5 Model Selection

Exercise 5.1

Consider the following snippet of code taking as input an `N` $\times$ `M` matrix `X` of data points and an `N`-dimensional vector `y` of binary targets, where `N` is the number samples, `M` is the number of features.

```
1  import numpy as np
2  classifier = LogisiticRegression()
3  classifier.fit(X, y)
4  yhat = classifier.predict(X)
5  accuracies = [sum(yhat = = y) / N]
6  for i in range(M):
7     Xi = np.delete(X, i, axis=1)
8     classifier.fit(Xi, y)
9     yhat = classifier.predict(Xi)
10    accuracies.append(sum(yhat = = y) / N)
```

Are the subsequent statements true or false? Provide motivations for your answers.

1. This snippet of code implements a portion of a well-known model selection procedure.

2. After having run the reported snippet, one should keep the classifier that led to the maximum value in the list of accuracies.

3. The classifier trained at line 3 is likely to suffer a larger bias than the classifiers trained at line 8.

4. The time of computation we need to run the snippet of code scales linearly with `M`.

Exercise 5.2

Answer to the following questions regarding feature selection. Provide motivation for your answers.

1. You have been asked to implement a feature selection process on a system with very limited computational resources. Would you opt for a filter approach or for a wrapper approach?

2. You have been asked to implement a feature selection process to improve as much as possible the performance of your model. Would you opt for a filter approach or for a wrapper approach?

3. If you want to rank (in order of importance) the features of a classification problem, which kind of feature selection process would you use among the ones presented in the course?

4. You trained two models on a problem with 5 features: Model A using all the 5 features and Model B using only 3 features. Assuming they have similar performances on the training set, do you expect Model A to perform better on the test set?

5. You trained two models on a problem with 9 features: Model A using only 5 features and Model B using all the 9 features. Do you expect Model A to have a smaller training error than Model B?

Exercise 5.3

We selected the features of a linear regression model by running a forward feature selection procedure which minimizes the validation error. For each of the following, tell if the elements corresponding to the learned model increase or decreases as the mode is increasing the number of feature (given that we trained on the same data):

1. The sum of the squared residuals on the same training set;

2. The variance;

3. The squared bias;

4. The sum of the squared residuals on a test set.

Exercise 5.4

Assume you are solving a regression problem with a linear regression model and you are considering to switch either to LASSO or Ridge regression. Tell which one you would choose basing on the following requirements (one at a time):

1. Too few data w.r.t. the number of parameters we are using for regression;

2. The final model is hard to interpret due to the large number of input features that have been used for the regression;

3. Bad conditioning of the design matrix $\Phi(x)^\top \Phi(x)$;

4. Problem with model bias of the current model.

Motivate your choices.

Exercise 5.5

Consider the following snippet of code taking as input an `N` × `M` matrix `X` of data points and an `N`-dimensional vector `y` of binary targets, where `N` is the number samples, `M` is the number of features. Are the subsequent statements true or false? Provide motivations for your answers.

```
1  import numpy as np
2  classifier = LogisiticRegression()
3  classifier.fit(X, y)
4  yhat = classifier.predict(X)
5  accuracies = [sum(yhat = = y) / N]
6  for i in range(M):
7  Xi = np.delete(X, i, axis=1)
8  classifier.fit(Xi, y)
9  yhat = classifier.predict(Xi)
10 accuracies.append(sum(yhat = = y) / N)
```

1. This snippet of code implements a portion of a well-known model selection procedure. (*Note*: in any case, specify the procedure and briefly describe how to complement the code).

2. After having run the reported snippet, one should keep the classifier that led to the maximum value in the list of accuracies.

3. The classifier trained at line 3 is likely to suffer a larger bias than the classifiers trained at line 8.

4. The time of computation we need to run the snippet of code scales linearly with `M`.

Exercise 5.6

Consider the following snippet of code:

```
coef = []
mse = []
for alpha in [0.001, 0.01, 0.1, 0.2, 0.5]:
model = linear_model.Lasso(alpha=alpha)
model.fit(x, y)
mse.append(mean_squared_error(y, model.predict(x)))
coef.append(model.coef_)
i = np.argmin(mse)
print(i, mse[i], coef[i])
```

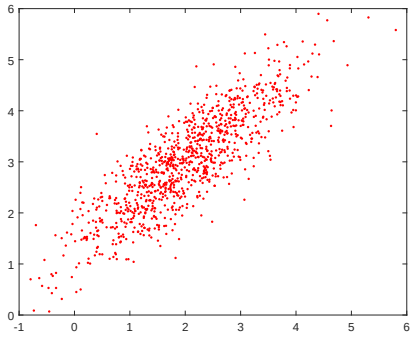
1. Describe the procedure provided by the snippet above. Is it correct? If not, propose a correction.

