Machine Learning

Lecture 2 : Introduction to Classification, Logistic Regression

Mohamad GHASSANY

EFREI PARIS

Classification

Classification examples

- Email: Spam / Not Spam?
- Online Transactions: Fraudulent (Yes/No)?
- Tumor: Malignant / Benign?
- Loan Demand (Credit Risk): Safe / Risky

Classification examples

- Email: Spam / Not Spam?
- Online Transactions: Fraudulent (Yes/No)?
- Tumor: Malignant / Benign?
- Loan Demand (Credit Risk): Safe / Risky

Classification: categorical output

- $y \in \{0, 1\}$
- 0: "Negative class"
- 1: "Positive Class"

- Email: Spam / Not Spam?
- Online Transactions: Fraudulent (Yes/No)?
- Tumor: Malignant / Benign?
- Loan Demand (Credit Risk): Safe / Risky

Classification: categorical output

- $y \in \{0, 1\}$
- 0: “Negative class”
- 1: “Positive Class”

.. and also multiclass classification

$$\text{Accuracy} = \frac{\text{Number of data points classified correctly}}{\text{all data points}}$$

Confusion Matrix

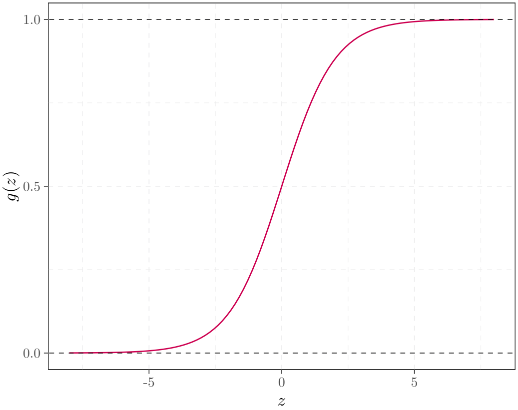
.. while in Regression (continuous output): Mean Squared Error (MSE).

Logistic Regression

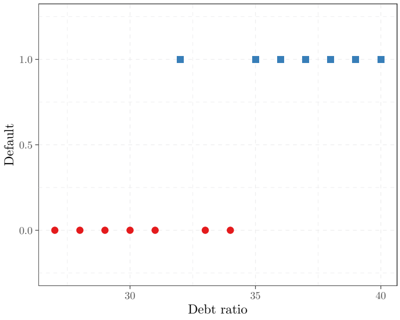
The logistic function (sigmoid)

$$g(z) = \frac{e^z}{1+e^z} = \frac{1}{1+e^{-z}}$$

$$g(z) = \frac{e^z}{1+e^z} = \frac{1}{1+e^{-z}}$$

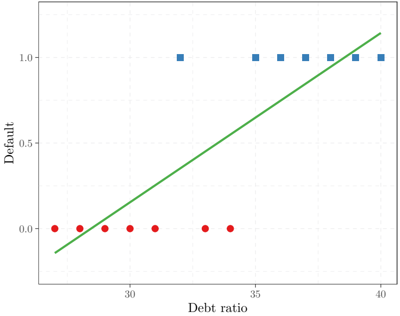


Logistic Regression: why not linear regression

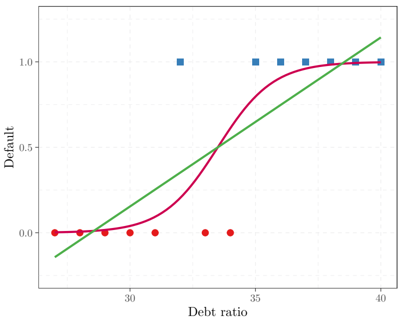


- $y \in \{0, 1\}$:
  - "0": Negative class (here no default)
  - "1": Positive class (here default)
- $f_\omega(x) = \omega' x$ can be > 1 ou < 0 !
- Ideally $0 \leqslant f_\omega(x) \leqslant 1$ s.t.:
  - If $f_\omega(x) \geqslant 0.5$, predict "$y = 1$"
  - If $f_\omega(x) < 0.5$, predict "$y = 0$"

- Let $f_\omega(x) = \omega' x$

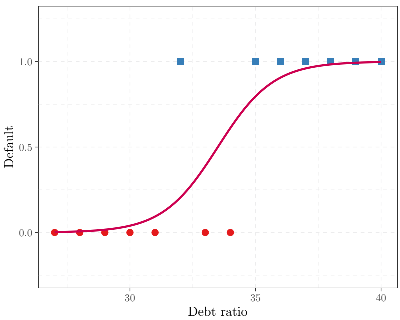


- Let $f_\omega(x) = \cancel{\omega' x} = g(\omega' x) = \dfrac{1}{1 + e^{-\omega' x}}$



Logistic Regression: intuition (3)

