

# Foundation of Intelligent Systems, Part I

## Regression 2

[mcuturi@i.kyoto-u.ac.jp](mailto:mcuturi@i.kyoto-u.ac.jp)

# Some Words on the Survey

## What is your main goal in taking this class?

Please check one or two boxes.

- I know nothing about machine learning, so I just need an introduction
- I know a few machine learning algorithms, but I would like to have a better theoretical understanding
- I know a few machine learning algorithms, but I would like to learn about more advanced ones
- I would like to understand how to use machine learning algorithms for a particular application (for instance, vision, bioinformatics etc..)

Not enough answers to say anything meaningful!

- Try again: **survey**.

# Last Week

**Regression:** highlight a functional relationship between a **predicted variable** and **predictors**

# Last Week

**Regression:** highlight a functional relationship between a **predicted variable** and **predictors**

find a function  $f$  such that

$\forall(\mathbf{x}, \mathbf{y})$  that can appear ,  $f(\mathbf{x}) \approx \mathbf{y}$

# Last Week

**Regression:** highlight a functional relationship between a **predicted variable** and **predictors**

to find an accurate function  $f$  such that

$\forall(\mathbf{x}, \mathbf{y})$  that can appear ,  $f(\mathbf{x}) \approx \mathbf{y}$

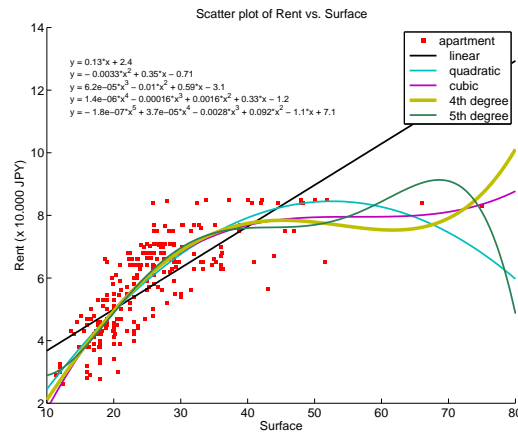
use a data set & the least-squares criterion:

$$\min_{f \in \mathcal{F}} \frac{1}{N} \sum_{j=1}^N (\mathbf{y}_j - f(\mathbf{x}_j))^2$$

# Last Week

**Regression:** highlight a functional relationship between a **predicted variable** and **predictors**

- when regressing a **real number** vs a **real number** :



- Least-Squares Criterion  $L(b, a_1, \dots, a_p)$  to fit **lines**, polynomials.
- results in solving a linear system.

$$\frac{\partial \mathbf{2}^{\text{nd}} \text{ order}(b, a_1, \dots, a_p)}{\partial a_p} = \mathbf{linear} \text{ in } (b, a_1, \dots, a_p)$$

- When setting  $\partial L / \partial a_p = 0$  we get  $p + 1$  **linear** equations for  $p + 1$  variables.

# Last Week

**Regression:** highlight a functional relationship between a **predicted variable** and **predictors**

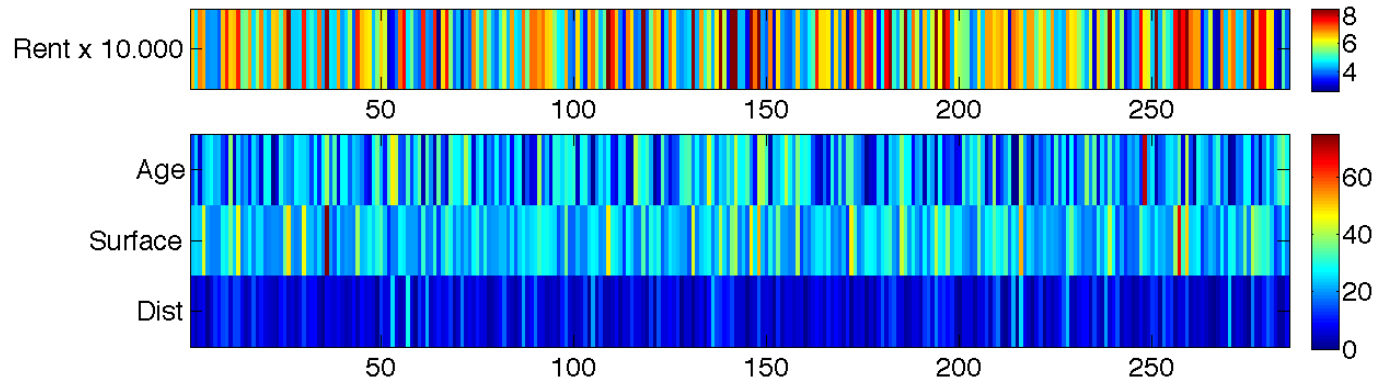
- when regressing a **real number** vs  $d$  **real numbers** (vector in  $\mathbb{R}^d$ ),
  - find best fit  $\alpha \in \mathbb{R}^n$  such that  $(\alpha^T \mathbf{x} + \alpha_0) \approx y$ .
  - Add to  $d \times N$  data matrix, a row of 1's to get the predictors  $\mathbf{X}$ .
  - The row  $\mathbf{Y}$  of **predicted** values
  - The Least-Squares criterion also applies:

$$L(\alpha) = \|\mathbf{Y} - \alpha^T \mathbf{X}\|^2 = \left( \alpha^T \mathbf{X} \mathbf{X}^T \alpha - 2 \mathbf{Y} \mathbf{X}^T \alpha + \|\mathbf{Y}\|^2 \right).$$

$$\nabla_{\alpha} L = 0 \quad \Rightarrow \quad \alpha^* = (\mathbf{X} \mathbf{X}^T)^{-1} \mathbf{X} \mathbf{Y}^T$$

- This works if  $\mathbf{X} \mathbf{X}^T \in \mathbb{R}^{d+1}$  is **invertible**.

# Last Week



```
>> (X*X') \ (X*Y')
```

```
ans =
```

```
-0.049332605603095    x age  
 0.163122792160298    x surface  
-0.004411580036614    x distance  
 2.731204399433800    + 27.300 JPY
```



# Today

- A **statistical / probabilistic** perspective on LS-regression
- A few words on **polynomials** in higher dimensions
- A **geometric** perspective
- **Variable co-linearity and Overfitting** problem
- Some solutions: **advanced regression techniques**
  - Subset selection
  - Ridge Regression
  - Lasso

---

**A (very few) words on the  
statistical/probabilistic interpretation of LS**

# The Statistical Perspective on Regression

- **Assume that** the values of  $y$  are stochastically linked to observations  $\mathbf{x}$  as

$$y - (\alpha^T \mathbf{x} + \beta) \sim \mathcal{N}(0, \sigma).$$

- This difference is a random variable called  $\varepsilon$  and is called a **residue**.

