2004-07-01

Learning Multiple Correct Classifications from Incomplete Data using Weakened Implicit Negatives

Dan A. Ventura
ventura@cs.byu.edu

Stephen Whiting

Original Publication Citation

Stephen Whiting and Dan Ventura, "Learning Multiple Correct Classifications from Incomplete Data using Weakened Implicit Negatives", Proceedings of the International Joint Conference on Neural Networks, pp. 2953-2958, July 24.

BYU ScholarsArchive Citation

Ventura, Dan A. and Whiting, Stephen, "Learning Multiple Correct Classifications from Incomplete Data using Weakened Implicit Negatives" (2004). *Faculty Publications*. 433.
https://scholarsarchive.byu.edu/facpub/433

Learning Multiple Correct Classifications from Incomplete Data using Weakened Implicit Negatives

Stephen Whiting and Dan Ventura
Department of Computer Science
Brigham Young University
Provo, UT 84602, USA
swhiting@cs.byu.edu, ventura@cs.byu.edu

Abstract—**Classification problems with output class overlap create problems for standard neural network approaches. We present a modification of a simple feed-forward neural network that is capable of learning problems with output overlap, including problems exhibiting hierarchical class structures in the output. Our method of applying weakened implicit negatives to address overlap and ambiguity allows the algorithm to learn a large portion of the hierarchical structure from very incomplete data. Our results show an improvement of approximately 58% over a standard backpropagation network on the hierarchical problem.**

I. INTRODUCTION

Classification problems normally involve the labeling of unclassified data with a specific output class. Often, this is difficult due to regions in the problem space that contain a mixture of overlapping classes. Class overlap in classification problems is traditionally thought of as noise, and many algorithms attempt to filter this noise, or simply ignore it. Other methods attempt to improve classification accuracy by providing for an unsure, or soft, classification in the overlapping regions, thereby improving accuracy without implicitly correcting the noisy data [1,3,4]. Our approach considers the possibility that such overlap is an inherent characteristic of the problem space, and that a network classifier should represent this overlap. Our algorithm continues to provide distinct answers, insofar as it is possible given the data. By intelligently allowing for a mixture of classes, the algorithm can more accurately classify data that is near the classification boundaries.

In general, classification problems are seen as learning a functional mapping from the input to the output space.

$$f : X \rightarrow C \tag{1}$$

This representation has one major flaw; it assumes that all of the output classes are mutually exclusive. The principle of mutual exclusion can be violated in at least two ways. The first is noise, which changes data values in such a way that the output classes incorrectly overlap. Secondly, if the outputs naturally have some mutual overlap, then assuming mutual exclusion will create errors in the learning process. While many problems do contain noise, rejecting the possibility that natural overlap can occur in real-world problems contradicts the complexity of such natural systems. While one answer may be the 'most correct', in complex systems, there can be many acceptable, or right, answers. Examples of such systems include hierarchy trees, decision making, and many natural language problems. Classification problems that contain multiple correct answers in some regions can be approximated as a relational mapping from the input to the output space. Alternately, we can continue to approximate the problem as a functional mapping as long as we change the output range as shown in Equation 2, where 2^C represents the power set of C:

$$g : X \rightarrow 2^C \tag{2}$$

While some existing algorithms, such as backpropagation [6], can learn relational mappings, they require the data to provide the complete output vector for every data instance. The current methods of hand-labeling data can already be unreliable and incomplete, without forcing them to provide complete output vectors for all data instances. This restriction makes it desirable to have an algorithm that can learn multiple correct classifications from data that is labeled with only a single output class. The Weakened Implicit Negatives (WIN) method learns to consider classification overlap as a natural part of the problem when it is encountered, thus fulfilling our requirement that it can learn multiple correct answers when provided data that only contains one classification for any given instance.

Section II covers a brief literature search into soft classification techniques and learning without assuming mutual exclusion. Section III details the implementation of the WIN method, including how a standard feed-forward network can be modified to account for class overlap. Section IV explains our experimental set up and our empirical results are presented in Section V. Section VI provides final conclusions and ideas for future research.

