Are we better off without Counter-Examples?

Nathalie Japkowicz
Computer and Information Science
The Ohio State University
395 Dreese Lab, 2015 Neil Avenue
Columbus, OH 43210, USA
e-mail: nat@cis.ohio-state.edu

Abstract

Concept-learning is commonly implemented using *discrimination*-based techniques which rely on both examples and counter-examples of the concept. Recently, however, a *recognition*-based approach that learns a concept in the absence of counter-examples was shown to be more accurate than its discrimination counterpart on two real-world domains and as accurate on the third. The purpose of this paper is to find out whether this recognition-based approach is *generally* more accurate than its discrimination counterpart or whether the results it obtained previously are purely coincidental. The analysis conducted in this paper concludes that the results obtained on the real-world domains were not coincidental, and this suggests that recognition-based approaches are promising techniques worth studying in greater depth.

1 Introduction

Concept-learning can be approached in one of two ways. The first way consists of considering both examples and counter-examples of the concept at hand and learning how to *discriminate* between these two classes. The second way consists of considering only instances of the concept and learning how to *recognize* those examples.

Although discrimination-based methods are usually favored in the fields of Pattern Recognition, Machine Learning, Neural Networks, and Data Mining, a recognition-based method which uses an autoassociator was recently shown to be more accurate than its discrimination counterpart—MLP—on two out of three real-world domains and as accurate as MLP on the third. The purpose of this paper is to find out whether these observations result from pure coincidence or whether the autoassociator can be expected to outperform MLP in a variety of domains, thus suggesting that recognition-based concept-learning should be given greater attention than it has in the past.

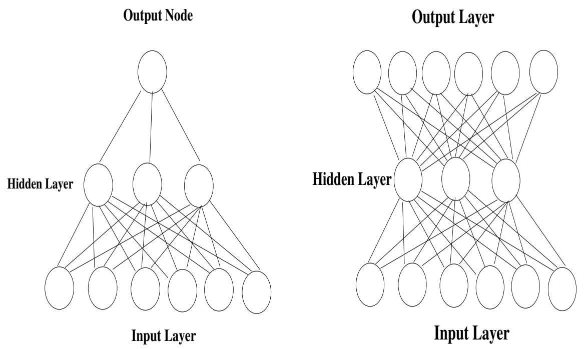
The paper is divided into four sections. Section 2 is a description of the discrimination and recognition systems contrasted in this work as well as a summary of the results previously obtained on real-world data. Section 3 states the particular problem considered in this paper. Section 4 discusses the experiments carried out and the results obtained. Section 5 is the conclusion.

2 Discrimination versus Recognition

This section begins by describing the two classification systems contrasted in this paper and it then reviews the results they obtained previously on real-world domains.

2.1 Systems' Description

A formal definition of the concept-learning problem typically involves an input vector x, and a response vector y which are such that the pair (x, y) belongs to some unknown joint probability distribution, P. The goal of concept learning is to induce a function $f(x)$ from a set $(x_1, y_1), (x_2, y_2)$, ... (x_N, y_N) of training examples, so that $f(x)$ approximates y. Multi-layer Perceptrons are a particular type of neural networks that are capable of approximating, as closely as desired, any such continuous functional mapping f. In this paper, we restrict our attention to multi-layer Perceptrons with a single hidden layer, i.e., with two layers of weights. Two examples of such feedforward neural networks are given in Figure 1. In figure 1(a), the input layer contains 6 units, the hidden layer, 3 units, and the output layer, a single unit. Figure 1(b) is similar except for the output layer which contains 6 units.



(a) MLP **(b) Autoassociator**

Figure 1: Examples of Feedforward Neural Networks: (a) The MLP Network; (b) The Autoassociator

The network of figure 1(a), which we refer to as "MLP" or "the MLP network", is the most commonly used network in the field of connectionist systems. It is typically used to implement a discrimination-based approach to classification. Using this scheme, each element y_i of y, the response vector, takes on a value of "1" or "0" which is interpreted as to mean that x_i, the input vector associated with y_i belongs or does not belong to the conceptual class of the problem. After training the network to approximate the function $f(x)$ from which the training examples are believed to have been generated, it is expected that, if the training set was appropriately designed, the network will be able to compute the appropriate label ("1" or "0") for any input vector of size d, the size of the input layer, even if that vector did not appear in the training set.

