Total: /20 marks. Comments: Well done.
Question 1-2: 7/7 marks. Comments: Great understanding and explanation.
Question 3: 5/5 marks. Comments: Well done.
Question 4-5: 8/8 marks. Comments: Well understanding and clear presentation.

Name: Yanislav Donchev
User id: ydd1g16

1. Separating Two Gaussians

The covariance matrices were calculated using (1) [1]:

$$\boldsymbol{\Sigma} = \begin{pmatrix} S_x^2 & \rho S_x S_y \\ \rho S_x S_y & S_y^2 \end{pmatrix} \qquad (1)$$

Figure 1 shows the histograms of the two classes projected on three different direction vectors. The expectation is that a vector $\boldsymbol{\omega}$ which is parallel to the centroids of the two distributions (Figure 1 - left) would result in a good separation of the projected histograms. On the other hand, $\boldsymbol{\omega}$ that is perpendicular to the centroids (Figure 1 - right) results in an overlap. The subplot in the middle shows a vector which is in between the mentioned extremes. The results match the expectations. The middle plot has a clearer separation than the right one, but a worse separation than the one to the left.

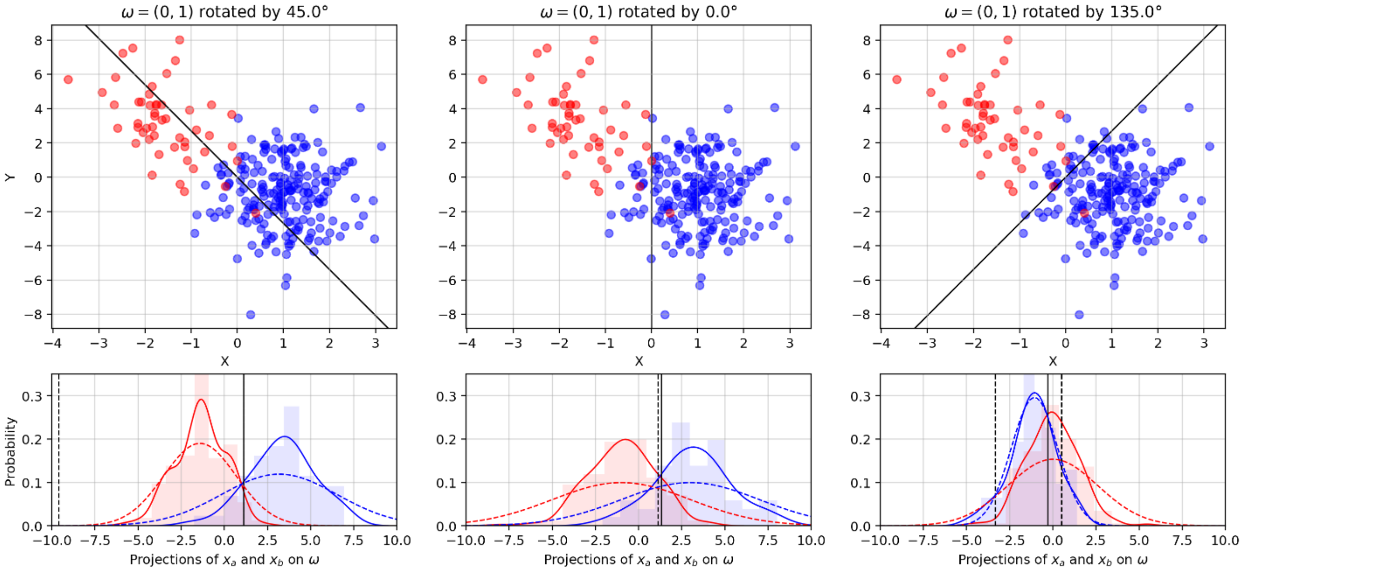

Figure 1. The probability densities of x_a and x_b projected on three different direction vectors $\boldsymbol{\omega}$. In addition to the histograms, there is a KDE plot (solid line) and the actual gaussian distribution derived from the generating parameters (dashed line). The black dashed lines show the optimal Bayes decision boundary, whilst the solid ones show the estimated boundary by the LDA. It is worth noting, that when the distributions are overlapping, and the standard deviation of one is much higher than the other (right plot), there are two decision boundaries. This can also be noticed in the 2D log-odds.

