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

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

March 28, 2018

1 Linear regression functions

1.1 Estimation parameter validation

1.2 Regularization verification

Given a regressor matrix $\mathbf{x}$ and an output matrix $\mathbf{y}$, the linear least squares estimate with L2-Regularization can be obtained by minimizing the sum of squared residuals,

$$\hat{\theta} = \arg\min_{\theta} (\mathbf{y} - \mathbf{x}\theta)^T(\mathbf{y} - \mathbf{x}\theta) + \lambda\theta^T\theta. \tag{1}$$

The estimate $\hat{\theta}$ can be found by setting the derivative of (1) w.r.t. θ to zero,

$$\frac{d}{d\theta}\left((\mathbf{y} - \mathbf{x}\theta)^T(\mathbf{y} - \mathbf{x}\theta) + \lambda\theta^T\theta\right) = 0, \tag{2a}$$

$$-\mathbf{x}^T\mathbf{y} + \mathbf{x}^T\mathbf{x}\theta + \lambda\theta = 0, \tag{2b}$$

$$(\mathbf{x}^T\mathbf{x} + \lambda\mathbf{I})^{-1}\mathbf{x}^T\mathbf{y} = \hat{\theta}. \tag{2c}$$

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

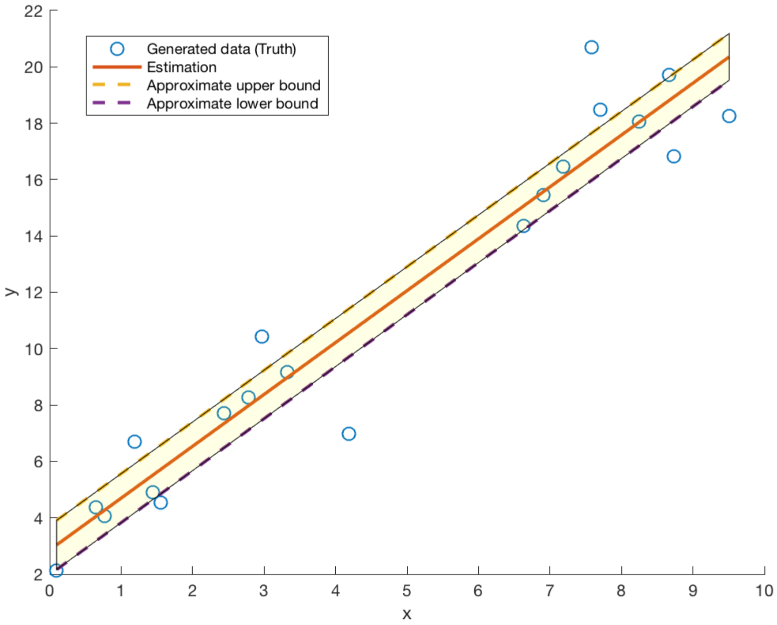
$$\hat{\theta} = (\mathbf{x}^T\mathbf{x})^{-1}\mathbf{x}^T\mathbf{y}, \tag{3}$$

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

1.3 Polynomial fitting validation

1.4 One dimensional model plotting

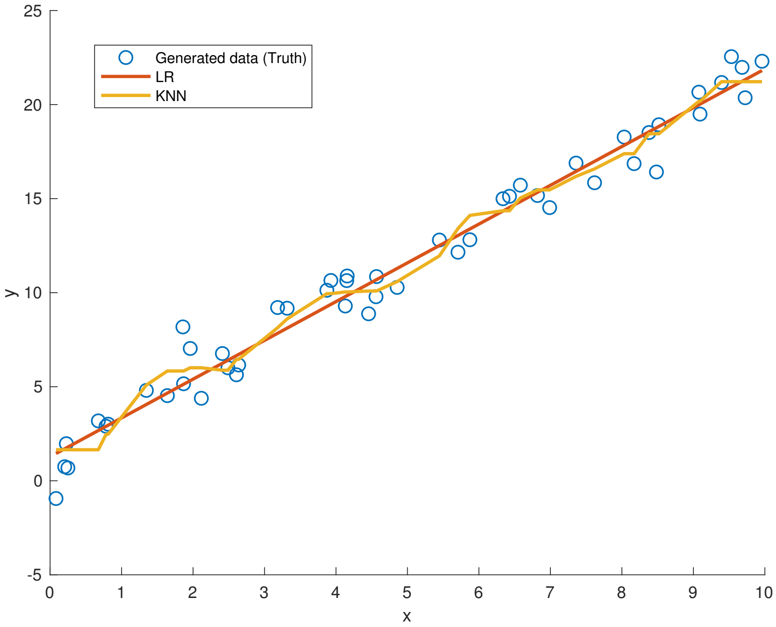
Figure 1: Estimated function using linear regression (LR). Uncertainty is illustrated as shaded area with confidence level 0.95. Each sample of regressor vector $\mathbf{x}$ is randomly drawn from uniform distribution $[0, 10]$. The true function is $\mathbf{y} = 2 + 2\mathbf{x}$, the noise variance is set to 2, and the number of samples is 20.



