UNIVERSITAT POLITÈCNICA DE CATALUNYA

FACULTAT D'INFORMÀTICA DE BARCELONA

Extension Report: Sampling

Analysis of the SVM-RFE algorithm for feature selection

Author:
Robert PLANAS

Director:
Luis A. BELANCHE

Bachelor Degree in Informatics Engineering
Specialization: Computing

May 8, 2021

Contents

Chapter 1

Sampling

In this experiment we want to see if using a different subset of observations each iteration can reduce computational cost without negatively affecting performance.

1.1 Description and reasoning

Sampling is the process of selecting a representative part of a population. In machine learning sampling is done when creating a dataset, where each observation is a *sample* of some unknown distribution.

Sampling may also be done, a second time, when preparing the dataset. Classification algorithms often require that all classes have a similar number of observations, when this condition is not meet it is said that the dataset has *sampling bias*. To avoid sampling bias, the dataset is resampled with another method called *stratified random sampling*. This method consists on making a new dataset with only a subset of the observations, while imposing the restriction that all classes must have the same number of observations.

Because SVM computational cost is directly related to the amount of examples (i.e. samples), using only a subset of observations on each iteration can reduce this cost. In doing that, it is expected that performance will decrease, but we hope that in choosing a different subset each iteration, the performance will increase to levels similar to those we would see while using all data.

The reasoning for this last statement is that, although the algorithm only uses a subset of the data each iteration, by the time the algorithm finishes all data should've been used. Also, it is known that sampling improves generalization, thus we hope that this approx will reduce the gap between test and train error compared to the version with all data.

Further improvements may be introduced by reusing support vectors after the first iteration. Also, a dynamic sampling size may also be considered.

1.2 Pseudocode formalization

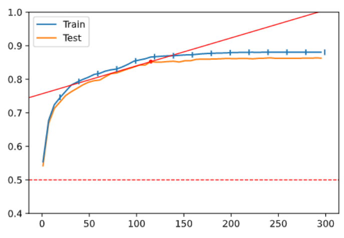
Definitions:

- $X_0 = [\vec{x_0}, \vec{x_1}, \ldots, \vec{x_k}]^T$ list of observations.
- $\vec{y} = [y_1, y_2, \ldots, y_k]^T$ list of labels.

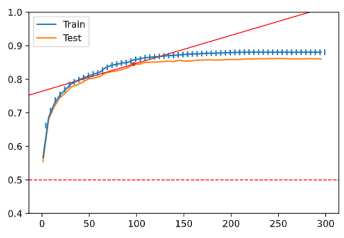
Algorithm 1: SVM-RFE with Random Sampling

```
   Input: $t, k$                // $t$ = step, $k$ = number of samples, $0 < k \leq |X_0|$
   Output: $\vec{r}$
   Data: $X_0, \vec{y}$
1  $\vec{s} = [1, 2, \ldots, n]$                     // subset of surviving features
2  $\vec{r} = []$                                          // feature ranked list
3  while $|\vec{s}| > 0$ do
       /* Determine subset of examples by random sampling              */
4      $idx = $ random_sample($|X_0|, k$)
       /* Restrict subset of examples to good feature indices          */
5      $X = X_0(idx, \vec{s})$
       /* Train the classifier                                         */
6      $\vec{\alpha} = $ SVM-train($X, y$)
       /* Compute the weight vector of dimension length $|\vec{s}|$    */
7      $\vec{w} = \sum_k \vec{\alpha}_k \vec{y}_k \vec{x}_k$
       /* Compute the ranking criteria                                 */
8      $\vec{c} = [(w_i)^2 \text{ for all } i]$
       /* Find the $t$ features with the smallest ranking criterion    */
9      $\vec{f} = $ argsort($\vec{c}$)( : $t$)
       /* Update the feature ranking list                              */
10     $\vec{r} = [\vec{s}(\vec{f}), \ldots \vec{r}]$
       /* Eliminate the features with the $t$ smallest ranking
       criterion                                                       */
11     $\vec{s} = [[\ldots \vec{s}(1 : f_i - 1), \ldots \vec{s}(f_i + 1 : |\vec{s}|)] \text{ for all } i]$
12 end
```

