Optimized Implementation of a Pattern Classifier using Feature Set Reduction

M. Aldape-Pérez, C. Yáñez-Márquez, and L.O. López Leyva

Center for Computing Research, CIC
National Polytechnic Institute, IPN
Mexico City, Mexico
maldapeb05@sagitario.cic.ipn.mx; cyanez@cic.ipn.mx
WWW home page: http://www.cornelio.org.mx

Abstract. Performance in most pattern classifiers is improved when redundant or irrelevant features are removed, however, this is mainly achieved by high demanding computational methods or successive classifiers construction. In this paper new results on coherent feature selection, obtained by applying the Hybrid Classification and Masking technique (HCM), are presented. Previous results showed that the HCM algorithm proves to be a feasible way to get an optimal subset of features represented by a mask value. This method identifies irrelevant or redundant features for classification purposes; an optimal subset of features allows register size optimization which clearly contributes to significant power savings. Moreover classifier accuracy of this new approach has been preserved or even improved in comparison to other experimental results formerly shown. Promising experimental results suggest that the HCM algorithm is an appropriate alternative for optimized pattern classifier hardware implementation.

Keywords: Feature Selection, Pattern Classifier, Masking Techniques, Supervised Learning

1 Introduction

Pattern recognition has existed for many years in a wide range of human activity, however, the general pattern recognition problem can be stated in the following form: Given a collection of objects belonging to a predefined set of classes and a set of measurements on these objects, identify the class of membership of each of these objects by a suitable analysis of the measurements (features) [1]. Although features are functions of the measurements performed on a class of objects, most of the times, the initial set of features consists of a large number of potential attributes that constitute an obstacle not only to the accuracy but to the efficiency of algorithms. In countless situations, it is a complicated task to find proper features for all patterns in a class (especially when intra-class variance is very high) [2]. In order to overcome this limitation, multiclassifier approach arises. Some of the most frequently used are: multiple models [3], [4], [5]; combining classifiers [6]; and classifier ensembles [7] among others. On behalf of these

Control, Virtual Instrumentation and Digital Systems.
Research in Computing Science 24, 2006, pp. 11-20

novel approaches a remarkable thing to mention is the notable classification rate achieved; nonetheless these methodologies lack of criterions that help to ignore redundant or irrelevant information.

In this paper an original method for pattern classifier hardware implementation, whose basic operation rests on a Hybrid Associative Memory and the HCM algorithm for optimality criterion evaluation, is presented.

Each obtained mask value, represents a different subset of features. It is to be said that the best mask value is the one that indicates the smallest subset of features, and hence to permit register size optimization for hardware implementation purposes.

From a register transfer level perspective, the smaller the register size is, the better architecture alternative is obtained.

In the following section, a brief description of *HCM* foundations is presented. In Section 3, experimental results are shown over several different data sets. The masking approach advantages will be discussed in section 4, and a short conclusion follows in Section 5.

2 Associative Memories

An associative memory M is a system that relates input patterns and output patterns as follows: $x \longrightarrow \boxed{\mathrm{M}} \longrightarrow y$ with x and y, respectively, the input and output pattern vectors. Each input vector forms an association with its corresponding output vector. For each k integer and positive, the corresponding association will be denoted as: (x^k, y^k). Associative memory M is represented by a matrix whose ij-th component is m_{ij} [9]. Memory M is generated from an *a priori* finite set of known associations, called the fundamental set of associations. If μ is an index, the fundamental set is represented as: $\{(x^\mu, y^\mu) \mid \mu = 1, 2, ..., p\}$ with p as the cardinality of the set. The patterns that form the fundamental set are called fundamental patterns. If it holds that $x^\mu = y^\mu \; \forall \mu \in \{1, 2, ..., p\}$ M is auto-associative, otherwise it is heteroassociative; in this case it is possible to establish that $\exists \mu \in \{1, 2, ..., p\}$ for which $x^\mu \neq y^\mu$. If we consider the fundamental set of patterns $\{(x^\mu, y^\mu) \mid \mu = 1, 2, ..., p\}$where n and m are the dimensions of the input patterns and output patterns, respectively, it is said that $x^\mu \in A^n$, $A = \{0, 1\}$ and $y^\mu \in A^m$. Then the j-th component of an input pattern is $x_j^\mu \in A$. Analogously, the j-th component of an output pattern is represented as $y_j^\mu \in A$. Therefore the fundamental input and output patterns are represented as follows:

$$x^\mu = \begin{pmatrix} x_1^\mu \\ x_2^\mu \\ \vdots \\ x_n^\mu \end{pmatrix} \in A^n \qquad y^\mu = \begin{pmatrix} y_1^\mu \\ y_2^\mu \\ \vdots \\ y_m^\mu \end{pmatrix} \in A^m$$

