

Statistical Machine Learning, Part I

Regression

mcuturi@i.kyoto-u.ac.jp

Fundamentals in Regression

- Can be studied from different viewpoints: statistical, linear algebra, AI... *etc..*
- Linear regression is currently revived by different ideas in **sparsity**
 - Lasso (1996→)
 - SVM for regression (1996→)
 - Compressed Sensing (2002→)

One of the most standard data analysis tasks: Regression

Data: many observations of the same data type

- We have a database $\{\mathbf{x}_1, \dots, \mathbf{x}_N\}$.

- Each datapoint \mathbf{x}_j can be encoded as a vector of features $\mathbf{x}_j = \begin{bmatrix} x_{1,j} \\ x_{2,j} \\ \vdots \\ x_{d,j} \end{bmatrix}$

- Each feature $x_{i,j}$ ($1 \leq i \leq d$) of a given point \mathbf{x}_j is a number.

One of the most standard data analysis tasks: Regression

This database can be seen as a $\mathbb{R}^{d \times N}$ **matrix**

$$\{\mathbf{x}^1, \dots, \mathbf{x}^N\} \iff \begin{bmatrix} \mathbf{x}_{1,1} & \mathbf{x}_{1,2} & \cdots & \mathbf{x}_{1,N} \\ \mathbf{x}_{2,1} & \mathbf{x}_{2,2} & \cdots & \mathbf{x}_{2,N} \\ \mathbf{x}_{3,1} & \mathbf{x}_{3,2} & \cdots & \mathbf{x}_{3,N} \\ \vdots & \vdots & \cdots & \vdots \\ \mathbf{x}_{d,1} & \mathbf{x}_{d,2} & \cdots & \mathbf{x}_{d,N} \end{bmatrix}$$

Examples



$$\text{Credit card holder } \mathbf{x}_j = \begin{bmatrix} \text{Income} \\ \text{Age} \\ \vdots \\ \text{Work history (months)} \\ \text{Family} \\ \# \text{ Credit Incidents} \end{bmatrix}$$



$$\text{Patient } \mathbf{x}_j = \begin{bmatrix} \text{height} \\ \text{weight} \\ \vdots \\ \# \text{ minutes exercise/week} \\ \text{LDL cholesterol} \\ \text{HDL cholesterol} \end{bmatrix}$$



$$\text{Blog } \mathbf{x}_j = \begin{bmatrix} \text{avg. pages view/month} \\ \# \text{ posts} \\ \vdots \\ \text{avg. } \# \text{ comments/month} \\ \text{revenue from ads/month} \end{bmatrix}$$

Within such variables...

- Some variables are very **cheap to measure**, others **very expensive**.
- Some variables might have a **causal** effect on other variables.

- In the regression setting, the d variables are split between
 - k **regressor** (or **predictor**) variables
 - $d - k$ **response** (or **predicted**) variables.

Regression

- In the regression setting, the d variables are split between
 - k **regressor** (or **predictor**) variables
 - $d - k$ **response** (or **predicted**) variables.

guess **response**(**expensive**) variables using **regressor**(**cheap**) ones.

The Regression Problem

- Given,

- A database $\{\mathbf{x}_1, \dots, \mathbf{x}_N\} \iff X = \begin{bmatrix} x_{1,1} & x_{1,2} & \cdots & x_{1,N} \\ x_{2,1} & x_{2,2} & \cdots & x_{2,N} \\ x_{3,1} & x_{3,2} & \cdots & x_{3,N} \\ \vdots & \vdots & & \vdots \\ x_{d,1} & x_{d,2} & \cdots & x_{d,N} \end{bmatrix}$

The Regression Problem

- Given,

- A database $\{\mathbf{x}_1, \dots, \mathbf{x}_N\} \iff X = \begin{bmatrix} \mathbf{x}_{1,1} & \mathbf{x}_{1,2} & \cdots & \mathbf{x}_{1,N} \\ \mathbf{x}_{2,1} & \mathbf{x}_{2,2} & \cdots & \mathbf{x}_{2,N} \\ \mathbf{x}_{3,1} & \mathbf{x}_{3,2} & \cdots & \mathbf{x}_{3,N} \\ \vdots & \vdots & & \vdots \\ \mathbf{x}_{d-1,1} & \mathbf{x}_{d-1,2} & \cdots & \mathbf{x}_{d-1,N} \\ \mathbf{x}_{d,1} & \mathbf{x}_{d,2} & \cdots & \mathbf{x}_{d,N} \end{bmatrix}$

- A set of k **regressors** variables $\mathbf{Reg} \subset \{1, \dots, d\}$
- A set of $d - k$ **response** variable $\mathbf{Res} \subset \{1, \dots, d\}$

The Regression Problem

- Regression = **build a function** $f : \mathbb{R}^k \rightarrow \mathbb{R}^{d-k}$ such that,

$$\forall \mathbf{x}, f((\mathbf{x}_i)_{i \in \text{Reg}}) \approx (\mathbf{x}_k)_{k \in \text{Res}}.$$

- *e.g.* if $d = 6$, $k = 4$, **Reg** = $\{1, 2, 3, 4\}$, **Res** = $\{5, 6\}$ we look for a function $f : \mathbb{R}^4 \rightarrow \mathbb{R}^2$,

$$f(\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \mathbf{x}_4) \approx (\mathbf{x}_5, \mathbf{x}_6)$$

Examples continued



$$\text{Credit card holder } \mathbf{x}_j = \begin{bmatrix} \text{Income} \\ \text{Age} \\ \vdots \\ \text{Work history (months)} \\ \text{Family} \\ \# \text{ Credit Incidents} \end{bmatrix}$$



$$\text{Patient } \mathbf{x}_j = \begin{bmatrix} \text{height} \\ \text{weight} \\ \vdots \\ \# \text{ minutes exercise/week} \\ \text{LDL cholesterol} \\ \text{HDL cholesterol} \end{bmatrix}$$



$$\text{Blog } \mathbf{x}_j = \begin{bmatrix} \text{avg. pages view/month} \\ \# \text{ posts} \\ \vdots \\ \text{avg. } \# \text{ comments/month} \\ \text{revenue from ads/month} \end{bmatrix}$$

Examples continued



$$\text{Credit card holder } \mathbf{x}_j = \begin{bmatrix} \text{Income} \\ \text{Age} \\ \vdots \\ \text{Work history (months)} \\ \text{Family} \\ \# \text{ Credit Incidents} \end{bmatrix}$$



$$\text{Patient } \mathbf{x}_j = \begin{bmatrix} \text{height} \\ \text{weight} \\ \vdots \\ \# \text{ minutes exercise/week} \\ \text{LDL cholesterol} \\ \text{HDL cholesterol} \end{bmatrix}$$



$$\text{Blog } \mathbf{x}_j = \begin{bmatrix} \text{avg. pages view/month} \\ \# \text{ posts} \\ \vdots \\ \text{avg. } \# \text{ comments/month} \\ \text{revenue from ads/month} \end{bmatrix}$$

In the following slides...

