(Adv.) Machine Learning Exercises

Week 1 Exercises

1.5.1: *Imagine we have two possibilities: We can scan and email the image, or we can use an optical character reader (OCR) and send the text file. Discuss the advantage and disadvantages of the two approaches in a comparative manner. When would one be preferable over the other?*

Solution:
Dependent on the OCR software, properties such as font, size, or personal touch if the text is handwritten will be lost - and while OCR is not perfect, it should identify ambiguous cases and transmit them as they are. However, text files are generally much smaller than image files obtained from scanning, but the scanned document may also contain visuals like diagrams, pictures, etc.
OCR is good if we have high volume, good quality documents; for documents of few pages with small amount of text, it is better to transmit the image.

1.5.5: *In basket analysis, we want to find the dependence between two items X and Y. Given a database of customer transactions, how can we find these dependencies? How would we generalize this to more than two items?*

Solution:
As mentioned in section 1.2.1, we are interested in finding association rules between items, i.e., an implication $X \to Y$, where X and Y are items from the database. There are several ways of doing this; section 3.5 mentions three measures that are frequently calculated: confidence, support, and lift.

Confidence indicates how often the association rule is true, or in our case the proportion of the transactions that contain X which also contain Y:

$$\text{Confidence}(X \to Y) \equiv P(Y|X) = \frac{P(X,Y)}{P(X)} = \frac{\#\{\text{customers who bought } X \text{ and } Y\}}{\#\{\text{customers who bought X}\}}.$$

This is also the conditional probability $P(Y|X)$. The closer this value is to 1 and the larger it is than $P(Y)$, the more likely it is for a customer who purchased item X to also have purchased item Y, and therefore the more confident we are that the rule is true.

Support indicates how frequently the items are sold:

$$\text{Support}(X,Y) \equiv P(X,Y) = \frac{\#\{\text{customers who bought } X \text{ and } Y\}}{\#\{\text{customers}\}}.$$

Support shows the statistical significance of the association rule, whereas confidence shows the strength of the rule. That's why we are also interested in maximizing support, because even if there is a dependency with a strong confidence value, if the number of customers is small, the rule is worthless.

Lift indicates whether X and Y are independent:

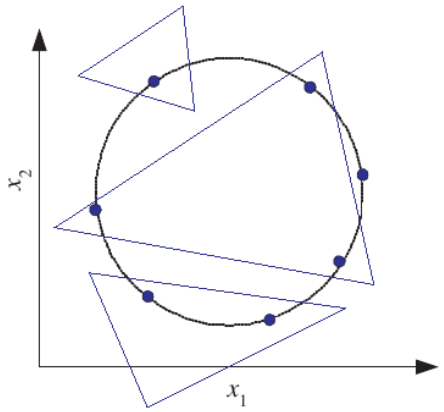
$$\text{Lift}(X \to Y) \equiv \frac{P(X,Y)}{P(X)P(Y)} = \frac{P(Y|X)}{P(Y)}.$$

If X and Y are independent, we expect the value of lift to be close to 1. If the value of lift is more than 1, we can say that makes X and Y more likely, and if it is less than 1, having X makes Y less likely.

All of these are easily generalized to more than two items. For example, with a three-item set $\{X, Y, Z\}$ we can look for a rule such as $X, Z \to Y$, i.e., $P(Y|X, Z)$.

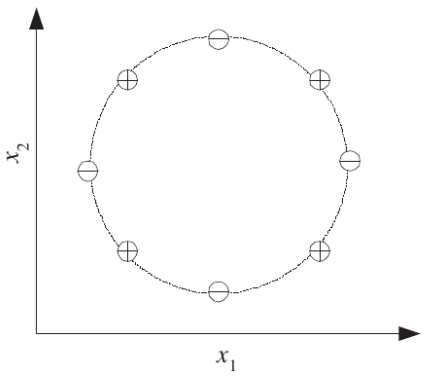
2.10.9: *Show that the VC dimension of the triangle hypothesis class is 7 in two dimensions. (Hint: For best separation, it is best to place the seven points equidistant on a circle.)*

If we place 7 points on a circle, all possible labelings of those 7 points can be separated using a triangle, as illustrated by the figure below. This is because one of the sets of positive and negative examples form at most 3 contiguous blocks. One edge of the triangle can therefore be used to cut off each block. Hence, there exists a set of 7 points in two dimensions that can be shattered.



These seven points can be separated using a triangle no matter how they are labeled.

The same logic applies for 8 points, however, the sets can now form up to 4 contiguous blocks. The figure below illustrates a labeling that cannot be separated using a triangle. Intuitively, you would have to intersect the circle at least 8 points, but any triangle only intersects a circle in at most 6 points.



These eight points with this labeling cannot be separated using a triangle.

2.10.11: *One source of noise is error in the labels. Can you propose a method to find data points that are highly likely to be mislabeled?*

Solution:
If an observation is surrounded by many observations with a different label, then it is most probably mislabeled. Using neighbour-based techniques makes it possible to check for this (see chapter 8). A related area of data mining called "anomaly detection" refers to the identification of observations that differ significantly from the rest of the data.

Week 2 Exercises

3.7.6: *Somebody tosses a fair coin and if the result is heads, you get nothing; otherwise, you get \$5. How much would you pay to play this game? What if the win is \$500 instead of \$5?*

Solution:
The idea of this exercises is to give a students the idea on how to calculate risks. As it can be seen from the description from the book we are trying to minimize the expected outcome of the different scenarios.

In this case there is no specific answer since there are no restrictions made in the instructions nor different scenarios. Because of this what it can be said is how much should you invest to have a cap on how much money should you invest at most (since we assume you cannot enter this deal for free).

At most, you want to go even (on average) from this deal, so we need to calculate

$$E[C_i|x] = 0$$

where:

$$E[C_i|x] = (0-x) * 1/2 + (5-x) * 1/2 = -x + 5/2$$

Then $x = 5/2$, which means that at most you should bet is \$2.5

3.7.8: *Generalize the confidence and support formulas for basket analysis to calculate k dependencies, namely, $P(Y|X_1, ..., X_k)$.*

Solution:
The idea of this exercise is to show that the apriori algorithm can be generalized to as many items as possible, it does not only work for pair of items.

The rule we are interested to find here is $X_1, ..., X_k \to Y$

Support will be generalized as follows:

$$\text{Support}(Y, X_1, X_2, ..., X_k) \equiv P(Y, X_1, X_2, ..., X_k) = \frac{\#\{\text{customers who bought } Y,\ X_1, ...,\ X_k\}}{\#\{\text{customers}\}}.$$

Confidence will be generalized to:

$$\text{Confidence}(X_1, ..., X_k \to Y) \equiv P(Y|X_1, ..., X_k) = \frac{P(X_1, \ldots, X_k, Y)}{P(X_1, \ldots, X_k)} = \frac{\#\{\text{customers who bought } Y,\ X_1, ...,\ X_k\}}{\#\{\text{customers who bought } X_1, X_2, ..., X_k\}}.$$

3.7.9: *Show that as we move an item from the ~~consequent to the antecedent~~ antecedent to the consequent, confidence can never increase: confidence($ABC \to D$) $\geq$ confidence($AB \to CD$)*

Solution:

To solve this problem we need to look at how the conditional probabilities look like. This are:

$$\text{Confidence}(A, B, C \to D) \equiv P(D|A, B, C) = \frac{P(D, A, B, C)}{P(A, B, C)} = \frac{\#\{\text{customers who bought } A, B, C, D\}}{\#\{\text{customers who bought } A, B, C\}}.$$

$$\text{Confidence}(A, B \to C, D) \equiv P(D, C|A, B) = \frac{P(D, A, B, C)}{P(A, B)} = \frac{\#\{\text{customers who bought } A, B, C, D\}}{\#\{\text{customers who bought } A, B\}}.$$

As it can be from here the numerators are the same, so this can be ignored, we just need to focus on the denominators. Thus, from set theory we know that the set of $(A \cap B \cap C) \leq (A \cap B)$ which means that $P(A, B, C) \leq P(A, B)$. This means:

$$\begin{aligned} P(A,B,C) &\leq & P(A,B) \\ \frac{1}{P(A,B,C)} &\geq & \frac{1}{P(A,B)} \\ \frac{P(A,B,C,D)}{P(A,B,C)} &\geq & \frac{P(A,B,C,D)}{P(A,B)} \end{aligned}$$

With this we can prove that confidence$(ABC \rightarrow D) \geq$ confidence$(AB \rightarrow CD)$ Which confirms the apriori principle for making pruning more efficient!

