THE UNIVERSITY of ADELAIDE

Regression

Lingqiao Liu
University of Adelaide

Outlines

- Introduction to regression
- Linear Regression
  - Regression to scalar values
  - Regression to vectors
- Regularized Regression
  - Ridge regression
  - Lasso
- Support Vector Regression

What is regression?

- Review
  - Types of Machine Learning?

What is regression?

- Supervised learning:
  - Known at the training stage (x, y)
  - Predict unknown y for x at the test stage
- Classification
  - Y is a discrete variable
- Regression
  - Y is a continuous variable

Example of Regression Tasks

Classification: Selling a house will make a profit or not ?(yes/no)

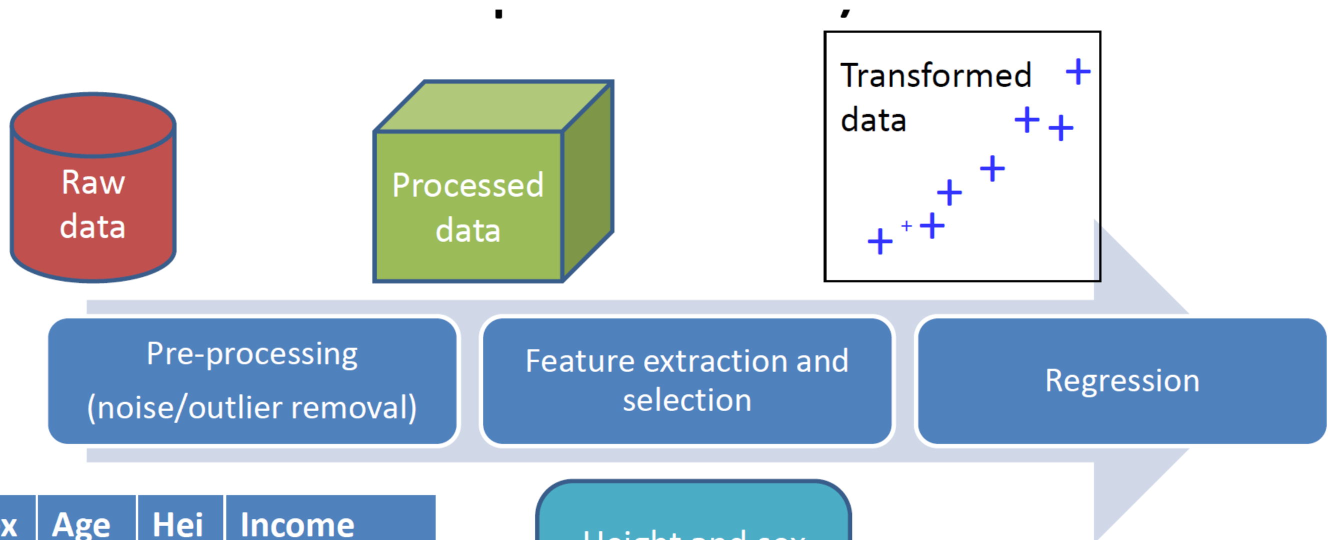
Regression: Sale price for a house? (dollar amount)

Input: $\mathbf{x} =$

building size	250 sq meters
land size	400 sq meters
# bedrooms	3
# bathrooms	2
# parking	2 (double garage)
# stories	1 (i.e. single story)
...	...