We only consider tasks with **one response** variable

- All other variables are **regressors**.
- We rename the **response** variable **y** and reassign x_1, \dots, x_d for the **regressors**
- predicting **more than one** variable? heavier mathematically, but similar.

In the following slides...

We assume that y takes **continuous values**.

- When y takes discrete values, notably binary $\{0, 1\}$ things change a bit.
- Yet... **binary** \subset **real** : regression techniques “work” on discrete data
- but **real** $\not\subset$ **binary**... we’ll discuss that later.

Today's Example: Your apartment

現在の条件に合う物件数 **1,226** 件中 **161~180** 件を表示しています。 [前へ](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [11](#) [12](#) [13](#) [14](#) [次へ](#)

[絞り込み条件をリセット](#) [一覧表示](#) [間取り表示](#) すべてにチェック チェックした物件をまとめて [詳細表示](#) [お問い合わせ](#)

画像	路線名/駅名 住所	バス 徒歩	賃料 管理費等	敷金または保証金 礼金(数引)	間取り 専有面積	築年月 (築年数)	選択
	京阪鴨東線/出町柳 京都市左京区田中大塚町 間取り図 写真	- 5分	4.20万円 3,000円	5万円 5万円(-)	1R 16.00m ²	'89/09 (築22年)	<input type="checkbox"/>
	叡山本線/修学院 京都市左京区山端滝ヶ鼻町 間取り図 写真	- 7分	4.30万円 2,000円	5万円 なし(なし)	1K 20.44m ²	'95/03 (築17年)	<input type="checkbox"/>
	叡山本線/修学院 京都市左京区山端滝ヶ鼻町 間取り図 写真	- 7分	4.30万円 2,000円	5万円 なし(なし)	1K 20.44m ²	'95/03 (築17年)	<input type="checkbox"/>
	叡山本線/一乗寺 京都市左京区一乗寺梅ノ木町 間取り図 写真	- 6分	4.30万円 2,000円	5万円 なし(-)	1K 20.00m ²	'88/03 (築24年)	<input type="checkbox"/>
	京阪鴨東線/出町柳 京都市左京区吉田上阿達町 間取り図 写真	- 7分	4.30万円 2,000円	5万円 5万円(なし)	1K 19.02m ²	'85/02 (築27年)	<input type="checkbox"/>
	烏丸線/今出川 京都市上京区柳園子町 間取り図 写真	- 3分	4.30万円 2,000円	なし 10万円(-)	1R 17.70m ²	'84/10 (築27年)	<input type="checkbox"/>
	烏丸線/今出川 京都市上京区北小路室町 間取り図 写真	- 2分	4.50万円 なし	5万円 5万円(-)	1K 16.00m ²	'94/03 (築18年)	<input type="checkbox"/>
	京阪鴨東線/出町柳 京都市左京区田中下柳町 間取り図 写真	- 3分	4.30万円 2,500円	5万円 8万円(-)	1K 20.00m ²	'85/03 (築27年)	<input type="checkbox"/>
	叡山本線/修学院 京都市左京区高野泉町 間取り図 写真	- 5分	4.05万円 5,500円	6万円 5万円(なし)	1K 14.58m ²	'79/01 (築33年)	<input type="checkbox"/>

選択路線

叡山本線

出町柳(824)

元田中(380) 茶山(388)

一乗寺(279)

修学院(187)

[路線を選びなおす](#) [駅を選びなおす](#)

基本条件

賃料 [選び方を変える](#)

安 高

下限なし~上限なし

管理費・共益費込み(1,226)

礼金なし(266)

敷金・保証金なし(224)

間取り

1R(102) 1K/1DK(778)

1LDK(66) 2K/2DK(73)

2LDK(99) 3K/3DK(22)

3LDK(78) 4K/4DK(0)

4LDK以上(8)

最寄り駅からの時間(徒歩)

1分以内(63) 5分以内(466)

7分以内(706) 10分以内(902)

15分以内(1,076)

指定なし(1,226)

Collected information about 285 (out of 1226) apartments close to Kyoto U.

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

Today's Example: Your apartment

現在の条件に合う物件数 **1,226** 件中 **161~180** 件を表示しています。 [前へ](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [11](#) [12](#) [13](#) [14](#) [次へ](#)

[絞り込み条件をリセット](#) [一覧表示](#) [間取り表示](#) すべてにチェック [チェックした物件をまとめて](#) [詳細表示](#) [お問い合わせ](#)

画像	路線名/駅名 住所	バス 徒歩	賃料 管理費等	敷金または保証金 礼金(数引)	間取り 専有面積	築年月 (築年数)	選択
	京阪鴨東線/出町柳 京都市左京区田中大塚町 間取り図 写真	- 5分	4.20万円 3,000円	5万円 5万円(-)	1R 16.00m ²	'89/09 (築22年)	<input type="checkbox"/>
	叡山本線/修学院 京都市左京区山端滝ヶ鼻町 間取り図 写真	- 7分	4.30万円 2,000円	5万円 なし(なし)	1K 20.44m ²	'95/03 (築17年)	<input type="checkbox"/>
	叡山本線/修学院 京都市左京区山端滝ヶ鼻町 間取り図 写真	- 7分	4.30万円 2,000円	5万円 なし(なし)	1K 20.44m ²	'95/03 (築17年)	<input type="checkbox"/>
	叡山本線/一乗寺 京都市左京区一乗寺梅ノ木町 間取り図 写真	- 6分	4.30万円 2,000円	5万円 なし(-)	1K 20.00m ²	'88/03 (築24年)	<input type="checkbox"/>
	京阪鴨東線/出町柳 京都市左京区吉田上阿達町 間取り図 写真	- 7分	4.30万円 2,000円	5万円 5万円(なし)	1K 19.02m ²	'85/02 (築27年)	<input type="checkbox"/>
	烏丸線/今出川 京都市上京区柳園子町 間取り図 写真	- 3分	4.30万円 2,000円	なし 10万円(-)	1R 17.70m ²	'84/10 (築27年)	<input type="checkbox"/>
	烏丸線/今出川 京都市上京区北小路室町 間取り図 写真	- 2分	4.50万円 なし	5万円 5万円(-)	1K 16.00m ²	'94/03 (築18年)	<input type="checkbox"/>
	京阪鴨東線/出町柳 京都市左京区田中下柳町 間取り図 写真	- 3分	4.30万円 2,500円	5万円 8万円(-)	1K 20.00m ²	'85/03 (築27年)	<input type="checkbox"/>
	叡山本線/修学院 京都市左京区高野泉町 間取り図 写真	- 5分	4.05万円 5,500円	6万円 5万円(なし)	1K 14.58m ²	'79/01 (築33年)	<input type="checkbox"/>

