Exercise Sheet 1

Integer Programming
Academic Year 2021/22: Quarter 4

Lecturer: Christopher Hojny

Problem Description

In this assignment, we want to find optimal solutions for a problem arising in data science. Suppose we are given a set of data points Z each having a label $+$ or $-$. One can think of Z being a set of measurements, e.g., within a clinical study, and the labels $+$ and $-$ correspond to whether the data extracted from measurement $z \in Z$ indicates a disease or not. From a mathematical perspective, however, the context of data points Z does not matter. For this reason, we just split Z into the set of "positive" measurements X and "negative" measurements Y and forget about a concrete application.

The aim of supervised learning is to find a model that distinguishes the points in X from the points in Y. In particular, if new measurements/data points arise, the model can be used to predict the label of this measurement, i.e., whether it belongs to X or Y. If the model detected by supervised learning methods is accurate enough, then this model can be used to classify new data points. This might be favorable if finding the correct label using experiments is expensive from a monetary or time perspective.

Depending on the application, one specific supervised learning method might be more appropriate than another. One of the simplest models is the model of a support vector machine.

Definition 1. *Let $X, Y \subseteq \mathbb{R}^n$ and $\varepsilon > 0$. An ε-*support vector machine (SVM) *is a linear inequality $a^\top z \leq \beta$ such that $a^\top x \leq \beta$ for all $x \in X$ and $a^\top y \geq \beta + \varepsilon$ for all $y \in Y$.*

Despite their simplicity, SVMs have been used successfully, e.g., in the context of classifying emails as spam. In certain applications, however, SVMs might not be accurate enough. This is, for example, the case if the set X admits a geometric structure.

Definition 2. *Let $X, Y \subseteq \mathbb{Z}^n$ such that $\mathrm{conv}(X) \cap Y = \emptyset$ and let $\varepsilon > 0$. An ε-*polyhedral classifier *of order k is a set of k inequalities ${a_i}^\top z \leq \beta_i$ for $i \in \{1, \dots, k\}$ such that*

- *$x \in Z$ is contained in X if ${a_i}^\top x \leq \beta_i$ for all $i \in \{1, \dots, k\}$;*
- *$y \in Z$ is contained in Y if there exists $i \in \{1, \dots, k\}$ with ${a_i}^\top y \geq \beta_i + \varepsilon$.*

Of course, a polyhedral classifier can be evaluated more easily if its order is as small as possible. The aim of this assignment is to use integer programming techniques to compute the smallest order k of a polyhedral classifier for given sets X and Y.

Exercises

As we will see below, integer programming allows us to compute the minimum order of a polyhedral classifier. However, there is no unique way of modeling this problem as a (mixed-) integer program, and we will compare two different models: a compact model and a column generation based model. The idea of the column generation model is based on the following observation. Suppose we are given an ε-polyhedral classifier with k inequalities encoded by $(a_1, \beta_1), \dots, (a_k, \beta_k)$. Each of these inequalities classifies some points of set Z to be contained in the set Y; for inequality (a_i, β_i), these points are exactly $Y_i = \{y \in Z : {a_i}^\top y \geq \beta_i + \varepsilon\}$. Moreover, because the inequalities define a polyhedral classifier, for each $y \in Y$, there exists such a set Y_i with $y \in Y_i$.

We can exploit this observation as follows. We say that a set $I \subseteq Y$ is *linearly separable* from X if there exists an inequality $a^\top z \leq \beta$ with $a^\top x \leq \beta$ for every $x \in X$ and $a^\top y \geq \beta + \varepsilon$ for all $y \in I$. Let $\mathcal{I} = \{I \subseteq Y : I \text{ is linearly separable from } X\}$. Moreover, for $y \in Y$, let $\mathcal{I}_y = \{I \in \mathcal{I} : y \in I\}$. Then,

$$
\begin{aligned}
\min \sum_{I \in \mathcal{I}} & w_I \\
\sum_{I \in \mathcal{I}_y} & w_I \geq 1, && y \in Y, \\
& w_I \in \mathbb{Z}_+, && I \in \mathcal{I},
\end{aligned}
$$

is an integer programming formulation for finding a minimum order polyhedral classifier. Note that the number of variables in this model can be extremely large. Our aim is to compare this model with a more compact model.

Exercise 1 (A compact model) 2 points

Let $X, Y \subseteq \mathbb{Z}^n$ be such that $\operatorname{conv}(X) \cap Y = \emptyset$. Let K be an upper bound on the minimum order of a polyhedral classifier for X and Y, and let $\varepsilon > 0$. Develop a mixed-integer linear program to model the problem of finding a minimum order ε-polyhedral classifier for X and Y. To this end, make use of the following variables:

- for $i \in \{1, \dots, K\}$, variables $(a_i, \beta_i) \in \mathbb{R}^n \times \mathbb{R}$ model the coefficients of the i-th inequality, i.e., ${a_i}^\top z \leq \beta_i$ is the i-th inequality used by the classifier;
- for $i \in \{1, \dots, K\}$, variable $u_i \in \{0, 1\}$ models whether the i-th inequality is used by the polyhedral classifier;
- for $y \in Y$ and $i \in \{1, \dots, K\}$, variable s_{yi} models whether the i-th inequality is violated by point y.

