Predictive Analytics

Week 3: K-Nearest Neighbours

Semester 2, 2018

Discipline of Business Analytics, The University of Sydney Business School

Week 3: K-Nearest Neighbours

1. K-Nearest Neighbours (KNN)
2. KNN properties
3. Comparison with linear regression
4. Summary

Reading: Chapter 3.5 of ISL.

Exercise questions: Chapter 3.7 of ISL, Q1, Q3 and Q4.

Introduction

In our first lecture, we introduced the distinction between parametric and nonparametric models. To review:

- A **parametric model** has a fixed number of parameters. Parametric models are faster to use and more interpretable, and make strong assumptions about the data.

- In a **nonparametric model**, the number of parameters grows with the size of the training data. Nonparametric methods are more flexible, but have larger variance and can be computationally infeasible for large datasets.

Introduction

In this lecture, we consider the K-nearest neighbours method, a nonparametric approach.

Studying this method will also allow you to learn and consolidate important concepts in statistical learning.

K-Nearest Neighbours (KNN)

K-nearest neighbours

- K-nearest neighbours can be used for regression or classification. This week we will be focusing on regression.
- KNN requires no distribution assumptions.
- Supervised or unsupervised learning algorithm?

K-nearest neighbours

We define the **K-nearest neighbours** (KNN) regression prediction for an input point $\boldsymbol{x}$ as

$$\widehat{f}(\boldsymbol{x}) = \frac{1}{k} \sum_{\boldsymbol{x}_i \in N_k(\boldsymbol{x}, \mathcal{D})} y_i,$$

for a training sample $\mathcal{D} = \{(y_i, \boldsymbol{x}_i)\}_{i=1}^N$.

Interpretation: the KNN prediction is the sample average of the response values for the k training observations which are closest to the query point $\boldsymbol{x}$.

K-nearest neighbours

To understand the intuition behind the method, suppose that $k = 1$ and there is an observation i for which $\boldsymbol{x}_i = \boldsymbol{x}$. Then,

$$E(Y|X = \boldsymbol{x}) = f(\boldsymbol{x}).$$

- That is, the KNN method can potentially approximate any regression function $f(\cdot)$ without making assumptions the form of this function.

- If there are input values in the training sample which are similar to $\boldsymbol{x}$ in some sense, the KNN will therefore be a low bias method.

- The bais and variance of KNN is dependent on the selection of k. More details will be discussed later.

Illustration

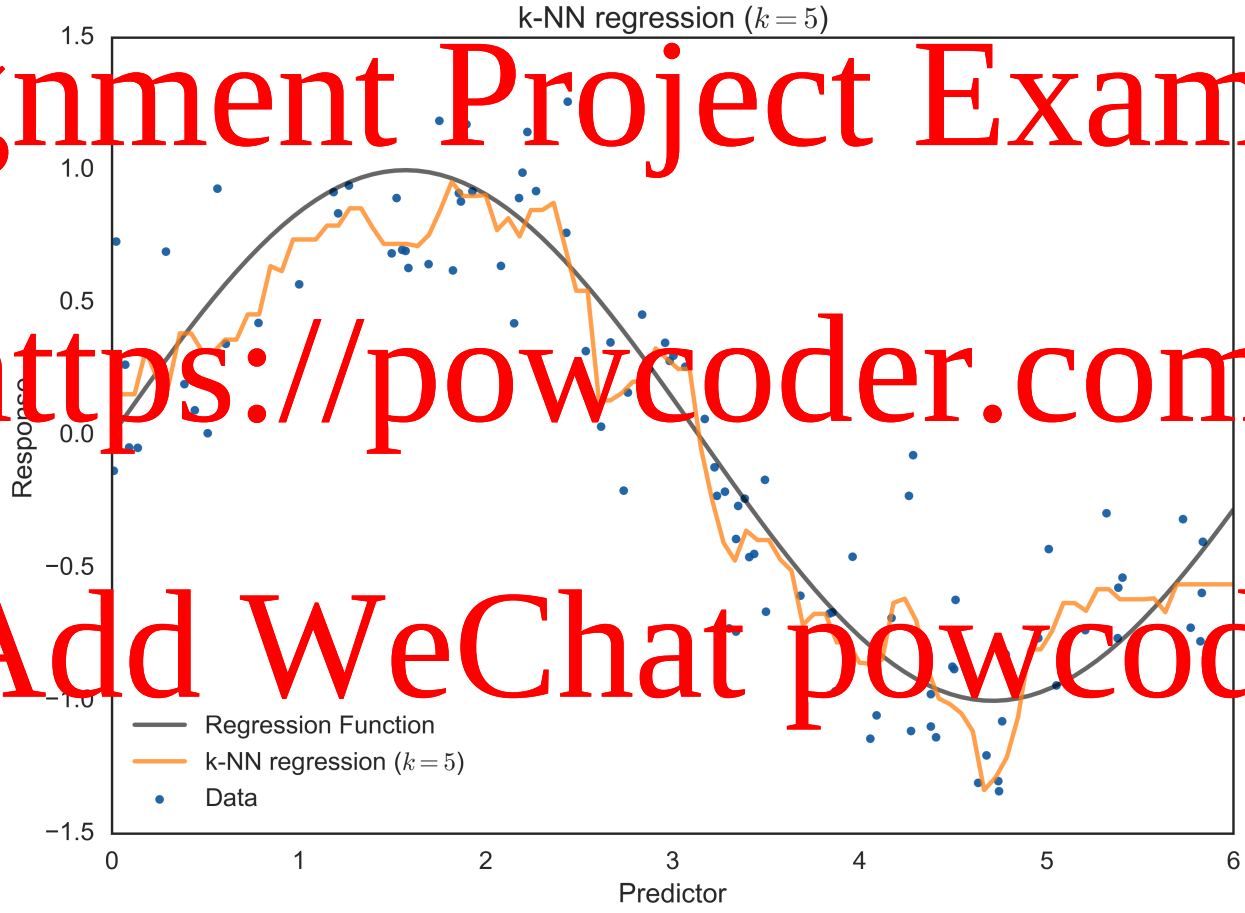
- We consider the regression model:

$$Y_i = \sin(X_i) + \varepsilon_i, \qquad \varepsilon_i \sim N(0, \sigma^2),$$

where we set $\sigma^2 = 0.1$ and restrict X to be between 0 and 6.

- In the next slide, we simulate $N = 100$ observations from the model and fit a KNN regression to this training sample.

Illustration ($k = 5$)



KNN properties

Modelling choices

$$\widehat{f}(\boldsymbol{x}) = \frac{1}{k} \sum_{\boldsymbol{x}_i \in N_k(\boldsymbol{x}, \mathcal{D})} y_i$$

for a training sample $\mathcal{D} = \{(y_i, \boldsymbol{x}_i)\}_{i=1}^{N}$.

The KNN method requires us to specify:

1. A distance metric.

2. The number of neighbours k.

3. The predictors.

Distance

A common distance measure is the Euclidean distance:

$$d(\boldsymbol{x}_i, \boldsymbol{x}_l) = \sqrt{\sum_{j=1}^{p} (x_{ij} - x_{lj})^2}$$

We will often use the notation $\|\boldsymbol{x}_i - \boldsymbol{x}_l\|_2$ which denotes the ℓ^2 (Euclidean) norm of the vector $\boldsymbol{x}_i - \boldsymbol{x}_l$.

Choosing the number of neighbours

- The number of neighbours k is a **hyperparameter**: we need to specify it prior to the learning process.
- We cannot use the training data to pick k since we would then always choose $k = 1$ and fit the data perfectly!
- Instead, we select k based on bias-variance trade-off considerations.
- We will use model selection and estimate the test error for each candidate value of k. We then select the value of k with the lowest error according to this criterion.

Overfitting and underfitting

- The model complexity decreases with the number of neighbours k.
- With a small k, the bias will be relatively small since the regression function evaluated at the neighbours $f(x_\ell)$ will be close to $f(x_0)$ (x_0 is prediction point). However, a small k means that we averaging only a few observations, leading to high variance $=>$ Tend to **overfitting**.
- As we increase k we reduce the variance, at the cost of higher bias. $=>$ Tend to **underfitting**.

Illustration

Play the animation to see how the estimates change as we vary k.

Curse of dimensionality

- The KNN method is subject to a **curse of dimensionality**: it breaks down with high-dimensional inputs (high p).

- The reason is that as we increase the number of predictors, it becomes exponentially more difficult to find training observations are reasonably local to the prediction point $\boldsymbol{x}$.

- The curse of dimensionality is a prevalent problem with nonparametric methods.

