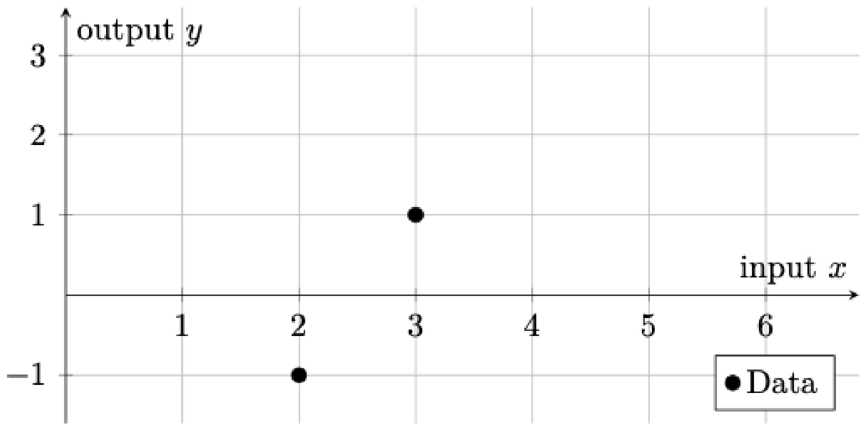
ARTIFICIAL INTELLIGENCE (E016350A)
GHENT UNIVERSITY
AY 2024/2025

Aleksandra Pizurica
Asst: Nicolas Nadisic
E-mail: ai@lists.ugent.be

Solutions: Intro to supervised learning – Part 1

1. **Linear regression**

   (a) Assume that you record a scalar input x and a scalar output y. First, you record $x^{(1)} = 2$, $y^{(1)} = -1$, and thereafter $x^{(2)} = 3, y^{(2)} = 1$. Fit a linear regression model $\hat{y} = h_{\mathbf{w}}(x) = w_0 + w_1 x$ using the squared error loss. Use the model to predict the output for the test input $x_t = 4$, and add the model to the plot below:



   (b) Now, assume you have made a third observation $y^{(3)} = 2$ for $x^{(3)} = 4$ (is that what you predicted in (a)?). Update the parameters $\mathbf{w}$ to all 3 data samples, add the new model to the plot (together with the new data point) and find the prediction for $x_t = 5$.

   (c) Repeat (b), but this time using a model without intercept term, i.e., $h_{\mathbf{w}}(x) = w_1 x$.

   (d) Repeat (b), but now using Ridge Regression with the regularization parameter $\lambda = 1$ instead of the ordinary least squares.

Solution:

(a) Our data matrix is $\mathbf{X} = \begin{bmatrix} 1 & 2 \\ 1 & 3 \end{bmatrix}$ and the vector of target values is $\mathbf{y} = \begin{bmatrix} -1 \\ 1 \end{bmatrix}$.

The least squares solution for the weights is given by the normal equation $\mathbf{X}^\top \mathbf{X}\mathbf{w}^* = \mathbf{X}^\top \mathbf{y}$. We thus solve it

$$\begin{bmatrix} 1 & 2 \\ 1 & 3 \end{bmatrix}^\top \begin{bmatrix} 1 & 2 \\ 1 & 3 \end{bmatrix} \mathbf{w}^* = \begin{bmatrix} 1 & 2 \\ 1 & 3 \end{bmatrix}^\top \begin{bmatrix} -1 \\ 1 \end{bmatrix} \quad \Rightarrow \quad \begin{bmatrix} 2 & 5 \\ 5 & 13 \end{bmatrix} \mathbf{w}^* = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \quad \Rightarrow \quad \mathbf{w}^* = \begin{bmatrix} -5 \\ 2 \end{bmatrix}$$

The predictor becomes $h_{\mathbf{w}}(x) = -5 + 2x$, and thus for $x_t = 4$ the predicted output is $\hat{y}_t = -5 + 2 \times 4 = 3$.

(b) Again, the solution is given by the normal equations

$$\mathbf{X}^\top \mathbf{X}\mathbf{w}^* = \mathbf{X}^\top \mathbf{y} \Rightarrow \left(\begin{bmatrix} 1 & 1 & 1 \\ 2 & 3 & 4 \end{bmatrix} \begin{bmatrix} 1 & 2 \\ 1 & 3 \\ 1 & 4 \end{bmatrix} \right) \mathbf{w}^* = \begin{bmatrix} 1 & 1 & 1 \\ 2 & 3 & 4 \end{bmatrix} \begin{bmatrix} -1 \\ 1 \\ 2 \end{bmatrix} \Rightarrow \mathbf{w}^* = \frac{1}{6} \begin{bmatrix} -23 \\ 9 \end{bmatrix}$$

The prediction for $x_t = 5$ is hence $\hat{y}_t = \frac{1}{6}(-23 + 9 \times 5) = \frac{11}{3} \approx 3.67$.

