Sparse Support Vector Machines with PySCIPOpt

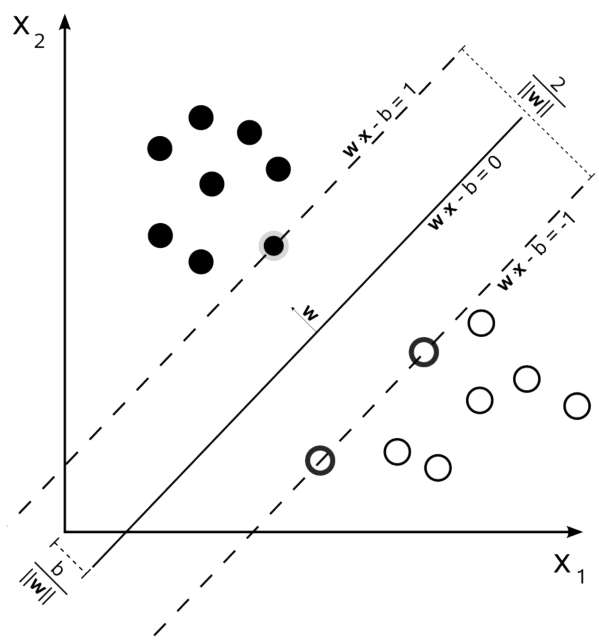
3rd IMI-ISM-ZIB Modal Workshop, Tokyo, Japan
Sep. 26, 2018

Introduction

Given a set X of n d-dimensional data points labeled by $y \in \{-1, 1\}^n$, we want to find a function that classifies each point to be in one of the two sets based on its location. In this exercise, we use a *linear classification model*, i.e. we look for a hyperplane that separates these two sets[1]. Additionally, we are interested in a classifier that is sparse, ie that uses only as few of the available data dimensions as possible.

A hyperplane h is given by a normal vector ω and a translation b and the classification $h_{\omega,b}$ is defined as follows:

$$h_{\omega,b}(x) = \operatorname{sgn}(\omega^T x + b) \in \{-1, 1\}$$



We require that this evaluation coincides with the given classification of the points[2]. Additionally we require the parameter ω and b to be such that no points lie in the *margin* which is defined as the following set of points x[3]:

$$\{x : |h_{\omega,b}(x)| < 1\}$$

We want to reduce the dimension of the original space and consider only a subset of features[4]. In our model this implies that a certain fraction of weights is required to be zero resulting in a *sparse classifier*.

A sparse classifier with sparsity ρ is a linear classifier where a fraction of the ω entries is equal to 0:

$$\rho(\omega) = \frac{|\{i \colon \omega_i = 0\}|}{d}$$

[1] graphic taken from: https://en.wikipedia.org/wiki/File:Svm_max_sep_hyperplane_with_margin.png.

[2] This may not always be possible. In that unlucky case we require the condition for as many datapoints x as possible and penalize misclassifications.

[3] It is easy to see that the width of the margin decreases when the length of ω increases.

[4] This classification to be successful meaning that not all features are relevant.

Advantages of a sparse classifier are *a smaller cost* of the classification and the fact that it results in a *simpler model*[5]

Material

The sub directory `scip-workshop/support-vector-machine` is the place where you should place your python script. It also contains a subdirectory `data`, which contains means to read in the data by the following python commands:

```
from data.load_cancer import load_cancer

dataset = load_cancer()
X = np.array(dataset.data)
y = np.array(dataset.targets)
```

The dataset is the classification of benign ($y = -1$) or malignant ($y = 1$) breast cancer based on 30 features and contains 569 data points. Out of these 212 are malignant and 357 are benign. It is taken from:

`http://archive.ics.uci.edu/ml/machine-learning-databases/breast-cancer-wisconsin/`

For all the exercises you should split the dataset into two parts, one that you train on and one that you predict.

Exercise 1

Your first task is to implement a linear svm with the following model.
Let the set of datapoints consist of n d-dimensional features $X \in \mathbb{R}^{n,d}$, labeled by $y \in \{-1, +1\}^n$ and let $C > 0$ be a regularization parameter. To penalize wrongly classified datapoints, consider as a loss function the *Hinge loss*[6]:

$$l^i(t) := \max\{0, 1 - y^i t\} \quad \text{for } i \in \{1, \dots, n\}$$

As you want to minimize the penalty, and maximize the margin (equivalently minimize the length of ω, since the width of the margin is given by $\frac{2}{\|\omega\|}$), the model can now be written as the following optimization problem:

$$\min_{\omega,b} \frac{C}{n} \sum_{i=1}^{n} l^i(\omega^T X^i + b) + \frac{1}{2}\|\omega\|_2^2$$

Substituting the Hinge loss for a variable

$$\begin{aligned} \xi^i &\geq l^i(\omega^T X^i + b) \\ &= \max\{0, 1 - y^i(\omega^T X^i + b)\}, \end{aligned}$$

the above problem is equivalent to:

[5]. *Occam's razor*: from a set of solutions to a problem select the one that makes the fewest assumptions.
[6] Here t is the evaluation of the classifier on datapoints.

$$
\begin{aligned}
\min_{\omega,b} \quad & \tfrac{C}{n}\sum_{i=1}^{n} \xi^i + \tfrac{1}{2}\|\omega\|_2^2 \\
\text{such that} \quad & 1 - y^i(\omega^T X^i + b) \le \xi^i, \quad && i \in \{1,\dots,n\} \\
& 0 \le \xi^i, && i \in \{1,\dots,n\}
\end{aligned}
$$

Report your results in terms of the accuracy (percentage of misclassified test examples), and the individual numbers of misclassified positive and negative test samples, respectively.

Hints The regularization parameter C is usually optimized to produce the best prediction. We can start with a value of 1.0. Once the model works, you can play around with different powers of 10 to produce the best result.

Exercise 2

Modify the model from Exercise 1 to produce a sparse classifier.
To implement a sparse classifier with sparsity ρ, add additional constraints and variables to the model[7].

$$
\begin{aligned}
& \textstyle\sum_{j\in d} v_j \le \rho \cdot d \\
& -B \cdot v_j \le \omega_j \le B \cdot v_j, \quad && j \in \{1,\dots,d\} \\
& v_j \in \{0,1\}, && j \in \{1,\dots,d\}
\end{aligned}
$$

For $i \in \{1,\dots,d\}$ assume the weights ω_j to be bounded by $-B$ and B for a bound $B > 0$. Only a fraction of these new binary indicator variables v_j are allowed to be nonzero. Then all the v_j that are zero will force their corresponding ω_j to be zero.
How sparse can you make the classifier to produce results comparable to Exercise 1?

Hint A good first choice on B would be 10, as the optimal solutions usually lie in within the interval $[-10, 10]$.

Exercise 3

Depending on the number of positive and negative samples in the data we might want to weight the penalties differently, ensuring that points from one of the sets have a higher probability to be classified correctly[8]. This correction c_i is applied in the objective function:

$$
\frac{C}{n}\sum_{i=1}^{n} c_i \xi^i + \frac{1}{2}\|\omega\|_2^2, \quad \text{where } c_i = \begin{cases} \alpha & \text{if } y_i = 1 \\ \beta & \text{if } y_i = -1 \end{cases}
$$

Your task is to balance the data.

[7] Another possibility would be to prefer sparse solutions using an $L1$-norm in the objective function.
[8] An application would be medical tests, where a false negative should be highly unlikely, whereas a false positive is not disastrous.