SSY 230, System Identification
Project 1: Estimating functions from noisy data

Yuxuan Xia
yuxuan.xia@chalmers.se
Emil Staf
emil.staf@chalmers.se

April 6, 2018

1 Linear regression functions

1.1 Estimation parameter validation

Verifying the parameter estimates and the uncertainty of the estimates requires some calculations by hand. In matrix form the parameter estimate is given by

$$\hat{\theta} = (X^\mathsf{T}X)^{-1}X^\mathsf{T}Y \tag{1}$$

where X is the regressor matrix and Y is the dependant variables. The uncertainty is given by

$$\text{cov}(\hat{\theta}) = \lambda^2(X^\mathsf{T}X)^{-1} \tag{2}$$

with λ^2 is the variance of the signal noise. Since λ^2 is unknown it has to be estimated using the data and an unbiased estimate is given by

$$s^2 = 2 * V(\hat{\theta})/(N-n) \tag{3}$$

where N is the number of feature vectors, n is the number of regressor dimensions and $V(\hat{\theta})$ is given by

$$V(\hat{\theta}) = \frac{1}{2}[Y - X\theta]^\mathsf{T}[Y - X\theta]. \tag{4}$$

The verification data is given by

$$X = \begin{bmatrix} 1 & 1 \\ 1 & 2 \\ 1 & 3 \end{bmatrix}, Y = \begin{bmatrix} 1.1 \\ 1.8 \\ 3.1 \end{bmatrix}. \tag{5}$$

Using (1) it is easy to verify that the estimate is $\hat{\theta} = [0, 1]^\mathsf{T}$. Using the found estimate (4) gives $V(\hat{\theta}) = 0.03$, which is used in (3) together with $N = 3$ and $n = 2$ and gives $s^2 = 0.06$. Now the covariance of the estimates is found using (2) with $s^2 = 0.06$ and

$$(X^\mathsf{T}X)^{-1} = \begin{bmatrix} 3 & 6 \\ 6 & 14 \end{bmatrix}^{-1} = \frac{1}{3 \cdot 14 - 6 \cdot 6}\begin{bmatrix} 14 & -6 \\ -6 & 3 \end{bmatrix} = \begin{bmatrix} \frac{7}{3} & -1 \\ -1 & \frac{1}{2} \end{bmatrix}. \tag{6}$$

This gives an estimated covariance of

$$s^2(X^\mathsf{T}X)^{-1} = 0.06\begin{bmatrix} \frac{7}{3} & -1 \\ -1 & \frac{1}{2} \end{bmatrix}. \tag{7}$$

1.2 Regularization verification

The loss function used in linear regression, using matrix notation, is

$$V_\lambda(\theta; X, Y) = \frac{1}{2}[(Y - X\theta)^\mathsf{T}(Y - X\theta) + \lambda\theta^\mathsf{T}\theta] \tag{8}$$

where λ is the regularization parameter and

$$X = \begin{bmatrix} | & | & & | \\ x_1 & x_2 & \cdots & x_d \\ | & | & & | \end{bmatrix}, Y = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{bmatrix} \tag{9}$$

where X $(N \times d)$, Y $(N \times m)$. Verify that $V_\lambda(\theta; X, Y)$ in (8) is the same as $V_0(\theta; X_2, Y_2)$, where

$$X_2 = \begin{bmatrix} X \\ \Lambda \end{bmatrix}, \Lambda = \begin{bmatrix} \sqrt{\lambda} & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & \sqrt{\lambda} \\ 0 & 0 & 0 \\ \vdots & \vdots & \vdots \end{bmatrix} Y_2 = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \\ 0 \\ \vdots \end{bmatrix} \tag{10}$$

where X_2 $(2N \times d)$, Y_2 $(2N \times m)$.

$$V_0(\theta; X_2, Y_2) = \frac{1}{2}(Y_2 - X_2\theta)^\mathsf{T}(Y_2 - X_2\theta) = \frac{1}{2}\begin{bmatrix} Y - X\theta \\ \mathbf{0} - \Lambda\theta \end{bmatrix}^\mathsf{T}\begin{bmatrix} Y - X\theta \\ \mathbf{0} - \Lambda\theta \end{bmatrix} = \tag{11}$$

$$= \frac{1}{2}[(Y - X\theta)^\mathsf{T}(Y - X\theta) + (\Lambda\theta)^\mathsf{T}(\Lambda\theta)] = \frac{1}{2}[(Y - X\theta)^\mathsf{T}(Y - X\theta) + \lambda\theta^\mathsf{T}\theta]. \tag{12}$$

Given a regressor matrix X and an output matrix Y, the linear least squares estimate with L2-Regularization can be obtained by minimizing the sum of squared residuals,

$$\hat{\theta} = \arg\min_\theta (Y - X\theta)^T(Y - X\theta) + \lambda\theta^T\theta. \tag{13}$$