The network of figure 1(b) is not as universal as the MLP network, but it has been used previously to implement a compression scheme (e.g., [Cottrell *et al.*1987]). As well, and this is the application of interest here, it can be used to implement a recognition approach to concept learning as proposed by [Hanson and Kegl1987]. In this scheme, each response vector y_i of y is set to x_i, the corresponding input vector. In other words, the network is trained to reproduce the input at the output layer. Such a network is called an autoassociator. For concept-learning to take place, the network is trained to reconstruct conceptual data only and classification is performed on a new vector x_{Test} by comparing its *reconstruction error*[1] to a threshold, and assigning it to the conceptual class if the reconstruction error is smaller than this threshold and to the other class otherwise. The idea behind this recognition-based classification scheme is that since the autoassociator is trained to compress and decompress examples of the conceptual class only, when tested on a novel data point, it will compress and decompress it appropriately if this example belongs to the conceptual-class, but it will not do so appropriately if the example does not belong to the conceptual class.

2.2 Previous Findings

MLP and autoassociation-based classification were previously compared and contrasted on a series of three real-world domains: helicopter gearbox monitoring, DNA promoter recognition, and sonar target recognition. Details of these experiments can be found in [Japkowicz *et al.*1995], but what is of interest here is the classification error rates obtained by the two systems on 5-fold cross-validation experiments on the three domains. These results are reported in Table 1. These results are interesting in that they show that in both the helicopter and the sonar target recognition case studies, autoassociation-based classification performed much more accurately than MLP. In the promoter case study, it did not, but the two systems performed equally well. The performance of the au-

Case Study	*Autoassociator Error (%)*	*MLP Error (%)*
Helicopter	3.125	10.9
Promoter	20	20
Sonar	20	32

Table 1: Summary of the Results for the three case studies.

[1] The reconstruction error corresponds to $\Sigma_{j=1}^{d}[x^j_{Test} - f(x^j_{Test})]^2$, where x^j_{Test} and $f^j(x_{Test})$ represent input unit j and output unit j of the network, respectively.

toassociator in relation to MLP is a bit unexpected given the fact that MLP uses more information than the autoassociator in the form of counter-examples. The purpose of this paper is thus to find out whether these results are only coincidental or whether they can be expected on other domains as well.

3 Problem Statement

In order to find out whether or not the results reported in the previous section are meaningful in a more general context, it is useful to reflect about the nature of concept-learning by recognition and discrimination. A useful framework for understanding the induction process of a generic classifier is to decompose it into two opposing processes in search of an equilibrium. The first process seeks to *generalize* from the examples of the concept given to the classifier; while the second one seeks to *specialize* these examples. Generalization amounts to seeking a concept description which characterizes all the examples of the concept while specialization consists of seeking a concept description which excludes all its counter-examples.[2] In this framework, it can be easily seen that discrimination-based classifiers learn concepts by generalizing from examples of these concepts while specializing using their counter-examples. Although recognition-based learners are also able to generalize from examples of the concept, they do not, however, have access to counter-examples for specialization. Their ability to specialize thus comes from certain internal capabilities ([Japkowicz *et al.*1999] discusses these capabilities in detail for the autoassociator).

The aforementioned contrast between discrimination- and recognition-based classification methods suggests that differences in the classification performance of the two types of systems may be introduced by domains requiring particularly strong or particularly weak specialization biases. In more detail, we expect that the MLP network should be better suited than the autoassociator to learning domains that require a strong specialization bias caused by the conceptual class, provided that the counter-conceptual class contains meaningful specialization information. This is because, the MLP network has the possibility to rely on the counter-examples during the inductive process, whereas the autoassociator does not. On the contrary, if domain requirements for a strong specialization come from the counter-conceptual class, then the autoassociator should not be affected since its learning strategy is independent of the conceptual class, whereas the MLP network, which relies on the counter-conceptual class, should. Finally, in cases where a weak specialization is required (caused by a sparse subconcept), then we expect that the autoassociator is more likely to learn this subconcept than the MLP network. This is because, once again, the autoassociator focuses on characterizing the conceptual class independently of the counter-conceptual class, whereas the MLP network may overlook a particularly sparse subconcept, especially if the counter-conceptual class is dense.

