Workshop 3

COMP90051 Statistical Machine Learning

Semester 1, 2019

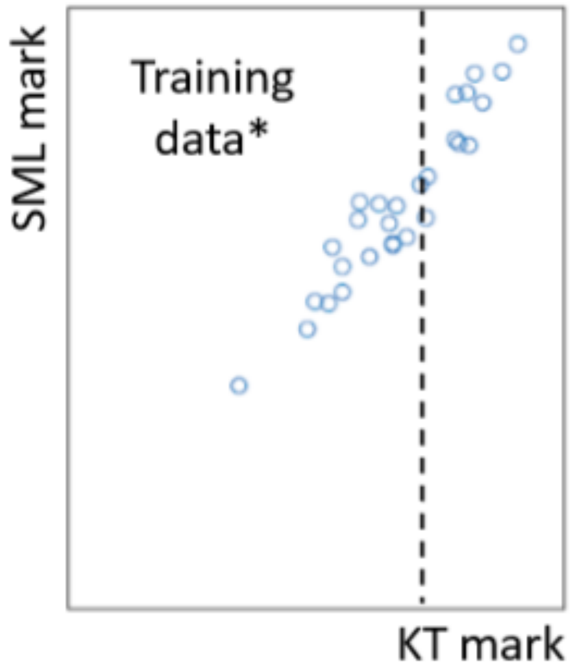
Learning Outcomes

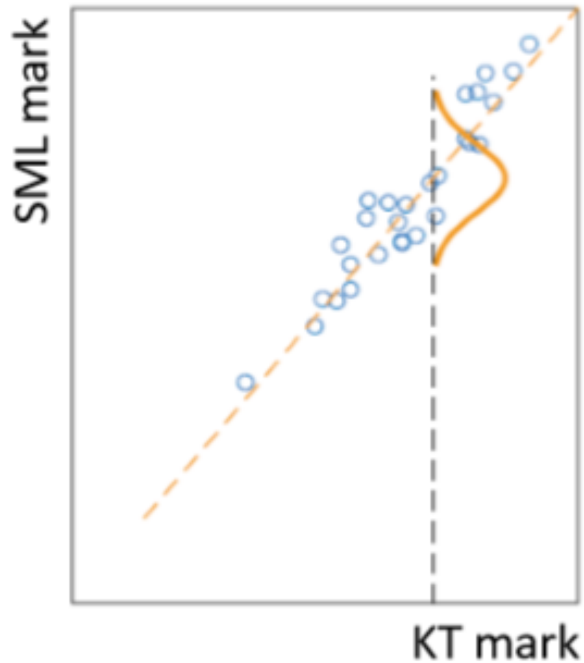
At the end of this workshop you should:

1. Be able to implement linear regression using analytic solution.
2. Be able to apply basis expansion to turn linear regression into polynomial regression

Data is noisy!

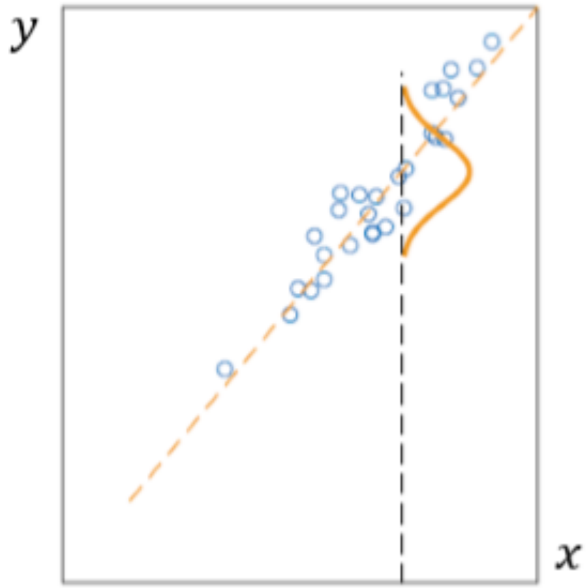
Example: predict mark for Statistical Machine Learning (SML) from mark for Knowledge Technologies (KT)





* synthetic data :)

Regression as a probabilistic model



- Assume a probabilistic model: $Y = \boldsymbol{X}'\boldsymbol{w} + \varepsilon$
  - Here $\boldsymbol{X}, Y$ and ε are r.v.'s
  - Variable ε encodes noise
- Next, assume Gaussian noise (indep. of $\boldsymbol{X}$): $\varepsilon \sim \mathcal{N}(0, \sigma^2)$