II. Related Works

Recently, a number of papers have discussed the benefits of finding and analyzing regions of uncertainty, or overlap, in classification problems. Support Vector Machines were used in [3] to identify such regions of classification uncertainty in the problem space. By identifying such regions, the system was able to avoid some misclassifications, but in the uncertainty region, no answer could be given, and the algorithm was still treating the uncertainty as noise in the functional mapping.

Other algorithms can produce more accurate solutions to many problems by allowing the network to defer classification if the instance is within an uncertainty, or class boundary, region [1,4]. Essentially, the benefit of indecision is that by doing so, the algorithm is able to avoid overfitting the network to the training data, and generalize better on the overall problem. Baram reports an average improvement to classification accuracy to stock exchange information in [1]. In [4], Ishibuchi and Nii present a soft decision network that can return a subset of possible classes based on interval regions. Their algorithm works by rejecting classes that are clearly invalid given a possibility analysis. The possibility analysis will unfortunately give no indication as to the probability of encountering a specific class in an overlap region.

In all of these cases, the problem of what to do inside of the uncertainty region remains. Deferring decision making, or providing an accurate estimate of the uncertain regions, can improve classification, but fails to address the real problem of overlapping output classes. Giving an accurate representation of output strength inside an uncertainty region is left to the end user. Our algorithm, however, possess the ability to give a best guess inside uncertainty regions, as well as represent the presence of other likely output classifications.

A separate paper [5] also addresses regions of uncertainty and how to correctly classify instances when various outputs are independent of each other. Regier's work looks at learning without the assumption of implicit negativity. Implicit negativity assumes that every instance is labeled with all corresponding correct labels and that any other classification is therefore incorrect. As long as classification learning is cast as the learning of functional mappings, the implicit negativity assumption holds. Real-world data, however, can contradict this assumption, and so [5] proposes a method of weakening the implicit negatives. The paper proposes a simple, hand picked constant to serve as a weakening tool. The algorithm is unable to dynamically learn the optimal value, and is limited its scope of applicability.

Our research expands the scope of weakened implicit negatives, allowing the system itself to learn better parameters.

III. Methods

Neural networks that make use of weakened implicit negatives are able to learn relational mappings. While weakened implicit negatives could be used with any supervised learning technique, we restrict ourselves to implementing them on a feed-forward neural network using the backpropagation learning algorithm [6].

We make use of a sigmoid activation function at both the hidden and output layers given by:

$$\sigma(net) = \frac{1}{1+e^{-net}} \tag{3}$$

The algorithm uses the standard feed-forward method except in the calculation of the error values of the output nodes. The normal sum squared error function for backpropagation learning is giving by:

$$error_k = (target_k - output_k) * \frac{d}{dx}\left(\sigma\left[\sum_j \left(hidden_j * w_{kj}\right)\right]\right) \tag{4}$$

Weakening the implicit negative assumption is done by reducing the error values for output nodes that aren't given a specific target value by a data instance. [5] proposes that this can be done with a hand-picked parameter as in Equation 5.

$$error_k = (target_k - output_k) * \frac{d}{dx}\left(\sigma\left[\sum_j \left(hidden_j * w_{kj}\right)\right]\right) * \beta \tag{5}$$

The problem with Equation 5 is that the backpropagation learning technique is based on minimizing the error values. If the system attempts to learn the optimal β value for the data, an error-based learning technique will attempt to minimize error by reducing β to zero. While any form of direct weakening of the implicit negatives can serve to improve network accuracy, the system has no arbitrary means to choose a good value for the parameters.

Our method of Weakened Implicit Negatives avoids directly modifying error values. The algorithm instead learns to calculate new target values for any unknown output value. The calculated target values are allowed access to all of the information provided by the training data. The new target values are learned by a separate component of the system referred to as the Target Correlation Black Box (TCBB). The TCBB can be represented as the function h in the following Equation:

$$h(\vec{x}, c) \rightarrow \vec{y} \tag{6}$$

where $\vec{x} \in X, c \in C, \vec{y} \in [0,1]^m$ and $m = |C|$.