選択路線

叡山本線

出町柳(824) 元田中(380) 茶山(388)

一乗寺(279) 修学院(187)

[路線を選びなおす](#) [駅を選びなおす](#)

基本条件

賃料 [選び方を考える](#)

安 高

下限なし~上限なし

管理費・共益費込み(1,226)

礼金なし(266)

敷金・保証金なし(224)

間取り

1R(102) 1K/1DK(778)

1LDK(66) 2K/2DK(73)

2LDK(99) 3K/3DK(22)

3LDK(78) 4K/4DK(0)

4LDK以上(8)

最寄り駅からの時間(徒歩)

1分以内(63) 5分以内(466)

7分以内(706) 10分以内(902)

15分以内(1,076)

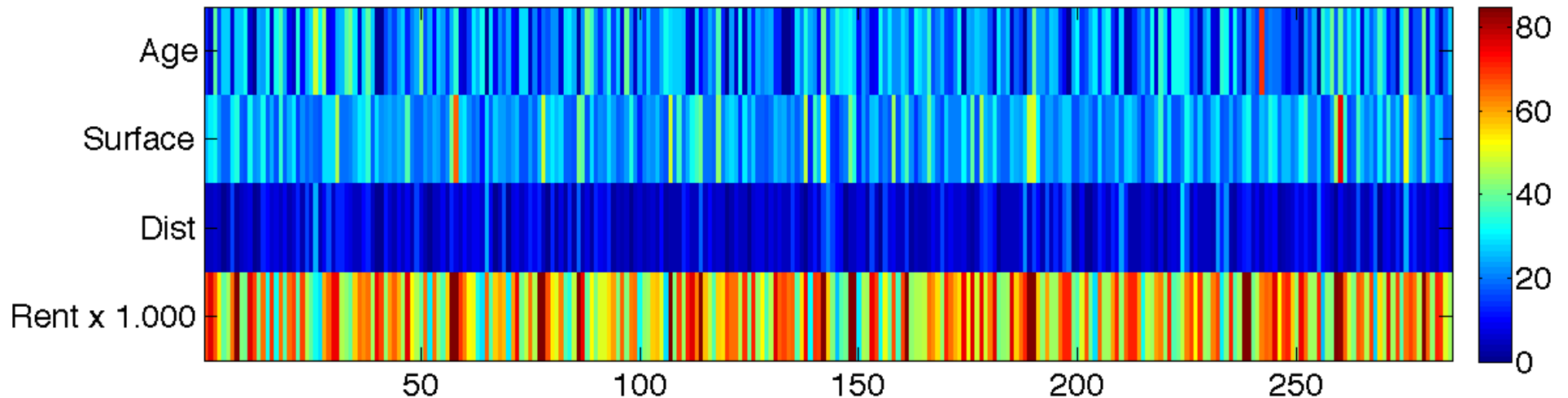
指定なし(1,226)

Collected information about 285 (out of 1226) apartments close to Kyoto U.

Kept 4 variables: **Surface**, **Rent**, **Age of Building**, **Walking distance to station**.

What does the matrix look like?

```
imagecs(H); colorbar;
```

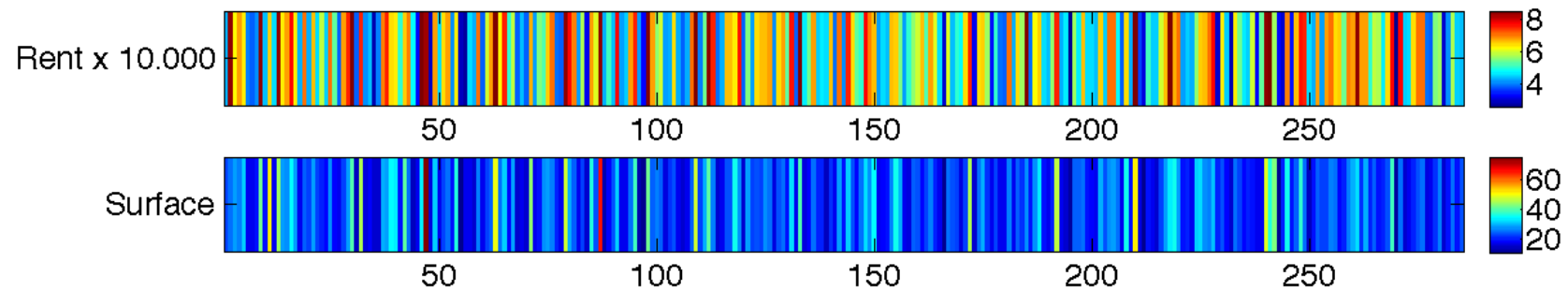


285 columns, 4 lines.

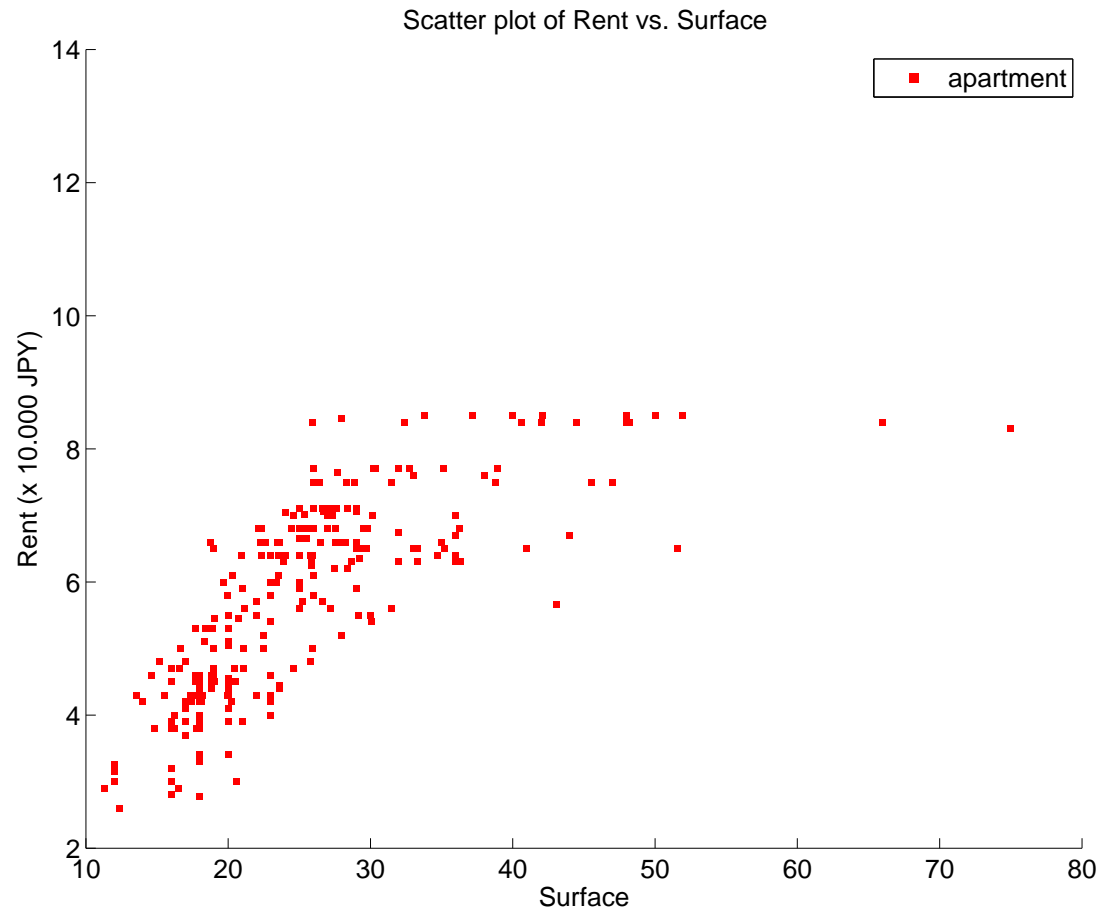
Each column represents one apartment.