Learning to predict housing price y $y = f(\mathbf{x})$

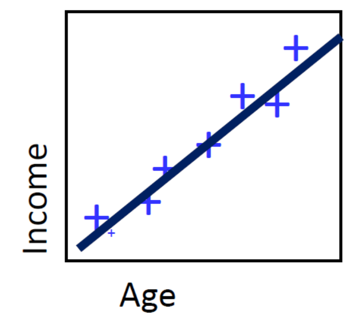
Practical workflow



Sex	Age	Height	Income
M	20	1.7	25,000
F	30	1.6	55,000
M	**125**	1.8	30,000
…			

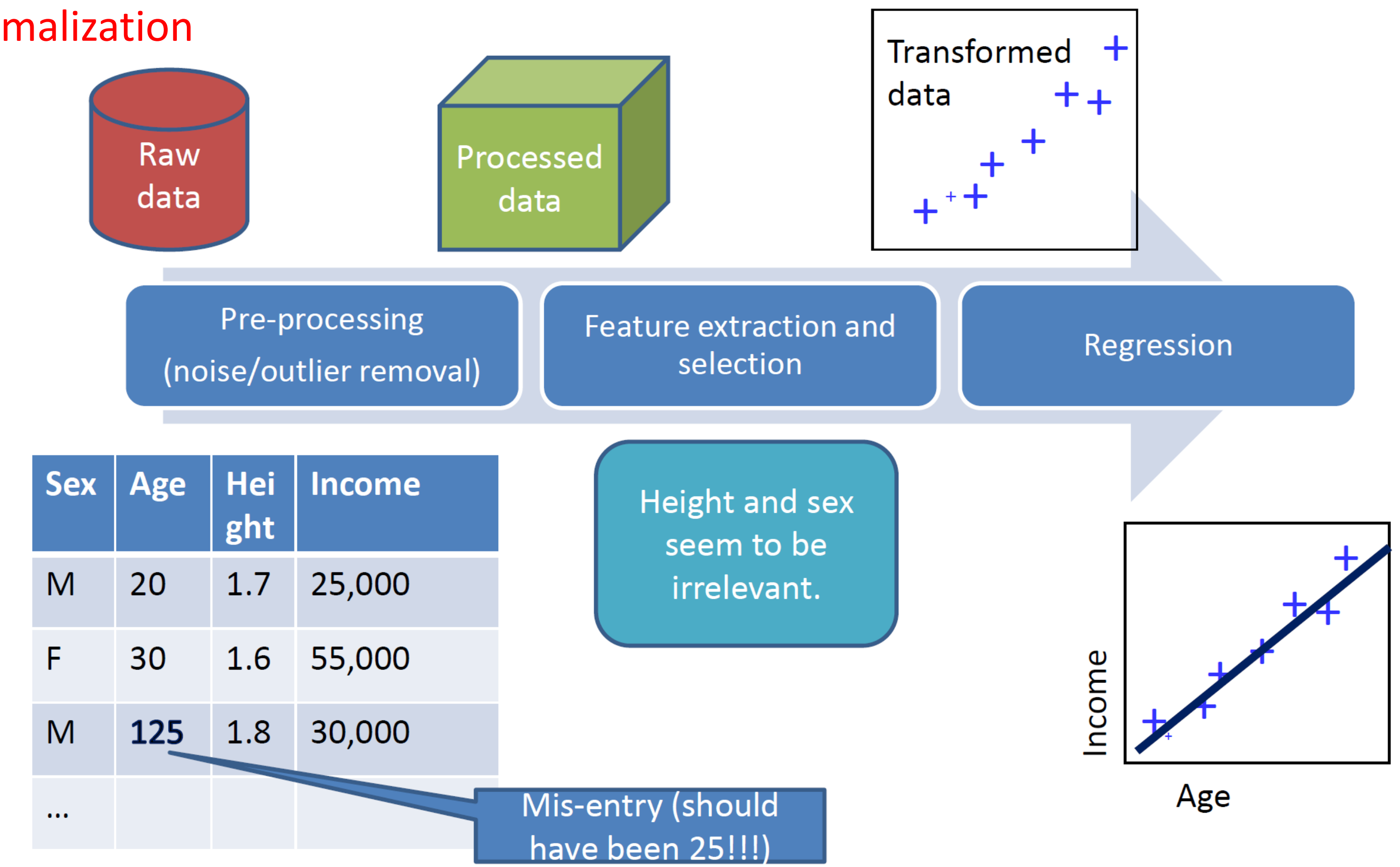
Mis-entry (should have been 25!!!)

Height and sex seem to be irrelevant.



Practical workflow

For some methods, we also need to perform normalization



Sex	Age	Height	Income
M	20	1.7	25,000
F	30	1.6	55,000
M	**125**	1.8	30,000
…			

Example of Regression Tasks

Less obvious task: Image processing

Example of Regression Tasks

Less obvious task: crowd counting

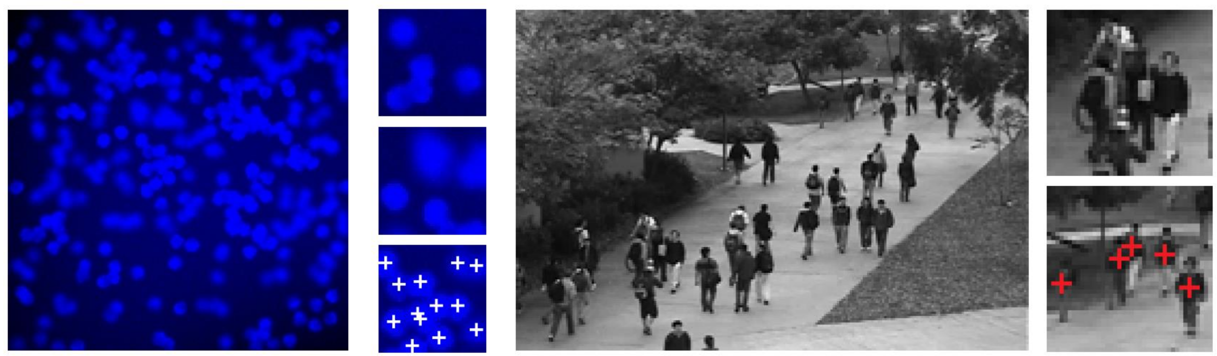
(a)

Example of Regression Tasks

Learning To Count Objects in Images

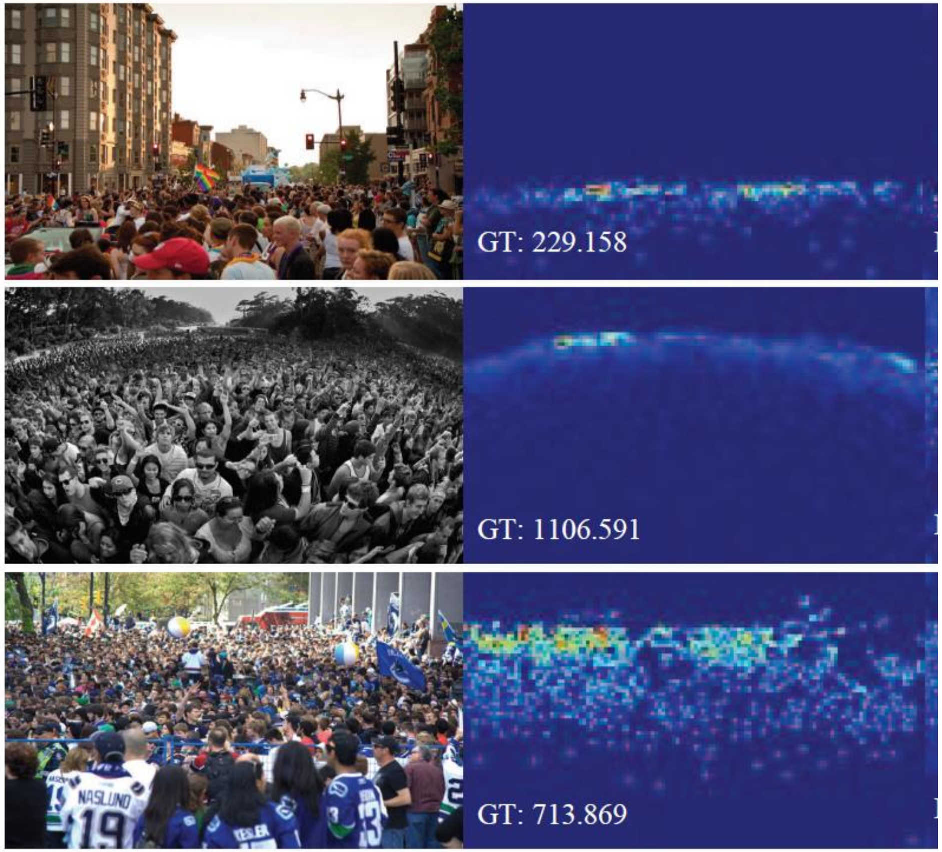
Victor Lempitsky
Visual Geometry Group
University of Oxford