# The Statistical Perspective on Regression

- This can be rewritten as,

$$y = (\alpha^T \mathbf{x} + \beta) + \varepsilon, \quad \varepsilon \sim \mathcal{N}(0, \sigma),$$

- We **assume** that the difference between  $y$  and  $(\alpha^T \mathbf{x} + b)$  behaves like a **Gaussian** (normally distributed) random variable.

**Goal as a statistician:** Estimate  $\alpha$  and  $\beta$  given observations.

# Identically Independently Distributed (i.i.d) Observations

- Statistical hypothesis: **assume that the parameters are**  $\alpha = a, \beta = b$

# Identically Independently Distributed (i.i.d) Observations

- Statistical hypothesis: **assume that the parameters are**  $\alpha = \mathbf{a}, \beta = b$
- In such a case, what would be the **probability** of **each** observation  $(\mathbf{x}_j, y_j)$ ?

# Identically Independently Distributed (i.i.d) Observations

- Statistical hypothesis: **assuming that the parameters are**  $\alpha = \mathbf{a}, \beta = b$ , what would be the **probability** of **each** observation?
  - For each couple  $(\mathbf{x}_j, y_j)$ ,  $j = 1, \dots, N$ ,

$$P(\mathbf{x}_j, y_j \mid \alpha = \mathbf{a}, \beta = b) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{\|y_j - (\mathbf{a}^T \mathbf{x}_j + b)\|^2}{2\sigma^2}\right)$$

# Identically Independently Distributed (i.i.d) Observations

- Statistical hypothesis: **assuming that the parameters are**  $\alpha = \mathbf{a}, \beta = b$ , what would be the **probability** of **each** observation?:
  - For each couple  $(\mathbf{x}_j, y_j)$ ,  $j = 1, \dots, N$ ,

$$P(\mathbf{x}_j, y_j \mid \alpha = \mathbf{a}, \beta = b) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{\|y_j - (\mathbf{a}^T \mathbf{x}_j + b)\|^2}{2\sigma^2}\right)$$

- Since each measurement  $(\mathbf{x}_j, y_j)$  has been **independently sampled**,

$$P(\{(\mathbf{x}_j, y_j)\}_{j=1, \dots, N} \mid \alpha = \mathbf{a}, \beta = b) = \prod_{j=1}^N \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{\|y_j - (\mathbf{a}^T \mathbf{x}_j + b)\|^2}{2\sigma^2}\right)$$



# Identically Independently Distributed (i.i.d) Observations

- Statistical hypothesis: **assuming that the parameters are**  $\alpha = \mathbf{a}, \beta = b$ , what would be the **probability** of **each** observation?:
  - For each couple  $(\mathbf{x}_j, y_j)$ ,  $j = 1, \dots, N$ ,

$$P(\mathbf{x}_j, y_j \mid \alpha = \mathbf{a}, \beta = b) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{\|y_j - (\mathbf{a}^T \mathbf{x}_j + b)\|^2}{2\sigma^2}\right)$$

- Since each measurement  $(\mathbf{x}_j, y_j)$  has been **independently sampled**,

$$P(\{(\mathbf{x}_j, y_j)\}_{j=1, \dots, N} \mid \alpha = a, \beta = b) = \prod_{j=1}^N \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{\|y_j - (\mathbf{a}^T \mathbf{x}_j + b)\|^2}{2\sigma^2}\right)$$

- A.K.A **likelihood** of the dataset  $\{(\mathbf{x}_j, y_j)_{j=1, \dots, N}\}$  as a function of  $a$  and  $b$ ,

$$\mathcal{L}_{\{(\mathbf{x}_j, y_j)\}}(\mathbf{a}, b) = \prod_{j=1}^N \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{\|y_j - (\mathbf{a}^T \mathbf{x}_j + b)\|^2}{2\sigma^2}\right)$$

# Maximum Likelihood Estimation (MLE) of Parameters

Hence, for  $\mathbf{a}, b$ , the **likelihood** function on the dataset  $\{(\mathbf{x}_j, y_j)_{j=1, \dots, N}\} \dots$

$$\mathcal{L}(\mathbf{a}, b) = \prod_{j=1}^N \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{\|y_j - (\mathbf{a}^T \mathbf{x}_j + b)\|^2}{2\sigma^2}\right)$$

# Maximum Likelihood Estimation (MLE) of Parameters

Hence, for  $\mathbf{a}, b$ , the **likelihood** function on the dataset  $\{(\mathbf{x}_j, y_j)_{j=1, \dots, N}\} \dots$

$$\mathcal{L}(\mathbf{a}, b) = \prod_{j=1}^N \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{\|y_j - (\mathbf{a}^T \mathbf{x}_j + b)\|^2}{2\sigma^2}\right)$$

Why not use the **likelihood** to **guess**  $(\mathbf{a}, b)$  given data?

# Maximum Likelihood Estimation (MLE) of Parameters

Hence, for  $\mathbf{a}, b$ , the **likelihood** function on the dataset  $\{(\mathbf{x}_j, y_j)_{j=1, \dots, N}\} \dots$

$$\mathcal{L}(\mathbf{a}, b) = \prod_{j=1}^N \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{\|y_j - (\mathbf{a}^T \mathbf{x}_j + b)\|^2}{2\sigma^2}\right)$$

...the **MLE** approach selects the values of  $(\mathbf{a}, b)$  which **mazimize**  $\mathcal{L}(\mathbf{a}, b)$

# Maximum Likelihood Estimation (MLE) of Parameters

Hence, for  $\mathbf{a}, b$ , the **likelihood** function on the dataset  $\{(\mathbf{x}_j, y_j)_{j=1, \dots, N}\} \dots$

$$\mathcal{L}(\mathbf{a}, b) = \prod_{j=1}^N \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{\|y_j - (\mathbf{a}^T \mathbf{x}_j + b)\|^2}{2\sigma^2}\right)$$

...the **MLE** approach selects the values of  $(\mathbf{a}, b)$  which **mazimize**  $\mathcal{L}(\mathbf{a}, b)$