In these slides, we will **regress** the rent using age, surface and distance

Regression: one variable vs. another

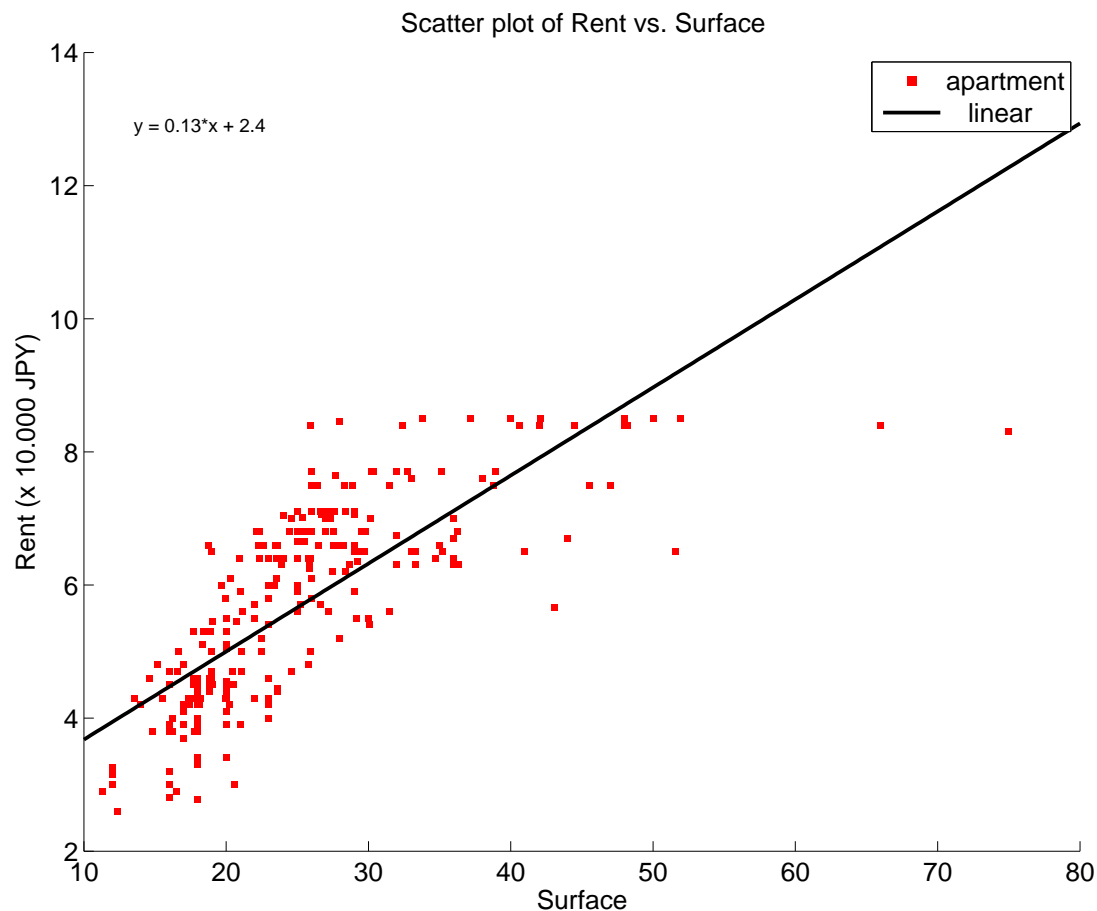


Rent vs. Surface



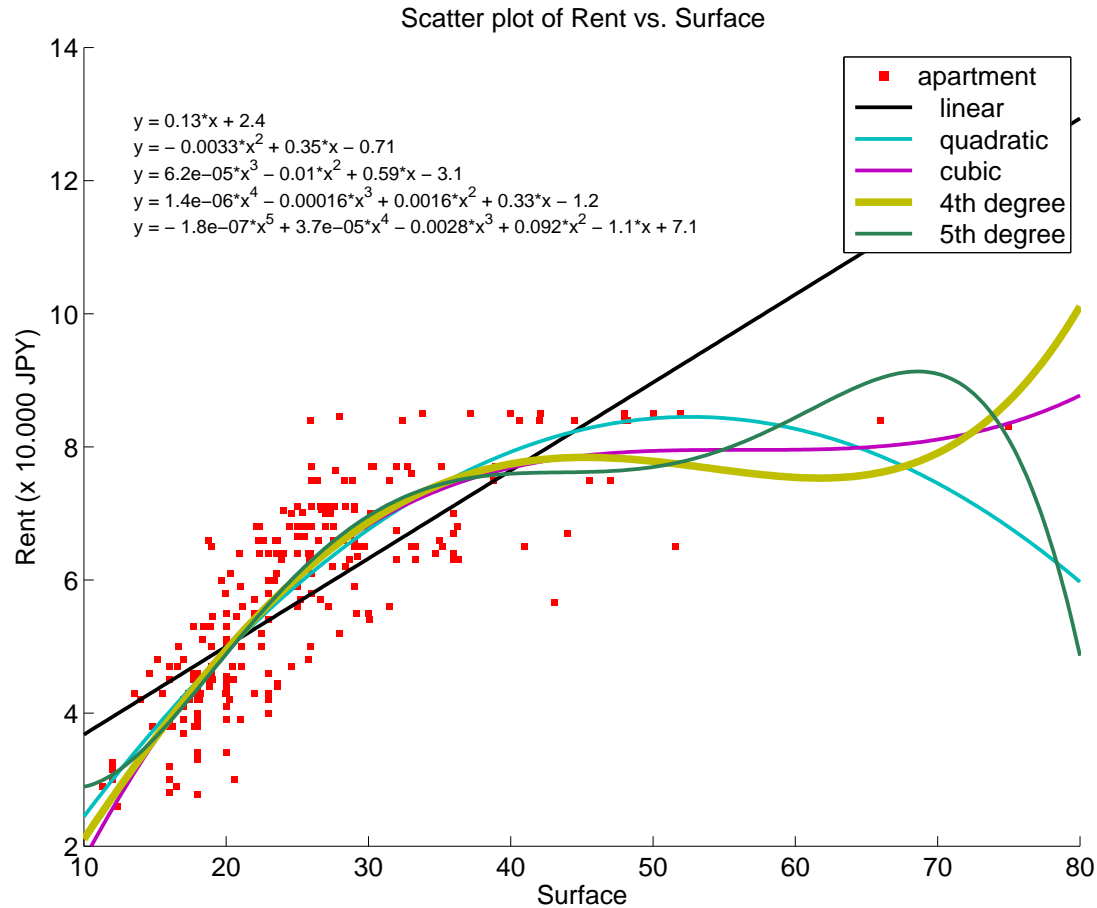
Note that the dataset has been censored above 85.000 JPY