The estimate $\hat{\theta}$ can be found by setting the derivative of (13) w.r.t. θ to zero,

$$\frac{d}{d\theta}\left((Y - X\theta)^T(Y - X\theta) + \lambda\theta^T\theta\right) = 0, \tag{14a}$$

$$-X^TY + X^TX\theta + \lambda\theta = 0, \tag{14b}$$

$$(X^TX + \lambda I)^{-1}X^TY = \hat{\theta}. \tag{14c}$$

It is easy to verify that, when $\lambda = 0$, $\hat{\theta}$ is equal to the linear least squares estimate without regularization, i.e.,

$$\hat{\theta} = (X^TX)^{-1}X^TY, \tag{15}$$

and that, when $\lambda \to +\infty$, $\hat{\theta} \to \mathbf{0}$.

1.3 Polynomial fitting validation

The estimated 3rd degree polynomial in Figure 1 is given by $\hat{y} = 0.4 + 5.56x + 0.41x - 0.096x$, close to the true function $y = 1 + 5x + 0.5x^2 - 0.1x^3$.

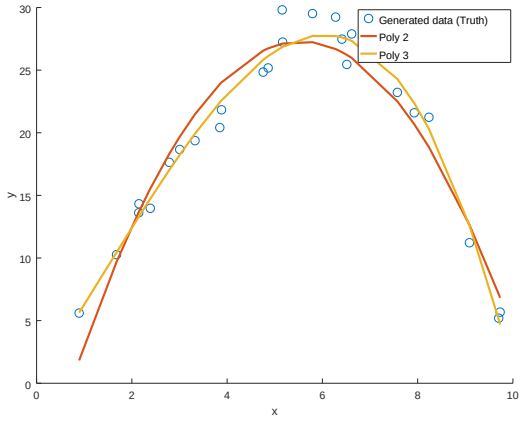


Figure 1: Estimated function using polynomial fitting. Each sample of regressor vector $\mathbf{x}$ is randomly drawn from uniform distribution $[0, 10]$. The true function is $\mathbf{y} = 1 + 5\mathbf{x} + 0.5\mathbf{x}^2 - 0.1\mathbf{x}^3$, the noise variance is set to 1, and the number of samples is 20.

1.4 (e) One dimensional model plotting

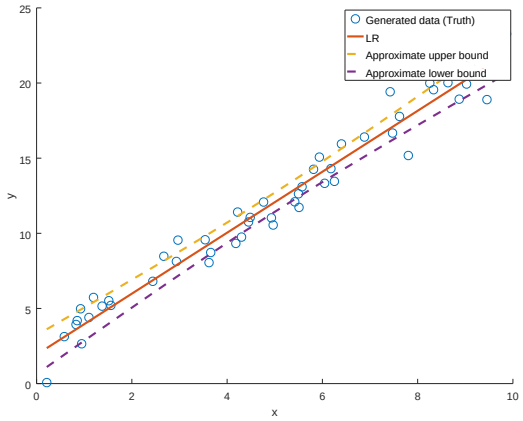


Figure 2: Estimated function using linear regression (LR). Uncertainty is illustrated as the region between the two dotted lines with confidence level 0.95. Each sample of regressor vector x is randomly drawn from uniform distribution $[0, 10]$. The true function is $y = 2 + 2x$, the noise variance is set to 2, and the number of samples is 20.

2 KNN-regression functions

The implemented KNN regressor is compared against linear regression in Figure 3.

2.1 KNN v.s. Linear regression

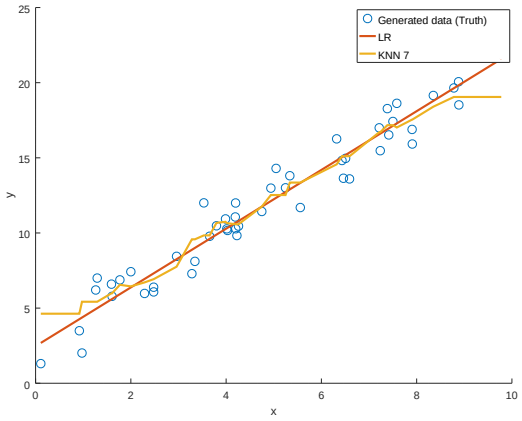


Figure 3: Estimated functions using linear regression (LR) and KNN regressor (K=7). Each sample of regressor vector $\mathbf{x}$ is randomly drawn from uniform distribution $[0, 10]$. The true function is $\mathbf{y} = 2 + 2\mathbf{x}$, the noise variance is set to 1, and the number of samples is 50.

3 Estimating one dimensional functions

3.1 Linear data

3.1.1 Linear regression model (constant + linear term)

Table 1: Estimation results of a linear regression model with only constant and linear terms.

