The Hausdorff Distance Measure for Feature Selection in Learning Applications

Selwyn Piramuthu
Operations and Information Management
The Wharton School, University of Pennsylvania
1300 SH-DH, Philadelphia, PA 19104-6366
selwyn@opim.wharton.upenn.edu

Abstract

Recent advances in computing technology in terms of speed, cost, as well as access to tremendous amounts of computing power and the ability to process huge amounts of data in reasonable time has spurred increased interest in data mining applications. Machine learning has been one of the methods used in most of these data mining applications. It is widely acknowledged that about 80% of the resources in a majority of data mining applications are spent on cleaning and preprocessing the data. However, there have been relatively few studies on preprocessing data used as input in these data mining systems. In this study, we present a feature selection method based on the Hausdorff distance measure, and evaluate its effectiveness in preprocessing input data for inducing decision trees. The Hausdorff distance measure has been used extensively in computer vision and graphics applications, to determine the similarity of patterns. Two real-world financial credit scoring data sets are used to illustrate performance of the proposed method.

1 Introduction

It is widely recognized that around 80% of the resources in data mining applications are spent on cleaning and preprocessing the data. The actual mining or extraction of patterns from the data requires the data to be clean since input data are the primary, if not the only, source of knowledge in these systems. Cleaning and preprocessing data involves a few or several steps including procedures for handling incomplete, noisy, or missing data; sampling of appropriate data; feature selection; feature construction; and also formatting the data as per the representational requirements of techniques used to extract knowledge from these data.

Invariably, and unknowingly for the most part, irrelevant as well as redundant variables are introduced along with relevant variables to better represent the domain in these applications. A relevant variable is neither irrelevant nor redundant to the target concept of interest (John, et al., 1994). Whereas an irrelevant feature does not affect describing the target concept in any way, a redundant feature does not add anything new to describing the target concept while possibly adding more noise than useful information in concept learning.

Feature Selection is the problem of choosing a small subset of features that ideally is necessary and sufficient to describe the target concept (Kira and Rendell, 1992). Feature selection is of paramount importance for any learning algorithm which when poorly done (i.e., a poor set of features is selected) may lead to problems associated with incomplete information, noisy or irrelevant features, not the best set/mix of features, among others. The learning algorithm used is slowed down unnecessarily due to higher dimensions of the feature space, while also experiencing lower prediction accuracies due to learning irrelevant information. The ultimate objective of feature selection is to obtain a feature space with (1) low dimensionality, (2) retention of sufficient information, (3) enhancement of separability in feature space for examples in different categories by removing effects due to noisy features, and (4) comparability of features among examples in same category (Meisel, 1972).

Feature selection method using the Hausdorff distance measure is presented and evaluated in this study. The Hausdorff distance measure is widely used in computer vision and graphics applications, due to its excellent properties. It has, however, not received much attention in the feature selection literature.

This paper is organized as follows: Section 2 provides a brief overview of recent developments in feature selection methods. The proposed feature selection method using the Hausdorff method is presented in section 3. This is followed by illustration of the proposed method using two real-world financial credit scoring data sets in section 4. Section 5 concludes the paper with a brief discussion of this study.

0-7695-0001-3/99 $10.00 (c) 1999 IEEE

2 Recent Developments in Feature Selection

Feature selection is the problem of choosing a small subset of features that ideally is necessary and sufficient to describe the target concept (Kira and Rendell, 1992). A goal of feature selection is to avoid selecting too many or too few features than is necessary. If too few features are selected, there is a good chance that the information content in this set of features is low. On the other hand, if too many (irrelevant) features are selected, the effects due to noise present in (most real-world) data may overshadow the information present. Hence, this is a tradeoff which must be addressed by any feature selection method.

There exists a vast amount of literature on feature selection. Researchers have attempted feature selection through varied means, such as statistical (e.g., Kittler, 1975), geometrical (e.g., Elomaa and Ukkonen, 1994), information-theoretic measures (e.g., Battiti, 1994), mathematical programming (e.g., Bradley, et al., 1998), among others.

