Task Sheet 3

Mathias Weyland (mathias.weyland@uzh.ch)

Notes

The questions in this task sheet are very similar to the ones that you will find in the final examination. Please remember to write both your name and student ID number at the top of each sheet before it is handed in. You are encouraged to work in groups of two. If you do so, please hand in only one solution and write both names and student IDs on it. Please make an effort to write legibly.

When you hand in the solutions, please staple the sheets together. Alternatively, you can send your solution as one single PDF file by email to mathias.weyland@uzh.ch.

A. Lagrangian

The generalised Lagrangian method plays an important role in the derivation of support vector machines (SVMs). In this section, we revisit Lagrange multipliers and their generalisation in order to tackle constrained optimisation problems.

A.1 Let us consider the minimisation problem 1P

$$\min_{\boldsymbol{x}} f(\boldsymbol{x}) \text{ subject to the constraints } h_i(\boldsymbol{x}) = 0,\ i = 1, \ldots, l.$$

We set up the Lagrangian

$$\mathcal{L}(\boldsymbol{x}, \boldsymbol{\beta}) = f(\boldsymbol{x}) + \sum_{i=1}^{l} \beta_i h_i(\boldsymbol{x})$$

and solve

$$\nabla_{\boldsymbol{x},\boldsymbol{\beta}} \mathcal{L}(\boldsymbol{x}, \boldsymbol{\beta}) = \mathbf{0}$$

to find the solution. Compute the partial derivatives $\frac{\partial \mathcal{L}}{\partial \beta_i}$ and use them to show how the constraints are embedded into $\mathcal{L}$.

A.2 Let us now consider the minimisation problem 2P

$$\min_{\boldsymbol{x}} f(\boldsymbol{x}) \text{ subject to } g_i(\boldsymbol{x}) \leq 0, i = 1, ..., k \text{ and } h_j(\boldsymbol{x}) = 0, i = 1, ..., l.$$

The difference between the previous problem and this one are the additional inequality constraints $g_j(\boldsymbol{x}) \leq 0$. We define the generalised Lagrangian

$$\mathcal{L}(\boldsymbol{x}, \boldsymbol{\alpha}, \boldsymbol{\beta}) = f(\boldsymbol{x}) + \sum_{i=1}^{k} \alpha_i g_i(\boldsymbol{x}) + \sum_{i=1}^{l} \beta_i h_i(\boldsymbol{x})$$

and consider the primal problem

$$p^* = \min_{\boldsymbol{x}} \underbrace{\max_{\boldsymbol{\alpha}, \boldsymbol{\beta}, \alpha_i \geq 0} \mathcal{L}(\boldsymbol{x}, \boldsymbol{\alpha}, \boldsymbol{\beta})}_{\theta_{\mathcal{P}}(\boldsymbol{x})} = \min_{\boldsymbol{x}} \theta_{\mathcal{P}}(\boldsymbol{x}).$$

Note that $\theta_{\mathcal{P}}(\boldsymbol{x})$ is a problem in $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$, but not in $\boldsymbol{x}$. Show that $\min_{\boldsymbol{x}} \theta_{\mathcal{P}}(\boldsymbol{x})$ is our original minimisation problem. Hints: What is $\theta_{\mathcal{P}}(\boldsymbol{x})$ if an inequality constraint is violated (i.e. $g_i(\boldsymbol{x}) > 0$)? What is $\theta_{\mathcal{P}}(\boldsymbol{x})$ if an equality constraint is violated (i.e. $h_i(\boldsymbol{x}) \neq 0$)? And finally, what is $\theta_{\mathcal{P}}(\boldsymbol{x})$ if none of the constraints are violated?

A.3 While deriving the algorithm for support vector machines, we define 1P

$$\theta_{\mathcal{D}}(\alpha, \beta) = \min_{\boldsymbol{x}} \mathcal{L}(\boldsymbol{x}, \boldsymbol{\alpha}, \boldsymbol{\beta})$$

and maximise it:

$$\max_{\boldsymbol{\alpha}, \boldsymbol{\beta}, \alpha_i \geq 0} \theta_{\mathcal{D}}(\alpha, \beta)$$

Comparing the primal and the dual problem,

$$\begin{aligned} p^* &= \min_{\boldsymbol{x}} \max_{\boldsymbol{\alpha}, \boldsymbol{\beta}, \alpha_i \geq 0} \mathcal{L}(\boldsymbol{x}, \boldsymbol{\alpha}, \boldsymbol{\beta}) \\ d^* &= \max_{\boldsymbol{\alpha}, \boldsymbol{\beta}, \alpha_i \geq 0} \min_{\boldsymbol{x}} \mathcal{L}(\boldsymbol{x}, \boldsymbol{\alpha}, \boldsymbol{\beta}) \end{aligned}$$

we see that we end up with two problems that are very similar. State one reason why we bother with the dual problem instead of sticking to the primal one.