Rent vs. Surface



Using the linear tool in curve fitting, we obtain the approximation $y = 0.13x + 2.4$

Rent vs. Surface



We can use higher order polynomials... yet look at the results.

Behind the curve fitting tool

- Matlab selects these curves using the **least-squares** criterion e.g

$$\min_{f \in \mathcal{F}} \sum_{j=1}^N (y_j - f(x_j))^2$$

where \mathcal{F} is a **class of functions**

- Matlab considers a few function classes for \mathcal{F} .

Behind the curve fitting tool

- Matlab selects these curves using the **least-squares** criterion e.g

$$\min_{f \in \mathcal{F}} \sum_{j=1}^N (y_j - f(x_j))^2$$

where \mathcal{F} is a **class of functions**

- Matlab considers a few function classes for \mathcal{F} . Among them..

- **Linear** $\min_{b, a_1 \in \mathbb{R}} \sum_{j=1}^N (y_j - (b + a_1 x_j))^2$

Behind the curve fitting tool

- Matlab selects these curves using the **least-squares** criterion e.g

$$\min_{f \in \mathcal{F}} \sum_{j=1}^N (y_j - f(x_j))^2$$

where \mathcal{F} is a **class of functions**

- Matlab considers a few function classes for \mathcal{F} . Among them..

- **Linear** $\min_{b, a_1 \in \mathbb{R}} \sum_{j=1}^N (y_j - (b + a_1 x_j))^2$

- **Quadratic** $\min_{b, a_1, a_2 \in \mathbb{R}} \sum_{j=1}^N (y_j - (b + a_1 x_j + a_2 x_j^2))^2$

Behind the curve fitting tool

- Matlab selects these curves using the **least-squares** criterion e.g

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

where \mathcal{F} is a **class of functions**

- Matlab considers a few function classes for \mathcal{F} . Among them..

- **Linear** $\min_{b, a_1 \in \mathbb{R}} \sum_{j=1}^N (\mathbf{y}_j - (b + a_1 \mathbf{x}_j))^2$

- **Quadratic** $\min_{b, a_1, a_2 \in \mathbb{R}} \sum_{j=1}^N \left(\mathbf{y}_j - (b + a_1 \mathbf{x}_j + a_2 \mathbf{x}_j^2) \right)^2$

- **Cubic** $\min_{b, a_1, a_2, a_3 \in \mathbb{R}} \sum_{j=1}^N \left(\mathbf{y}_j - (b + a_1 \mathbf{x}_j + a_2 \mathbf{x}_j^2 + a_3 \mathbf{x}_j^3) \right)^2$

- *etc.*

How can we solve this? The linear case

- Let's take a look at the function

$$(a, b) \mapsto \sum_{j=1}^N (\mathbf{y}_j - (\mathbf{b} + \mathbf{a}x_j))^2.$$

- Using the notations

$$\text{Rent } Y = [y_1 \quad y_2 \quad \cdots \quad y_N]$$

$$\text{Surface } X = [x_1 \quad x_2 \quad \cdots \quad x_N]$$

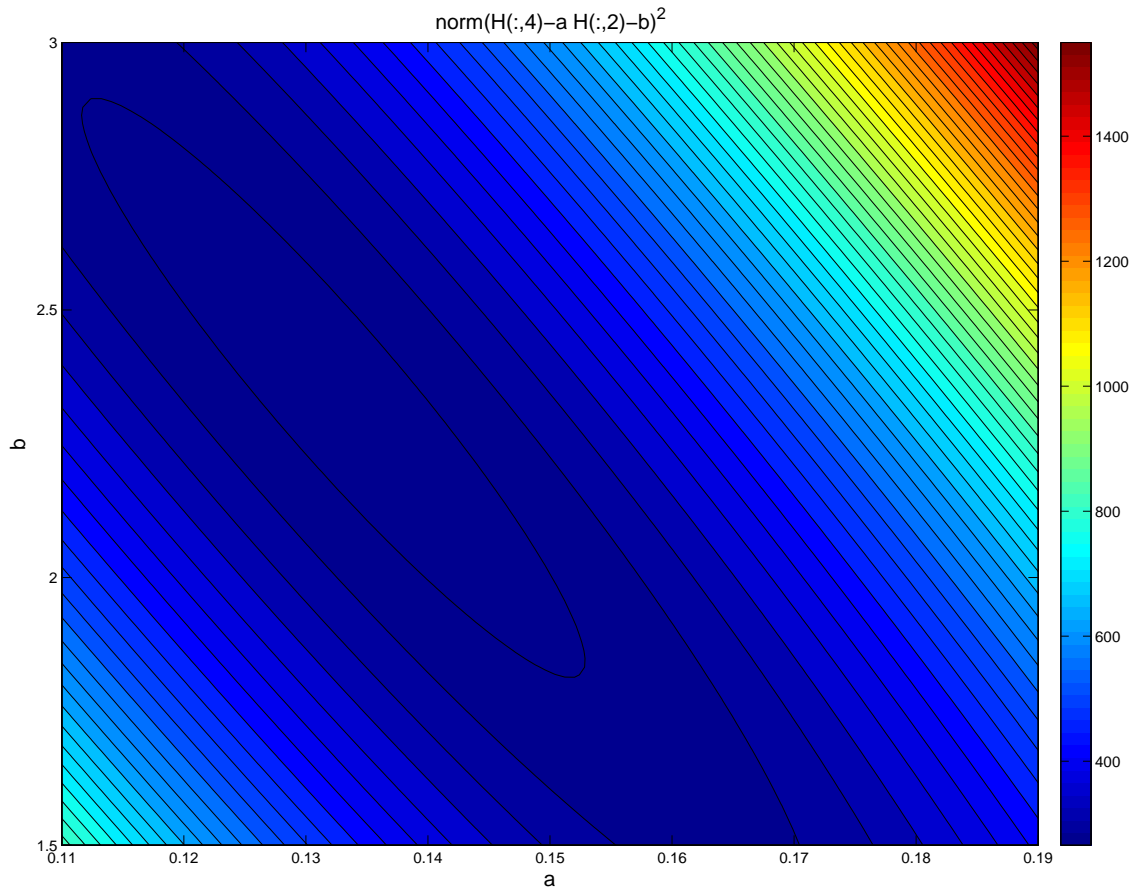
$$\text{Constant } \mathbf{1}_N = [1 \quad 1 \quad \cdots \quad 1]$$

we have that

$$\sum_{j=1}^N (\mathbf{y}_j - (\mathbf{a}x_j + \mathbf{b}))^2 = \|\mathbf{Y} - \mathbf{a}\mathbf{X} - \mathbf{b}\mathbf{1}_N\|^2$$

Contour plot of $(a, b) \rightarrow \|Y - aX - b\mathbf{1}_N\|^2$

- Since we only handle 2 parameters, we can make a contour plot



- This validates the equation $\mathbf{y} = 0.13x + 2.4$. How to get there?