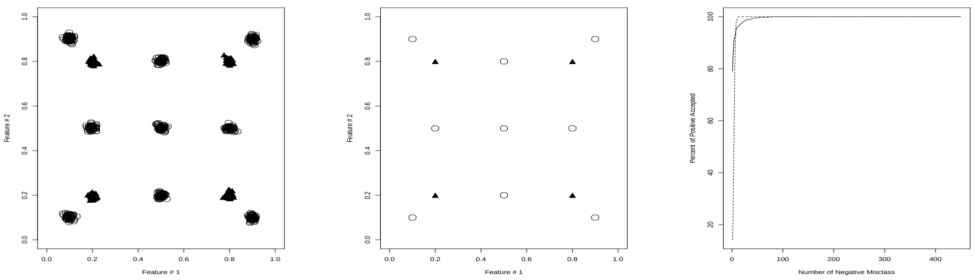
In order to test the assumptions just described, MLP and the autoassociator2r were compared on a series of artificial domains presenting the three types of characteristics deemed significant in the above discussion:

- Domains with strong specialization need coming from the conceptual class
- Domains with strong specialization need coming from the counter-conceptual class
- Domains with weak specialization need coming from the conceptual class

These domains were obtained by applying certain transformations to a *neutral* domain, labeled as such since it can be classified almost as accurately by the MLP network and the autoassociator. This domain is illustrated in Figures 2(a) and (b)[3] and the relative classification performance of MLP and the autoassociator is reported in Figure 2(c). The

[2] These ideas are formalized in [Mitchell1982] using the notion of "Version Spaces".

[3] Figure 2(a) represents the entire neutral domain. It is composed of four conceptual and nine counter-conceptual components. More specifically, Figure 2(b) shows that the means of the conceptual components are located at points: (.2, .2), (.2, .8), (.8, .2), and (.8, .8). The means of the counter-conceptual components are located at points: (.5, .5), (.1, .1), (.1, .9), (.9, .1), (.9, .9), (.5, .2), (.2, .5), (.5, .8), and (.8, .5). As shown in Figure 2(a), each mean actually stands for a cluster of data points normally distributed around that mean and of variance $\sigma^2 = 0.01$. This domain was inspired from the real-world domains used in [Japkowicz *et al.*1995] as described in [Japkowicz1999].



(a) Neutral **(b) Means Only** **(c) ROC Analysis**

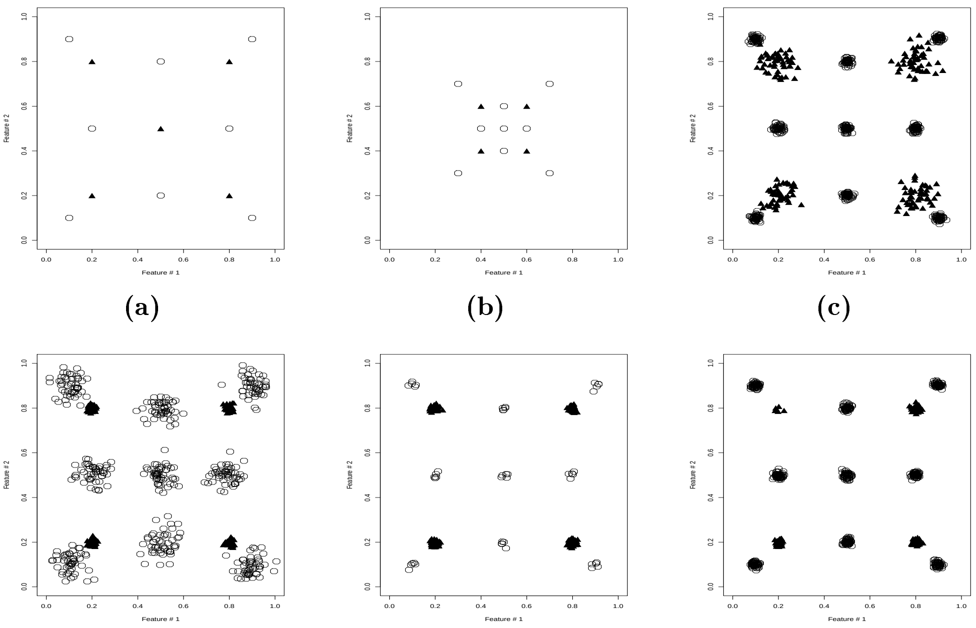
Figure 2: (a) and (b): The Original Artificial Domain; and (c): the result of a ROC analysis of the MLP and autoassociator on this domain.