4.10.8: *When the training set is small, the contribution of variance to error may be more than that of bias and in such a case, we may prefer a simple model even though we know that it is too simple for the task. Can you give an example?*

Solution:
Here the model will have higher variance as the model is more complex, so it will be wise to reduce the complexity to be able to reduce the variance. But if the model is too simple it tends into the possibility of having a high bias.

Here are some images showing the problem.

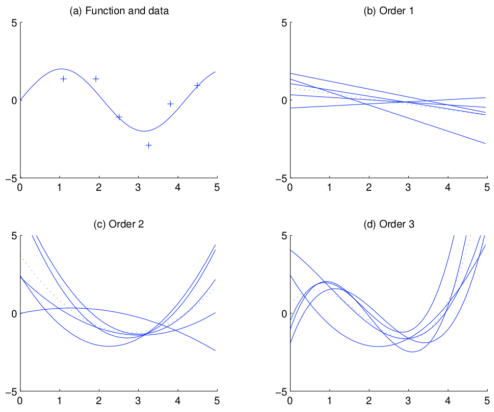
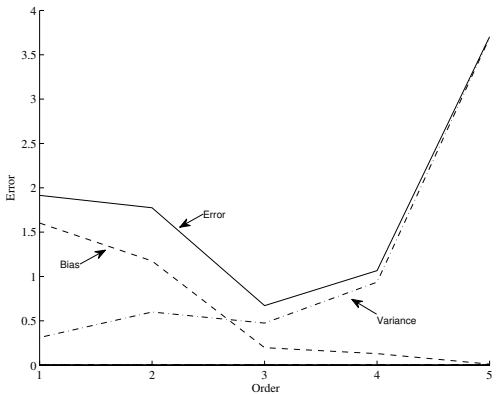


Figure 4.4 Function, one random noisy sample and five fitted polynomials of order 1, 2 and 3.

```
Order 1 B=1.66 V=0.37 E=2.03
Order 2 B=1.31 V=1.59 E=2.90
Order 3 B=3.74 V=2.86 E=6.60
```

In most cases the relation bias, variance as we increase the order of the polynomial will be like the following:



4.10.9: *Let us say, given the samples* $X_i = x_i^t, r_i^t$*, we define* $g_i(x) = r_i^1$*, namely, our estimate for any* x *is the* r *value of the first instance in the (unordered) dataset Xi. What can you say about its bias and variance, as compared with gi(x) = 2 and* $g_i(x) = \sum_t r_i^t/N$*? What if the sample is ordered, so that* $g_i(x) = min_t r_i^t$ *?*

Solution:
Taking an observation within the data points will make the bias go down as compared to one random constant value outside of them. Taking the minimum value will also make the bias will be relatively high. As compared to the mean value which mill make it have less bias.

Here are some images showing the problem, it is important to note that order 0 takes the average value of that group of values:

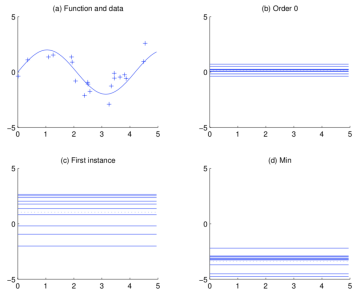


Figure 4.5 Function, one random noisy sample and the fitted models are: $\sum_t r^t/N$, r^1, $\min_t r^t$.

```
Order 0 B=1.87 V=0.10 E=1.97
First inst B=2.63 V=2.30 E=4.93
Min inst B=14.42 V=0.53 E=14.95
```

Week 3 Exercises

5.10.5: *Another possibility using Gaussian densities is to have them all diagonal but allow them to be different. Derive the discriminant for this case.*
OBS. You don't need to derive a formulae! Look at Fig. 5.7. of the book. How would the new version change the appearance of the ellipsoids?

Solution:
When the covariance matrices are diagonal (as in figure 5.7 "Diag. covar"), the distributions are axis-aligned. But they can still vary in how their distributions are. Again, we get a linear discriminant if the two matrices are equal, as their contours will be the same, and otherwise, we get a quadratic discriminants.

5.10.6: *Let us say in two dimensions, we have two classes with exactly the same mean. What type of boundaries can be defined?*
OBS. You don't need to derive a formulae! Present your solution graphically and describe.

Solution:
The means are identical but the matrices are different: Distributions contours are shown by one line for visibility. In (a), when one matrix is just a scaled version of the other, we get a circular discriminant, otherwise we get hyperboles.

6.14.4: *In Sammon mapping, if the mapping is linear, namely, $g(x|W) = W^T x$, how can W that minimizes the Sammon stress be calculated?*
OBS. You don't need to derive a formulae! Give the steps to a solution, i.e. how to you optimize.

Solution:
MDS already does this linear mapping.

Another possibility, if the mapping is non-linear, is to plug in the (non)linear model in Sammon stress (equation 6.37) and find W that minimizes it. Sammon Stress works by finding difference in the distances between two point, in the original space and in the lower dimensional space, then squaring it. finally it is normalized by dividing it with the squared distances from the original space.

One can for example use an iterative, gradient-descent procedure; this is especially interesting when the mapping is nonlinear. We discuss this in chapter 11 when we use multilayer perceptrons (p. 304).

Week 4 Exercises

7.11.11: *Having generated a dendrogram, can we "prune" it?*

Solution:
At any level, if one of the branches has a relatively small number of descendants when compared to others, we may ignore that branch; it will correspond to a cluster containing too few instances.

8.11.10: *Generalize kernel smoother to multivariate data.*

Solution:
We can just use a multivariate kernel, for example, d-variate Gaussian, in equation 8.26. Again, we have the usual concerns of the suitability of using Euclidean distance vs. estimating a local covariance and using Mahalanobis distance.

4.2: *Assume data points $\mathbf{x} \in \mathbb{R}^D$ are given and you know the data is separated into K classes. The goal is to (1) estimate the classes, each represented by one cluster $\mathbf{m}_k \in \mathbb{R}^D$, and (2) to estimate to which of these cluster centers each data point $\mathbf{x}_n$ belongs to, hence minimize:*

$$J = \sum_{n=1}^{N} \sum_{k=1}^{K} r_{nk} \|\mathbf{x}_n - \mathbf{m}_k\|_2^2, \tag{1}$$

where r_{nk} is an indicator variable, which is one if $\mathbf{x}_n$ belongs to $\mathbf{m}_k$, else zero. In the EM-algorithm the estimation of the unknowns $\mathbf{m}_k$ and r_{nk} is alternated, by fixing one of them while estimating the other.

(a): *Assume $\mathbf{m}_k$ are known, estimate the binary values r_{nk}. Keep in mind that for each n one k must be assigned.*
Hint: Consider that this can be done independently for each single data point $\mathbf{x}_n$:

$$J_n(r_{nk}) = \sum_{k=1}^{K} r_{nk} \|\mathbf{x}_n - \mathbf{m}_k\|_2^2. \tag{2}$$

Solution:
Corresponding to the **expectation step** of the EM-algorithm, we have to minimize J with respect to r_{nk} while keeping $\mathbf{m}_k$ fixed. Since J is a linear function of r_{nk} and each term involving n are independent, we can optimize for each n separately by choosing r_{nk} to be 1 for whichever value of k gives the minimum value of $\|\mathbf{x}_n - \mathbf{m}_k\|_2^2$, i.e. we assign the nth data point to the closest cluster. We can express this as

$$r_{nk} = \begin{cases} 1 & \text{if } k = \arg\min_j \|\mathbf{x}_n - \mathbf{m}_j\|_2^2 \\ 0 & \text{otherwise.} \end{cases} \tag{3}$$

(b): *Assume r_{nk} are known, estimate $\mathbf{m}_k$. Compute the derivative of J with respect to $\mathbf{m}_k$, set it to zero, and then deduce the estimates for $\mathbf{m}_k$. By rewriting*

$$J = \sum_{k=1}^{K} J_k(\mathbf{m}_k), \tag{4}$$

we can focus on one cluster k at a time by:

$$J_k(\mathbf{m}_k) = \sum_{n=1}^{N} r_{nk} \|\mathbf{x}_n - \mathbf{m}_k\|_2^2. \tag{5}$$

Hint: $\|\mathbf{x}_n - \mathbf{m}_k\|_2^2 = (\mathbf{x}_n - \mathbf{m}_k)^\intercal (\mathbf{x}_n - \mathbf{m}_k)$.