- $0 \leqslant g(\omega' x) \leqslant 1$
- $f_\omega(x) = g(\omega' x) =$ estimated probability that $y = 1$ on input x
- Probability that $y = 1$, given x, parameterized by ω
- $g(\omega' x) = p(y = 1 \mid x) = p(x)$
- $y \in \{0, 1\}$ so $p(y = 1 \mid x) + p(y = 0 \mid x) = 1$



logistic score

$$p(x) = p(y = 1 \mid x) = \frac{e^{\omega' x}}{1 + e^{\omega' x}} = \frac{1}{1 + e^{-\omega' x}}$$

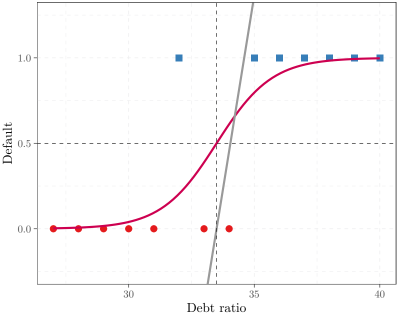
odds (côtes)

$$\frac{p(x)}{1 - p(x)} = e^{\omega' x}$$

log-odds or logit (logarithme des côtes)

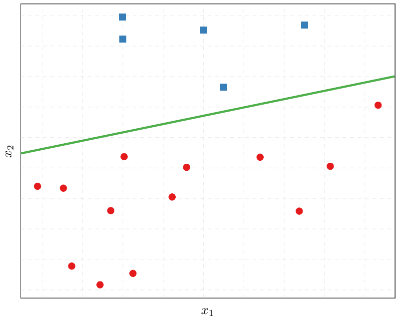
$$\log\left(\frac{p(x)}{1 - p(x)}\right) = \omega' x$$

Logistic Regression: decision boundary



- We predict "$y = 1$" if $p(x) \geqslant 0.5$ which means $\omega' x \geqslant 0$
- $\omega_0 + \omega_1 x \geqslant 0 \Rightarrow x \geqslant -\dfrac{\omega_0}{\omega_1}$

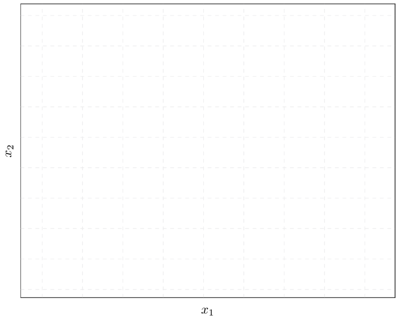
Logistic Regression: decision boundary (2 features)



- $p(x) = p(y = 1 \mid x) = f_\omega(x) = g(\omega' x)$
- Predict "$y = 1$" if $p(x) \geqslant 0.5$ which means $\omega' x \geqslant 0$
- $\omega_0 + \omega_1 x_1 + \omega_2 x_2 \geqslant 0$ So

$$x_2 \geqslant -\frac{\omega_1}{\omega_2} x_1 - \frac{\omega_0}{\omega_2}$$

Non linear decision boundaries

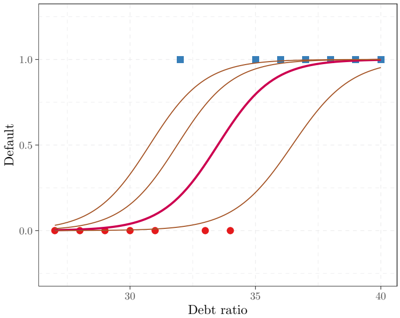


- Let $f_\omega(x) = g(\omega_0 + \omega_1 x_1 + \omega_2 x_2 + \omega_3 x_1^2 + \omega_4 x_2^2)$
- For example, predict "$y = 1$" if $-1 + x_1^2 + x_2^2 \geqslant 0$
- Or, $f_\omega(x) = g(\omega_0 + \omega_1 x_1 + \omega_2 x_2 + \omega_3 x_1^2 + \omega_4 x_1^2 x_2 + \omega_5 x_1^2 x_2^2 + \ldots)$

Logistic Regression: model estimation

Logistic Regression: model estimation[1]

- Parameters to estimate: $\omega = \{\omega_0, \omega_1\}$ if univariate
- $\omega = \{\omega_0, \omega_1, \ldots, \omega_p\}$ if multivariate with p features
- How to choose parameters ω?



[1] check: https://shinyserv.es/shiny/log-maximum-likelihood/, by Eduardo García Portugués

Cost function of simple linear regression

- Model: $f_\omega(x) = \omega_0 + \omega_1 x = \omega' x$
- Parameters: ω_0 and ω_1
- Cost function: $J(\omega_0, \omega_1) = \frac{1}{n}\sum_{i=1}^{n} \frac{1}{2}\left(f_\omega\left(x^{(i)}\right) - y^{(i)}\right)^2$
- Goal: $\min_{\omega_0,\omega_1} J(\omega_0, \omega_1)$

Non-convex in case of logistic regression !