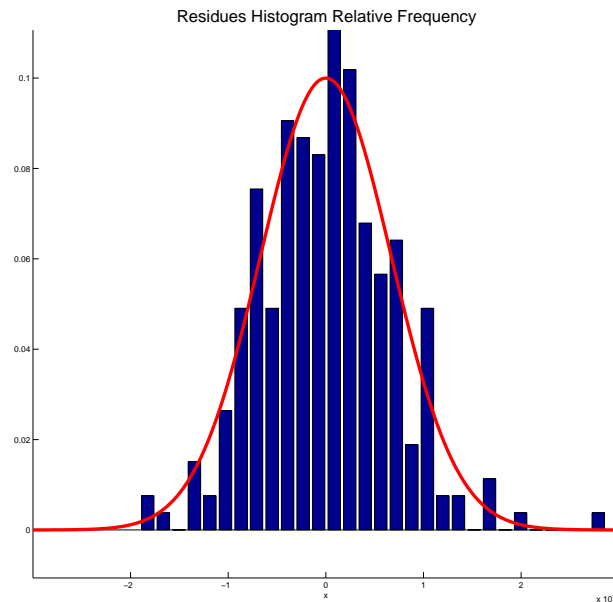
- **Since**  $\max_{(\mathbf{a}, b)} \mathcal{L}(\mathbf{a}, b) \Leftrightarrow \max_{(\mathbf{a}, b)} \log \mathcal{L}(\mathbf{a}, b)$

$$\log L(\mathbf{a}, b) = C - \frac{1}{2\sigma^2} \sum_{j=1}^N \|y_j - (\mathbf{a}^T \mathbf{x}_j + b)\|^2.$$

- **Hence**  $\max_{(\mathbf{a}, b)} \mathcal{L}(\mathbf{a}, b) \Leftrightarrow \min_{(\mathbf{a}, b)} \sum_{j=1}^N \|y_j - (\mathbf{a}^T \mathbf{x}_j + b)\|^2 \dots$

# Statistical Approach to Linear Regression

- Properties of the MLE estimator: convergence of  $\|\alpha - \mathbf{a}\|$ ?
- Confidence intervals for coefficients,
- Tests procedures to assess if model “fits” the data,



- Bayesian approaches: instead of looking for **one** optimal fit  $(\mathbf{a}, b)$  juggle with a whole density on  $(\mathbf{a}, b)$  to make decisions
- *etc.*

---

# **A few words on polynomials in higher dimensions**

# A few words on polynomials in higher dimensions

- For  $d$  variables, that is for points  $\mathbf{x} \in \mathbb{R}^d$ ,
  - the space of polynomials on these variables up to degree  $p$  is generated by

$$\{\mathbf{x}^{\mathbf{u}} \mid \mathbf{u} \in \mathbb{N}^d, \mathbf{u} = (u_1, \dots, u_d), \sum_{i=1}^d u_i \leq p\}$$

where the monomial  $\mathbf{x}^{\mathbf{u}}$  is defined as  $x_1^{u_1} x_2^{u_2} \dots x_d^{u_d}$

- Recurrence for dimension of that space:  $\dim_{p+1} = \dim_p + \binom{p+1}{d+p}$
- For  $d = 20$  and  $p = 5$ ,  $1 + 20 + 210 + 1540 + 8855 + 42504 > 50.000$

Problem with polynomial interpolation in **high-dimensions** is the **explosion** of relevant variables (one for each monomial)



---

# Geometric Perspective

# Back to Basics

- Recall the problem:

$$X = \begin{bmatrix} 1 & 1 & \cdots & 1 \\ \vdots & \vdots & \cdots & \vdots \\ \mathbf{x}_1 & \mathbf{x}_2 & \cdots & \mathbf{x}_N \\ \vdots & \vdots & \cdots & \vdots \end{bmatrix} \in \mathbb{R}^{d+1 \times N}$$

and

$$Y = [y_1 \quad \cdots \quad y_N] \in \mathbb{R}^N.$$

- We look for  $\alpha$  such that  $\alpha^T X \approx Y$ .

## Back to Basics

- If we transpose this expression we get  $X^T \alpha \approx Y^T$ ,

$$\begin{bmatrix} 1 & x_{1,1} & \cdots & x_{d,1} \\ 1 & x_{1,2} & \cdots & x_{d,2} \\ \vdots & \vdots & \vdots & \vdots \\ 1 & x_{1,k} & \cdots & x_{d,k} \\ \vdots & \vdots & \vdots & \vdots \\ 1 & x_{1,N} & \cdots & x_{d,N} \end{bmatrix} \times \begin{bmatrix} \alpha_0 \\ \vdots \\ \alpha_d \end{bmatrix} = \begin{bmatrix} y_1 \\ \vdots \\ y_2 \\ \vdots \\ y. \\ \vdots \\ y_N \end{bmatrix}$$

- Using the notation  $\mathbf{Y} = Y^T$ ,  $\mathbf{X} = X^T$  and  $\mathbf{X}_k$  for the  $(k + 1)^{\text{th}}$  column of  $\mathbf{X}$ ,

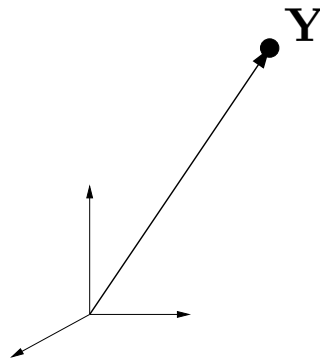
$$\sum_{k=0}^d \alpha_k \mathbf{X}_k \approx \mathbf{Y}$$

- Note how the  $\mathbf{X}_k$  corresponds to **all** values taken by the  $k^{\text{th}}$  variable.
- **Problem:** approximate/reconstruct Reconstructing  $\mathbf{Y} \in \mathbb{R}^N$  using  $\mathbf{X}_0, \mathbf{X}_1, \cdots, \mathbf{X}_d \in \mathbb{R}^N$ ?

# Linear System

Reconstructing  $\mathbf{Y} \in \mathbb{R}^N$  using  $\mathbf{X}_0, \mathbf{X}_1, \dots, \mathbf{X}_d$  vectors of  $\mathbb{R}^N$ .