Solution:
In the **maximization step** we have to minimize J with respect to $\mathbf{m}_k$ while keeping r_{nk} fixed. Since J is a quadratic function of $\mathbf{m}_k$, we can minimize it by finding the partial derivative with respect to $\mathbf{m}_i$, setting it equal to zero, and solving for $\mathbf{m}_i$. The derivative is

$$\frac{\partial}{\partial \mathbf{m}_i} J = 2 \sum_{n=1}^{N} r_{ni}(\mathbf{x}_n - \mathbf{m}_i). \tag{6}$$

Then we set it equal to zero and solve for $\mathbf{m}_i$

$$2 \sum_{n=1}^{N} r_{ni}(\mathbf{x}_n - \mathbf{m}_i) = 0 \tag{7}$$

$$\sum_{n=1}^{N} r_{ni}(\mathbf{x}_n - \mathbf{m}_i) = 0 \tag{8}$$

$$\sum_{n=1}^{N} r_{ni}\mathbf{x}_n = \sum_{n=1}^{N} r_{ni}\mathbf{m}_i \tag{9}$$

$$\frac{\sum_{n=1}^{N} r_{ni}\mathbf{x}_n}{\sum_{n=1}^{N} r_{ni}} = \mathbf{m}_i. \tag{10}$$

The numerator of (10) is the sum of all points assigned to the ith cluster, and the denominator is the number of points assigned to the ith cluster. Hence, (10) can be interpreted as the mean of all points assigned to the ith cluster.

Week 5 Exercises

9.8.4: *In generating a univariate tree, a discrete attribute with n possible values can be represented by n 0/1 dummy variables and then treated as n separate numeric attributes. What are the advantages and disadvantages of this approach?*

Solution:
If one decides to do this, what happens is that the nodes start having binary branches instead of n-ary branches. This does not necessarily mean that the model will become simpler to read, this will depend on the data.

10.11.1: *For each of the following basis functions, describe where it is nonzero:*

Solution:

- $\sin(x_1)$
  This is used when a trigonometric mapping is necessary, for example, in robot arm kinematics, or in recurring phenomena, for example, seasonal repeating behavior in time-series. (zero in multiples of pi, nonzero everywhere else, $2\pi n < x < 2\pi n + \pi, n \in \mathbb{Z}$)
- $\exp\left(-\frac{(x_1-a)^2}{c}\right)$
  This is a bell-shaped function with a as its center and c its spread. Roughly, speaking it is nonzero between $(a - 2\sqrt{c}, a + 2\sqrt{c})$.
- $\exp\left(\frac{\|\mathbf{x}\text{-}\mathbf{a}\|^2}{c}\right)$
  This is a d-dimensional bell-shaped function with a as its center and c its spread in d dimensions.
- $\log(x_2)$
  This is useful when x2 has a wide scale and a log transformation is useful. $x < -1 \vee 1 < x$
- $1(x_1 > c)$
  This is similar to a univariate split of a decision tree node.
- $1(ax_1 + bx_2 > c)$
  This defines a multivariate oblique split.

10.11.2: *For the two-dimensional case of figure 10.2, show equations 10.4 and 10.5.*

Solution:
The main idea here is to do a scalar projection (Figure 1) to get the distance since r represents the distance between the lines. The visual concept of this exercise is given by Figure 2.

since $g(x_0) = 0$, and $r = ||x_0||cos(a_0)$, then we have:

$$\begin{aligned} g(x_0) = \quad & \mathbf{w}^T\mathbf{x}_0 + w_0 = \quad ||\mathbf{w}|| * ||\mathbf{x}||cos(a_0) + w_0 \\ r_0 = \quad & ||\mathbf{x}_0||cos(a_0) \end{aligned}$$

Then using equality $||\mathbf{w}|| * ||\mathbf{x}||cos(a_0) + w_0 = 0$ we have that:

$$\begin{aligned} ||\mathbf{w}||||\mathbf{x}||cos(a_0) + w_0 = \quad & 0 \\ ||\mathbf{x}||cos(a_0) = \quad & \frac{-w_0}{||\mathbf{w}||} \\ r_0 = \quad & \frac{-w_0}{||\mathbf{w}||} \end{aligned}$$

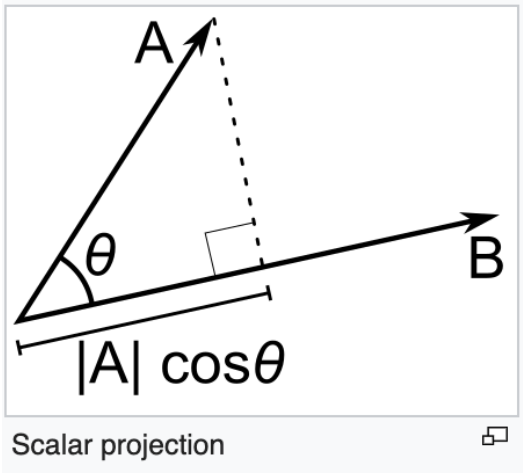


Figure 1: scalar projection

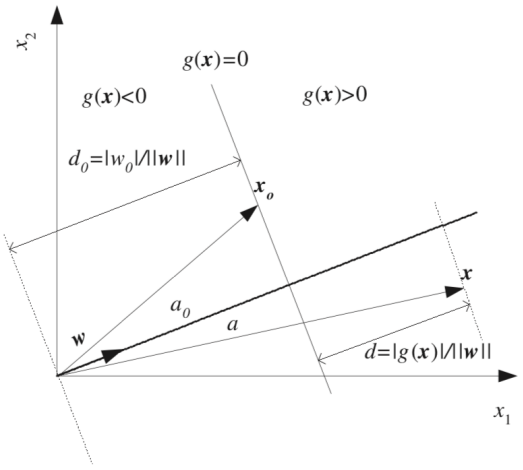


Figure 2: geometric interpretation of linear discriminant

Now for the second case note that $g(x) \neq 0$ (does not equal), still we have:

$$\begin{aligned} g(x) = \quad & \mathbf{w}^T\mathbf{x} + w_0 = \quad ||\mathbf{w}||||\mathbf{x}||cos(a) + w_0 \\ r = \quad & ||\mathbf{x}||cos(a) - r_0 = \quad ||\mathbf{x}||cos(a) - \frac{-w_0}{||\mathbf{w}||} \end{aligned}$$

Then making the substraction:

$$r = \quad ||\mathbf{x}||cos(a) - \frac{-w_0}{||\mathbf{w}||} = \quad \frac{||\mathbf{w}||||\mathbf{x}||cos(a) + w_0}{||\mathbf{w}||} = \quad \frac{g(\mathbf{x})}{||\mathbf{w}||}$$

10.11.3: *Show that the derivative of the softmax,* $y_i = exp(a_i)/\sum_j exp(a_j)$*, is* $\partial y_i/\partial a_j = y_i(\delta_{ij} - y_j)$*, where* δ_{ij} *is 1 if* $i = j$ *and 0 otherwise.*

Solution:

First let's try when i = j, then knowing that the quotient rule is: $\frac{d}{dx}\left[\frac{f(x)}{g(x)}\right] = \frac{f'(x)g(x) - f(x)g'(x)}{g(x)^2}$

then:

$$\frac{\partial y_i}{\partial a_i} = \frac{exp(a_i)\sum_j exp(a_j) - exp(a_i)exp(a_i)}{(\sum_j exp(a_j))^2} = \frac{exp(a_i)}{\sum_j exp(a_j)}\left(\frac{\sum_j exp(a_j) - exp(a_i)}{\sum_j exp(a_j)}\right)$$
$$\frac{\partial y_i}{\partial a_i} = y_i(1 - y_i)$$

Now when $i \neq j$ we have:

$$\frac{\partial y_i}{\partial a_j} = \frac{0\sum_j exp(a_j) - exp(a_i)exp(a_j)}{(\sum_j exp(a_j))^2} = \frac{exp(a_i)}{\sum_j exp(a_j)}\left(\frac{-exp(a_j)}{\sum_j exp(a_j)}\right)$$
$$\frac{\partial y_i}{\partial a_i} = y_i(0 - y_j)$$

10.11.6: *In using quadratic (or higher-order) discriminants as in equation 10.34, how can we keep variance under control?*

Solution:
For a quadratic discriminant, $\mathbf{W}_i$ of equation 10.34 has $O(d^2)$ terms. If we know that certain pairs of input variables x_j , x_k are not correlated, we can have the corresponding off-diagonal $\mathbf{W}_{ijk}$ set to 0. We can have different discriminants share a common $\mathbf{W}$, or, we can write a low-rank approximation to $\mathbf{W}_i$.