Andrew Zisserman
Visual Geometry Group
University of Oxford



Example of Regression Tasks

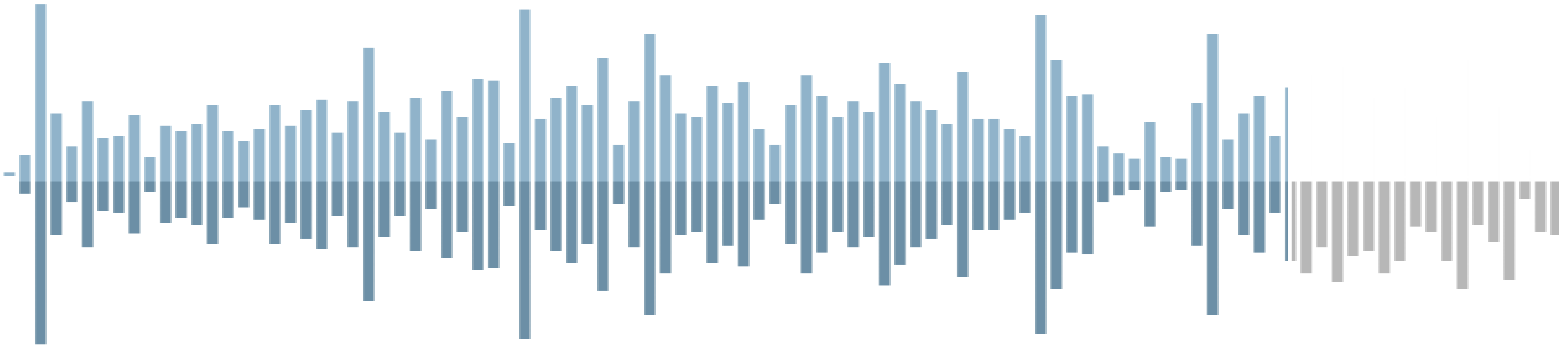
Less obvious task: crowd counting



$$\int D(x)dx = N$$

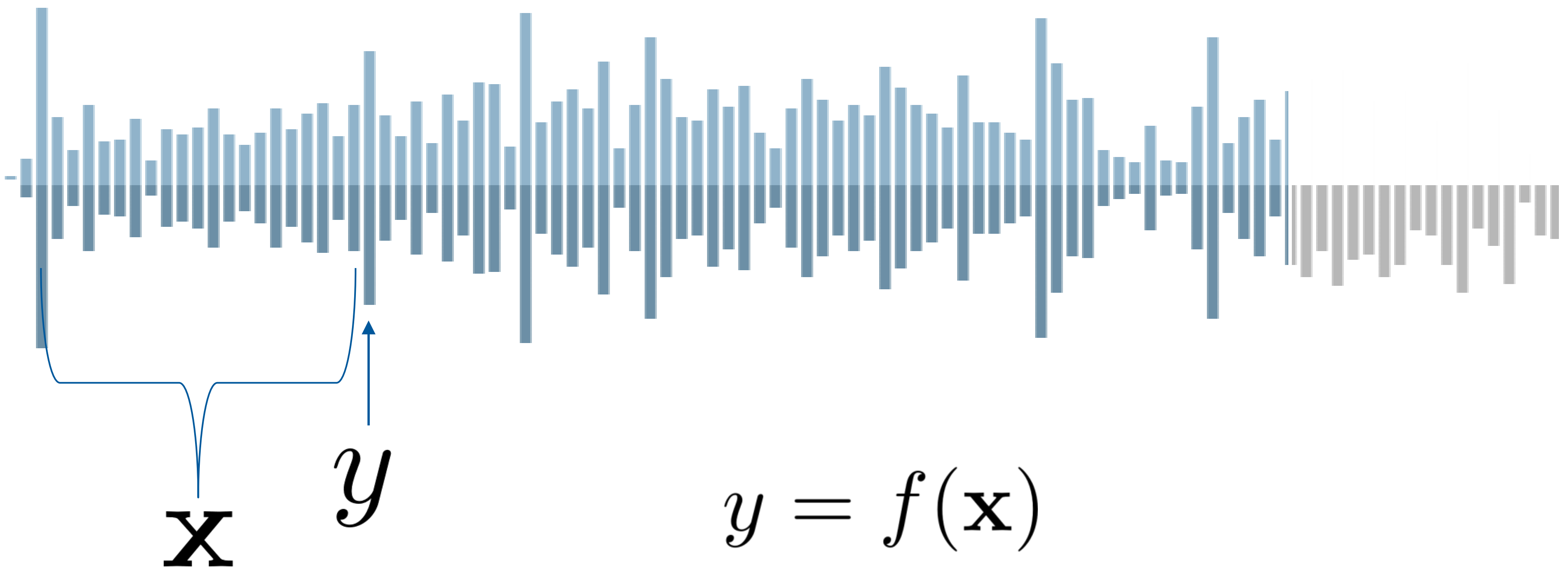
Example of Regression Tasks

Less obvious task: generating sounds



Example of Regression Tasks

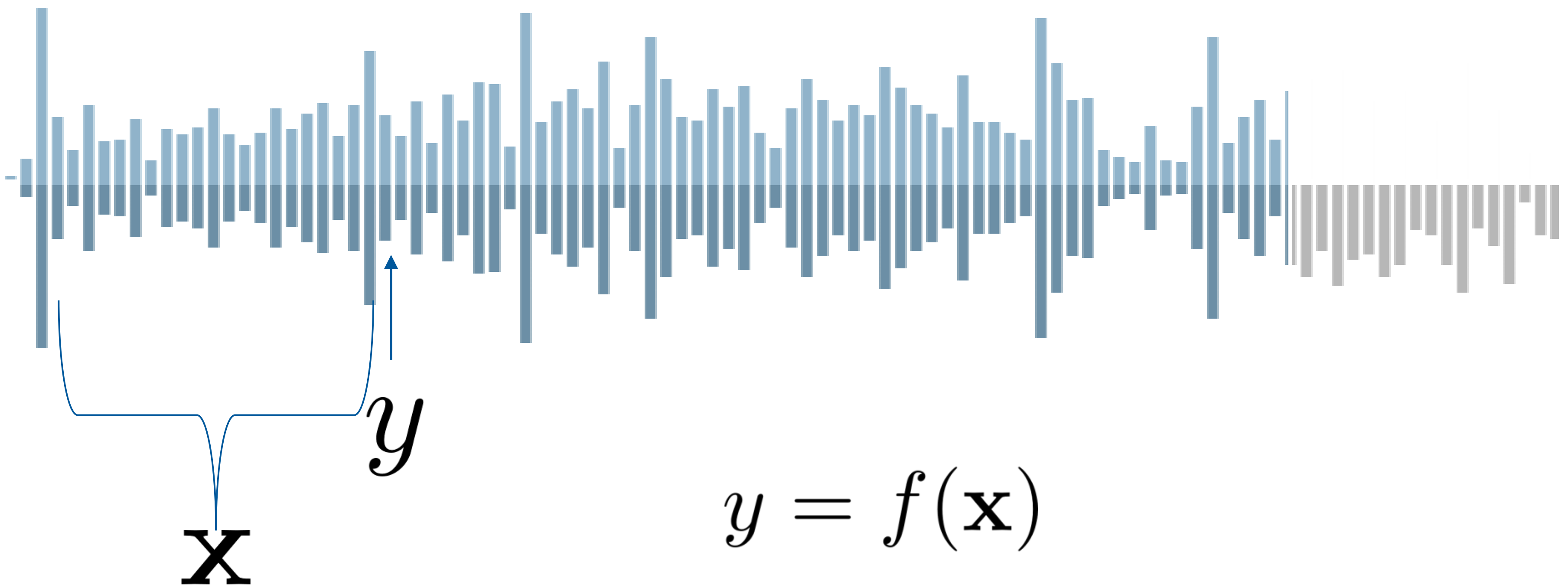
Less obvious task: generating sounds



$$y = f(\mathbf{x})$$

Example of Regression Tasks

Less obvious task: generating sounds (auto-regressive model)



$$y = f(\mathbf{x})$$

Regression framework

$$y = f(\mathbf{x})$$

- Goal:
  - What to fit?
- Model: how to fit?
  - Linear model
  - Nonlinear model
- Loss: how to measure the fitting error? Or additional objectives?
  - E.g. MSE loss

Linear regression

$$y = f(\mathbf{x})$$

- Goal:
  - Fit the scalar value y
- Model:

$$y = \mathbf{w}^T \mathbf{x} = \sum_{k=1}^{D} w_k x_k$$

$\sum_{k=1}^{D} w_k x_k + b$ can be achieved by setting $x_{D+1} = 1$

- Loss:

$$err = (f(\mathbf{x}) - \hat{y})^2$$

Linear regression

- Loss for N training samples:

$$\mathcal{L} = \sum_{i=1}^{N} (f(\mathbf{x}_i) - \hat{y}_i)^2$$

$$= \sum_{i=1}^{N} (\mathbf{w}^T \mathbf{x}_i - \hat{y}_i)^2$$

