MSDD as a Tool for Classification

Tim Oates

July 28, 1994

The Multi-Stream Dependency Detection algorithm has been applied to a variety of classification problems from the UC Irvine repository to assess performance and operating characteristics on "real world", rather than artificial, data sets. Although MSDD was not designed to be a classifier, its performance on a few initial problems prompted further exploration. In this memo I describe the performance of MSDD on the various problems that have been tested to date, and compare that performance to other results published in the machine learning literature. The majority of the problems discussed herein were chosen from a list of thirteen presented in [3] as being a minimal representative set that covers several important features that distinguish problem domains.

The MSDD algorithm was used with non-redundant child generation, left-to-right instantiation of wildcards, and the S_2 heuristic in all cases. Real valued features in the data sets were binned into between five and twenty equally sized bins. The size was chosen after briefly experimenting to see where the best accuracy was obtained. Unless otherwise noted, each data set was randomly distributed into a set containing 2/3 of the instances for training and another set containing the remaining 1/3 for testing. MSDD was run on ten different random splits resulting in a mean classification accuracy.

Breast Cancer

This data set involves classifying breast cancer cases from the University of Wisconsin at Madison Hospital as either benign or malignant. Each training instance contains ten features in addition to the class label.

MSDD accuracy: 95.15%

- 10,000 search nodes

Other results from literature

- 93.5% – two pairs of parallel hyperplanes (1 trial only)
- 95.9% – three pairs of parallel hyperplanes (1 trial only)
- 93.7% – 1 nearest neighbor

Diabetes

This data set involves identifying patients who show signs of the onset of diabetes based on eight features. The default accuracy is 65.1%.

MSDD accuracy: 71.33%

- 10,000 search nodes

Other results from literature

- 76% – ADAP

Heart Disease

This data set involves separating patients who have heart disease from those that do not based on thirteen features. The default accuracy is 54.1%.

MSDD accuracy: 79.21%

- 20,000 search nodes

Other results from literature

[1] lists results for seventeen different algorithms with only two achieving better accuracy than MSDD. Their scores were 79.4% and 80.6% respectively.

Hepatitis

This data set requires determination of whether patients with hepatitis will either live or die based on nineteen features. The default accuracy is 79.35%. Although MSDD only achieves default accuracy, it was not the case that the rules used for prediction always indicated the majority class.

MSDD accuracy: 80.77%

- 10,000 search nodes

Other results from literature

[1] presents results for a number of algorithms with accuracy ranging from 71.3% to 85.8%. Of those reported, thirteen had lower accuracy than MSDD and sixteen had higher accuracy. The mean accuracy of the 29 algorithms was 81.26%.

LED-7

Recall that LED displays contain seven light-emitting diodes from which the ten decimal digits can be constructed. Training instances from this domain consist of seven boolean values indicating the state of the seven LED's, and the corresponding decimal digit. To complicate matters, each of the seven boolean values for a given training instance have been reversed with a 10% probability. The class label, however, is always correct for the training instance before noise is added.

MSDD accuracy: 70.54%

- 1000 training instances
- 500 test instances
- 5000 search nodes

Other results from literature

- 74% – Bayes Optimal
- 71% – CART decision tree
- 71% – Nearest neighbor
- 72.6% – C4 decision tree with pessimistic pruning
- 73.3% – AND-OR algorithm

LED-24

This problem is the same as LED-7 except that an additional seventeen boolean attributes whose values are randomly assigned either 0 or 1 are added to each training instance.

MSDD accuracy: 71.28%

- 1000 training instances
- 500 test instances
- 5000 search nodes

Other results from literature

- 74% – Bayes Optimal
- 70% – CART decision tree
- 41% – Nearest neighbor
- 70.7% – NTgrowth+ with 700 training instances
- 71.5% – NTgrowth+ with 1000 training instances

Lymphography

This is yet another medical diagnosis domain that has a four-valued class label that must be ascertained from eighteen other features. The four classes appear with rather different frequencies: 1.4%, 2.7%, 41.2%, and 54.7%.

MSDD accuracy: 78.16%

- 15000 search nodes

Other results from literature

- 76% – Assistant-86
- 83% – Simple Bayes
- 82% – CN2
- 80-82% – AQ15

NetTalk

This data set involves translation of letters situated in English words into the correct phoneme. It was used in the famous NetTalk study in which Sejnowski and Rosenberg trained a neural network to perform the translation. My approach to encoding the input was taken from that study. Precursor multi-tokens are eight tokens wide. The first seven contain part of a word taken from the data set, and the last token contains the phoneme corresponding to the letter (token) in the fourth position. Words are padded with blanks as needed. For example, the word "and" whose phonetic transcription is "@nd" would yield three possible input to MSDD:

- -,-,-,A,n,d,-,@
- -,-,a,N,d,-,-,n
- -,a,n,D,-,-,-,d

Included in the data is a list of the 1000 most common English words. Those words contained a total of 5048 letters that were used to create the same number of training instances for MSDD. The resulting rules were then used to test classification accuracy on the training set and the full corpus of 20,008 words (146,943 letters).