2. What is the purpose of the procedure above? Can you tell which property have the coefficients `coef` computed in the snippet above?

3. Can you list at least two other approaches that can be used for purposes similar to the one provided by the procedure above? Motivate your answers.

Exercise 5.7

Consider the following dataset:



Draw the direction of the principal components and provide an approximate and consistent guess of the values of the loadings. Are they unique?

Exercise 5.8

Consider the following statement regarding PCA and tell if they are true or false. Provide motivation for your answers.

1. Even if all the input features are on very similar scales, we should still perform mean normalization (so that each feature has zero mean) before running PCA.

2. Given only scores t_i and the loadings W, there is no way to reconstruct any reasonable approximation to x_i.

3. Given input data $x_i \in \mathbb{R}^d$, it makes sense to run PCA only with values of k that satisfy $k \leq d$.

4. PCA is susceptible to local optima, thus trying multiple random initializations may help.

Exercise 5.9

Which of the following is a reasonable way to select the number of principal components k in a datset with N samples?

1. Choose k to be 99% of N, i.e., $k = \lceil 0.99N \rceil$;
2. Choose the value of k that minimizes the approximation error $\sum_{i=1}^{N} ||x_i - \hat{x}_i||_2^2$;
3. Choose k to be the smallest value so that at least 99% of the variance is retained;
4. Choose k to be the smallest value so that at least 1% of the variance is retained;
5. Identify the elbow of the cumulated variance function.

What changes if the purpose of PCA is visualization?

Exercise 5.10

Consider the following snippet of code taking as input a dataset `X` having 100 samples with 5 features. Are the subsequent statements true or false? Motivate your answers.

```
1 import numpy as np
2 X_tilde = X - np.mean(X, axis=0)
3 C = np.dot(X_tilde.T, X_tilde)
4 eigenvalues, W = np.linalg.eig(C)
5 T = np.dot(X_tilde, W[:,:2])
```

1. The code snippet above is implementing a feature selection technique.
2. Line 2 is unnecessary if the data in `X` are scaled.
3. A model trained with the inputs `T` is likely to display a lower bias than a model trained with inputs `X`.
4. It is possible to recover, by computing $X = W[:, :2]^\top \cdot T$, the original dataset `X` from `T`.

Exercise 5.11

Consider the following statement regarding **Principal Component Analysis** (PCA) and tell if they are true or false. Provide motivation for your answers.

1. PCA might get stuck into local optima, thus trying multiple random initializations might help;
2. Even if all the input features have similar scales, we should still perform mean normalization (so that each feature has zero mean) before running PCA;

3. Given the scores $\tilde{x}_i, \forall i$ and the loadings matrix W, there is no way to reconstruct the original samples $x_i, \forall i$;

4. PCA can used either for data compression, data visualization or feature extraction.

Exercise 5.12

Tell if the following statement about the Principal Component Analysis (PCA) procedure are true or false. Motivate your answers.

1. The set of the Principal Components vectors are providing an orthonormal base for the original feature space.

2. Using as features for regression/classification problems the projection of the original features into the principal components provided by the PCA reduces the phenomenon of overfitting.

3. The percentage of the variance explained by a Principal Component is inversely proportional to the value of the corresponding eigenvalue.

4. The procedure to apply PCA to a dataset is deterministic.

Exercise 5.13

Assume you have been given a dataset with input matrix X and target vector y. Consider the following snippet of code:

```
1  pca = PCA()
2  pca.fit(X, y)
3  explained = pca.explained_variance_
4  T = pca.transform(X)
5  explained_variance = np.cumsum(explained) / sum(explained)
6  T_tilde = T[:, explained_variance < 0.95]
```

1. Describe the procedure and the purpose of the above code snippet. Is it correct?

2. Line 6 consists in a selection procedure. Explain the rationale behind this operation and suggest other viable options to perform the selection procedure.

3. Do you think this code requires some preliminary operations on X and y before being executed? In the case of a positive answer, tell which ones and why should they be performed. In the case of a negative answer, motivate adequately.

Exercise 5.14

Imagine having a dataset with a small amount of data you suspect is corrupted. Moreover, you have a specific application that allows you to consider only simple models in the system's training and operational life (e.g., embedded microcontroller).