Curse of dimensionality

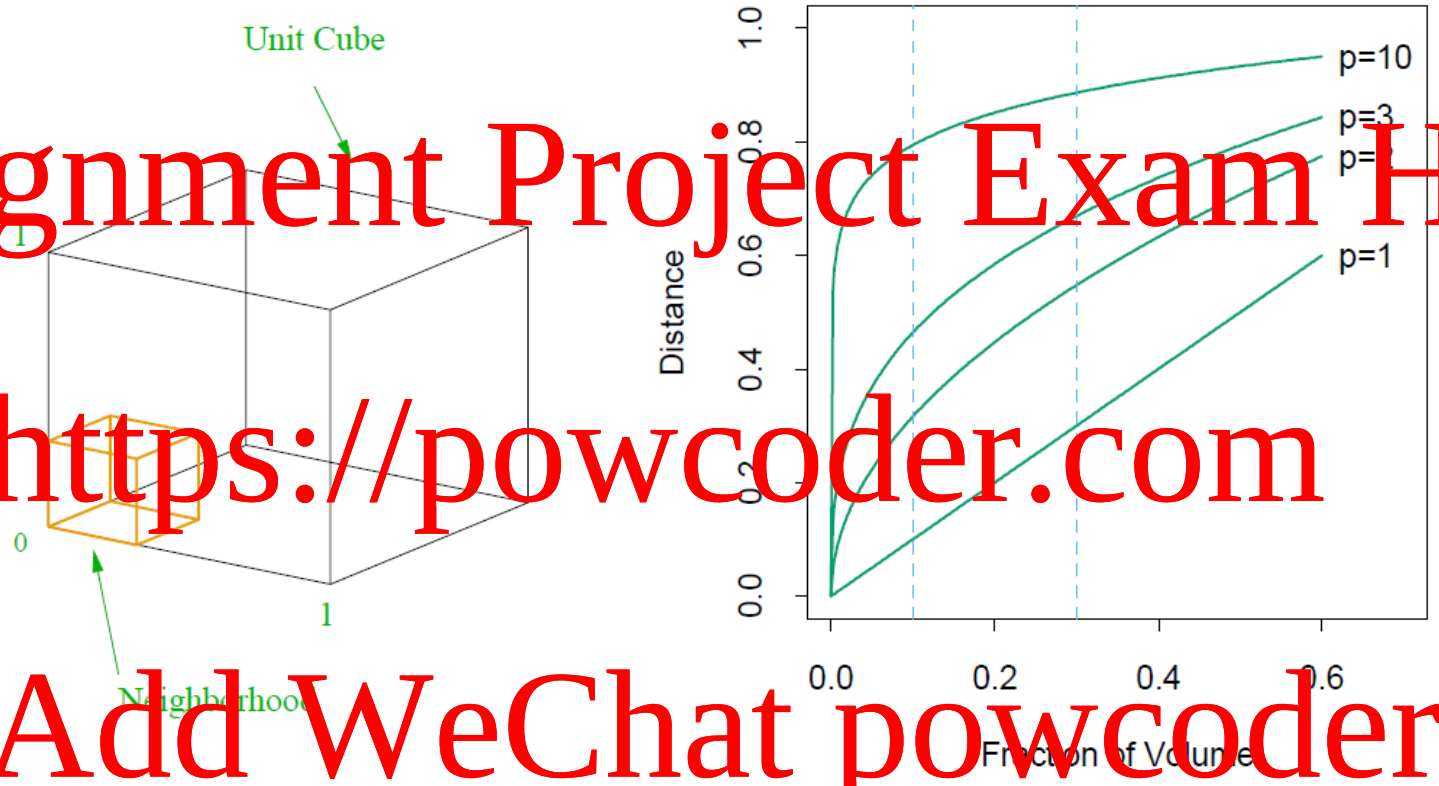


FIGURE 2.6. *The curse of dimensionality is well illustrated by a subcubical neighborhood for uniform data in a unit cube. The figure on the right shows the side-length of the subcube needed to capture a fraction r of the volume of the data, for different dimensions p. In ten dimensions we need to cover 80% of the range of each coordinate to capture 10% of the data.*

(Figure from "The Elements of Statistical Learning")

Computational considerations

- The KNN method is a memory intensive method. It requires us to keep the entire training sample in the memory for computing predictions.

- Generating predictions is computationally costly. For each new input point, we need to compute distances to all the training points, and sort these values. That contrasts with linear regression, where computing predictions is cheap.

- Sorting algorithms have require a number of computations which is proportional to $N \log(N)$ on average, such that the KNN method does not scale well to large datasets.