Generated data size (N)	Constant estimation	Linear term estimation
N=10	1.62	0.53
N=100	1.49	0.50
N=1000	1.48	0.50
N=10000	1.52	0.50

As suggested by the results shown in Table 1, the estimates converge to the true function $\mathbf{y} = 1.5 + 0.5\mathbf{x}$ when the number of data increases.

3.1.2 Linear regression model (5th order polynomial)

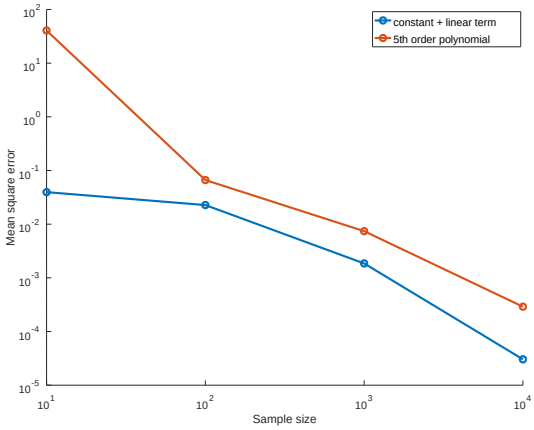


Figure 4: Model quality (mean square error) v.s. Number of data

As can be seen in Figure 4 the linear regression model with only constant and linear terms has better model quality in terms of mean square error than the linear regression model with polynomial terms. The difference between these two models, in general, becomes smaller as the number of data increases. This is because that the frequency of overfitting decreases as the data size increases.

3.1.3 Parameter variance validation via Monte Carlo simulation

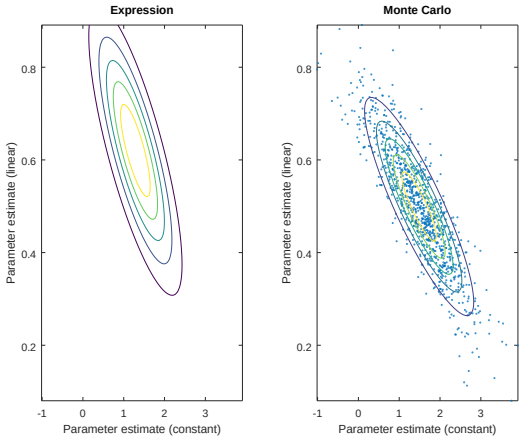


Figure 5: Contour plots of parameter estimates using 10 generated data over 1000 Monte Carlo trials. In the left subplot is the theoretical result on one realization (training data) and in the left plot is the distribution of found parameter estimates on different data sets.

In Figure 5, the expression uncertainty

3.1.4 KNN models

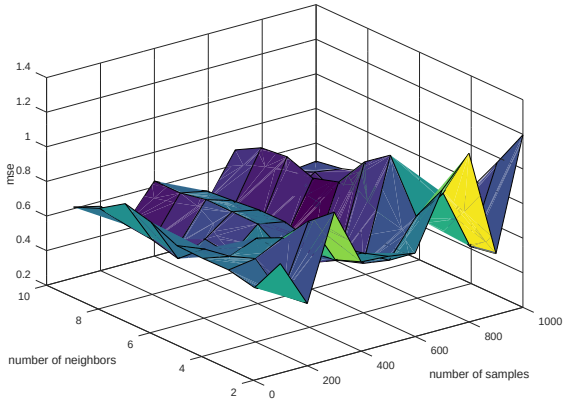


Figure 6: Model quality (mean square error) of KNN model v.s. number of neighbors and number of data. Noise variance is 1.

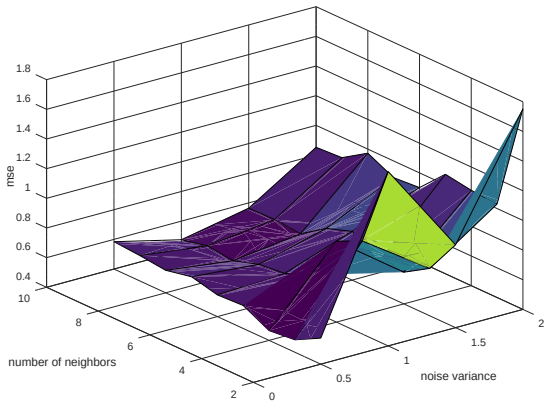


Figure 7: Model quality (mean square error) of KNN model v.s. number of neighbors and noise variance. Data size is 1000.

It can be seen in Figure 6 and Figure 7 that the model quality of KNN model depends on both the data size and the noise variance. The results also suggest that the larger the noise variance and the larger the data size, the more the number of neighbors we should choose. A heuristically optimal number K of nearest neighbors can be found based on the variance-bias trade-off.

