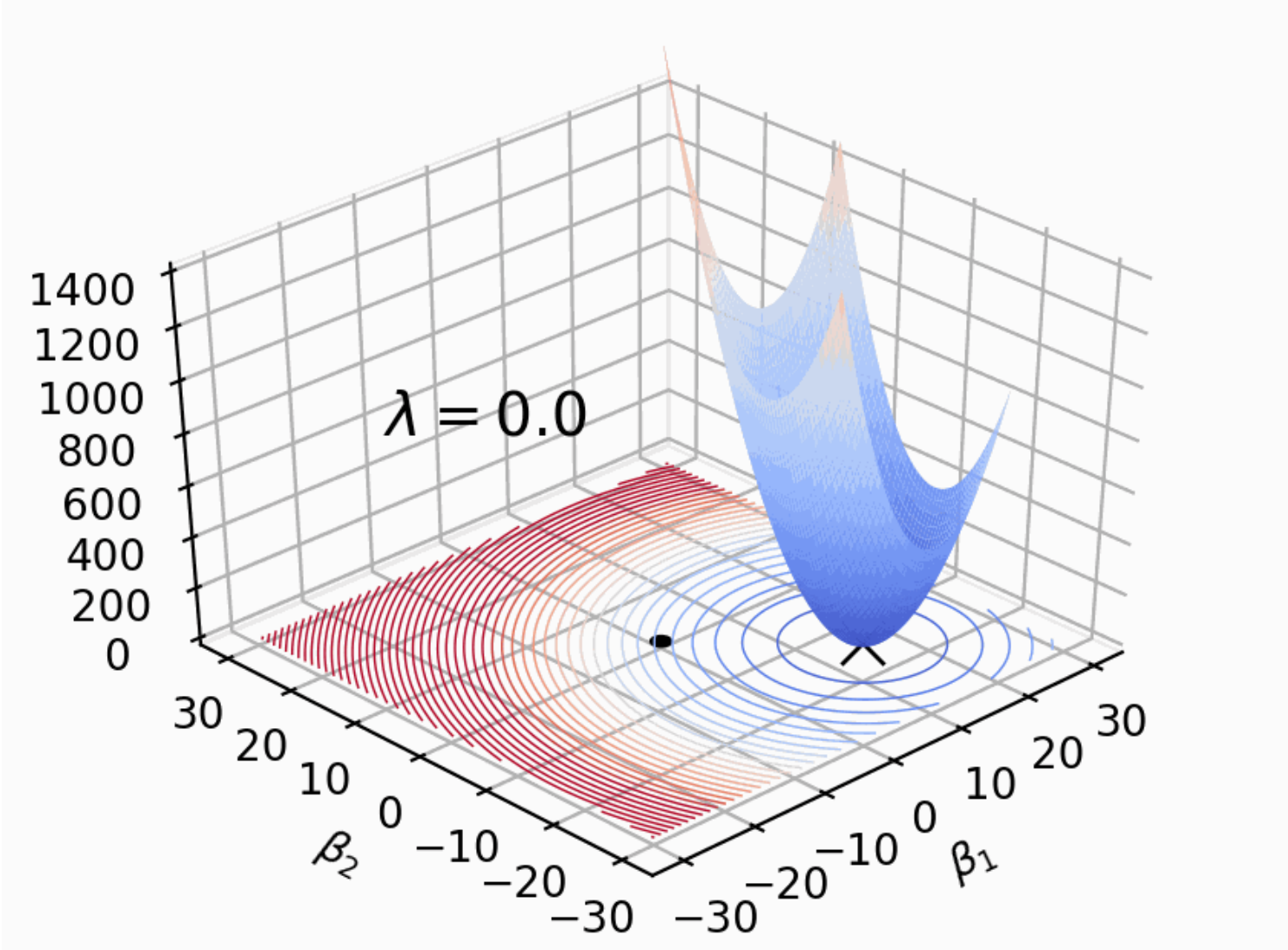
Is always invertible!!

Why?