Week 6 Exercises

Exercise W6.2 (math and programming): *For $\mathbf{x} \in \mathbb{R}^D$ the PDF for a multivariate (D-dimensional) Gaussian distribution is defined as:*

$$N(\mathbf{x}|\boldsymbol{\mu}, \boldsymbol{\Sigma}) = \frac{1}{\sqrt{(2\pi)^D|\boldsymbol{\Sigma}|}} \exp\left(-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^T\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})\right) \tag{11}$$

where $\boldsymbol{\mu} \in \mathbb{R}^D$ is the mean, $\boldsymbol{\Sigma} \in \mathbb{R}^{D\times D}$ is the covariance matrix, and $|\boldsymbol{\Sigma}|$ is the determinant of the covariance matrix. In the bivariate (2-dimensional) case, we have $\mathbf{x} = (x_1, x_2)^T \in \mathbb{R}^D$, and we can write the parameters as

$$\boldsymbol{\mu} = \begin{pmatrix}\boldsymbol{\mu}_1\\ \boldsymbol{\mu}_2\end{pmatrix} \quad \textit{and} \quad \boldsymbol{\Sigma} = \begin{bmatrix}\sigma_1^2 & \rho\sigma_1\sigma_2\\ \rho\sigma_1\sigma_2 & \sigma_2^2\end{bmatrix} \tag{12}$$

where $\boldsymbol{\Sigma}_1, \boldsymbol{\Sigma}_2 \in \mathbb{R}^+$ and $\rho \in]-1, 1[$.

(a): *Show that in the 2D case we have that $\frac{1}{\sqrt{(2\pi)^D|\boldsymbol{\Sigma}|}} = \frac{1}{2\pi\sigma_1\sigma_2\sqrt{1-\rho^2}}$.*

Solution:
The determinant of a 2x2 matrix $A = \begin{bmatrix}a & b\\ c & d\end{bmatrix}$ is given by $|A| = ad - bc$ so:

$$\begin{aligned}
|\boldsymbol{\Sigma}| &= \sigma_1^2\sigma_2^2 - ((\rho\sigma_1\sigma_2)(\rho\sigma_1\sigma_2)) && (13)\\
&= \sigma_1^2\sigma_2^2 - \rho^2\sigma_1^2\sigma_2^2 && (14)\\
&= \sigma_1^2\sigma_2^2(1-\rho^2) && (15)\\
& && (16)
\end{aligned}$$

Now one can insert the new term for the determinant of $\boldsymbol{\Sigma}$.

$$\begin{aligned}
\frac{1}{\sqrt{(2\pi)^D|\boldsymbol{\Sigma}|}} &= \frac{1}{2\pi\sqrt{|\boldsymbol{\Sigma}|}} && (17)\\
&= \frac{1}{2\pi\sqrt{\sigma_1^2\sigma_2^2(1-\rho^2)}} && (18)\\
&= \frac{1}{2\pi\sigma_1\sigma_2\sqrt{1-\rho^2}} && (19)\\
& && (20)
\end{aligned}$$

(b): *For the 2D case, implement a Python function that take the following arguments $(x_1, x_2, \boldsymbol{\mu}_1, \boldsymbol{\mu}_2, \sigma_1, \sigma_2, \rho)$ and returns the value of $N(\mathbf{x}|\boldsymbol{\mu}, \boldsymbol{\Sigma})$.*

Solution:
See notebook.

(c): *Make six individual plots of the contours for the PDF where $x \in [-3, 3]^2$.*

- $(\boldsymbol{\mu}_1, \boldsymbol{\mu}_2, \boldsymbol{\Sigma}_1, \boldsymbol{\Sigma}_2, \rho) = (0, 0, 1, 1, 0)$,
- $(\boldsymbol{\mu}_1, \boldsymbol{\mu}_2, \boldsymbol{\Sigma}_1, \boldsymbol{\Sigma}_2, \rho) = (1, 1, 1, 1, 0)$,
- $(\boldsymbol{\mu}_1, \boldsymbol{\mu}_2, \boldsymbol{\Sigma}_1, \boldsymbol{\Sigma}_2, \rho) = (0, 0, 1, 2, 0)$,
- $(\boldsymbol{\mu}_1, \boldsymbol{\mu}_2, \boldsymbol{\Sigma}_1, \boldsymbol{\Sigma}_2, \rho) = (0, 0, 2, 1, 0)$,
- $(\boldsymbol{\mu}_1, \boldsymbol{\mu}_2, \boldsymbol{\Sigma}_1, \boldsymbol{\Sigma}_2, \rho) = (0, 0, 1, 1, 0.5)$ *and*
- $(\boldsymbol{\mu}_1, \boldsymbol{\mu}_2, \boldsymbol{\Sigma}_1, \boldsymbol{\Sigma}_2, \rho) = (0, 0, 1, 1, -0.75)$.

Solution:
See notebook.

(d): *Use the plots to explain the meaning of the parameters $\mu_1, \mu_2, \sigma_1, \sigma_2$ and ρ.*

Solution:
Just as before, μ_1, μ_2 correspond to each of the variables' expected values, σ_1^2, σ_2^2 correspond to how spread out the distribution is in that dimension, and ρ can be thought of as the two variables dependency, i.e. how correlated they are (positively or negatively).

Exercise W6.3 (math): *The logical operator exclusive or is defined as $p \oplus q = (p \vee q) \wedge \neg(p \vee q)$. The expression $p \oplus q$ is true exactly when one of the variables p or q are true. The corresponding truth table is:*

p	q	$p \oplus q$
0	0	0
1	0	1
0	1	1
1	1	1

Consider a two layer neural network with two dimensional input $\mathbf{x} \in \mathbb{R}^2$, two hidden nodes, and one dimensional output $y \in \mathbb{R}$, defined by

$$y(x, w) = \sum_{j=1}^{2} w_j^{(2)} h\left(\sum_{i=1}^{2} w_{ji}^{(1)} x_i + w_{j0}^{(1)}\right) + w_0^{(2)} \tag{21}$$

where

- $w_{10}^{(1)} = 0$ *and* $w_{20}^{(1)} = -1$,
- $w_{ij}^{(1)} = 1$ *for* $i, j \in \{1, 2\}$,
- $w_0^{(2)} = 0$,$w_1^{(2)} = 1$ *and* $w_2^{(2)} = -2$.

and $h(a) = max(0, a)$ is the rectifier activation function. Very by hand calculations that this network is equivalent to the exclusive or operator, i.e. that $y((0,0)^\intercal, \mathbf{w}) = 0, y((1,0)^\intercal, \mathbf{w}) = 1, y((0,1)^\intercal, \mathbf{w}) = 1$ and $y((1,1)^\intercal, \mathbf{w}) = 0$.

