Statistical Machine Learning: Exercise 4

Group A: Kexin Wang (2540047), Paul Philipp Seitz (2337506)

TECHNISCHE UNIVERSITÄT DARMSTADT

Summer Term 2022

Task 1: Support Vector Machines (35 Points)

1a) Definition (3 Points)

SVMs are algorithms that are used to dinstiguish two classes by a hyperplane with a maximized margin. SVMs can also be used to construct non-linear models via Kernel Trick. SVMs usually give better results than other linear methods. As it can be represented as a convex optimisation problem, a global optimal solution can be found using known efficient algorithms. In addition, because it depends on only a small number of support vectors, it is computationally simple and robust.

1b) Quadratic Programming (2 Points)

Constrained Optimization Problem:

$$arg \max_{\mathbf{W},b} \frac{1}{2} \parallel \mathbf{w} \parallel^2 \tag{1}$$

$$s.t. \quad y_i(\mathbf{w}^\mathsf{T}\mathbf{x}_i + b) - 1 \geq 0 \quad \forall i \tag{2}$$

1c) Slack Variables (2 Points)

We can only solve the above conditional optimisation problem if the data can be perfectly linearly separated. When the data cannot be separated linearly, we often use Kernel Trick to implement a non-linear model. However, for approximately linearly separated data, not only is the construction of the above conditional optimization problem impossible to use, but it also tends to lead to overfitting. We can add slack variables to the optimisation problem so that some of the data points can be within the margins, which will allow for linear separation of the data. At the same time we add a regularisation factor C to regulate the extent to which the slack variables can be applied, avoiding the situation where any one W can satisfy the conditions of the optimisation problem. Constrained optimization problem (Soft-Margin):

$$arg \max_{\mathbf{W},b} \frac{1}{2} \parallel \mathbf{w} \parallel^2 + C \sum_{i=1}^{N} \xi_i \tag{3}$$

$$s.t. \quad y_i(\mathbf{w}^\mathsf{T}\mathbf{x}_i + b) - 1 + \xi_i \geq 0 \quad with \quad \xi_i \geq 0 \quad \forall i \tag{4}$$

1d) Optimization with Slack Variables (8 Points)

We can construct a dual problem from the original problem (1)(2) based on the Lagrange Multiplier Method:

$$\min L(\mathbf{w}, b, \alpha) = \min(\frac{1}{2} \parallel \mathbf{w} \parallel^2 - \sum_{i=1}^{N} \alpha_i (y_i(\mathbf{w}^\mathsf{T}\mathbf{x}_i + b) - 1)) \tag{5}$$

Optimization of the w, b: We can get the hyperplane:

$$\frac{\partial L}{\partial \mathbf{w}} = 0 \Rightarrow \mathbf{w} = \sum_{i=1}^{N} \alpha_i y_i \mathbf{x}_i \tag{6}$$

$$\frac{\partial L}{\partial b} = 0 \Rightarrow \sum_{i=1}^{N} \alpha_i y_i = 0 \tag{7}$$

Use the constraint $\sum_{i=1}^{N} \alpha_i y_i = 0$ and the heperplane above:

$$\max \widetilde{L}(\alpha) = \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j y_i y_j (\mathbf{x_j}^\mathsf{T} \mathbf{x_i}) \tag{8}$$

Finally, the dual optimization problem will be:

$$\max \widetilde{L}(\alpha) = \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j y_i y_j (\mathbf{x_j}^\mathsf{T} \mathbf{x_i})$$

$$s.t. \quad 0 \leq \alpha_i \leq C$$

$$s.t. \quad \sum_{i=1}^{N} \alpha_i y_i = 0$$

1e) The Dual Problem (4 Points)

Compared to the original problem, the constraints of the dual problem are very simple. This is because the dual problem takes into account the constraints of the original problem in the loss function. In addition the format of the dual problem facilitates the introduction of kernel functions for the construction of non-linear models.Accordingly, the nonlinear SVM in dual form is given below:

$$\widetilde{L}(\alpha) = \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j y_i y_j (\mathbf{x_j}^\mathsf{T} \mathbf{x_i})$$

Compared to the original form, there are fewer unknown variables in the pairwise Lagrangian, so it is easier to solve. Since we use few data points in the SVM, it is more efficient to compute by using only data points with $\alpha_i \neq 0$.

1f) Kernel Trick (6 Points)

Kernel tricks are a class of methods that transform a nonlinearly differentiable problem in a low-dimensional space into a linearly differentiable problem in a high-dimensional space.
The kernel function has non-mapped original data as an input and computes mapping the data to a higher-dimensional space and calculating a scalar product. The use of kernel tricks in support vector machines not only allows a freer implementation of various models, especially non-linear models, but can also replace the complex process of spatial projection and reduce much of the computational effort.

1g) Implementation (10 Points)