Logistic Regression: how to estimate the parameters

- How to choose parameters ω?
- $y \in \{0, 1\}$, Let's assume:

$$p(y = 1 \mid x, \omega) = f_\omega(x)$$
$$p(y = 0 \mid x, \omega) = 1 - f_\omega(x)$$

Logistic Regression: how to estimate the parameters

- How to choose parameters ω?
- $y \in \{0, 1\}$, Let's assume:

$$p(y = 1 \mid x, \omega) = f_\omega(x)$$
$$p(y = 0 \mid x, \omega) = 1 - f_\omega(x)$$

- We represent $y \mid x, \omega \sim \mathcal{B}(f_\omega(x))$
- We can write:

$$p(y \mid x, \omega) = (f_\omega(x))^y \, (1 - f_\omega(x))^{1-y} \qquad y \in \{0, 1\}$$

- How to choose parameters ω?
- $y \in \{0, 1\}$, Let's assume:

$$p(y = 1 \mid x, \omega) = f_\omega(x)$$
$$p(y = 0 \mid x, \omega) = 1 - f_\omega(x)$$

- We represent $y \mid x, \omega \sim \mathcal{B}(f_\omega(x))$
- We can write:

$$p(y \mid x, \omega) = (f_\omega(x))^y \, (1 - f_\omega(x))^{1-y} \qquad y \in \{0, 1\}$$

- Given the n observations and assuming independance, we estimate ω by maximizing the likelihood:

$$\mathcal{L}(\omega) = \prod_{i=1}^{n} p\left(y^{(i)} \mid x^{(i)}, \omega\right)$$

- The likelihood:

$$\begin{aligned}
\mathcal{L}(\omega) &= \prod_{i=1}^{n} p\left(y^{(i)} \mid x^{(i)}, \omega\right) \\
&= \prod_{i=1}^{n} \left(f_\omega\left(x^{(i)}\right)\right)^{y^{(i)}} \left(1 - f_\omega\left(x^{(i)}\right)\right)^{1-y^{(i)}}
\end{aligned}$$

- The likelihood:

$$\mathcal{L}(\omega) = \prod_{i=1}^{n} p\left(y^{(i)} \mid x^{(i)}, \omega\right)$$
$$= \prod_{i=1}^{n} \left(f_\omega\left(x^{(i)}\right)\right)^{y^{(i)}} \left(1 - f_\omega\left(x^{(i)}\right)\right)^{1-y^{(i)}}$$

- Maximizing the likelihood is same as maximizing its log:

$$\ell(\omega) = \log\left(\mathcal{L}(\omega)\right)$$
$$= \sum_{i=1}^{n} y^{(i)} \log f_\omega\left(x^{(i)}\right) + \left(1 - y^{(i)}\right) \log\left(1 - f_\omega\left(x^{(i)}\right)\right)$$

- Maximizing $\ell(\omega)$ is same as minimizing: $-\frac{1}{n}\ell(\omega)$

Logistic Regression: model estimation

- The likelihood:

$$
\begin{aligned}
\mathcal{L}(\omega) &= \prod_{i=1}^{n} p\left(y^{(i)} \mid x^{(i)}, \omega\right) \\
&= \prod_{i=1}^{n} \left(f_\omega\left(x^{(i)}\right)\right)^{y^{(i)}} \left(1 - f_\omega\left(x^{(i)}\right)\right)^{1-y^{(i)}}
\end{aligned}
$$

- Maximizing the likelihood is same as maximizing its log:

$$
\begin{aligned}
\ell(\omega) &= \log\left(\mathcal{L}(\omega)\right) \\
&= \sum_{i=1}^{n} y^{(i)} \log f_\omega\left(x^{(i)}\right) + \left(1 - y^{(i)}\right) \log\left(1 - f_\omega\left(x^{(i)}\right)\right)
\end{aligned}
$$

- Maximizing $\ell(\omega)$ is same as minimizing: $-\frac{1}{n}\ell(\omega)$
- Let $J(\omega) = -\frac{1}{n}\ell(\omega)$, a convex cost function for the logistic regression model (known as *binary cross entropy*).

- **Goal**: Find ω s.t. $\omega = \text{argmin}_{\omega} J(\omega)$
- $J(\omega) = -\frac{1}{n}\sum_{i=1}^{n} y^{(i)} \log f_{\omega}\left(x^{(i)}\right) + \left(1 - y^{(i)}\right) \log\left(1 - f_{\omega}\left(x^{(i)}\right)\right)$
- Contrary to the linear regression, this cost function **does not** have an analytical solution. We need an optimization technique.

Logistic Regression: optimization of the cost function

- **Goal**: Find ω s.t. $\omega = \text{argmin}_\omega J(\omega)$
- $J(\omega) = -\frac{1}{n}\sum_{i=1}^{n} y^{(i)} \log f_\omega\left(x^{(i)}\right) + \left(1 - y^{(i)}\right) \log\left(1 - f_\omega\left(x^{(i)}\right)\right)$
- Contrary to the linear regression, this cost function **does not** have an analytical solution. We need an optimization technique.