To limit the scope of our work, we ignore the input vector when computing the new target values, causing h to represent a target correlation matrix. The learned target values replace the assumed values in Equation 4 during training. The following equation replaces the implicit output target from Equation 4 with an output target value learned by our system.

$$error_k = [h_k(\bar{x}, c) - output_k] * \frac{d}{dx}\left(\sigma\left[\sum_j \left(hidden_j * w_{kj}\right)\right]\right) \quad (7)$$

As mentioned before, $\bar{x}$ is currently ignored by h. Also, since $c \in C$, we can reduce Equation 7 to

$$error_k = [h_k(c) - output_k] * \frac{d}{dx}\left(\sigma\left[\sum_j \left(hidden_j * w_{kj}\right)\right]\right) \quad (8)$$

While equation 8 makes the weakening of implicit negatives indirect, it introduces the added complexity of learning the appropriate target values for unknown output nodes. To solve the problem of learning the target values, the system uses the current network to approximate appropriate target values. The network and the unknown target values alternate learning in a manner reminiscent of the EM algorithm [7]. Figure 1 illustrates this process.

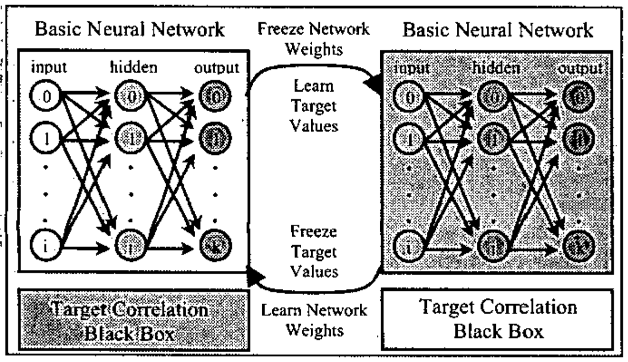

Fig. 1. The network training cycle alternates between training the basic network and the target correlation values.

Since the network itself has the information provided by the listed outputs, the learned target values need only provide a rough estimate of how strongly the various output classes overlap. The learned target values form a matrix of output class correlations. This matrix, γ, represents an increase in the probability of class k being concurrently active when the output class c is present in the training data.

The algorithm starts with the basic assumption of implicit negativity (target values of zero), and then weakens this assumption over time by increasing these target values. The target values are updated according to the following equation.

$$\gamma_{c,k} = \gamma_{c,k} + (net_k(\bar{x}) - h_k(c)) * \eta_{TCBB} \quad (9)$$

The learned target values are trained on the same data as the network, with the target values of the matrix being given by the current neural network that has been learned. Generally, we allow the network to train for some percentage (5-50%) of the total training time before beginning to learn the target values in an attempt to improve learning.

The target values can be periodically reset to zero in order to allow the network to compensate for incorrect learning in the early stages of learning. Experimentation has shown that resetting the target values occasionally helps classes that overlap slightly to avoid falsely inflating the unknown target values to one. This false inflation is the equivalent of saying that the two classes are always concurrently active.

IV. Experiments

Our algorithm is compared on a pair of problems from the UCI Machine Learning Repository [3]. The first problem is the Iris data set. The iris problem is chosen because of its simplicity and the fact that one output is linearly separable, and the other two outputs overlap only slightly. The problem is one that can be learned almost perfectly with a standard network using the backpropagation learning technique. We use this as a sanity check to test that our algorithm's performance isn't degraded on problems that exhibit mutual exclusion amongst the output classes.

The second problem we test on is the Glass data set. This data set contains instances of various kinds of glass which need to be identified. The data set itself is broken up into seven output classes, of which one is not represented in the data. This data set is much more complex than the Iris set and contains large amounts of overlap between output classes. In addition, the data set includes a description of the types of glass. Using these descriptions, we have been able to create an additional data set which contains hierarchical classifications based on information provided in the original set.

The modified glass set has the following outputs added: Building glass, Vehicle glass, Window glass, Non-window glass, float-processed, and non-float-processed. The 214 instances from the original data set were duplicated (multiple times if necessary) and relabeled with another applicable label, as given in the description of the original data set as well as having the complete output vector for testing purposes. For example, an instance that was originally labeled as class 1 (float-processed building window) is now listed four times in the modified set with labels 1, 8 (float-processed), 10 (window), and 12 (building). The new data set contains a total of 805 instances, with the average point having 3-4 classifications that should be activated concurrently. The concurrent activation represents the hierarchical structure of the data given in the description of the Glass data set. Training and testing accuracy results on both the modified and original glass data sets will be provided in the next section. The modified glass data set allows us to illustrate the adaptability of our algorithm in learning in the presence of multiple correct answers.