- Our ability to approximate  $\mathbf{Y}$  depends implicitly on the space spanned by  $\mathbf{X}_0, \mathbf{X}_1, \dots, \mathbf{X}_d$

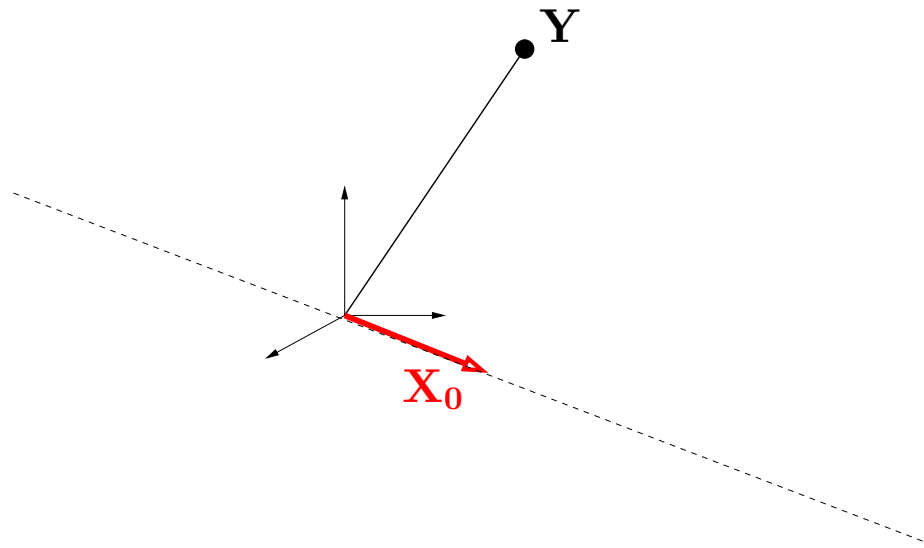


Consider the observed vector in  $\mathbb{R}^N$  of predicted values

# Linear System

Reconstructing  $\mathbf{Y} \in \mathbb{R}^N$  using  $\mathbf{X}_0, \mathbf{X}_1, \dots, \mathbf{X}_d$  vectors of  $\mathbb{R}^N$ .

- Our ability to approximate  $\mathbf{Y}$  depends implicitly on the space spanned by  $\mathbf{X}_0, \mathbf{X}_1, \dots, \mathbf{X}_d$

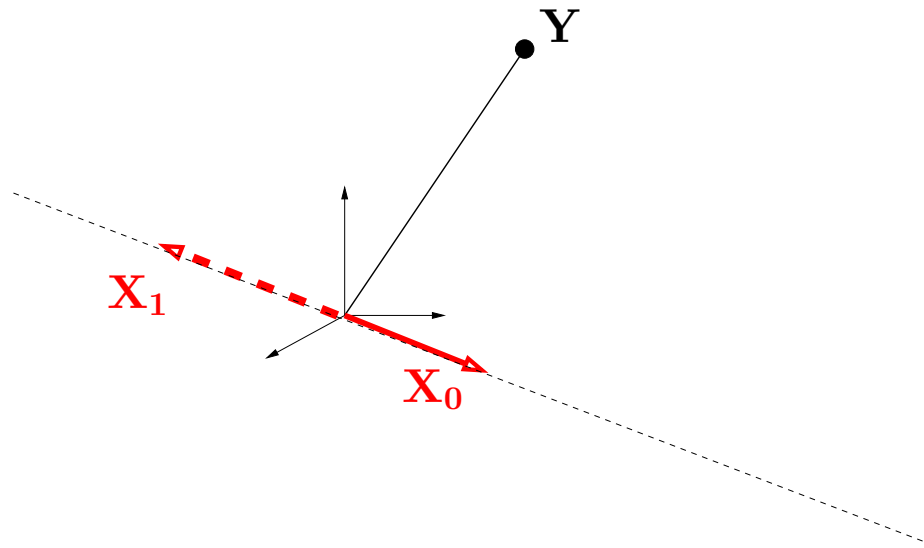


Plot the first regressor  $\mathbf{X}_0$ ...

# Linear System

Reconstructing  $\mathbf{Y} \in \mathbb{R}^N$  using  $\mathbf{X}_0, \mathbf{X}_1, \dots, \mathbf{X}_d$  vectors of  $\mathbb{R}^N$ .

- Our ability to approximate  $\mathbf{Y}$  depends implicitly on the space spanned by  $\mathbf{X}_0, \mathbf{X}_1, \dots, \mathbf{X}_d$

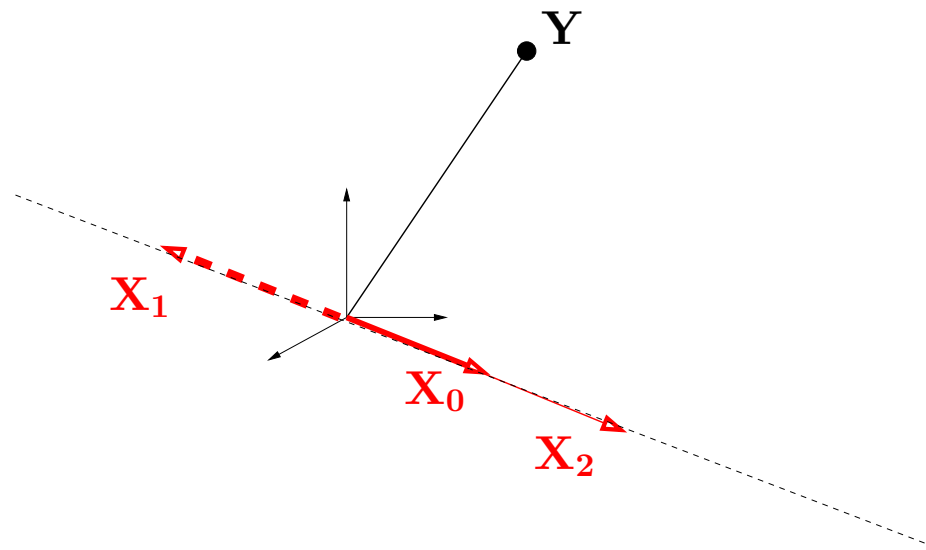


Assume the next regressor  $\mathbf{X}_1$  is colinear to  $\mathbf{X}_0$ ...

# Linear System

Reconstructing  $\mathbf{Y} \in \mathbb{R}^N$  using  $\mathbf{X}_0, \mathbf{X}_1, \dots, \mathbf{X}_d$  vectors of  $\mathbb{R}^N$ .