This separation can be scored by the Fisher ratio, which takes the square of the difference of the projected means and scales it in accordance to the standard deviations and the relative sizes of the classes. To do this, any vector $\boldsymbol{\omega}$ (in this case $\boldsymbol{\omega} = (0,1)$) can be picked and rotated by a range of angles between $0\ rad$ and $\pi\ rad$ (see Figure 2). Since we only care about the direction of the vector, the function will be periodic with a period of $\pi\ rad$. The ideal direction vector from the Fisher ratio is reasonably close to the one predicted in Figure 1.

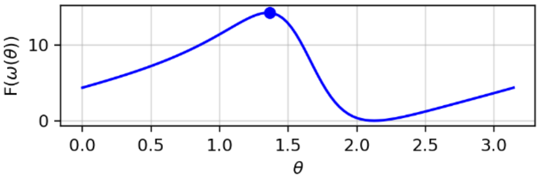

Figure 2. Fischer ratio for a range of θ angles. The peak value is 13.48 and it corresponds to $\boldsymbol{\omega}^* = (-0.96,\ 0.27)$. This is the $(0, 1)$ vector rotated by $1.30\ rad$

Figure 3 shows the optimal $\boldsymbol{\omega}$ vector together with equiprobable contour lines of the two distributions. The clear separation of the projections (Figure 3 - right) further confirms that this is the optimal vector. The overlap in the projection exists because the original 2D data is also slightly overlapped.

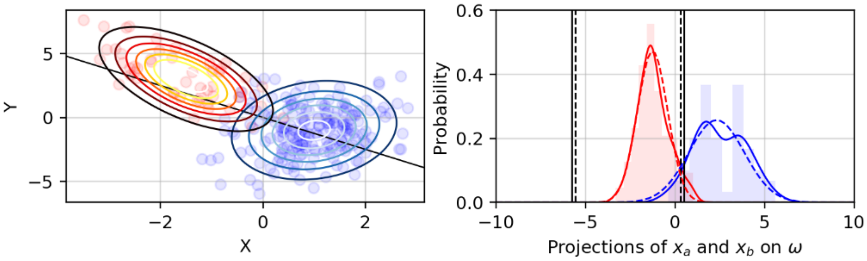

Figure 3. Equiprobable lines for the two classes and the optimal $\boldsymbol{\omega}$ vector (left) and histograms of the projections (right). It is worth noting that $\boldsymbol{\omega}$ is not exactly parallel to the centroids of the distributions. This is because of the difference between the covariance matrices and the relative sizes of the distributions (this affects the denominator of the Fisher ratio).

The log-odds (2) of the distributions can be rearranged using Bayes' theorem. This reveals that the log-odds are the sum of the logarithms of the ratios between the prior and posterior probabilities. The prior probability is $P(c) = \frac{n_c}{n}$, where n is the sum of elements in all classes [2].

$$\ln\left(\frac{P(c=a|x^n)}{P(c=b|x^n)}\right)=\ln\left(\frac{P(x^n|c=a)}{P(x^n|c=b)}\frac{P(c=b)}{P(c=a)}\right)=\ln\left(\frac{P(x^n|c=a)}{P(x^n|c=b)}\right)+\ln\left(\frac{P(c=a)}{P(c=b)}\right) \qquad (2)$$

The posterior probability (3) is [3]:

$$P(x^n|c)=\frac{\sqrt{|\boldsymbol{\Sigma}_c^{-1}|}}{2\pi}\exp\left(-\frac{1}{2}(\boldsymbol{x^n}-\boldsymbol{m_c})^{\boldsymbol{T}}\boldsymbol{\Sigma_c^{-1}}(\boldsymbol{x^n}-\boldsymbol{m_c})\right) \qquad (3)$$

Figure 4 shows the log-odds decision boundaries. The fact that the boundary gets pushed by the bigger class is due to the fact that the prior probability of that class increases, and hence the log-odds shift either up or down.