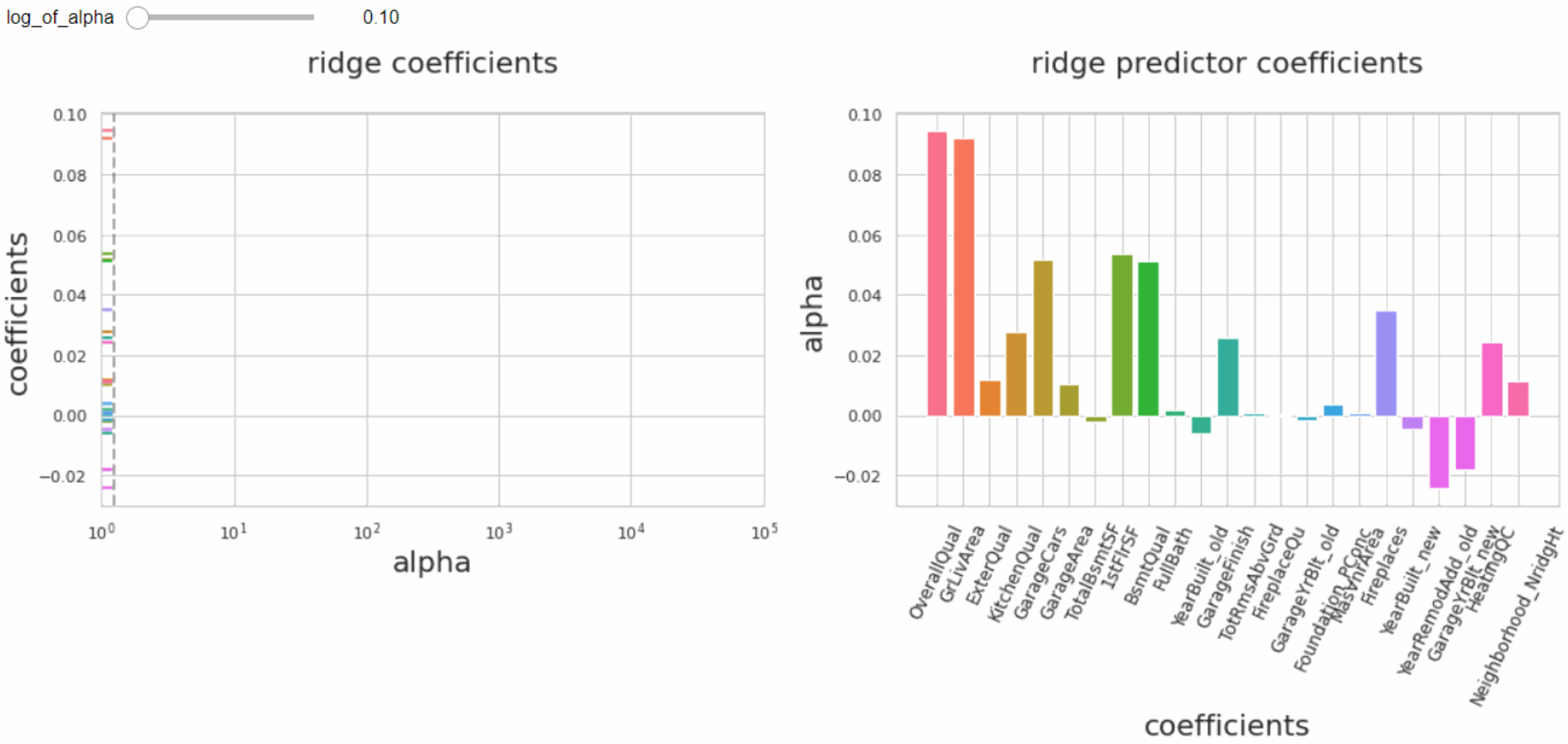
Geometric interpretation



Geometric interpretation



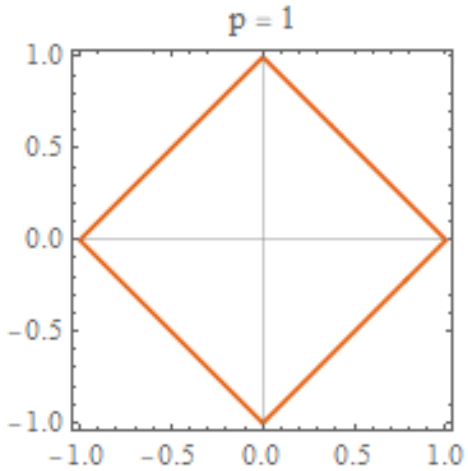
Geometric interpretation



Lasso regression

$$\min \|w\|_0 \quad s.t. \quad y_i = w^T x_i$$

$$\arg\min_w \|y - X^T w\|_2^2 + \lambda \|w\|_1$$



Lasso optimization problems

Convexity

Both the sum of squares and the lasso penalty are convex, and so is the lasso loss function. Consequently, there exist a global minimum. However, the lasso loss function is not strictly convex. Consequently, there may be multiple β's that minimize the lasso loss function.

Problem

In general, there is no explicit solution that optimizes the lasso loss function.

Solution

Resort to numerical optimization procedures, e.g., gradient ascent.

Least angle regression (a possible solution)

1. Standarize the predictors (x_i) to have mean zero and unit norm. Start with the residual $r = y - \hat{y}$ and $w_1, w_2 \ldots, w_d = 0$.
2. Find the predictor x_j most correlated with r.
3. Move $w_j$towards its least-squares coefficient until some other competitor x_k has as much correlation with the *current* residual as does x_j.
4. Move w_j and $w_k$in the direction defined by their joint least squares coefficient of the current residual on (x_j, x_k)until some other competitor x_l has as much correlation with the *current* residual.
    a) If a non-zero coefficient hits zero, drop its variable from the active set of variables and recompute the current joint least squares direction.
5. Continue in this way until all d predictors have been entered (Full least squares solution).

Least angle regression

1. Standarize the predictors (x_i) to have mean zero and unit norm. Start with the residual $r = y - \hat{y}$ and $w_1, w_2 \ldots, w_d = 0$.

$$x_i = \frac{\dot{x}_i - \langle \dot{x}_i \rangle}{\|\dot{x}_i\|_2}$$

Least angle regression

1. Standarize the predictors (x_i) to have mean zero and unit norm. Start with the residual $r = y - \hat{y}$ and $w_1, w_2 \ldots, w_d = 0$.
2. Find the predictor x_j most correlated with r.

$$c(x_j, y) = \frac{\sum_{j=1}^{N} x_{ij}\, y_j}{\sum_{j=1}^{N} {x_{ij}}^2 \sum_{j=1}^{N} {y_j}^2}$$

Least angle regression

1. Standarize the predictors (x_i) to have mean zero and unit norm. Start with the residual $r = y - \hat{y}$ and $w_1, w_2 \ldots, w_d = 0$.
2. Find the predictor x_j most correlated with r.
3. **Move $w_j$towards its least-squares coefficient until some other competitor x_k has as much correlation with the *current* residual as does x_j.**
4. **Move w_j and $w_k$in the direction defined by their joint least squares coefficient of the current residual on (x_j, x_k)until some other competitor x_l has as much correlation with the *current* residual.**

Moving in the weight space: The direction

- We need a direction of moving and a step size (like in the steepest descent algorithm)
- If $\mathcal{A}_k$ is the active set of variables at the beginning of step kth (just after identifying x_k) and $w_{\mathcal{A}_k}$ the weights at this moment (the weight correspondent with x_k would be equal to zero at this point):

$$r_k = y - \hat{y}_k = y - X^T{}_{\mathcal{A}_k} w_{\mathcal{A}_k}$$

And the direction for the next step would be:

$$\delta_k = \left(X_{\mathcal{A}_k} X^T{}_{\mathcal{A}_k}\right)^{-1} X_{\mathcal{A}_k} r_k$$

This direction warrants that the correlation of the residuals with all the active variables is the same

So our weights should evolve as:

$$w_{\mathcal{A}_k}(\alpha) = w_{\mathcal{A}_k} + \alpha \delta_k$$

Moving in the weight space: Step size

$$w_{\mathcal{A}_k}(\alpha) = w_{\mathcal{A}_k} + \alpha\left(\left(X_{\mathcal{A}_k}X^T{}_{\mathcal{A}_k}\right)^{-1}X_{\mathcal{A}_k}r_k\right)$$

Move until some other competitor x_l has as much correlation with the current residual.

$$r_{k+1}(\alpha) = y - \hat{y}_{k+1} = y - X^T{}_{\mathcal{A}_k}w_{\mathcal{A}_k}(\alpha)$$

We have to find the value of α and the variable l for which the above condition holds.