(c) With no intercept term, $\mathbf{w}$ reduces to w_1, the data matrix is $\mathbf{X} = \begin{bmatrix} 2 \\ 3 \\ 4 \end{bmatrix}$, and thus

$$\mathbf{X}^\top \mathbf{X}\mathbf{w}^* = \mathbf{X}^\top \mathbf{y} \Rightarrow \begin{bmatrix} 2 & 3 & 4 \end{bmatrix} \begin{bmatrix} 2 \\ 3 \\ 4 \end{bmatrix} \mathbf{w}^* = \begin{bmatrix} 2 & 3 & 4 \end{bmatrix} \begin{bmatrix} -1 \\ 1 \\ 2 \end{bmatrix} \Rightarrow \mathbf{w}^* = w_1 = \frac{9}{29}$$

The prediction is now $\hat{y} = w_1^* x$ and for $x_t = 5$ we have $\hat{y}_t = \frac{45}{29} \approx 1.55$.

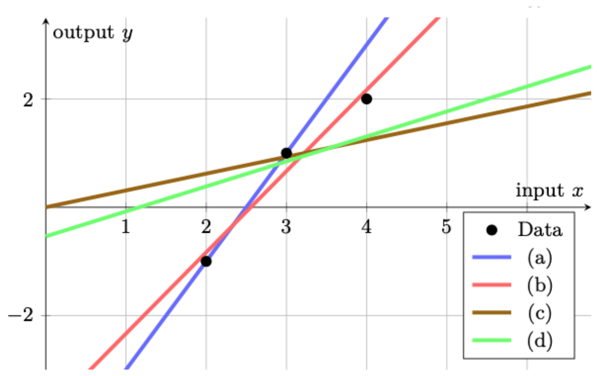
(d) Ridge Regression also has a closed form solution, and in particular, the normal equation now becomes $(\mathbf{X}^\top \mathbf{X} + \lambda I)\mathbf{w}^* = \mathbf{X}^\top \mathbf{y}$, i.e., $\mathbf{w}^* = (\mathbf{X}^\top \mathbf{X} + \lambda I)^{-1}\mathbf{y}$, where I is the identity matrix. Here, $\lambda = 1$ and $I = I_2$ (it is of size 2×2). We thus have:

$$(\mathbf{X}^\top \mathbf{X} + I_2)\mathbf{w}^* = \mathbf{X}^\top \mathbf{y} \quad \Rightarrow \quad \left(\begin{bmatrix} 1 & 1 & 1 \\ 2 & 3 & 4 \end{bmatrix} \begin{bmatrix} 1 & 2 \\ 1 & 3 \\ 1 & 4 \end{bmatrix} + \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \right) \mathbf{w}^* = \begin{bmatrix} 1 & 1 & 1 \\ 2 & 3 & 4 \end{bmatrix} \begin{bmatrix} -1 \\ 1 \\ 2 \end{bmatrix}$$

$$\Rightarrow \quad \mathbf{w}^* = \frac{1}{39} \begin{bmatrix} -21 \\ 18 \end{bmatrix}$$

The prediction for $x_t = 5$ is hence $\hat{y}_t = w_0^* + w_1^* \times 5 = \frac{69}{39} \approx 1.77$.

Solution:
(Continued) The four predictors from (a) – (d) are shown in the figure below:



2. **Deriving least squares linear regression from maximum likelihood**

   Assume a linear regression model

$$y = w_0 + w_1 x_1 + \ldots w_d x_d + \epsilon$$

   where the errors ϵ are independent, identically distributed (i.i.d.) and follow a normal distribution with zero mean $\epsilon \sim \mathcal{N}(0, \sigma_\epsilon^2)$.

   Show that the maximum likelihood estimate for the weights in this case is equivalent to the least squares solution, i.e., to the weight optimization under the squared error loss.