- Our ability to approximate  $\mathbf{Y}$  depends implicitly on the space spanned by  $\mathbf{X}_0, \mathbf{X}_1, \dots, \mathbf{X}_d$

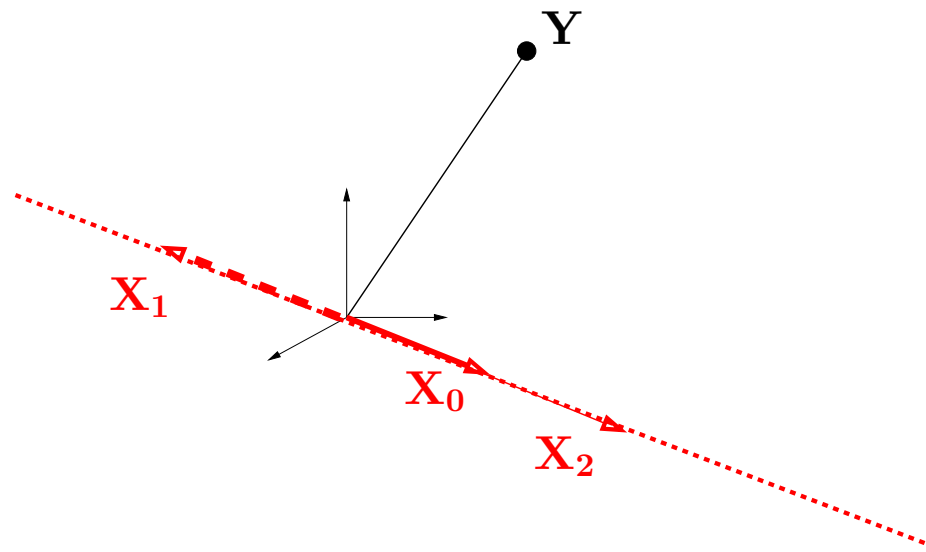


and so is  $\mathbf{X}_2 \dots$

# Linear System

Reconstructing  $\mathbf{Y} \in \mathbb{R}^N$  using  $\mathbf{X}_0, \mathbf{X}_1, \dots, \mathbf{X}_d$  vectors of  $\mathbb{R}^N$ .

- Our ability to approximate  $\mathbf{Y}$  depends implicitly on the space spanned by  $\mathbf{X}_0, \mathbf{X}_1, \dots, \mathbf{X}_d$



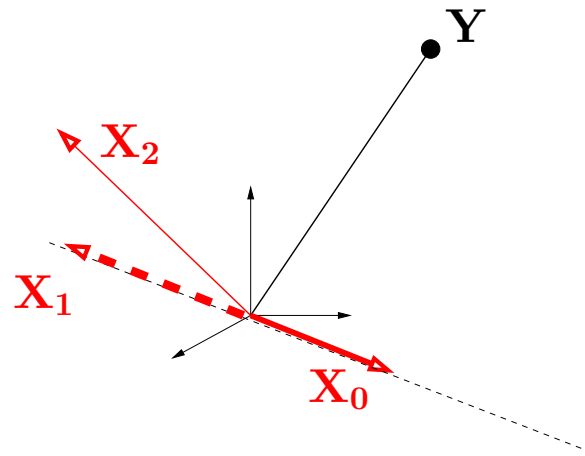
Very little choices to approximate  $\mathbf{Y}$ ...



# Linear System

Reconstructing  $\mathbf{Y} \in \mathbb{R}^N$  using  $\mathbf{X}_0, \mathbf{X}_1, \dots, \mathbf{X}_d$  vectors of  $\mathbb{R}^N$ .

- Our ability to approximate  $\mathbf{Y}$  depends implicitly on the space spanned by  $\mathbf{X}_0, \mathbf{X}_1, \dots, \mathbf{X}_d$

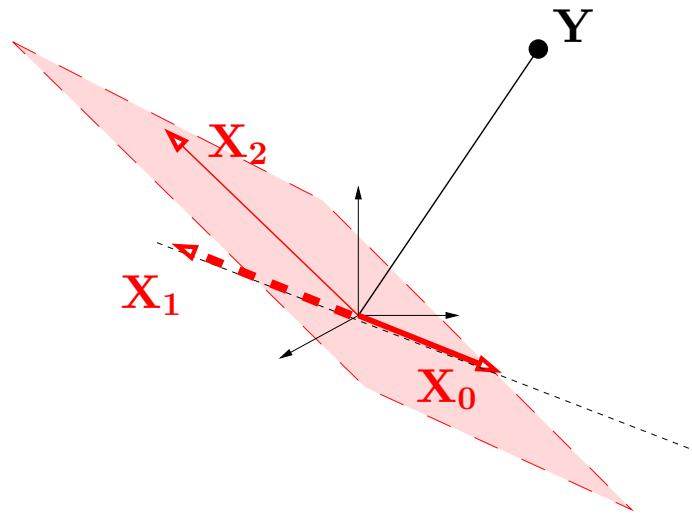


Suppose  $\mathbf{X}_2$  is actually not colinear to  $\mathbf{X}_0$ .

# Linear System

Reconstructing  $\mathbf{Y} \in \mathbb{R}^N$  using  $\mathbf{X}_0, \mathbf{X}_1, \dots, \mathbf{X}_d$  vectors of  $\mathbb{R}^N$ .

- Our ability to approximate  $\mathbf{Y}$  depends implicitly on the space spanned by  $\mathbf{X}_0, \mathbf{X}_1, \dots, \mathbf{X}_d$

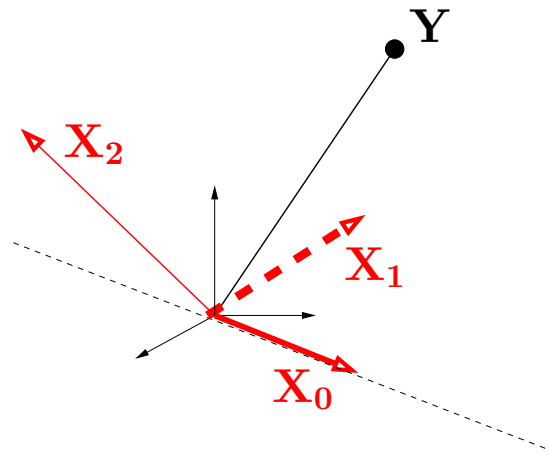


This opens new ways to reconstruct  $\mathbf{Y}$ .

# Linear System

Reconstructing  $\mathbf{Y} \in \mathbb{R}^N$  using  $\mathbf{X}_0, \mathbf{X}_1, \dots, \mathbf{X}_d$  vectors of  $\mathbb{R}^N$ .