1. Propose a solution for the above-described setting.
2. Instead, assume to have a long time for training. What techniques would you consider to reduce the influence of these corrupted samples on the used model?
3. Does the prediction phase of the learner increase with the proposed technique? By how much?

Exercise 5.15

Describe the **advantages** and **drawbacks** of the following choices about model selection in Machine Learning:

1. Increase the model complexity and fix the number of samples;
2. Fix the model complexity and the number of samples, but use ensemble techniques.

Exercise 5.16

State whether the following claims about Bagging and Boosting are true or false, motivating your answers:

1. Since Boosting and Bagging are ensemble methods, they can be both parallelized.
2. Bagging should be applied with weak learners.
3. The central idea of Boosting consists in using bootstrapping.
4. It is not a good idea to use Boosting with a deep neural network as a base learner.

Exercise 5.17

Tell which technique or approach would you use for the following purposes. Motivate your choice.

1. Reduce the variance of a model;
2. Select a model without retraining it on a different set of data;
3. Reduce the bias of the model, without increasing its variance;
4. Select a model by exploiting the huge computational power available to you.

Exercise 5.18

Answer to the following questions about the bias-variance decomposition, model selection, and related topics. Motivate your answers.

1. If your linear regression model underfits the training data (i.e., the model is not complex enough to explain the data), would you apply PCA to compute a more suitable feature space for your model?

2. If solving a regression problem, the design matrix $X^\top X$, is singular, would you apply PCA to solve this issue?

3. Assuming a classifier fits very well the training data but underperforms on the validation set, would you apply Bagging or Boosting to improve it?

4. Assuming that you trained a classifier with a K-fold cross-validation and it consistently has poor performances both on training and on validation folds, would you apply Bagging or Boosting to improve it?

5. You applied ridge regression to train a linear model using a rather large regularization coefficient, would you think that bagging would improve your model?

6. You have been asked to implement a feature selection process on a system with very limited computational resources. Would you opt for a filter approach or for a wrapper approach?

7. You have been asked to implement a feature selection process to improve as much as possible the performance of your model. Would you opt for a filter approach or for a wrapper approach?

8. You need to train a linear regression model using as input the readings of several sensors. Assuming that you know that some of these sensors might be faulty (i.e., resulting in meaningless readings), which linear regression approach would you use to train your model?

9. A linear regression model, computed using ordinary least squares, has a validation error that is much larger than training error. Assuming that you do not want to change neither the input features nor the kind of model, what would you do to improve it?

10. If you have to choose among a few models knowing only the training error (assuming you cannot retrain them or evaluate them on a different dataset), what would you do?

Exercise 5.19

Answer to the following questions about the bias-variance decomposition, model selection, and related topics. Motivate your answers.

1. You trained two models on the same dataset on a problem with 5 features: Model A using all the 5 features and Model B using only 3 features. Assuming they have similar performances on the training set, do you expect Model A to perform better on the test set?

2. You trained two models on the same dataset on a problem with 9 features: Model A using only 5 features and Model B using all the 9 features. Do you expect Model A to have a smaller training error than Model B?

3. You trained a K-NN classifier and the performance on the validation set is much worse than the one on the training set. Would you increase the value of K?

4. You trained a model with ridge regression and the performance on the validation set is much worse than the one on the training set. Would you decrease the regularization coefficient?

5. You used 10-fold cross-validation to tune the hyper-parameter K of a classifier. The model trained on the third fold with $K = 3$ achieved the best performance overall. Based on this, would you set $K = 3$?

6. You used 10-fold cross-validation to select a classification model among several ones. Once you selected the model, would it be a good idea to re-train it on the whole dataset (i.e., all the 10 folds together)?

7. You trained 10 regression models applying different basis functions to the problem inputs. Assuming you do not have enough time to perform additional training, would you select the model with the lowest training error?

8. You need to assess the performance of a model on a very large dataset. You discover that training your model on the whole dataset is not very (computationally) expensive. Would you use Leave-one-out (LOO) cross-validation?

Answers

Answer of exercise 5.1

1. TRUE the code implements the first iteration of a typical *backward feature selection* procedure, which is used for model selection. However, some aspects are missing in the snippet, such as the computation of the accuracy on a validation set or cross-validation.

2. FALSE model selection cannot be performed on the training accuracy, otherwise we would always select the model that keeps all the `M` features.

3. FALSE the classifier trained on the full set of features is likely to suffer a larger variance and a lower bias w.r.t. the classifiers trained on a subset of the features.

