Solution of the CoIL Challenge 2000 Task Using Support Vector Machines

S.Sathiya Keerthi* and Chong Jin Ong†
Dept. of Mechanical & Production Engineering
National University of Singapore
10 Kentridge Crescent
Singapore 119260

Abstract

This document briefly describes details of our solution of the CoIL Challenge 2000 task using support vector machines (SVMs).

The task is a two category classification problem with 85 input variables. Each data vector of dimension 85 consists of information about one person. Broadly, the aim is to develop a predictor which uses knowledge about the 85 variables associated with a person to determine if he will take a caravan policy or not. The training data consists of 5822 training data vectors (for persons associated with these vectors it is known whether they took a caravan policy or not) and the test set consists of 4000 data vectors. The specific aim is to choose a subset of size 800 (20% of 4000) from the test set which has the most number of caravan policy takers.

All variables are discrete. We adopted the '1-of-m' representation for each input variable, i.e., if a variable took the integer values, 1, 2 and 3, then we formed a three dimensional vector for this variable with (1,0,0), (0,1,0) and (0,0,1) representing the values 1, 2 and 3 respectively. The SVM algorithm/code that we used is the one in Keerthi et al. 2000. The hyperparameters in the SVM formulation, such as the penalty parameter, the degree of the polynomial kernel, the width parameter in the gaussian kernel etc., were tuned using 10-fold CV (cross validation) to optimize the hitrate. (Given a cross validational sample and an SVM solution, the hitrate is

* *mpessk@guppy.mpe.nus.edu.sg*
† *mpeongcj@nus.edu.sg*

defined as the percentage of caravan policy holders in the top 20% of that sample, chosen using the SVM rating.)

Our first try was to apply the SVM algorithm on the data using all input variables. We started with the simple linear kernel and then tried polynomial kernels of varying degrees and gaussian kernels of varying widths. The best average CV hitrate that we achieved was around 16%. We tried increasing the weight given to the examples corresponding to caravan policy holders, but it only worsened the results somewhat. Next we decided to have a careful look at the variables and see if we can remove some irrelevant ones. Variables from 6 till 43 are derived from a sociodemographic database, e.g., zip code information. *So we decided to throw away these variables altogether.* (This will be assumed in the rest of the report.)

Simple statistics told us that variables 47 and 59 (contributions to car and fire policies) are very important. Let us make the following definition. If, in a data vector, the condition, *variable 47's value is not 6 and variable 59's value is neither 3 nor 4* holds, we will say that property P holds. We noted the following statistics.

- The set of persons whose data vector satisfies property P is a large percentage of the whole population (41.15%).
- Within this set of persons whose data vectors satisfy property P, the percentage of those taking a caravan policy is very small (1.8%). *Hence we decided to completely leave out all persons whose data vectors satisfy property P.*[1] This operation led to the removal of 1591 persons from the test set having 4000 persons.

After the above removal we found that persons having *any one* of the following (variable,value) pairs had a large (decent) percentage of caravan policies: (44,3), (45,3), (51,2), (57,3), (58,6), (60,1), (61,1-4), (62,1), (63,1), (64,2-4), (68,3), (76,3-4), (79,1), (81,1), (82,1-2), (83,1), (83,3) and (85,1). *So we decided to fully choose all such persons for our final select group.* (This will be assumed in the rest of the report.) In the training data there were 242 such persons (56 of whom had caravan policies) and, in the test data there were 171 such persons. (We expected that this part of the test set has about $171 \times 56/242 \approx 40$ persons taking caravan policies. Unfortunately, the test data set finally had only 25 caravan policy holders in this group.)

[1] If some time is spent analysing this omitted data, it is, of course, possible to recover some caravan policy holders. But we did not pursue that direction due to lack of time.

	Variables 1-5	Kernel Function	Overall Hitrate on Training Set	Overall Hitrate on Validation Set	Hitrate on Test Set
SVM-1	Used	Quadratic	20.94	18.26	13.25
SVM-2	Used	Cubic	23.21	18.24	13.38
SVM-3	Not Used	Quadratic	19.16	18.30	14.38
SVM-4	Not Used	Linear	18.26	18.13	14.13

Table 1: Description and performance of the best three methods

After the operations described above we used SVMs on the remaining data. Recall that the goal is to choose 800 members from the test set of size 4000. Since $1591 + 171 = 1762$ members were removed from the earlier statistical decisions the remaining problem is to choose $800 - 171 = 629$ persons from a group of size, $4000 - 1762 = 2238$. Hence the SVMs were tuned to optimize the average hitrate on the top 28.106% ($100 \times 629/2238$) of the CV samples. Again, 10-fold cross validation was used. If h is the (average CV) hitrate achieved then the overall (average CV) hitrate can be defined as

$$\text{Overall}\quad\text{hitrate} = \frac{40 + 629 \times h/100}{800} \times 100$$

This was done to compare the new method against our original raw SVM approach and also against other methods described by others.

We also had some doubt about the usefulness of variables from 1 till 5. So we designed several SVMs, using the first 5 variables in some and leaving them out in others. We also tried a number of choices for the kernel functions. Finally, four SVM designs looked promising. Their choices, together with the results achieved by them are described in Table 1.

The values, 13.25, 13.38, 14.38 and 14.13, of hitrate on the test set correspond to 106, 107, 115 and 113 caravan policy holders in the group of 800 members chosen by the SVMs from the test set of size 4000. With no knowledge of these numbers, we had to choose based on the overall hitrate on validation set. So, initially we submitted the solution given by SVM-3. (We would have been runners-up in the competition using this solution.) But then, we got a bit greedy. Since SVM-2 gives nearly the same value for overall hitrate on validation set while it gives a significantly better value for the overall hitrate on the training set, we felt SVM-2 was the better design. So, we finally recalled our first submission and sent in the solution given by SVM-2, which turned out to be a mistake. Though SVM-2 and SVM-3 gave very close average performance on the validation set, they gave quite a different performance on the test set; also, a look at the 10-fold CV details showed that on some validation samples SVM-2 did much better than SVM-3 while on some other validation samples SVM-2 did much worse than SVM-3. Obviously, SVM-3 had much better luck with the test set than SVM-2! Furthermore, the overall hitrate on the validation set and the hitrate on the test set are quite different. These facts indicate that the test set is not well correlated with the training set.

In future competitions we suggest that the organizers give several test datasets and evaluate groups on the average performance on those test datasets. For example, 4 test datasets, each comprising a random collection of 1000 members could have been given and each group asked to select the best 200 members from each test set.

Overall, we conclude that our methods using simple statistics-based ideas in combination with SVMs gave very good solutions, competitive with those given by the best groups in the competition. Post-competition analysis has revealed the fact that the choice of features is an important criterion. If even upto 4 or 5 variables had been carefully chosen, they can provide excellent results. If some extra effort had been put in this direction, we believe that SVMs could have given even better results.

References

CoIL 2000 homepage: *http://www.dcs.napier.ac.uk/coil/challenge/*

S.S. Keerthi, S.K. Shevade, C. Bhattacharyya and K.R.K. Murthy, A fast iterative nearest point algorithm for support vector machine classifier design, IEEE Trans. Neural Networks, Vol. 11, pp. 124-136, Jan. 2000. For a detailed technical report and a code, see: *http://guppy.mpe.nus.edu.sg/˜mpessk*