- Our ability to approximate  $\mathbf{Y}$  depends implicitly on the space spanned by  $\mathbf{X}_0, \mathbf{X}_1, \dots, \mathbf{X}_d$

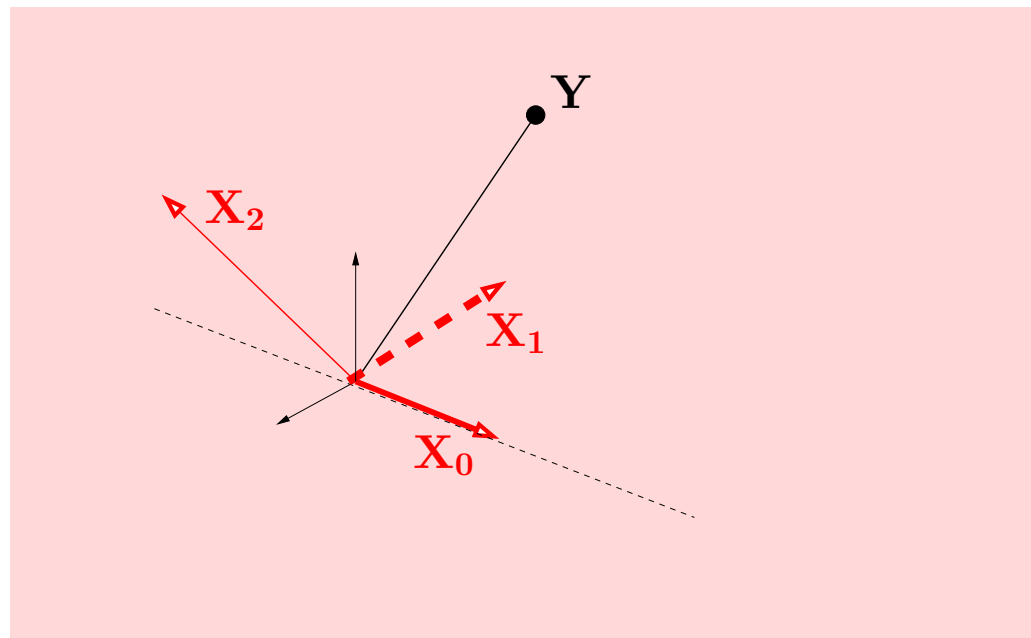


When  $\mathbf{X}_0, \mathbf{X}_1, \mathbf{X}_2$  are linearly independent,

# Linear System

Reconstructing  $\mathbf{Y} \in \mathbb{R}^N$  using  $\mathbf{X}_0, \mathbf{X}_1, \dots, \mathbf{X}_d$  vectors of  $\mathbb{R}^N$ .

- Our ability to approximate  $\mathbf{Y}$  depends implicitly on the space spanned by  $\mathbf{X}_0, \mathbf{X}_1, \dots, \mathbf{X}_d$



$\mathbf{Y}$  is in their span since the space is of dimension 3

# Linear System

Reconstructing  $\mathbf{Y} \in \mathbb{R}^N$  using  $\mathbf{X}_0, \mathbf{X}_1, \dots, \mathbf{X}_d$  vectors of  $\mathbb{R}^N$ .

- Our ability to approximate  $\mathbf{Y}$  depends implicitly on the space spanned by  $\mathbf{X}_0, \mathbf{X}_1, \dots, \mathbf{X}_d$

The dimension of that space is **Rank( $\mathbf{X}$ )**, the rank of  $\mathbf{X}$

$$\mathbf{Rank}(\mathbf{X}) \leq \min(d + 1, N).$$

# Linear System

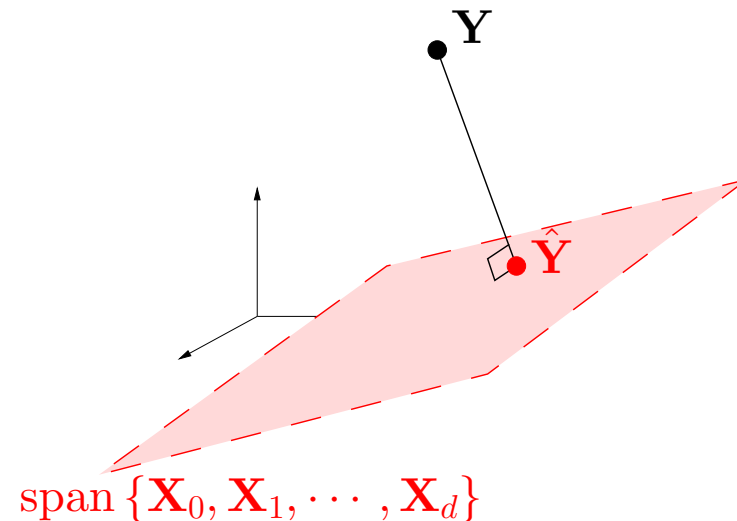
Three cases depending on **Rank X** and  $d, N$

1. **Rank X**  $< N$ .  $d + 1$  **column vectors do not span**  $\mathbb{R}^N$ 
  - For arbitrary  $Y$ , there is **no solution** to  $\alpha^T X = Y$
2. **Rank X**  $= N$  and  $d + 1 > N$ , **too many variables span the whole of**  $\mathbb{R}^N$ 
  - **infinite** number of solutions to  $\alpha^T X = Y$ .
3. **Rank X**  $= N$  and  $d + 1 = N$ , **# variables = # observations**
  - Exact and unique solution:  $\alpha = \mathbf{X}^{-1}\mathbf{Y}$  we have  $\alpha^T X = Y$

In most applications,  $d + 1 \neq N$  so we are either in case 1 or 2

## Case 1: Rank $\mathbf{X} < N$

- **no solution** to  $\alpha^T \mathbf{X} = \mathbf{Y}$  (equivalently  $\mathbf{X}\alpha = \mathbf{Y}$ ) in general case.
- What about the **orthogonal projection** of  $\mathbf{Y}$  on the **image** of  $\mathbf{X}$



- Namely the point  $\hat{\mathbf{Y}}$  such that

$$\hat{\mathbf{Y}} = \underset{\mathbf{u} \in \text{span}\{\mathbf{X}_0, \mathbf{X}_1, \dots, \mathbf{X}_d\}}{\text{argmin}} \|\mathbf{Y} - \mathbf{u}\|.$$

## Case 1: Rank $\mathbf{X} < N$

**Lemma 1.**  $\{\mathbf{X}_0, \mathbf{X}_1, \dots, \mathbf{X}_d\}$  is a **l.i.** family  $\Leftrightarrow \mathbf{X}^T \mathbf{X}$  is invertible



## Case 1: Rank $\mathbf{X} < N$

- Computing the **projection**  $\hat{\omega}$  of a point  $\omega$  on a **subspace**  $V$  is well understood.
- In particular, if  $(\mathbf{X}_0, \mathbf{X}_1, \dots, \mathbf{X}_d)$  is a **basis** of  $\text{span}\{\mathbf{X}_0, \mathbf{X}_1, \dots, \mathbf{X}_d\}$ ...

(that is  $\{\mathbf{X}_0, \mathbf{X}_1, \dots, \mathbf{X}_d\}$  is a **linearly independent** family)

... then  $(\mathbf{X}^T \mathbf{X})$  is invertible and ...

$$\hat{\mathbf{Y}} = \mathbf{X}(\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{Y}$$

- This gives us the  $\alpha$  vector of weights we are looking for:

$$\hat{\mathbf{Y}} = \mathbf{X} \underbrace{(\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{Y}}_{\hat{\alpha}} = \mathbf{X} \hat{\alpha} \approx \mathbf{Y} \text{ or } \hat{\alpha}^T \mathbf{X} = \mathbf{Y}$$

- What can go wrong?