GD for logistic regression

- `initialize` ω 'randomly"
- `repeat until convergence`{

$$\omega_i^{new} = \omega_i^{old} - \alpha \frac{\partial J(\omega)}{\partial \omega_i}$$

`simultaneously for` $i = 0, \ldots, p$ }

Logistic Regression: optimization of the cost function

- Recall that $g(z) = \dfrac{e^z}{1+e^z} = \dfrac{1}{1+e^{-z}}$
- Notice that $g'(z) = g(z)(1-g(z))$

- Recall that $g(z) = \dfrac{e^z}{1+e^z} = \dfrac{1}{1+e^{-z}}$
- Notice that $g'(z) = g(z)(1-g(z))$
- $\dfrac{\partial J(\omega)}{\partial \omega_i} = (y - f_\omega(x))x_i$

- Recall that $g(z) = \frac{e^z}{1+e^z} = \frac{1}{1+e^{-z}}$
- Notice that $g'(z) = g(z)(1-g(z))$
- $\frac{\partial J(\omega)}{\partial \omega_i} = (y - f_\omega(x))x_i$

GD for logistic regression

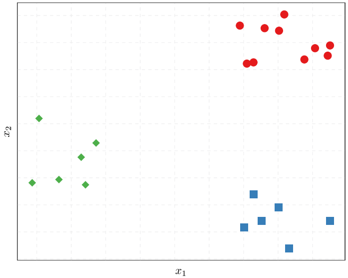
- `initialize` ω `randomly`
- `repeat until convergence`$\{$

$$\omega_i^{new} = \omega_i^{old} - \alpha \frac{1}{n} \sum_{i=1}^{n} \left(f_\omega\left(x^{(i)}\right) - y^{(i)}\right) .x_i^{(i)}$$

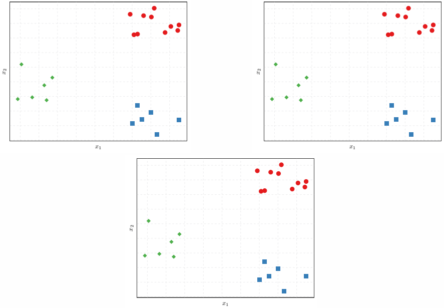
`simultaneously for` $i = 0, \ldots, p$ $\}$

Multi-class classification: One-vs-all

- Weather: Sunny, Cloudy, Rain, Snow
- Medical diagrams: Not ill, Cold, Flu
- News articles: Sport, Education, Technology, Politics



- $f_{\omega}^{(i)}(x) = P(y = 1|x, \omega) \quad$ for $i = 1, 2, 3$
- Train a logistic regression classifier for each class i to predict the probability that $y = i$
- On a new input x, to make a prediction, pick the class i that maximizes $f_{\omega}^{(i)}(x)$



- Very famous method and maybe the most used
- Adapted for a binary y
- Relation with linear regression
- Linear decision boundary, but can be non linear using other hypothesis
- Direct calculation of $p(y = 1 \mid x)$

