An Evolutionary Approach to Constructive Induction for Link Discovery

Tim Weninger, William H. Hsu, Jing Xia, Waleed Aljandal
234 Nichols Hall
Kansas State University
Manhattan, KS 66506
{weninger, bhsu, xiajing, waleed}@ksu.edu

ABSTRACT

This paper presents a genetic programming-based symbolic regression approach to the construction of relational features in link analysis applications. Specifically, we consider the problems of predicting, classifying and annotating friends relations in friends networks, based upon features constructed from network structure and user profile data. We explain how the problem of classifying a user pair in a social network, as directly connected or not, poses the problem of selecting and constructing relevant features. We use genetic programming to construct features, represented by multiple symbol trees with base features as their leaves. In this manner, the genetic program selects and constructs features that may not have been originally considered, but possess better predictive properties than the base features. Finally, we present classification results and compare these results with those of the control and similar approaches.

General Terms

Design, Experimentation

Categories and Subject Descriptors

I.2.6 [**Computing Methodologies**]: Learning; I.1.1 [**Computing Methodologies**]: Expressions and Their Representation—*symbolic regression*

Keywords

machine learning, classification, genetic programming

1. INTRODUCTION

We present a genetic programming (GP) approach to the construction of features in order to improve link discovery, *i.e.*, predicting the existence of links between objects. In order to discover links not previously known to exist, a GP is used to construct features that appropriately leverage the knowledge contained in the presence or absence of links. With this approach we show a statistically significant improvement in the AUC of most classifiers. We refer readers unfamiliar with one or more of the following topics: constructive induction, social networks, and link mining, to [8, 6, 2, 7, 1].

GECCO'09, July 8–12, 2009, Montréal Québec, Canada.
ACM 978-1-60558-505-5/09/07.

2. METHODOLOGY

Our approach for link mining constructive induction using genetic programming was performed in four distinct parts. First, we crawled the social network site *LiveJournal* and retrieved 39,024 users with 2,992,607 directed links (friendships). Second, we extracted features in the same manner as in [3]. Third, we generated, at random, three sets of candidate pairs: 2000 for training, 2000 for testing, and 2000 for validation. The ratio of positive to negative examples in our dataset is 1.5%, therefore the training dataset was forced to into a 50/50 positive/negative distribution, while the testing and validation datasets contained examples of the original distribution [5]. Finally, five set operators (U, V, $(U \cap V)$, $(U \cup V)$, $(U \setminus V)$), five statistical operations (*sum*, *mean*, *min*, *max*, *count*), and four mathematical operators ($+$, $-$, $\times$, $\div$) are used as GP-nodes to evolve the original features into new synthetic features. The fitness function was the inverse of WEKA [9] implementations of OneR, J48, IB1, Logistic and NaiveBayes classification AUC scores trained with the GP-generated features. We compare the GP results with results from the same classifiers and meta-learning algorithms (bagging, boosting, random forests) trained on only base-features.

3. RESULTS

In each experiment the total number of constructed features was exactly 10, the genetic program's population was exactly 100 individuals, the number of generations was set to 50, and the probability of mutation, crossover, etc. were set to the ECJ defaults [4]. As explained in earlier sections, the learning algorithm was trained with 2000 examples of a 50/50 distribution and tested on an independent set of 2000 examples of the original distribution; because of this "wrapper" approach, the test examples influenced the learning algorithm, therefore another independent validation set of 2000 examples of the original distribution was used to score the final performance of each algorithm. The scores reported in this section are from the holdout validation data. The tests were repeated 100 times so the resulting fitness scores were averaged with other scores of the same generation.

The average fitness scores for the entire population were averaged with other scores of the same generation. Figure 1 shows the mean fitness progression for all populations in each generation. Almost immediately the populations began to converge. The fitness scores (1-AUC) have been inverted in order to display the results more clearly, that is, the higher the score the better the fitness.

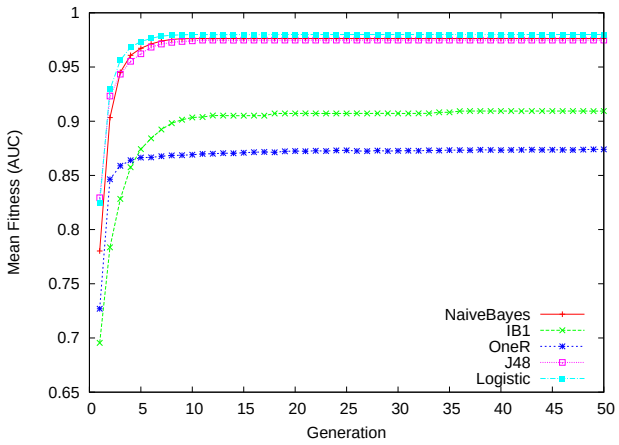

Figure 1: Inverse fitness scores of the population per generation, averaged among 100 repetitions.