V. RESULTS

We ran our tests using 10-fold cross-validation and averaging over 30 trials. We also recorded maximum and minimum results over the 30 trials to provide bounds for the accuracy measures. Our results are presented as classification accuracies over the training and test sets.

For both the standard Glass and the Iris data sets, results are shown for a winner-take-all scheme where the highest output node is selected and compared to the listed output. The modified Glass data set is rated on an accuracy measure that thresholds the individual outputs and checks against the output vector provided in the data set. Reported results show the average accuracy over all outputs.

Using the winner-take-all scheme allows us to ensure that our algorithm doesn't negatively affect the performance of the backpropagation learning on simple problems with little or no overlap. The Iris data set was run for 5,000 training epochs with an η value of 0.001 and 8 hidden nodes. The results shown are accuracies reported on a separate run after training concluded. The standard Glass data set ran for 50,000 training epochs with the same η value and using 15 hidden nodes. As before, the accuracies reported were computed from an epoch during which no training occurred. Table I shows the results of both tests using the standard backpropagation approach, and using weakened implicit negatives (WIN). The "Max" and "Min" rows represent the maximum and minimum accuracies measured over the 30 trials. The performance of the two algorithms is nearly indistinguishable on the problems when testing using the winner-take-all scheme.

TABLE I
CLASSIFICATION ACCURACY ON IRIS AND GLASS

		Weakened Implicit Negatives		Standard Backpropagation	
		Train	Test	Train	Test
Iris	Max	+0.06%	+0.62%	+0.07%	+0.62%
	Mean	98.01%	97.38%	98.01%	97.38%
	Min	-0.01%	-0.04%	-0.01%	-0.04%
Glass	Max	+1.28%	+3.40%	+1.46%	+4.21%
	Mean	74.63%	63.75%	74.60%	63.41%
	Min	-1.31%	-4.22%	-1.34%	-3.89%

The results from the modified Glass data set, however, provide us with a look at the true power of the algorithm. The modified Glass set allowed us to train the two algorithms with only simple functional learning examples, but test them on the complete output vector. Because our algorithm is designed to interpret areas of the output space that overlap, and adjust the target values appropriately, it should be capable of learning to correctly classify multiple outputs as active (high) from the purely functional data. In order to verify how closely we are able to correctly classify all of the appropriate outputs, we also ran a test using a standard backpropagation network, but provided it the full output vectors for training and testing.

The modified glass data set is tested using an activation threshold of 0.5. Output nodes are compared against the target values provided by the completed output vector, but were provided only a single correct answer during training. Using a maximum likelihood estimator (MLE) would result in a selection of zero for all outputs regardless of input. This selection method produces an accuracy rating of approximately 70%. The number of training epochs was also increased to 100,000. The number of hidden nodes and η values were kept the same as they were on the standard Glass problem. Figure 2 shows a graph of testing accuracy over the course of training. The points on the graph are the mean value taken every 100 epochs over the 30 trials and averaged over the 10 folds.

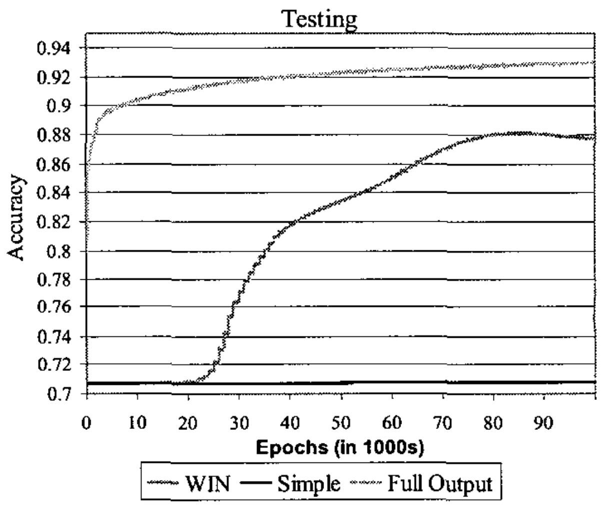

Fig. 2. Testing accuracy results over the course of training on the Modified Glass data set.