Comparison with linear regression

Comparison with linear regression

Linear regression: $\widehat{f}(\boldsymbol{x}) = \widehat{\beta}_0 + \sum_{j=1}^{p} \widehat{\beta}_j x_j$

KNN: $\widehat{f}(\boldsymbol{x}) = \frac{1}{k} \sum_{\boldsymbol{x}_i \in N_k(\boldsymbol{x}, D)} y_i$

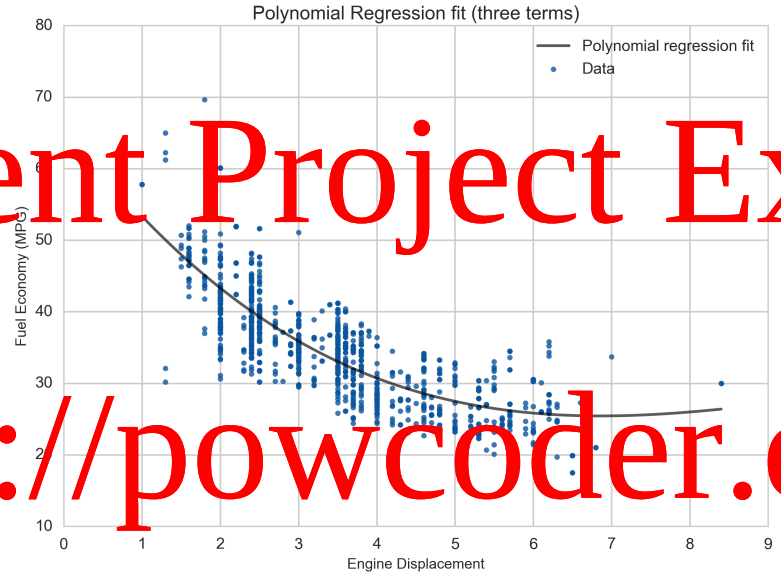
Comparison with linear regression

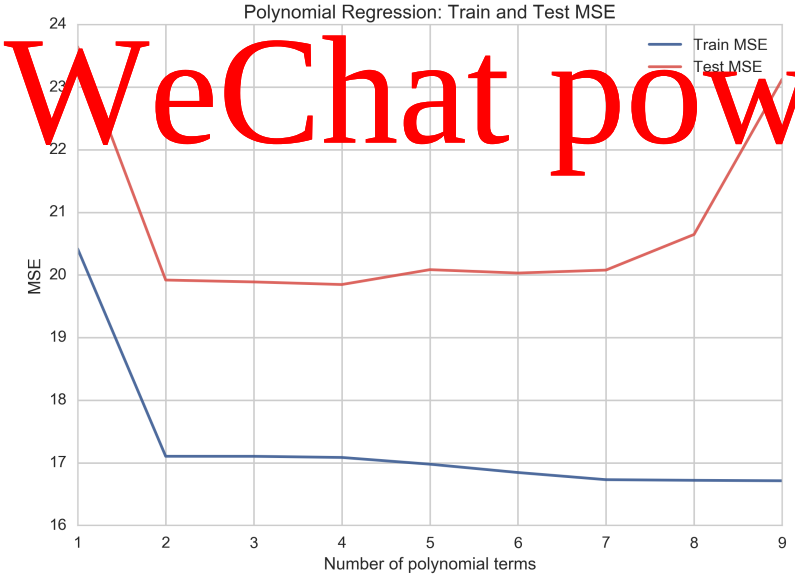
- The linear regression and KNN methods represent two opposite approaches to supervised learning.
- Linear regression assumes a linear form for the regression function $f(x)$. This assumption leads to stable predictions $\hat{f}(x)$, but the model can be highly inaccurate (high bias) if the assumption of linearity in the parameters is incorrect.
- KNN makes no structural assumptions about $f(x)$, leading to highly flexible predictions (low bias). But its predictions can be very unstable (high variance), since only a few training observations contribute to each prediction.

Linear regression and KNN: illustration

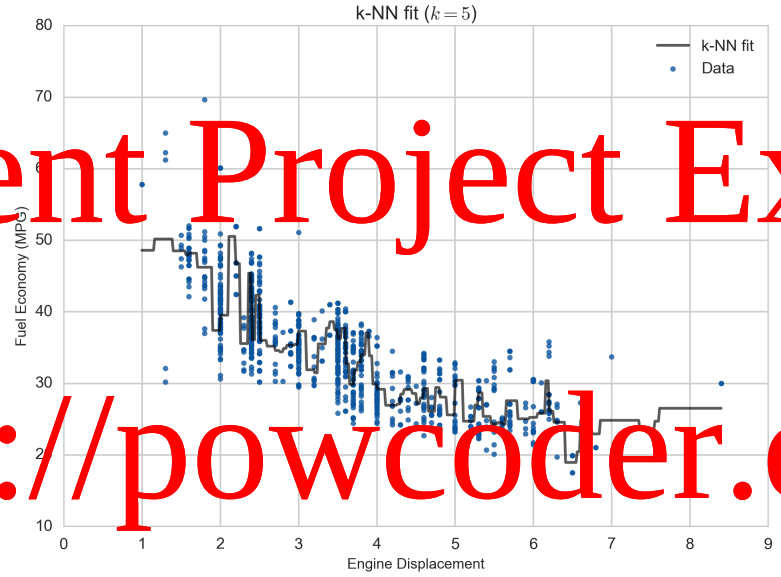
- We again consider the fuel economy dataset from lecture 1.
- Polynomial regressions and KNN are two alternatives for modeling the relationship between highway miles per gallon (MPG) of the car and engine displacement.
- The two methods have very similar test performance for reasonable choices of model complexity.