Solution:
The maximum likelihood estimate of $\mathbf{w}$ is

$$\mathbf{w}_{ML} = \arg\max_{\mathbf{w}} P(\mathbf{y}|\mathbf{X};\mathbf{w})$$

where $\mathbf{X}$ is the data matrix, whose rows are the input examples $(\mathbf{x}^{(i)})^\top = [x_1^{(i)} \dots x_d^{(i)}]$, $i = 1, \dots N$, and $\mathbf{y} = [y^{(1)} \dots y^{(N)}]^\top$ are the corresponding target values. Since ϵ is independent for each data point, we can write the joint probability in the expression above as the product of the independent terms, for each data point and thus we have

$$\mathbf{w}_{ML} = \arg\max_{\mathbf{w}} \prod_{i=1}^{N} P(y^{(i)}|\mathbf{x}^{(i)};\mathbf{w})$$

Since the errors are assumed to be normally distributed, $\epsilon \sim \mathcal{N}(0, \sigma_\epsilon^2)$, we can further specify the likelihood as

$$\begin{aligned}
P(\mathbf{y}|\mathbf{X};\mathbf{w}) &= \prod_{i=1}^{N} P(y^{(i)}|\mathbf{x}^{(i)};\mathbf{w}) \\
&= \prod_{i=1}^{N} \mathcal{N}(\underbrace{w_0 + w_1 x_1^{(i)} + \ldots + w_d x_d^{(i)} - y^{(i)}}_{\epsilon}|0, \sigma_\epsilon^2) \\
&\propto \prod_{i=1}^{N} e^{-\frac{1}{2\sigma_\epsilon^2}(w_0 + w_1 x_1^{(i)} + \ldots + w_d x_d^{(i)} - y^{(i)})^2} \\
&= \exp\left(\sum_{i=1}^{N} -\frac{1}{2\sigma_\epsilon^2}\left(w_0 + w_1 x_1^{(i)} + \ldots + w_d x_d^{(i)} - y^{(i)}\right)^2\right)
\end{aligned}$$

Instead of maximizing this distribution directly, we can equivalently maximize its logarithm (since log is a monotonically increasing function)

$$\begin{aligned}
\arg\max_{\mathbf{w}} P(\mathbf{y}|\mathbf{X};\mathbf{w}) &= \arg\max_{\mathbf{w}} \log P(\mathbf{y}|\mathbf{X};\mathbf{w}) \\
&= \arg\max_{\mathbf{w}} \left\{ \sum_{i=1}^{N} -\frac{1}{2\sigma_\epsilon^2}\left(w_0 + w_1 x_1^{(i)} + \ldots + w_d x_d^{(i)} - y^{(i)}\right)^2 \right\} \\
&= \arg\min_{\mathbf{w}} \left\{ \sum_{i=1}^{N} \left(w_0 + w_1 x_1^{(i)} + \ldots + w_d x_d^{(i)} - y^{(i)}\right)^2 \right\}
\end{aligned}$$

and this is exactly the least squares solution for $\mathbf{w}$. So, we gave the maximum likelihood interpretation to the weight optimization under the squared error loss function.

3. Consider the following training data

x	y
1	3
2	1
3	0.5

Suppose the data comes from a model $y = cx^\beta + noise$, for unknown constants c and β. Use least squares linear regression to find an estimate of c and β.

Solution:

Write $\underbrace{\log y}_{y'} = \underbrace{\log c}_{w_0} + \underbrace{\beta}_{w_1} \underbrace{\log x}_{x'}$

Data becomes:

x'	y'
0	$\log 3$
$\log 2$	0
$\log 3$	$\log 0.5$

The data matrix, the vector of target vales and the weight vector are:

$$\mathbf{X} = \begin{bmatrix} 1 & 0 \\ 1 & \log 2 \\ 1 & \log 3 \end{bmatrix} ; \ \mathbf{y} = \begin{bmatrix} \log 3 \\ 0 \\ \log 5 \end{bmatrix} ; \ \mathbf{w} = \begin{bmatrix} \log c \\ \beta \end{bmatrix}$$

Solve $\min_{\mathbf{w}} \|\mathbf{y} - \mathbf{X}\mathbf{w}\|^2$

Solution is given by $\mathbf{w}^* = (\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{X}^\top \mathbf{y}$

$$\mathbf{w}^* = \begin{bmatrix} 0.899 \\ -0.5 \end{bmatrix}$$

$c = e^{0.899} = 2.45; \quad \beta = -0.5$

4. Consider the following data set

Item	x_1	x_2	Class y
A	1	2	yes=1
B	2	1	yes=1
C	1	1	no=0
D	1	0	no=0

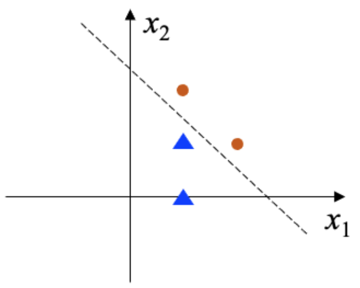
(a) Is this dataset linearly separable? Explain.