Some linear algebra

- We define the function L as

$$L : (a, b) \mapsto \sum_{j=1}^N (\mathbf{y}_j - (\mathbf{b} + \mathbf{a}x_j))^2$$

- The partial derivatives of L can be computed.

$$\frac{\partial L}{\partial a} = -2 \sum_{j=1}^N (\mathbf{y}_j - (\mathbf{b} + \mathbf{a}x_j)) \mathbf{x}_j$$

$$\frac{\partial L}{\partial b} = -2 \sum_{j=1}^N \mathbf{y}_j - (\mathbf{b} + \mathbf{a}x_j)$$

- Any minimum (a^*, b^*) of L must be a saddle point.

Some linear algebra

- Namely, the partial derivatives of L at (a^*, b^*) must be zero

$$\frac{\partial L}{\partial a} = 2 \left(a \sum x_j^2 + b \sum x_j - \sum y_j x_j \right)$$

$$\frac{\partial L}{\partial b} = 2 \left(Nb - \sum y_j + a \sum x_j \right)$$

- Hence (a^*, b^*) **must satisfy** the linear system

$$0 = a^* \sum x_j^2 + b^* \sum x_j - \sum y_j x_j$$

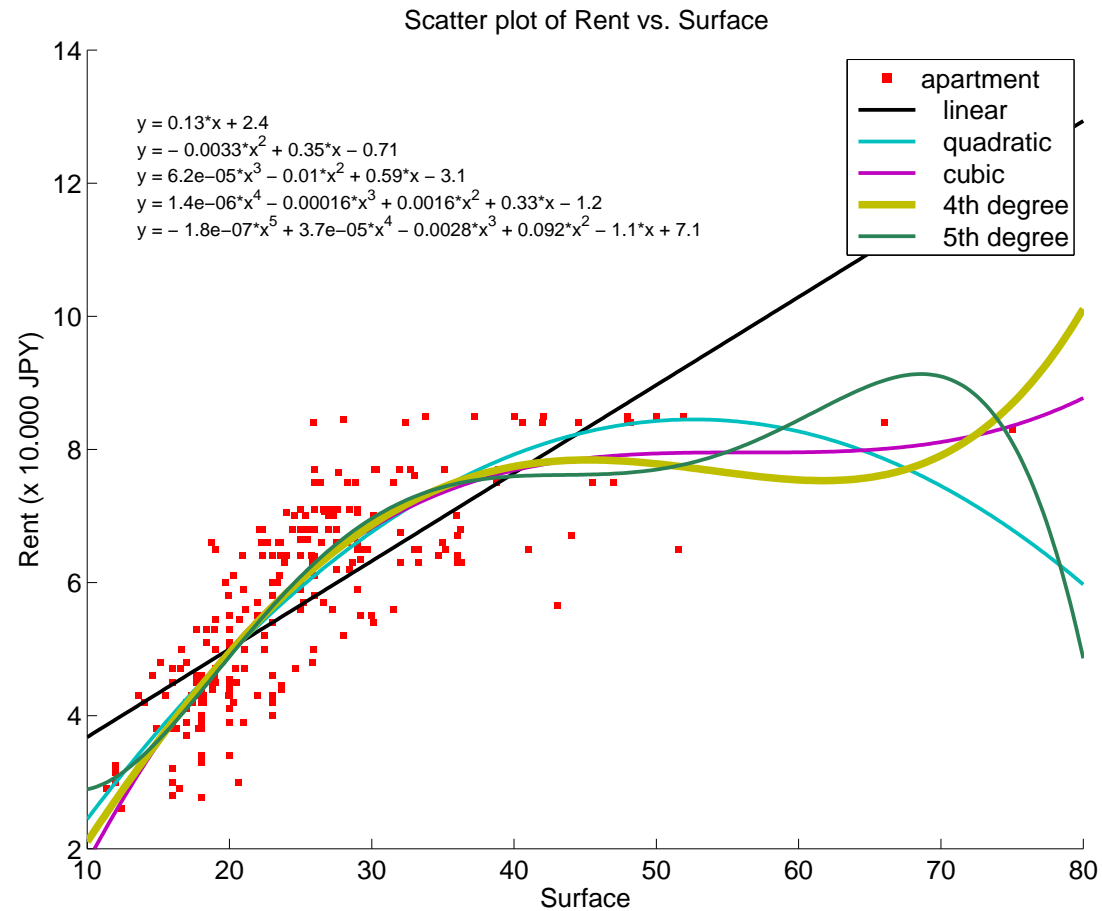
$$0 = Nb^* - \sum y_j + a^* \sum x_j$$

- Namely,

$$\begin{bmatrix} a^* \\ b^* \end{bmatrix} = \begin{bmatrix} \sum x_j^2 & \sum x_j \\ \sum x_j & N \end{bmatrix}^{-1} \begin{bmatrix} \sum y_j x_j \\ \sum y_j \end{bmatrix}$$

- ans = 0.132248772789152 2.354203561671262

Rent vs. Surface



We understood how to get the linear curve. What about the quadratic?

What about the quadratic case?

$$\text{Quadratic } \min_{b, a_1, a_2 \in \mathbb{R}} \sum_{j=1}^N \left(y_j - (b + a_1 x_j + a_2 x_j^2) \right)^2$$

- same idea... define

$$L : (a_1, a_2, b) \mapsto \sum_{j=1}^N \left(y_j - (b + a_1 x_j + a_2 x_j^2) \right)^2$$

- look at the objective's derivatives...

$$\frac{\partial L}{\partial a_2} = -2 \sum_{j=1}^N \left(y_j - (b + a_1 x_j + a_2 x_j^2) \right) x_j^2$$

$$\frac{\partial L}{\partial a_1} = -2 \sum_{j=1}^N \left(y_j - (b + a_1 x_j + a_2 x_j^2) \right) x_j$$

$$\frac{\partial L}{\partial b} = -2 \sum_{j=1}^N \left(y_j - (b + a_1 x_j + a_2 x_j^2) \right)$$

What about the quadratic case?