We need to use the result that the correlations for all the variables in the active set are equal, so:

$$c_a(\alpha) = x_a^T r_{k+1}(\alpha) = x_a^T\left(y - X^T{}_{\mathcal{A}_k}w_{\mathcal{A}_k}(\alpha)\right) = x_a^T\left(y - X^T{}_{\mathcal{A}_k}\left(w_{\mathcal{A}_k} + \alpha\delta_k\right)\right)$$
$$= x_a^T\left(y - X^T{}_{\mathcal{A}_k}w_{\mathcal{A}_k} - \alpha X^T{}_{\mathcal{A}_k}\delta_k\right) = x_a^T r_k(\alpha) - \alpha x_a^T X^T{}_{\mathcal{A}_k}\delta_k$$

Where a is any of the active variables. For non active variables (b) the same formula is valid

$$c_b(\alpha) = x_b^T\, r_k(\alpha) - \alpha x_b^T X^T{}_{\mathcal{A}_k}\delta_k$$

Moving in the weight space: Step size

We have to find for alll the non-active variables the α in which

$$|c_a(\alpha)| = |c_b(\alpha)|$$

Depending on the signs

$$\alpha = \frac{x_b^T r_k(\alpha) - x_a^T r_k(\alpha)}{x_b^T X^T{}_{\mathcal{A}_k} \delta_k - x_a^T X^T{}_{\mathcal{A}_k} \delta_k} \; or$$

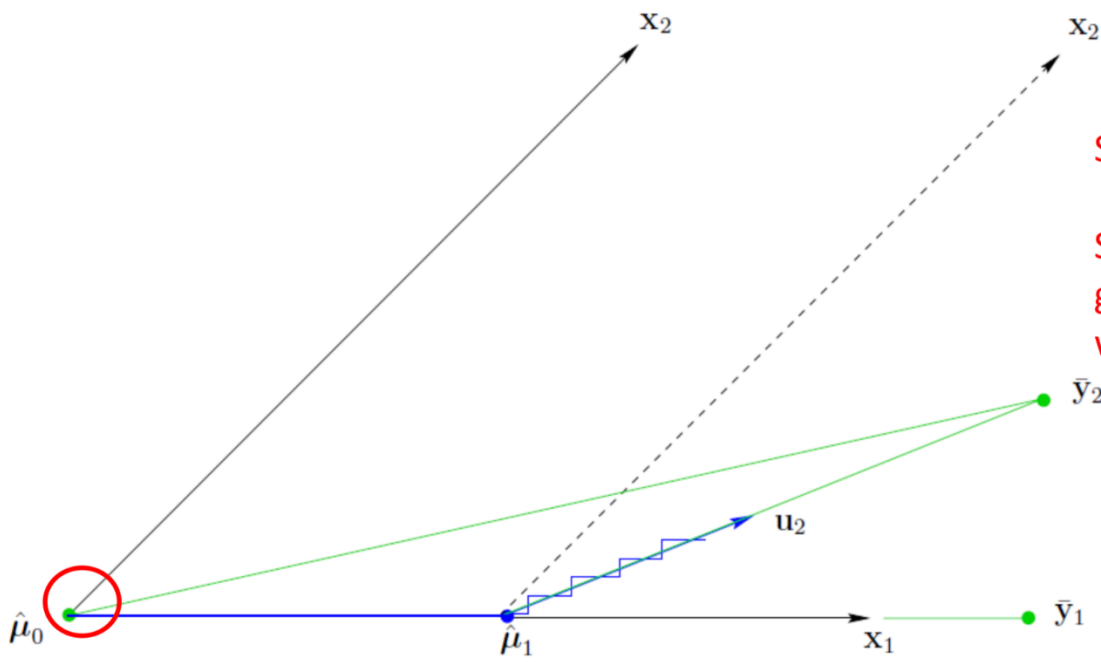
$$\alpha = \frac{x_b^T r_k(\alpha) + x_a^T r_k(\alpha)}{x_b^T X^T{}_{\mathcal{A}_k} \delta_k + x_a^T X^T{}_{\mathcal{A}_k} \delta_k}$$

And then find b as

$$\max_b |c_b(\alpha)|$$

A geometric interpretation of LARS algorithm

LARS(Least Angle Regression Shrinkage)

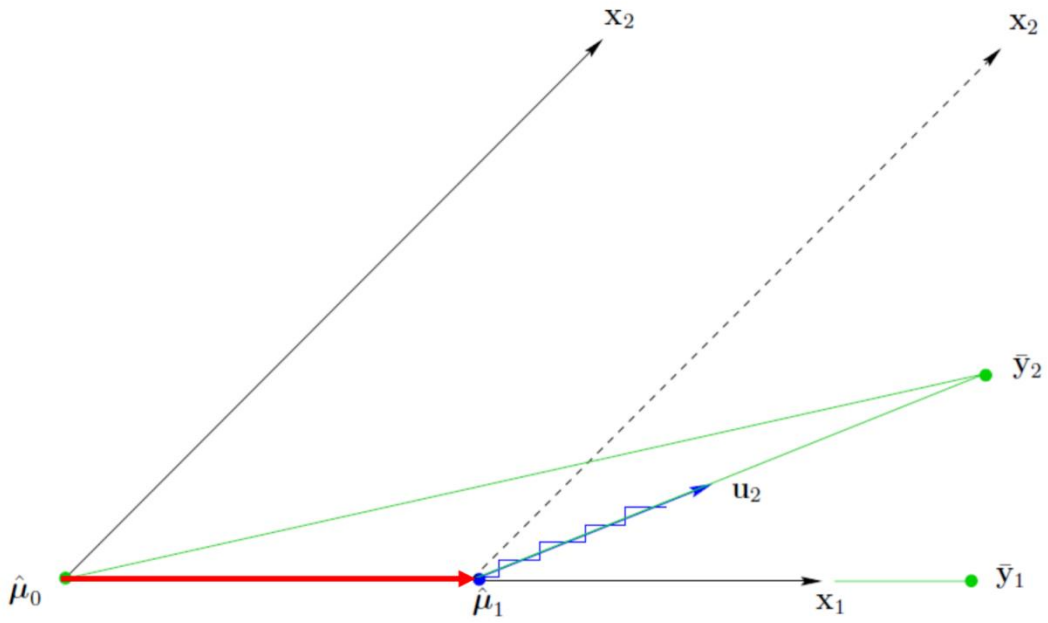


Step 1: Start with $\hat{u}_0=0$

Step 2: The residual $\hat{y}_2-\hat{u}_0$ has a greater correlation with x_1 than with x_2

