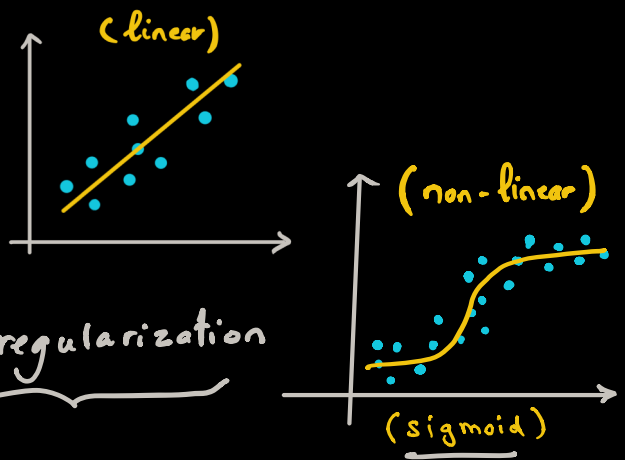
# $\dot{M}\dot{\infty}\dot{L}_6$

1. Basics of linear regression

2. Non-linear regression and $l_2$ regularization

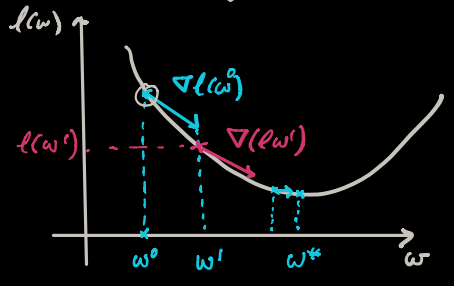
3. Feature design for linear regression



# $\mathcal{M} \infty \mathcal{L}_5 \looparrowright ?$

## Gradient descent

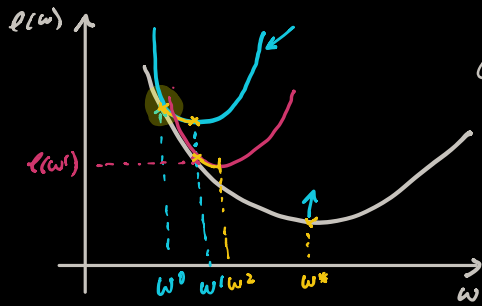
- $\alpha$ : learning rate
- $w^0$ : initial guess
- $\varepsilon$ : stopping "value"




$$w^k = w^{k-1} - \alpha \nabla \ell(w^{k-1})$$

## Newton's method

- $w^0$
- $\varepsilon$ ; $\| \nabla \ell(w^*) \| \leq \varepsilon$




⊕ if $\ell$ is convex → faster convergence
⊕ no $\alpha$
⊖ if $\ell$ is non-convex ⇒ not converge
⊖ $[\nabla^2 \ell(w)]^{-1}$ computationally expensive

$$w^k = w^{k-1} - \left[\nabla^2 \ell(w^{k-1})\right]^{-1} \nabla \ell(w^{k-1})$$

**Estimating a predictive learner** { classifier / regressor

Think about how to define $\ell(w)$ ← ① define loss function & its set of parameters $\ell(w)$

optimization ? ← ② minimize $\ell(w)$ using the training data $w$

cross-validation ← ③ Once the optimal set of model parameters $w^*$ is estimated, evaluate model on testing data.

# ① Basics of linear regression

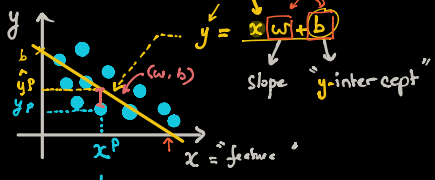
## Training step

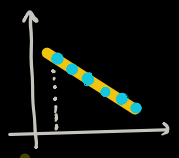
- **Training samples:**

$$\left\{ \begin{matrix} (x^1, y^1) \\ (x^2, y^2) \\ (x^n, y^n) \end{matrix} \right\} \quad \{(x^1,y^1), (x^2,y^2), \dots (x^n,y^n)\} = \{(x^i,y^i)\}_{i=1}^{n}$$

$x^i$: feature vector (input) $\in \mathbb{R}^d$ — $y^i$: scalar-valued target (output) $\in \mathbb{R}$ — $f_w$

- **Case** $d = 1$ (scalar input)



$$y^p \approx x^p w + b = \hat{y}^p$$

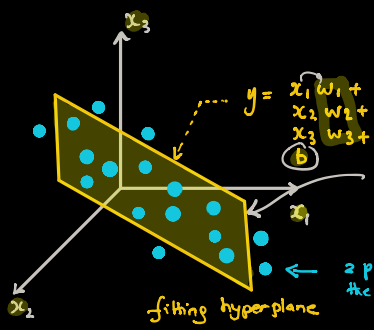
* what is the approximation or prediction of the target $y^p$ by this model?

- **Case** $d \geq 1$ (general case)

$$x^p = \begin{bmatrix} x_1^p \\ x_2^p \\ \vdots \\ x_d^p \end{bmatrix} \Rightarrow \begin{cases} y^p \approx x^{pT} w + b \\ p = 1, \dots, n \end{cases}$$

($w$: weight, $b$: bias)

$$f(w,b) = f_{w,b}$$
$$f_{w,b}(x^p) \approx \hat{y}^p$$




$\in \mathbb{R}^{d+1}$ $w$ "$b$"

a point $(x^i, y^i)$ lies close to the hyperplane $y = x^T w + b$ if $y^i \approx x^{iT} w + b$.

