Prof. Dr. Volker Roth
volker.roth@unibas.ch

Vitali Nesterov
vitali.nesterov@unibas.ch

Department of Mathematics and Computer Science
Spiegelgasse 1
4051 Basel

# Exercise 9

Due date: **Wednesday, May 6th 2020**

## 9.1: Support Vector Machine

Recall the Perceptron learning algorithm, a solution vector is found if the data is linearly separable and no misclassification occurs. However, the found solution is arbitrary and if the data is linearly not separable, the algorithm will never converge. A Support Vector Machine finds an optimal solution vector, within a solution space, by maximizing the margin of the hyperplane. Further relaxation of the classification constraints allow to find a decision boundary for linearly non-separable data. Assume $d$-dimensional observations $(x_1, x_2, \ldots, x_d) \in \mathbb{R}^d$, such that a labeled dataset is $\boldsymbol{\mathcal{D}} = ((\boldsymbol{x_1}, y_1), \ldots, (\boldsymbol{x_n}, y_n))$ with $y_i \in \{-1, 1\}$ and the model parameters $\boldsymbol{w} = (w_1, w_2, \ldots, w_d)^\top \in \mathbb{R}^d$.

### Definitions

- The **relaxed classification constraints** are defined as follows:

  $$y_i \left(\boldsymbol{x}_i^\top \boldsymbol{w}\right) \geq 1 - \xi_i \quad \xi_i \geq 0,$$

  sucht that data points lie on the correct side of the margin with an additional penalty term $\xi_i$, a so called slack variable, allowing violation of the classification constraints.

- The **primal optimization problem** of a Support Vector Machine with a soft margin is given with:

  $$\begin{aligned} \underset{w,\xi}{\text{minimize}} \quad & \frac{1}{2}\|\boldsymbol{w}\|^2 + C\sum_{i=1}^{n} \xi_i \\ \text{subject to} \quad & y_i \left(\boldsymbol{w}^\top \boldsymbol{x}_i\right) \geq 1 - \xi_i \\ & \xi_i \geq 0, \end{aligned}$$

  where $C > 0$ is a fixed parameter that regulates the sensitivity of the model to the outliers in the data.

- The **dual optimization problem** is defined as follows:

$$\begin{aligned}\underset{\alpha}{\text{maximize}} \quad & \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} y_i y_j \alpha_i \alpha_j \boldsymbol{x}_i^\top \boldsymbol{x}_j \\ \text{subject to} \quad & \sum_{i=1}^{n} \alpha_i y_i = 0 \\ & 0 \leq \alpha_i \leq C,\end{aligned}$$

  where $\alpha_i$ are Lagrangian multipliers.

**Exercise**

Derive the dual optimization problem given the primal with relaxed classification constraints. Use the Lagrangian formalism to express the primal constraint as a loss term. Given a min-max-problem, swap the optimization operators to simplify the optimization task. Why is this a valid operation? Minimize the expression with respect to model parameters and the slack variables and rearrange the terms. Lastly, write down the dual optimization problem.

## 9.2: Python implementation

In the above derived results, the dual optimization problem is given. Rewrite it into a generic quadratic optimization problem by using a matrix notation form. Write a Python program for a relaxed binary classification task.

- Generate a labeled dataset from two independent 2-d Gaussians.
- Use a cvxopt package for solving a quadratic optimization problem.
- Compute required parameters for the quadratic solver and optimize the objective function with respect to the Lagrangian parameters.
- Compute the model parameters and the intercept.
- Evaluate the decision boundary given the model parameters. Plot the corresponding line and the observed data points.