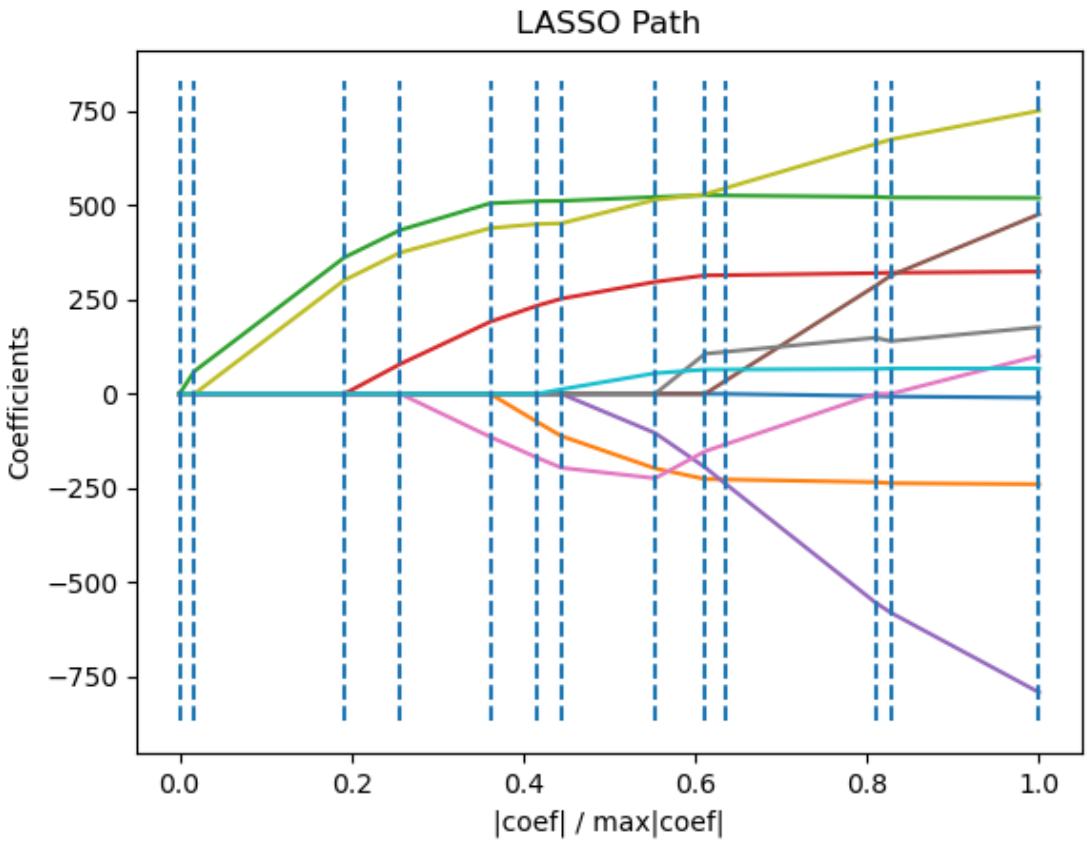
A geometric interpretation of LARS algorithm

LARS(Least Angle Regression Shrinkage)