- **Example**

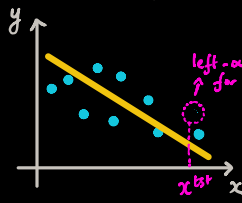

Guggenmos et al. Translational Psychiatry (2017)7:1279
DOI 10.1038/s41398-017-0037-y
Translational Psychiatry
ARTICLE Open Access


**Quantitative neurobiological evidence for accelerated brain aging in alcohol dependence**


Matthias Guggenmos[1], Katharina Schmack[1], Maria Sekutowicz[1], Maria Garbusow[1], Miriam Sebold[1], Christian Sommer[2], Michael N. Smolka[2,3], Hans-Ulrich Wittchen[4], Ulrich S. Zimmermann[2], Andreas Heinz[1] and Philipp Sterzer[1]


## Testing step

- Given an available dataset $\{(x^i, y^i)\}_{i=1}^{n}$
- Use leave-one-out cross validation to train our linear regressor:




(A) how to predict the target $y^{tst}$?

(B) how to evaluate our prediction? (good or bad?)

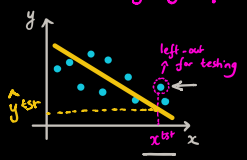
Hint:

$$\tilde{w}^* = \left[\sum_{i=1}^{n-1} \left(\tilde{x}^i \tilde{x}^{iT}\right)\right]^{-1} \sum_{i=1}^{n-1} \tilde{x}^i y^i$$

$$\begin{cases} \tilde{x}^i = \begin{bmatrix} 1 \\ x^i \end{bmatrix} ; \tilde{w} = \begin{bmatrix} b \\ w \end{bmatrix} \end{cases}$$

$\tilde{w}^* \Rightarrow w^*$

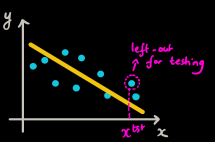
(A) how to predict the target $y^{tst}$? [step-by-step]




$$\hat{y}^{tst} = x^{tst\,T} w^* + b^*$$

($n-1$ samples)

(B) how to evaluate our prediction? (good or bad?)




- Given the optimal solution:

$$\tilde{w}^* = \begin{bmatrix} b^* \\ w^* \end{bmatrix}$$

↓ evaluate our prediction for each left-out subject using

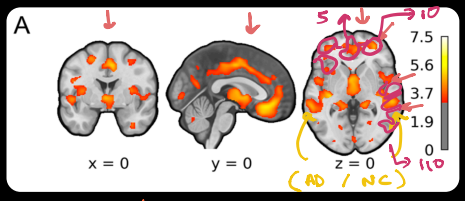
MSE = Mean Squared Error

$$MSE = \frac{1}{n} \sum_{i=1}^{n} (x^{iT} w^* - y^i)^2$$

$$\tilde{w}^* = \left[\sum_{i=1}^{n-1} \tilde{x}^i \tilde{x}^{iT}\right]^{-1} \left[\sum_{i=1}^{n-1} \tilde{x}^i y^i\right]$$

what if we leave 3 subjects out? Would this change the value we get for MSE?

$$MSE_{LOO} \neq MSE_{5\text{-fold}}$$




* one training sample :

$$\left(x^s = \begin{bmatrix} 10 \\ 5 \\ \vdots \\ 110 \end{bmatrix}, \; y^s = ?\right)$$

**Fig. 1** Correspondence between AD-related and age-related grey matter loss (GML). **a** and **b** show t-maps for univariate whole-brain analyses, thresholded at $p < 0.001$ uncorrected, for illustration. **a** T-map for AD-related grey matter volume loss based on a two-sample *t* test between AD subjects and control subjects, controlling for age, gender, site, smoking (FTND sum score) and general health status (WHODAS-

Results

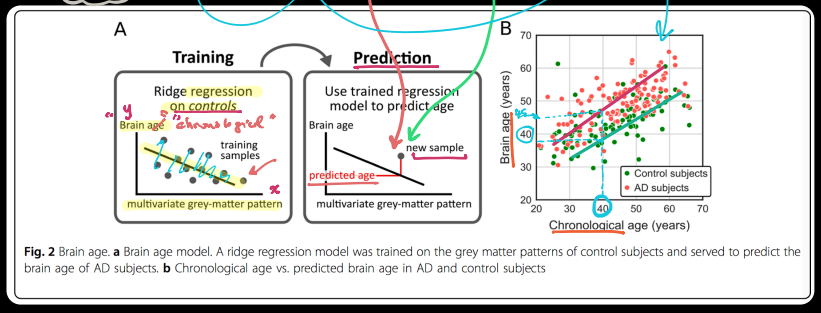
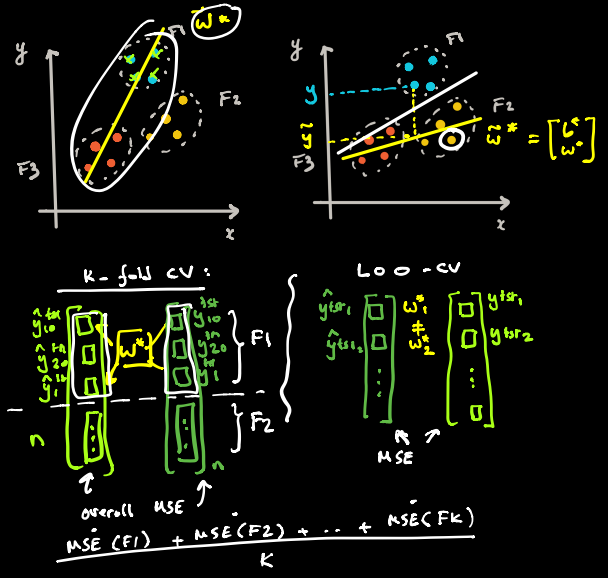



**Fig. 2** Brain age. **a** Brain age model. A ridge regression model was trained on the grey matter patterns of control subjects and served to predict the brain age of AD subjects. **b** Chronological age vs. predicted brain age in AD and control subjects