In statistical analyses, forward and backward stepwise multiple regression (SMR) are widely used to select features, with forward SMR being used more often due to the lesser magnitude of calculations involved. The output here is the smallest subset of features resulting in an R^2 (correlation coefficient) value that explains a significantly large amount of the variance. In forward SMR, the analyses proceeds by adding features to a subset until the addition of a new feature no longer results in a significant (usually at the 0.05 level) increment in explained variance (R^2 value). In backward SMR, the full set of features are used to start with, while seeking to eliminate features with the smallest contribution to R^2.

Malki and Moghaddamjoo (1991) apply the K-L transform on the training examples to obtain the initial training vectors. Training is started in the direction of the major eigenvectors of the correlation matrix of the training examples. The remaining components are gradually included in their order of significance. The authors generated training examples from a synthetic noisy image and compared the results obtained using the proposed method to those of standard backpropagation algorithm. The proposed method converged faster than standard backpropagation with comparable classification performance.

Siedlecki and Sklansky (1989) use genetic algorithms for feature selection by encoding the initial set of n features as n-element bit string with 1 and 0 representing the presence and absence respectively of features in the set. They used classification accuracy, as the fitness function (for genetic algorithms while selecting features) and obtained good neural network results compared to branch and bound and sequential search (Stearns, 1976) algorithms. They used a synthetic data as well as digitized infrared imagery of real scences, with classification accuracy as the objective function. Yang and Honavar (1997) report a similar study. However, later Hopkins et al. (1994) show that classification accuracy may be a poor fitness function measure when searching for reducing the dimension of the feature set.

Using Rough Sets theory (Pawlak, 1982), PRESET (Modrzejewski, 1993) determines the degree of dependency (γ) of sets of attributes for selecting binary features. Features leading to a minimal preset decision tree, which is the one with minimal length of all path from root to leaves, are selected. Kohavi and Frasca (1994) use best-first search, stopping after a predetermined number of nonimproving node expansions. They suggest that it may be beneficial to use a feature subset that is not a reduct, which has a property that a feature cannot be removed from it without changing the independence property of features. A table-majority inducer was used with good results.

The *wrapper* method (Kohavi, 1995) searches for a good feature subset using the induction algorithm as a black box. The feature selection algorithm exists as a wrapper around the induction algorithm. The induction algorithm is run on data sets with subsets of features, and the subset of feature with the highest estimated value of a performance criterion is chosen. The induction algorithm is used to evaluate the data set with the chosen features, on an independent test set.

Almuallim and Dietterich (1991) introduce MIN-FEATURES (if two functions are consistent with the training examples, prefer the function that involves fewer input features) bias to select features in the FOCUS algorithm. They used synthetic data to study the performance of the FOCUS, ID3, and FRINGE algorithms using sample complexity, coverage, and classification accuracy as performance criteria. They increased the number of irrelevant features and showed that FOCUS performed consistently better.

The IDG algorithm (Elomaa and Ukkonen, 1994) takes the positions of examples in the instance space to select features for decision trees. They limit their attention to boundaries separating examples belonging to different classes, while rewarding (penalizing) rules that separate examples from different (same) classes. Eight data sets are used to compare the performance (% accuracy, number of nodes in decision tree, time) of decision trees constructed using the proposed algorithm with ID3 (Quinlan, 1987). Decision trees generated using the proposed algorithm had better accuracy whereas those with ID3 had fewer number of nodes and took more than an order of magnitude less time.

Based on the positions of instances in instance space, the Relief algorithm (Kira and Rendell, 1992) selects features that are statistically relevant to target concept, using a relevancy threshold that is selected by the user. Relief is noise-tolerant and is unaffected by feature interaction. The complexity of relief is O(pn), where n and p are the number of instances and number

of features respectively. Relief was studied using two 2-class problems with good results, compared to FOCUS (Almuallim and Dietterich, 1991) and heuristic search (Devijver and Kittler, 1982), Kononenko (1994) extended RELIEF to deal with noisy, incomplete, and multi-class data sets.

Milne (1995) used neural networks to measure the contribution of individual input features to the output of the neural network. A new measure of input features' contribution to output is proposed, and evaluated using data mapping species occurrence in a forest. Using a scatter plot of contribution to output, subsets of features were removed and the remaining feature sets were used as input to neural networks. Setino and Liu (1997) present a similar study using neural networks to select features.