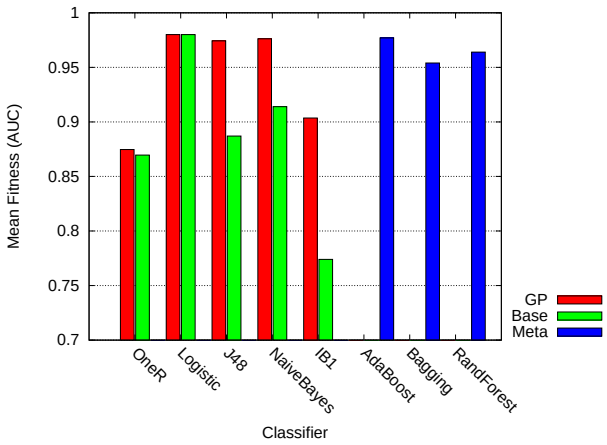

Figure 2: Comparison of AUC scores for different classifiers

A two-tailed paired T-test was performed in order to gauge the statistical significance of these results in comparison with results of the non-GP classifiers (control). The test list was comprised of the inverse fitness (AUC) scores of the best individual for each of the 100 repetitions. The control list was a list of the base scores, wherein the base score was enumerated 100 times. Table 1 shows the results of the T-test for each classifier. The results of all classifiers, except Logistic, were shown to have performed significantly better than the control.

Table 1: Test of statistical significance using two-tailed paired T-test

Learning algorithm	Mean	Variance	Paired t-test
OneR	87.47	0.03	2.03×10^{-5}
J48	97.43	0.01	3.60×10^{-92}
IB1	90.36	0.03	7.23×10^{-86}
Logistic	98.0	5.55×10^{-5}	0.9844
NaiveBayes	97.62	5.60×10^{-5}	2.07×10^{-93}

A final comparison of the results is shown in Figure 2.

4. CONCLUSIONS

We considered the problem of using a GP to enhance a learning algorithm's ability to train a classifier. We have identified the need to construct and select features based on information from the friends network. To that end, we employed a genetic program capable of evolving new features from primitive features. Finally, we show that classifiers that have been constructed by the GP perform significantly better than classifiers constructed in lieu of the GP. Moreover, GP-evolved classifiers generally performed better than meta-learning techniques although tests for significance were not attempted.

One limitation of this work is the expressiveness of the constructed features. One avenue for future research is to allow the GP more degrees of freedom in its construction of features. This could be done by adjusting the evolution parameters in system, or by using alternative fitness measures. This is likely a very difficult task due to the type differences inherent in set, statistical and mathematical operations.

5. ACKNOWLEDGEMENTS

This research was partially funded by a grant from the Defense Intelligence Agency.

6. REFERENCES

[1] L. Getoor. Link mining: A new data mining challenge. *SIGKDD Explorations*, 4(2):1–6, 2003.

[2] J. Han and M. Kamber. *Data Mining: Concepts and Techniques*. Morgan Kaufmann, 2nd edition, 2006.

[3] W. H. Hsu, A. King, M. S. Paradesi, T. Pydimarri, and T. Weninger. Structural link analysis from user profiles and friends networks: A feature construction approach. In *International Conference on Weblogs and Social Media (ICWSM)*, pages 75–80, Boulder, CO, 2007.

[4] S. Luke. *An EC and GP system in Java*, 2001. `http://www.cs.umd.edu/projects/plus/ec/ecj`.

[5] R. H. M. Kubat and S. Matwin. Learning when negative examples abound. In *ECML, Lecture Notes in Artificial Intelligence*, pages 146–153. Springer Verlag, 1997.

[6] T. M. Mitchell. *Machine Learning*. McGraw Hill, 1997.

[7] S. Wasserman and K. Faust. *Social Network Analysis: Methods and Applications*. Cambridge University Press, Cambridge, NY, 2007.

[8] T. Weninger. Link discovery in very large graphs by constructive induction using genetic programing. Master's thesis, Kansas State University, Manhattan, KS, USA, December 2008.

[9] I. H. Witten and E. Frank. *Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations*. Morgan Kaufmann, San Francisco, CA, 1999.