$$= \|\mathbf{w}^T \mathbf{X} - \hat{\mathbf{y}}\|_2^2$$

$$\mathbf{X} \in \mathbf{R}^{d \times N}$$
$$\hat{\mathbf{y}} \in \mathbf{R}^{1 \times N}$$
$$\mathbf{w} \in \mathbf{R}^{d \times 1}$$

Linear regression

- Solution

$$\mathbf{w}^* = \underset{\mathbf{w}}{argmin} \, \|\mathbf{w}^T\mathbf{X} - \hat{\mathbf{y}}\|_2^2$$

$$\frac{\partial \|\mathbf{w}^T\mathbf{X} - \hat{\mathbf{y}}\|_2^2}{\partial \mathbf{w}} = 0$$

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

Linear regression

- Extension to vector outputs

$$\mathcal{L} = \sum_{i=1}^{N} \|f(\mathbf{x}_i) - \hat{\mathbf{y}}_\mathbf{i}\|_2^2$$
$$= \|\mathbf{W}^T\mathbf{X} - \hat{\mathbf{Y}}\|_2^2$$

$$\mathbf{X} \in \mathbf{R}^{d\times N}$$
$$\hat{\mathbf{Y}} \in \mathbf{R}^{c\times N}$$
$$\mathbf{W} \in \mathbf{R}^{d\times c}$$

- Similar solution

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

Classification as Regression

- Vector encoding of classification target
  - One hot vector

$$y = \left\{ \begin{matrix} -1 \\ 1 \\ . \\ . \\ -1 \\ -1 \end{matrix} \right\}$$

$$\|\mathbf{W}^T\mathbf{X} - \hat{\mathbf{Y}}\|_2^2$$

Discussion: What is the drawback of this solution?