3.2 Polynomial data

3.2.1 Linear regression model (constant + linear term)

Using a linear model it does not converge to the true model, which can be seen in Figure 8.

When we regress non-linear data on linear regressors, the model does not converge to the correct function while the parameter uncertainty goes to zero. This is due to the fact that when the number

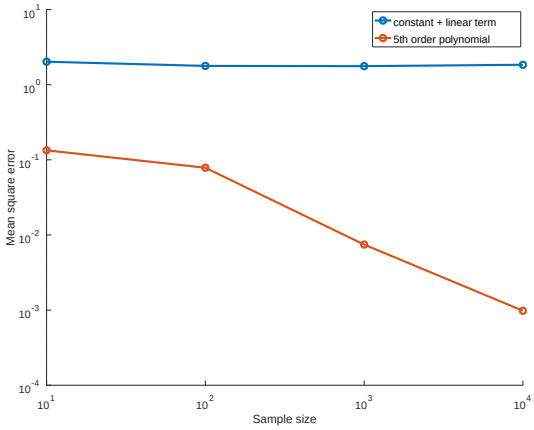


Figure 8: Mean square error of linear and 5th degree polynomial.

of data goes to infinity the model uncertainty decreases, which can be seen in Figure 9. Recall the

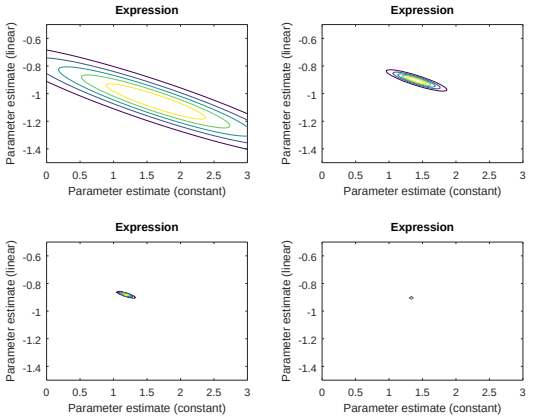


Figure 9: Model uncertainty for 10 samples (upper left), 100 samples (upper right), 1000 samples (lower left), and 10000 samples (lower right).

expression for parameter estimation variance,

$$\text{var}(\hat{\theta}) = (X^T X)^{-1}\sigma^2, \tag{16}$$

where σ^2 is the noise variance. It can be found that the parameter uncertainty depends on $(X^T X)^{-1}$, which further depends on the dimension of X, i.e., the data size. The larger the data size, the smaller $(X^T X)^{-1}$ and the smaller the parameter uncertainty.

3.2.2 Linear regression model (polynomial + regularization)

In Figure 10 the regression results of multiple polynomials of varying degrees and linear regression on the polyData set is shown. It is regressed on the underlying data using 100 samples and from the results both the polynomials of degree 3 and 4 provide a good fit.

As can be seen from Figure 11, the best result for 100 samples is obtained when using a polynomial model with degree 4.

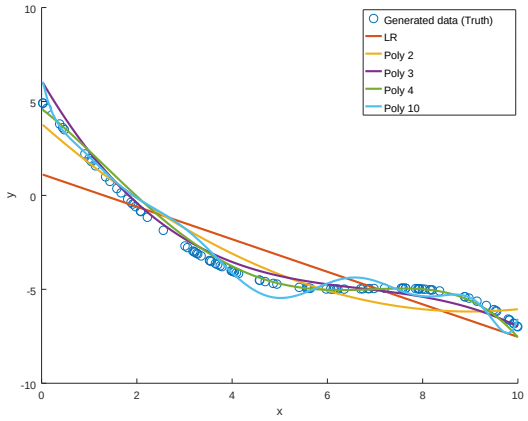


Figure 10: Multiple polynomials of various degree.

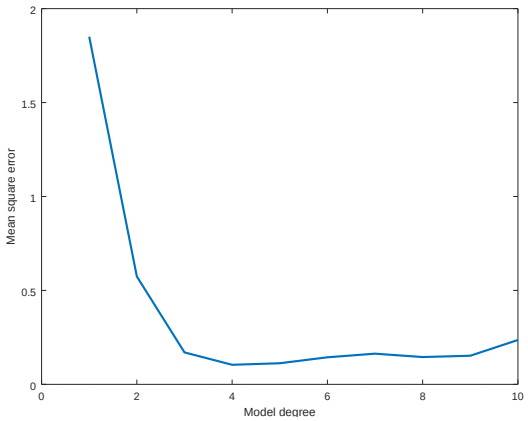


Figure 11: Model degree vs mean squared error.