meaning of Figure 2(c) will be clarified in Section 4, but the important point to notice for now is that the two curves are almost co-located, which means that the two systems are practically equally accurate.

The transformations applied to the neutral domain are illustrated in Figures 3 and 4. In more detail, modifications (a), (b) and (c) represent three domains in need of strong specialization coming from the conceptual class. These three modifications can be thought of as void reduction methods in that they all—in one way or another—decrease the void occurring between the four conceptual clusters of the original domain. Decreasing this void calls for a stronger specialization effect since it creates a tendency for classifiers to integrate the various conceptual clusters into a single larger cluster, a step that would cause a high classification error to occur since internal counter-examples would be misclassified as conceptual rather than counter-conceptual. Modifications (d) and (e) represent two domains in need of strong specialization coming from the counter-conceptual class. The purpose of modification (d) is to increase the amount of surface of each conceptual cluster surrounded by counter-conceptual data. This modification calls for greater specialization strength because the counter-conceptual data are not all located in a few localized spots, as in the neutral domain, but rather, they wrap around the conceptual clusters. The purpose of modification (e) is to decrease the density of each counter-conceptual cluster, thereby decreasing the amount of external specialization provided to systems which take such information into consideration, and calling for increased internal specialization. Finally, modification (f) represents a domain in need of weak specialization coming from the conceptual class. This modification was applied by decreasing the cardinality of one of the conceptual clusters in order to test whether or not, in case of unbalance in the density of the conceptual clusters, the small cluster is correctly identified or misclassified as a noisy occurrence.[4] Please note that the various types of domain characteristics embodied by these modifications do occur in real-world domains and, in particular, in the real-world domains of section 2.2 ([Japkowicz1999]).

According to our expectations, the domains of Figure 3(a), (b) and (c) should get more accurately classified by the MLP network and the domains of Figure 3(d), (e) and (f) should get more accurately classified by the autoassociator. The next section describes the experiments conducted on these domains as well as their results.

[4]In more detail, modification (a) was implemented by creating a concave, but connected pattern. It consisted of turning the central internal counter-conceptual cluster of figures 2(a) and (b) into a conceptual one. Modification (b) was implemented by moving the conceptual clusters to a distance of 0.2 units from each other (we recall that the original distance between each conceptual cluster was of 0.6 units). Modification (c) was implemented by increasing the variances of each conceptual cluster to 0.04 (we recall that the variance was 0.01 in the original domain). Modification (d) was implemented by increasing the variance of each counter-conceptual cluster to 0.04 (we recall that the variance was 0.01 in the original domain). Modification (e) was implemented by decreasing the size of the counter-conceptual clusters to 5 examples each (we recall that each cluster contained 50 data points in the original domain). Finally, modification (f) was implemented by decreasing the cardinality of the left top conceptual cluster of Figures 2(a) and (b) from 50 to 5 in order to test whether or not, in case of imbalance in the density of the conceptual clusters, the small cluster gets correctly identified.



(a) (b) (c)

(d) (e) (f)

Figure 3: (a), (b), (c): Modifications that should affect the autoassociator more than MLP. (d), (e), (f): Modifications that should affect MLP more than the autoassociator.