- We consider the equations that a saddle point must verify:

$$0 = \sum_{j=1}^N \left(y_j - \left(b^* + a_1^* x_j + a_2^* x_j^2 \right) \right) x_j^2$$

$$0 = \sum_{j=1}^N \left(y_j - \left(b^* + a_1^* x_j + a_2^* x_j^2 \right) \right) x_j$$

$$0 = \sum_{j=1}^N \left(y_j - \left(b^* + a_1^* x_j + a_2^* x_j^2 \right) \right)$$

$$\begin{bmatrix} a_2^* \\ a_1^* \\ b^* \end{bmatrix} = \begin{bmatrix} \sum x_j^4 & \sum x_j^3 & \sum x_j^2 \\ \sum x_j^3 & \sum x_j^2 & \sum x_j \\ \sum x_j^2 & \sum x_j & N \end{bmatrix}^{-1} \begin{bmatrix} \sum y_j x_j^2 \\ \sum y_j x_j \\ \sum y_j \end{bmatrix}$$

- ans = -0.003306463076068 0.347969105896777 -0.705157514974559

Higher order polynomials

- Intuitively, for polynomial up to degree p we would have to
 - Build the corresponding Toeplitz Matrix
 - Build the corresponding vector with y and x combined at different exponents
 - Solve the linear system
- Surprisingly

Finding the **best p^{th} order polynomial** with **least-squares**



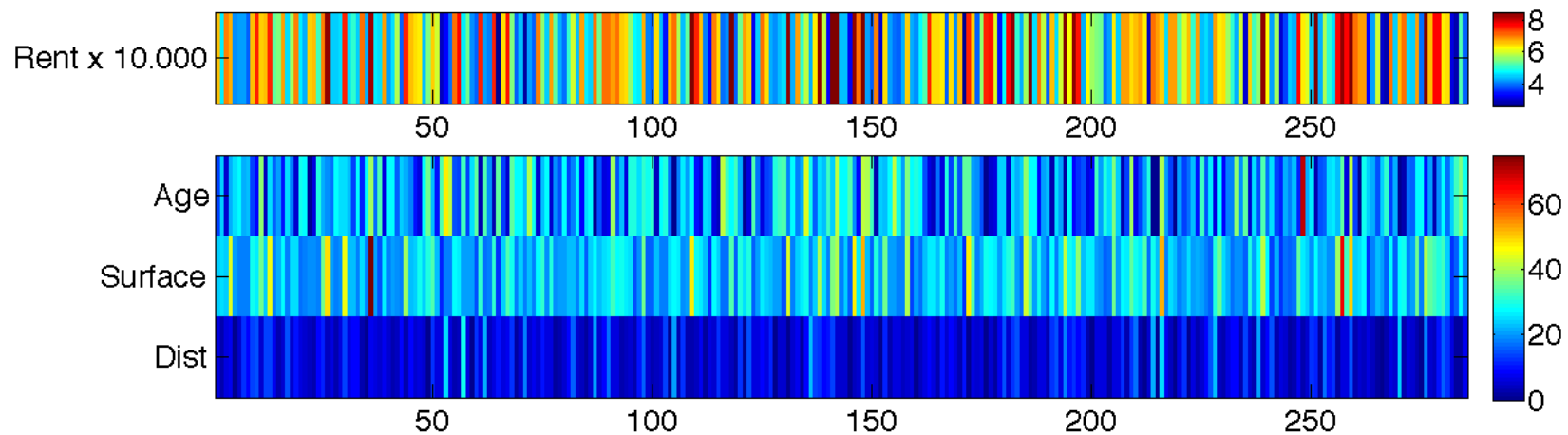
Solving a **p dimensional** linear system

- Not so surprising after all:
 - Least-squares: objective of degree 2 in coefficients
 - Minimum \Leftrightarrow saddle point \Leftrightarrow system of degree 1..
 - Least-squares has been chosen **because** it yields a linear system...

The general case: one vs. all rest

- What about using all other variables?

$$\begin{array}{l} \text{Rent} \\ \text{All other variables} \end{array} \quad Y = \begin{bmatrix} y_1 & y_2 & \cdots & y_N \\ \vdots & \vdots & \cdots & \vdots \\ \mathbf{x}_1 & \mathbf{x}_2 & \cdots & \mathbf{x}_N \\ \vdots & \vdots & \cdots & \vdots \end{bmatrix}$$



The general case

- We assume that we have d **regressor** variables, 1 **response** variable.
- Consider again the **linear** approach. We look for a function f of the form

$$f(\mathbf{x}) = \alpha_0 + \alpha_1 x_1 + \alpha_2 x_2 + \cdots + \alpha_d x_d.$$

- We want to determine $d + 1$ weights,
 - a constant α_0
 - $1 \leq i \leq d, \alpha_i$ weights for each variable.
- **Least squares:**

$$L(\alpha_0, \alpha_1, \alpha_2, \cdots, \alpha_d) = \sum_{j=1}^N \left(\mathbf{y}_j - (\alpha_0 + \alpha_1 x_{1,j} + \alpha_2 x_{2,j} + \cdots + \alpha_d x_{d,j}) \right)^2$$

The general case

- Notice that

$$L(\alpha_0, \alpha_1, \alpha_2, \dots, \alpha_d) \rightarrow \sum_{i=1}^N \left(y_i - \left(\alpha_0 + \begin{bmatrix} \alpha_1 \\ \vdots \\ \alpha_d \end{bmatrix}^T \mathbf{x}_i \right) \right)^2 = \left\| \begin{bmatrix} \alpha_0 \\ \vdots \\ \alpha_d \end{bmatrix}^T X - Y \right\|^2,$$

where

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

and

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

- We write α for the $d + 1$ dimensional vector $\begin{bmatrix} \alpha_0 \\ \vdots \\ \alpha_d \end{bmatrix}$.