Machine Learning

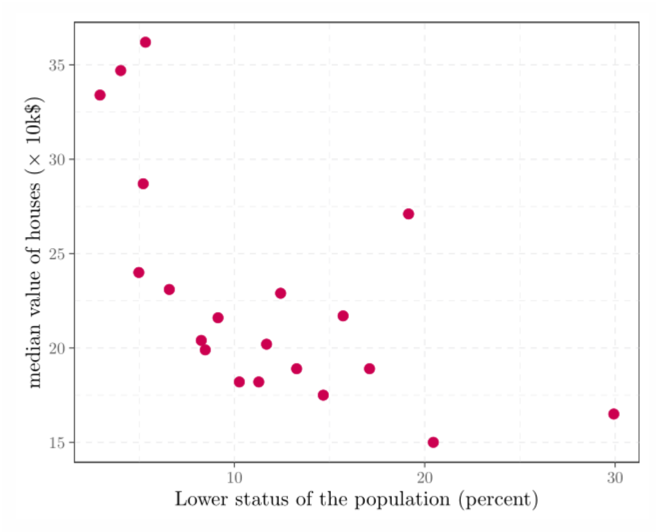
Lecture 2bis: Regularization

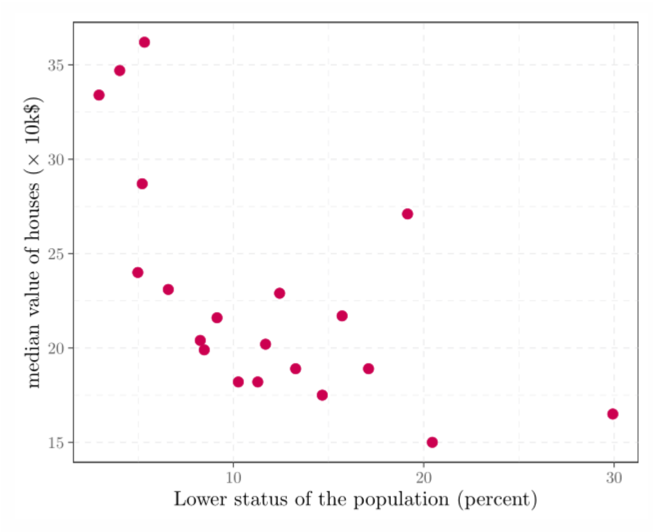
Mohamad GHASSANY

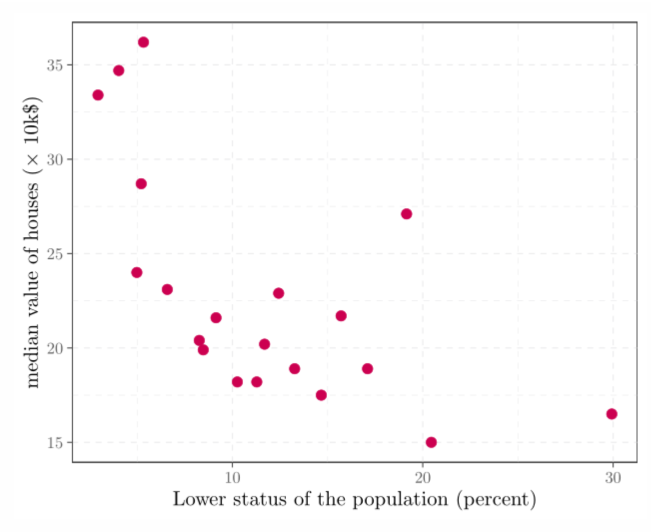
EFREI Paris

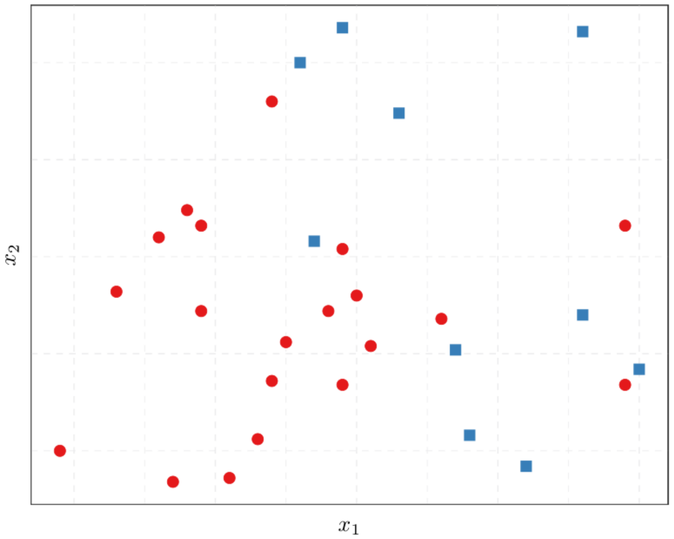
- Supervised learning
- Target variable type
- Hypothesis
- Cost function
- Optimization
- Sampling

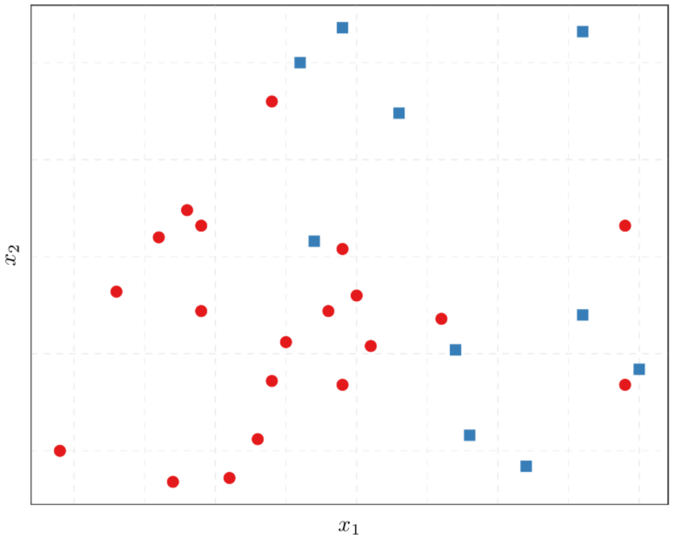
The problem of overfitting

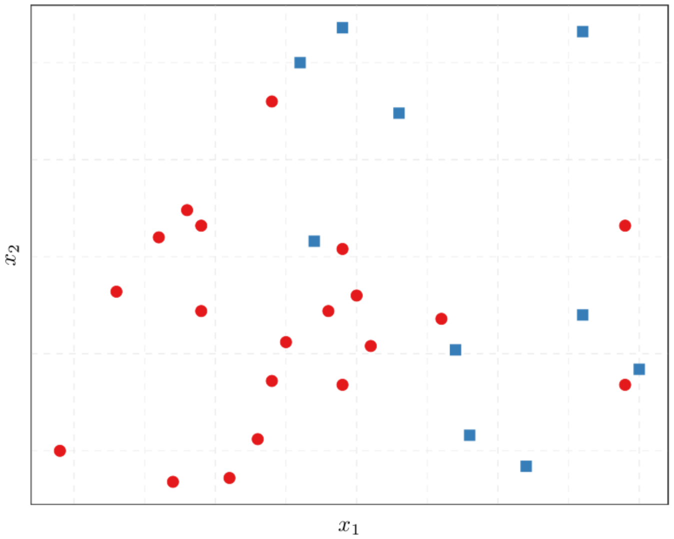

median value of houses (× 10k$)
Lower status of the population (percent)
35
30
25
20
15
10
20
30


median value of houses (× 10k$)
Lower status of the population (percent)
35
30
25
20
15
10
20
30