When using a linear regressor with 10th order polynomial on a small data size, e.g., 15, the estimation does not fit the model due to overfitting. As shown in Figure 12, the estimation error can be reduced by adding regularization term; however, the size of the regularization term that gives the best model quality depends on the data. The regressors with varying regularization parameter are presented in Figure 13. Having a small number of data points N=15, in the range of the degree of polynomials n=10, the linear regression model is very likely to overfit the training data. By increasing the regularization parameter some of the variance is traded against a larger bias, which is favorable in this situation and seen in both Figure 12 and Figure 13. Increasing the regularization parameter forces the regressor to be less flexible, which in the case of having few training data is a great tool for acheiving an acceptable fit. There is an optimal regularization parameter value when the increase in bias error is larger than the decrease in variance error.

3.2.3 Regress unsymmetrical data using linear regression

The unsymmetrical data is generating data where 90 % lies within the interval [0,5] and 10 % within [5,10], in contrast to the symmetrical data previously used where the data is uniformly distributed over the whole range [0, 10]. This could be problematic, since using linear regression without regularization all data points are equally weighted. This will contribute to a better fit of

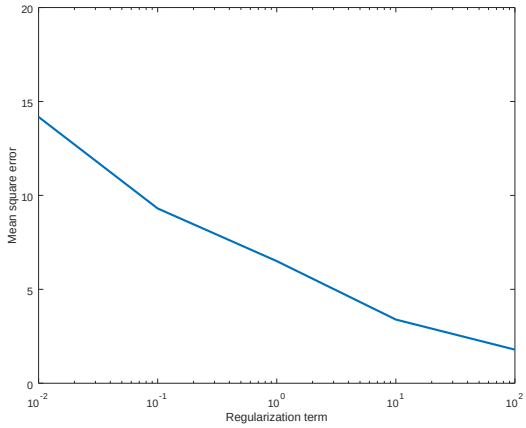


Figure 12: Model quality (mean square error) v.s. Regularization term.

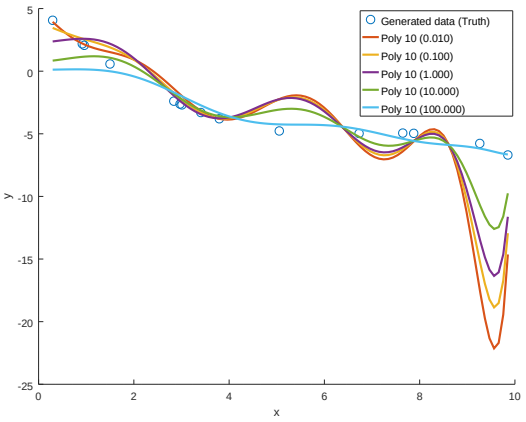


Figure 13: Polynomial regressors of degree 10 with varying regularization parameter.

the underlying function in [0,5], but a worse fit in [5,10]. To solve this problem we could e.g. add a regularization term or make sure that the training data is uniformly distributed withing the range of the regressors. As can be seen from Figure 14, the linear regression does not fit the data, especially the last part. In order to solve this problem, we can either increase the polynomial degree of the linear regression model we use or use a KNN model instead.

3.2.4 Regress unsymmetrical data using KNN

Figure. 15a shows how the model quality of KNN model varies with the number of neighbors and the number of data with fixed noise variance, and Figure. 15b shows how the model quality of KNN model varies with the number of neighbors and the number of data with fixed data size. A general rule is that the larger the noise variance and the larger the data size, the more the number of neighbors we should choose. A heuristically optimal number K of nearest neighbors can be found based on the variance-bias trade-off.

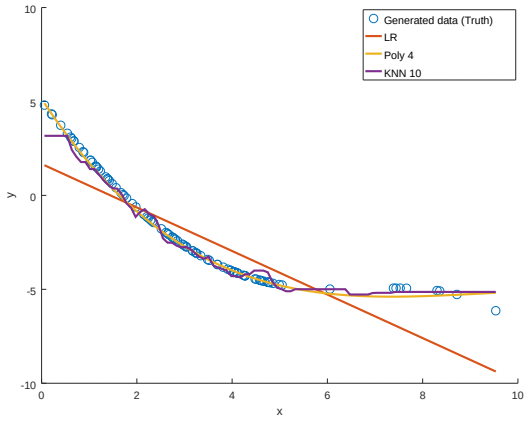
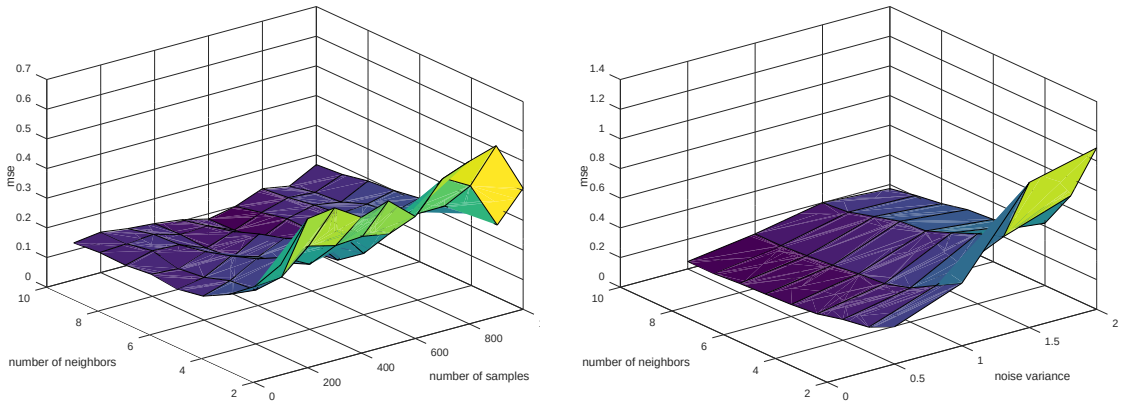