Create a PDF document that contains your model as well as an explanation thereof. Moreover, implement your model in Gurobi.

Exercise 2 (A weighted SVM) 1 point

Let $X, Y \subseteq \mathbb{Z}^n$ be non-empty and let $c \in \mathbb{R}^Y$ be a vector that assigns each point in Y a weight. If X and Y are not linearly separable, then we cannot find an ε-SVM distinguishing both sets. Instead, we can look for an inequality $a^\top z \leq \beta$ that separates the points from X with the largest weight, i.e., an inequality with

- $a^\top x \leq \beta$ for all $x \in X$;
- the weight of $Y^{(a,\beta)} = \{y \in Y : a^\top y \geq \beta + \varepsilon\}$ is maximal among all possible choices of (a, β).

Model this problem as a mixed-integer linear program. Add your solution together with an explanation to the PDF of the previous exercise.

Exercise 3 (A column-generation model) 3.5 points

Using Gurobi, implement a column generation procedure to solve the LP relaxation of the above column generation based model, i.e., replace the requirement $w_I \in \mathbb{Z}_+$ by $w_I \geq 0$. To this end,

- introduce a suitable subset of variables w_I, $I \in \mathcal{I}$, such that the initial LP relaxation is feasible. The number of initial variables must not exceed $|Y|$;
- implement a pricing procedure to decide whether new variables need to be added to the current LP relaxation.

Hint: Gurobi can be terminated early if a solution with objective value at least x has been found, using the command `model.Params.BestObjStop = x`. This might be useful if, e.g., a pricing problem is difficult to solve. Here, it is not necessary to find an optimal solution. Any solution with negative reduced cost is sufficient.

Exercise 4 (Finding integer solutions) 0.5 points

The column generation procedure only allows to solve the LP relaxation, i.e., some variables might be fractional in the solution found by Gurobi. Use Gurobi to find an optimal integer solution that uses only the variables that have been generated during column generation.

Exercise 5 (Estimating the Strength of the Column Generation Model) 1+1+1 points

Let $v^\star_{\mathrm{CG}}$ and $v^\star_{\mathrm{comp}}$ be the optimal values of the LP relaxations of the column generation and a compact model, respectively. One reason to use the column generation model is that $v^\star_{\mathrm{CG}}$ is usually much larger than $v^\star_{\mathrm{comp}}$. The aim of this exercise is to find a concrete lower bound on $v^\star_{\mathrm{CG}}$. To this end, we consider special subsets of Y. A set $S \subseteq Y$ is called *special* if $\mathrm{conv}(\{y_1, y_2\}) \cap \mathrm{conv}(X) \neq \emptyset$ for all distinct $y_1, y_2 \in Y$. Add your solutions to the following questions to your PDF.

1. Let $a^\top z \leq \beta$ be an inequality that holds for all $x \in X$. Prove that, for any special set $S \subseteq Y$, there is at most one $y \in S$ such that $a^\top y > \beta$.
2. Prove that
$$v^\star_{\mathrm{CG}} \geq \max_{\substack{S \subseteq Y:\\ S \text{ is special}}} |S|.$$
That is, the size of any special set is a lower bound on $v^\star_{\mathrm{CG}}$.
3. Let $X = \{0,1\}^n$ and let $Y = \mathbb{Z}^n \setminus X$. Show that, in this case, $v^\star_{\mathrm{CG}} \geq n$.

Test Instances

We provided a template file and a few test instances on CANVAS. The template file already contains a routine for reading an instance of our problem in the section "Auxiliary Functions" as well as methods `exercise1()`, `exercise3()`, and `exercise4()`. Extend this template by your own routines and optimization models. Please put your own methods in the corresponding sections (Auxiliary Functions, Exercise 1, Exercise 3, etc.) of this file and submit your solution via CANVAS. Your solution should make use of the previously mentioned functions by providing them suitable input. For example, the model of Exercise 1 shall be built and solved by calling `exercise1()` with a suitable input. These functions are called in the main part of the template, so only adjust the input of these functions in `main`.

To run the code using the template file, you have to specify a file containing data, see CANVAS for exemplary data; depending on your operating system, a call of the template might look like this

```
python3 column_generation.py /path/to/datafile.cg
```

The format of the data file is as follows. The first line contains an upper bound on the order of a minimal classifier. The elements of the sets X and Y are encoded by lines starting with "x" and "y", respectively. For all test instances, assume that we want to compute an ε-polyhedral classifier with $\varepsilon = 0.001$.

Submission and Grading

To submit your solution on CANVAS,

- for the practical exercises, extend the provided template file by adding your own methods and data structures to the corresponding sections of the template;
- for the theoretical exercises, prepare a PDF file containing your solution;
- work in groups of up to 3 students, list all group members in the header section of the template file and your PDF;
- submit your solution until May 12, 20:00.

To achieve a top score, your solution has to satisfy the following criteria:

- Answers to all exercises are provided;
- The implementation and answers to theoretical questions are correct;
- The pricing problem in Exercise 3 is not build from scratch in each iteration. Instead, only the objective function is updated;
- The column generation problem in Exercise 3 is not build from scratch in each iteration. Instead, it is extended by a new column for each $I \in \mathcal{I}$ that is added to the problem (except for the initial sets I).
- Solving the column generation model's LP relaxation does not take longer than solving the compact model with integrality constraints.