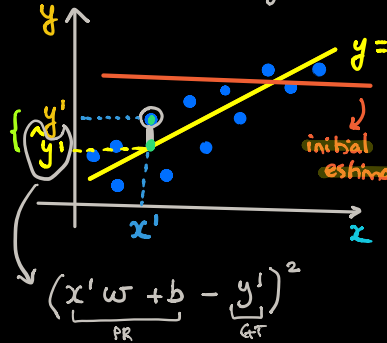

## ② Least squares loss function for linear regression

$$arg\,min_{\mathbf{w}} \mathcal{L}(\mathbf{w}) = \frac{1}{n} \sum_{(\mathbf{x}^i, y^i) \in \mathcal{D}} \underbrace{E(f_{\mathbf{w}}(\mathbf{x}^i), y^i)}_{\ell(w,b)} + R(\dots)$$

($\hat{y}$ : $f_{\mathbf{w}}(\mathbf{x}^i)$)

- Our goal ⇒ find the parameters of the hyperplane which best fits a regression dataset.
- Sol° ⇒ these parameters $(w, b) \in \mathbb{R}^{d+1}$

- Think of 2 simple loss function that we can minimize to find the optimal $(w^*, b^*)$.




$y = x^T w^* + b^* = f_{w,b}(x)$

initial estimation

$y = x^T w^0 + b^0$

minimize $_{w,b}$ $\ell(w,b) = ?$

$(x'w + b - y')^2$

PR GT

$$\underset{b,w}{\text{minimize}} \underbrace{\sum_{i=1}^{n}}_{\text{sum of}} \underbrace{\left( x^i w + b - y^i \right)^2}_{\text{least squares}}$$

- Minimization of the loss function $\ell(w, b)$:

$$\tilde{x}^{iT} \tilde{w} = ?$$

$$\tilde{x}^i = \begin{bmatrix} 1 \\ x^i \end{bmatrix} \in \mathbb{R}^{d+1}, \quad \tilde{w} = \begin{bmatrix} b \\ w \end{bmatrix} \in \mathbb{R}^{d+1}$$

$$\begin{bmatrix} 1 & x^i \end{bmatrix} \begin{bmatrix} b \\ w \end{bmatrix} = b + x^i w$$

→ using this compact notation where we have only one variable $\tilde{w}$ to minimize:

$$\ell(\tilde{w}) = \sum_{i=1}^{n} \left( \tilde{x}^{iT} \tilde{w} - y^i \right)^2$$

$(a-b)^2 = a^2 - 2ab + b^2$

Reminder

$$\frac{\partial x^T w}{\partial w} = x$$

→ compute the gradient of $\ell(\tilde{w})$:

$$\frac{d f^2}{dx} = 2 f f'$$

$$\nabla \ell(\tilde{w}) = \frac{\partial \ell(\tilde{w})}{\partial \tilde{w}} = \frac{\partial}{\partial \tilde{w}} \left[ \sum_{i=1}^{n} \left( \left( \tilde{x}^{iT} \tilde{w} \right)^2 - 2 \tilde{x}^{iT} \tilde{w} \, y^i + y^{i2} \right) \right]$$

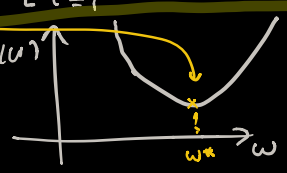
$$= \sum_{i=1}^{n} \left[ 2 \tilde{x}^i \tilde{x}^{iT} \tilde{w} - 2 \tilde{x}^i y^i \right]$$

$\tilde{w}^*$ ?

$$= 0$$

$$\sum_{i=1}^{n} \tilde{x}^i \tilde{x}^{iT} \tilde{w} = \sum_{i=1}^{n} \tilde{x}^i y^i$$

$$w^* = \left[ \sum_{i=1}^{n} \tilde{x}^i \tilde{x}^{iT} \right]^{-1} \sum_{i=1}^{n} \tilde{x}^i y^i$$



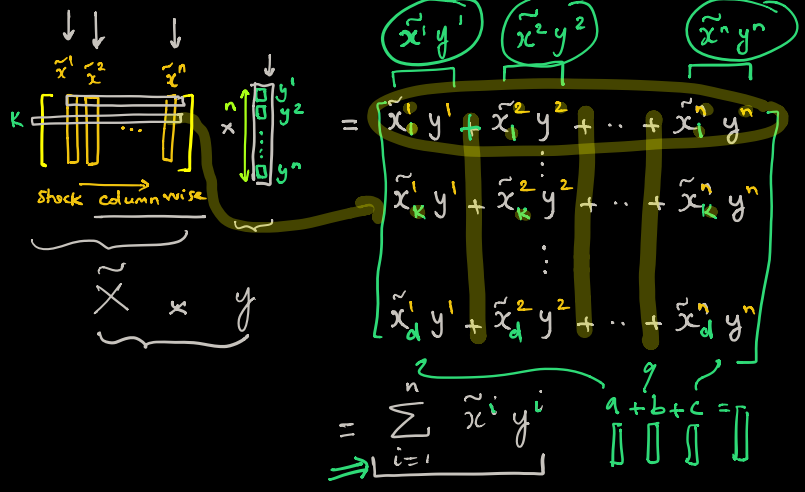

closed form solution

Algebra trick

$$\left( \sum_{i=1}^{n} \tilde{x}^i \tilde{x}^{iT} \right) \tilde{w} = \sum_{i=1}^{n} \tilde{x}^i y^i$$

linear system to solve

$$\Rightarrow \tilde{X} \tilde{X}^T \tilde{w} = \tilde{X} y, \quad y = \begin{bmatrix} y^1 \\ \vdots \\ y^n \end{bmatrix}$$




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

Algebraic solution to the linear least squares loss function

$$l(\tilde{w}) = \sum_{i=1}^{n} (\tilde{x}^{iT} \tilde{w} - y^i)^2$$

$$w^* = [\tilde{X}\tilde{X}^T]^{-1} \tilde{X} y$$

closed form sol$^n$ if $\tilde{X}\tilde{X}^T$ is invertible.

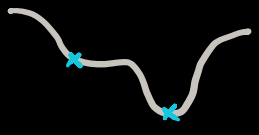
∴ Still, better to solve the original linear system as computing the inverse of a matrix might be computationally expensive.