## Case 1: Rank $\mathbf{X} < N$

- If  $\mathbf{X}^T \mathbf{X}$  is invertible,

$$\hat{\mathbf{Y}} = \mathbf{X}(\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{Y}$$

- If  $\mathbf{X}^T \mathbf{X}$  is not invertible... we have a problem.

- If  $\mathbf{X}^T \mathbf{X}$ 's condition number

$$\frac{\lambda_{\max}(\mathbf{X}^T \mathbf{X})}{\lambda_{\min}(\mathbf{X}^T \mathbf{X})},$$

is very large, a small change in  $\mathbf{Y}$  can cause dramatic changes in  $\alpha$ .

- In this case the linear system is said to be **badly conditioned**...

- Using the formula

$$\hat{\mathbf{Y}} = \mathbf{X}(\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{Y}$$

might return garbage as can be seen in the following Matlab example.

## Case 2: Rank $\mathbf{X} = N$ and $d + 1 > N$

### high-dimensional low-sample setting

- **Ill-posed inverse problem**, the set

$$\{\alpha \in \mathbb{R}^d \mid \mathbf{X}\alpha = \mathbf{Y}\}$$

is a whole **vector space**. We need to choose **one** from **many admissible** points.

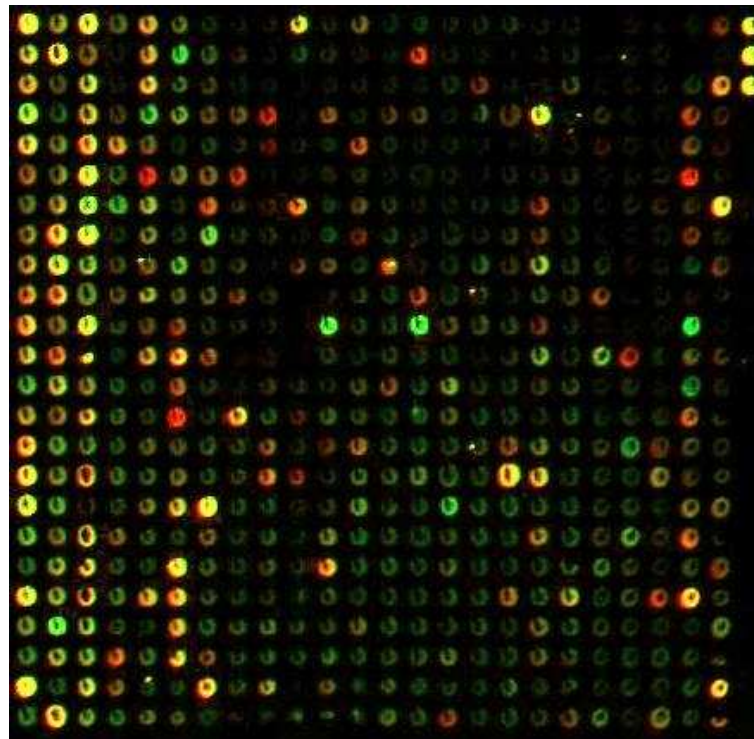
- When does this happen?
  - High-dimensional low-sample case (DNA chips, multimedia *etc.*)
- How to solve for this?
  - Use something called regularization.

---

# **A practical perspective: Colinearity and Overfitting**

# A Few High-dimensions Low sample settings

- DNA chips are very long vectors of measurements, one for each gene



- Task: regress a health-related variable against gene expression levels

Image:<http://bioinfo.cs.technion.ac.il/projects/Kahana-Navon/DNA-chips.htm>



# Correlated Variables

- Suppose you run a real-estate company.



- For each apartment you have compiled a **few hundred** predictor variables, *e.g.*
  - distances to conv. store, pharmacy, supermarket, parking lot, *etc.*
  - distances to all main locations in Kansai
  - socio-economic variables of the neighborhood
  - characteristics of the apartment
- Some are obviously **correlated** (correlated= “almost” colinear)
  - distance to Post Office / distance to Post ATM
- In that case, we may have some problems (Matlab example)

Source: <http://realestate.yahoo.co.jp/>

# Overfitting

- Given  $d$  variables (including constant variable), consider the least squares criterion

$$L_d(\alpha_1, \dots, \alpha_d) = \sum_{j=1}^j \left\| y_j - \sum_{i=1}^d \alpha_i x_{i,j} \right\|^2$$

- Add **any** variable vector  $\mathbf{x}_{d+1,j}, j = 1, \dots, N$ , and define

$$L_{d+1}(\alpha_1, \dots, \alpha_d, \boldsymbol{\alpha}_{d+1}) = \sum_{j=1}^j \left\| y_j - \sum_{i=1}^d \alpha_i x_{i,j} - \boldsymbol{\alpha}_{d+1} \mathbf{x}_{d+1,j} \right\|^2$$



# Overfitting

- Given  $d$  variables (including constant variable), consider the least squares criterion

$$L_d(\alpha_1, \dots, \alpha_d) = \sum_{j=1}^j \left\| y_j - \sum_{i=1}^d \alpha_i x_{i,j} \right\|^2$$

- Add **any** variable vector  $\mathbf{x}_{d+1,j}, j = 1, \dots, N$ , and define

$$L_{d+1}(\alpha_1, \dots, \alpha_d, \boldsymbol{\alpha}_{d+1}) = \sum_{j=1}^j \left\| y_j - \sum_{i=1}^d \alpha_i x_{i,j} - \boldsymbol{\alpha}_{d+1} \mathbf{x}_{d+1,j} \right\|^2$$

**THEN**  $\min_{\alpha \in \mathbb{R}^{d+1}} L_{d+1}(\alpha) \leq \min_{\alpha \in \mathbb{R}^d} L_d(\alpha)$

# Overfitting

- Given  $d$  variables (including constant variable), consider the least squares criterion

$$L_d(\alpha_1, \dots, \alpha_d) = \sum_{j=1}^j \left\| y_j - \sum_{i=1}^d \alpha_i x_{i,j} \right\|^2$$

- Add **any** variable vector  $\mathbf{x}_{d+1,j}, j = 1, \dots, N$ , and define

$$L_{d+1}(\alpha_1, \dots, \alpha_d, \boldsymbol{\alpha}_{d+1}) = \sum_{j=1}^j \left\| y_j - \sum_{i=1}^d \alpha_i x_{i,j} - \boldsymbol{\alpha}_{d+1} \mathbf{x}_{d+1,j} \right\|^2$$