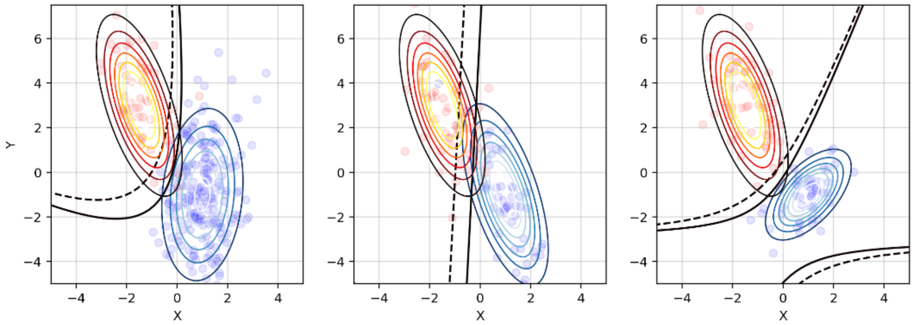

Figure 4. Decision boundaries for balanced (solid) and unbalanced (dashed) distributions. When the two classes have different covariances, the boundary is a parabola (left). When the classes have the same covariances, the boundary is a straight line (middle). When the classes are unbalanced (one class is 10 times bigger), the decision boundary gets "pushed" by the bigger class. An effect that was mentioned in Figure 1 can be seen on the right subplot, where two boundaries exist.

The unbalanced Fisher ratio would only give a different result if the projected variances are different. If they are the same, the balanced and unbalanced ratios will just be proportional $\frac{(\mu_a-\mu_b)^2}{2\sigma^2} \propto \frac{(\mu_a-\mu_b)^2}{(\pi_a+\pi_b)\sigma^2}$ (which is not important since we only care about $\boldsymbol{\omega}$ at the highest ratio and not the actual ratio). The datasets must also be unbalanced (one set having more elements than the other).

Figure 5 - left shows that the maximum Fisher ratio is at a different angle when the unbalanced formulation is used. Once again, the decision boundary gets "pushed" by the bigger class. The consequences of accounting for the different fractions

of data in each class is accounted for in the log-odds (2) in the last term, which is the logarithm of the ratio of the prior probabilities.

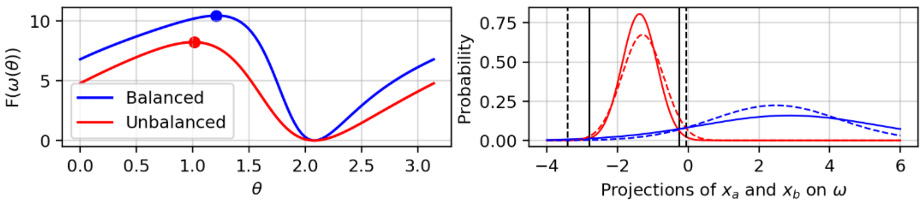

Figure 5. Fisher ratios (left) and projected probability densities (right). The dashed lines show the projections on the balanced $\boldsymbol{\omega}$. It is obvious that when the prior probabilities are considered, the decision boundary expands in favour of the bigger dataset.