## ② Non-linear regression an $l_2$ regularization

A) Previously, the loss function is convex & linear in $w$

$$\begin{cases} \ell(w,b) = (x^T w + b - y)^2 \Leftarrow \\ \text{or} \\ \ell(\tilde{w}) = (\tilde{x}^T \tilde{w} - y)^2 \quad \tilde{w} = \begin{bmatrix} b \\ w \end{bmatrix} \end{cases}$$

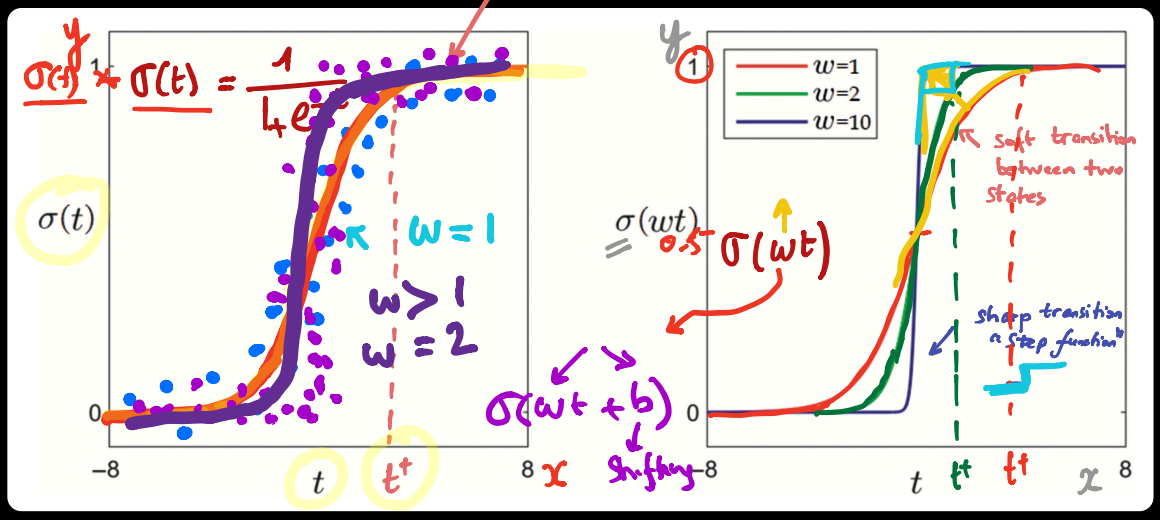

B) loss function is non-convex & nonlinear in $w$:



this requires an additional regularization term to improve loss optimization.

# Example : Logistic regression

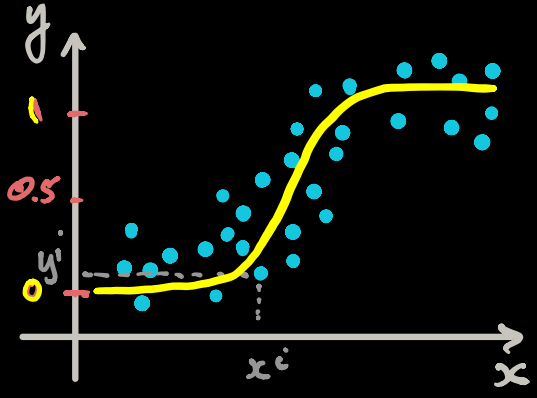
## classic non-linear & nonconvex problem




logistic sigmoid function

- $\sigma$ was designed to model population growth over time.
- At some point $t \gtrsim t^\dagger$, the growth stabalizes and levels off due to lack of limited resources (e.g., food, space, etc).

- Given a dataset $\{(x^i, y^i)\}_{i=1}^n$ that is distributed like a sigmoid function:

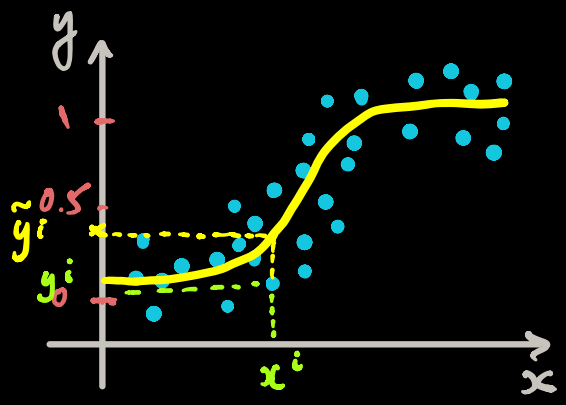



**Goal** = estimate the **best** sigmoid function fitting our data.

**Sol°** = parameterize $\sigma(w,b)$

$x \longrightarrow wx + b$

$w$ ↓ Smoothness of the transition $y=0 \to y=1$




The data satisfies

$$\tilde{y}^i = \sigma(x^i w + b) \approx y^i$$

$$\Downarrow$$

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

$$\sigma(\underbrace{x^{iT}}_{\in \mathbb{R}^q} \underbrace{w}_{\in \mathbb{R}^q} + \underbrace{b}_{\text{value (shifting factor)}}) \approx y^i$$

$$\begin{cases} \ell(w, b) = \sum_{i=1}^{n} \left(\sigma(x^{iT} w + b) - y^i\right)^2 \\ \text{training set: } \{(x^i, y^i)\}_{i=1}^{n} \end{cases}$$