Battiti (1994) developed MIFS to use mutual information for evaluating the information content of each individual feature with respect to the output class. The features thus selected were used as input in neural networks. The author shows that the proposed method is better than those feature selection methods that use linear dependence (e.g., correlations as in Principal Components Analysis) measures. Koller and Sahami (1996) use cross-entropy to minimize the amount of predictive information lost during feature selection. Piramuthu and Shaw (1994) use C4.5 (Quinlan, 1990), to select features used as input in neural networks. Their results showed improvements, over just backpropagation, both in terms of classification accuracy and time taken by neural networks to converge.

The most popular feature selection methods in machine learning literature are variations of Sequential Forward Search (SFS) and Sequential Backward Search (SBS) as described in Devijver and Kittler (1982) and its variants (e.g., Pudil et al., 1994). SFS (SBS) obtains a chain of nested subsets of features by adding (subtracting) the locally best (worst) feature in the set. These methods are particular cases of the more general 'plus l - take away r' method (Stearns, 1976). Results from previous studies indicate that the performance using forward and backward searches are comparable. In terms of computing resources, forward search has the advantage since fewer number of features are evaluated at each iteration, compared to backward search where the process begins using all the features.

3 Feature Selection & Hausdorff Distance

In this section, the proposed method of feature selection using the Hausdorff (FSH) method is presented after a brief introduction to the Hausdorff distance measure.

3.1 Hausdorff Distance

The Hausdorff distance (e.g., Nadler, 1978) is a measure of the similarity, with respect to their position in metric space, of two non-empty compact sets A and B. It measures the extent to which each point in a set is located relative to those in another set. Let $X_1 = \{x_{11}, x_{12}, ..., x_{1m}\}$ and $X_2 = \{x_{21}, x_{22}, ..., x_{2n}\}$ be two finite point sets and d a distance over this space. Here, d can be any distance including the 1-norm[1], the Euclidean norm, as well as simple difference between corresponding coordinates in each dimension, among others. The Hausdorff distance is defined as follows:

$$\forall x_1 \epsilon X_1, D(x_1, X_2) = min_{x_2 \epsilon X_2}\{d(x_1, X_2)\} \quad (1)$$

$$h(X_1, X_2) = max_{x_1 \epsilon X_1}\{D(x_1, X_2)\} \quad (2)$$

$$H(X_1, X_2) = max\{h(X_1, X_2), h(X_2, X_1)\} \quad (3)$$

Here, $h(X_1, X_2)$ is the *directed* Hausdorff distance from X_1 to X_2. It identifies the point $x^* \epsilon X_1$ that is farthest (using a pre-specified norm) from any point in X_2 and measures the distance from x^* to its nearest neighbor in X_2. Essentially, $h(X_1, X_2)$ ranks each point in X_1 based on its distance from the nearest point in X_2 and then uses the largest ranked such point (x^*, the point in X_1 farthest away from X_2) as the distance. If $h(X_1, X_2) = A$, then each point in X_1 has at least one point in X_2 in the neighborhood of radius A. For smaller values of A, X_1 is nearly included in X_2. Hence, $h(X_1, X_2)$ is a measure of inclusion of X_1 in X_2. The Hausdorff distance itself, $H(X_1, X_2)$ is the maximum of the directed Hausdorff distances $h(X_1, X_2)$ and $h(X_2, X_1)$. $H(X_1, X_2)$ can be calculated in O(m,n) for two point sets of size m and n respectively. Alt et al. (1991) improve this to O((m+n)log(m+n)).

The Hausdorff distance H is a metric over the set of all closed, bounded sets (Csaszar, 1978). Being a true distance, it also obeys the properties of identity, symmetry, and triangle inequality. In the context of classification, it follows that description of a concept is identical only to its own description, the order of comparing different concepts does not matter, and the descriptions of two different concepts cannot be similar to some third concept.

3.2 Feature Selection with Hausdorff distance

The algorithm FSH assumes the data set to include k variables. Step 1 calculates the Hausdorff distance $H_i(X_1, X_2)$ between examples belonging to classes 1 and 2 (assuming a binary concept learning problem), individually for each of the k variables in the data set. This is followed by sorting the H_i values, and the corresponding variables are noted (s_1, .. , s_k) in step 2. This is followed by evaluation of the feature set by

[1] The 1-norm between two points $A(x_1, y_1)$ and $B(x_2, y_2)$ is defined as $d(A, B) = |x_1 - x_2| + |y_1 - y_2|$.

inducing decision tree. The trees are generated iteratively as more variables are added to the data set, in ascending order, based on their corresponding H_i values. The quality of the decision trees (e.g., classification accuracy on heretofore unseen examples) thus generated are evaluated. The algorithm stops when a pre-specified stopping criterion (e.g., classification accuracy) is reached, and the variables corresponding to this decision tree are returned as the selected set.