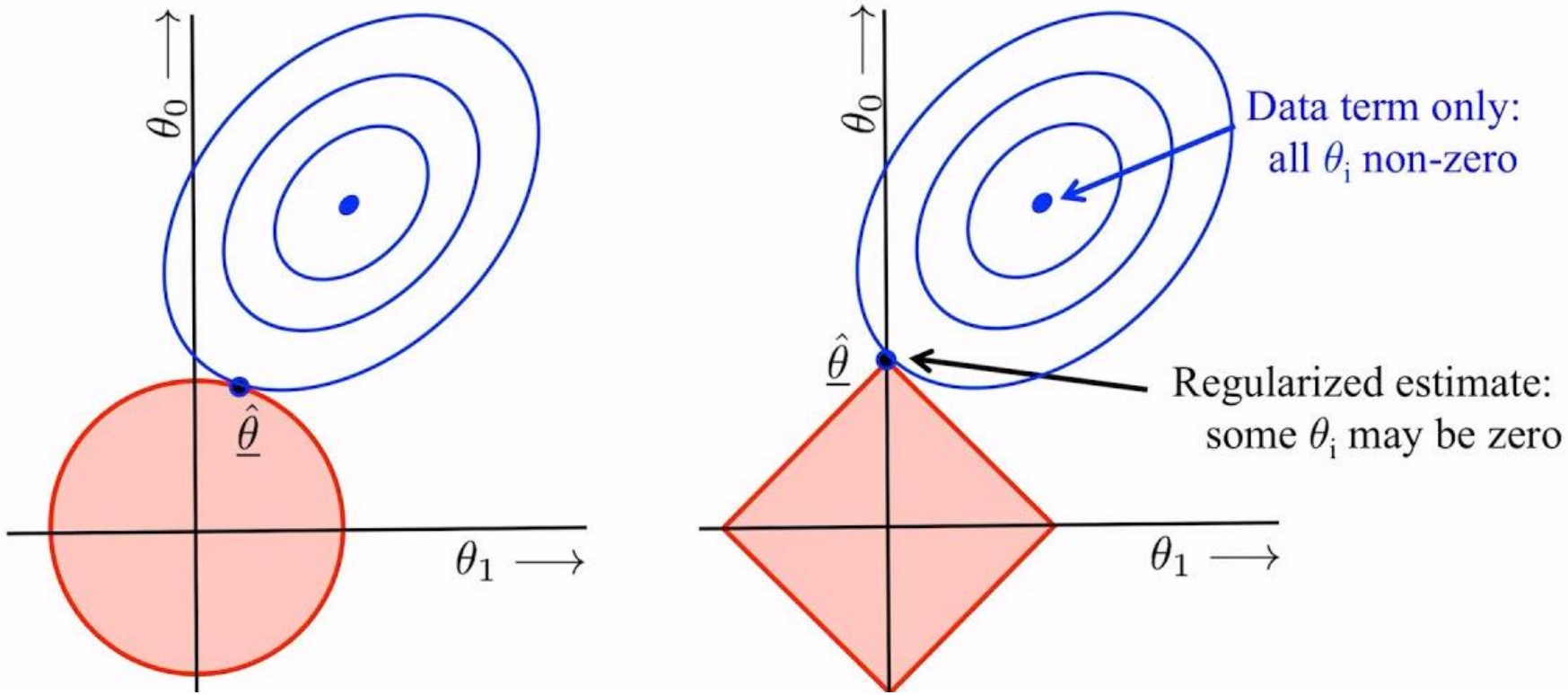
Step 3: Move in the direction of x_1

What is the result?

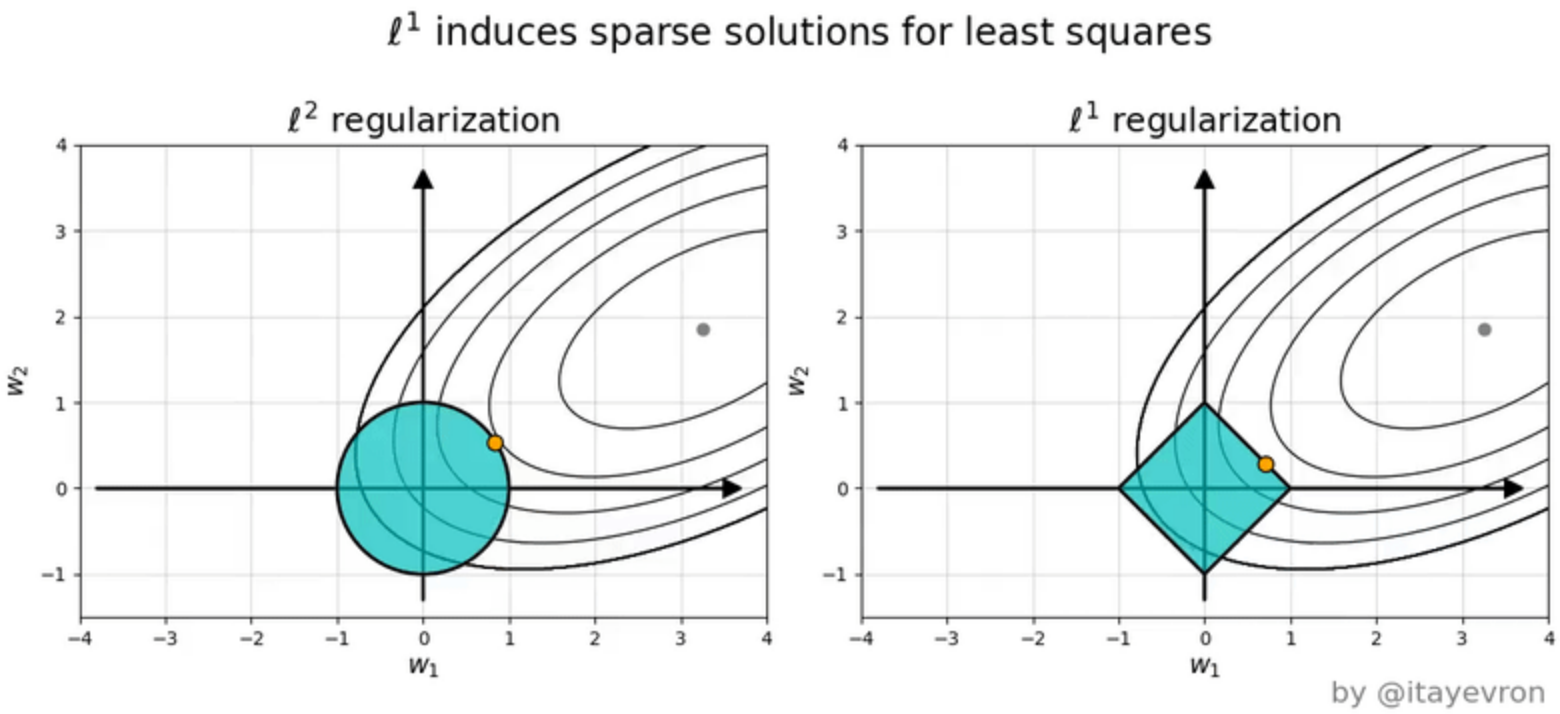


Ridge vs Lasso geometric interpretation

- L1 tends to generate sparser solutions than a quadratic regularizer

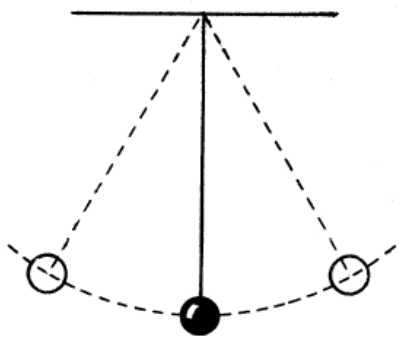


Ridge vs Lasso geometric interpretation



Data-driven discovery of new interpretable dynamical models

Suppose you want to learn the harmonic oscillator equation from data:



$$\{u((x,t)_i) \equiv u_i\}_{i=1}^{N}$$
$$\downarrow$$
$$(\partial_t - \omega^2 \partial_{xx})u \equiv Lu = 0$$

A way to proceed is to make the hypothesis that the differential equation is of the form:

$$\partial_t u = \{1, u, u^2, u_x, u_x^2, uu_x, u_{xx}, u_{xx}^2, uu_{xx}, u_x, u_{xx}\}\alpha = D\alpha$$

and minimize the number of active terms in α. Minimization problem:

Dictionary Approach

$$\alpha^* = \arg\min_{\alpha} \|\partial_t \mathbf{u_i} - \mathbf{D_i}\alpha\|_2 + \|\alpha\|_0$$

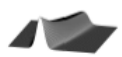
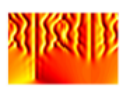
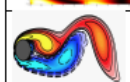
Data-driven discovery of new interpretable dynamical models

SCIENCE ADVANCES | RESEARCH ARTICLE

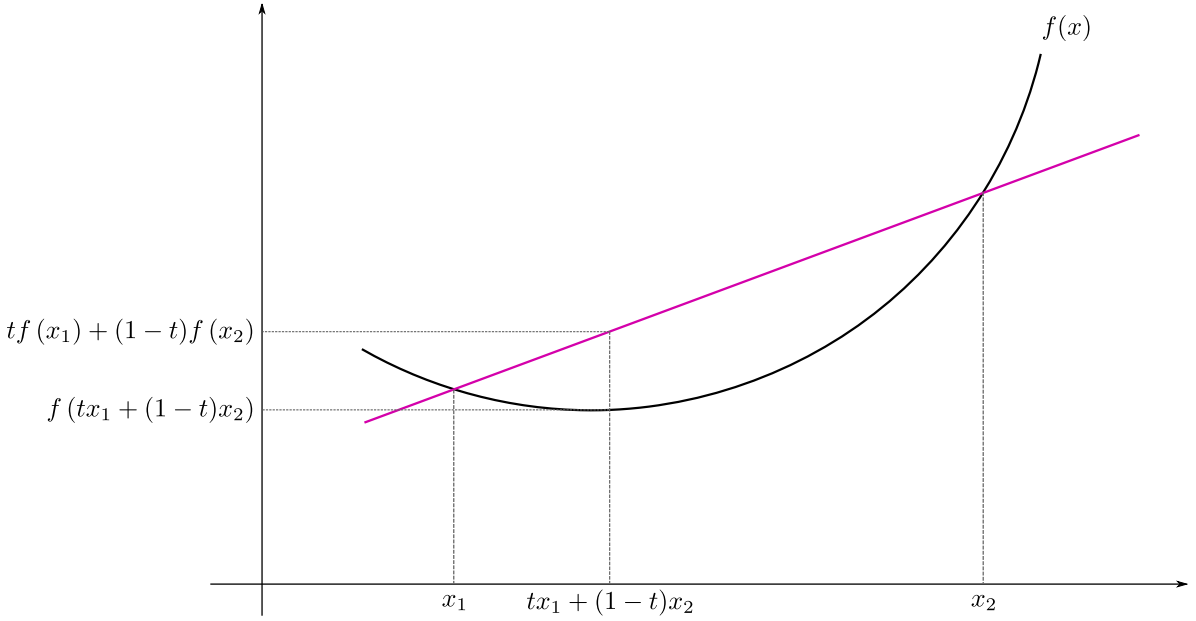
APPLIED MATHEMATICS

Data-driven discovery of partial differential equations

Samuel H. Rudy,[1]* Steven L. Brunton,[2] Joshua L. Proctor,[3] J. Nathan Kutz[1]

PDE	Form	Error (no noise, noise)	Discretization
 KdV	$u_t + 6uu_x + u_{xxx} = 0$	1%±0.2%, 7%±5%	x∈[−30, 30], n=512, t∈[0, 20], m=201
 Burgers	$u_t + uu_x - \epsilon u_{xx} = 0$	0.15%±0.06%, 0.8%±0.6%	x∈[−8, 8], n=256, t∈[0, 10], m=101
 Schrödinger	$iu_t + \frac{1}{2}u_{xx} - \frac{x^2}{2}u = 0$	0.25%±0.01%, 10%±7%	x∈[−7.5, 7.5], n=512, t∈[0, 10], m=401
 NLS	$iu_t + \frac{1}{2}u_{xx} + \lvert u\rvert^2 u = 0$	0.05%±0.01%, 3%±1%	x∈[−5, 5], n=512, t∈[0, π], m=501
 KS	$u_t + uu_x + u_{xx} + u_{xxxx} = 0$	1.3%±1.3%, 70%±27%	x∈[0, 100], n=1024, t∈[0, 100], m=251
u  v Reaction Diffusion	$u_t = 0.1\nabla^2 u + \lambda(A)u - \omega(A)v$ $v_t = 0.1\nabla^2 v + \omega(A)u + \lambda(A)v$ $A^2 = u^2 + v^2, \omega = -\beta A^2, \lambda = 1 - A^2$	0.02% ± 0.01%, 3.8% ± 2.4%	x, y∈[−10, 10], n=256, t∈[0, 10], m=201 subsample 1.14%
 Navier Stokes	$\omega_t + (\mathbf{u}\cdot\nabla)\omega = \frac{1}{Re}\nabla^2\omega$	1% ± 0.2% , 7% ± 6%	x∈[0, 9], n_x=449, y∈[0, 4], n_y=199, t∈[0, 30], m=151, subsample 2.22%