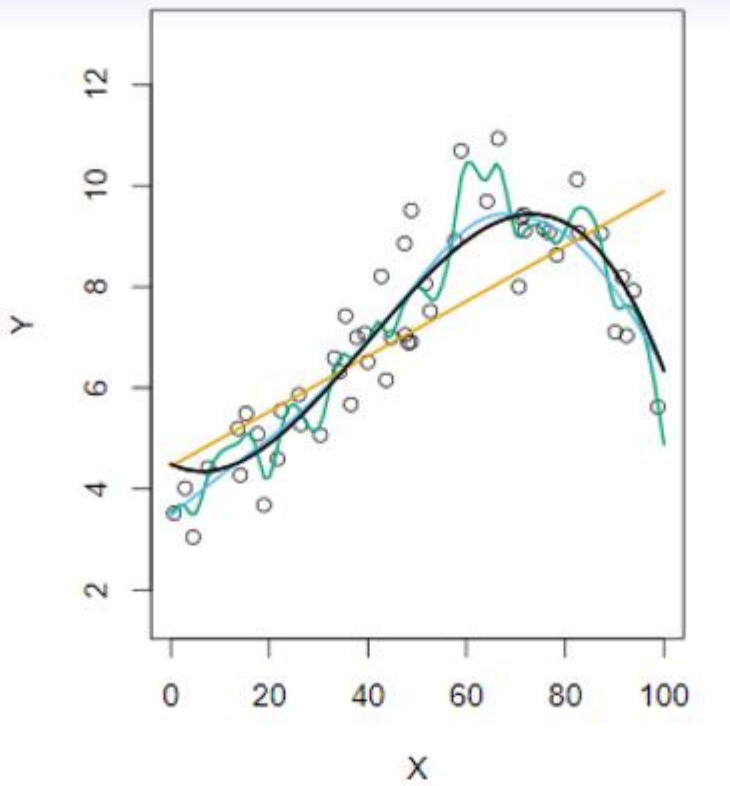

median value of houses (× 10k$)
Lower status of the population (percent)
35
30
25
20
15
10
20
30