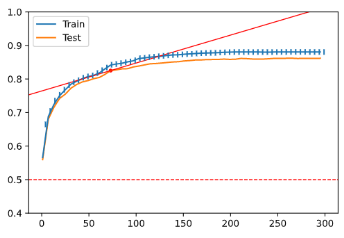
1.3 Results



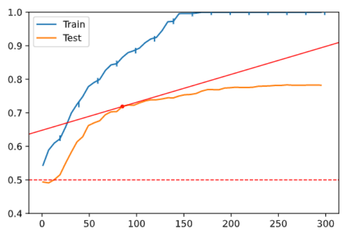
A1. $t = 20$



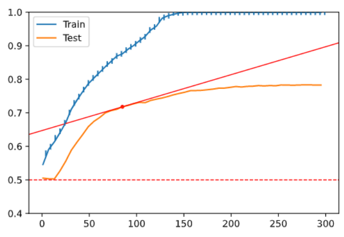
A2. $t = 5$



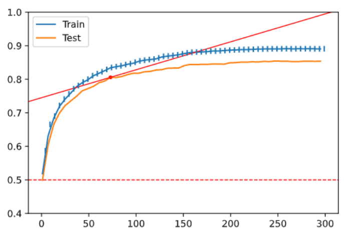
A3. $t = 5$



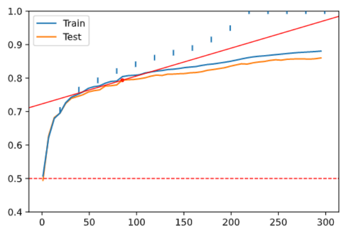
B1. $t = 20, k = 10\%$ PRE



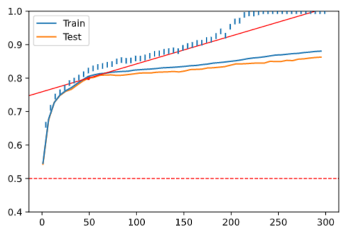
B2. $t = 5, k = 10\%$ PRE



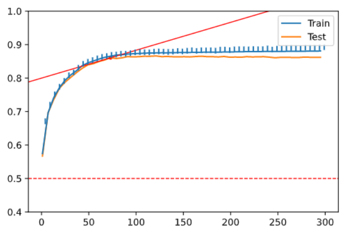
B3. $t = 5, k = 50\%$ PRE



C1. $t = 20, k = 10\%$



C2. $t = 5, k = 10\%$



C3. $t = 5, k = 50\%$

FIGURE 1.1: Grid comparing algorithms on each row **A-B-C** and the hyperparameter configuration for each **1** (step = 20), **2** (step = 5, sample = 10%) and **3** (step = 5, sample = 50%).

	1			**2**			**3**		
	Feat.	Acc.	Score	Feat.	Acc.	Score	Feat.	Acc.	Score
A	115	85.25%	0.194	97	84.51%	0.189	73	82.51%	0.189
B	85	71.88%	0.281	85	71.79%	0.282	73	80.58%	0.204
C	85	79.38%	0.221	49	79.99%	0.193	73	86.08%	0.160

TABLE 1.1: Perfomrance.

	1			**2**			**3**		
	Iter.	Time (s)	Var. (s^2)	Iter.	Time (s)	Var. (s^2)	Iter.	Time (s)	Var. (s^2)
A	15	07.82	0.26	60	30.35	7.68	60	30.87	2.21
B	15	01.46	0.16	60	05.29	1.86	60	16.26	4.70
C	15	03.18	2.14	60	13.43	5.51	60	19.64	1.75

TABLE 1.2: Times.

Accuracy trade-off is 80%. Problem is 8000 observations with 20-fold cross validation, 300 features, 100 informative.