Figure 14: Regressing unsymmetrical data (100 training data and 100 validation data).



(a) Varying number of samples, constant noise of variance 1.

(b) Varying noise, constant number of samples of 1000.

Figure 15: Number of neighbors depending on noise and number of samples.

3.3 Chirp data

3.3.1 Influences of data size and noise level on polynomial model

Empirical results show that better higher order model can be obtained by increasing the data size and decreasing the noise level. The "best" polynomial degree according to the findings in Figure 16 is 16, while this is highly sensitive to the amount of data should the data size become closer to the degree of the polynomial. When the data size is small and the noise level is high, overfitting is likely to happen because the model is working too hard to find patterns in the training data which are just cause by random chance. Hence, increasing the data size and lowering the noise level help reduce the chance of overfitting, which further provides a better estimation.

3.3.2 High-degree polynomial with regularization v.s. Low-degree polynomial without regularization

Is a high degree polynomial with regularization preferred over a low degree polynomial without regularization? Generally speaking, if the degree of a polynomial model is too low to fit the data well, we then should choose a higher degree to have smaller estimation bias. The increased estimation variance can be reduced by regularization. By applying regularization, i.e., shrinking the estimated parameters, we can often substantially reduce the variance at the cost of a negligible increase in bias.

3.3.3 Linear regression v.s. KNN

We have tested a linear regression model with polynomial degree two to twenty and a KNN model with number of neighbors two to twenty on generated data with size 50 and 1000 respectively. The results are shown in Figure 16, where model degree is plotted against mean square error on validation data. When the data size is small, the best performance of KNN is achieved by choosing the number

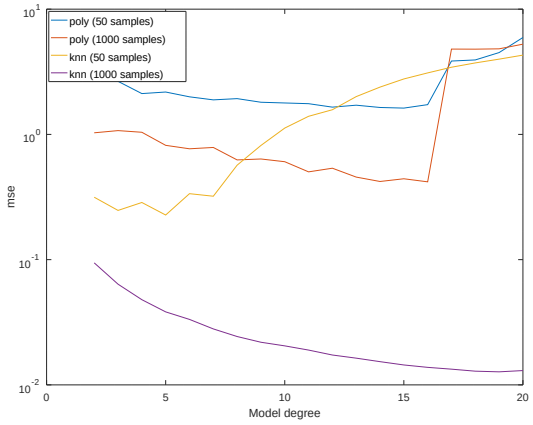


Figure 16: Model degree VS mean square error. For a KNN regressor the model degree corresponds to the number of neighbors while the degree of polynomials considered in the case of a polynomial regressor. Using a regularization factor of 10.

of neighbors around $K = 5$. KNN performs slightly better than linear regression on a small data set. The specific comparison result depends on the polynomial degree of the linear regression, the randomly generated data and the noise level. However, when the data size is large, KNN outperforms linear regression with various degrees. In this case, the mean square error of the estimation result obtained using KNN is orders of magnitude less than the one using linear regression.