Linear least squares

- Expanding this expression,

$$L(\alpha) = (\alpha^T X X^T \alpha - 2Y X^T \alpha + \|Y\|^2)$$

- Consider the **gradient** of that function

$$\nabla L = 2X X^T \alpha - 2X Y^T$$

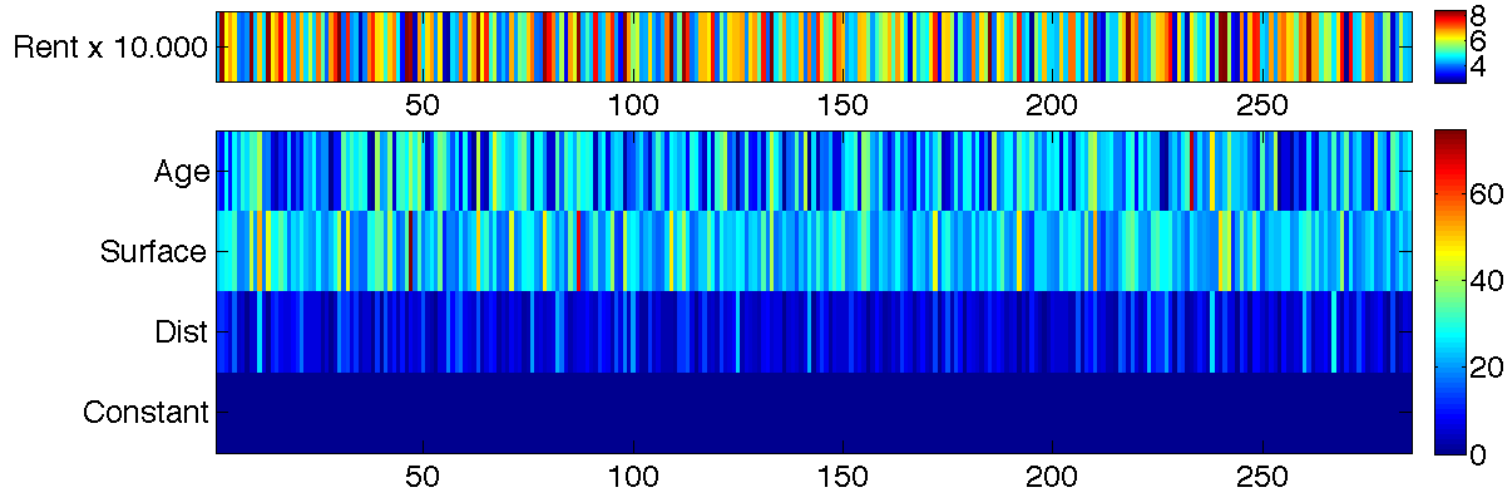
- Hence this gradient is zero for

$$\alpha^* = (X X^T)^{-1} X Y^T$$

- $X X^T \in \mathbf{S}_+^n$, that is $X X^T$ is a positive (semi)definite matrix.
- This works if $X X^T \in \mathbb{R}^{d+1}$ is **invertible**, that is $X X^T \in \mathbf{S}_{++}^n$.

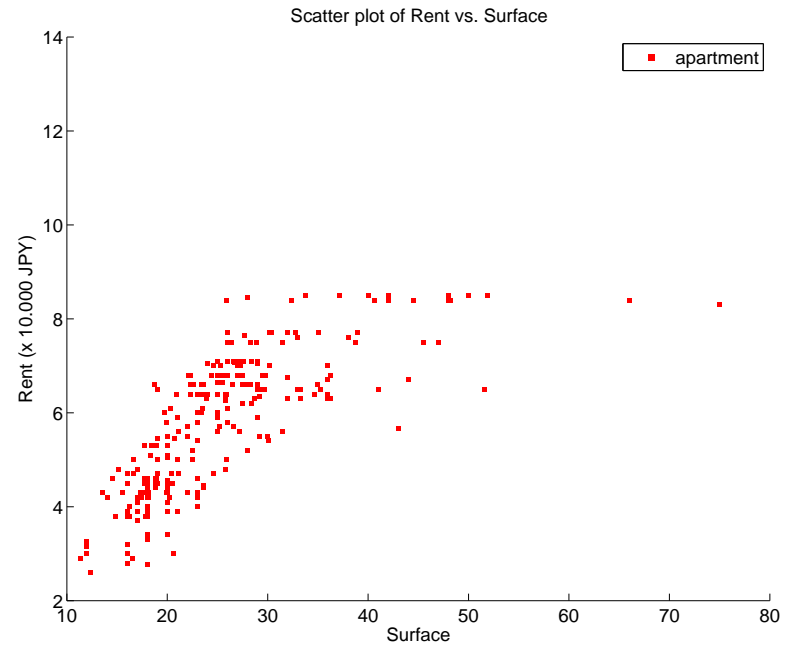
Considering again rents vs the rest

- Getting the data again, adding a line of 1's



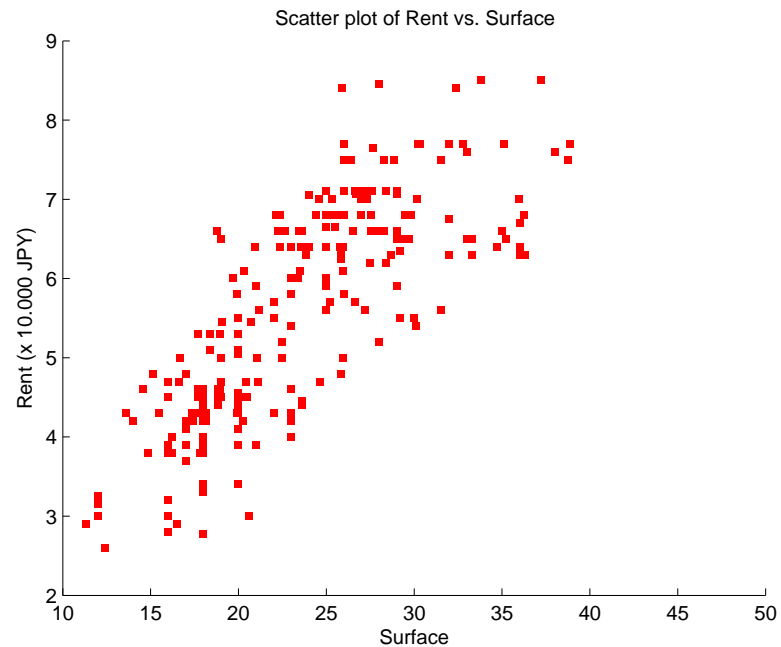
```
>>> (X*X') \ (X*Y')
ans =
    0.000141678721821
    0.004226687659299
   -0.012599982792209
    5.611128285287092
```

What went wrong?



$$\text{rent} = 1.4 \text{ age} + 42.2 \text{ surf} - 125 \text{ dist} + 56.110 \text{ JPY}$$

What happens if we remove outliers? (surf > 40)



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

```
ans =
```

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

Moral of the story: easy to draw wrong conclusions **even with simple tools**

What else can go wrong? Next time...

- What happens when $d \gg n$? (XX^T) is **no longer invertible**...
 - high-dimensional data in genomics,
 - images analysis (lots of features)

- What happens when (XX^T) is **badly conditioned** ($\frac{\lambda_{\min}(XX^T)}{\lambda_{\max}(XX^T)} \approx 0$)?
 - if $\lambda_{\min}(XX^T) = 1e-10$, $\lambda_{\max}((XX^T)^{-1}) = 1e10!!$
 - Very bad numerical stability of the solution...

- When $d \gg n$, we might want to do **variable selection**,
 - *i.e.* pick a subset d' of the d variables which is relevant to predict \mathbf{y} .
 - *i.e.* favor vectors α such that $\|\alpha\|_0 = \text{card}\{\alpha_i \neq 0\}$ is small.