Classification as Regression

- Limitation:
  - Put unnecessary requirements to the predicted output
  - May increase the fitting difficulty and lead to bad training result
- But why it is commonly used in practice?
  - Close-form solution, less storage for low-dimensional data
  - Quick update for incremental learning, distributed learning

$$\mathbf{w} = (\underbrace{\mathbf{X}\mathbf{X}^T}_{d \times d})^{-1}\underbrace{\mathbf{X}\hat{\mathbf{Y}}^T}_{d \times c}$$

Classification as Regression

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

If $\mathbf{X} = [\mathbf{X}_1, \mathbf{X}_2]$

$$\mathbf{X}\mathbf{X}^T = \mathbf{X}_1\mathbf{X}_1^T + \mathbf{X}_2\mathbf{X}_2^T$$

$$\mathbf{X}\hat{\mathbf{Y}}^T = \mathbf{X}_1\hat{\mathbf{Y}}^T + \mathbf{X}_2\hat{\mathbf{Y}}^T$$

Classification as Regression

- Discussion: Simple distributed learning
  - Consider the case: N = 10^9, d = 1000, c = 10
  - Data are equally distributed in two servers
  - Cost of sending raw data vs sending $\mathbf{X}\mathbf{X}^T$ and $\mathbf{X}\hat{\mathbf{Y}}^T$?

Regularized linear regression model

- Why regularization?
  - Avoid overfitting
  - Enforce certain property of solution

$$\mathcal{L} = \sum_{i=1}^{N} \|\mathbf{w}^T \mathbf{x}_i - \hat{\mathbf{y}}_i\|_2^2 + \Omega(\mathbf{w})$$

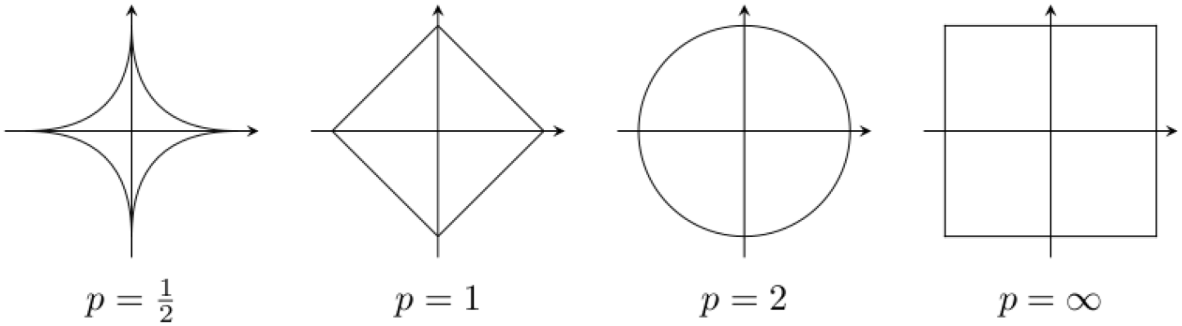
p-Norm

Regulariser $\Omega(\mathbf{w})$ is often in a form of p-norm.

Definition (p-norm)

Let $p \geq 0$ be a real number. The p-norm of $\mathbf{x} \in \mathbb{R}^d$ is

$$\|\,\mathbf{x}\,\|_p := \left(\sum_{j=1}^{d} |x^j|^p \right)^{1/p}$$



Images of $\|\mathbf{x}\|_p = 1$ (Consider d = 2)

Ridge regression

$$\mathcal{L} = \sum_{i=1}^{N} \|\mathbf{w}^T \mathbf{x}_i - \hat{\mathbf{y}}_i\|_2^2 + \lambda \|\mathbf{w}\|_2^2$$

How to understand $\|w\|_2^2$?
If we perturb the input x by adding a random noise vector ξ, i.e. $\mathbf{x}' = \mathbf{x} + \xi$
the prediction will become $\mathbf{w}^T(\mathbf{x} + \xi)$ which is a random variable.
The variance of $\mathbf{w}^T(\mathbf{x} + \xi)$ is $\|w\|_2^2$, if $\xi \sim \mathcal{N}(0, \mathbf{I})$

In plain English, it measures the sensitivity of the regressor w.r.t the perturbation of inputs. Minimizing it reflects our expectation of a smooth predictor.

Ridge regression: Solution

Practice: Derive the solution of ridge regression

$$\mathcal{L} = \sum_{i=1}^{N} \|\mathbf{w}^T \mathbf{x}_i - \hat{\mathbf{y}}_i\|_2^2 + \lambda \|\mathbf{w}\|_2^2$$

$$\text{Hint: } \|\mathbf{w}\|_2^2 = \mathbf{w}^T \mathbf{w}$$

Ridge regression: Solution

Solution

$$\mathcal{L} = \sum_{i=1}^{N} \|\mathbf{w}^T \mathbf{x}_i - \hat{\mathbf{y}}_i\|_2^2 + \lambda \|\mathbf{w}\|_2^2$$

$$\mathbf{w} = (\mathbf{X}\mathbf{X}^T + \lambda \mathbf{I})^{-1} \mathbf{X} \hat{\mathbf{y}}^T$$

Compare with the solution of linear regression

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

Discussion

- An issue I didn't tell you about the linear regression solution

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

What if $\mathbf{X}\mathbf{X}^T$ is not invertible?