(a): *Do the verification by expanding the sum in equation (W4.4) and inserting the different values of $\mathbf{x}$.*

Solution:
Firstly, let's write out the sum in its entirety as it's not that long, and it'll make the calculations easier,

$$\begin{aligned} y(\mathbf{x}, \mathbf{w}) &= \sum_{j=1}^{2} w_j^{(2)} h\left(w_{j1}^{(1)} x_1 + w_{j2}^{(1)} x_2 + w_{j0}^{(1)}\right) + w_0^{(2)} \\ &= w_1^{(2)} h\left(w_{11}^{(1)} x_1 + w_{12}^{(1)} x_2 + w_{10}^{(1)}\right) + w_2^{(2)} h\left(w_{21}^{(1)} x_1 + w_{22}^{(1)} x_2 + w_{20}^{(1)}\right) + w_0^{(2)}. \end{aligned}$$

Putting the values of the weights into the equation yields

$$y(\mathbf{x}, \mathbf{w}) = 1h\left(1x_1 + 1x_2 + 0\right) - 2h\left(1x_1 + 1x_2 + -1\right) + 0 = h(x_1 + x_2) - 2h(x_1 + x_2 - 1),$$

show that $y((0,0), w) = 0$.

$$\begin{aligned} y((0,0)) &= 1 * h(1 * 0 + 1 * 0) - 2h(1 * 0 + 1 * 0 - 1) \\ &= 1 * max(0, 0) - 2max(0, -1) \\ &= 0 \end{aligned}$$

so $y((0,0),w) = 0$

show that $y((1,0),w) = 1$.

$$\begin{aligned} y((1,0)) &= 1 * h(1*1 + 1*0) - 2h(1*1 + 1*0 - 1) \\ &= 1 * max(0,1) - 2max(0,0) \\ &= 1 \end{aligned}$$

so $y((1,0),w) = 1$

show that $y((0,1),w) = 1$.

$$\begin{aligned} y((0,1)) &= 1 * h(1*0 + 1*1) - 2h(1*0 + 1*1 - 1) \\ &= 1 * max(0,1) - 2max(0,-1) \\ &= 1 \end{aligned}$$

so $y((0,1),w) = 1$.

show that $y((1,1),w) = 0$.

$$\begin{aligned} y((1,1)) &= 1 * h(1*1 + 1*1) - 2h(1*1 + 1*1 - 1) \\ &= 1 * max(0,2) - 2max(0,1) \\ &= 0 \end{aligned}$$

so $y((1,1),w) = 0$.

(b): *Do the verification by drawing the network, inserting the different values of* $\mathbf{x}$ *and performing a forward propagation (i.e. calculating the values of all the nodes).*

Solution:
See Figure 3, Figure 4, Figure 5 and Figure 6

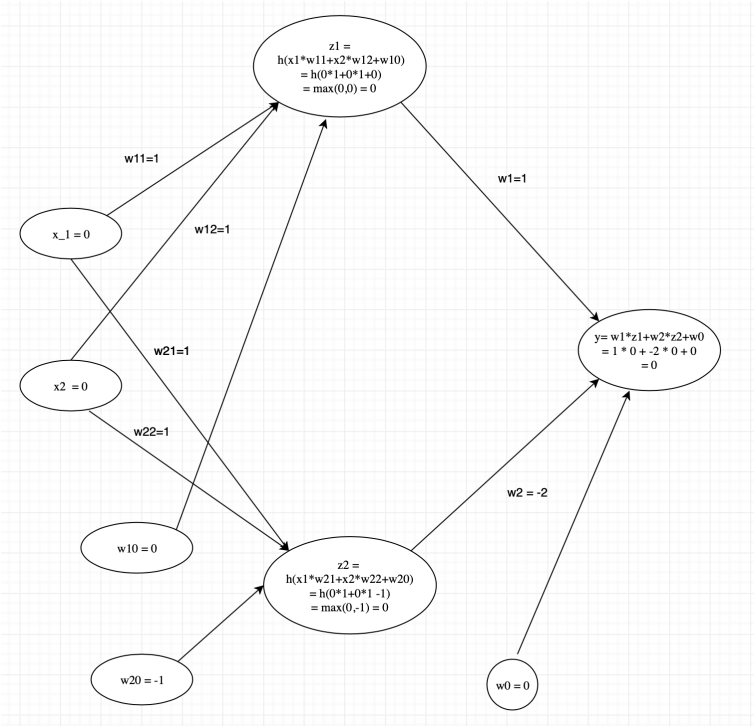


Figure 3: y(0,0)

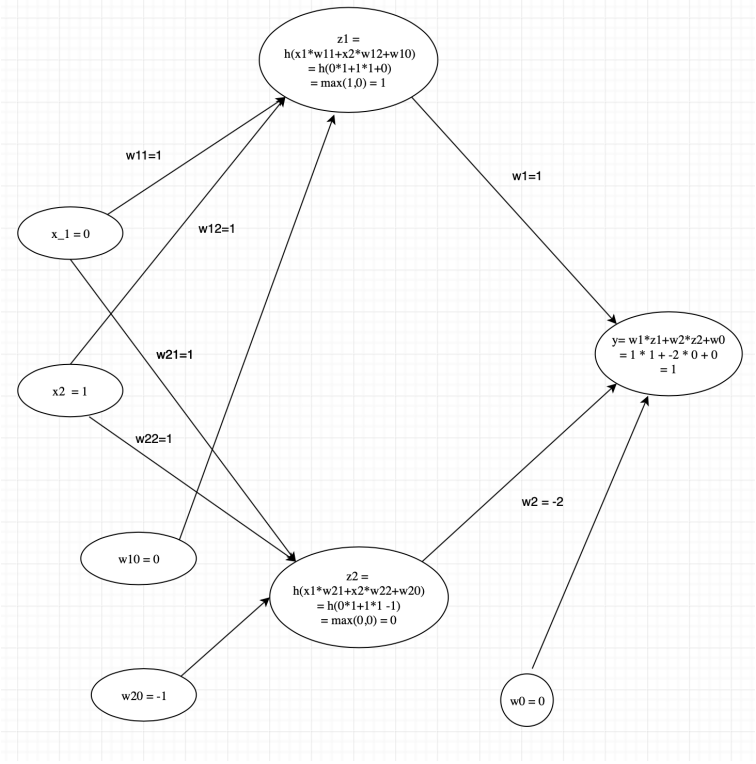


Figure 4: y(0,1)

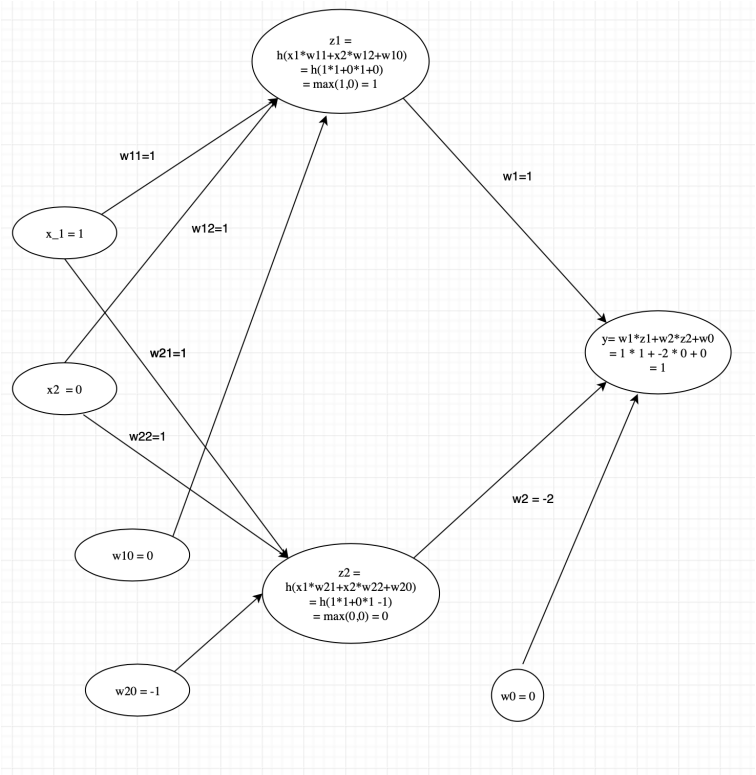


Figure 5: y(1,0)

Exercise W6.5 (math): *Do exercise 12.11.3 in Alpaydin. Note that the exercise text is misleading: the update rules should be derived in the classification case where (12.19) and (12.20) are assumed.*

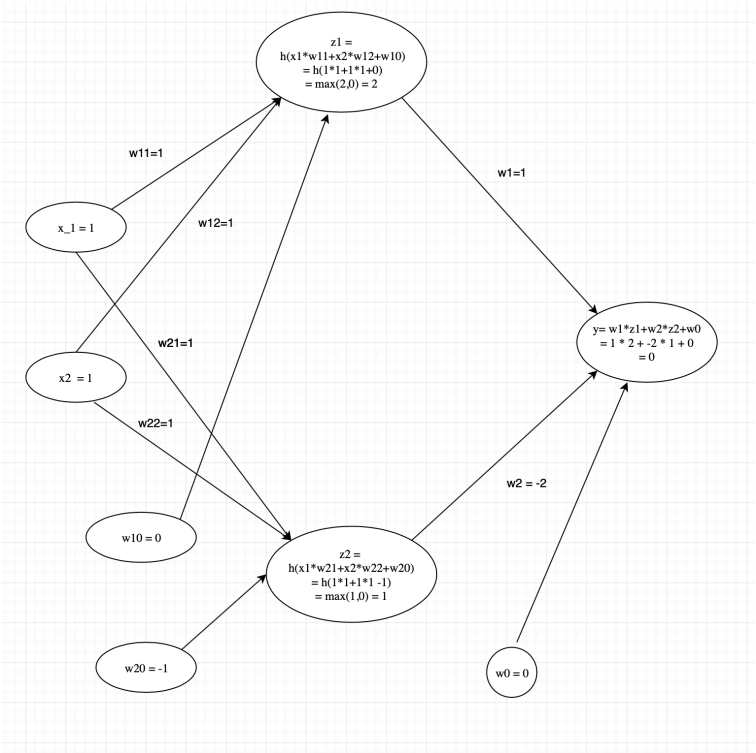


Figure 6: y(1,1)

Derive the update equations for the RBF network for classification(equations 12.20 and 12.21) where (12.19) and (12.20) are assumed.