Algorithm FSH (Feature Selection with Hausdorff distance)

Variables in data: v_1, v_2, ... v_k.
S = set of all input variables = Ø.

1. Set i=1; While i < k+1, do
   (a) Calculate $H_i(X_1, X_2)$ for v_i.
   (b) i=i+1.
   (c) Go to 1(a).
2. Sort $H_i(.,.)$, in ascending order, with the corresponding features (s_1, .. , s_k).
3. Set j=k; Until stopping criterion is met, do
   (a) S = S + s_j
   (b) Induce decision tree with input S.
   (c) Evaluate quality of decision tree.
   (d) j=j-1.
   (e) go to 3(a).
4. Return final set of features.

---

4 Experimental Results

Using two financial credit scoring data sets with different characteristics - one on loan default prediction and the other on bank failure prediction - we illustrate the performance of the proposed feature selection method. To facilitate comparison of results from previous studies using these data sets (e.g., Abdel-Khalik and El-Sheshai, 1980; Piramuthu et al., 1998; Tam and Kiang, 1992), we follow the same split of training and testing (holdout) samples in accordance with these previous studies.

4.1 Loan Default Data

This data has been used in previous studies (e.g., Abdel-Khalik and El-Sheshai, 1980), to classify a set of firms into those that would default and those that would not default on loan payments. The source of this data is the Index of Corporate Events in the 1973-1975 issues of Disclosure Journal. Sixteen defaulted firms were matched with sixteen non-defaulted firms to obtain data for the study. Another set of sixteen examples, all belonging to the non-default case, were used as the holdout set in line with previous studies using this data set. There are 18 variables in this data: (1) net income/total assets, (2) net income/sales, (3) total debt/total assets, (4) cash flow/total debt, (5) long-term debt/net worth, (6) current assets/current liabilities, (7) quick assets/sales, (8) quick assets/current liabilities, (9) working capital/sales, (10) cash at year-end/total debt, (11) earnings trend, (12) sales trend, (13) current ratio trend, (14) trend of L.T.D./N.W., (15) trend of W.C./sales, (16) trend of N.I./T.A., (17) trend of N.I./sales, and (18) trend of cash flow/T.D. For detailed description of this data, the reader is referred to (Abdel-Khalik and El-Sheshai, 1980).

Table 1: Results using loan default data

Feature Selection Method	Input Variables (for C4.5)	tree size (C4.5)	Classification Accur. of Decision Trees (%) Training	Testing
none	x_1 .. x_{18}	15	96.9	87.5
FSH	x_2, x_5, x_6, x_7, x_9, x_{10}, x_{13}, x_{16}, x_{17}, x_{18}	7	84.4	87.5
Nonlinear	x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8, x_{13}, x_{15}	5	78.1	81.2

In order to compare FSH with a comparable method, the nonlinear sequential forward search method with Parzen and hyperspheric kernel is used. The nonlinear sequential forward search method with Parzen measure using hyperspheric kernel has been shown in previous studies (e.g, Piramuthu, 1998) to result in good performance compared to several other inter-class as well as probabilistic distance-based feature selection methods for induced decision trees. The number of variables chosen by the nonlinear method was guided by the number of variables chosen by the FSH method, for comparison purposes.

Table 1 provides results from decision trees generated after pre-processing input through the feature selection methods. In Table 1, 'none' corresponds to the case where no pre-processing (here, feature selection) was done. The classification accuracy on heretofore unseen testing (holdout) examples are of primary interest, and the number of input variables as well as the size of decision trees generated are also important though to a lesser degree. The classification accuracy of the decision tree after pre-processing through FSH is the same as that generated without any pre-processing. However, the same accuracy was obtained with fewer (10 compared to 18) features as well as a smaller (7 compared to 15) decision tree. Clearly, the ability to learn to describe a concept with fewer features as well as smaller decision tree is preferable in terms of the Occam's Razor principle as well the resources necessary to gather, store, maintain, analyze, and interpret results. Although the nonlinear method resulted in a smaller tree, the classification accuracy on holdout examples is not as good as the other two methods.