4. FALSE whereas we need to run the loop 6-10 exactly `M` times, the time of computation we need to run the fit function of the classifier grows with `M` as well, being the logistic regression a parametric method.

Answer of exercise 5.2

1. FILTER because a wrapper approach involves solving an optimization problem that requires training several models. In contrast, filter approaches only require to compute statistics on the features.

2. WRAPPER because filter approaches usually assume that the features are independent and might not find the best subset.

3. FILTER because it involves computing a metric for each feature (e.g., correlation or information gain) such that on the basis of this metric features can be ranked and selected from the most relevant to the least one.

4. NO Model A is more complex and hence has a larger variance and probability of overfitting training data. So it would probably result in a worse test error.

5. NO Model A is simpler and will probably have a larger bias resulting in a larger training error.

Answer of exercise 5.3

1. DECREASES: as the model becomes more and more complex the training data are more prone to overfitting. If we use the closed form solution for linear regression (up to numerical problems) we are sure that the RSS for the training set is decreasing.

2. INCREASES: the variance of a model is strictly related to its complexity. In this case as we add more and more features we are also increasing the hypothesis space dimension. Therefore, such models will have an increased variance (given that they are using the same amount of data).

3. DECREASES: more complex models are more likely to decrease their bias, and, as said before, increasing the number of the variables makes the model more complex.

4. DECREASE AND THEN INCREASE: the behaviour of the test set error should follow the same behaviour of the one of the validation one. Since we will stop the feature selection procedure as soon as the validation error is increasing, the test set should have a similar pattern.

Answer of exercise 5.4

1. LASSO OR Ridge: to reduce the overfitting phenomenon we might use regularization, no matter which approach we use.

2. LASSO: since it provides a feature selection procedure intrinsically.

3. Ridge: the addition of the diagonal element to the design matrix stabilize the model. (LASSO would change the optimization method, so we would not be sure this problem would be solved, but it would be irrelevant in this case).

4. NONE: regularizing the current solution would not decrease the problem of bias, instead it would worse the problem.

Answer of exercise 5.5

1. TRUE, the code implements the first iteration of a typical *backward feature selection* procedure, which is used for model selection. However, some aspects are missing in the snippet, such as the computation of the accuracy on a validation set or cross-validation.

2. FALSE, model selection cannot be performed on the training accuracy, otherwise we would always select the model that keeps all the `M` features.

3. FALSE, the classifier trained on the full set of features is likely to suffer a larger variance and a lower bias w.r.t. the classifiers trained on a subset of the features.

4. FALSE, whereas we need to run the loop 6-10 exactly `M` times, the time of computation we need to run the fit function of the classifier grows with `M` as well, being the logistic regression a parametric method.

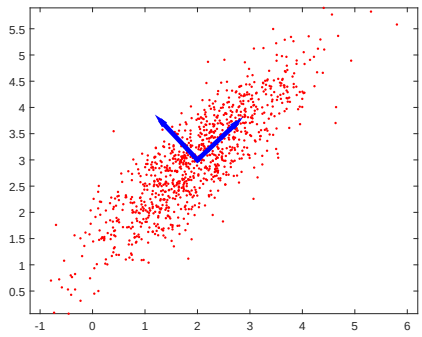
Answer of exercise 5.6

1. The above procedure is evaluating the regularization parameter for a LASSO model for the problem of regression. It fits models with different values of α and computes the training RSS to decide which one is the most suitable for the given dataset. Clearly this procedure would give as a result that the model with the smallest parameter $\alpha = 0.001$ is the one to chose, since we are validating our possible models on the training set. A correct procedure would have to validate on independent data (for instance, using validation or crossvalidation).

2. The purpose of the procedure is to introduce regularization into a regression model to avoid overfitting. The LASSO procedure produces a final parameter vector which is sparse. Indeed in Line 5 the resulting printed parameter would have less and less non-zero elements as the value of α (λ) increases.

3. To avoid overfitting one may apply other regularization techniques, like Ridge or model selection techniques, like the backward/forward feature selection approach. Differently from the LASSO we would have that the final solution is not necessarily sparse with Ridge, while it might be with the wrapper approaches.

Answer of exercise 5.7



The computed principal components loadings are:

$$\begin{matrix} 0.7287 & -0.6849 \\ 0.6849 & 0.7287 \end{matrix}$$

and a reasonable guess would be:

$$\begin{matrix} \frac{\sqrt{2}}{2} & -\frac{\sqrt{2}}{2} \\ \frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} \end{matrix}$$

The properties satisfied by the principal components are the unity norm and that they are orthogonal.