4 Experiments and Results

The results of the comparisons of MLP with the autoassociator on each of the domains of Figure 3 are displayed in Figure 4. These results are reported in the form of Receiver-Operating Characteristic (ROC) curves which plot the percent of positive instances of the concept correctly characterized versus the actual number of false positives. For each domain, the curves for the MLP network and autoassociator are plotted on the same graph and the curve located above the other represents the most accurate classifier on that domain. The autoassociator's performance is represented by a full curve while MLP's is represented by a broken curve. All the experiments were repeated five times and the curves displayed represent the average results obtained.[5]

These graphs show that, according to the expectations expressed in Section 3, MLP outperforms the autoassociator in the domains in need of strong specialization caused by the conceptual class (Figures 4(a), (b) and (c)); the autoassociator outperforms MLP in the domains in need of strong specialization caused by the counter-conceptual class (Figures 4(d)[6] and 4(e)); and the autoassociator outperforms MLP in the domains in need of weak

[5]The optimal number of hidden units was determined by running the two systems on each domain with different capacities, and retaining the capacity yielding the best result. In particular, the capacities considered are those of 1, 2, 4, 8, 16, 32 and 64 hidden units (for a list of the capacities retained for each domain and each system, see [Japkowicz1999]). The momentum of the networks was set to the standard value of 0.9 for each system, but while the learning rate of the autoassociator was set to the standard value of 0.05, it was increased to 0.2 in the MLP network, in order to speed up learning. The stopping criterion was established on the neutral domain at 5000 epochs for MLP and 2000 epochs for the autoassociator. This stopping criterion was deemed "practically reasonable" for all the domain variations studied in this paper since 1) actual convergence on the neutral domain takes place at least 1000 epochs earlier in both MLP and the autoassociator, and thus, the stopping criterion should impart sufficient time to all the other (closely related) domains as well; 2) Overfitting is unlikely to be too strong on any domain since they are all noise-free, and thus, the stopping criterion should not impart too much time to any domain. Further experiments establishing optimal stopping points for each domain and each system have been left for future work.

[6]Note that in the case of figure 4(d), the difference in accuracy of the two systems is very slight. However, since MLP was slightly more accurate than the autoassociator in the neutral domain, the difference in accuracy observed in domain 4(d) is relatively meaningful.