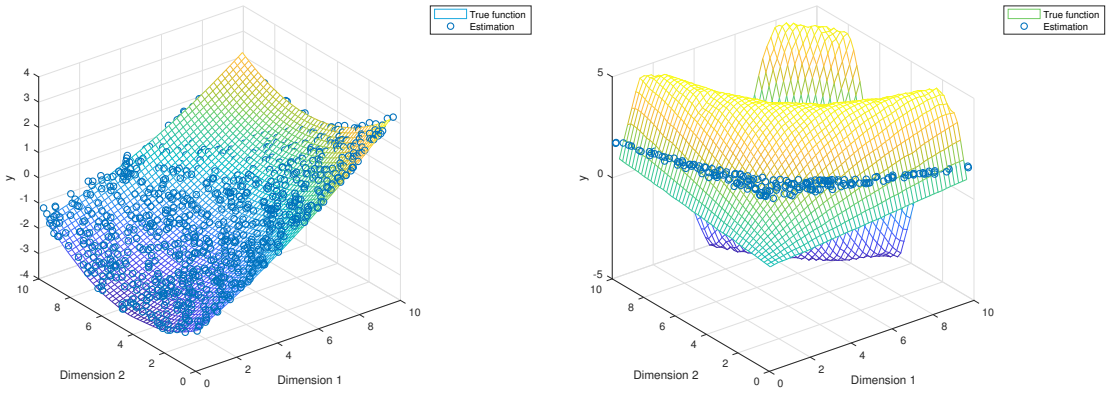
Analysis: linear regression is an example of a parametric approach because it assumes a linear functional form for $f(\mathbf{x})$. Because the chirp data is highly non-linear, the resulting model after linear regression will provide a poor fit to the data. The non-linearity of the chirp data can be verified by finding the Taylor series of $\sin(\mathbf{x}^2)$, which has polynomial order towards infinity. As for the KNN model, a non-parametric regression approach, no explicit form for $f(\mathbf{x})$ is assumed, thus providing a more flexible estimation.

4 Estimating two and high dimensional functions

4.1 Two dimensional data

The model quality can still be measured by calculating the mean square error.

4.1.1 Linear regression model

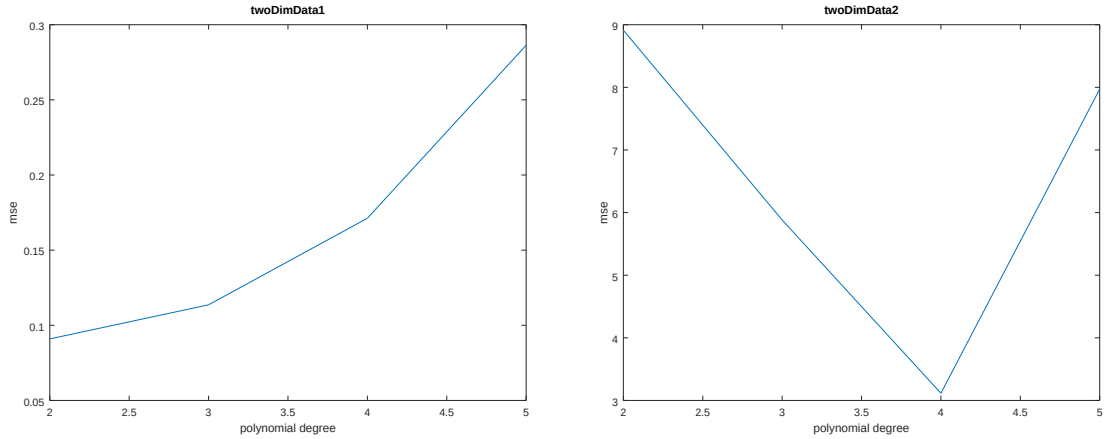


(a) twoDimData1 (b) twoDimData2

Figure 17: Linear regression on two data sets.

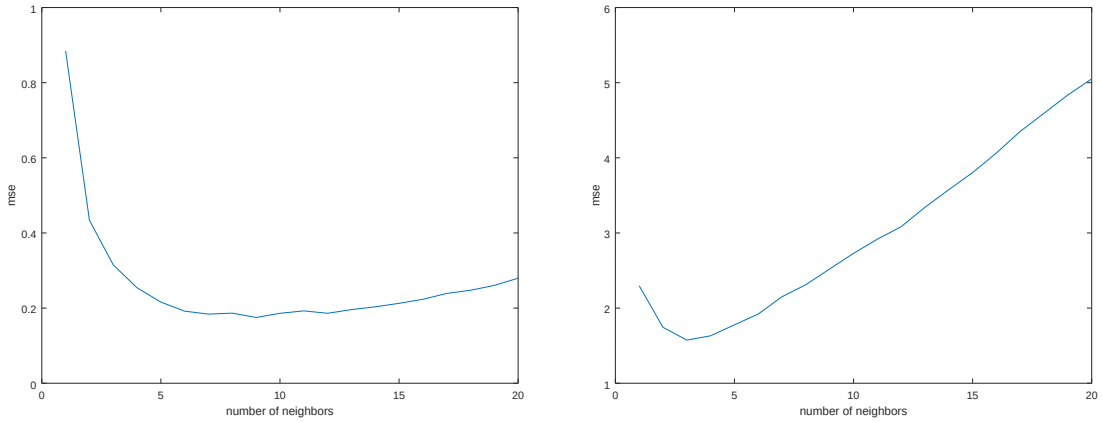
4.1.2 Polynomial models

In Figure 18 the mean square error for varying degrees of polynomials is presented. Best regression results are obtained using polynomials of degree 2 for twoDimData1 and polynomials of degree 4 for twoDimData2.