- Compact form:

$$\tilde{x}^i = \begin{bmatrix} 1 \\ x^i \end{bmatrix}, \quad \tilde{w} = \begin{bmatrix} b \\ w \end{bmatrix}$$

$$\ell(\tilde{w}) = \sum_{i=1}^{n} \left(\underbrace{\sigma(\tilde{x}^{iT} \tilde{w})}_{f} - y^i\right)^2$$

$$\frac{\partial x^T w}{\partial w} = x$$

$$\rightarrow \nabla \ell(\tilde{w}) = ? \; f$$

Hint: $\frac{\partial \sigma(x^T w)}{\partial w} = \sigma(x^T w)\left(1 - \sigma(x^T w)\right) x$

$$\ell(\tilde{w}) = \sum_{i=1}^{n} \left[ \sigma(\tilde{x}^{iT}\tilde{w})^2 - 2\,\sigma(\tilde{x}^{iT}\tilde{w})\,y^i + y^{i2} \right]$$

$$\frac{\partial \ell(\tilde{w})}{\partial \tilde{w}} = \sum_{i=1}^{n} \Big[ 2\,\sigma(\tilde{x}^{iT}\tilde{w})\,\sigma(\tilde{x}^{iT}\tilde{w})\,(1-\sigma(\tilde{x}^{iT}\tilde{w}))\,\tilde{x}^i - 2y^i\,\sigma(\tilde{x}^{iT}\tilde{w})\,(1-\sigma(\tilde{x}^{iT}\tilde{w}))\,\tilde{x}^i \Big]$$

$$= 2\sum_{i=1}^{n} \left[ (\sigma(\tilde{x}^{iT}\tilde{w}) - y^i)\,\sigma(\tilde{x}^{iT}\tilde{w})\,(1-\sigma(\tilde{x}^{iT}\tilde{w}))\,\tilde{x}^i \right]$$

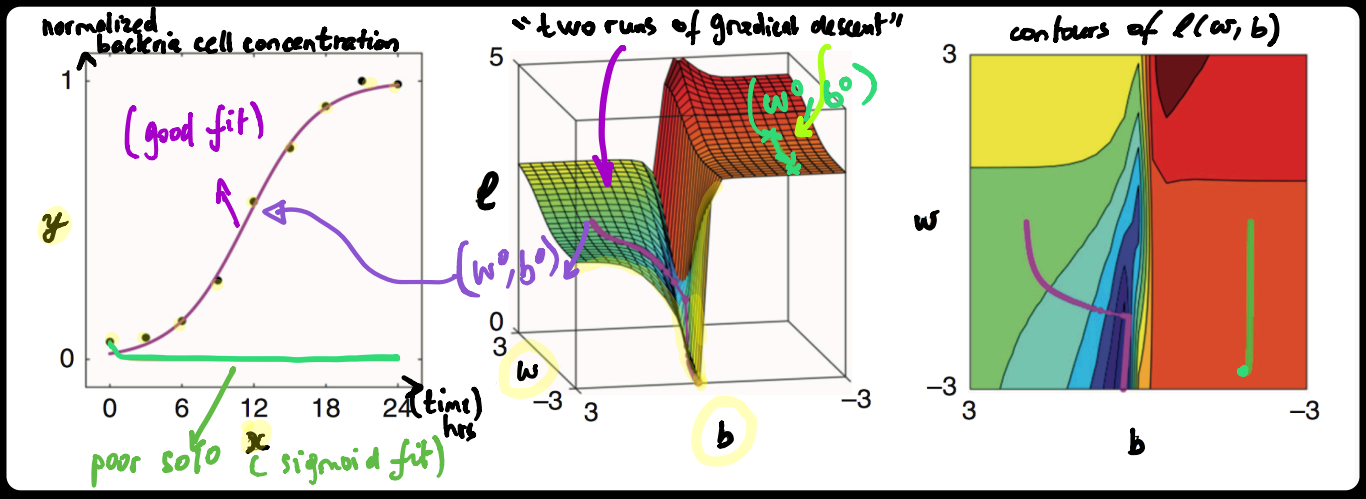
$$\Rightarrow \nabla \ell(\tilde{w}) = 2\sum_{i=1}^{n} (\sigma(\tilde{x}^{iT}\tilde{w}) - y^i)\,\sigma(\tilde{x}^{iT}\tilde{w})\,(1-\sigma(\tilde{x}^{iT}\tilde{w}))\,\tilde{x}^i$$

where $\sigma(\cdot) = \frac{1}{1+e^{-(\cdot)}}$

$\tilde{w}^0$ $\qquad$ $\tilde{w}^* = [\quad]$

Gradient descent can be used to optimize $\ell$!

## Example 1

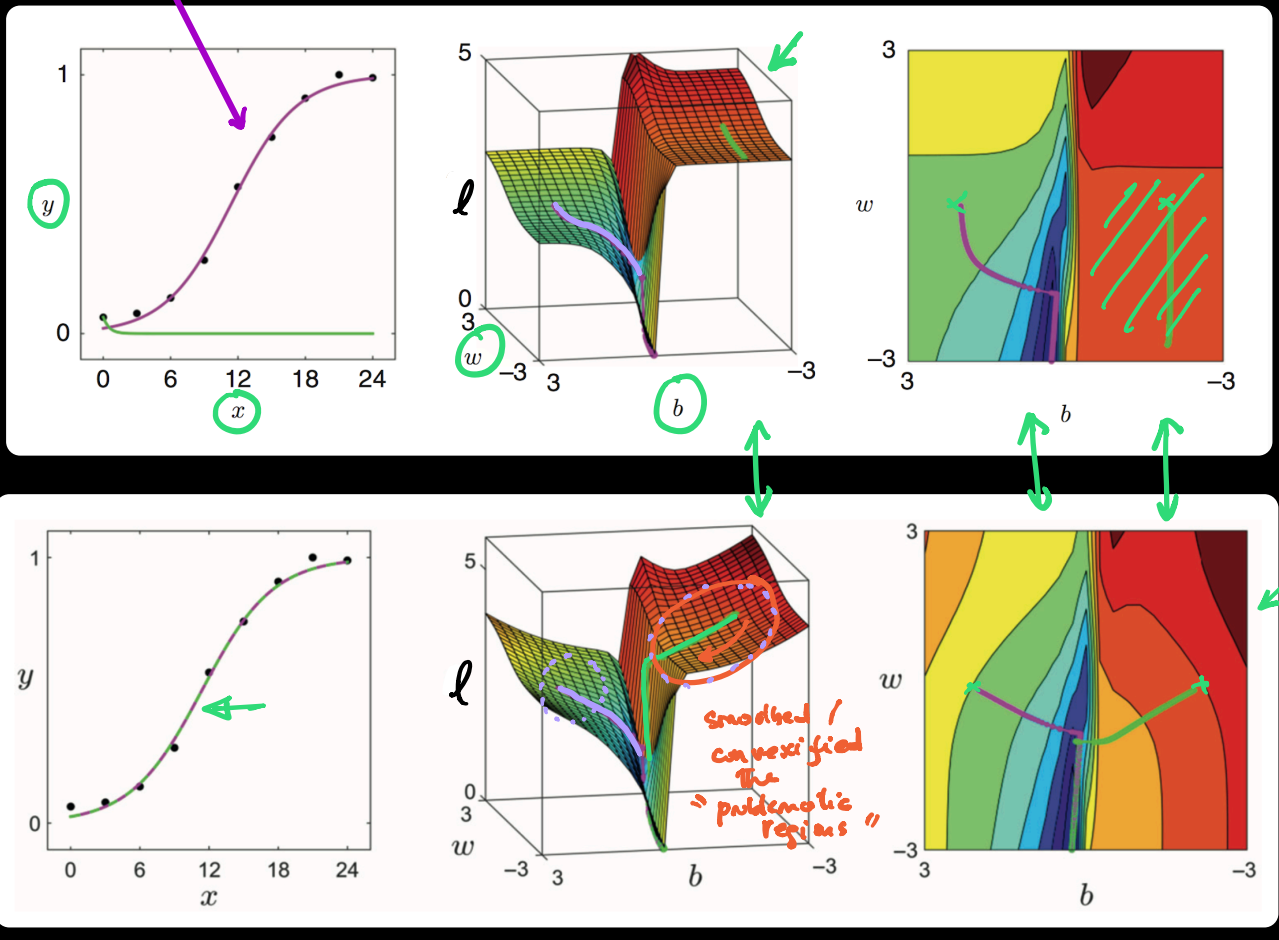



☹ our soln $(w, b)$ depends on the initialization of the gradient descent algorithm.

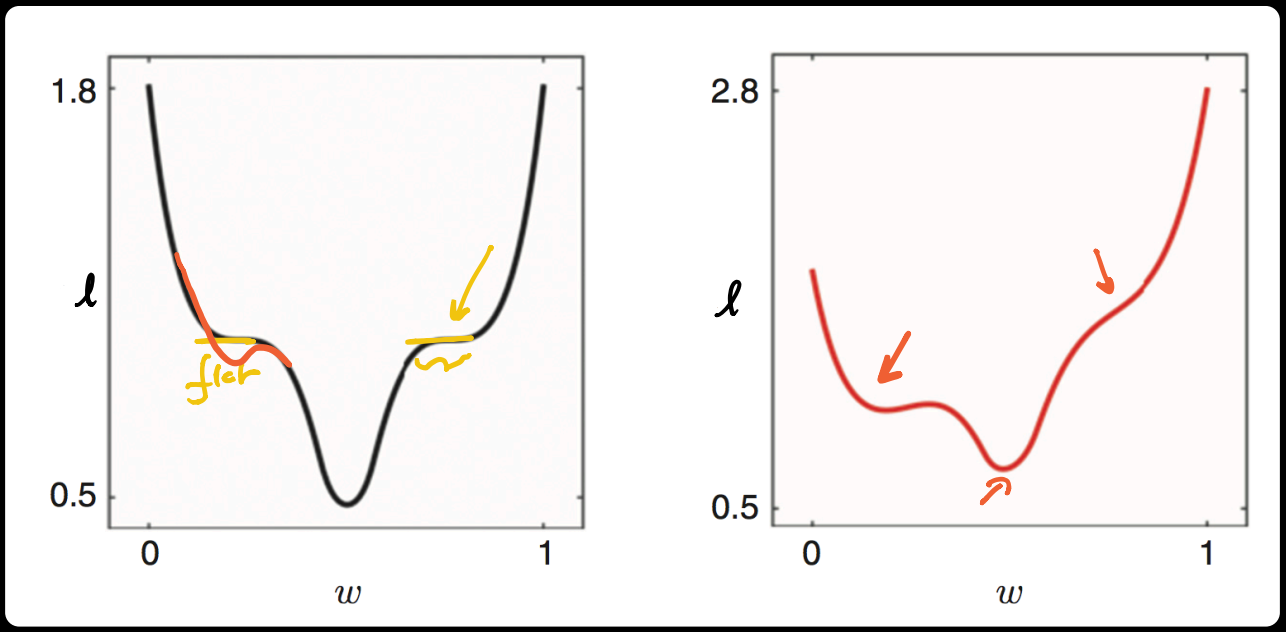
☹ initialization in the large flat orange region leads to a poor solution.