B. Support Vector Machines (SVMs)

In this section, we revisit the intuitions behind support vector machines. Assume that the training set is linearly separable in the input space unless stated otherwise. The notation used is as follows: $\left(\boldsymbol{\xi}^{(i)}, \zeta^{(i)}\right)$ is the i^{th} training example with features $\boldsymbol{\xi}^{(i)} \in \mathbb{R}^n$ and label $\zeta^{(i)} \in \{-1, 1\}$. The decision boundary (separating hyperplane) is denoted by $\boldsymbol{w}^\mathsf{T}\boldsymbol{x} + b = 0$. $||\boldsymbol{v}||$ denotes the norm of $\boldsymbol{v}$ and $\langle \boldsymbol{a}, \boldsymbol{b} \rangle$ denotes the inner product of $\boldsymbol{a}$ and $\boldsymbol{b}$.

You might find the following helpful: Videos of the Machine Learning class at Stanford by Andrew Ng., CS229:
`http://cs229.stanford.edu/`

B.1 For the SVM to work properly, we want to ensure that the following two properties hold true for all i: 1P

$$\boldsymbol{w}^\mathsf{T}\boldsymbol{\xi}^{(i)} + b > 0 \text{ if } \zeta^{(i)} = 1,\ \boldsymbol{w}^\mathsf{T}\boldsymbol{\xi}^{(i)} + b < 0 \text{ if } \zeta^{(i)} = -1.$$

Explain why this is.

B.2 Figure 1 shows data from two classes separated by a hyperplane $\boldsymbol{w}^\mathsf{T}\boldsymbol{x} + b = 0$. 2P

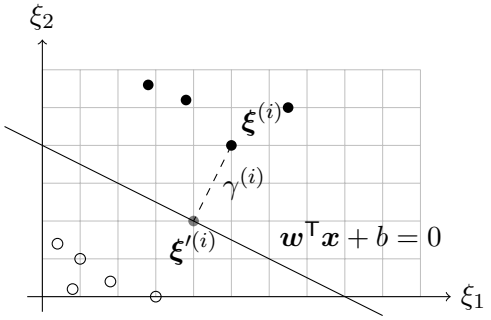

Figure 1: Example of a decision boundary for a linearly separable training set in $\mathbb{R}^2$.

Look at the point $\boldsymbol{\xi}'^{(i)}$ which is the orthogonal projection of $\boldsymbol{\xi}^{(i)}$ onto the hyperplane. The distance between $\boldsymbol{\xi}'^{(i)}$ and $\boldsymbol{\xi}^{(i)}$ is called the *geometric margin* with respect to a training example, $\gamma^{(i)}$. Note that $\gamma^{(i)}$, being a distance, is always non-negative.

Express $\boldsymbol{\xi}'^{(i)}$ in terms of $\boldsymbol{\xi}^{(i)}$, $\gamma^{(i)}$, $\zeta^{(i)}$ and $\boldsymbol{w}$. Then derive a closed form solution for $\gamma^{(i)}$. Hint: What is the normal of the hyperplane? Use the fact that $\boldsymbol{\xi}'^{(i)}$ lies on the hyperplane.

B.3 Let us define the geometric margin with respect to the whole training set, γ, as follows: 2P

$$\gamma = \min_i \gamma^{(i)}.$$

This is in some sense the "worst case". Our goal now is to maximise γ, i.e.

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

Explain why this is an intuitive objective function. Why do we need the constraint $||\boldsymbol{w}|| = 1$? Relate your answer to the previous two questions.

B.4 The optimization stated in the previous question could be solved, but its non-convex nature may lead to the convergence towards non-global optima. Hence, the optimization is rewritten such that it can be solved in a more elegant way: 2P

$$\min_{\boldsymbol{w},b} \frac{1}{2}||\boldsymbol{w}||^2 \text{ s.t. } \zeta^{(i)} \left(\boldsymbol{w}^\mathsf{T}\boldsymbol{\xi}^{(i)} + b\right) \geq 1 \ (i = 1, ..., m).$$

Explain why this is just another way of formalising the same optimization. You do not have to care about the factor $\frac{1}{2}$, this is just a scaling factor that will lead to a slightly prettier derivative later on.

