Dimensionality Reduction using Symbolic Regression

Extended Abstract

Ilknur Icke[1]
[1] The Graduate Center
City University of New York
365 Fifth Avenue
New York, NY 10016
iicke@gc.cuny.edu

Andrew Rosenberg[2,1]
[2] Queens College
City University of New York
65-30 Kissena Blvd.
Flushing, NY 11367−1575
andrew@cs.qc.cuny.edu

ABSTRACT

In this paper, we propose a symbolic regression approach for data visualization that is suited for classification tasks. Our algorithm seeks a visually and semantically interpretable lower dimensional representation of the given dataset that would increase classifier accuracy as well. This simultaneous identification of easily interpretable dimensionality reduction and improved classification accuracy relieves the user of the burden of experimenting with the many combinations of classification and dimensionality reduction techniques.

Categories and Subject Descriptors

I.2.8 [**Artificial Intelligence**]: Problem Solving, Search

General Terms

Algorithms

Keywords

symbolic regression, dimensionality reduction, classification

1. INTRODUCTION

In exploratory data analysis settings the dimensionality of the data causes various problems. First, it is not possible to visually inspect the dataset if it has more than three dimensions. Second, due to the *curse of dimensionality*, the number of samples needed for a classification task increases greatly as the number of dimensions increases. To address these two problems, various techniques have been introduced in order to reduce the dimensionality of the data before applying classification algorithms. In this paper we are proposing a symbolic regression based approach that performs dimensionality reduction for data visualization using classifier accuracy as the fitness function. Our algorithm evolves a visually and semantically interpretable transformation of the higher dimensional dataset into a lower dimensional representation such that the classifier accuracy would be improved. Genetic programming (GP) has been used for data classification tasks in various ways. One approach is to evolve the classifier directly, such as a program that would contain rules similar to a decision tree [3]. Another approach is to use genetic programming as a feature selection and/or extraction mechanism coupled with a classifier [7, 5]. Our approach is similar to the second group of techniques. This user-focused approach has the goal of simultaneously identifying the most interpretable data representation and the optimal classifier.

Section 2 describes our approach. Section 3 describes the experiments performed and the results are presented in Section 4. Conclusions are discussed in Section 5.

GECCO'10, July 7–11, 2010, Portland, Oregon, USA.
ACM 978-1-4503-0073-5/10/07.

2. SYMBOLIC REGRESSION APPROACH

Our symbolic regression (SR) based approach to dimensionality reduction task is an exploratory algorithm that aims to satisfy the following three important criteria simultaneously: 1) *classification accuracy*, 2) *visual interpretability*, 3) *semantic interpretability – the complexity of the dimensions.*

We explicitly optimize the classification accuracy by using cross-validation accuracy as the fitness measure to the symbolic regression algorithm. By limiting the dimensionality reduction to 2 or 3 dimensions, we ensure that we give users easy visualization of the data. This is a highly desirable property since data visualization is an incredibly useful tool for exploratory data analysis. By semantic interpretability, we refer to the complexity of the projection from an initial feature set to the reduced dimensionality. The symbolic expressions evolved by the algorithm provide relatively clear correlates to the initial feature representation.

Our algorithm aims to maintain visual interpretability by searching for those 2 or 3 dimensional representations of the data that achieve highest classifier accuracy. Each dimension is a symbolic expression representing a 1D projection of the data. In order to maintain semantic interpretability, we introduce a size limit on these symbolic expressions in order to keep the dimension concise and easily understood.

3. EXPERIMENTS

We conducted our experiments using a custom toolkit named Flubber, that utilizes ECJ [6] genetic programming and Weka [4] machine learning packages. In this paper we report our experiments on two datasets: Wisconsin Breast Cancer (9 features, 683 samples, 2 classes) [1] and Leptograpsus crabs (5 features, 200 samples, 4 classes) [2].

Symbolic regression by definition, finds a function that maps the multidimensional input data into a single value. We cast the dimensionality reduction task as a symbolic regression problem where the goal is to simultaneously evolve 2 (or 3) functions that map the input dataset into a lower

dimensional representation. Each expression is made up of $+, -, *, \%$ operators over the initial features and represents a 1D projection of the data. When guiding the symbolic regression algorithm we use cross-fold classifier accuracy as a fitness measure.

We experiment with the following classifiers: Naive Bayes, Logistic, SMO (support vector machine), Multilayer PerceptronCS (neural network), RBF Network, IBk (k-Nearest Neighbors), Simple Cart and J48 (decision tree). We examine three ways to compute the fitness of each individual: 1) maximum, 2) minimum and 3) average accuracy achieved by any classifier. We use 10-fold stratified cross-validation scheme to generate these fitness values.