2 KNN-regression functions

2.1 KNN v.s. Linear regression

Figure 2: 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)

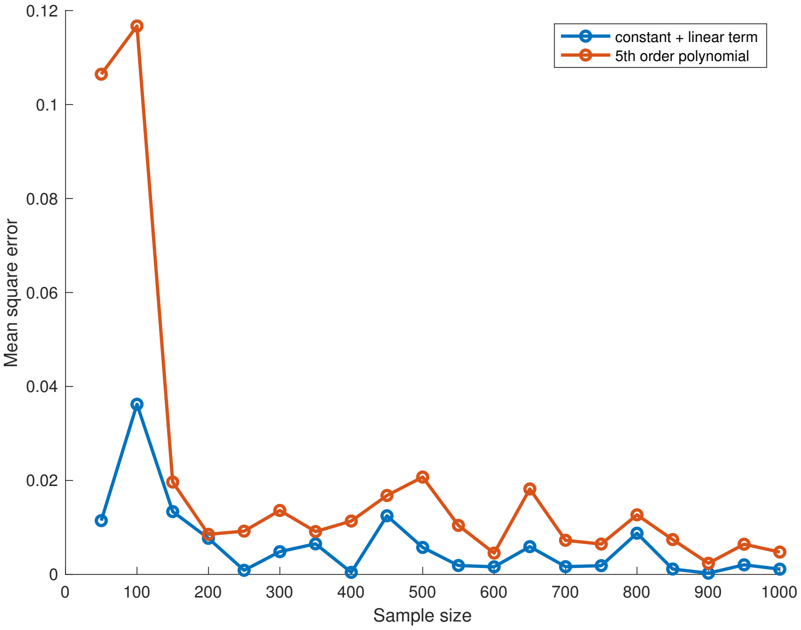
Table 1: Estimation results of a linear regression model with only constant and linear terms.

Generated data size (N)	Constant estimation	Linear term estimation
N=10	1.74	0.45
N=100	1.34	0.53
N=1000	1.63	0.48
N=10000	1.47	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 goes to infinity.

3.1.2 Linear regression model (5th order polynomial)

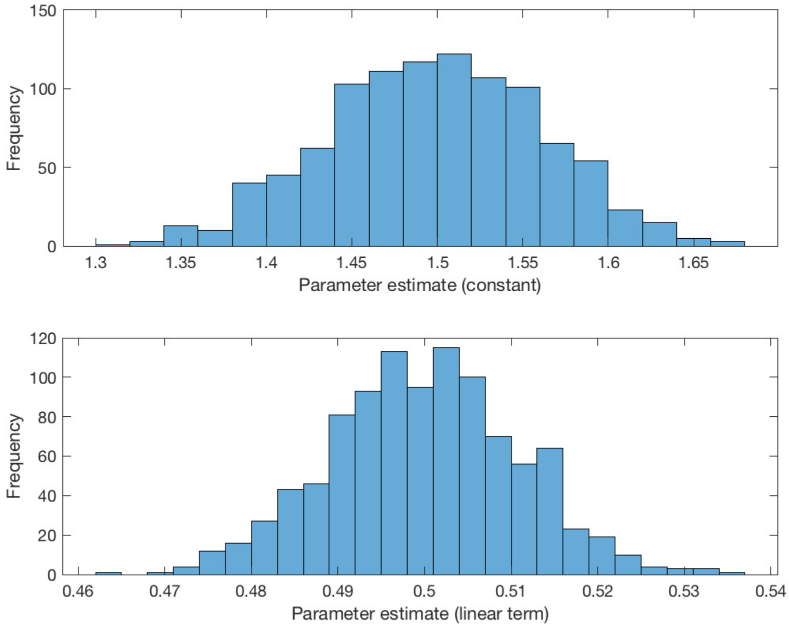
Figure 3: Model quality (mean square error) v.s. Number of data



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 4: Histogram of parameter estimates using 1000 generated data over 1000 Monte Carlo trials.