L1 vs L0. Convexity



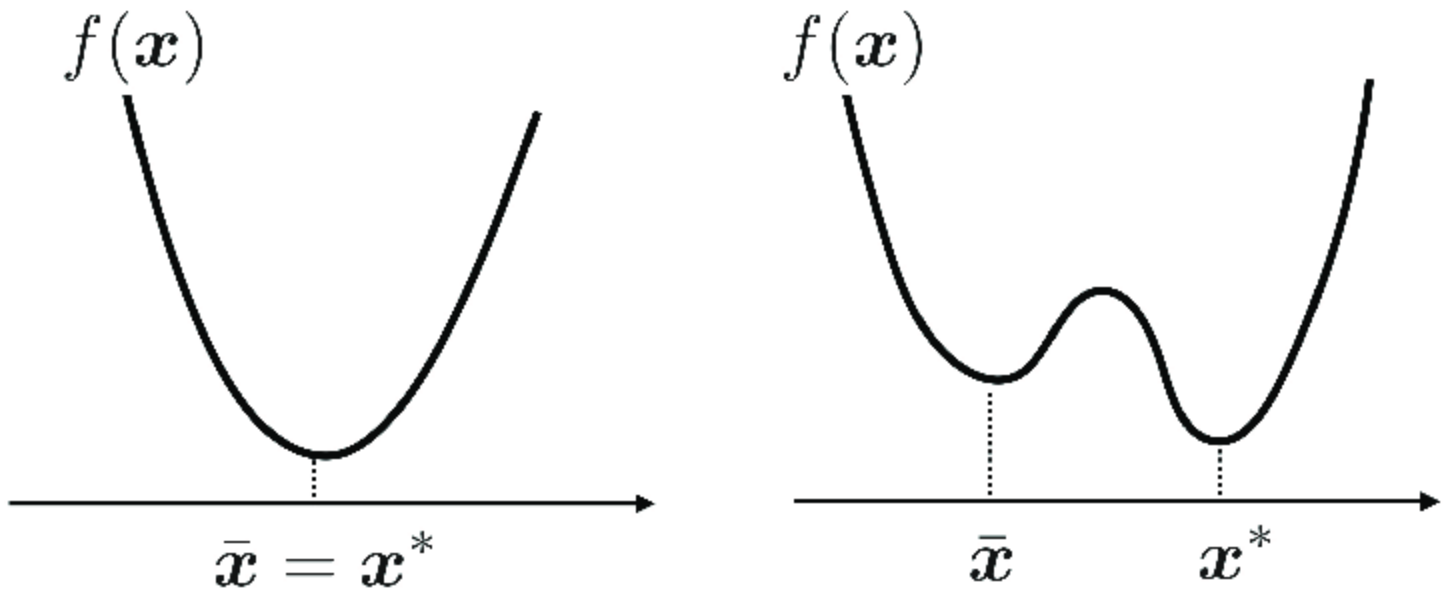
What is L0 of a vector counting?

For all $0 \leq t \leq 1$ and all $x_1, x_2 \in X$:

$$f(tx_1 + (1-t)x_2) \leq tf(x_1) + (1-t)f(x_2)$$

Why is convexity important?

Convexity: global= local minima



Why not l0?

Lemma 1.11 (ℓ_0 "norm"). *The ℓ_0 "norm" is not convex.*

Proof. We provide a simple counterexample. Let $x := \binom{1}{0}$ and $y := \binom{0}{1}$, then for any $\theta \in (0,1)$

$$||\theta x + (1-\theta)\, y||_0 = 2 > 1 = \theta\, ||x||_0 + (1-\theta)\, ||y||_0 \,. \tag{11}$$

□

Ridge vs lasso

- The ridge regression helps only in avoiding the model getting overfitted by reducing the coefficients of the features and keeping all the features present in the model. But in the case of lasso regression, apart from reducing the complexity of the model, it helps in automatic feature selection also. Lasso regression transforms the coefficient values to zero. The feature whose coefficients become equal to zero is less important in predicting the target variable and hence it can be dropped from the model.

- **Limitations of Ridge and Lasso Regressions :**

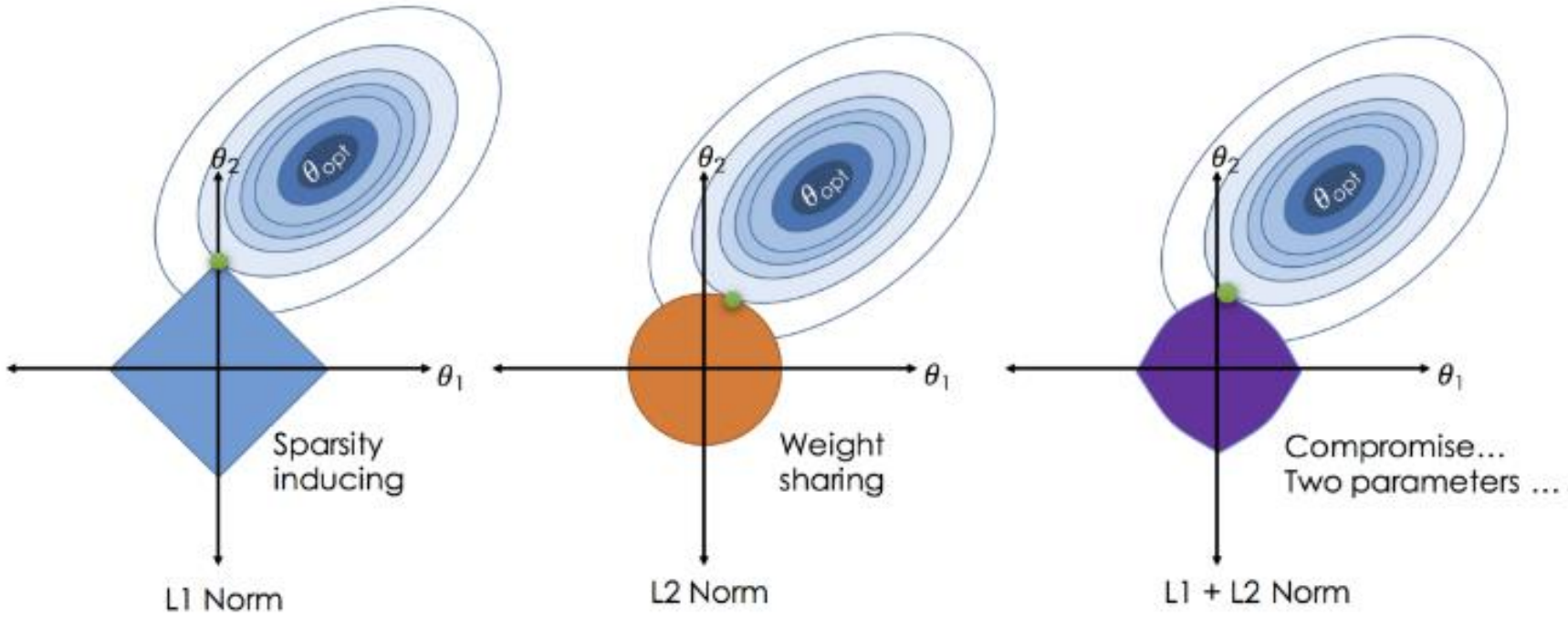
  - Ridge regression does not help in feature selection.