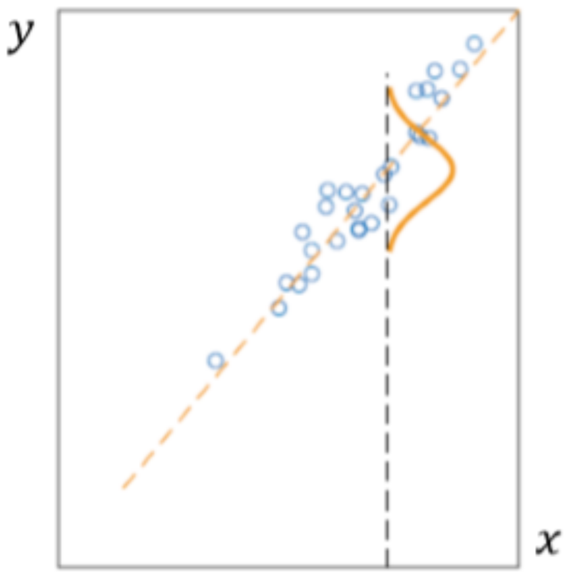
$Y \sim \mathcal{N}(w'X, \sigma^2)$

this is a squared error!

- Recall that $\mathcal{N}(x; \mu, \sigma^2) \equiv \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$
- Therefore

$$p_{\boldsymbol{w},\sigma^2}(y|\boldsymbol{x}) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{\boxed{(y - \boldsymbol{x}'\boldsymbol{w})^2}}{2\sigma^2}\right)$$

Parametric probabilistic model



- Using simplified notation, discriminative model is:

$$p(y|\boldsymbol{x}) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(y - \boldsymbol{x}'\boldsymbol{w})^2}{2\sigma^2}\right)$$

- Unknown parameters: $\boldsymbol{w}, \cancel{\sigma^2}$

- Given observed data $\{(\boldsymbol{X}_1, Y_1), \dots, (\boldsymbol{X}_n, Y_n)\}$, we want to find parameter values that "best" explain the data

- Maximum likelihood estimation: choose parameter values that maximise the probability of observed data

Maximum likelihood estimation

- Assuming independence of data points, the probability of data is

$$p(y_1, \dots, y_n | \boldsymbol{x}_1, \dots, \boldsymbol{x}_n) = \prod_{i=1}^{n} p(y_i | \boldsymbol{x}_i)$$

- For $p(y_i|\boldsymbol{x}_i) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(y_i - \boldsymbol{x}_i{}'\boldsymbol{w})^2}{2\sigma^2}\right)$

- "Log trick": Instead of maximising this quantity, we can maximise its logarithm (why?)

$$\sum_{i=1}^{n} \log p(y_i | \boldsymbol{x}_i) = -\frac{1}{2\sigma^2} \boxed{\sum_{i=1}^{n} (y_i - \boldsymbol{x}_i{}'\boldsymbol{w})^2} + C$$

here C doesn't depend on $\boldsymbol{w}$ (it's a constant)

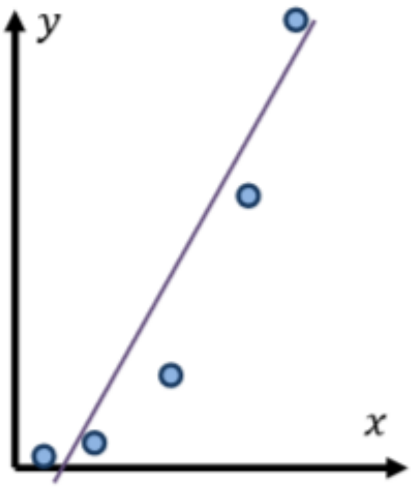
the sum of squared errors!

- Under this model, maximising log-likelihood as a function of $\boldsymbol{w}$ is equivalent to minimising the sum of squared errors

Basis expansion for linear regression

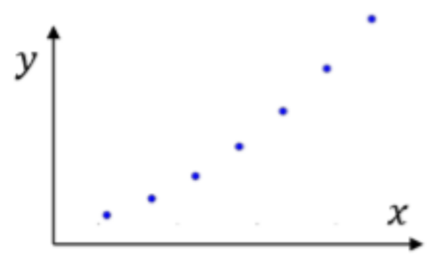
- Let’s take a step back. Back to linear regression and least squares
- Real data is likely to be non-linear
- What if we still wanted to use a linear regression?
  - It’s simple, easier to understand, computationally efficient, etc.
- How to marry non-linear data to a linear method?