- When it happens, it means there are multiple solutions to achieve minimal

$$\mathcal{L} = \sum_{i=1}^{N} \|\mathbf{w}^T\mathbf{x}_i - \hat{\mathbf{y}}_i\|_2^2$$

Discussion

- When it happens, it means there are multiple solutions to achieve minimal

$$\mathcal{L} = \sum_{i=1}^{N} \|\mathbf{w}^T \mathbf{x}_i - \hat{\mathbf{y}}_i\|_2^2$$

- Adding regularization makes it always invertible.
- It essentially provides a criterion for choosing optimal solution among multiple equivalent solutions of the first term.

$$\mathcal{L} = \sum_{i=1}^{N} \|\mathbf{w}^T \mathbf{x}_i - \hat{\mathbf{y}}_i\|_2^2 + \lambda \|\mathbf{w}\|_2^2$$

Lasso

$$\mathcal{L} = \sum_{i=1}^{N} \|\mathbf{w}^T \mathbf{x}_i - \hat{\mathbf{y}}_i\|_2^2 + \lambda \|\mathbf{w}\|_1$$

- L1 norm encourages sparse solution. This could be useful for understanding the impact of various factors, e.g. perform feature selection.
- Sometimes it can lead to an improved performance since it can suppressing noisy factors.
- Unfortunately, it does not have a close-form solution

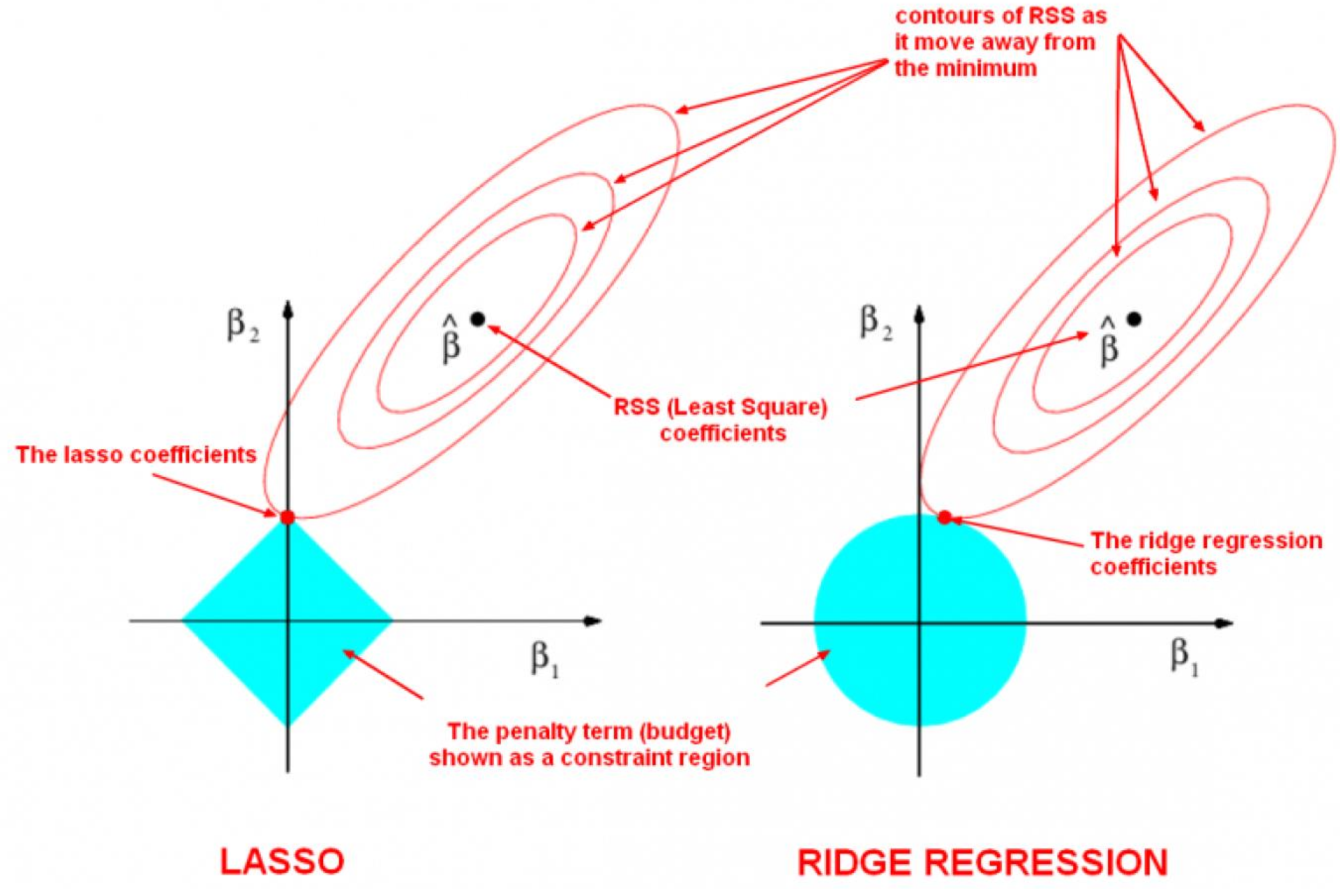
Sparse vs Dense solution

- Sparse solution: most parameters are 0
- Why l1 norm enforces sparsity?
  - Let's consider an example in 2D

$$
\begin{aligned}
\mathcal{L} &= \sum_{i=1}^{N} \|w_1 \mathbf{x}_i^1 + w_2 \mathbf{x}_i^2 - \hat{\mathbf{y}}_i\|_2^2 + \lambda \|\mathbf{w}\|_1 \\
&= a_1 (w_1 - c_1)^2 + a_2 (w_2 - c_2)^2 + e + \lambda \|\mathbf{w}\|_1 \\
&= f(w_1, w_2) + \lambda \|\mathbf{w}\|_1
\end{aligned} \tag{1}
$$

$f(w_1, w_2) = z$ is an ellipsoid equation. In other words, all w leading to the same z will form an ellipsoid in the parameter space

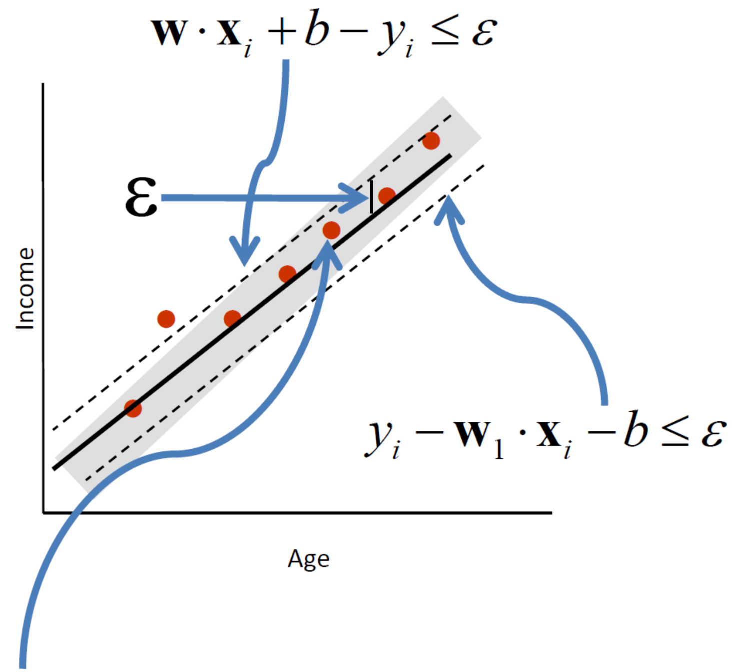
Sparse vs Dense solution



Support vector regression

- Key idea: if the fitting error is already small enough, do not make it smaller

A different way of measuring the fitting error

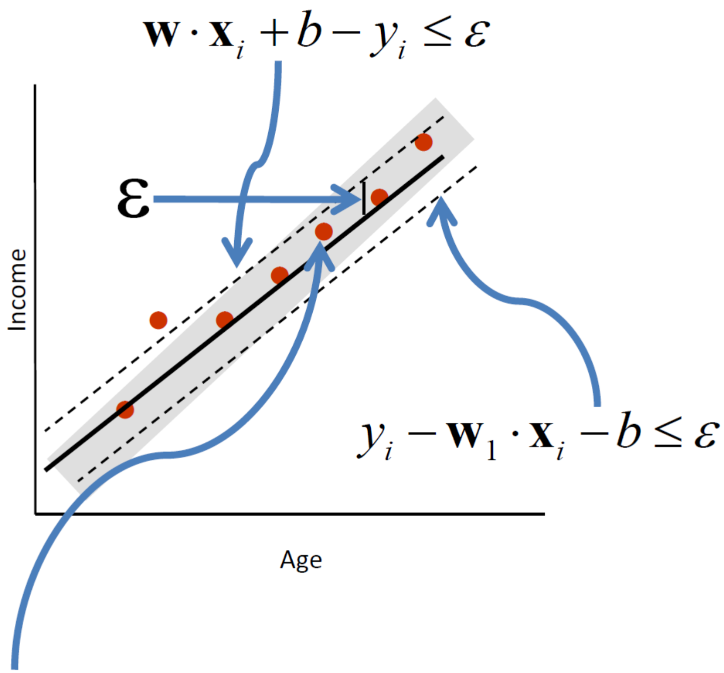


We do not care about errors as long as they are less than ε

Support vector regression: hard margin

- Idea: finding the linear so all the samples falling into the shadowed region

Assume linear parameterization $f(\mathbf{x}, \omega) = \mathbf{w} \cdot \mathbf{x} + b$



Support vector regression: hard margin

- Formulation

The problem can be written as a convex optimization problem

$$\min \frac{1}{2} \| \mathbf{w} \|^2$$

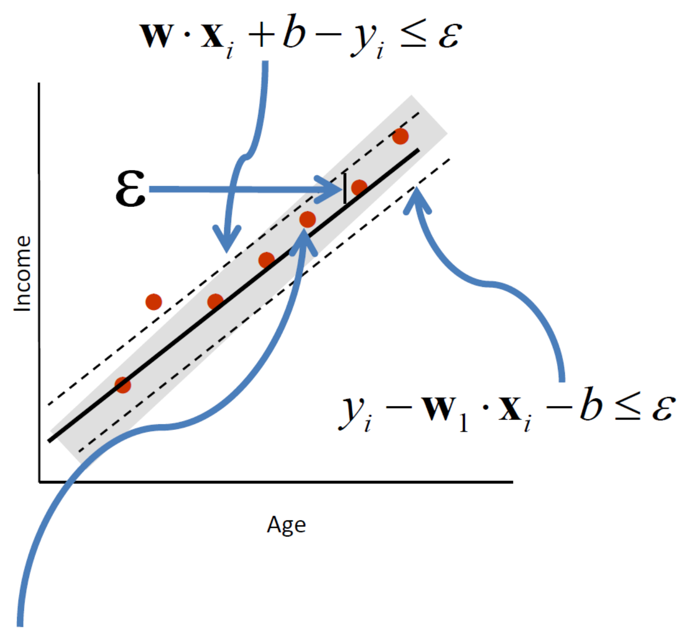
$$s.t.\ y_i - \mathbf{w}_1 \cdot \mathbf{x}_i - b \le \varepsilon;$$

$$\mathbf{w}_1 \cdot \mathbf{x}_i + b - y_i \le \varepsilon;$$

C: trade off the complexity

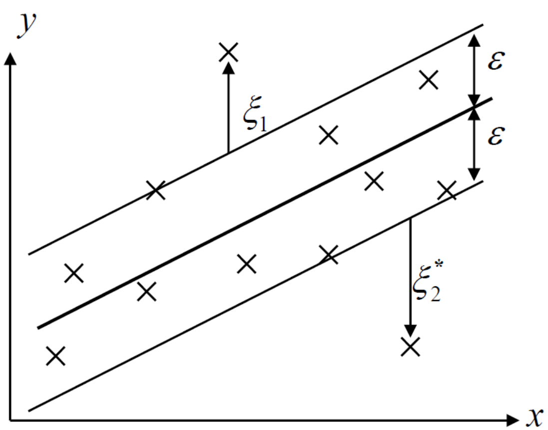
What if the problem is not feasible?

