Humboldt University of Berlin

Bootstrapping the SVM

Patricia Craja, Thomas Goerttler, Christian Koopmann

May 18, 2016

1 Problem Setting

1.1 Introduction

Support Vector Machines are one of the most successful methods of Machine Learning. By reducing non-linear complex decisions problems to linear problems through application of the Kernel-Trick, they represent a computationally efficient way to tackle these problems. SVMs based on certain kernels (e.g. Gaussian RBF Kernel) are non parametric methods. Since the distribution of the underlying data is generally unknown so is the finite sample distribution of these methods. There has been considerable research on the asymptotic distribution of SVMs, which have been shown to be asymptotically normally distributed under certain conditions. An alternative idea to estimate these distributions is using Efrons empirical bootstrap. The idea behind this method is to repeatedly draw samples with replacement from the full data according to the empirical distribution function of the data. Through the repeated calculation of the statistic of interest one can get an estimate of its distribution. For the SVM this estimate has been shown to be consistent under relatively mild conditions.

1.2 Goal

The goal of this project is to apply the bootstrap method to the SVM algorithm to estimate the uncertainty of its predictions and the way in which this uncertainty is correlated with other aspects of the SVM (in particular the number of support vectors). This will be done using both simulated as well as real datasets. On simulated datasets the resulting confidence intervals can be compared to known actual values, according to the distribution used for simulation, whereas in the case of real data it will be compared to asymptotic results. The project will be implemented in Python using the Liblinear-Algorithm for training individual SVMs, and applying this algorithm to different Bootstrap samples in a parallelized manner. The results will be summarized in a short paper.

2 Agenda

2.1 Overview

1. **Parallel SVM Training on Bootstrap Samples**

   Time budget: 40h

   Time frame: 11.05 - 07.06.

   Content: In this work package we will implement python code that will draw random samples of full size with replacement from our data and train an SVM on each of these samples. The training of SVMs will be parallelized (not each individual SVM though) to speed up the process.

2. **Calculation of Bootstrap Confidence Intervals**

   Time budget: 50h

   Time frame: 21.05 - 10.06.

   Content: Here we will develop code that will generate confidence intervals and variance measures based on the individual SVMs generated in Work Package 1

3. **Simulation Study**

   Time budget: 60h

   Time frame: 28.05. - 20.06.

   Content: In this section first we will design code that generates input variables and labels from a known distribution. Here we will ensure that the input variables vary in their influence on the outcome variable and are distributed so that it is fairly straightforward to determine the true distribution of the SVM. Afterwards we will apply our results from phase 1 and 2 to estimate the distribution and compare the estimate to the true distribution.

4. **Empirical Study**

   Time budget: 60h

   Time frame: 28.05. - 20.06.

   Content: This work package is somewhat similar to work package 3 with the difference that here we will use a real dataset. Here a significant challenge will lie in identifying a suitable dataset, since we will need data with a at least asymptotically known distribution. Also we will restrict our choice of data to applications where SVMs have high predictive performance.

5. **Empirical Study**

   Time budget: 60h

   Time frame: 28.05. - 20.06.

   Content: This work package is somewhat similar to work package 3 with the difference that here we will use a real dataset. Here a significant challenge will lie in identifying a suitable dataset, since we will need data with a at least asymptotically known distribution. Also we will restrict our choice of data to applications where SVMs have high predictive performance.