2. Iris Dataset

First, the generalised eigenvalue problem needs to be set. For this, the within-class (4) and between-class (5) scatter matrices need to be computed [4]. $\boldsymbol{m_t}$ is the mean of all means. Since we only want to retain the discriminative parts of the data, the goal is to transform the data, so that the between class scatter is maximised, whilst the within class scatter gets minimised. This can be done by solving the generalised eigenvalue problem and projecting the data onto the optimal eigenvector.

$$\boldsymbol{\Sigma_w} = \sum_{c=1}^{3} \sum_{n=1}^{50} (\boldsymbol{x^n} - \boldsymbol{m_c})(\boldsymbol{x^n} - \boldsymbol{m_c})^{\boldsymbol{T}} \quad (4)$$

$$\boldsymbol{\Sigma_b} = \sum_{c=1}^{3} \boldsymbol{n_c}(\boldsymbol{m_c} - \boldsymbol{m_t})(\boldsymbol{m_c} - \boldsymbol{m_t})^{\boldsymbol{T}} \quad (5)$$

Since the rank of $\boldsymbol{\Sigma_w}$ is three (there are three classes), there can be no more than 2 non-zero eigenvalues and corresponding directions [5]. This means that the last two, almost-zero, eigenvalues that SciPy returns after solving the generalized eigenvalue problem are floating point errors and should be ignored. This leaves us with the two eigenvalues $\lambda_1 = 32.2$ and $\lambda_2 = 0.85$, with corresponding eigenvectors (6) and (7).

$$\boldsymbol{\omega_1} = (0.0684 \quad 0.1266 \quad -0.1816 \quad -0.2318) \quad (6)$$

$$\boldsymbol{\omega_2} = (0.002 \quad 0.1785 \quad -0.0769 \quad 0.2342) \quad (7)$$

The eigenvectors form the eigenbasis of the projected feature space. Their corresponding eigenvalues tell how much the data is stretched in this new feature space. Therefore, the eigenvector with highest eigenvalue should be the optimal direction vector ($\boldsymbol{\omega}^* = \boldsymbol{\omega_1}$). The projection of all classes is shown in Figure 6.

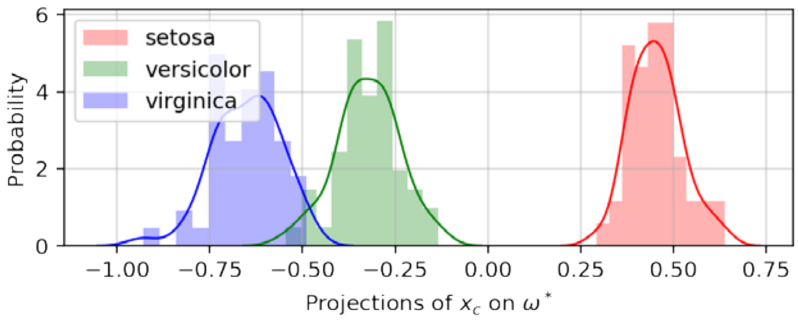

Figure 6. Projections of the three classes onto the optimal direction $\boldsymbol{\omega}^*$. There is a clear separation between the classes, with a slight overlap between the blue and the green class.

To confirm that this is in fact the best direction vector, a new projection is shown in Figure 7 on a vector $\boldsymbol{\omega} = \boldsymbol{\omega}^* + \boldsymbol{\alpha}$, where $\boldsymbol{\alpha}$ is the other generalized eigenvector. It is apparent that there is a greater overlap between *versicolor* and *virginica*, and the *setosa* class has been pulled closer to the other two. What is more, the projected variances have seemingly increased.

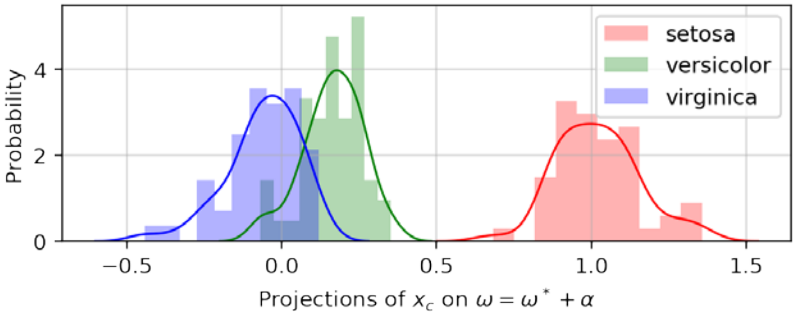

Figure 7. The projections on this dataset are worse than the one on Figure 6

The two eigenvectors are the basis vectors (forming an eigenbasis) of a two-dimensional plane in the four-dimensional feature space. Therefore, we can describe any vector in this plane by the linear combination of the two eigenvectors $\boldsymbol{\omega} = \boldsymbol{\omega}^* + \boldsymbol{\alpha}$. This is the reason that $\boldsymbol{\alpha}$ must be constructed of the other generalised eigenvector.

The generalised eigenvalue problem $\boldsymbol{\Sigma_b \omega} = \boldsymbol{\lambda \Sigma_w \omega}$ finds the eigenvectors, which form a two-dimensional basis in which the between scatter matrix is maximised and the within scatter matrix is minimised. The generalised matrix form of the Fisher ratio is (8) [5]. Plotting the Fisher ratio for different linear combinations of $\boldsymbol{\omega}$ (Figure 8) confirms that $\boldsymbol{\omega_1}$ (the eigenvector with highest eigenvalue) is in fact the best direction for highest separation.

$$F(w) = \frac{\omega^T \Sigma_b \omega}{\omega^T \Sigma_\mathrm{w} \omega} \qquad (8)$$

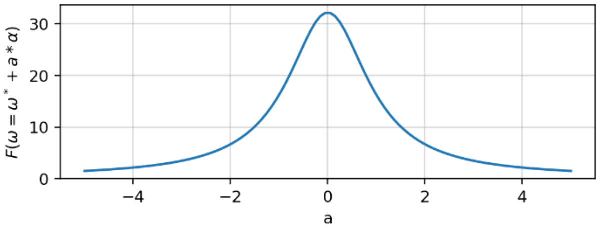

Figure 8. The variation of the Fisher ratio when different fractions of the second eigenvector are added to the optimal one. Highest ratio is at $a = 0$, meaning that $\boldsymbol{\omega}^* = \boldsymbol{\omega_1}$.

The benefits of this method over the one in the previous section is that it analytically returns the optimal direction vector, saving computational effort. The Fisher approach in the previous section relies on calculating all possible direction vectors and their corresponding Fisher ratios, which can be time consuming, and the final result depends on how fine the rotation angle step is. Both methods result in a linear decision boundary.

In comparison, the two-dimensional log-odds approach (the quadratic discriminant analysis (QDA)) results in a quadratic decision boundary. Therefore, there will be cases where the QDA more accurately models the problem, when compared to the linear approaches [3].

3. Linear Regression with non-Linear Functions

3.1. Performing Linear Regression

To perform gradient descent, the gradient (10) of the loss function (9) must be calculated. The last term (square of L_2 norm) penalises the weights for getting large. The λ factor gauges how much to penalise.

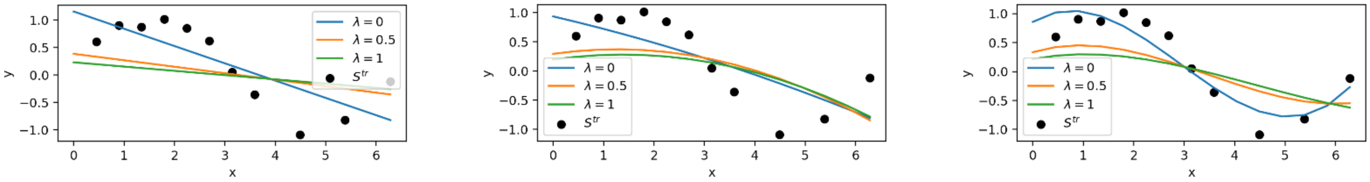

Figure 9. Gradient descend for $p = 1$ ($rate = 60e - 3$) (left), $p = 2$ ($rate = 2e - 5$)(middle), and $p = 3$ ($rate = 2e - 5$) (right). It is expected that the third order would have the best fit since it has two concaves (similarly to the sinusoid). The generalisation factor has the effect of squishing the weights towards zero. The number of iterations is 300000.