We can introduce slack variables
(similar to soft margin loss function).



We do not care about errors as long as they are less than ε

Support vector regression: soft-margin



Given training data

$$(\mathbf{x}_i, y_i) \quad i = 1, \ldots, m$$

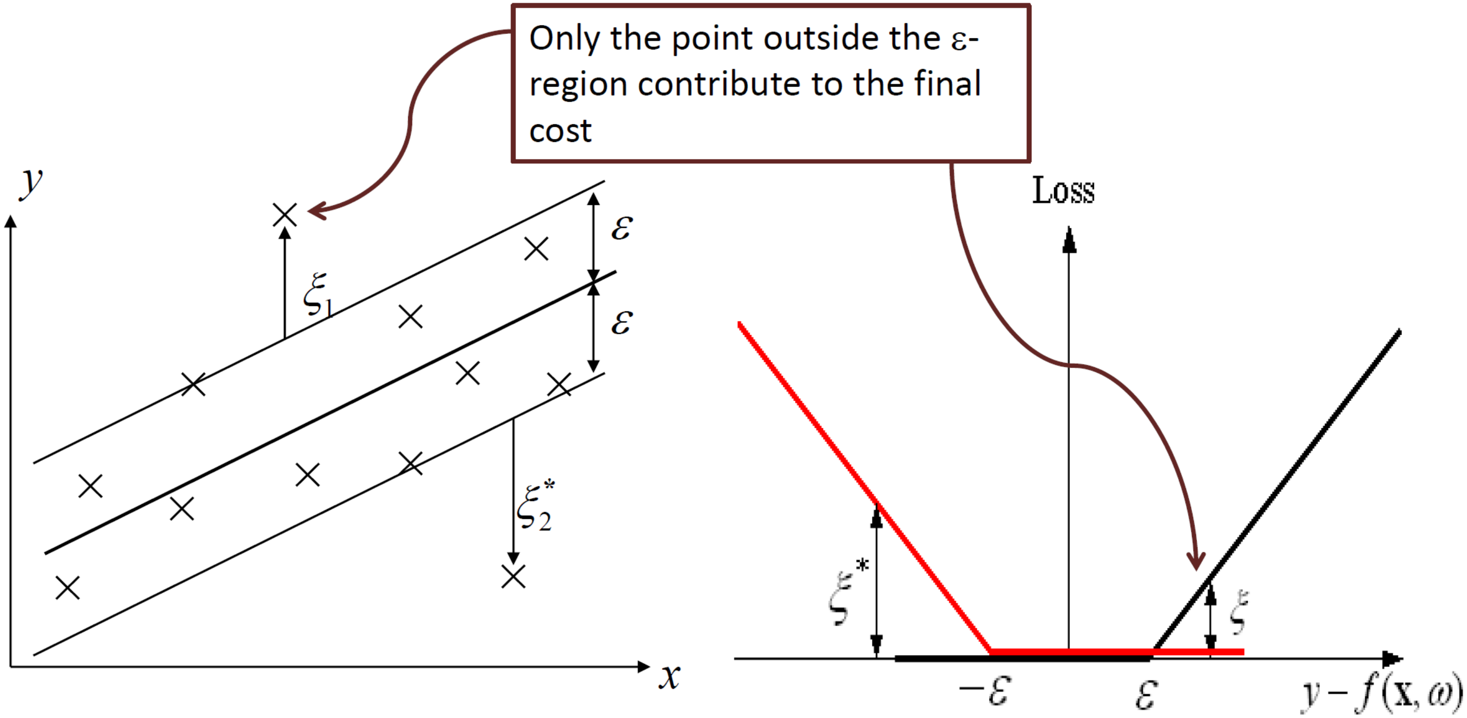
Minimize

$$\frac{1}{2} \| \mathbf{w} \|^2 + C \sum_{i=1}^{m} (\xi_i + \xi_i^*)$$

Under constraints

$$\begin{cases} y_i - (\mathbf{w} \cdot \mathbf{x}_i) - b \leq \varepsilon + \xi_i \\ (\mathbf{w} \cdot \mathbf{x}_i) + b - y_i \leq \varepsilon + \xi_i^* \\ \xi_i, \xi_i^* \geq 0, i = 1, \ldots, m \end{cases}$$

Support vector regression



$$L_\varepsilon(y, f(\mathbf{x},\omega)) = \max\left(|\, y - f(\mathbf{x},\omega)\,| - \varepsilon, 0\right)$$

Dual form of SVR

- Primal

$$\min \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{m}(\xi_i + \xi_i^*)$$

$$s.t.\begin{cases} y_i - (\mathbf{w}\cdot\mathbf{x}_i) - b \le \varepsilon + \xi_i \\ (\mathbf{w}\cdot\mathbf{x}_i) + b - y_i \le \varepsilon + \xi_i^* \\ \xi_i\ ,\xi_i^* \ge 0, i = 1,\ldots,m \end{cases}$$

- Dual

$$\max\begin{cases} \frac{1}{2}\sum_{i,j=1}^{m}(\alpha_i - \alpha_i^*)(\alpha_j - \alpha_j^*)\langle x_i, x_j\rangle \\ -\varepsilon\sum_{i=1}^{m}(\alpha_i + \alpha_i^*) + \sum_{i=1}^{m} y_i(\alpha_i - \alpha_i^*) \end{cases}$$

$$s.t.\sum_{i=1}^{m}(\alpha_i - \alpha_i^*) = 0;\ \ 0 \le \alpha_i, \alpha_i^* \le C$$

Primal variables: w for each feature dim	Dual variables: α, α* for each data point
Complexity: the dim of the input space	Complexity: Number of support vectors