

# Convex Optimization

MTech IS 2019

## 1 Presentation

The goal of this test is to design and test an automatic spam filter using convex programming. In order to design such a filter, you will design function that can automatically predict if a given email is a spam or not. The functions will be automatically optimized to be as accurate as possible using a dataset of known undesirable and normal emails.

Each email is a text. In order to process them easily, each text is converted into a vector of positive scalar of dimension 57. Most of these 57 features are the frequencies of particular words that have been observed to be important to decide whether an e-mail is a spam or not, e.g., words like *credit*, *free* or *meeting*<sup>1</sup>.

In order to design your spam filter you can use a set of 500 emails converted into 57-dimensional vectors, together with their classes +1 (spam) or -1 (non-spam). These data are available in the respective variables `xtrain` and `ytrain`. Once you have designed a nice filter, you can test it on a set of 2000 additional e-mails stored in the variable `xtest`, and compare your prediction with the correct class stored in the variable `ytest`. All data are stored in the file

`spamdata.mat`

A precise description is available at <http://www.ics.uci.edu/>

`mllearn/databases/spambase/spambase names`

---

## 2 Building the filter: general strategy

Let  $x_1, \dots, x_n$  be the  $n = 500$  vectors of dimension  $p = 57$  available for training. We will investigate filters that compute a score for each vector  $x$  by an affine function:

$$f(x) = w^\top x + b,$$

and predict that the e-mail represented by the vector  $x$  is a spam if  $f(x) > 0$ , or is a normal e-mail if  $f(x) \leq 0$ . We denote by  $y_1, \dots, y_n$  the spam indicator variable, i.e.,  $y_i = 1$  if  $x_i$  is a spam,  $y_i = -1$  otherwise. The function  $f$  is defined by the vector  $w$  and the scalar  $b$  which we must set by fitting the training set of e-mails according to the following principles:

- $f(x)$  should be positive if  $y = +1$ , negative if  $y = -1$ . This means that in all cases  $yf(x)$  should be "as positive as possible".
- Because the value of  $f(x)$  can be arbitrarily increased by changing the scale of  $w$ , we must control this scaling, e.g., make sure  $\|w\|$  is not too large.

We will therefore investigate below filters  $f$  based on the minimization of the function:

$$\min_{w,b} \frac{1}{n} \sum_{i=1}^n L(y_i(w^\top x_i + b)) + \gamma \|w\|^2 \quad (1)$$

where  $L$  is usually a decreasing function and  $\gamma > 0$  is a parameter. This formulation will find a trade-off between the first term (increasing  $yf(x)$  on the training set) and the second one (making sure that  $f$  is not arbitrarily large because  $\|w\|$  is large). Different functions  $L$  will lead to different algorithms.

## 3 Hard-margin support vector machines

Here we take:

$$L(u) = \begin{cases} +\infty & \text{if } u < 1, \\ 0 & \text{otherwise.} \end{cases}$$

3.1. In this case rewrite (1) as a quadratic program (QP). When is it feasible? (give a geometric interpretation)

3.2. Write the dual problem. Does strong duality hold?

3.3. Write KKT conditions. Can you give an interpretation to the complementary slackness conditions?

- 3.4. How to you recover  $w$  and  $b$  if you solve the dual problem?
- 3.5. Implement the primal problem and solve it on the training set of 500 emails. What do you observe?
- 3.6. Implement the dual problem and solve it on the training set of 500 emails. What do you observe?
- 3.7.. Repeat 3.4. and 3.5. on the first 50 examples of the training set only. What do you observe?

## 4 Soft-margin support vector machines

Here we take:

$$L(u) = \begin{cases} 1 - u & \text{if } u < 1, \\ 0 & \text{otherwise.} \end{cases}$$

- 4.1. In this case rewrite (1) as a quadratic program (QP). Is it feasible? (hint: introduce slack variables)
- 4.2. Write the dual problem. Does strong duality hold?
- 4.3. Write KKT conditions. Can you give an interpretation to the complementary slackness conditions?
- 4.4. How do you recover  $w$  and  $b$  from a solution of the dual?
- 4.5. Implement the primal and the dual, optimize them on the training set for  $\gamma = 10^{-3}$ . Compare  $w$  and  $b$  obtained by the primal and the dual.
- 4.4. Optimize the primal problem on the training set, and plot the curves of percentage of errors on the training and on the test set as a function of  $\log_2(\gamma)$ , for  $\log_2(\gamma)$  in the range  $[-15 : 5]$ . What do you observe?

## 5 Regularized logistic regression

Here we take:

$$L(u) = \log(1 + e^u) - u$$

- 5.1. Show that (1) is then an unconstrained convex problem with smooth (twice differentiable) objective function.
- 5.2. Compute the gradient and the Hessian of the objective function.
- 5.2. Implement the problem and plot the curves of training and testing errors as a function of  $\log_2(\gamma)$ , for  $\log_2(\gamma)$  in the range  $[-15 : 5]$ . What do you observe?