ESANN 2019 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 24-26 April 2019, i6doc.com publ., ISBN 978-287-587-065-0. Available from http://www.i6doc.com/en/.

A best-first branch-and-bound search for solving the transductive inference problem using support vector machines

Hygor Xavier Araújo, Raul Fonseca Neto and Saulo Moraes Villela

Department of Computer Science, Federal University of Juiz de Fora, Brazil

Abstract. In this paper, we present a new method for solving the transductive inference problem whose objective is predicting the binary labels of a subset of points of interest of an unknown decision function. We attempt to learn a decision boundary using SVM. To obtain the maximal-margin hypothesis over labeled and unlabeled samples, we employ an admissible best-first search based on margin values. Empirical evidence suggests that this globally optimal solution can obtain excellent results in the transduction problem. Due to the selection strategy used, the search algorithm explores only a small fraction of unlabeled samples making it efficiently applicable to median-sized datasets. We compare our results with the results obtained from the TSVM demonstrating better results in margin values.

1 Introduction

In many applications, the process of labeling samples on a dataset is very difficult, expensive or time-consuming, in some cases requiring the manual classification by an expert. In these cases, there is usually a small set of labeled data and a large number of unlabelled data. Semi-supervised learning emerges as a solution to this type of situation. In this type of learning, some labeled data (training set) is required for the construction of the model and, in addition, it is also possible to use unlabeled data (working set) to build a model. With this setting, it is expected that the acquired solution is better than would be possible with only the labeled or unlabeled data.

Therefore, we can resume the use of semi-supervised learning in classification problems as an attempt to improve the generalization capacity using simultaneously labeled and unlabeled data. Methods related to semi-supervised learning usually employ the cluster assumption: the decision boundary should lie in low-density regions [1]. Thus, it makes sense to use a large margin classifier, like the Support Vector Machines (SVM), to find a maximum margin separator hyperplane on the training and working sets. In this way, the transductive support vector machines [2] implements the cluster assumption directly by trying to find a decision surface which is far away from the labeled and unlabeled samples. We can say that transductive induction is a special case of semi-supervised learning if the transductive hypothesis is used to infer unseen samples.

However, finding the exact transductive SVM optimal solution or the best scheme of labels for the working set is a combinatorial NP-hard problem, becoming computationally prohibitive for datasets with a large number of unlabeled samples. Given a binary classification problem and a working set of size n we have 2^n possible labeling schemes.

To overcome this problem, we propose a best-first search that efficiently explores the space of all labeling schemes finding the maximal margin hypothesis. The algorithm employs as evaluation function the margin values, which is a monotone function since the margin values are monotonically decreasing when new points are inserted in the problem space, and thereby the search algorithm is admissible. We provide an extensive evaluation of the model performance making a set of transductive inference experiments. We compare our results with the results obtained from the Transductive Support Vector Machines (TSVM) proposed in [3] demonstrating better results in margin values.

Following this brief introduction, we present in Section 2 some related work. In Section 3 we describe preliminary concepts such as the binary classification problem and the semi-supervised learning task. Section 4 presents the proposed transduction algorithm and Section 5 reports the experiments and results. Finally, Section 6 presents the discussion and perspectives of future work.

2 Related work

In [4] the Semi-Supervised Support Vector Machine (S^{3}VM) is presented. It is shown that the SVM optimization problem can be modified to include the working set and transformed into a mixed-integer programming problem, which can be solved by integer programming methods. To make the problem easier to solve, the authors attempt to minimize the L_1 norm of the normal vector defining a robust linear programming model with binary variables. This method is practical only for solving small-sized problems.

The TSVM is presented in [3], which performs a local search by labeling the entire working set and then performing changes of the given label while there is an improvement on the objective function. Because it is not an exact method and uses a form of local search, it is designed to handle large-sized datasets.

Finally, [5] presents a formulation of S^{3}VM using the Branch-and-Bound technique for obtaining the globally optimal solution attempting to learn the low-density separator assumption. The method is very similar to our proposal, but it differs in the two main processes: branching and bounding, and is appropriate only for small-sized datasets. As a control strategy, this method uses a depth-first search and selects to branch the unlabeled sample which results in a big increase in the objective function. However, this requires the solution of several SVM problems, one to each unlabeled sample. As will be seen in Section 4, we implement alternatives strategies for these processes making the proposed model more efficient and applicable in median-sized datasets.

3 Preliminaries

3.1 Binary classification problem

Given a set of samples X of size m belonging to an input space $\mathbb{R}^d$ of dimension d with each sample x_i associated with a scalar $y_i \in Y$, we can define the training set of a classification problem as $Z = \{z_i = (x_i, y_i) \mid i \in \{1, \ldots, m\},\ x_i \in X \text{ and } y_i \in Y\}$. In a binary classification problem $y_i = -1$ or $+1$. The main goal in a classification problem is to find a function that generalizes from a set of data used for training. We can define the hyperplane by its normal vector $w \in \mathbb{R}^d$, also called the weight vector, and a constant $b \in \mathbb{R}$ called bias. This hyperplane has to separate the space such that $\{(x_i, y_i) \in Z \mid y_i = +1\}$ stays in one-half space separated by it and $\{(x_i, y_i) \in Z \mid y_i = -1\}$ in the other.

For a linearly separable training set we want to find (w, b) subject to $y_i(w \cdot x_i) \geq 0, \forall (x_i, y_i) \in Z$. One possible way to find this hyperplane is to use a large margin classifier. This class of algorithm is capable of defining a distance between the decision boundary and the samples. Its solution gives a hyperplane with the maximum distance between it and the nearest samples.

3.2 Semi-supervised learning and transduction

Semi-supervised learning can be considered to be in between supervised and unsupervised learning. The reason for this is because of its learning phase, where not only is a training set X_l used, with all the samples already labeled, but also a working set X_u of unlabeled samples. The goal of using both these sets is to have a better classifier than would be possible using only one of them.

We can define the training set X_l for semi-supervised learning as $X_l = \{(x_i, y_i) \mid i \in \{1, \ldots, m\}\}$ and the working set as $X_u = \{x_j \mid j \in \{1, \ldots, k\}\}$.

A learning algorithm can have as a result either an inductive or transductive function. More commonly we find algorithms with an inductive setting, which means that after its learning phase it is capable of outputting a function $f : \mathcal{X} \to y$ defined in all space $\mathcal{X}$. On the contrary, with a transductive setting, the result would be a function $f : X_u \to y_u$ that is only capable of labeling the samples from the working set.

For new samples in an inductive setting, we can use the resulting function to make predictions about the labels. In a transductive setting, it would be necessary to retrain the model including the new samples to the working set. On the other hand, the main idea of transductive learning follows the fact that if there is a limit with a restrict amount of information, do not solve the particular problem by solving a more general problem [2].

4 Transduction algorithm

4.1 State space and heuristic search

An efficient paradigm to deal with the combinatorial nature of the transduction inference problem is the heuristic search, where each problem hypothesis is represented by a state in the search state space. Among the main search methods, we can cite the best-first search that employs as selection strategy the choice of the best of all. However, this method requires an evaluation function in order to measure the merit of the states and the condition that this function is monotonically decreasing for solving maximization problems. Following the algorithm proposed in [6] we develop the Best-First Branch-and-Bound Transductive Classifier (BFBB-TC) algorithm coupled with a hard margin linear SVM.

The BFBB-TC algorithm uses the margin values from an SVM as an evaluation function that is monotonically decreasing, satisfying the admissibility property and ensuring the optimality of the search. Then, let γ^{m+1} be the real value of the maximal margin for a child state hypothesis and γ^{m} be the real value of the maximal margin for its parent's hypothesis. Thus, in a training set with $m + 1$ samples we have $\gamma^{m+1} \leq \gamma^{m}$. The generated states, ranked by the margin values, are stored in a priority queue, implemented as a heap structure.

4.2 Branching

The branching process can be explained as follows: take from the queue the current solution of superior value margin. Next, introduce in training space the unlabeled samples from the working set. If the new solution is feasible and does not force the margin, then the optimal solution was found. Otherwise, we have an unfeasible margin solution or error margin, and it is possible to update the lower bound by computing the margin value of the sample that is closest to the hyperplane. Notice that the feasible solution fulfills the margin constraint. Then, this sample is selected to be labeled and generates two new states S_{+} and S_{-} which must be inserted, after evaluation, in the queue.

4.3 Evaluation and pruning

The branching process produces two new training sets X_{l+} and X_{l-}, each one has the previous training set plus the selected sample with one of the labels. We run the SVM with X_{l+} and X_{l-} to obtain the new solutions with margins γ_{+} and γ_{-}. The parent's solution γ defines a new upper bound for this sample. If there is no solution, then the margin value will be negative, and the respective state must be eliminated and not inserted in the heap structure. Also, all the states whose margin value is smaller than the lower bound must be eliminated. Since we are always selecting the sample which is closer to the separating hyperplane, when the margin value is reduced, then this sample is a potential candidate to be a support vector in the final solution.

In this sense, the algorithm selects only a small fraction of the unlabeled samples. The monotonicity property of the margin values is proved considering the fact that the new maximal margin problem is more restricted than the parent's problem observing the fact that the addition of a new constraint reduces the hypothesis space. Therefore, the new solution shall be equal to or less than the parent's solution. Every time the lower bound is updated the states in the queue, with a margin value smaller than it, are removed from the search.

5 Computational experiments and results

In the experiments, we made a comparison between the BFBB-TC and the TSVM algorithm proposed in [3] using the SVM^{light}. The BFBB-TC uses as classifier the SMO algorithm [7] implemented in Scikit-Learn library. For both implementations, the hyper-parameters set were the regularization parameter C, with a value of 10000, and the kernel chosen was linear. For each of the four datasets selected from the benchmark created in [1], the experiments were made with working sets (WS) of sizes 50, 100, 200 and 300, except the BCI dataset due to its small size. These working sets were created from the original data, making ten random splits of it to select the samples for the training set and working set. The objective was to analyze the margin size and how much of the working set was explored on the solution. To run the experiments the only preprocessing made was to normalize the feature values in the range [-1, 1].

5.1 Results

Table 1 shows the mean values for the margin obtained with the ten executions of the TSVM and the BFBB-TC. The column "WS" indicates the working set size. The column "Not exp." indicates what percentage of the working set was not explored in the final solution of the BFBB-TC algorithm with the hyperplane that correctly separates the classes while considering the training and working sets. The column "%" indicates how much the BFBB-TC margin was greater than the TSVM margin. The best results are highlighted in bold. Table 1 also shows some information about the selected datasets.

Table 1: Comparison between BFBB-TC and TSVM.

Set	Dim.	Samples			WS	TSVM	BFBB-TC		
		Pos.	Neg.	Total		Margin	Margin	%	Not exp.
Digit1	241	734	766	1500	50	0.05249 ± 0.00259	**0.05391 ± 0.00155**	2.71%	91.60%
					100	0.05265 ± 0.00345	**0.05486 ± 0.00127**	4.20%	92.90%
					200	0.05559 ± 0.00224	**0.05794 ± 0.00211**	4.23%	91.55%
					300	0.05723 ± 0.00497	**0.06044 ± 0.00259**	5.61%	91.27%
USPS	241	1200	300	1500	50	0.01272 ± 0.00047	**0.01289 ± 0.00046**	1.30%	98.20%
					100	0.01502 ± 0.00113	**0.01529 ± 0.00107**	1.74%	96.70%
					200	0.01946 ± 0.00133	**0.01996 ± 0.00161**	2.53%	95.10%
					300	0.02452 ± 0.00237	**0.02513 ± 0.00247**	2.51%	93.23%
COIL$_2$	241	750	750	1500	50	0.00798 ± 0.00052	**0.00828 ± 0.00048**	3.75%	95.00%
					100	0.00832 ± 0.00057	**0.00864 ± 0.00040**	3.85%	93.20%
					200	0.00980 ± 0.00051	**0.01015 ± 0.00048**	3.56%	91.85%
					300	0.01092 ± 0.00089	**0.01135 ± 0.00076**	3.91%	91.37%
BCI	117	200	200	400	50	0.00627 ± 0.00076	**0.00646 ± 0.00087**	3.04%	90.80%
					100	0.00837 ± 0.00134	**0.00870 ± 0.00139**	3.90%	87.60%
					200	0.01645 ± 0.00387	**0.01853 ± 0.00477**	12.66%	84.95%

As shown in Table 1 the BFBB-TC algorithm achieved a larger margin in all cases, as expected. Given that a larger margin is achieved it is expected that the classifier will also have a better generalization. Although the TSVM is capable

ESANN 2019 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 24-26 April 2019, i6doc.com publ., ISBN 978-287-587-065-0. Available from http://www.i6doc.com/en/.

of finding a solution, even for larger datasets, it is not the optimal one.

A very important question related to this method is that only 1.8 to 15.05% of the working set was really necessary to find the solution in the experiments. With a working set of size n we would have 2^n possible labeling schemes to expand in total, but if we only need to expand at most 10% of that, we would have 2^k where $k = 0.1 \cdot n$. Taking this into consideration, it makes it possible to solve problems with larger working sets and may also indicate which maximum size of a working set could be used to solve a problem. Another interesting detail about this is that you would not need to know previously which samples of your working set are the most important, the algorithm will determine it accordingly to your training and working sets data distribution.

6 Discussion

In this work, we proposed the BFBB-TC algorithm which combines a best-first search strategy with the branch-and-bound technique and the SVM to find the optimal labeling scheme that solves the transduction problem. The results, as shown in Table 1, were very promising encouraging the continuity of the studies.

Considering the fact that the monotonicity property of the evaluation function is preserved in the feature space, as future work, we intend to develop the dual implementation of the model allowing the possibility of making the nonlinear transductive inference with the use of kernel functions. We also consider the possibility of changing the SVM by a large margin classifier implemented in an iterative setting. In this case, we can solve the optimization problem starting from a previous solution [8], which could improve the efficiency of the method.

References

[1] O. Chapelle, B. Schölkopf, and A. Zien, editors. *Semi-Supervised Learning.* MIT Press, Cambridge, MA, 2006.

[2] Vladimir N. Vapnik. *The Nature of Statistical Learning Theory.* Springer-Verlag, Berlin, Heidelberg, 1995.

[3] Thorsten Joachims. Transductive inference for text classification using support vector machines. In *Proceedings of the Sixteenth International Conference on Machine Learning*, pages 200–209, San Francisco, CA, USA, 1999. Morgan Kaufmann Publishers Inc.

[4] Kristin P. Bennett and Ayhan Demiriz. Semi-supervised support vector machines. In *Proceedings of the 1998 Conference on Advances in Neural Information Processing Systems II*, pages 368–374, Cambridge, MA, USA, 1999. MIT Press.

[5] Olivier Chapelle, Vikas Sindhwani, and S. S. Keerthi. Branch and bound for semi-supervised support vector machines. In B. Schölkopf, J. C. Platt, and T. Hoffman, editors, *Advances in Neural Information Processing Systems 19*, pages 217–224. MIT Press, 2007.

[6] Saulo Moraes Villela, Saul C. Leite, and Raul Fonseca Neto. Feature selection from microarray data via an ordered search with projected margin. In *Proceedings of the 24th International Conference on Artificial Intelligence*, pages 3874–3881. AAAI Press, 2015.

[7] John C. Platt. Fast training of support vector machines using sequential minimal optimization. In Bernhard Schölkopf, Christopher J. C. Burges, and Alexander J. Smola, editors, *Advances in Kernel Methods*, pages 185–208. MIT Press, Cambridge, MA, USA, 1999.

[8] Saulo Moraes Villela, Saul C. Leite, and Raul Fonseca Neto. Incremental p-margin algorithm for classification with arbitrary norm. *Pattern Recognition*, 55:261–272, 2016.