Table 2: Bank Failure Prediction Data

Feature Selection Method	Input Variables (for C4.5)	tree size (C4.5)	Classification Accuracy of Decision Trees (%) Training	Testing
none	x_1 .. x_{19}	29	99.2	79.5
FSH	x_1, x_5, x_6, x_7, x_8, x_9, x_{15}	9	84.7	86.4
Nonlinear	x_1, x_2, x_3, x_4, x_5, x_{10}, x_{17}	21	91.5	81.8

4.2 Bank Failure Prediction Data

This data set was used in the Tam and Kiang (1992) study. Texas banks that failed during 1985-1987 were the primary source of data. Data from a year prior to their failure were used. Data from 59 failed banks were matched with 59 non-failed banks, which were comparable in terms of asset size, number of branches, age and charter status. Tam and Kiang had also used holdout samples. The 1 year prior case consists of 44 banks, 22 of which belongs to failed and the other 22 to nonfailed banks. The data describes each of these banks in terms of 19 financial ratios. For a detailed overview of the data set, the reader is referred to Tam and Kiang (1992).

Table 2 provides results from decision trees generated after pre-processing input through the feature selection methods using bank failure prediction data. Unlike with the loan default data set, the classification accuracy of the decision tree after pre-processing through FSH is slightly better than that with the nonlinear feature selection method as well as that generated without any pre-processing. The size of the decision tree is also significantly smaller in the case of FSH compared to the other two methods. This is a slightly larger data set compared to the loan default data set, with more examples (118 training and 44 holdout examples in the bank failure prediction data, compared to 32 training and 16 testing examples in the loan default data set). The performance of the nonlinear method also is better than that using no pre-processing at all, in terms of tree size, using relatively fewer number of input variables.

5 Discussion

We developed and evaluated a feature selection method based on the Hausdorff distance measure, as to its effects on selecting features for inducing decision trees. This method was compared with a comparable sequential forward search algorithm as well as the case when no feature selection was used at all. In terms of classification accuracy on previously unseen examples, FSH performed slightly better than the nonlinear method with smaller decision trees.

The results also show that induced decision trees are sensitive to the input data used. By selecting appropriate features through pre-processing, the performance of induced decision trees can be improved without much effort since most of these pre-processing techniques are not time/computing intensive. This is true for any learning algorithm, because the complexity of the data used directly affects the learning algorithm's performance. Feature selection, when used along with *any* learning system, can help improve performance of these systems even further with minimal additional effort.

By selecting useful features from the data set, we are essentially reducing the number of features needed for learning tasks. This in turn translates to reduction in data gathering costs as well as storage and maintenance costs associated with features that are not necessarily useful for the decision problem of interest.

References

[1] Abdel-Khalik, A. R., and K. M. El-Sheshai, "Information Choice and Utilization in an Experiment on Default Prediction," *Journal of Accounting Research*, Autumn, pp. 325-342, 1980.

[2] Almuallim, H.M., and T. G. Dietterich, "Learning with Many Irrelevant Features," *Proceedings of the Ninth National Conference on Artificial Intelligence*, pp. 547-552, 1991.

[3] Battiti, R., "Using Mutual Information for Selecting Features in Supervised Neural Net Learning," *IEEE Transactions on Neural Networks*, 5, 4, pp. 537-550, 1994.

[4] Bradley, P. S., O. L. Mangasarian, and W. N. Street, "Feature Selection in Mathematical Programming," *INFORMS Journal on Computing*, 10, 2, 1998.

[5] Csaszar, A, *General Topology*, Bristol: Adam Hilger, 1978.

[6] Devijver, P. A., and J. Kittler, *Pattern Recognition: A Statistical Approach*, Prentice-Hall: 1982.

[7] Elomaa, T., and E. Ukkonen, "A Geometric Approach to Feature Selection," in *Proceedings of the European Conference on Machine Learning*, pp. 351-354, 1994.

[8] Hopkins, C., T. Routen, and T. Watson, "Problems with Using Genetic Algorithms for Neural Network Feature Selection," *11th European Conference on Artificial Intelligence*, pp. 221-225, 1994.