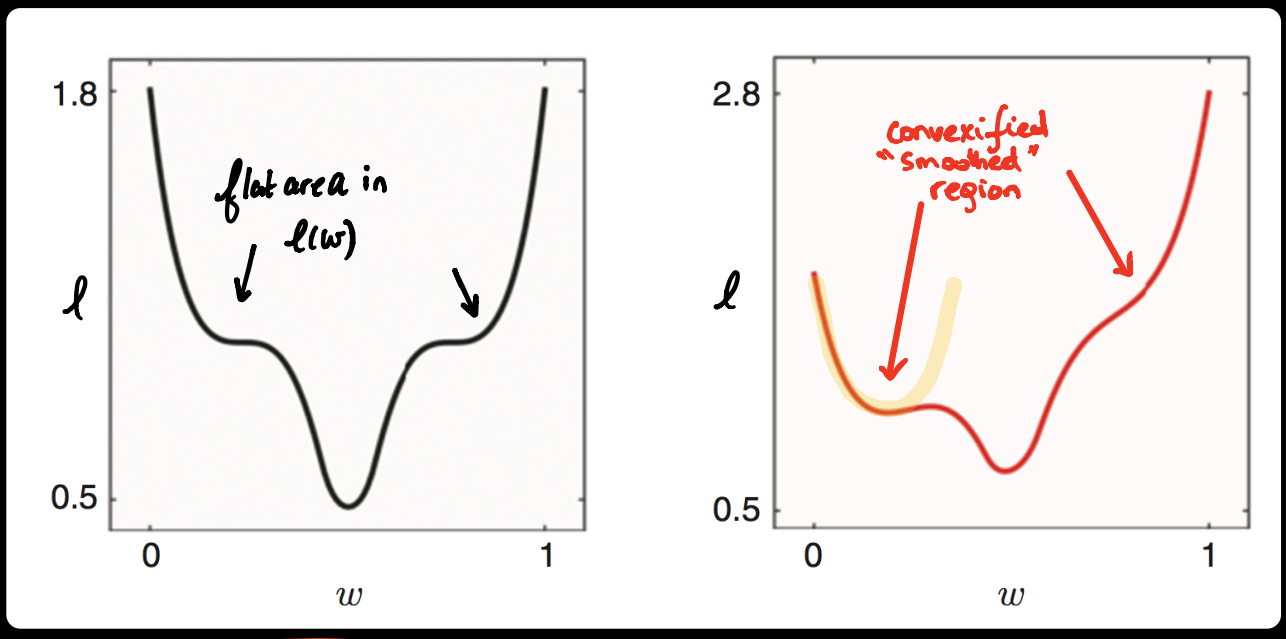
THINKING CAPS ON!

logistic




## 2D example






global

- $\ell(w, b)$ + $\lambda \| w \|_2^2$ } squared $l_2$ norm of the weights
  - $\ell(w, b)$: fit to data samples
  - $\lambda \| w \|_2^2$: $l_2$ regularizer of the loss function
- $\lambda > 0$
- $\| w \|_2^2 = \sum_{i=1}^{d} w_i^2$

- if $\lambda = 0$ ⇒ only loss ⇒ trade-off parameter (term 1, term 2)
- if $\lambda$ very large ⇒ the regularizer drowns out the loss function and we have

$$\underbrace{\ell(w,b)}_{0.01} + \underbrace{\lambda \| w \|_2^2}_{10^{10}\text{ too large}} \approx \underbrace{\lambda \| w \|_2^2}_{\text{"takes over"}}$$

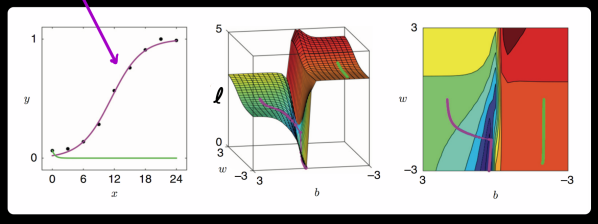
- Typically, $\lambda$ is set fairly small ( $\lambda = 0.1$ or smaller)

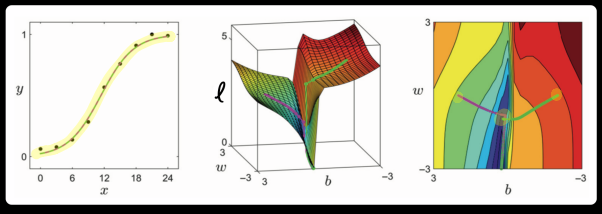
---

Sol° ⇒ Improving problematic flat areas using $l_2$ regularizer

$l_2 \| . \|_2$ ⇒ a simple convex function that is added to the non-convex function to smooth it and convexify it in problematic non-convex regions.

# * Logistic regression *



$$\ell(w,b) = \sum_{i=1}^{n} \left(\sigma(x^{i\top} w + b) - y^i\right)^2$$

$$\ell(w,b) = \sum_{i=1}^{n} \left(\sigma(x^{i\top} w + b) - y^i\right)^2 + \lambda \|w\|_2^2$$

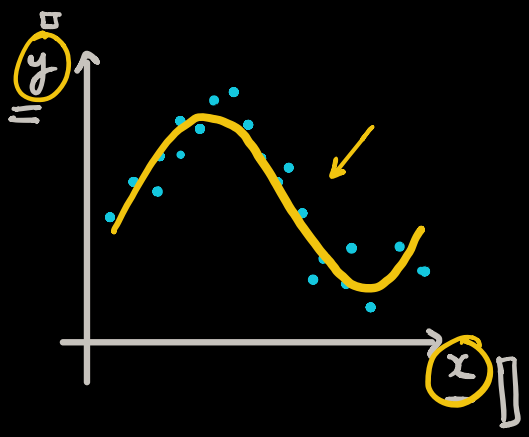
$$arg\ min_{\mathbf{w}} \mathcal{L}(\mathbf{w}) = \frac{1}{n} \sum_{(\mathbf{x}^i, y^i) \in \mathcal{D}} E(f_{\mathbf{w}}(\mathbf{x}^i, y^i)) + R(\dots)$$

Practice & code it up!
- exercise 3.11
- exercise 3.13

③ Nonlinear relationship between input features and output values

⇓

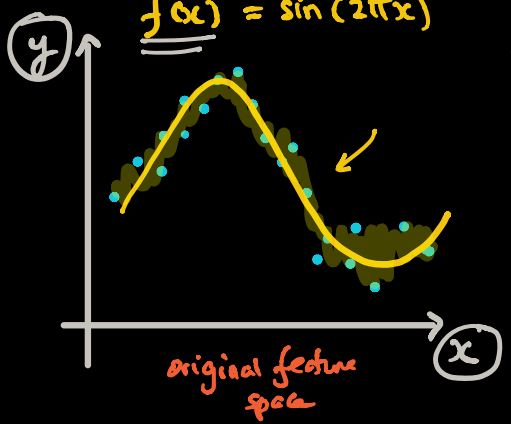
Knowledge-driven feature design for regression