3.1.4 KNN models

Figure 5: Model quality (mean square error) of KNN model v.s. number of neighbors and number of data. Noise variance is 1.

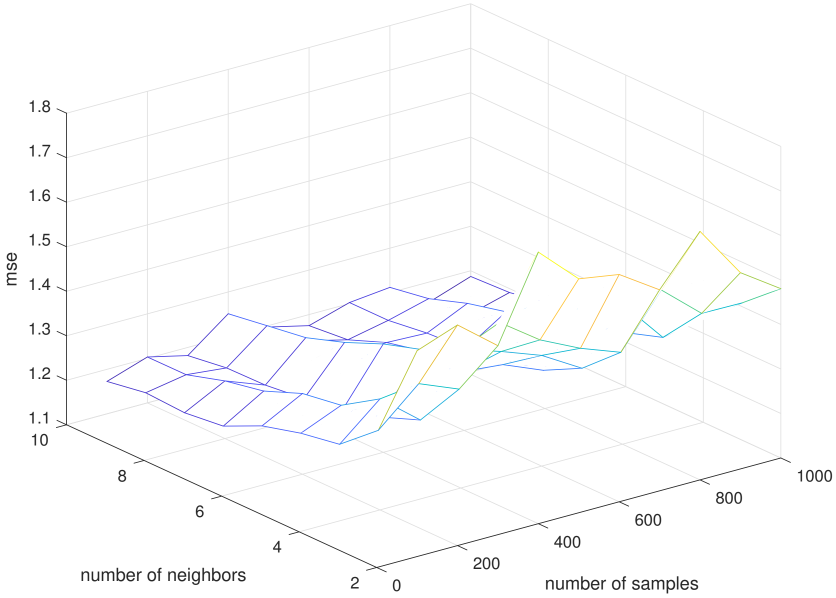
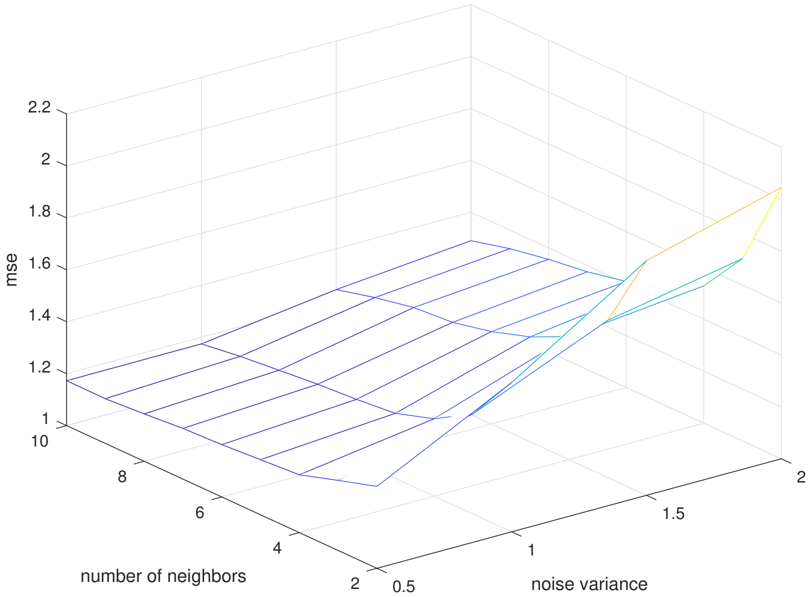


Figure 6: Model quality (mean square error) of KNN model v.s. number of neighbors and noise variance. Data size is 1000.



It can be seen from the results 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)

When we regress non-linear data on linear regressors, the model does not converge to the correct function and the parameter uncertainty goes to zero when the number of data goes to infinity. Recall the expression for parameter estimation variance,

$$\mathrm{var}(\hat{\theta}) = (\mathbf{x}^T\mathbf{x})^{-1}\sigma^2, \tag{4}$$

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

3.2.2 Linear regression model (polynomial + regularization)

Figure 7: Model quality (mean square error) v.s. Polynomial degree.