Answer of exercise 5.8

1. TRUE Since the principal components are identifying the directions where the most of the variance of the data is present, where the directions is defined as a vector with tail in the origin, we should remove the mean values for each component in order to identify correctly these directions.

2. FALSE By applying again the loadings matrix W to the scores t_i, thanks to the orthogonality property of W, we are able to reconstruct perfectly the original mean normalized vectors. If we want to reconstruct the original vectors we should also store the mean values for each dimension.

3. TRUE Running it with $k = d$ is possible but usually not helpful and $k > d$ does not make sense.

4. FALSE There is no source of randomization and no initialization point in the algorithm to perform PCA.

Answer of exercise 5.9

1. FALSE This way we could either include principal components which provides explanation for small amount of variance or exclude some of the most important ones.

2. FALSE This would mean to include all the principal components, which are able to perfectly reconstruct the original dataset;

3. TRUE The cumulated variance of the principal components provides us an estimates on how much of the variance of the original dataset is considered. Keeping an high percentage of it would imply that we are discarding components which does not vary too much or noise.

4. FALSE The same reason of the previous point.

5. TRUE If there is an elbow in the cumulated variance, the inclusion of the following principal components would not improve the representation of the data too much.

Answer of exercise 5.10

1. FALSE the code snippet is implementing the PCA, which is a feature extraction technique.

2. FALSE the data in `X` has to be centered anyway to compute the covariance matrix.

3. FALSE the dataset `T` has lower dimensionality than the dataset `X`. Thus, a model trained on `T` is likely to have a larger bias but a smaller variance.

4. FALSE in this way we are minimizing the reconstruction error, but it would be greater than zero in general.

Answer of exercise 5.11

1. FALSE: There is no source of stochasticity in the process of performing PCA. Thus the use of multiple initialization would not produce different results;

2. TRUE: The process considers the case in which the points are centered in the origin (have zero mean). If one of the component has an average value far different from zero, we might bias the direction of the first eigen–vector towards this dimension;

3. FALSE/TRUE: If we consider a zero mean dataset, we can exploit the orthogonality of the matrix W to reconstruct the original values ($x_i = \tilde{x}_i W$), otherwise we also need the values of the expected values of each dimension μ_i ($x_i = \tilde{x}_i W + \mu_i$);

4. TRUE: PCA naturally tries to minimize the euclidean error between the reconstructed and the original data, thus it can be used for data compression (considering the first k columns of the loadings matrix W). If we consider only the first 2–3 principal components, we are able to visualize the data, even if their original dimensionality did not allow to represent them. At last, PCA can be considered also as an unsupervised techniques for feature extraction, where we consider the first k principal components as new features.

Answer of exercise 5.12

1. TRUE: The procedure for the PCA looks for the direction in the dataset which is providing the most variance and extracts the (first) Principal Component (PC) as the unit vector identifying that direction. Iteratively, it checks the direction, orthogonal to the previous one, with the most variance and extracts another (second) PC. The process iterates over a number of PC equal to the number of dimensions of the dataset. This produces an orthogonal base to the initial dataset.

2. FALSE: only selecting the first K PC one has the chance to remove the noise from the data and avoid overfitting. If we are keeping all the PC, we would have a linear transformation of the original dataset, which is likely to behave as good as the original one for the supervised task.

3. FALSE: The variance explained by each PC is directly proportional to the corre-

sponding eigenvalue. Indeed, its formula is $\frac{\lambda_i}{\sum_i \lambda_i}$, where λ_i is the eigenvalues corresponding to the i-th PC.

4. TRUE: The procedure includes a mean-scaling procedure and the inversion of the covariance matrix, which are both deterministic procedures.

Answer of exercise 5.13

1. The code is implementing the PCA algorithm for feature extraction. It starts computing the loadings matrix, and applying them to the original data input X. Finally, it selects the principal components retaining at least 95% of the variance present in the data. The issue of this code is that the fit procedure is using also the target y, while this procedure is unsupervised and therefore it does not require the real target.

2. As mentioned above, the line selects the principal components retaining at least 95% of the variance present in the data. Other viable options are the detection of an "elbow" in the cumulated variance or retaining all the principal components explaining at least $k\%$ of the variance, with a fixed parameter $k \in (0, 1)$.

3. We need to remove the empirical mean from the input otherwise the principal components might be strongly affected by it.