$\{(x^i, y^i)\}_{i=1}^n$

(non-convex)
(non-linear)

$f(x) = \sin(2\pi x)$



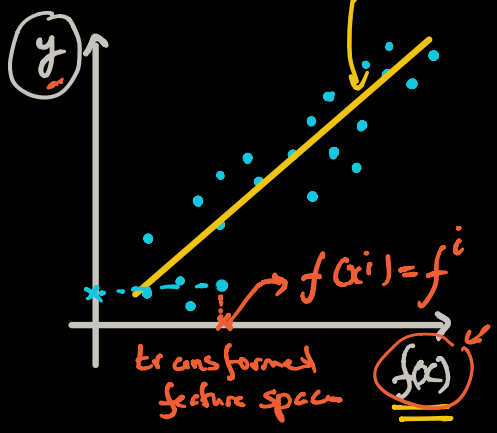

$\{(x^i, y^i)\}_{i=1}^n$

$y^i \simeq \sin(2\pi x^i)$

non linear

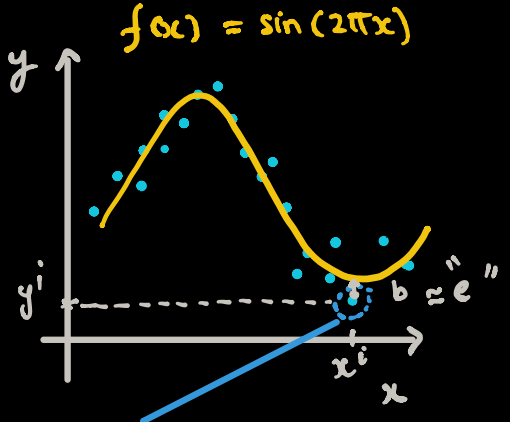
transformation $f$

$y = xw + b$




$\{(f^i, y^i)\}_{i=1}^n$

$y^i \simeq f^i w + b$




each sample can be viewed as a noisy observation drawn from $f$ distribution.

Knowledge-driven feature design

our "prior" knowledge about the data

⇓

This might originate from our prior understanding / assumption about the data distribution.

$x^T \in \mathbb{R}^a \rightarrow f(x^T) \in \mathbb{R}^q$

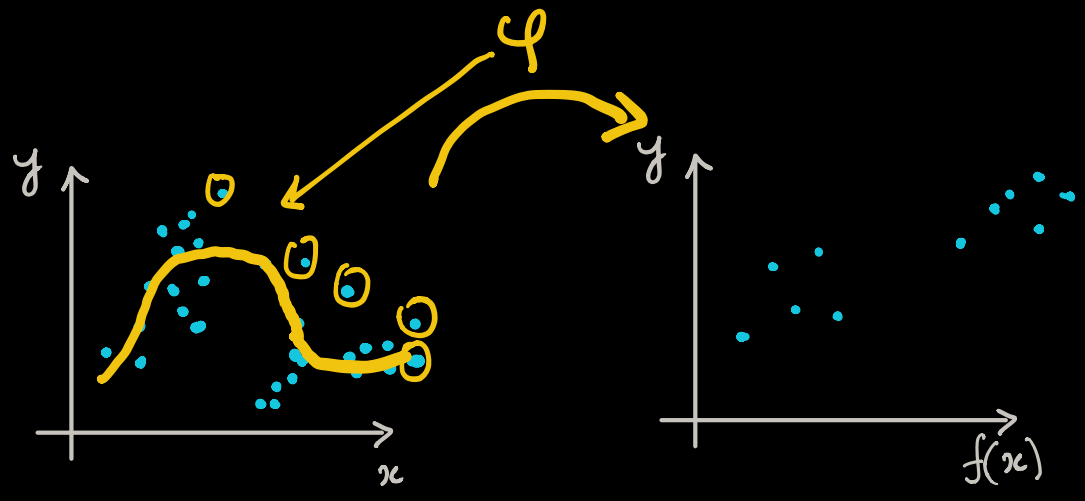
mapped / transform feature

$$\text{minimize}_{b,w} \sum_{i=1}^{n} \left(f^i w - y^i\right)^2$$

Linear in $w$ ☺

↓

" The trick based on our knowledge of feature distribution in the original space "




$f$ is also called "estimated data generating function"

(it is our estimated best fit to the dataset)

- Solve for $w, b$:

$$\underset{b,w}{\text{minimize}} \sum_{i=1}^{n} \left( f^i w + b - y^i \right)^2$$

- Compact notation:

$$\tilde{f}^i = \begin{bmatrix} 1 \\ f^i \end{bmatrix} \; ; \quad \tilde{w} = \begin{bmatrix} b \\ w \end{bmatrix}$$

$$\Rightarrow \quad \ell(\tilde{w}) = \sum_{i=1}^{n} \left( \tilde{f}^{iT} w - y^i \right)^2$$

$$\Rightarrow \quad \left[ \sum_{i=1}^{n} \tilde{f}^i \tilde{f}^{iT} \right] \tilde{w} = \sum_{i=1}^{n} \tilde{f}^i y^i$$

$$\boxed{\tilde{F} \tilde{F}^T \tilde{w} = \tilde{F} y}$$

- Recap

A properly designed feature (or set of features) for linear regression provides a good *nonlinear* fit in the original feature space and, simultaneously, a good *linear* fit in the transformed feature space.

☹ just because we can visualize a low-dimensional regression dataset does not mean we can easily design a proper feature transformation $f$ "by eye".

☹ it might not be obvious how to formulate a proper feature $f(x)$ to recover the original data generating function $y(x)$.

☹ when $x \in \mathbb{R}^d$, $d > 2$ $\Rightarrow$ it is even harder to "see"!

Example :