MSDD accuracy on training set: 73.42%
MSDD accuracy on full corpus: 70.11%

- 50,000 search nodes

NetTalk results

- 98% – 1000 word training set
- 77% – full corpus

Monks-2

The second monks problem involves binary classification based on six features x_1 - x_6 where $x_{1,2,4} \in \{1,2,3\}$, $x_{3,6} \in \{1,2\}$, and $x_5 \in \{1,2,3,4\}$. The task is to separate those instances for which *exactly two of the attributes are 1* from the remainder. This is very similar to the parity problem and was designed to be difficult for symbolic learning algorithms. All results reported for this data set are based on a single trial since it is explicitly divided into a training set and a test set.

MSDD accuracy: 79.17%

- 169 training instances
- 432 test instances
- 5000 search nodes

Other results from literature
[2] lists results from a competition involving 24 different algorithms of which nine had higher classification accuracy than MSDD (including three neural network algorithms with 100% accuracy) and fifteen had lower accuracy. Of the nine that were better than MSDD, six were at least ten percentage points higher. Of the fifteen that were worse than MSDD, twelve were at least ten percentage points lower.

Mushroom

This problem involves classification of mushrooms as either poisonous or edible based on 22 features. Although the default accuracy for this data set is 50%, it appears to be a relatively easy problem with many algorithms achieving very high accuracy.

MSDD accuracy: 99.49%

- 500 training instances
- 7,624 test instances
- 30,000 search nodes

Other results from literature

- 92.7% – Marsh
- 95.0% – HILLARY
- 95.0% – STAGGER

- 98.6% – GINI
- 98.6% – Info Gain
- 99.1% – neural nets
- 99.9% – ID3, C4
- 100% – C4 pruned

Thyroid

This medical diagnosis domain requires classification of patients as either hypothyroid or normal. Note that the default accuracy is 95.0%. It appears that MSDD predicts the majority class for all test instances, thereby achieving close to default accuracy.

MSDD accuracy: 95.46% (see note below)

- 20,000 search nodes

NOTE: Although the MSDD algorithm easily falls victim to pursuing the majority class, the flexibility of the algorithm makes at least two fixes to the problem trivial. First, one could restrict MSDD so that it generates rules only for the minority class by not telling it about the token that represents the majority class. The most accurate rules generated can then be combined with a single default rule that guesses the majority class. I implemented this approach and achieved 97.25% accuracy on two different random splits of the data. The standard deviation of the ten trials used to generate the 95.46% figure given above was 0.326. The fact that the two new trials are more than 5 standard deviations away from the original mean is indicative of a significant improvement. The second method for dealing with unequal class sizes is to simply sample them so that the training data is composed of equal numbers of each. I did not apply that method in this case.

Waveform-40

Training instances from this domain consist of twenty noisy samples from one of three different waveforms, with an additional nineteen "features" that are random noise sampled from a gaussian distribution with mean 0.0 and variance 1.0. The task is to classify an instance according to the waveform used to generate it.

MSDD accuracy: 73.02%

- 1000 training instances
- 500 test instances
- 15,000 search nodes

Other results from literature

- Bayes optimal – 86%
- CART decision tree – 72%
- Nearest neighbor – 38%

Conclusions and Future Work

How should one interpret the results presented above? Given the fact that MSDD was not designed as a classifier, and that the amount of tuning that went into these experiments was minimal, the results are quite good. Our accuracy on Breast Cancer, Heart Disease, LED-24, Monks-2 (compared to other symbolic learning algorithms) and Mushroom is among the best reported. Our accuracy on the other data sets is comparable to the average performance of the other algorithms taken together. On none of the problems does MSDD stand out as a poor performer.

Of the thirteen data sets listed in [3], results for eleven are given above. I am in the process of gathering results for the two that remain which are Promoter and Soybean. The Soybean data set, which is known among the ML community to be an easy one, is difficult for MSDD. The large number of features cause the algorithm to focus exclusively on the first class expanded.

References

[1] Holte, Robert C. Very simple classification rules perform well on most commonly used data sets. In *Machine Learning*, (11), pp. 63-91, 1993.

[2] Thrun, S.B. The MONK's problems: A performance comparison of different learning algorithms. Carnegie Mellon University, CMU-CS-91-197.

[3] Zheng, Zijian. A benchmark for classifier learning. Basser Department of Computer Science, University of Sydney, NSW.