Solution:
Equation 12.20 is defined as:

$$y_i^t = \frac{exp\left[\sum_h w_{ih} p_h^t + w_{i0}\right]}{\sum_k exp\left[\sum_h w_{kh} p_h^t + w_{k0}\right]} \tag{22}$$

where $\sum_h w_{ih} p_h^t + w_{i0} = z_i^t$.
Equation 12.21 is defined as:

$$E(\{\mathbf{m}_h, s_h, w_{ih}\}_{i,h} | \mathcal{X}) = -\sum_t \sum_i r_i^t \log(y_i^t) \tag{23}$$

Then the update equation is given by:

$$\Delta w_{ih} = -\eta \frac{dE}{dw_{ih}} \tag{24}$$

where $\frac{dE}{dw_{ih}}$ can be solved using the chainrule:

$$\frac{dE}{dw_{ih}} = \frac{dE}{dy_i^t} \frac{dy_i^t}{dz_i^t} \frac{dz_i^t}{dw_{ih}} \tag{25}$$

So now we can find the derivatives.

$$\frac{dz_i^t}{dw_{ih}} = \frac{d}{dw_{ih}} \sum_h w_{ih} p_h^t + w_{i0} \tag{26}$$

$$= p_h^t \tag{27}$$

$$\frac{dy_i^t}{dz_i^t} = \frac{exp\left[\sum_h w_{ih} p_h^t + w_{i0}\right]}{\sum_k exp\left[\sum_h w_{kh} p_h^t + w_{k0}\right]} \tag{28}$$

$$= y_i^t(\delta_{ij} - y_j^t) \tag{29}$$

where δ_{ij} is Kronecker delta, which is 1 if $i = j$ and 0 is $i \neq j$.

$$\frac{dE}{dy_i^t} = -\sum_t \sum_i r_i^t log(y_i^t) \tag{30}$$

$$= -\sum_t \sum_i \frac{r_i^t}{y_i^t} \tag{31}$$

where $\sum_i r_i^t = 1$ as it is the identification vector.

$$\frac{dE}{dw_{ih}} = \frac{dE}{dy_i^t} \frac{dy_i^t}{dz_i^t} \frac{dz_i^t}{dw_{ih}} \tag{32}$$

$$= -\sum_t \sum_i \frac{r_i^t}{y_i^t} y_i^t(\delta_{ij} - y_i^t) p_h^t \tag{33}$$

$$= -\sum_t \sum_i r_i^t(\delta_{ij} - y_i^t) p_h^t \tag{34}$$

$$= -\sum_t (\sum_i r_i^t \delta_{ij} - y_i^t \sum_i r_i^t) p_h^t \tag{35}$$

$$= -\sum_t (r_i^t - y_i^t \sum_i r_i^t) p_h^t \tag{36}$$

$$= -\sum_t (r_i^t - y_i^t) p_h^t \tag{37}$$

Now we have the $\frac{dE}{dw_{ih}}$ we can write the final update function:

$$\Delta w_{ih} = -\eta \frac{dE}{dw_{ih}} \tag{38}$$

$$= \eta \sum_t (r_i^t - y_i^t) p_h^t \tag{39}$$

Similarly, for Δw_{i0}:

$$\Delta w_{i0} = -\eta \frac{dE}{dw_{ih}} \tag{40}$$

$$= \eta \sum_t (r_i^t - y_i^t) \tag{41}$$

because

$$\frac{dz_i^t}{dw_{i0}} = \frac{d}{dw_{i0}} \sum_h w_{ih} p_h^t + w_{i0} \tag{42}$$

$$= 1 \tag{43}$$

Week 7 Exercises

13.16.10: *Let us say we have two representations for the same object and associated with each, we have a different kernel. How can we use both to implement a joint dimensionality reduction using kernel PCA?*

Solution 1:
We could do a separate kernel PCA to get two lower-dimensional vectors and then concatenate them.

Solution 2:
By taking a sum of the kernel functions, we could perform PCA on the sum.

14.10.1: *With two independent inputs in a classification problem, that is, $p(x_1, x_2|C) = p(x_1|C)p(x_2|C)$, how can we calculate $p(x_1|x_2, C)$? Derive the formula for $p(x_j|C_i) \sim \mathcal{N}(\mu_{ij}, \sigma^2_{ij})$.*

Solution:
Since x_1 and x_2 are independent, then $p(x_1|x_2, C_i) = p(x_1|C_i) \sim \mathcal{N}(\mu_{i1}, \sigma^2_{i1})$ (note that this is a univariate Gaussian distribution). However, if x_1 and x_2 were dependent, then

$$p(x_1|x_2, C_i) = \frac{p(x_1, x_2|C_i)}{p(x_2|C_i)}$$

(note that this is a bivariate Gaussian distribution).

14.10.7: *Write down the graphical model for linear logistic regression for two classes in the manner of figure 14.7.*

Solution:

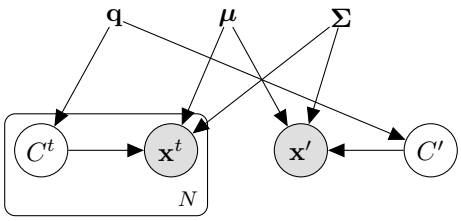


Figure 7: Graphical representation of logistic regression.

Theoretical exercise 7.2. SVM: *The XOR problem consists of four points from two classes, which are not linearly separable, as follows:*

- *class 1:* $\mathbf{x}_1$, $\mathbf{x}_2$,
- *class 2:* $\mathbf{x}_3$, $\mathbf{x}_4$,

given the four points:

$$\mathbf{x}_1 = \begin{pmatrix} 1 \\ 1 \end{pmatrix}, \qquad \mathbf{x}_2 = \begin{pmatrix} -1 \\ -1 \end{pmatrix}, \qquad \mathbf{x}_3 = \begin{pmatrix} -1 \\ 1 \end{pmatrix}, \qquad \mathbf{x}_4 = \begin{pmatrix} 1 \\ -1 \end{pmatrix},$$

with labels:

$$r_1 = +1, \qquad r_2 = +1, \qquad r_3 = -1, \qquad r_4 = -1.$$

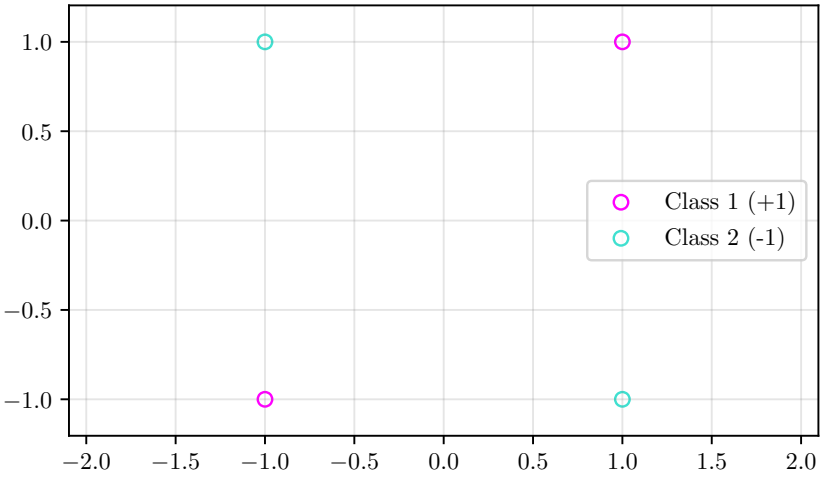
The goal of this exercise is to compute the discriminant:

$$g(\mathbf{x}) = \mathbf{w}^\intercal \boldsymbol{\phi}(\mathbf{x}),$$

which enables a linear classification in a higher dimension, enabled by the basis function $\boldsymbol{\phi}$.

(a): *Draw the points and highlight to which class which points belongs.*

Solution:



(b): *Since the points are not linearly separable in 2D, they should be transferred to a higher dimension, such that they become. Use the following basis function to transfer each of the four 2D points to 6D:*

$$\begin{aligned}\boldsymbol{\phi} &: \mathbb{R}^2 \to \mathbb{R}^6 \\ \boldsymbol{\phi}(\mathbf{x}) = \boldsymbol{\phi}(x_1, x_2) &= \left(1, \sqrt{2}x_1, \sqrt{2}x_2, \sqrt{2}x_1x_2, x_1^2, x_2^2\right)^\intercal,\end{aligned}$$

i.e. calculate $\mathbf{z}_i = \boldsymbol{\phi}(\mathbf{x}_i), i = 1, \ldots, 4$.

Solution:

$$\begin{aligned}\mathbf{z}_1 = \boldsymbol{\phi}(\mathbf{x}_1) &= \left(1, \sqrt{2}, \sqrt{2}, \sqrt{2}, 1, 1\right)^\intercal \\ \mathbf{z}_2 = \boldsymbol{\phi}(\mathbf{x}_2) &= \left(1, -\sqrt{2}, -\sqrt{2}, \sqrt{2}, 1, 1\right)^\intercal \\ \mathbf{z}_3 = \boldsymbol{\phi}(\mathbf{x}_3) &= \left(1, -\sqrt{2}, \sqrt{2}, -\sqrt{2}, 1, 1\right)^\intercal \\ \mathbf{z}_4 = \boldsymbol{\phi}(\mathbf{x}_4) &= \left(1, \sqrt{2}, -\sqrt{2}, -\sqrt{2}, 1, 1\right)^\intercal\end{aligned}$$

(c): *Use the known values to complete Eq. (13.26):*

$$\begin{aligned}L_d(\boldsymbol{\alpha}) = L_d(\alpha_1, \alpha_2, \alpha_3, \alpha_4) &= \sum_{i=1}^{4} \alpha_i - \frac{1}{2}\sum_{i=1}^{4}\sum_{j=1}^{4} \alpha_i\alpha_j r_i r_j \mathbf{z}_i^\intercal \mathbf{z}_j \\ &= \sum_{i=1}^{4} \alpha_i - \frac{1}{2}\sum_{i=1}^{4}\sum_{j=1}^{4} \alpha_i\alpha_j r_i r_j \boldsymbol{\phi}(\mathbf{x}_i)^\intercal \boldsymbol{\phi}(\mathbf{x}_j)\end{aligned}$$

Solution:
We start by noting that

$$\mathbf{z}_i^\intercal \mathbf{z}_j = 8\delta_{ij} + 1 = \begin{cases} 9 & \text{if } i = j, \\ 1 & \text{otherwise.} \end{cases}$$

We can now evaluate L_d:

$$\begin{aligned}
L_d(\boldsymbol{\alpha}) &= \sum_{i=1}^{4} \alpha_i - \frac{1}{2} \sum_{i=1}^{4} \sum_{j=1}^{4} \alpha_i \alpha_j r_i r_j \mathbf{z}_i^\intercal \mathbf{z}_j \\
&= \sum_{i=1}^{4} \alpha_i - \frac{1}{2} \left(9 \sum_{i=1}^{4} \alpha_i^2 + \sum_{i=1}^{4} \sum_{\substack{j=1 \\ j \neq i}}^{4} r_i r_j \alpha_i \alpha_j \right) \\
&= \sum_{i=1}^{4} \alpha_i - \frac{9}{2} \sum_{i=1}^{4} \alpha_i^2 - (\alpha_1\alpha_2 - \alpha_1\alpha_3 - \alpha_1\alpha_4 - \alpha_2\alpha_3 - \alpha_2\alpha_4 + \alpha_3\alpha_4) \\
&= \mathbf{1}^\intercal \boldsymbol{\alpha} - \frac{9}{2} \sum_{i=1}^{4} \alpha_i^2 - \alpha_1\alpha_2 + \alpha_1\alpha_3 + \alpha_1\alpha_4 + \alpha_2\alpha_3 + \alpha_2\alpha_4 - \alpha_3\alpha_4.
\end{aligned}$$

(d): *Compute the derivative of $L_d(\boldsymbol{\alpha})$ with respect to α_i, i.e. the four components of the gradient:*

$$\nabla L_d(\boldsymbol{\alpha}) = \begin{pmatrix} \frac{\partial}{\partial \alpha_1} L_d(\alpha_1, \alpha_2, \alpha_3, \alpha_4) \\ \frac{\partial}{\partial \alpha_2} L_d(\alpha_1, \alpha_2, \alpha_3, \alpha_4) \\ \frac{\partial}{\partial \alpha_3} L_d(\alpha_1, \alpha_2, \alpha_3, \alpha_4) \\ \frac{\partial}{\partial \alpha_4} L_d(\alpha_1, \alpha_2, \alpha_3, \alpha_4) \end{pmatrix}.$$

Solution:
The gradient is

$$\begin{aligned}
\nabla L_d(\boldsymbol{\alpha}) &= \begin{pmatrix} \frac{\partial}{\partial \alpha_1} L_d(\alpha_1, \alpha_2, \alpha_3, \alpha_4) \\ \frac{\partial}{\partial \alpha_2} L_d(\alpha_1, \alpha_2, \alpha_3, \alpha_4) \\ \frac{\partial}{\partial \alpha_3} L_d(\alpha_1, \alpha_2, \alpha_3, \alpha_4) \\ \frac{\partial}{\partial \alpha_4} L_d(\alpha_1, \alpha_2, \alpha_3, \alpha_4) \end{pmatrix} \\
&= \begin{pmatrix} 1 - 9\alpha_1 - \alpha_2 + \alpha_3 + \alpha_4 \\ 1 - 9\alpha_2 - \alpha_1 + \alpha_3 + \alpha_4 \\ 1 - 9\alpha_3 + \alpha_1 + \alpha_2 - \alpha_4 \\ 1 - 9\alpha_4 + \alpha_1 + \alpha_2 - \alpha_3 \end{pmatrix}.
\end{aligned}$$

(e): *Derive the equation system from $\nabla L_d(\boldsymbol{\alpha}) = 0$ and solve for $\boldsymbol{\alpha} = (\alpha_1, \alpha_2, \alpha_3, \alpha_4)^\intercal$.*

Solution:
We first rearrange the equation system to the form $\mathbf{A}\boldsymbol{\alpha} = \mathbf{b}$:

$$\begin{aligned}
L_d(\boldsymbol{\alpha}) &= 0 \\
\begin{pmatrix} 1 - 9\alpha_1 - \alpha_2 + \alpha_3 + \alpha_4 \\ 1 - 9\alpha_2 - \alpha_1 + \alpha_3 + \alpha_4 \\ 1 - 9\alpha_3 + \alpha_1 + \alpha_2 - \alpha_4 \\ 1 - 9\alpha_4 + \alpha_1 + \alpha_2 - \alpha_3 \end{pmatrix} &= 0 \\
\begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \end{pmatrix} + \begin{pmatrix} -9\alpha_1 - \alpha_2 + \alpha_3 + \alpha_4 \\ -\alpha_1 - 9\alpha_2 + \alpha_3 + \alpha_4 \\ \alpha_1 + \alpha_2 - 9\alpha_3 - \alpha_4 \\ \alpha_1 + \alpha_2 - \alpha_3 - 9\alpha_4 \end{pmatrix} &= 0 \\
\begin{pmatrix} 9 & 1 & -1 & -1 \\ 1 & 9 & -1 & -1 \\ -1 & -1 & 9 & 1 \\ -1 & -1 & 1 & 9 \end{pmatrix} \boldsymbol{\alpha} &= \mathbf{1}
\end{aligned}$$

Solving this for $\boldsymbol{\alpha}$ gives us

$$\boldsymbol{\alpha} = \frac{1}{8}\mathbf{1}^\intercal = \left(\frac{1}{8}, \frac{1}{8}, \frac{1}{8}, \frac{1}{8}\right)^\intercal.$$

(f): *Which of the four training points are support vectors? How do the values of α_i answer this question?*

Solution:
The set of points, whose corresponding α are larger than zero, are the support vectors (Alpaydin pp. 353). This means all our points are support vectors.

