Task Sheet 3, Solution

Mathias Weyland (mathias.weyland@uzh.ch)

A. Lagrangian

1. When computing the partial derivative

\[ \frac{\partial L}{\partial \beta_i} = h_i(x) \]

we see that every term of the Lagrangian disappears except from the one related to \( h_i(x) \) in which \( \beta_i \) disappears. Furthermore, setting \( \frac{\partial L}{\partial \beta_i} = 0 \) leads to

\[ h_i(x) = 0 \]

which is the original constraint.

2. If any of the constraints are violated, we end up with

\[ \theta_P = \max_{\alpha, \beta, \alpha_i \geq 0} L(x, \alpha, \beta) = \infty \]

The reason for this is that the \( \alpha_i \) in \( \alpha_i g_i(x) \) and the \( \beta_i \) in \( \beta_i h_i(x) \), respectively, can be chosen arbitrarily; the terms do not have an upper bound. If on the other hand the constraints hold true, the sum related to the equality constraint is 0 always and the maximum of each \( \alpha_i g_i(x) \) is found at \( \alpha_i = 0 \). The optimisation problem therefore simplifies to

\[ \min_x \theta_P(x) = \min_x f(x) \]

which is the original problem. Minimising \( \theta_P(x) \) will “pick” from the solutions for which the constraints are satisfied because any other solution is \( \infty \) and hence not a decent minimum.

3. In the context of SVMs, reasons for dealing with the dual problems are:
   
   - We can identify support vectors by looking at the \( \alpha_i \) (see question B.8).
• Stating the problem in terms of inner products allows us to use the kernel trick (see question B.9).

• The computation-effort is reduced due to lower dimensionality, although we did not really look at that.

(On a side note, \( p^* = d^* \) does not hold true in general, but it does so for the SVM problem.)

B. Support Vector Machines (SVMs)

1. The separating hyperplane is denoted by \( w^T \xi^{(i)} + b = 0 \). Any other \( s = w^T \xi^{(i)} + b \neq 0 \) is on either side of the hyperplane, depending on the sign of \( s \). Naturally, we want this hyperplane to be located such that all training examples with \( \zeta^{(i)} = 1 \) are found on one side of the hyperplane, while the remaining training examples with \( \zeta^{(i)} = -1 \) are found on the other side. This is described by the properties

\[
\begin{align*}
& w^T \xi^{(i)} + b > 0 \text{ if } \zeta^{(i)} = 1, \\
& w^T \xi^{(i)} + b < 0 \text{ if } \zeta^{(i)} = -1.
\end{align*}
\]

2. The unit-length normal of the hyperplane, \( n \), can be found by normalising \( w \) which is orthogonal to the hyperplane:

\[
n = \frac{w}{\|w\|}.
\]

\( n \) can then be scaled by the unknown \( \gamma \) to find the vector denoted by the dashed line in the sketch. Looking further at the sketch, it appears as if we could find \( \xi' \) by vector-subtraction:

\[
\xi'^{(i)} = \xi^{(i)} - \frac{w}{\|w\|} \gamma^{(i)}.
\]

However, this does not work if \( \xi \) were on the other side of the hyperplane. Hence, we have to use \( \zeta \) to cover both cases simultaneously:

\[
\xi'^{(i)} = \xi^{(i)} - \frac{w}{\|w\|} \gamma^{(i)} \zeta^{(i)}.
\]

Because \( \xi'^{(i)} \) is on the hyperplane, we can plug it into the hyperplane equation and rearrange:
0 = \mathbf{w}^T \left( \xi^{(i)} - \frac{\mathbf{w}}{\|\mathbf{w}\|} \gamma^{(i)} \zeta^{(i)} \right) + b = \mathbf{w}^T \xi^{(i)} - \|\mathbf{w}\| \gamma^{(i)} \zeta^{(i)} + b. \\

Solving for $\gamma^{(i)}$ and using the property $\zeta = \zeta^{-1}$ for restructuring yields:

$$
\gamma^{(i)} = \frac{\mathbf{w}^T \xi^{(i)} + b}{\|\mathbf{w}\| \zeta^{(i)}} = \zeta^{(i)} \frac{\mathbf{w}^T \xi^{(i)} + b}{\|\mathbf{w}\|}.
$$