If you can’t beat’em, join’em

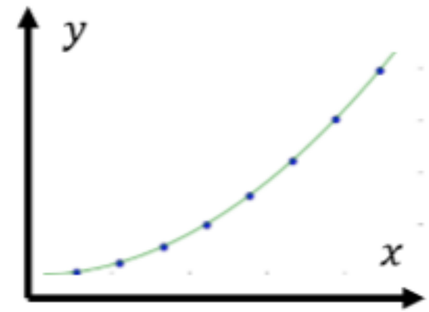


Transform the data

- The trick is to transform the data: Map data onto another features space, s.t. data is linear in that space
- Denote this transformation $\varphi: \mathbb{R}^m \to \mathbb{R}^k$. If $\boldsymbol{x}$ is the original set of features, $\varphi(\boldsymbol{x})$ denotes new feature set
- Example: suppose there is just one feature x, and the data is scattered around a parabola rather than a straight line



Example: Polynomial regression



- No worries, mate: define

$$\varphi_1(x) = x$$
$$\varphi_2(x) = x^2$$

- Next, apply linear regression to φ_1, φ_2

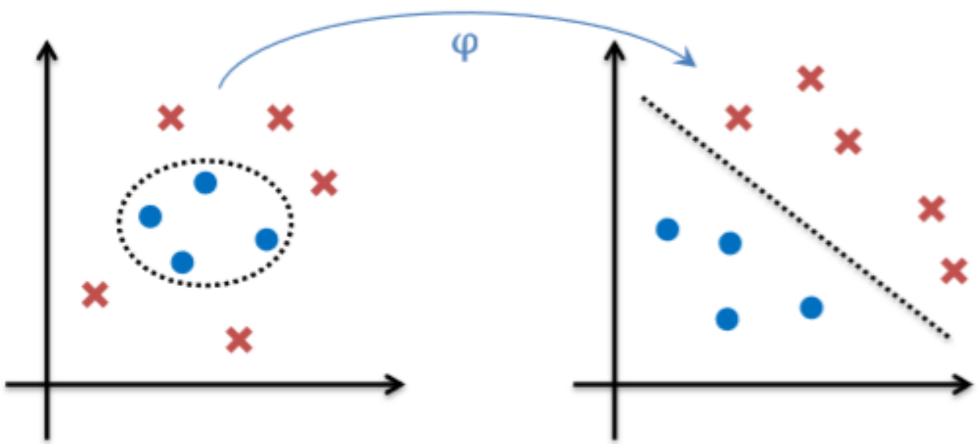
$$y = w_0 + w_1\varphi_1(x) + w_2\varphi_2(x) = w_0 + w_1 x + w_2 x^2$$

and here you have **quadratic regression**

- More generally, obtain **polynomial regression** if the new set of attributes are powers of x

Basis expansion

- Data transformation, also known as basis expansion, is a general technique
  - We'll see more examples throughout the course
- It can be applied for both regression and classification
- There are many possible choices of φ