The input dataset has 30 points with additive gaussian noise which has a mean of 0 and a standard deviation of 1. The training – testing ratio is 70/30. Figure 9 shows the learned models for $p \in [1,3]$ and different regularisation strengths.

$$\boldsymbol{L}(\boldsymbol{w}) = \sum_{\boldsymbol{n}=\boldsymbol{1}}^{\boldsymbol{N}} \left(\boldsymbol{y}^{\boldsymbol{n}} - \sum_{\boldsymbol{j}=\boldsymbol{1}}^{\boldsymbol{p}+\boldsymbol{1}} \boldsymbol{A}_{\boldsymbol{nj}}\, \boldsymbol{\omega}_{\boldsymbol{j}} \right)^2 + \lambda \|\boldsymbol{\omega}\|_{\boldsymbol{2}}^{\boldsymbol{2}} \qquad (9)$$

$$\frac{\partial L}{\partial \omega_i} = -2\left(\boldsymbol{A}^{\boldsymbol{T}}(\boldsymbol{y} - \boldsymbol{A\omega})\right)_i + 2\lambda\boldsymbol{\omega}_{\boldsymbol{i}} \qquad (10)$$

Increasing the order of the polynomial increases the constraints on the learning rate. In particular, higher orders require smaller learning rates. This is because, very high powers of x exist and small changes in ω result in large changes in the gradient, pushing it to infinity. However, a small learning rate results in a very slow convergence and a high number of iterations. Fortunately, an analytical approach is available, which gives the optimal weights without iterations (Figure 10). It can be seen in this figure that for a low order, the model can not overfit and hence the regularisation results in a worse model.

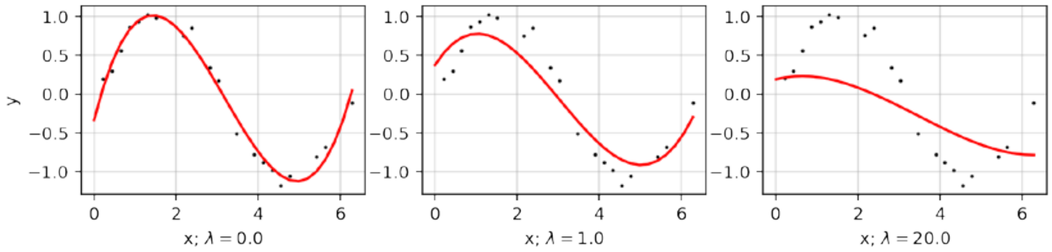

Figure 10. 3^{rd} order polynomial fit using the analytical formula with different regularisation factors.

Table 1 compares the weights for models trained on the same data. After 300000 iterations, the gradient descent is still far from the analytical weights.

Method	Order	Weights
Gradient descent	1	$[1.15 \quad -0.314]$
Gradient descent	2	$[0.933 \quad -0.201 \quad -0.012]$
Gradient descent	3	$[0.846 \quad 0.487 \quad -0.386 \quad 0.045]$
Analytical	1	$[0.647 \quad -0.218]$
Analytical	2	$[0.683 \quad -0.254 \quad 0.006]$
Analytical	3	$[-0.6133 \quad 2.123 \quad -0.907 \quad 0.0942]$

Table 1. Comparison of weights for three orders of polynomials.