Answer of exercise 5.14

At first, if you can state which of them are corrupted, you should remove them from the dataset, since they are not providing information to your learner. This would solve your problem.

1. Using a non-parametric method would completely remove the issue about the training of the model and, due to the small amount of data, even the prediction phase would not be too computationally expensive. For instance, K-NN would be a good idea. Regarding the corrupted data, we need to base our prediction on a large-enough number of samples. In the case of K-NN, it would mean that we need to set the parameter K to a large number so that the corrupted data do not influence the final prediction too much.

2. If we want to reduce the influence of such data in the learning process, we might resort to bootstrapping so that the amount of corrupted data, on average, is even smaller than in the original dataset. Moreover, the corrupted data would likely lead to a badly-performing method, therefore, they may be weighted less for the final decision.

3. The learning phase increases linearly with the number of datasets we are boot-

strapping and the computation required to learn the aggregating function (if this is required).

Answer of exercise 5.15

1. Increasing the model complexity should reduce the bias of the model (advantage), at the cost of increasing the variance (drawback).
2. Using ensemble technique might help in reducing the bias or/and the variance of the model (advantage) at the cost of increasing the computational time required to train the model (drawback).

Answer of exercise 5.16

1. False, only Bagging can be parallelized, since training is done on different datasets, while Boosting is sequential by nature.
2. False, weak learners are good candidate for Boosting, since they have low variance. Typically, one uses instead bagging when more complex and unstable learners are needed, to reduce their variance.
3. False, bootstrapping is used in bagging, whose name derived indeed from "boosting aggregation".
4. True, it is not a good idea to do that, since deep neural networks are very complex predictor, which can have large variance. Therefore, you may not succeed in lowering bias without increasing variance. Moreover, since you need to train the network multiple times, the procedure may require a lot of time.

Answer of exercise 5.17

1. We can increase the number of points used in the training set. This would, in principle, reduce the model variance. Otherwise, we could use Bagging;
2. We could use an adjustment technique to correct the value of the loss function provided by the already trained models penalizing their complexity;
3. In this case Boosting is a viable option, since it uses a set of simple models, whose variance is limited, to generate a more complex overall model;
4. in this case Leave One Out is a viable option, which scales with the number of samples, and, therefore requires a lot of computational power to be applied. None the less, the estimates provided by this method of the method error is the most accurate one among the methods we covered.

Answer of exercise 5.18

1. NO, because if linear regression does not fit training data (underfitting), the features computed with PCA will not solve the problem as they are linear combinations of input variables.
2. YES, applying PCA and selecting the top K components we will allow to avoid collinearity in the resulting feature space.
3. BAGGING, because it might reduce the variance of the model. In contrast, boosting allows to reduce bias without increasing (significantly) the variance (however, in this example we have a low bias and high variance).
4. BOOSTING, because it can successfully reduce bias of a stable learner without increasing the variance which seems to be the problem of the learner in this case.
5. NO, because ridge regression with large regularization coefficient will be very stable (limited variance) and this would not allow to exploit significantly bagging.
6. FILTER, because a wrapper approach involves solving an optimization problem that requires training several models. In contrast, filter approaches only require to compute statistics on the features.
7. WRAPPER, because filter approaches assume features are independent and might not find the best subset.
8. LASSO, because it implicitly performs a feature selection that will get rid of the faulty sensors.
9. REGULARIZATION can help me improve the performance by reducing the variance. In particular RIDGE regression would be the most obvious choice. Another solution, if viable, is to increase the number of samples used for training.
10. ADJUSTED COMPLEXITY METRICS can be used to *correct* the training error taking into account also the model complexity.

Answer of exercise 5.19

1. NO, Model A is more complex and hence has a larger variance and probability of overfitting training data. So it would probably result in a worse test error.
2. NO, Model A is simpler and will probably have a larger bias resulting in a larger training error.
3. YES, the model is overfitting training data. Increasing K will decrease variance and could reduce overfitting.

4. NO, the model is overfitting training data. Decreasing the regularization coefficient will increase variance and possibly also overfitting.

5. NO, you should choose the value of K based on the average performance compute on all the 10 folds.

6. YES, cross-validation provides an unbiased estimate of the test error of each model. Once selected the model, it still is a good idea to use the whole data to train a possibly better model.

7. NO, the training error does not provide a useful estimate of the true error (test error) of the model. You could, instead, use some adjusted error measure to select the model.

8. NO, because even if a single training process is not expensive, if the dataset is very large, LOO will not be feasible. Instead, we should use K-fold cross-validation.