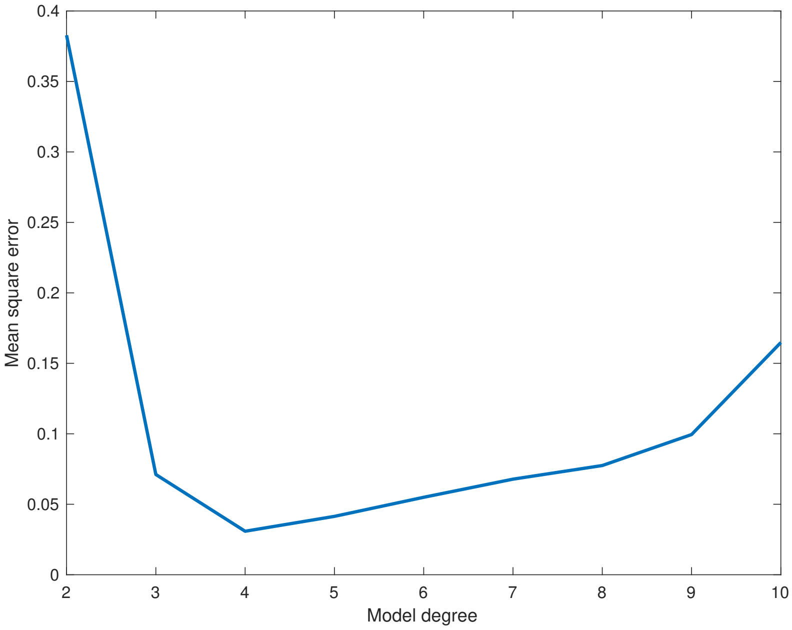
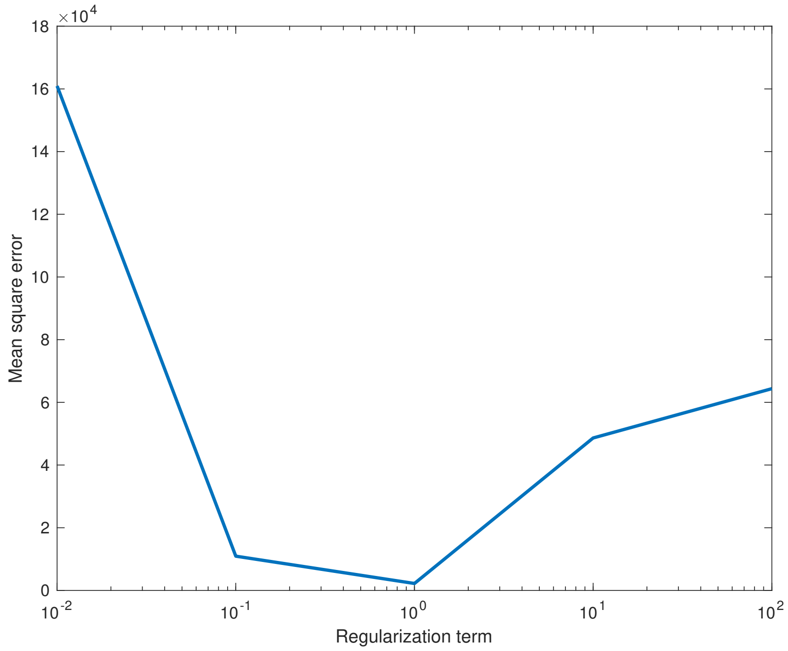


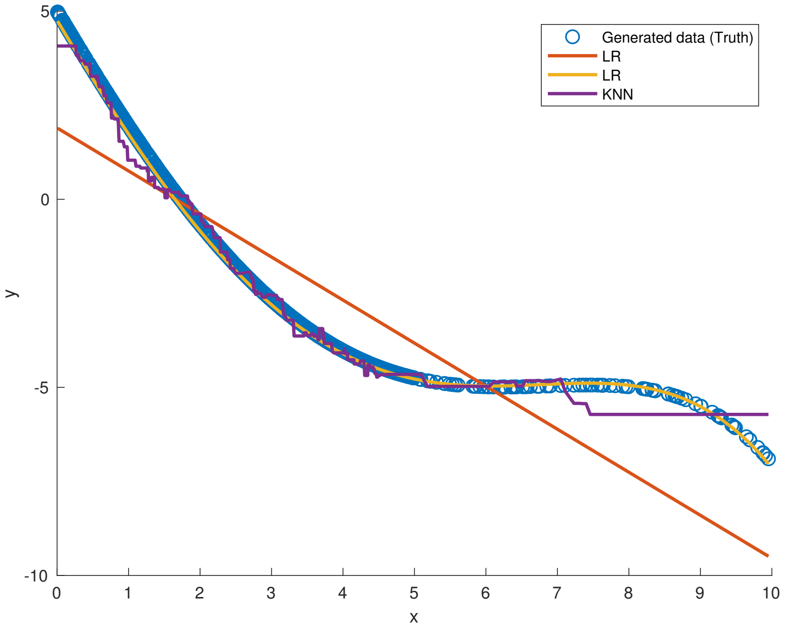
Figure 8: Model quality (mean square error) v.s. Regularization term.