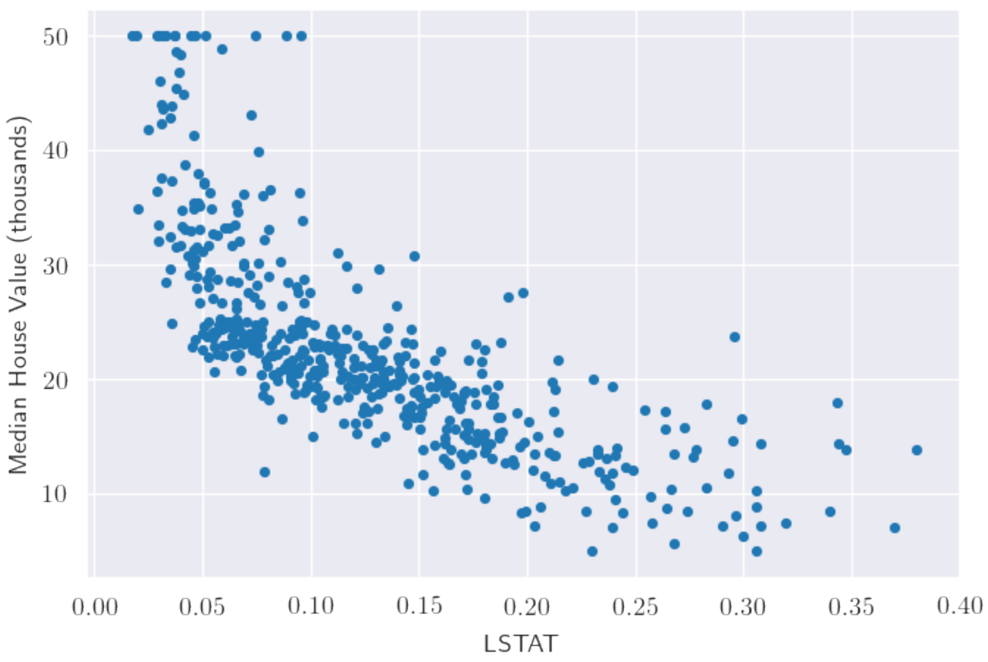


Dataset

- What is your goal?
  - Train models to predict house prices
  - Regression Task!
- What does the dataset look like?
  - Features: 13 different features
  - Target: median house in the given suburb(MEDV)
- Start with one feature.
  - LSTAT

Data Visualization

- Plot the data to see the relationship between the feature(s) and target.



What is the relationship between MEDV and LSTAT?

Split the dataset

- New Dataset:
  - One feature: LSTAT
  - One target: MEDV

- Split the dataset into 2 different sets.
  - What is the size of training set and test set?
  - Why are we doing this?

Linear regression

- Two solution approaches
  - Analytic solution
  - Approximate iterative solution

- Find the optimal weights w*

$$\mathbf{w}^* = \left[\mathbf{X}^\top \mathbf{X}\right]^{-1} \mathbf{X}^\top \mathbf{y}$$

Numpy

- A=np.array([[1,2],[3,4]])
- A = $\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$
- A.T = $\begin{bmatrix} 1 & 3 \\ 2 & 4 \end{bmatrix}$
- np.dot(A,B) = $\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \cdot \begin{bmatrix} a & b \\ c & d \end{bmatrix}$ = $\begin{bmatrix} 1 * a + 2 * c & 1 * b + 2 * d \\ 3 * a + 2 * c & 3 * b + 4 * d \end{bmatrix}$
- np.linalg.solve = ?
  - Try “np.linalg.solve?”, or “help(np.linalg.solve)”
  - Try this code, what do you get?

```
A = np.array([[1,2],[3,4]])
B = np.dot(A,A.T)
np.linalg.solve(A,B)
```

Make prediction

- Now, you got w*
- How to make prediction?
  - y = ?

- Now, you have a trained model, what's next?

Evaluation

- Choose an evaluation metric
  - Mean square error (MSE)
  - Train MSE and Test MSE, which we are more interested in?
- Make prediction with unseen data (X_test)
  - Why use unseen?
- Implement mean_square_error function
  - For more numpy, check numpy-basics.ipynb under workshop 1a

Solving using scikit-learn

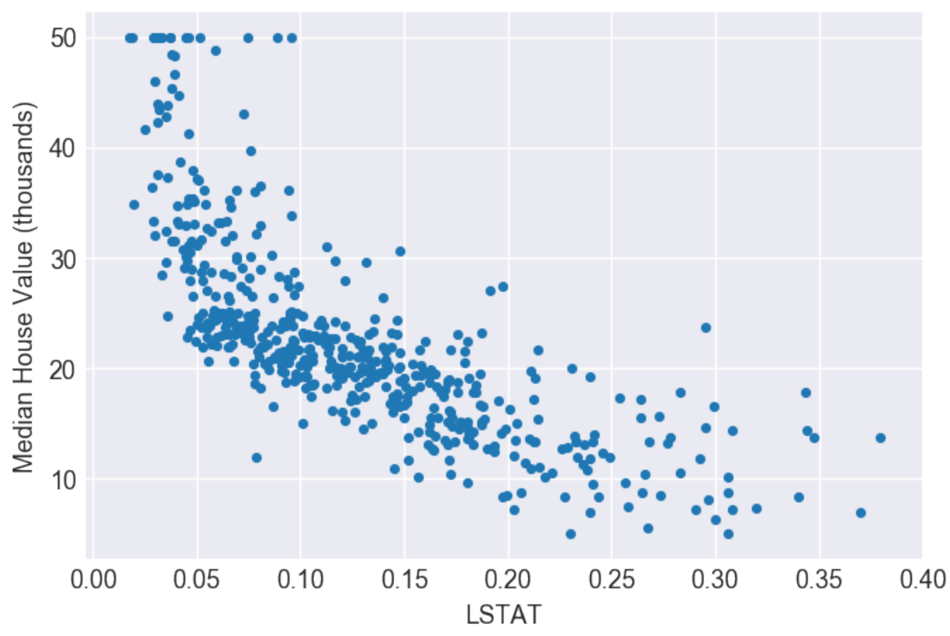
- Scikit-learn
  - **Scikit-learn** (formerly **scikits.learn**) is a free software machine learning library for the Python programming language.