For comparison, we generate 2D and 3D representations of the datasets using three widely utilized dimensionality reduction techniques: principal components analysis (PCA), multidimensional scaling (MDS) and random projections (RP). We also report the 10-fold cross-validation performance of each classifier on these lower dimensional representations of the data as well as the original dataset.

4. RESULTS

Every classifier we examined demonstrates high accuracy on the full Wisconsin dataset and none of the dimensionality reduction methods show a significant improvement. The best combination of dimensionality reduction and classification technique was the 2D MDS and J48 pair but MDS does not produce interpretable expressions for the dimensions.

Classifier	PCA (2D)	PCA (3D)	MDS (2D)	MDS (3D)	RP (2D)	RP (3D)	All features
N. Bayes	96.78	96.63	97.07	96.78	95.75	95.75	96.34
Logistic	96.63	96.93	97.07	97.22	95.17	94.88	96.78
SMO	96.78	96.63	97.07	96.63	95.9	95.75	**97.07**
MLP	97.36	96.93	96.78	96.93	96.34	96.05	96.05
RBF	96.34	96.49	96.63	96.05	95.31	94.73	95.75
kNN	95.31	96.05	96.49	95.61	94.29	93.7	95.75
CART	96.78	96.34	97.22	97.22	95.17	94.88	95.17
J48	97.22	97.22	**97.51**	97.07	95.02	94.88	96.05
Avg(std)	96.65(0.6)	96.65(0.4)	96.98(0.3)	96.69(0.6)	95.37(0.6)	95.08(0.8)	96.12(0.6)

Table 1: Accuracy(%) on Wisconsin dataset

Table 2 shows the mean and standard deviation of the classifier accuracy for the best SR solutions over 30 runs of 10 generations with a population size of 20 and using the three fitness functions described in Section 3. The best performing classifier in each case is marked in bold.

The highest classifier accuracy on this dataset was reached by the MLP (%98.1) on the following 3D transformation of the data found by the SR algorithm: f_1=V3+(V1+V4) * (V6+V9), f_2=(V6-V5) - (V9 / ((V9+V7-V3-V8) * V7)) / V5, f_3=((V1+V4) * (V6+V9) * (V3*V7)*(V9+V8))

Classifier	All features	Fitness:Maximum SR (2D)	Fitness:Maximum SR (3D)	Fitness:Minimum SR (2D)	Fitness:Minimum SR (3D)	Fitness:Average SR (2D)	Fitness:Average SR (3D)
N. Bayes	96.34	95.51(1.64)	95.66(1.16)	96.19(0.53)	96.35(0.57)	96.39(0.53)	96.52(0.64)
Logistic	96.78	95.48(1.48)	95.82(0.7)	95.88(0.46)	96.15(0.53)	96.02(0.48)	96.27(0.58)
SMO	**97.07**	94.42(3.09)	94.89(1.99)	95.83(0.43)	96.64(0.5)	95.86(0.52)	96.32(0.57)
MLP	96.05	**96.70(0.9)**	**96.84(1.33)**	**96.6(0.5)**	**96.23(0.53)**	96.82(0.40)	**96.90(0.4)**
RBF	95.75	95.72(1.42)	95.73(1.38)	96.17(0.55)	96.07(0.6)	96.57(0.5)	96.50(0.46)
kNN	95.75	95.53(0.74)	95.43(0.75)	95.78(0.63)	96.39(0.54)	95.92(0.59)	96.12(0.69)
CART	95.17	96.35(0.81)	96.51(0.70)	96.27(0.49)	96.39(0.54)	96.99(0.42)	96.63(0.45)
J48	96.05	96.73(0.84)	96.55(0.7)	96.24(0.5)	96.33(0.6)	**97(0.5)**	96.72 (0.41)
Avg(std)	96.12(0.6)	95.83(1.70)	95.93(1.32)	96.12(0.58)	96.3(0.56)	96.45(0.66)	96.50(0.58)

Table 2: SR results on Wisconsin dataset (30 runs)

The Crabs dataset represents a more confusable class of data. Although 3D representations using the PCA and MDS algorithms resulted improvement over the average accuracy, neither of the dimensionality reduction techniques were able to improve the classifier accuracy over the initial features.

Table 4 shows results of the symbolic regression approach. The 2D results are much better compared to the 2D results using PCA, MDS and RP.