2.1 The Steinbuch's Lernmatrix

Lernmatrix is a heteroassociative memory that can easily work as a binary pattern classifier if output patterns are appropriately chosen [13]. Typically it ac-

cepts binary patterns suchlike $\mathbf{x}^\mu \in A^n$, $A = \{0,1\}$ as input and returns binary patterns suchlike $\mathbf{y}^\mu \in A^m$ as output; it is worth pointing out that there are m different classes, each one coded by a simple rule: class $k \in \{1,2,...,m\}$ will be represented by a column vector which components will be assigned by $y_k^\mu = 1$, so $y_j^\mu = 0$ for $j = 1,2...,k-1,k+1,...m$.

The following matrix will keep the pattern association values after the Learning Phase for the Steinbuch's *Lernmatrix* is done:

	x_1^μ	x_2^μ	$\cdots$	x_j^μ	$\cdots$	x_n^μ
y_1^μ	m_{11}	m_{12}	$\cdots$	m_{1j}	$\cdots$	m_{1n}
y_2^μ	m_{21}	m_{22}	$\cdots$	m_{2j}	$\cdots$	m_{2n}
$\vdots$	$\vdots$	$\vdots$		$\vdots$		$\vdots$
y_i^μ	m_{i1}	m_{i2}	$\cdots$	m_{ij}	$\cdots$	m_{in}
$\vdots$	$\vdots$	$\vdots$		$\vdots$		$\vdots$
y_m^μ	m_{m1}	m_{m2}	$\cdots$	m_{mj}	$\cdots$	m_{mn}

(1)

Each one of the m_{ij} components of M is initialized with zero and will be modified by the following rule: $m_{ij} = m_{ij} + \Delta m_{ij}$ where:

$$\Delta m_{ij} = \begin{cases} +\epsilon \text{ if } y_i^\mu = 1 = x_j^\mu \\ -\epsilon \text{ if } y_i^\mu = 1 \text{ and } x_j^\mu = 0 \\ 0 \text{ otherwise} \end{cases} \tag{2}$$

and ϵ a positive constant, previously chosen.

The Recalling Phase for the Steinbuch's *Lernmatrix* consists of finding the class which an input pattern $\mathbf{x}^\omega \in A^n$ belongs to. Finding the class means getting $\mathbf{y}^\omega \in A^m$ that corresponds to $\mathbf{x}^\omega$; accordingly to the construction method of all $\mathbf{y}^\mu$, the class should be obtained without ambiguity. The i-th component of y_i^ω is obtained according to the following rule, where $\vee$ is the *maximum* operator:

$$y_i^\omega = \begin{cases} 1 \text{ if } \sum_{j=1}^n m_{ij}.x_j^\omega = \bigvee_{h=1}^m \left[\sum_{j=1}^n m_{hj}.x_j^\omega\right] \\ 0 \text{ otherwise} \end{cases} \tag{3}$$

2.2 Linear Associator.

It is worth pointing out that James A. Anderson and Teuvo Kohonen obtained amazingly similar results known nowadays as *Linear Associator*. Lets consider the fundamental set as $\{(\mathbf{x}^\mu, \mathbf{y}^\mu) \mid \mu = 1,2,...,p\}$ with

$$\mathbf{x}^\mu = \begin{pmatrix} x_1^\mu \\ x_2^\mu \\ \vdots \\ x_n^\mu \end{pmatrix} \in A^n \quad \text{y} \quad \mathbf{y}^\mu = \begin{pmatrix} y_1^\mu \\ y_2^\mu \\ \vdots \\ y_m^\mu \end{pmatrix} \in A^m$$

The *Learning Phase* is done in two stages.

