Intelligence Artificial and the Web,
Machine Learning Exam
MOSIG 2014-2015

Massih-Reza Amini (Part 1), Ahlame Douzal (Part 2)

Duration: 2 hours, Documents: authorized

Provide parts 1 and 2 in separate sheets

1 Perceptron Ranking

The Perceptron Ranking model (PRanking) proposed by (Crammer and Singer, 2002)[1] is an adaptation of the Perceptron algorithm for ranking problems. To each example $\mathbf{x}$ is associated a rank y within the set $\{1, \ldots, k\}$ indicating a preference over $\mathbf{x}$. The aim of the algorithm is to find a weight vector $\mathbf{w}$ and a set of k thresholds $\mathbf{b} = (b_1, \ldots, b_k)$ verifying $b_1 \leq \ldots \leq b_{k-1} \leq b_k$ in a way that the following rule reflects the best the desired preference y for a given $\mathbf{x}$[2]:

$$\hat{y} = \min_{r \in \{1,\ldots,k\}} \{r \mid \langle \mathbf{w}, \mathbf{x} \rangle - b_r < 0\}$$

Hence, $\hat{y}$ is the predicted index of the lowest threshold $b_{\hat{y}}$ verifying $\langle \mathbf{w}, \mathbf{x} \rangle < b_{\hat{y}}$ and we are looking to have $\hat{y} = y$.

1.1 the algorithm

As Perceptron, the PRanking algorithm finds these weights and thresholds in an on-line mode for every example for which the current weights and thresholds do not predict the right ranking.

1. Suppose that $(\mathbf{x}^{(t)}, y^{(t)})$ is a randomly chosen example and that $(\mathbf{w}^{(t)}, \mathbf{b}^{(t)})$ are the current weights and thresholds verifying $b_1^{(t)} \leq \ldots \leq b_{k-1}^{(t)} \leq b_k^{(t)}$.

[1] K. Crammer & Y. Singer. Pranking for Ranking (2002) *Advances in Neural Information Processing Systems 14*. pp. 641–647.

[2] In order that this minimum be well defined, we set $b_k = \infty$.

What would be the sign of $\langle \mathbf{w}^{(t)}, \mathbf{x}^{(t)} \rangle - b_r^{(t)}$ for $r = 1, \ldots, y^{(t)} - 1$ and for $r = y^{(t)}, \ldots, k-1$?

2. In order to express the previous inequalities by a single one, let introduce the following binary variables $z_1^{(t)}, \ldots, z_{k-1}^{(t)}$ such that :

$$\forall r, z_r^{(t)} = \begin{cases} +1, & \text{if } \left\langle \boldsymbol{w}^{(t)}, \mathbf{x}^{(t)} \right\rangle > b_r^{(t)} \\ -1, & \text{if } \left\langle \boldsymbol{w}^{(t)}, \mathbf{x}^{(t)} \right\rangle < b_r^{(t)} \end{cases}$$

What should be the largest rank r for which $z_r^{(t)} = +1$? In this case, why the following statement is always true :

$$\begin{gathered} \text{The rank of } \mathbf{x}^{(t)} \text{ is well predicted if and only if} \\ \forall r, z_r^{(t)} \left(\left\langle \boldsymbol{w}^{(t)}, \mathbf{x}^{(t)} \right\rangle - b_r^{(t)} \right) > 0 \end{gathered}$$

Hence, an example $(\mathbf{x}^{(t)}, y^{(t)})$, is mis-ranked if there exists a threshold $b_r^{(t)}$ for which the value of $\left\langle \boldsymbol{w}^{(t)}, \mathbf{x}^{(t)} \right\rangle$ is on the wrong side of $b_r^{(t)}$, i.e. $z_r^{(t)} \left(\left\langle \boldsymbol{w}^{(t)}, \mathbf{x}^{(t)} \right\rangle - b_r^{(t)} \right) \leq 0$. In this case the algorithm forces the values $\left\langle \boldsymbol{w}^{(t)}, \mathbf{x}^{(t)} \right\rangle$ and $b_r^{(t)}$ to go towards one another. $b_r^{(t)}$ is thus replaced by $b_r^{(t)} - z_r^{(t)}$ and the weight vector is modified using the following update rule:

$$\boldsymbol{w}^{(t+1)} = \boldsymbol{w}^{(t)} + \left(\sum_{r | z_r^{(t)} \left(\left\langle \boldsymbol{w}^{(t)}, \mathbf{x}^{(t)} \right\rangle - b_r^{(t)} \right) \leq 0} z_r^{(t)} \right) \mathbf{x}^{(t)}$$

As Perceptron, the weights and thresholds are not changed on examples for which the ranks are well predicted. An interesting result is that it can be shown that the on-line algorithm with the previous update rule convergences in the following situation.

Let $(\mathbf{x}^{(1)}, y^{(1)}), \ldots, (\mathbf{x}^{(T)}, y^{(T)})$ *be an input sequence of examples, such that all examples are within an hypersphere of radius* $R; \forall t, \|\mathbf{x}^{(t)}\| \leq R$. *Suppose that there exists a ranking rule* $\mathbf{v}^* = (\mathbf{w}^*, \mathbf{b}^*)$ *such that* $b_1^* \leq \ldots b_{k-1}^*$ *and* $\|\mathbf{w}^*\| = 1$ *that rank perfectly the examples with a given margin* $\rho = \min_{r,t} \left\{ z_r^{(t)} \left(\left\langle \boldsymbol{w}^*, \mathbf{x}^{(t)} \right\rangle - b_r^* \right) \right\} > 0$. *Further suppose that* $\mathbf{w}^{(1)} = \mathbf{0}$ *and* $b_1^{(1)} = 0, \ldots b_{k-1}^{(1)} = 0, b_k^{(1)} = \infty$

1.2 The proof of convergence

3. Suppose that at the t^{th} iteration, the chosen example $(\mathbf{x}^{(t)}, y^{(t)})$, is misranked. Let

$$\forall r, s_r^{(t)} = \begin{cases} z_r^{(t)}, & \text{if } \left\langle \boldsymbol{w}^{(t)}, \mathbf{x}^{(t)} \right\rangle - b_r^{(t)} \leq 0 \\ 0, & \text{otherwise.} \end{cases}$$

Then, show that we have :

$$\boldsymbol{w}^{(t+1)} = \boldsymbol{w}^{(t)} + \left(\sum_r s_r^{(t)} \right) \mathbf{x}^{(t)}$$
$$\forall r \in \{1, \ldots, k-1\}, b_r^{(t+1)} = b_r^{(t)} - s_r^{(t)}$$

And also :

$$\left\langle \boldsymbol{v}^*, \boldsymbol{v}^{(t+1)} \right\rangle = \left\langle \boldsymbol{v}^*, \boldsymbol{v}^{(t)} \right\rangle + \sum_{r=1}^{k-1} s_r^{(t)} \left(\left\langle \boldsymbol{w}^*, \mathbf{x}^{(t)} \right\rangle - b_r^* \right)$$

4. For the misranked example $(\mathbf{x}^{(t)}, y^{(t)})$, let $n^{(t)} = |\hat{y}^{(t)} - y^{(t)}|$ be the absolute error between the predicted and the true rank of the example $\mathbf{x}^{(t)}$. In this case, show that

$$\sum_{r=1}^{k-1} s_r^{(t)} \left(\left\langle \boldsymbol{w}^*, \mathbf{x}^{(t)} \right\rangle - b_r^* \right) \geq n^{(t)} \rho$$

5. using the fact that $\mathbf{w}^{(1)} = \mathbf{0}$ and $b_1^{(1)} = 0, \ldots b_{k-1}^{(1)} = 0, b_k^{(1)} = \infty$, prove the following :

$$a) \left\langle \boldsymbol{v}^*, \boldsymbol{v}^{(t+1)} \right\rangle \geq \left\langle \boldsymbol{v}^*, \boldsymbol{v}^{(t)} \right\rangle + n^{(t)} \rho$$
$$b) \left\langle \boldsymbol{v}^*, \boldsymbol{v}^{(T+1)} \right\rangle \geq \left(\sum_t n^{(t)} \right) \rho$$
$$c) \|\boldsymbol{v}^{(T+1)}\| \geq \left(\sum_t n^{(t)} \right) \rho \quad (1)$$

Using the fact that examples are within a hypersphere of radius R, we can also show that (**to not be proven**):

$$\|\boldsymbol{v}^{(T+1)}\|^2 \leq R^2 \sum_t (n^{(t)})^2 + \sum_t n^{(t)} \tag{2}$$

6. From equations (1) and (2) and using the fact that $n^{(t)} \leq k-1$ show that the total number of misrankings $\sum_t n^{(t)} = \sum_t |y^{(t)} - \hat{y^{(t)}}|$ is upper-bounded by

$$\sum_t n^{(t)} \leq \frac{R^2(k-1)+1}{\rho^2}$$

7. From the previous result, say why the algorithm converges?

2 Temporal metrics and large margin classifiers

2.1 Toward temporal metrics

1. Let x_i and x_j be two time series of the same length T, generated by a synchronized underlying process (i.e. without delays). Does the Euclidean distance $d_E(x_i, x_j)$ be equivalent to $DTW(x_i, x_j)$? Answer by true or false, and justify your answer.

2. Explain how the following algorithm, proposed by Salvador and Chan[3], performs to fasten the DTW.

Function FastDTW()

Input: *X* – a TimeSeries of length |*X*|
Y – a TimeSeries of length |*Y*|
radius – distance to search outside of the projected warp path from the previous resolution when refining the warp path

Output: 1) A min. distance warp path between *X* and *Y*
2) The warped path distance between *X* and *Y*

```
1  // The min size of the coarsest resolution.
2  Integer minTSsize = radius+2
3
4  IF (|X|≤minTSsize OR |Y|≤minTSsize)
5  {
6    // Base Case: for a very small time series run
7    //   the full DTW algorithm.
8    RETURN DTW(X, Y)
9  }
10 ELSE
11 {
12   // Recursive Case: Project the warp path from
13   //   a coarser resolution onto the current
14   //   current resolution. Run DTW only along
15   //   the projected path (and also 'radius' cells
16   //   from the projected path).
17   TimeSeries shrunkX = X.reduceByHalf()
18   TimeSeries shrunkY = Y.reduceByHalf()
19
20   WarpPath lowResPath =
21       FastDTW(shrunkX,shrunkY, radius)
22
23   SearchWindow window =
24       ExpandedResWindow(lowResPath, X, Y,
25                         radius)
26
27   RETURN DTW(X, Y, window)
28 }
```

Figure 7. The FastDTW algorithm.

Figure 1: Fast DTW algorithm

[3]Salvador, S., & Chan, P. (2007). Toward accurate dynamic time warping in linear time and space. *Intelligent Data Analysis*, 11(5), 561-580.

2.2 About SVM

Let us consider the two classes training data given in Figure 2 to be classified by an SVM. Answer the following questions.

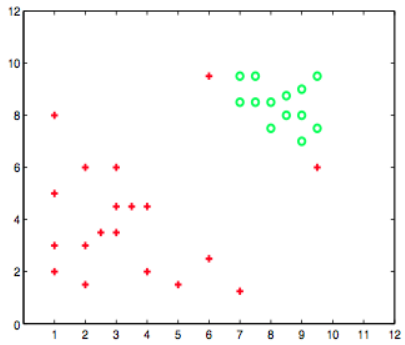

Figure 2: Binary classes

1. For an SVM trained with a Gaussian kernel, reproduce the figure above and draw, if possible, where would the decision boundary be for very large values of C (i.e., $C \rightarrow \infty$)? Justify your answer.

2. For an SVM trained with a Gaussian kernel, indicate by a circle those points that will not change the decision boundary learned for very large values of C. Justify your answer.

3. For an SVM trained with a Gaussian kernel, and for $C \approx 0$, indicate in the figure , where you would expect the decision boundary to be? Justify your answer.

4. For an SVM trained with a linear kernel, reproduce the figure above and draw, if possible, where would the decision boundary be for very large values of C (i.e., $C \rightarrow \infty$)? Justify your answer.