Then  $\min_{\alpha \in \mathbb{R}^{d+1}} L_{d+1}(\alpha) \leq \min_{\alpha \in \mathbb{R}^d} L_d(\alpha)$

why?  $L_d(\alpha_1, \dots, \alpha_d) = L_{d+1}(\alpha_1, \dots, \alpha_d, \mathbf{0})$

# Overfitting

- Given  $d$  variables (including constant variable), consider the least squares criterion

$$L_d(\alpha_1, \dots, \alpha_d) = \sum_{j=1}^j \left\| y_j - \sum_{i=1}^d \alpha_i x_{i,j} \right\|^2$$

- Add **any** variable vector  $\mathbf{x}_{d+1,j}, j = 1, \dots, N$ , and define

$$L_{d+1}(\alpha_1, \dots, \alpha_d, \boldsymbol{\alpha}_{d+1}) = \sum_{j=1}^j \left\| y_j - \sum_{i=1}^d \alpha_i x_{i,j} - \boldsymbol{\alpha}_{d+1} \mathbf{x}_{d+1,j} \right\|^2$$

Then  $\min_{\alpha \in \mathbb{R}^{d+1}} L_{d+1}(\alpha) \leq \min_{\alpha \in \mathbb{R}^d} L_d(\alpha)$

why?  $L_d(\alpha_1, \dots, \alpha_d) = L_{d+1}(\alpha_1, \dots, \alpha_d, \mathbf{0})$

**Residual-sum-of-squares** goes down... but is it **relevant** to add variables?

# Occam's razor formalization of overfitting

Minimizing least-squares (RSS) is **not clever enough**.  
We need **another idea** to avoid **overfitting**.

- **Occam's razor:** *lex parsimoniae*



- **law of parsimony:** principle that recommends selecting the hypothesis that makes the fewest assumptions.

*one should always opt for an explanation in terms of the fewest possible causes, factors, or variables.*

Wikipedia: William of Ockham, born 1287- died 1347

---

# Advanced Regression Techniques

# Quick Reminder on Vector Norms

- For a vector  $\mathbf{a} \in \mathbb{R}^d$ , the Euclidian norm is the quantity

$$\|\mathbf{a}\|_2 = \sqrt{\sum_{i=1}^d a_i^2}.$$

- More generally, the  $q$ -norm is for  $q > 0$ ,

$$\|\mathbf{a}\|_q = \left( \sum_{i=1}^d |a_i|^q \right)^{\frac{1}{q}}.$$

- In particular for  $q = 1$ ,

$$\|\mathbf{a}\|_1 = \sum_{i=1}^d |a_i|$$

- In the limit  $q \rightarrow \infty$  and  $q \rightarrow 0$ ,

$$\|\mathbf{a}\|_\infty = \max_{i=1, \dots, d} |a_i|. \quad \|\mathbf{a}\|_0 = \#\{i | a_i \neq 0\}.$$

# Tikhonov Regularization '43 - Ridge Regression '62

- Tikhonov's motivation : solve **ill-posed inverse problems** by **regularization**
- If  $\min_{\alpha} L(\alpha)$  is achieved on many points... consider

$$\min_{\alpha} L(\alpha) + \lambda \|\alpha\|_2^2$$

- We can show that this leads to selecting

$$\hat{\alpha} = (\mathbf{X}^T \mathbf{X} + \lambda \mathbf{I}_{d+1})^{-1} \mathbf{X} \mathbf{Y}$$

- The condition number has changed to

$$\frac{\lambda_{\max}(\mathbf{X}^T \mathbf{X}) + \lambda}{\lambda_{\min}(\mathbf{X}^T \mathbf{X}) + \lambda}$$

# Subset selection : Exhaustive Search

- Following Ockham's razor, ideally we would like to know for any value  $p$

$$\min_{\alpha, \|\alpha\|_0=p} L(\alpha)$$

- → select the **best** vector  $\alpha$  which **only** gives weights to  $p$  **variables**.
- → Find the **best** combination of  $p$  variables.

## Practical Implementation

- For  $p \leq n$ ,  $\binom{n}{p}$  possible combinations of  $p$  variables.
- Brute force approach: generate  $\binom{n}{p}$  regression problems and select the one that achieves the best RSS.

**Impossible in practice** with moderately large  $n$  and  $p \dots \binom{30}{5} = 150.000$



# Subset selection : Forward Search

Since the **exact** search is **intractable in practice**, consider the **forward** heuristic

- **In Forward search:**

- define  $I_1 = \{0\}$ .
- given a set  $I_k \subset \{0, \dots, d\}$  of  $k$  variables, **what is the most informative variable one could add?**
  - ▷ Compute for each variable  $i$  in  $\{0, \dots, d\} \setminus I_k$

$$t_i = \min_{(\alpha_k)_{k \in I_k}, \alpha} \sum_{j=1}^N \left\| y_j - \left( \sum_{k \in I_k} \alpha_k x_{k,j} + \alpha x_{i,j} \right) \right\|^2$$

- ▷ Set  $I_{k+1} = I_k \cup \{i^*\}$  for any  $i^*$  such that  $i^* = \min t_i$ .
- ▷  $k = k + 1$  until desired number of variables

# Subset selection : Backward Search

... or the **backward** heuristic

- In Backward search:

- define  $I_d = \{0, 1, \dots, n\}$ .
- given a set  $I_k \subset \{0, \dots, d\}$  of  $k$  variables, what is the **least informative variable one could remove**?
  - ▷ Compute for each variable  $i$  in  $I_k$

$$t_i = \min_{(\alpha_k)_{k \in I_k \setminus \{i\}}} \sum_{j=1}^N \left\| y_j - \left( \sum_{k \in I_k \setminus \{i\}} \alpha_k x_{k,j} \right) \right\|^2$$

- ▷ Set  $I_{k-1} = I_k \setminus \{i^*\}$  for any  $i^*$  such that  $i^* = \mathbf{max} t_i$ .
- ▷  $k = k - 1$  until desired number of variables

# Subset selection : LASSO

Naive Least-squares

$$\min_{\alpha} L(\alpha)$$

Best fit with  $p$  variables (Occam!)

$$\min_{\alpha, \|\alpha\|_0=p} L(\alpha)$$

Tikhonov regularized Least-squares

$$\min_{\alpha} L(\alpha) + \lambda \|\alpha\|_2^2$$

**LASSO** (least absolute shrinkage and selection operator)

$$\min_{\alpha} L(\alpha) + \lambda \|\alpha\|_1$$