3. If we plug the constraint $\|\mathbf{w}\| = 1$ into $\gamma^{(i)}$ from the previous question, we get

$$
\hat{\gamma}^{(i)} = \frac{\mathbf{w}^T \xi^{(i)} + b}{\|\mathbf{w}\|} \frac{\zeta^{(i)}}{\zeta^{(i)}} = \gamma^{(i)} \left( \mathbf{w}^T \xi^{(i)} + b \right)
$$

which is exactly what we are maximising. Intuitively, maximising this margin is exactly what we want. We are looking for $\mathbf{w}$ and $b$ such that the distance between the hyperplane and the points closest to the hyperplane is maximal.

Furthermore, the constraint $\|\mathbf{w}\| = 1$ prevents us from scaling $\mathbf{w}$ and $b$ arbitrarily without affecting the position of the hyperplane. Without this constraint, $\mathbf{w}$ and $b$ were free in the sense that several pairs of $\mathbf{w}$ and $b$ would describe the same hyperplane.

4. First, let us define

$$
\gamma = \frac{\hat{\gamma}}{\|\mathbf{w}\|}
$$

as we did before. The optimisation problem now can be written as

$$
\max_{\gamma, \mathbf{w}, b} \frac{\hat{\gamma}}{\|\mathbf{w}\|} \quad \text{s.t.} \quad \zeta^{(i)} \left( \mathbf{w}^T \xi^{(i)} + b \right) \geq \hat{\gamma} \quad (i = 1, \ldots, m).
$$

This allows us to move the scaling constraint discussed above, $\|\mathbf{w}\| = 1$, away from $\mathbf{w}$ and imposing it onto $\hat{\gamma}$: $\hat{\gamma} = 1$. Reformulating the optimisation once more leads to

$$
\max_{\mathbf{w}, b} \frac{1}{\|\mathbf{w}\|} \quad \text{s.t.} \quad \zeta^{(i)} \left( \mathbf{w}^T \xi^{(i)} + b \right) \geq 1 \quad (i = 1, \ldots, m).
$$

Finally, instead of maximising $\frac{1}{\|\mathbf{w}\|}$, we might just as well minimise $\|\mathbf{w}\|^2$. Furthermore, the only thing the factor $\frac{1}{2}$ does is scaling the objective function; the location of the minimum remains unchanged. Hence, it follows that
\[
\min_{\mathbf{w}, b} \|\mathbf{w}\|^2 \text{ s.t. } \zeta^{(i)} \left( \mathbf{w}^T \xi^{(i)} + b \right) \geq 1 \quad (i = 1, \ldots, m)
\]

is the very same optimisation problem, just with nicer (=convex) properties.

5. The generalised Lagrangian is:
\[
\mathcal{L}(\mathbf{w}, b, \alpha) = \frac{1}{2} \|\mathbf{w}\|^2 - \sum_{i=1}^{m} \alpha_i \left( \zeta^{(i)} \left( \mathbf{w}^T \xi^{(i)} + b \right) - 1 \right).
\]

6. The primal problem is
\[
\min_{\mathbf{w}, b} \max_{\alpha} \mathcal{L}(\mathbf{w}, b, \alpha).
\]

The dual problem is
\[
\max_{\alpha} \min_{\mathbf{w}, b} \mathcal{L}(\mathbf{w}, b, \alpha).
\]

Solving \(\theta_D\):

\[
\nabla_{\mathbf{w}} \mathcal{L}(\mathbf{w}, b, \alpha) = \mathbf{w} - \sum_{i=1}^{m} \alpha_i \xi^{(i)} \zeta^{(i)} = 0 \quad \Rightarrow \quad \mathbf{w} = \sum_{i=1}^{m} \alpha_i \xi^{(i)} \zeta^{(i)}
\]

\[
\frac{\partial \mathcal{L}(\mathbf{w}, b, \alpha)}{\partial b} = -\sum_{i=1}^{m} \alpha_i \zeta^{(i)} = 0 \quad \Rightarrow \quad \sum_{i=1}^{m} \alpha_i \zeta^{(i)} = 0
\]
7. This requires a few steps of simplification and rearrangement:

\[
L = \frac{1}{2} \| w \|^2 - \sum_{i=1}^{m} \alpha_i \left( \zeta^{(i)} \left( w^T \xi^{(i)} + b \right) - 1 \right)
\]

\[
= \frac{1}{2} \left\| \sum_{i=1}^{m} \alpha_i \zeta^{(i)} \right\|^2 - \sum_{i=1}^{m} \alpha_i \left( \zeta^{(i)} \left( \sum_{j=1}^{m} \alpha_j \xi^{(j)} \right)^T \xi^{(i)} + b \right) - 1
\]

\[
= \frac{1}{2} \left\| \sum_{i=1}^{m} \alpha_i \zeta^{(i)} \right\|^2 - \sum_{i=1}^{m} \alpha_i \left( \sum_{j=1}^{m} \alpha_j \xi^{(j)} \right)^T \xi^{(i)} - b \sum_{i=1}^{m} \alpha_i \zeta^{(i)} + \sum_{i=1}^{m} \alpha_i
\]

\[
= \frac{1}{2} \left( \sum_{j=1}^{m} \alpha_j \xi^{(j)} \right)^T \left( \sum_{i=1}^{m} \alpha_i \zeta^{(i)} \right) - \left( \sum_{j=1}^{m} \alpha_j \xi^{(j)} \right)^T \left( \sum_{i=1}^{m} \alpha_i \zeta^{(i)} \right) + \sum_{i=1}^{m} \alpha_i
\]

\[
= \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \left( \sum_{j=1}^{m} \alpha_j \xi^{(j)} \right)^T \left( \sum_{i=1}^{m} \alpha_i \zeta^{(i)} \right)
\]

8. This question is about identifying support vectors (see A.3).

a) \( g_i \) is the constraint ensuring that the hyperplane is located such that the closest training examples are as far away as possible. Note that \( g_i(w, b) = 0 \) for support vectors and \( g_i(w, b) < 0 \) for any other training example. You can compare \( g_i \) to the formulae derived in B.2 and B.3 to see why this is.

b) Because of \( \alpha_i^* g_i(w^*, b^*) = 0 \), any \( \alpha_i^* \neq 0 \) implies \( g_i(w, b) = 0 \), hence the corresponding training example is a support vector.

c) Terms of sums with coefficients \( \alpha_i = 0 \) do not have to be considered when summing, which is an advantage. However, many critical quantities can be pre-computed anyway.
9. The key message of this question is to understand the kernel trick. In particular, expressing the problem in terms of inner products only allows for efficient computation.

a) For general \( \xi \in \mathbb{R}^n \):

\[
\langle \phi(\xi^{(i)}), \phi(\xi^{(j)}) \rangle = \sum_{k=1}^{n} \sum_{l=1}^{n} (\xi_k^{(i)} \xi_l^{(j)})^2 = \sum_{k=1}^{n} \sum_{l=1}^{n} \xi_k^{(i)} \xi_l^{(i)} \xi_l^{(j)} \xi_k^{(j)}
\]

\[
\sum_{m=1}^{n} \xi_m^{(i)} \xi_m^{(j)} \left( \sum_{m=1}^{n} \xi_m^{(i)} \xi_m^{(j)} \right) = (\xi^{(i)T} \xi^{(j)})^2
\]

b) Projecting \( \xi \) into feature-space is of polynomial complexity, \( O(n^2) \).

c) However, the inner product of two projections can be computed in linear time \( (O(n)) \) using the identity proven in a).

C. Hopfield Nets

Sadly, the weight matrix on the task sheet was not properly aligned; it was meant to be

\[
W = \frac{1}{5} \begin{pmatrix}
0 & -3 & 3 \\
-3 & 0 & -3 \\
3 & -3 & 0
\end{pmatrix}
\]

We will use this matrix throughout the following solutions.

1. The network is shown in the following figure:

2. Each additional neuron doubles the total number of states. With one neuron, there are two states. Two neurons lead to four states and the third neuron doubles to number of states again to \( 2^3 = 8 \).
3. Using the notation \((x_1, x_2, x_3)\) for the state \(x_i\) of node \(i\), there are two stable states: \((-1, 1, -1)\) and \((1, -1, 1)\). This can be visualised nicely in 3D-state space where arrows from one state to another denote how the state changes for unstable states: