Local Application of One-Level Trees

D. Anyfantis, M. Karagiannopoulos, S. B. Kotsiantis, P. E. Pintelas

Educational Software Development Laboratory
Department of Mathematics, University of Patras, Hellas

Abstract

We propose a technique of local application of one-level decision and regression trees. We recognize local regions having similar characteristics and then build local expert on each of these regions describing the relationship between the data characteristics and the target value. We performed a comparison with other well known lazy methods on standard benchmark datasets and the proposed technique produced the most accurate results.

Key words: classifier, machine learning, data mining, regressor

1 Introduction

Instance-based (lazy) learners classify an instance by comparing it to a database of pre-classified examples. Local learning [1] can be understood as a general principle that allows extending learning techniques designed for simple models, to the case of complex data for which the model's assumptions would not necessarily hold globally, but can be thought as valid locally. In this paper, we propose a technique of local application of one-level decision and regression trees (decision stumps) [7]. We performed a comparison with other well known lazy methods on standard benchmark datasets and the proposed technique produced the most accurate results. In the next section, we describe the proposed method and we evaluate the proposed method on several UCI datasets by comparing it with other lazy methods. Finally, section 4 concludes the paper and suggests further directions.

Email address: {dany, mariosk, sotos, pintelas}@math.upatras.gr (D. Anyfantis, M. Karagiannopoulos, S. B. Kotsiantis, P. E. Pintelas).

2 Proposed Algorithm and Experiments

Local methods have significant advantages when the probability measure defined on the space of symbolic objects is very complex, but can still be described by a collection of less complex local approximations. Some theoretical results and experimental results [4], [9] indicate that a local learning algorithm provides a feasible solution to this problem. The proposed algorithm builds a model for each point to be estimated, taking into account only a subset of the training points. This subset is chosen on the basis of the preferable distance metric between the testing point and the training point in the input space. For each testing point, a decision stump learner is thus learned using only the training points lying close to the current testing point. Generally, the proposed method consists of the four steps (see Fig. 1).

1. Determine a suitable distance metric.
2. Find the k nearest neighbors using the selected distance metric.
3. Apply the DS algorithm using as training instances the k instances.
4. The 5answer of the model is the prediction for the testing instance.

Fig. 1. Local Decision Stump

In our experiments, we used the most well known -Euclidean similarity function- as distance metric. The proposed algorithm also requires choosing the value of K. In the current implementation we decided to use a fixed value for K (=50): a) in order to keep the training time low and b) about this size of instances is appropriate for a simple algorithm, to build a precise model according to [6], [8]. We have experimented with a number of classification datasets from the UCI repository [2]. In order to calculate the classifiers' accuracy, cross validation was run 10 times for each algorithm and the average value was calculated. It must be mentioned that we used the free available source code for most of the algorithms by [10] for our experiments. We compare the proposed methodology with K-nearest neighbors using k=3 (most common used number of neighbors), as well as k=50 because the proposed algorithm uses 50 neighbors. In addition, we tested Kstar: another instance-based learner which uses entropy as distance measure [5]. In following Tables, we represent as "v" that the specific algorithm performed statistically better than the proposed method according to t-test with $p < 0.05$. On the other hand, "*" indicates that the proposed method performed statistically better than the specific algorithm according to t-test with $p < 0.05$. In all the other cases, there is no significant statistical difference between the results (Draws). As one can see, the performance of the presented method is more accurate than the other techniques. Subsequently, we experimented with a number of datasets from the UCI repository [2]. We compared the proposed methodology with Simple

DS algorithm, K-nearest neighbors using k=50 because the proposed algorithm uses 50 neighbors. In addition, we tested Kstar: another instance-based learner which uses entropy as distance measure [5]. Similarly, in order to calculate the models' correlation coefficient for our experiments, cross validation was run 10 times for each algorithm and the average value was calculated. The performance of the proposed method is better than the other tested techniques.

Table I. Comparing local decision stumps with other learners

Datasets	Local DS	Kstar	3NN	DS	50NN
audiology	72.68	80.32 v	67.97 *	46.46 *	35.95 *
autos	74.82	72.01*	67.23 *	44.9 *	48.18 *
colic	80.87	75.71*	80.95	81.52	84.04 v
credit-rating	83.61	79.1 *	84.96	85.51	86.16 v
Glass	70.58	75.31 v	70.02	44.89 *	56.16 *
heart-c	78.29	75.18 *	81.82 v	72.93 *	81.58 v
ionosphere	88.24	84.64 *	86.02 *	82.57 *	71.65 *
Iris	94	94.67	95.2	66.67 *	90.53 *
monk3	93.44	86.22 *	86.72 *	76.01 *	82.46 *
Vehicle	69.58	70.22 *	70.21 *	39.81 *	63.47 *
Vote	95.4	93.22 *	93.08 *	95.63 *	90.41 *

Table II. Comparing the Algorithms

Dataset	Local DS	Kstar		50NN		DS	
servo	0.89	0.86	*	0.65	*	0.79	*
autoHorse	0.92	0.90		0.85	*	0.72	*
autoMpg	0.89	0.91		0.86	*	0.74	*
bodyfat	0.94	0.87	*	0.91	*	0.82	*
cholesterol	0.12	0.04	*	0.17	v	0.04	*
fishcatch	0.94	0.99	v	0.78	*	0.83	*
housing	0.84	0.90	v	0.77	*	0.60	*

lowbwt	0.78	0.62	*	0.75	*	0.78	
pbc	0.43	0.30	*	0.52	v	0.43	
pwLinear	0.84	0.72	*	0.85		0.68	*
quake	0.09	0.08		0.06	*	0.09	
sensory	0.47	0.39	*	0.36	*	0.29	*
auto93	0.72	0.77	v	0.71		0.59	*

3 Conclusion

Our experiment in real datasets shows that the proposed method outperforms other lazy classification and regression methods. In a following work we will focus on the problem of reducing the size of the stored set of instances [3] while trying to maintain or even improve generalization accuracy by avoiding noise and overfitting.

References

[1] C.G. Atkeson, A.W. Moore & S. Schaal, Locally weighted learning. Artificial Intelligence Review, 11(1-5), 11-73, 1997.

[2] C. Blake & C. Merz, UCI Repository of machine learning databases. Irvine, CA: University of California, Department of Information and Computer Science. http://www.ics.uci.edu/ mlearn/MLRepository.html

[3] H. Brighton, C. Mellish, Advances in Instance Selection for Instance-Based Learning Algorithms, Data Mining and KDD, 6, 153-172, 2002.

[4] L. Bottou and V. Vapnik, Local learning algorithm, Neural Computation, vol. 4, no. 6, pp. 888-901, 1992.

[5] C. John and L. Trigg, K*: An Instance- based Learner Using an Entropic Distance Measure", Proc. of the 12th Inter. Conf. on ML, pp. 108-114, 1995.

[6] E. Frank, M. Hall and B. Pfahringer, Locally weighted naive Bayes. Proc. of the 19th Conf. on Uncertainty in AI. Acapulco, Mexico. Morgan Kaufmann, 2003.

[7] W. Iba, & P. Langley, Induction of one-level decision trees. Proc. of the Ninth Intern. ML Conf (1992). Aberdeen, Scotland: Morgan Kaufmann.

[8] Loader, C., Local Regression and Likelihood. Springer, New York, (1999).

[9] V.N. Vapnik, Statistical Learning Theory, Wiley, New York, 1998.

[10] Ian H. Witten and Eibe Frank, Data Mining: Practical machine learning tools and techniques, 2nd Edition, Morgan Kaufmann, San Francisco, 2005.