B.5 Set up the generalised Lagrangian $\mathcal{L}(\boldsymbol{w}, b, \boldsymbol{\alpha})$. You can use the formula from question A.2, but note that $\boldsymbol{x}$ translates to $\boldsymbol{w}$ and b and that there are no equality constraints. The latter means that this Lagrangian is lacking the second sum. 1P

B.6 State the dual problem and solve $\theta_\mathcal{D}$ by computing $\nabla_{\boldsymbol{w}}\mathcal{L} = \mathbf{0}$ and solving for $\boldsymbol{w}$. Also find $\frac{\partial \mathcal{L}}{\partial b}$. 2P

B.7 Plug the $\boldsymbol{w}$ obtained in the previous question into $\mathcal{L}$ and rearrange to get 2P

$$\mathcal{L}(\boldsymbol{w}, b, \boldsymbol{\alpha}) = \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i=1}^{m} \sum_{j=1}^{m} \zeta^{(i)}\zeta^{(j)}\alpha_i\alpha_j \left\langle \xi^{(i)}, \xi^{(j)} \right\rangle.$$

Hence the final optimisation problem is to maximize this function subject to $\sum y_i \alpha_i = 0$[1]. (This constraint follows from $\frac{\partial \mathcal{L}}{\partial b} = 0$.) This will result in $\boldsymbol{\alpha}$, $\boldsymbol{w}$ and b.

B.8 A new input $\tilde{\boldsymbol{\xi}}$ can be classified as follows:

$$\tilde{y}\left(\tilde{\boldsymbol{\xi}}^{(i)}\right) = \begin{cases} 1 & \text{if } \boldsymbol{w}^\mathsf{T}\tilde{\boldsymbol{\xi}} + b > 0 \\ -1 & \text{otherwise} \end{cases}$$

By plugging the $\boldsymbol{w}$ obtained above into this equation, we get

$$\boldsymbol{w}^\mathsf{T}\tilde{\boldsymbol{\xi}} + b = \sum_{i=1}^{m} \alpha_i \zeta^{(i)} \left\langle \boldsymbol{\xi}^{(i)}, \tilde{\boldsymbol{\xi}} \right\rangle + b.$$

You do not have to prove this identity. Note however that after all this math, all $\boldsymbol{\xi}^{(i)}$ in both the optimisation problem as well as the classification rule are expressed in terms of inner products only. In this and the next question, we will look at how this can be exploited.

Coming back to the Lagrangian method, it can be shown that the identity

$$\alpha_i^* g_i(\boldsymbol{w}^*, b^*) = 0$$

holds for any solution α_i^*, $\boldsymbol{w}^*$, b^*.

a) What is the geometric meaning of $g_i(\boldsymbol{w}^*, b^*) = 0$? 2P

b) If $\alpha_i^* \neq 0$, what can you conclude about g_i and its corresponding input? 1P

c) Usually, most α_i^* are 0. Does this make classification efficient? Please explain. 1P

B.9 In a last step, we will have a look at kernels. So far, we have assumed that our problem is linearly separable in input space. Let us relax this assumption and assume that the problem is linearly separable in feature space. Let us define a function $\phi(\boldsymbol{\xi})$ that maps an input to a high-dimensional feature space. For instance, we could have

$$\phi(\boldsymbol{\xi}) = (\xi_1\xi_1, \xi_1\xi_2, \xi_1\xi_3, \xi_2\xi_1, \xi_2\xi_2, \xi_2\xi_3, \xi_3\xi_1, \xi_3\xi_2, \xi_3\xi_3)^\mathsf{T}$$

for $\boldsymbol{\xi} \in \mathbb{R}^3$. $\phi(\boldsymbol{\xi})$ is called a *feature*. Note however that we are not required to compute $\phi(\boldsymbol{\xi})$ for a single $\boldsymbol{\xi}$. What we actually have to do instead is computing

[1] You don't have to do that.

the inner product $\left\langle \phi\left(\boldsymbol{\xi}^{(i)}\right), \phi\left(\boldsymbol{\xi}^{(j)}\right)\right\rangle$. While this looks computationally more intensive, it actually is not if we use the *kernel trick*:

a) Show that $\left\langle \phi\left(\boldsymbol{\xi}^{(i)}\right), \phi\left(\boldsymbol{\xi}^{(j)}\right)\right\rangle = \left(\boldsymbol{\xi}^{(i)\mathsf{T}}\boldsymbol{\xi}^{(j)}\right)^2$ for the example above, but with $\boldsymbol{\xi} \in \mathbb{R}^n$. 2P

b) How many operations are required to compute $\phi(\boldsymbol{\xi})$ in terms of n? 1P

c) How many operations are required to compute $\left(\boldsymbol{\xi}^{(i)\mathsf{T}}\boldsymbol{\xi}^{(j)}\right)^2$ in terms of n? 1P

C. Hopfield Nets

For the questions on Hopfield nets, consider a network consisting of 3 neurons with weight matrix

$$W = \frac{1}{5}\begin{pmatrix} 0 & -3 & 3 \\ -3 & 0 & -3 \\ 3 & -30 & \end{pmatrix}$$

C.10 Draw the network including its connections and weights. 1P

C.11 The neurons can only have two states, either 1 or -1. How many possible states can the network have? 2P

C.12 Which of all the possible states are stable? Unstable states converge towards stable states. calculate every step, until a stable state is reached. Draw also the arrow going from every unstable to its corresponding stable state into the 3D state-space. 3P