x_2
x_1

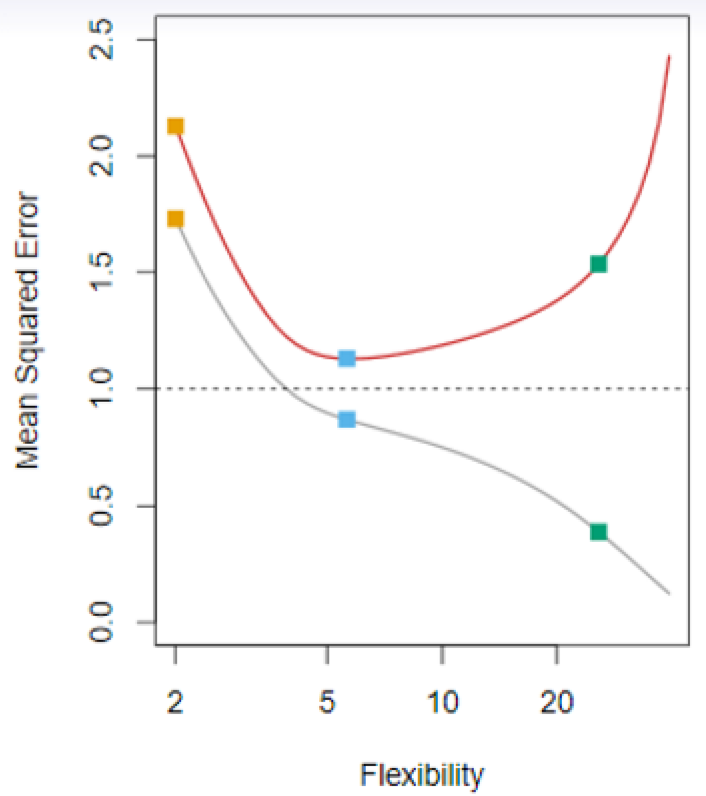

x_2
x_1


x_2
x_1

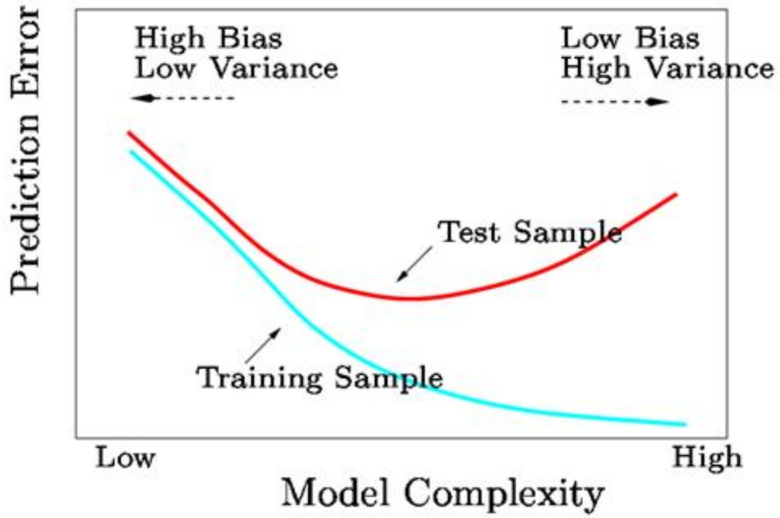
Underfitting/Overfitting Trade off



Black: Truth
Orange: Linear Estimate
Blue: smoothing spline
Green: smoothing spline (more flexible)



RED: Test MES
Grey: Training MSE
Dashed: Minimum possible test MSE (irreducible error)



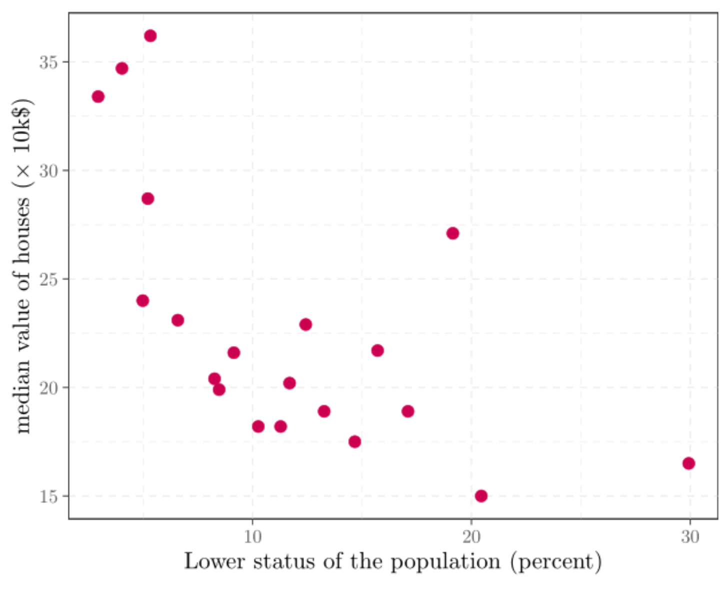
We must always keep this picture in mind when choosing a learning method. More flexible/complicated is not always better!

Adressing overfitting

- Options:

1. **Reduce number of features**
   - Manually
   - Model selection
2. **Regularization**
   - Keep all the features, but reduce magnitude/values of parameters ω_j
   - Works well when we have a lot of features, each of which contributes a bit to predicting y
   - It may not be immediately obvious why such a constraint should improve the fit, but it turns out that shrinking the coefficient estimates can significantly reduce their variance

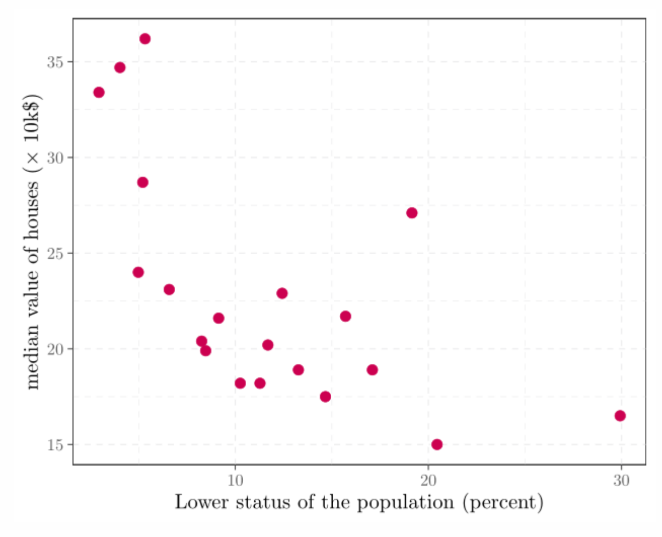
Regularization: Cost function



$$J(\omega) = \frac{1}{2n}\sum_{i=1}^{n}\left(f_{\omega}\left(x^{(i)}\right) - y^{(i)}\right)^2 +$$

- $J(\omega) = \frac{1}{2n}\left[\sum_{i=1}^{n}\left(f_{\omega}\left(x^{(i)}\right) - y^{(i)}\right)^2 + \lambda \sum_{j=1}^{p} w_j^2\right]$

- Choice of λ
  - What happens if λ is large ?