When provided only the single output, a simple backpropagation network struggles and improves only slightly over the course of training. If provided with the full output vector during training (the top arc on Figure 2), the backpropagation network quickly learns the correct outputs, finally reaching a point where ambiguity and overlap in the original output classes makes further learning impossible. Using the full output vector, the simple network attains a testing accuracy of approximately 93%.

Using weakened implicit negatives, we greatly improve accuracy even though we are learning on data with a single output. Our algorithm achieves a final accuracy of approximately 88%. This represents approximately 77% of what the backpropagation network learned when allowed to train on data with a full output vector.

The train and test accuracies are virtually identical on this problem, but are provided for completeness. Table II provides a complete summary of accuracy results including the maximum and minimum accuracies over the 30 trials.

TABLE II
CLASSIFICATION ACCURACY FOR THE MODIFIED GLASS DATA SET

		Train	Test
Weakened Implicit Negatives	Max	+0.31%	+0.49%
	Mean	88.00%	87.71%
	Min	-0.51%	-0.58%
Standard Backprop	Max	+0.03%	+0.10%
	Mean	70.71%	70.77%
	Min	-0.01%	-0.05%
Standard Backprop Trained w/ Full Output Vector	Max	+0.65%	+0.81%
	Mean	94.03%	92.99%
	Min	-0.57%	-0.64%

VI. CONCLUSIONS AND FUTURE WORK

Using weakened implicit negatives can dramatically improve network performance on problems with multiple correct answers and our ability to actively learn the appropriate parameters enables us to utilize the adaptive power of the WIN theory to learn complex classification problems.

In addition, the WIN algorithm demonstrates little loss in classification accuracy on problems that exhibit mutual exclusion in the output classes. We have shown that our algorithm provides a relative improvement of almost 58% over the simple backpropagation network when trained with only a single output class. If we compare our results to those of the backpropagation network trained on the full output vector, our algorithm shows the ability to learn approximately 77% of the additional information without being told about the hierarchical nature of the data.

Providing sufficient data for a learning algorithm is difficult enough without insisting that the data provide completed output vectors for all instances. Our algorithm provides a means of learning complex, overlapping classification problems without a completed output vector.

Although our algorithm performs well on classification problems with multiple correct answers, there is still room for improvement. We are currently looking at both simpler and more complex correlation structures for approximating the target values of a complete output vector. The algorithm also needs to be tested on a wider range of problems, although the question of analyzing results in the absence of accurate, and complete, output vectors is a difficult one. Another of our directions for future research is in applying the WIN method toward problems in Natural Language Processing such as Part-of-Speech tagging where ambiguity in output classes is known to exist.

REFERENCES

[1] Baram, Yoram. "Partial Classification: The Benefit of Deferred Decision." *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 20, No. 8, August 1998.

[2] Blake, C.L. & Merz, C.J. *UCI Repository of machine learning databases* [http://www.ics.uci.edu/~mlearn/MLRepository.html]. Irvine, CA: University of California, Department of Information and Computer Science. 1998.

[3] Drago, Gian Paolo and Marco Muselli. "Support Vector Machines for uncertainty region detection." In *Neural Nets: WIRN VIETRI '01, 12th Italian Workshop on Neural Nets* (Vietri sul Mare, Italy, 17-19 May 2001) Eds. M. Marinaro, R. Tagliaferri, London: Springer, 108-113.

[4] Ishibuchi, Hisao and Manabu Nii. "Neural Networks for soft decision making." *Fuzzy Sets and Systems*, 115 (2000), 121-140.

[5] Regier, Terry. "Learning without Explicit Negative Evidence." In *The Human Semantic Potential: Spatial Language and Constrained Connectionism*. Cambridge, MA: MIT Press. 1996.

[6] Rumelhart, David, Geoffrey Hinton, and Ronald Williams. "Learning internal representations by error propagation". In *Parallel distributed processing: Explorations in the microstructure of cognition*. Vol 2, *Foundations*, by David Rumelhart, James McClelland, and the PDP Research Group. Cambridge, MA: MIT Press. 1986.

[7] A.P. Dempster, N. M. Laird, and D.B. Rubin. "Maximum likelihood from incomplete data via the EM algorithm." *Journal of the Royal Statistical Society*, B 39:1-39, 1977.