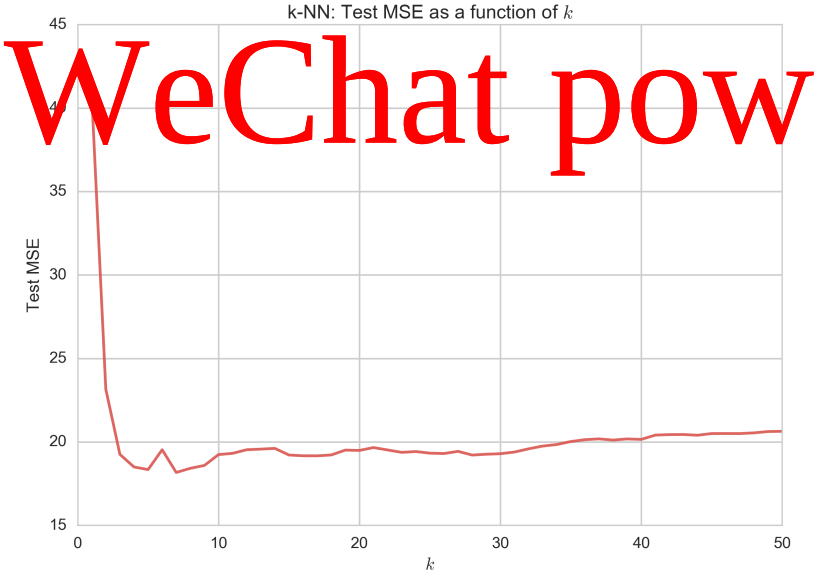
Linear regression and KNN: illustration





Linear regression and KNN: Illustration





Parametric or nonparametric?

- As always, the choice of method should be based on estimating the test performance with the available data.

- In general, we can say that a parametric model will outperform the nonparametric approach if the parametric form assumed for the regression function is close to the true form of f.

Advanced: Mahalanobis distance

The Euclidean distance only makes sense if the predictors are in the same scale. An alternative is use the normalised Euclidean distance:

$$d(\boldsymbol{x}_i, \boldsymbol{x}_l) = \sqrt{\sum_{j=1}^{p} \left(\frac{x_{ij} - x_{lj}}{s_{x_j}} \right)^2},$$

where s_{x_j} is the sample standard deviation of predictor j in the training sample.

A more complicated measure that often works better in practice is the Mahalanobis distance

$$d(\boldsymbol{x}_i, \boldsymbol{x}_l) = \sqrt{(\boldsymbol{x}_i - \boldsymbol{x}_l)^T S^{-1} (\boldsymbol{x}_i - \boldsymbol{x}_l)},$$

where S^{-1} is the sample covariance matrix of the predictors.

Advanced: effective number of parameters

- At first glance, it may difficult to relate our formal definition of nonparametric models with the KNN method, since it seems the KNN method only has one **hyperparameter**: the number of neighbours k.
- For this, we need the more advanced concept of the **effective number of parameters** of a model. For the KNN method, it is N/k
- The intuition is the following: if the neighbourhoods were nonoverlapping, there would be N/k neighbourhoods. We fit one parameter (the mean) for each neighbourhood.

Summary

Summary

- The KNN algorithm is a highly flexible method that does not explicitly assume a form for the regression function $f(X)$.
- A smaller value of k provides the more flexible model, with low bias but high variance.
- A larger value of k provides smoother estimates, at the cost of a less flexible approximation, with low variance but high bias.

Summary

However, the KNN method has disadvantages that we need to keep in mind:

- The estimate of the regression function can be very unstable, since it is the average of only a few points. This is the price that we pay for flexibility.

- Curse of dimensionality.

- The predictive performance is sensitive to noisy or irrelevant predictors.
- Generating predictions is computationally expensive.

Review questions

- How does the KNN method compute predictions?
- What is a hyperparameter?

- What is the Euclidean distance between two points?
- What is the curse of dimensionality?

- What are the advantages and disadvantages of the KNN method?