  - Ridge regression use to shrink the coefficients, but never sets their values as absolute zero. The model will retain all the features and will remain complex, which may lead to poor model performance.

  - When we apply Lasso regression to a model which has highly correlated variables, then it will retain only a few variables and sets other variables to be zero. That will lead to some loss of information as well as lower accuracy of the model.

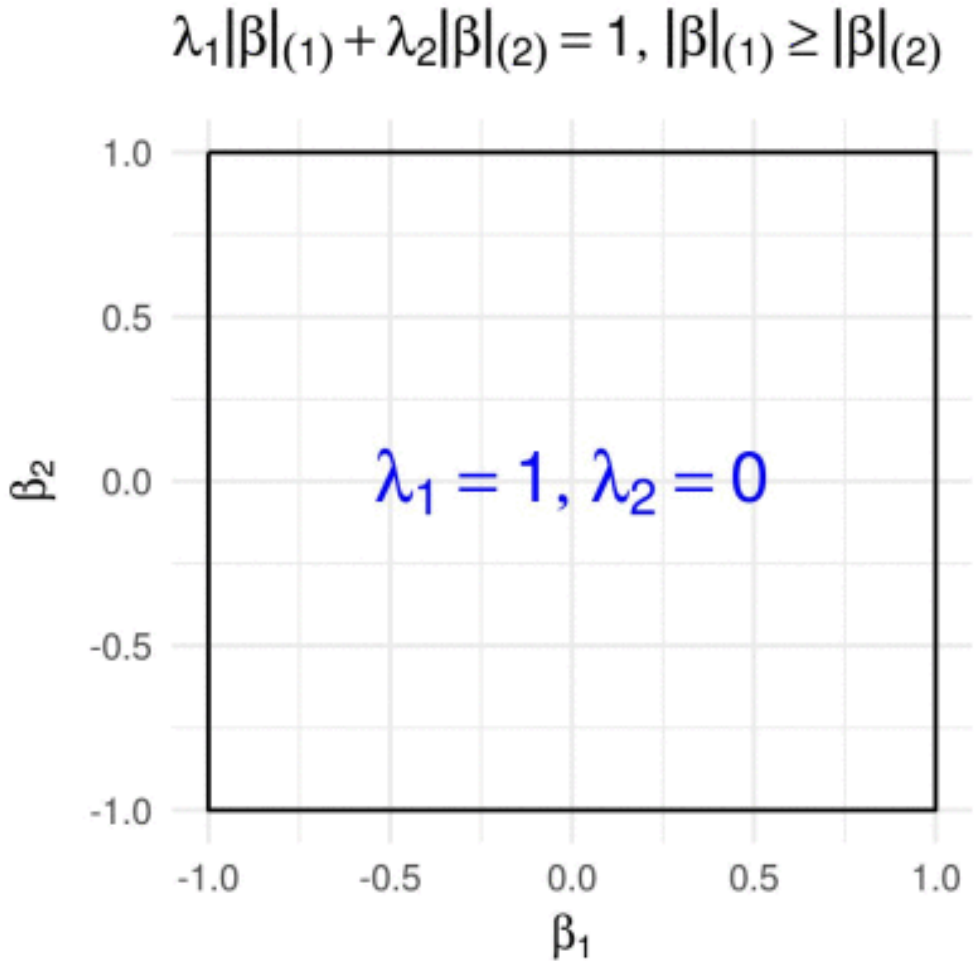
Elastic nets

$$\arg\min_{w} \|y - X^T w\|_2^2 + \lambda(\alpha\|w\|_1 + (1-\alpha)\|w\|_2)$$



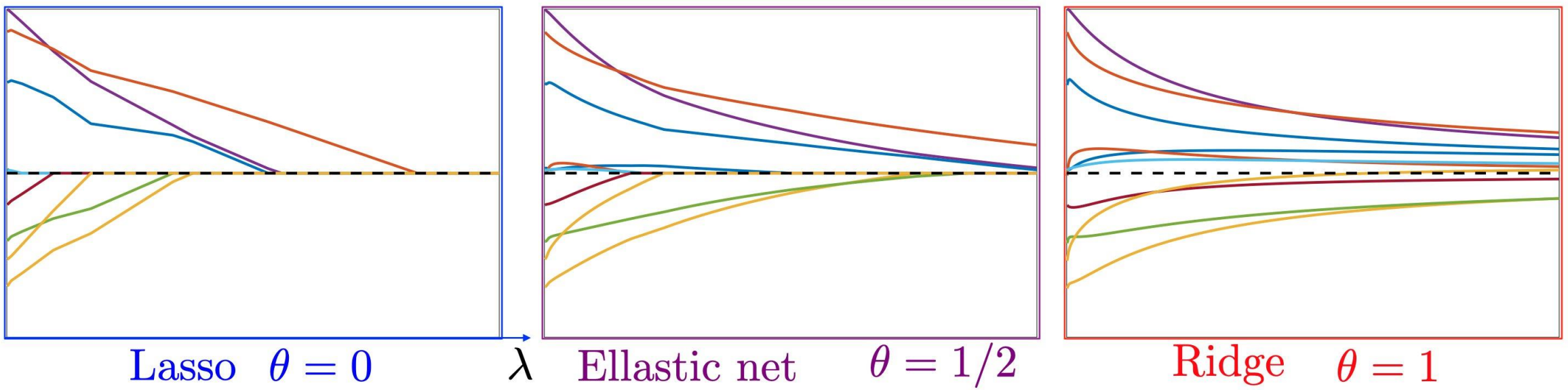
Which is the advantage?

Elastic nets



Elastic net: $x_\lambda \in \underset{x}{\operatorname{argmin}} \frac{1}{2\lambda}\|Ax-y\|^2 + (1-\theta)\|x\|_1 + \frac{\theta}{2}\|x\|_2^2$

Regularization path: $\lambda \longmapsto x_\lambda$



Elastic nets

$$\arg\min_{w} \|y - X^T w\|_2^2 + \lambda(\alpha\|w\|_1 + (1-\alpha)\|w\|_2)$$

Can you derive the gradient descent algorithm?