

# IN4320 Machine Learning Assignment 1

February 13, 2018

---

We are going to consider (what we will refer to as) the nearest representor classifier (NrC). You may want to compare this classifier to the nearest mean classifier (NMC). Like NMC, NrC determines a vector for every class at training time, which we call the representors. At test time, it assigns samples to the class for which the corresponding representor is closest in terms of the Euclidean distance.

In this assignment, we consider two-class data only. In the general setting, training data consists of  $d$ -dimensional feature vectors  $x_i$  together with their corresponding labels  $y_i$ , the latter of which we agree to encode with an element from the set  $\{+, -\}$ . The two representors for the two classes are also indexed  $+$  and  $-$ , i.e.,  $r_-$  belongs to the class with labels  $-$ , while  $r_+$  represents the  $+$  class.

Now, consider the following loss function (or objective function)  $L$ :

$$L(r_-, r_+) := \left( \sum_{i=1}^N \frac{1}{N_{y_i}} \|x_i - r_{y_i}\|^2 \right) + \lambda \|r_- - r_+\|_1, \quad (1)$$

with  $\lambda$  the regularization parameter,  $N$  the number of samples in the training set,  $N_-$  and  $N_+$  the number of samples in the  $-$  and  $+$  class respectively, and  $\|\cdot\|_1$  the  $L_1$ -norm. Minimizing  $L$  both over  $r_- \in \mathbb{R}^d$  and  $r_+ \in \mathbb{R}^d$  gives us the solution to the regularized loss (for that  $\lambda$ ). These optimal representors then fully define our regularized NrC.

**My guess:** if your report goes over 4 pages, you are probably on the wrong track.

## Some Optima & Some Geometry

- 1 Assume  $d = 1$ . Also assume that we are in a situation where we know  $r_-$  is fixed to 1, so we only have to optimize for  $r_+$ . The only observations that we have for that  $+$  class are  $x_1 = -1$  and  $x_2 = 1$ .
  - a Draw the loss function as a function of  $r_+$  for all  $\lambda \in \{0, 1, 2, 3\}$ . Be precise. A rough sketch or artist impression is not enough.
  - b Derive for every of the four functions the minimizer and their minimum values. Also determine all points where the derivative equals 0.
- 2 Generally, what does the regularizer in Equation (1) actually try to enforce and what will, therefore, eventually happen to the representors if  $\lambda$  gets larger and larger (i.e., what is the limiting behavior of the two solution representors)?

**3** We now consider the setting in which *both* representors have to be determined through a minimization of the loss. Still,  $d = 1$ , so we have  $L : \mathbb{R}^2 \rightarrow \mathbb{R}$ .

- a** Describe in words (but still precise and accurate!) how the contour lines for the general function  $L$  typically look like when we are trying to find two 1-dimensional representors. Hint: the contours can be described as the concatenation of two basic geometric shapes.
- b** Let us consider a handful of data points. For the  $+$  class we have the same observations as in Exercise 1. For the  $-$  class we now have observations  $x_3 = 3$  and  $x_4 = -1$ . Clearly, if we set  $\lambda$  to 0, we would find as optimal solution  $(r_-, r_+) = (1, 0)$ . Assume we have a large enough  $\lambda$  as under Exercise 2: determine the exact solution  $(r_-, r_+)$  in that case.

## Some Programming & Some Experimenting

Through the course page you can find a two-class digit classification task in 64 dimensions, i.e.,  $d = 64$ . It is named `optdigitssubset` and consists of the pixel values of small  $8 \times 8$  images, which are ordered in  $N = 1125$  rows of 64 columns wide. The first 554 rows contain the values of  $8 \times 8$  images of zeros, while the remaining block of 571 rows contains the 64 pixel values of 571 ones. The actual feature vectors are obtained by running through the rows of the images of the digits from top to bottom, concatenating all 8 rows of 8 pixels into a 64-dimensional vector.

**4** Implement an optimizer for the NrC from Equation (1) and convince yourself that it indeed optimizes what it should optimize. You can use gradient descent or any other approach of your liking. You are even allowed to use optimization toolboxes and the like. You can either implement a general version of the NrC or one that is completely dedicated to the data set that is given. (Note, however, that the latter is probably not necessarily easier to implement and it is probably harder to debug.)

- a** Describe *in no more than 200 words* the main ingredients of and/or considerations behind your optimization routine. In particular, sketch the search strategy that you employ and explain how you decide when you have reached the sought-after minimum.
- b** Similar to all 64-dimensional samples in the data sets, you can restructure the 64-dimensional solution representors that you find into an  $8 \times 8$  images again. (For instance, in Matlab you can just use something like `reshape(m, [8 8])`.) Plot the two solution representor images that you find for  $\lambda = 0$  and plot the two solution images for large  $\lambda$  for which the solution does not change anymore with an even further increase.
- c** Draw regularization curves, where you plot estimates of the apparent and true error (y-axis) against the regularization parameter  $\lambda$  (x-axis). Consider a single training example per class and all  $\lambda \in \{0, \frac{1}{10}, 1, 10, 100, 1000\}$ . Make sure you repeat the experiment so to get somewhat smooth curves: 20 times may be enough, but 100 is probably better.