(g): *Now that all four values of $\boldsymbol{\alpha}$ have been computed, employ Eq. (13.24) to compute $\mathbf{w}$:*

$$\mathbf{w} = \sum_{i=1}^{4} \alpha_i r_i \mathbf{z}_i = \sum_{i=1}^{4} \alpha_i r_i \boldsymbol{\phi}(\mathbf{x}_i).$$

Please note: $\mathbf{w}, \mathbf{z}_i \in \mathbb{R}^6$.

Solution:

$$\begin{aligned}
\mathbf{w} &= \sum_{i=1}^{4} \alpha_i r_i \boldsymbol{\phi}(\mathbf{x}_i) \\
&= \frac{1}{8}\begin{pmatrix} 1 \\ \sqrt{2} \\ \sqrt{2} \\ \sqrt{2} \\ 1 \\ 1 \end{pmatrix} + \frac{1}{8}\begin{pmatrix} 1 \\ -\sqrt{2} \\ -\sqrt{2} \\ \sqrt{2} \\ 1 \\ 1 \end{pmatrix} - \frac{1}{8}\begin{pmatrix} 1 \\ -\sqrt{2} \\ \sqrt{2} \\ -\sqrt{2} \\ 1 \\ 1 \end{pmatrix} - \frac{1}{8}\begin{pmatrix} 1 \\ \sqrt{2} \\ -\sqrt{2} \\ -\sqrt{2} \\ 1 \\ 1 \end{pmatrix} \\
&= \left(0, 0, 0, \frac{\sqrt{2}}{2}, 0, 0\right)^\intercal.
\end{aligned}$$

(h): *Give the discriminant function g based on the original input space:*

$$g(\mathbf{x}) = g(x_1, x_2) = \mathbf{w}^\intercal \boldsymbol{\phi}(\mathbf{x}) = \ldots$$

Solution:

$$\begin{aligned}
g(\mathbf{x}) &= \mathbf{w}^\intercal \boldsymbol{\phi}(\mathbf{x}) \\
&= \left(0, 0, 0, \frac{\sqrt{2}}{2}, 0, 0\right)\left(1, \sqrt{2}x_1, \sqrt{2}x_2, \sqrt{2}x_1x_2, x_1^2, x_2^2\right)^\intercal \\
&= \frac{\sqrt{2}}{2}\sqrt{2}x_1x_2 \\
&= x_1x_2
\end{aligned}$$

(i): *Apply the discriminant function and compute the resulting values for the training input samples $g(\mathbf{x}_i)$, $i = 1, \ldots, 4$. How are they classified? Are they correctly classified?*

Solution:
Yes, they are correctly separated and classified:

$$\begin{aligned}
g(\mathbf{x}_1) = g(\mathbf{x}_2) = 1 > 0 &\Rightarrow \text{class } 1 \\
g(\mathbf{x}_3) = g(\mathbf{x}_4) = -1 \leq 0 &\Rightarrow \text{class } 2.
\end{aligned}$$

Week 8 Exercises

17.13.2: *In bagging, to generate the L training sets, what would be the effect of using L-fold cross-validation instead of bootstrap?*

Solution:
With L-fold cross-validation[1], each training set has L - 1 parts of the original set and different sets share L - 2 parts. For example, for L = 10, each training set contains 90 percent of the dataset whereas this percentage is expected to be 63.2 percent[2] for bootstrap when we let a third of the dataset to be validation. Different training sets share 80 percent of the data which is higher than the expected percentage of data shared by two bootstrap samples.

17.13.4: *In the mixture of experts architecture, we can have different experts use different input representations. How can we design the gating network in such a case?*

Solution:
As an additional definition of gating we have: Gating is a generalization of Cross-Validation Selection. It involves training another learning model to decide which of the models in the bucket is best-suited to solve the problem. Often, a perceptron is used for the gating model. It can be used to pick the "best" model, or it can be used to give a linear weight to the predictions from each model in the bucket.

Then when doing this we might give all the different inputs used for training the models to the gating model, but this might be too big, so in such cases some dimensionality reduction might be worth doing or use a multilayered perceptron might be better.

18.9.1: *Given the grid world in figure 18.12, if the reward on reaching on the goal is 100 and $\gamma = 0.9$, calculate manually $Q(s, a), V(S)$, and the actions of optimal policy.*

Solution:
If we think of our policy just caring about one step forward, then our V(S) is only the reward gotten by the next move. Now Q(S,a) will be represented by 4 4x4 matrices where the 4x4 matrices represent the grid and we need 4 of them because there are four possible movements (up, left, right, up). This matrix will be initialized by zero, and there will only be a reward of 100 if the agent reaches the goal position (3,2)

[1] This is presented in the book in page 559 chapter 19.

[2] This percentage is presented on page 561. But this is a probability, so it is the expected value, is not deterministic like in k-fold (converges to infinity).

Week 9 Exercises

Theoretical Exercises 9.2.:
What are the fundamental differences between the Maximum-Likelihood and Bayesian approach of estimation?

Solution:

Short answer

ML finds the weights that either maximizes the log likelihood, or equivalently minimizes the sum of squared errors. In the Bayesian approach one finds weights that maximizes the posterior distribution w.r.t weights which gives the MAP (maximum-a-posteriori) of w (but only in some specific cases, where both the prior and the likelihood is given by a Gaussian dist.). In order to do find the MAP estimate one need a prior probability distribution for the weights and therefore are the weights not treated as a point estimate but as a random variable.

More details

A linear regression model is defined as following:

$$r = \mathbf{w}^T\mathbf{x} + \epsilon, \text{ where } \epsilon \sim N(0, 1/\beta) \tag{44}$$

where β is the precision of the additive noise and w are the weights. From this equation we have:

$$p(r_t|\mathbf{x}_t, w, \beta) \sim N(\mathbf{w}^T\mathbf{x}, 1/\beta) \tag{45}$$

where the log likelihood is:

$$L(w|X) = \log p(X|w) = \log p(\mathbf{r}, \mathbf{X}|\mathbf{w}) \tag{46}$$
$$= \log p(\mathbf{r}|\mathbf{X}, \mathbf{w}, \beta) + \log p(X) \tag{47}$$

where $\log p(X)$ is constant and independent of the parameters and is therefore ignored and only the first term is expanded:

$$\log p(\mathbf{r}|\mathbf{X}, \mathbf{w}, \beta) = \log \prod_t p(r_t x_t, w,) \tag{48}$$
$$= -N \log \sqrt{2\pi} + N \log \sqrt{\beta} - \frac{\beta}{2} \sum_t (r_t - \mathbf{w}^T x_t)^2 \tag{49}$$

So the case of ML one can either find w that maximizes the log likelihood, $\log p(r|X, w, \beta)$, or equivalently minimizes the sum of squared error (see page 457 for more info).

The solution for W_{ML} is therefore $w_{ML} = (X^T X)^{-1} X^T r$, which is a point estimate of the parameters wich maximizes the likelihood.

In the Bayesian approach to linear regression we wish to find the parameters Θ which maximized the posterior in in Bayes theorem:

$$p(\Theta|\mathbf{x}) = \frac{p(\mathbf{x}|\Theta)p(\Theta)}{p(\mathbf{x})} \tag{50}$$

$$posterior = \frac{likelihood \times prior}{evidence} \tag{51}$$

In order to calculate or approcimate the posterior distribution we need some prior probability distribution for the parameters. For reasons of simplicity, we will use an isotropic Gaussian distribution over parameters w with zero mean: $p(w) \sim N(0, (1/\alpha)I)$, where α is the precision of the prior. The prior is conjugate to the likelihood $p(t|x, w, \beta)$ meaning that the posterior distribution has the same functional form as the prior i.e. is also a Gaussian. In this special case, the posterior has an analytical solution with the following sufficient statistics

$$p(w|X, r) \sim N(\mu_N, \Sigma_N) \tag{52}$$

where

$$\Sigma_N = (\alpha I + \beta X^T X)^{-1} \tag{53}$$

$$\mu_N = \beta \Sigma_N X^T r \tag{54}$$

A major advantages of the Bayesian approach is that we can calculate a measure of uncertainty.

$$Var(\hat{r}) = \frac{1}{\beta} + (\hat{x})^T \Sigma_N \hat{x} \tag{55}$$