UNIVERSITAT POLITÈCNICA DE CATALUNYA

FACULTAT D'INFORMÀTICA DE BARCELONA

Extension Report: Dynamic Step

Analysis of the SVM-RFE algorithm for feature selection

Author:
Robert PLANAS

Director:
Luis A. BELANCHE

Bachelor Degree in Informatics Engineering
Specialization: Computing

March 31, 2021

Contents

Chapter 1

Dynamic Step

This modification is based on the constant step variant of the SVM-RFE algorithm, however, instead of using some constant number as the step in each iteration we calculate that number dynamically. The most straightforward way to do this is by using a percentage. Another possibility is to let the model score influence this percentage.

1.1 Description and reasoning

The percentage is a hyperparameter. It is used within every iteration to select a number of the first ranked features. A constant step has already been used in practice, but it is expected that this method will be significantly faster without effecting the accuracy performance, or even improving it.

The reasoning behind this is that when you have many features for which you only want to select a very small subset, the required amount of iterations will be of lineal complexity on the amount of features, however, by using a percentage, we get a logarithmic complexity.

Other similar modifications are also found in the literature, including using the square root of the remaining features `SQRT-RFE`, an entropy based function `E-RFE`, or `RFE-Annealing` which sets the step at $|\vec{s}|\frac{1}{i+1}$, thus, changing the percentage each iteration (Ding and Wilkins, 2006).

We assume that, the bigger the step in each iteration, the worse the performance of the final selection. However, since we're selecting the worst variables first, selecting more of them at once shouldn't affect performance because they would have been selected anyway with height probability in the following iterations. The fewer the iterations remaining, the riskier it becomes selecting multiple variables at once, and thus a smaller step is beneficial.

1.2 Pseudocode formalization

Definitions:

- $X_0 = [\vec{x}_0, \vec{x}_1, \ldots, \vec{x}_k]^T$ list of observations.
- $\vec{y} = [y_1, y_2, \ldots, y_k]^T$ list of labels.

Algorithm 1: SVM-RFE with DynamicStep

```
Input: $p$                                    // p = percentage, $0 \leq p \leq 1$
Output: $\vec{r}$
Data: $X_0, \vec{y}$
$\vec{s} = [1, 2, \ldots, n]$                 // subset of surviving features
$\vec{r} = []$                                // feature ranked list
while $|\vec{s}| > 0$ do
    /* Restrict training examples to good feature indices        */
    $X = X_0(:, \vec{s})$

    /* Train the classifier                                      */
    $\vec{\alpha}$ = SVM-train($X, y$)

    /* Compute the weight vector of dimension length $|\vec{s}|$      */
    $\vec{w} = \sum_k \vec{\alpha}_k \vec{y}_k \vec{x}_k$

    /* Compute the ranking criteria                              */
    $\vec{c} = [(w_i)^2$ for all $i]$

    /* Compute $t$ based on the percentage                         */
    $t = p|\vec{s}|$

    /* Find the $t$ features with the smallest ranking criterion   */
    $\vec{f}$ = argsort($\vec{c}$)( : $t$)

    /* Update the feature ranking list                           */
    $\vec{r} = [\vec{s}(\vec{f}), \ldots \vec{r}]$

    /* Eliminate the features with the $t$ smallest ranking
    criterion                                                    */
    $\vec{s} = [[\ldots\vec{s}(1 : f_i - 1), \ldots\vec{s}(f_i + 1 : |\vec{s}|)]$ for all $i]$
end
```

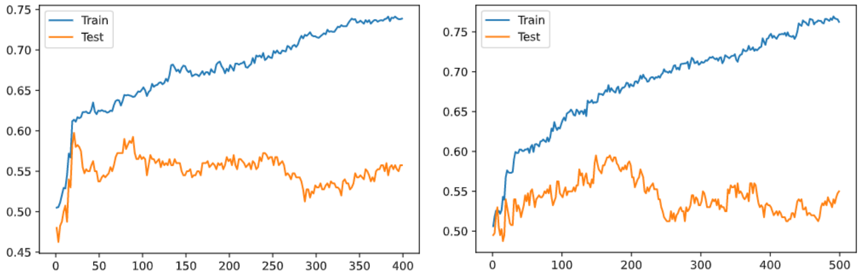
1.3 Results

1.3.1 Madelon Initial Analysis

First, we're using the MADELON dataset. We've initially run some tests to find how the hyper-parameters (C and step) may influence it.

The first test consists on simply doing a random feature selection and plotting the accuracy for the training and test splits. It is easy to see with figure 1.1 that the classifier is not working too well with test data, even though good results appear on training data. This is probably due to overfitting. Reducing the amount of dimensions, even randomly, results on a slightly improved performance with a peak at 200 features.

We tested different values for the regularization parameter, from 10^0 to 10^{12} in intervals of 1000 times each, but there is no indication that this has any major effect.



$C = 1$ $C = 10^3$

FIGURE 1.1: Accuracy of an SVM classifier with a random feature selection with the Madelon dataset.

Bibliography

Ding, Yuanyuan and Dawn Wilkins (Sept. 2006). “Improving the Performance of SVM-RFE to Select Genes in Microarray Data”. en. In: *BMC Bioinformatics* 7.2, S12. ISSN: 1471-2105. URL: https://doi.org/10.1186/1471-2105-7-S2-S12 (visited on 03/28/2021).