Next, the effect of changing the regularisation factor and the order of the polynomial is explored. This is done by evaluating the mean of the squared residuals for different parameters on a constant set of testing data (Figure 11). It is hard to find the optimal polynomial and regularisation factor, since this is dependent on the added noise and the chosen training set. However, it is apparent that polynomials of order $p \in [3, 10]$ give the lowest errors. Lower order polynomials underfit and higher order polynomials overfit to the training data. Regularisation factors $\lambda \in [0.1{,}0.3]$ result in the best results for high order polynomials. Low factors don't manage to prevent overfitting, but factors that are too high eliminate the important trends in the data [6]. Polynomials of order $p \in [3{,}4]$ give lowest errors with $\lambda = 0$. This is because the order is too low for the data to overfit and reducing the weights towards zero results in underfitting.

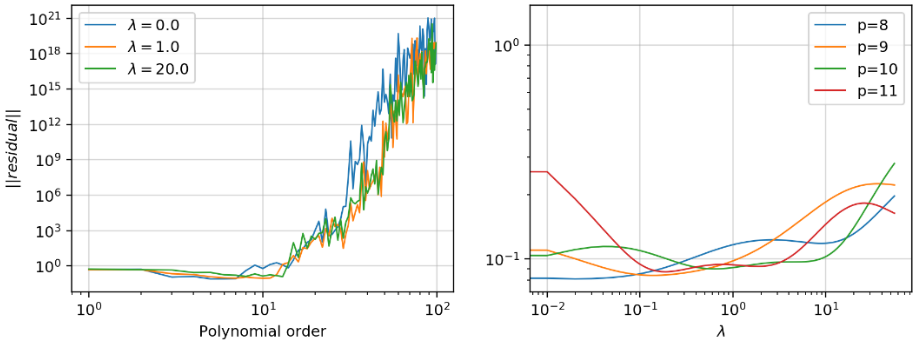

Figure 11. Change of the mean of squared residuals with polynomial order (left) and regularisation factor (right).

3.2. How Does Linear Regression Generalise

A new distribution of 100 points was split into $|S^{tr}|/|S^{ts}| = 4$. The training set was then split into 10 overlapping subsets S_i such that $|S_i|/|S^{tr}| = 1/3$. Therefore, $|S_i| = 20$ and $|S^{ts}| = 20$. As seen in Figure 12, training to different sets S_i, picked from a larger set, results in different models.

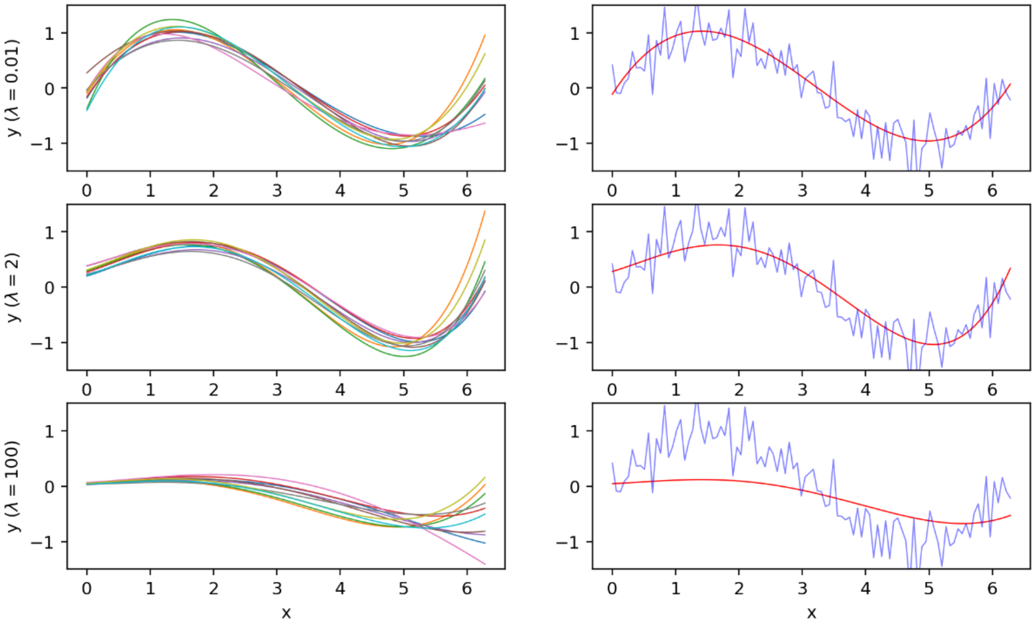

Figure 12. Ten 5th order polynomials trained on the S_i sets (left). The output of the average model (right). (adapted from [6])