Classifier	PCA (2D)	PCA (3D)	MDS (2D)	MDS (3D)	RP (2D)	RP (3D)	All features
N. Bayes	57.5	92	67	94	41	38	38
Logistic	59.5	94.5	63	94	65	92	**96.5**
SMO	54.5	91.5	59	91.5	47	50	63.5
MLP	62	**96**	67.5	95.5	61	94	**96.5**
RBF	67	95.5	69	93	49	50	49
kNN	57	95	67.5	93.5	50	70	89.5
CART	57.5	93	61	90.5	53	57.5	75.5
J48	56.5	93.5	59	92	51	59	73.5
Avg(std)	58.94(3.9)	93.88(1.6)	64.12(4.1)	93(1.6)	52.12(7.7)	63.81(20.2)	72.75(21.6)

Table 3: Accuracy(%) on Crabs dataset

In 3D case, the best dimensionality reduction/classifier pair was the maximum fitness and MLP classifier which performed better than the PCA and MDS on the average. The highest classifier accuracy on this dataset was reached by the MLP(%99) and on the following 3D transformation of the data found by the SR algorithm:

f_1=FL, f_2=RW / ((CL+BD) - (CW-FL)), f_3=CL / RW

Classifier	All features	Fitness:Maximum SR (2D)	Fitness:Maximum SR (3D)	Fitness:Minimum SR (2D)	Fitness:Minimum SR (3D)	Fitness:Average SR (2D)	Fitness:Average SR (3D)
N. Bayes	38	75.8(18.43)	65.07(21.89)	84.75(8.85)	87.53(7.46)	84.65(7.08)	86.45(9.47)
Logistic	**96.5**	86.75(7.09)	93.87(5.76)	**87.07(6.60)**	91.83(4.95)	**86.95(6.48)**	93.33(2.01)
SMO	63.5	77.02(16.53)	74.48(18.16)	84.92(7.83)	89.43(6.25)	84.13(8.19)	90.25(6.67)
MLP	**96.5**	**86.37(6.45)**	**96.72(1.81)**	86.18(7.37)	**92(5.27)**	86.8(6.80)	**94.25(2.42)**
RBF	49	82.13(11.64)	77.18(15.67)	85.55(7.58)	89.45(6.8)	85.78(7.02)	90.33(6.71)
kNN	89.5	80.92(10.09)	88.22(5.63)	83.8(8.27)	89.6(6.77)	83.11(8.70)	91.17(3.38)
CART	75.5	79.27(11.27)	77.33(10.46)	84.6(8.25)	87.05(5.72)	84.03(7.25)	86.17(5.9)
J48	73.5	79.65(11.48)	77.9(10.53)	84.92(7.43)	87.43(6.3)	84.07(7.34)	87.37(6.17)
Avg(std)	72.75(21.6)	80.99(12.72)	81.35(16.22)	85.22(7.80)	89.29(6.45)	84.94(7.45)	89.98(6.43)

Table 4: SR results on Crabs dataset (30 runs)

5. CONCLUSION

We outline an approach to dimensionality reduction that seeks to simultaneously optimize the human interpretability and the discriminative power of the resulting dimensions. We show that this approach is valuable for exploratory data analysis and data visualization, as well as having potential for automatic model selection.

We have evaluated the approach as a data visualization task, claiming that increased classifier performance on low dimensional data would lead to increased visualization results. Future work will examine other visualization measures including inter- and intra-cluster distance measures in the reduced dimension space. Also, we will explore the use of this approach for model selection – evaluating how well the derived dimensionality reduction is able to generalize to unseen data.

6. REFERENCES

[1] `www.ics.uci.edu/~mlearn`.

[2] `www.stats.ox.ac.uk/pub/PRNN`.

[3] J. Eggermont. *Data Mining using Genetic Programming: Classification and Symbolic Regression.* PhD thesis, Leiden University, Netherlands, 2005.

[4] Ian Witten Eibe, Ian H. Witten, Eibe Frank, Len Trigg, Mark Hall, Geoffrey Holmes, and Sally Jo Cunningham. Weka: Practical machine learning tools and techniques with java implementations, pages 192–196, 1999.

[5] César Estébanez, Ricardo Aler, and José María Valls. A method based on genetic programming for improving the quality of datasets in classification problems. *IJCSA*, 4(1):69–80, 2007.

[6] Sean Luke. Ecj 19-a java-based evolutionary computation research system. `http://cs.gmu.edu/~eclab/projects/ecj/`.

[7] D.P. Muni, N.R. Pal, and J. Das. Genetic programming for simultaneous feature selection and classifier design. 36(1):106–117, February 2006.