As can be seen from Figure. 7, the best result (using 1000 generated data) is obtained when using a polynomial model with degree 4. 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. 8, 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.

3.2.3 Regress unsymmetrical data using linear regression

Figure 9: Regressing unsymmetrical data (1000 training data and 10000 validation data).



As can be seen from Figure. 9, the estimation 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 10: Regressing unsymmetrical data on a KNN. Noise variance is 1.

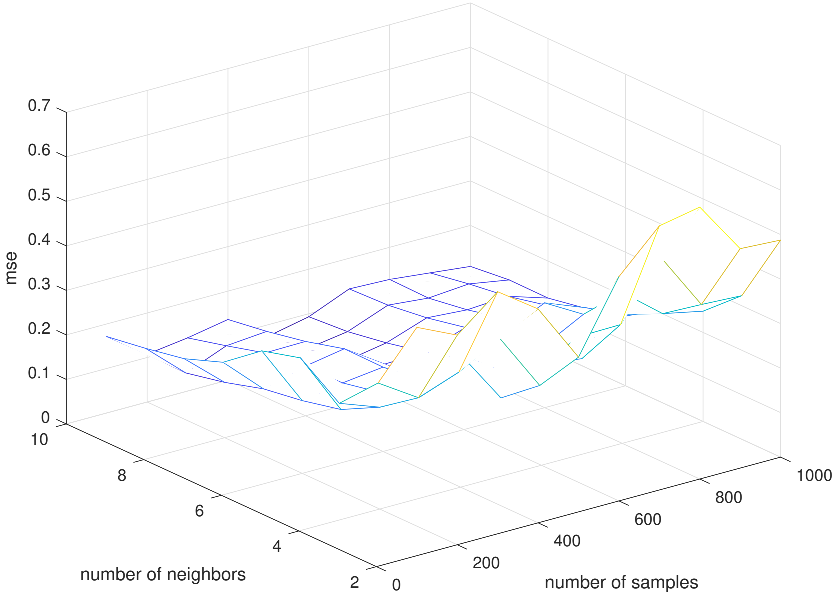


Figure 11: Regressing unsymmetrical data on a KNN. Data size is 1000.

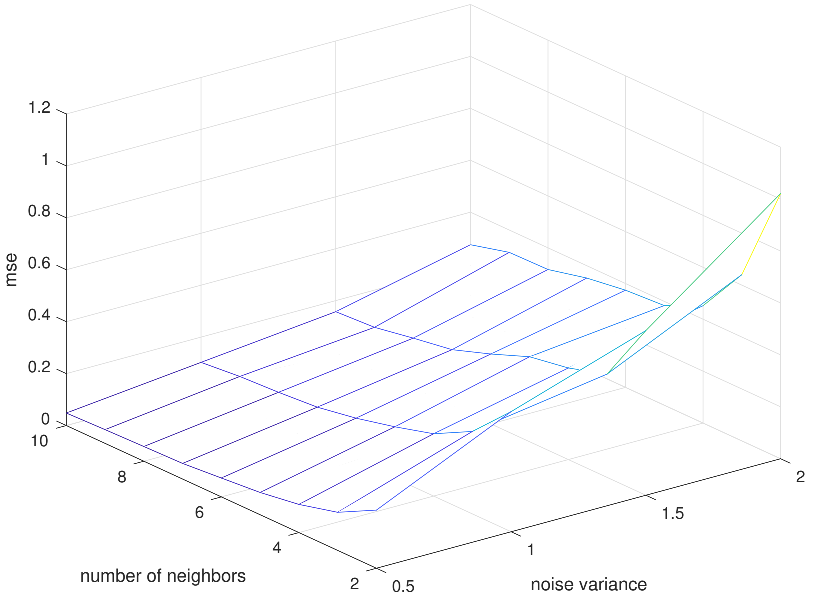


Figure. 10 shows how the model quality of KNN model varies with the number of neighbors and the number of data with fixed noise variance, and Figure. 11 shows how the model quality of KNN model varies with the number of neighbors and the number of data with fixed data size. A general rule is that the 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.