The bias is a measure that shows the property of the model to persistently learn wrong relations. The variance shows the consistency of the model, when different training subset is chosen. A high variance means that the model is very sensitive to the choice of S_i. Using (11) and (12) [6], the variance can be calculated. The bias is simply the mean of the average loss of all models M_i.

$$\bar{y}(x) = \frac{1}{10}\sum_{i=1}^{10} y^{i}(x) \tag{11}$$

$$variance = \frac{1}{N}\sum_{n=1}^{N} \frac{1}{10}\sum_{i=1}^{10}\left\{y^{i}(x_n) - \bar{y}(x_n)\right\}^2 \tag{12}$$

Intuitively, a larger regularisation factor (and hence smaller weights) should result in higher bias and smaller variance. However, the polynomials in Figure 12 do not manage to fit well to the right side of the sinusoid and even high regularisation factors do not reduce the variance. The bias and variance relationship is shown in Figure 13. According to [6], the sum of $bias^2$ and $variance$ is related to the loss of the model. Therefore, the lowest point of the curve corresponds to the optimal regularisation factor ($\lambda \sim 0.1$) and should result in the best bias-variance balance. Note that this value is also in the range defined in section 3.1.

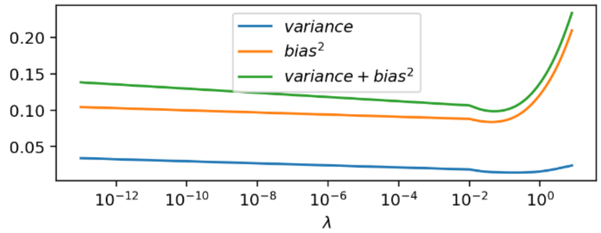

Figure 13. The change of bias and variance with λ (adapted from [6])

Another view on how the bias-variance changes can be seen in Figure 14. Increasing λ increases the bias.

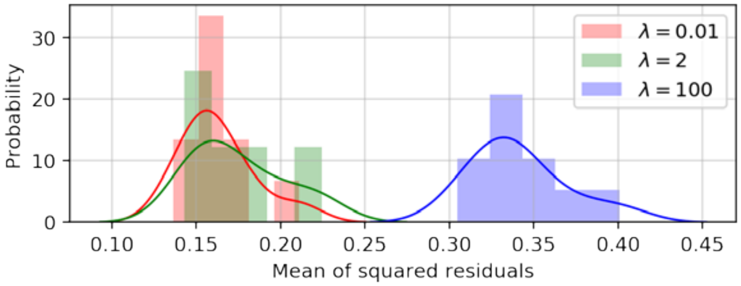

Figure 14. The bias-variance distribution for different regularisation factors.

References

[1] C. . M. Grinstead, “Continuous Random Variables,” in *Introduction to Probability*, p. 281.

[2] C. M. Bishop, “Probability Theory,” in *Pattern Recognition and Machine Learning*, Springer, 2006, p. 17.

[3] G. James, “Linear Discriminant Analysis for p>1,” in *An Introduction to Statistical Learning*, Springer, 2014, p. 156.

[4] C. M. Bishop, “Linear Models for Classification,” in *Pattern Recognition and Machine Learning*, Springer, 2006, p. 192.

[5] D. Barber, “Canonical variates,” in *Bayesian Reasoning and Machine Learning*, 2011, pp. 325-327.

[6] C. M. Bishop, “The Bias-Variance Decomposition,” in *Pattern Recognition and Machine Learning*, Springer, 2006, pp. 148-150.