[9] John, G. H., R. Kohavi, and K. Pfleger, "Irrelevant Features and the Subset Selection Problem," *Proceedings of the Eleventh International Conference on Machine Learning*, pp. 121-129, 1994.

[10] Kira, K., and L. A. Rendell, "A Practical Approach to Feature Selection," in *Proceedings of the Ninth International Conference on Machine Learning*, pp. 249-256, 1992.

[11] Kittler, J., "Mathematical Methods of Feature Selection in Pattern Recognition," *International Journal of Man-Machine Studies*, 7, pp. 609-637, 1975.

[12] Kohavi, R., and B. Frasca, "Useful Feature Subsets and Rough Sets Reducts," *Third International*

Workshop on Rough Sets and Soft Computing (RSSC 94), 1994.

[13] Kohavi, R. *Wrappers for Performance Enhancement and Oblivious Decision Graphs*, Ph.D. Dissertation, Computer Science Department, Stanford University, 1995.

[14] Koller, D., and M. Sahami, "Toward Optimal Feature Selection," *Machine Learning: Proceedings of the Thirteenth International Conference*, 1996.

[15] Kononenko, I., "Estimating Attributes: Analysis and Extensions of RELIEF," *Proceedings of the European Conference on Machine Learning*, pp. 171-182, 1994.

[16] Malki, H. A., and A. Moghaddamjoo, "Using the Karhunen-Loe've Transformation in the Back-Propagation Training Algorithm," *IEEE Transactions on Neural Networks*, 2, 1, pp. 162-165, 1991.

[17] Meisel, W. S., *Computer-Oriented Approaches to Pattern Recognition*, Academic Press, New York, 1972.

[18] Milne, L., "Feature Selection using Neural Networks with Contribution Measures," *AI'95*, Canberra, November 1995.

[19] Modrzejewski, M., "Feature Selection Using Rough Sets Theory," *European Conference on Machine Learning*, pp. 213-226, 1993.

[20] Nadler, S. B., jr., *Hyperspaces of Sets*, New York: Marcel Dekker, 1978.

[21] Pawlak, Z, "Rough Sets," *International Journal of Computer and Information Sciences*, 11, 5, pp. 341-356, 1982.

[22] Piramuthu, S. "Evaluating Feature Selection Methods for Learning in Data Mining Applications," *HICSS-31*, pp. V:294-301, 1998.

[23] Piramuthu, S. and M. J. Shaw, "On Using Decision Tree as Feature Selector for Feed-Forward Neural Networks," *International Symposium on Integrating Knowledge and Neural Heuristics*, pp. 67-74, 1994.

[24] Piramuthu, S., H. Ragavan, and M. J. Shaw, "Using Feature Construction to Improve the Performance of Neural Networks," *Management Science*, 44, 3, pp. 416-430, 1998.

[25] Pudil, P., F. J. Ferri, J. Novovicova, and J. Kittler, "Floating Search Methods for Feature Selection with Nonmonotonic Criterion Functions," *IEEE 12th International Conference on Pattern Recognition - Vol. II*, pp. 279-283, 1994.

[26] Quinlan, J. R., "Simplifying Decision Trees," *International Journal of Man-Machine Studies*, 27, pp. 221-234, 1987.

[27] Quinlan, J. R., "Decision Trees and Decision Making," *IEEE Transactions on Systems, Man and Cybernetics*, 20, 2, pp. 339-346, 1990.

[28] Setino, R., and H. Liu, "Neural Network Feature Selector," *IEEE Transactions on Neural Networks*, 8, 3, pp. 654-662, 1997.

[29] Siedlecki, W. and J. Sklansky, "A Note on Genetic Algorithms for Large-scale Feature Selection," *Pattern Recognition Letters*, 10, 5, pp. 335-347, 1989.

[30] Stearns, S. D., "On Selecting Features for Pattern Classifiers," *Third International Conference on Pattern Recognition*, pp. 71-75, 1976.

[31] Tam, K. Y., and M. Y. Kiang, "Managerial Applications of Neural Networks: The Case of Bank Failure Predictions," *Management Science*, 38, 7, pp. 926-947, 1992.

[32] Yang, J., and V. Honavar, "Feature Subset Selection using a Genetic Algorithm," *Proceedings of the Genetic Programming Conference*, GP'97, pp. 380-385, 1997.