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 depends on the data since it is scholastically generated. 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

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 ten and a KNN model with number of neighbors two to ten on generated data with size 50 and 1000 respectively. When the data size is small, the best performance of KNN is achieved by choosing the number of neighbors $K = 2$. In this case, it is hard to say that KNN is better than linear regression. 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

Figure 12: Regressing twoDimData1 on a linear regression model

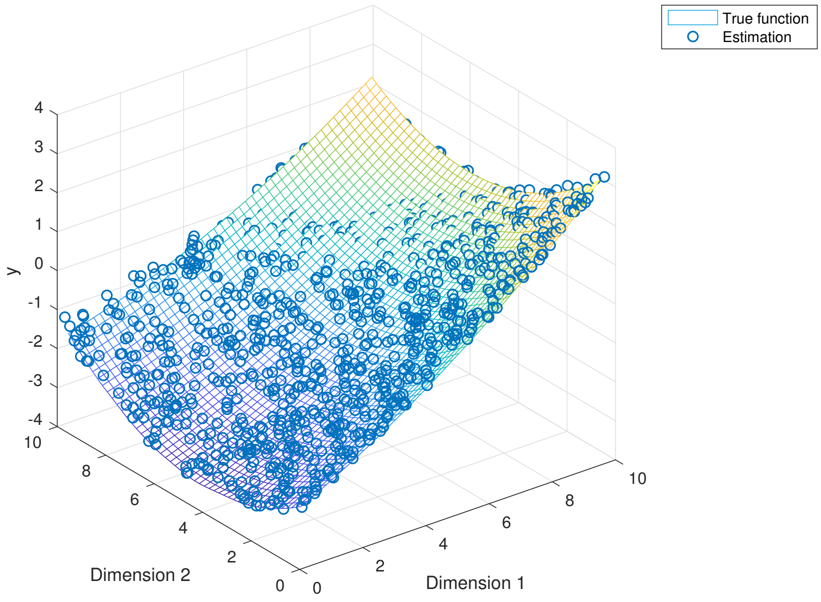
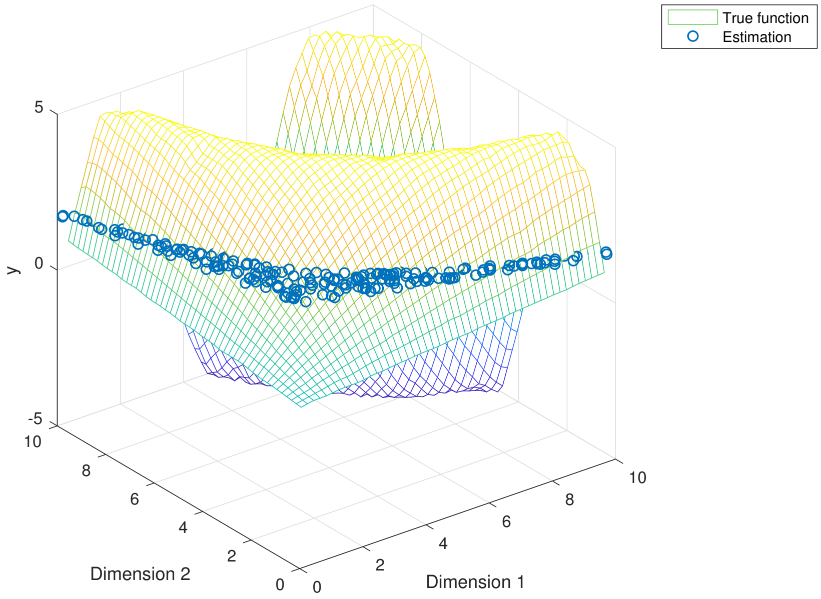


Figure 13: Regressing twoDimData2 on a linear regression model



4.1.2 Polynomial models

Table 2: Mean square error

Polynomial degree	2	3	4	5
twoDimData1	0.1	0.2	0.3	0.4
twoDimData2	8.5	5.0	4.2	2.9

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 $K = 10$, 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 $K = 3$, 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 model containing polynomial regressors up to degree 3, it has 1 regressor for constant term, $\binom{10}{1}$ regressors for linear term, $\binom{10}{1} + \binom{10}{2}$ regressors for quadratic term and $\binom{10}{1} + 2\binom{10}{2} + \binom{10}{3}$ for cubic term (in total 286 regressors).

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.

4.2.3 Testing result of KNN

Empirical results show that, given 1000 data, choosing number of neighbors around $K = 30$ gives the best result, and the linear regression with polynomial degree 3 and regularization parameter $\lambda = 10$ outperforms the KNN. 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.