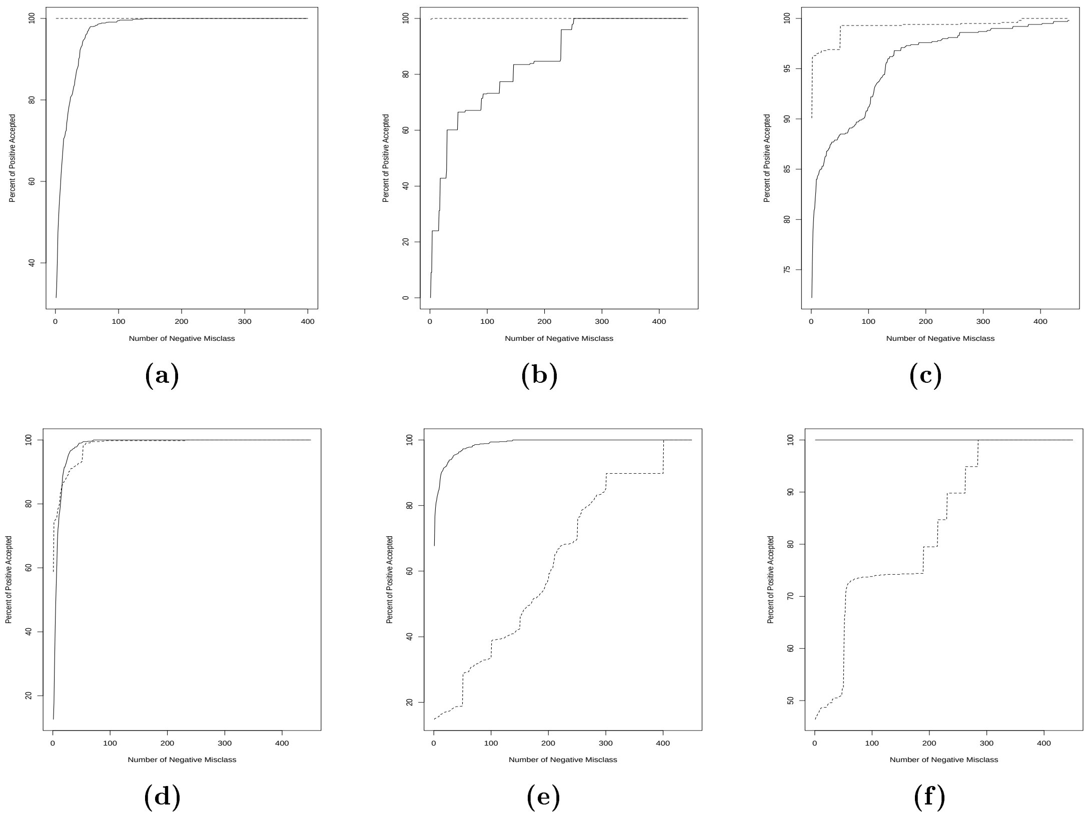


Figure 4: ROC Analysis for all the domains. The autoassociator's performance is represented by a full curve while MLP's is represented by a broken one. In domains (a), (b) and (c), as expected, MLP is more accurate than the autoassociator. In domains (d), (e), and (f), as expected, the autoassociator is more accurate than MLP.

specialization caused by the conceptual class (Figure 5(f)).

The results obtained in this section thus demonstrate that, at least on synthetic domains, autoassociation-based classification is more accurate than MLP in multi-modal domains presenting certain well-defined characteristics. As discussed in [Japkowicz1999] these results also extend to practical domains, and in particular, can be used to explain the accuracy results obtained on the real-world domains of Section 2.2.

5 Conclusion

This paper originated from the observation that autoassociation-based classification was reported to be more accurate than the MLP discrimination network on two out of three real-world domains and as accurate as MLP on the third, despite the fact that the autoassociator learns a concept in the absence of counter-examples [Japkowicz *et al.*1995]. The purpose of this paper was to find out whether this result was only coincidental or whether it could be expected in other domains. The results obtained suggest that the autoassociator is more accurate than MLP in domains that require particularly strong specialization coming from the counter-conceptual class or particularly weak specialization coming from the conceptual class. However, MLP is more accurate than the autoassociator in domains that require particularly strong specialization coming from the conceptual class.

The results of this study thus suggest that unsupervised learning techniques, which are often dismissed in favor of supervised ones in the context of binary classification, may present an interesting array of classification strengths. This observation comes in addition to the fact that unsupervised techniques learn concepts in the absence of negative data and are thus useful for a whole range of practical domains that cannot even be considered by supervised learning methods.

Acknowledgements

I would like to express my gratitude to Stephen J. Hanson and Casimir A. Kulikowski, my Ph.D. thesis advisors, for their helpful guidance and comments during the development of the ideas presented in this work.

References

[Cottrell *et al.*1987] Garrison W. Cottrell, Paul Munro, and David Zipser. Learning internal representations from gray-scale images: An example of extensional programming. In *Proceedings of the ninth AnnualConference of the Cognitive Science Society*, pages 462–473, Seattle, WA, 1987.

[Hanson and Kegl1987] Stephen J. Hanson and Judy Kegl. Parsnip: A connectionist network that learns natural language grammar from exposure to natural language sentences. In *Proceedings of the Ninth Annual Conference on Cognitive Science*, 1987.

[Japkowicz *et al.*1995] Nathalie Japkowicz, Catherine Myers, and Mark Gluck. A novelty detection approach to classification. In *Proceedings of the Fourteenth Joint Conference on Artificial Intelligence*, pages 518–523, 1995.

[Japkowicz *et al.*1999] Nathalie Japkowicz, Stephen J. Hanson, and Mark A. Gluck. Nonlinear autoassociation is not equivalent to pca. In *Submitted to Neural Computation (Revise and Resubmit)*, 1999.

[Japkowicz1999] Nathalie Japkowicz. *Concept-Learning in the Absence of Counter-Examples: An Autoassociation-Based Approach to Classification (in progress)*. PhD thesis, Rutgers University, 1999.

[Mitchell1982] Tom Michael Mitchell. Generalization as search. *Artificial Intelligence*, 18(2):203–226, March 1982.