- Use scikit-learn to check the results.
  - Import Linear Regression
  - Train and predict with fewer code

- How to improve the performance?

How to improve

- More data
  - Features: 1 -> 13
  - Train MSE: 38.63 -> ?
  - Test MSE: 38.00 -> ?
- The other models



Introducing Nonlinear Basis Functions

- Map the data onto a new space which the data is linear separable in there.
  - Use $\vec{\phi}(\mathbf{x})$ instead of x
- Polynomial regression

$$\vec{\phi}(x) = \left(1, x, x^2, \ldots, x^m\right)$$

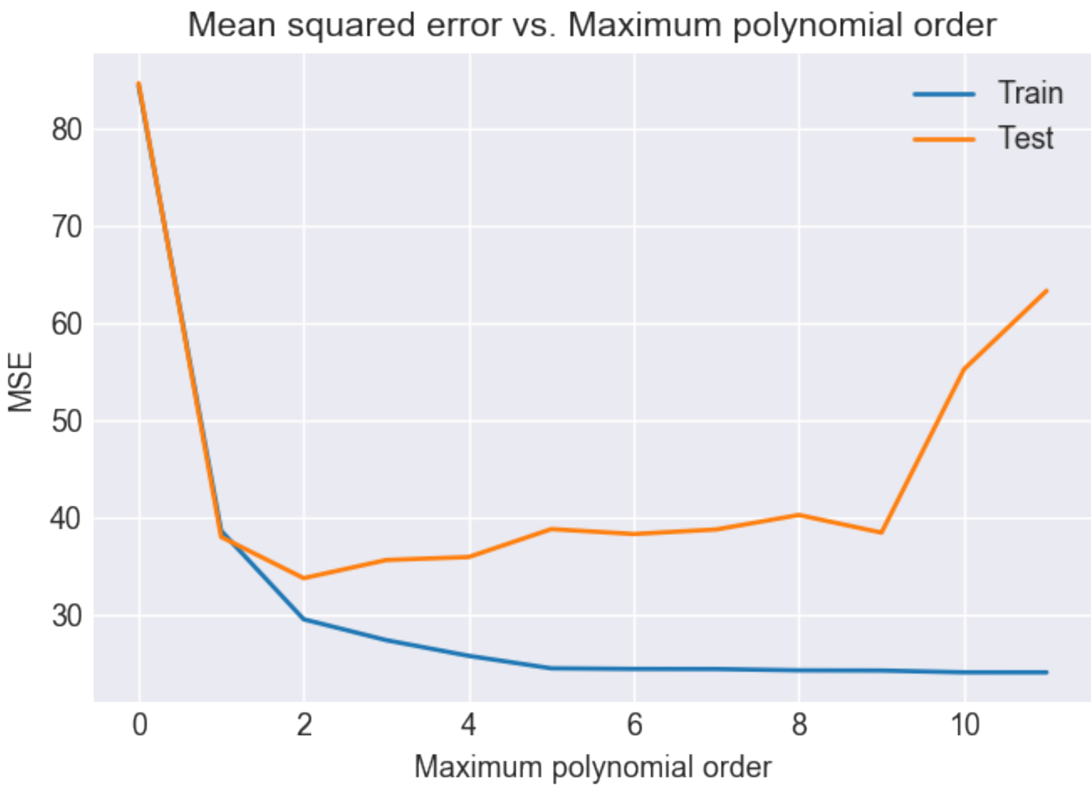
  - Add the x^2, x^3..x^m as the features
  - Build the design matrix

Polynomial Regression

- Start with order 3
  - Train MSE: 38.63 ->?
  - Test MSE: 38.00 -> ?
- What happened?
  - Discuss with your fellow students
- Higher order = better performance?
- What will happened to the model if you keep increasing the order?

Hyperparameters Tuning

- How to choose the m (maximum polynomial order)?
  - Grid search
  - Based on Train MSE or Test MSE?



Thank you!

- See you next week