(b) We are training a perceptron (linear classifier with a hard threshold) on this data. We append $x_0 = 1$ to each data point to account for the bias term (e.g. for item A, $(1, 2)$ becomes $(1, 1, 2)$). Suppose the current weigths to be $\mathbf{w} = (0, -1, 1)$. Assume a learning rate $\alpha = 0.2$. How should the weights be updated if point A is considered?

(c) How should the weights be updated if point B is considered?

Solution:

(a) Yes. The data points can be separated by a line, e.g., $x_1 + x_2 = 2.5$.



(b) If we denote the prediction of a linear classifier with a hard threshold by $\hat{y} = h_{\mathbf{w}}(\mathbf{x}) = Threshold(\mathbf{w} \cdot \mathbf{x})$, where $Threshold(z) = 1$ if $z \geq 0$ and 0 otherwise, the perceptron update rule is

$$w_j \leftarrow w_j + \alpha(y - h_{\mathbf{w}}(\mathbf{x}))x_j, \quad j = 0, 1, 2$$

Or we can write more compactly:

$$\mathbf{w} \leftarrow \mathbf{w} + \alpha(y - h_{\mathbf{w}}(\mathbf{x}))\mathbf{x}$$

For the point A, the score is $\mathbf{w} \cdot \mathbf{x} = (0, -1, 1) \cdot (1, 1, 2) = 1 \geq 0$. The prediction $\hat{y} = h_{\mathbf{w}}(\mathbf{x}) = 1$ is correct, we thus have $y - h_{\mathbf{w}}(\mathbf{x}) = 0$ and the weights will not be updated.

(c) For B, the score is $\mathbf{w} \cdot \mathbf{x} = (0, -1, 1) \cdot (1, 2, 1) = -1 \leq 0$ and thus the predicted label is $\hat{y} = h_{\mathbf{w}}(\mathbf{x}) = 0$, while the given target label is $y = 1$. The weights are now updated as

$$\begin{aligned}\mathbf{w} \leftarrow & \mathbf{w} + \underbrace{\alpha}_{0.2} \underbrace{(y - h_{\mathbf{w}}(\mathbf{x}))}_{1} \mathbf{x} \\ = & (0, -1, 1) + 0.2 \times 1 \times (1, 2, 1) \\ = & (0.2, -0.6, 1.2)\end{aligned}$$

5. You are asked to use regularized linear regression to predict the target $y \in \mathbb{R}$ from the eight-dimensional feature vector $\mathbf{x} \in \mathbb{R}^8$. Let us define the model $y = \mathbf{w}^T\mathbf{x}$ and consider the following objective functions:

$$\min_{\mathbf{w}} \sum_{i=1}^{n} \left(y^{(i)} - \mathbf{w}^T\mathbf{x}^{(i)}\right)^2 \tag{1}$$

$$\min_{\mathbf{w}} \sum_{i=1}^{n} \left(y^{(i)} - \mathbf{w}^T\mathbf{x}^{(i)}\right)^2 + \lambda \sum_{j=1}^{8} w_j^2 \tag{2}$$

$$\min_{\mathbf{w}} \sum_{i=1}^{n} \left(y^{(i)} - \mathbf{w}^T\mathbf{x}^{(i)}\right)^2 + \lambda \sum_{j=1}^{8} |w_j| \tag{3}$$

(a) Circle regularization terms in the objective functions above.

(b) For large values of λ in objective (2) the *bias* would:

- increase
- decrease
- remain unaffected?

(c) For large values of λ in objective (3) the *variance* would:

- increase
- decrease
- remain unaffected?

(d) The following table contains the weights learned for all three objective functions (not in any particular order). Beside each objective write the appropriate column label (A, B, or C):

	Column A	Column B	Column C
w_1	0.60	0.38	0.50
w_2	0.30	0.23	0.20
w_3	-0.10	-0.02	0.00
w_4	0.20	0.15	0.09
w_5	0.30	0.21	0.00
w_6	0.20	0.03	0.00
w_7	0.02	0.04	0.00
w_8	0.26	0.12	0.05

- Objective (1): ____________
- Objective (2): ____________
- Objective (3): ____________

Solution:

(a) The regularization term in (2) is $\lambda \sum_{j=1}^{8} w_j^2 = \lambda \|\mathbf{w}\|_2^2$ (Tikhonov).

The regularization term in (3) it is $\lambda \sum_{j=1}^{8} |w_j| = \lambda \|\mathbf{w}\|_1$ (LASSO).

(b) *Bias* is the expected deviation between the predicted value and the true value. When λ is large, then the regularization term dominates the objective function. Thus, the bias would increase.