- Ridge Regression

$$J(\omega) = \frac{1}{2n}\sum_{i=1}^{n}\left(f_\omega\left(x^{(i)}\right) - y^{(i)}\right)^2 +$$

- Lasso

$$J(\omega) = \frac{1}{2n}\sum_{i=1}^{n}\left(f_\omega\left(x^{(i)}\right) - y^{(i)}\right)^2 +$$

- Neither ridge regression nor the lasso will universally dominate the other
- In general, one might expect the lasso to perform better when the response is a function of only a relatively small number of predictors.
- However, the number of predictors that is related to the response is never known *a priori* for real data sets.
- A technique such as cross-validation can be used in order to determine which approach is better on a particular dataset.
- **Cross-validation**: we choose a grid of λ values, and compute the cross-validation error rate for each value of λ. We then select the value for which the cross-validation error is smallest.

Regularization for Linear Regression

Regularized Linear Regression: GD

- $J(\omega) = \frac{1}{2n}\left[\sum_{i=1}^{n}\left(f_\omega\left(x^{(i)}\right) - y^{(i)}\right)^2 + \lambda \sum_{j=1}^{p} w_j^2\right]$
- $\min_{\omega} J(\omega)$

- Using **GD**:
  - `initialize` ω_j `randomly`
  - `repeat until convergence{`
    - $\omega_0^{new} = \omega_0^{old} - \alpha \frac{1}{n}\sum_{i=1}^{n}\left(f_\omega\left(x^{(i)}\right) - y^{(i)}\right)$
    - $\omega_j^{new} = \omega_j^{old} - \alpha \ \frac{1}{n}\sum_{i=1}^{n}\left(f_\omega\left(x^{(i)}\right) - y^{(i)}\right).x_j^{(i)}$ simultaneously for every $j = 1, \dots, p$

    }

- $\omega_j^{new} = \omega_j^{old}(1 - \alpha\frac{\lambda}{n}) - \alpha\frac{1}{n}\sum_{i=1}^{n}\left(f_\omega\left(x^{(i)}\right) - y^{(i)}\right).x^{(i)}$

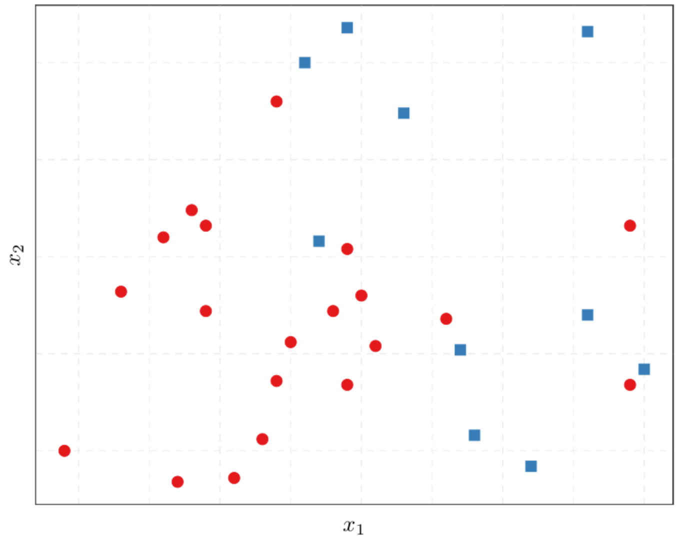
- $(1 - \alpha\frac{\lambda}{n})$ will always be less than 1

- $J(\omega) = \frac{1}{2n}\left[\sum_{i=1}^{n}\left(f_\omega\left(x^{(i)}\right) - y^{(i)}\right)^2 + \lambda \sum_{j=1}^{p} w_j^2\right]$
- $\min_{\omega} J(\omega)$
- $\omega = (X'X \qquad)^{-1} X'y$
- Using regularization takes care also of non-invertibility problem

Regularization for Logistic Regression

- $J(\omega) = -\frac{1}{n}\sum_{i=1}^{n} y^{(i)} \log f_\omega\left(x^{(i)}\right) + \left(1 - y^{(i)}\right)\log\left(1 - f_\omega\left(x^{(i)}\right)\right) + \frac{\lambda}{2n}\sum_{j=1}^{p} \omega_j^2$

- $\min_{\omega} J(\omega)$



- Using **GD**:
  - initialize ω_j randomly
  - repeat until convergence{
    - $\omega_0^{new} = \omega_0^{old} - \alpha\frac{1}{n}\sum_{i=1}^{n}\left(f_\omega\left(x^{(i)}\right) - y^{(i)}\right)$
    - $\omega_j^{new} = \omega_j^{old} - \alpha[\frac{1}{n}\sum_{i=1}^{n}\left(f_\omega\left(x^{(i)}\right) - y^{(i)}\right).x^{(i)} + \frac{\lambda}{n}\omega_j]$ simultaneously for every $j = 1, \dots, p$

    }