(a) twoDimData1 (b) twoDimData2

Figure 18: Mean square error VS degree of polynomials in polynomial regression for two data sets.



(a) twoDimData1 (b) twoDimData2

Figure 19: Mean square error VS number of neighbors for a KNN regressor for two different data sets.

4.1.3 KNN model

twoDimData1: Given 100 generated data, the KNN model gives the best estimation result when choosing the number of neighbors around $K = 10$ (see Figure 19a), and the polynomial model outperforms the KNN. Increasing the number of data will not change the comparison result. However, when the data size is very small, e.g., 10, both the polynomial model and the KNN cannot provide a good estimation. In this case, which model is better depends on the stochastically generated data.

twoDimData2: Given 100 generated data, the KNN model gives the best estimation result when choosing the number of neighbors around $K = 3$ (see Figure 19b), and the KNN outperforms the polynomial model. Empirical results show that increasing or decreasing the number of data may influence the optimal number of neighbors but will not change the comparison result.

4.2 Ten dimensional data

4.2.1 Number of regressors

Given a ten dimensional data, a linear regression model has 11 regressors, one for constant, and ten for linear terms. For a linear regression of polynomials up to degree 3 this is a bit more complicated. The formula $\binom{r+n-1}{r}$, which gives the number of combinations to choose r items among n items **with replacement** will be used to verify that the number of regressors is 286. There is 1 constant regressor, $\binom{1+10-1}{1} = 10$ linear regressors, $\binom{2+10-1}{2} = 55$ quadratic regressors, and $\binom{3+10-1}{3} = 220$ qubic regressors, giving a total of 1+10+55+220=286. This match the number of regressors in our implemented code.

4.2.2 Testing result of linear regression model

A linear regression model outperforms a polynomial model with degree 3 without regularization. Adding regularization to the polynomial model can improve the estimation performance substantially, see Figure 20.

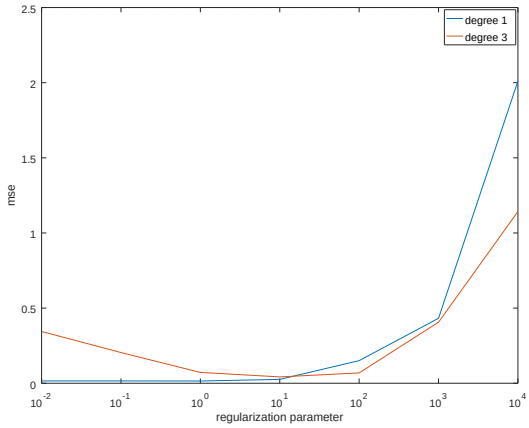


Figure 20: Regularization on linear regression models of polynomials of degree 1 and 3 given in the legend.

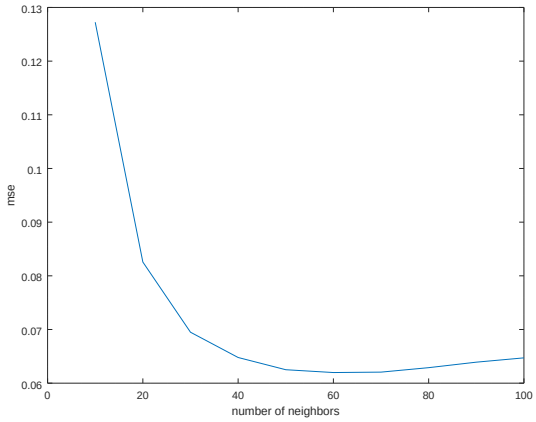


Figure 21: Number of neighbours for KNN regressor on 10 dimensional data.

4.2.3 Testing result of KNN

In Figure 21 it can be seen that given 1000 data choosing the number of neighbors around $K = 50$ gives the best result, and the linear regression with polynomial degree 3 and regularization parameter $\lambda = 10$ outperforms the KNN. The best model however is the linear model using only polynomials to the degree 1 and regularization parameter 1. Increasing the data size will not change the comparison result. However, we found that, when the data size decreases to 500, KNN and linear regression start to have similar performance, and that, if we further decrease the data size, KNN will instead outperform linear regression.

Analysis: As a general rule, linear regression will tend to outperform KNN when there is a small number of observations per predictor. This rule especially holds for high-dimensional data since the KNN suffers from the curse of dimensionality. However, our observation obeys this general rule. In the regression, we use all the possible regressors up to degree 3 to fit the model. However, it is more often the case that the response is only related to a subset of the regressors. In order to fit a single model involving regressors that are associated with the response, variable selection should have been done. Using redundant regressors will increase the chance to fit unexpected pattern in the data that are randomly generated by noise; thus estimation performance might be deteriorated. This explains

our observation that the KNN outperforms the linear regression model without variable selection even when the data size is small.