(c) *Variance* is an error from sensitivity to small changes in the training set. We can similarly as in (b) reason that for large λ, the variance will decrease.

(d) Columns A, B and C correspond to the objectives (1), (2) and (3), respectively. The regularization term in the objective tells us which w solutions are preferred.

6. Alice is building a model to know if a student will pass the exam or not. She has the hypothesis that the hours of study will be a good indicator of passing or not. She surveys her friends, 7 who failed the exam and 7 who passed, and asks them how many hours they spent studying. Here is what she finds:

   - Number of hours of study for each student who failed: $[1, 2, 2, 3, 5, 5, 6]$
   - Number of hours of study for each student who passed: $[5, 5, 7, 9, 9, 10, 11]$

   She trains a logistic regression classifier on the data and plots the classifier against the data, see Figure 1.

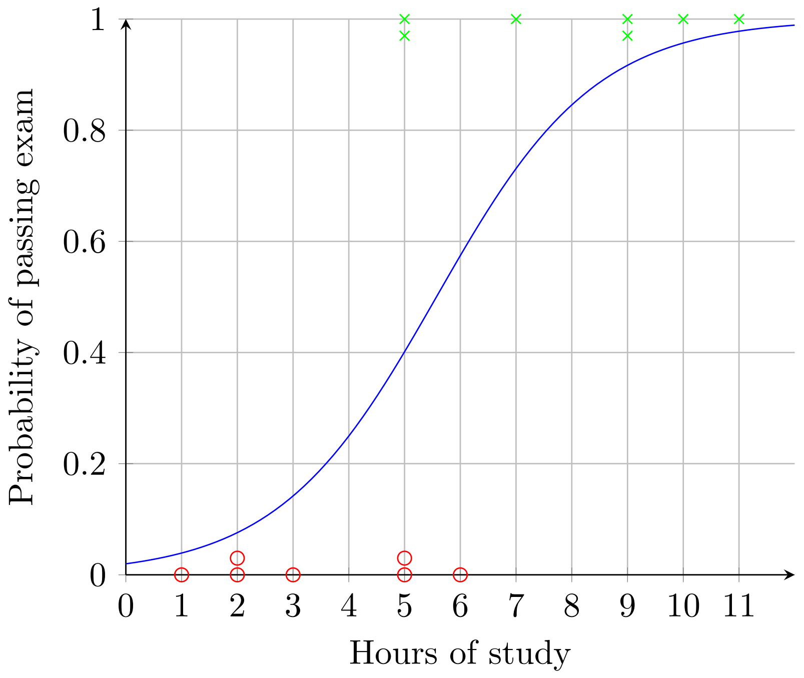


Figure 1: Plot of Alice's classifier model. The blue line represents the output of the model.

   Answer the following questions assuming the logistic regression model:

   (a) Consider a student who spent 5 hours studying. According to the model, what is roughly the probability that he will fail the exam? The probability that he will pass?

   (b) How many hours a student must study for the model to guarantee without a doubt that she will pass?

   (c) How is a logistic regression model normally turned into a binary classifier? If you turn the model into a classifier in this way, what is the accuracy of the classifier on the training data?

   (d) It is most important to reduce false negatives, i.e. we want to avoid that a pass is classified as a fail. To achieve that, we want to avoid false negatives in the training dataset. How can this goal be described in terms of training precision and recall? How can the logistic regression classifier be modified to try to achieve this goal?

Solution:

(a) We read from the graph that $P(0|x = 5) = 0.6$ and $P(1|x = 5) = 0.4$.

(b) You can never be sure with a logistic regression model, there is no way to get a probability of 1.

(c) This is normally done by choosing the class 1 if $P(1|x) \geq 0.5$. We see from the graph that this classify 5 pass correctly, 2 pass incorrectly, 6 fails correctly and 1 incorrectly. Altogether 11 out of 14 are classified correctly, yielding and accuracy of 11/14.

(d) The precision and recall are defined as follows:

$$Precision = \frac{tp}{tp + fp}; \qquad Recall = \frac{tp}{tp + fn}$$

where tp denotes 'true positive', fp 'false positive', and fn 'false negative'.

To minimize the false negatives for pass we should aim at a good recall (small fn for pass). Equivalently, we can formulate this as aiming at a good precision for fail (because then we minimize the false positive for fail, i.e., the fraction of pass that was detected as fail, which are the false negatives for pass).

We achieve this by changing the threshold to a value different from 0.5. For the training data, a threshold of 0.4 would suffice to remove the false negatives for pass completely. If we want to be prepared for more variation in the test data, we could set the threshold even lower. Of course, we risk having more false positives that way.