1. Consider each one of the p associations $(\mathbf{x}^\mu, \mathbf{y}^\mu)$, so an $m \times n$ matrix is obtained by $\mathbf{y}^\mu \cdot (\mathbf{x}^\mu)^t$

$$\mathbf{y}^\mu \cdot (\mathbf{x}^\mu)^t = \begin{pmatrix} y_1^\mu \\ y_2^\mu \\ \vdots \\ y_m^\mu \end{pmatrix} \cdot (x_1^\mu, x_2^\mu, ..., x_n^\mu) = \begin{pmatrix} y_1^\mu x_1^\mu & y_1^\mu x_2^\mu & \cdots & y_1^\mu x_j^\mu & \cdots & y_1^\mu x_n^\mu \\ y_2^\mu x_1^\mu & y_2^\mu x_2^\mu & \cdots & y_2^\mu x_j^\mu & \cdots & y_2^\mu x_n^\mu \\ \vdots & \vdots & & \vdots & & \vdots \\ y_i^\mu x_1^\mu & y_i^\mu x_2^\mu & \cdots & y_i^\mu x_j^\mu & \cdots & y_i^\mu x_n^\mu \\ \vdots & \vdots & & \vdots & & \vdots \\ y_m^\mu x_1^\mu & y_m^\mu x_2^\mu & \cdots & y_m^\mu x_j^\mu & \cdots & y_m^\mu x_n^\mu \end{pmatrix} \tag{4}$$

2. M memory is obtained by adding all the p matrices

$$\mathbf{M} = \sum_{\mu=1}^{p} \mathbf{y}^\mu \cdot (\mathbf{x}^\mu)^t = [m_{ij}]_{m \times n} \tag{5}$$

in this way the ij-th component of M memory is expressed as:

$$m_{ij} = \sum_{\mu=1}^{p} y_i^\mu x_j^\mu \tag{6}$$

The *Recalling Phase* for the *Linear Associator* is done by operating the M memory with an input pattern $\mathbf{x}^\omega$, where $\omega \in \{1, 2, ..., p\}$; operate $\mathbf{M} \cdot \mathbf{x}^\omega$ as follows:

$$\mathbf{M} \cdot \mathbf{x}^\omega = \left[\sum_{\mu=1}^{p} \mathbf{y}^\mu \cdot (\mathbf{x}^\mu)^t \right] \cdot \mathbf{x}^\omega \tag{7}$$

$$\mathbf{M} \cdot \mathbf{x}^\omega = \mathbf{y}^\omega \cdot \left[(\mathbf{x}^\omega)^t \cdot \mathbf{x}^\omega \right] + \sum_{\mu \neq \omega} \mathbf{y}^\mu \cdot \left[(\mathbf{x}^\mu)^t \cdot \mathbf{x}^\omega \right] \tag{8}$$

Expression 8 lets us know about which restrictions have to be observed thus perfect recalling is achieved. These restrictions are expressed as:

$$(\mathbf{x}^\mu)^t \cdot \mathbf{x}^\omega = \begin{cases} 1 \text{ if } \mu = \omega \\ \\ 0 \text{ if } \mu \neq \omega \end{cases} \tag{9}$$

If condition 9 is met, then a perfect recalling is expected. So 8 is expressed as:

$$\mathbf{M} \cdot \mathbf{x}^\omega = \mathbf{y}^\omega.$$

2.3 Hybrid Classification and Masking approach

As it was said *Hybrid Classification and Masking* technique (HCM) are presented as a new feature selection approach to provide a mask that identifies the optimal subset of features, the best mask value is the one that indicates the smallest subset of features, and hence to permit register size optimization for hardware implementation purposes. In order to explain how optimal mask is found, some definitions are required.

Definition 1. *Let* $\mathbf{f}$ *be the number of features from the original set of data.*

Definition 2. *Let* $\mathbf{r}$ *be an index where* $r \in \{1, 2, ..., (2^f - 1)\}$

Definition 3. *Let* $\mathbf{e}^r$ *be a masking vector of size* n *represented as:*

$$\mathbf{e}^r = \begin{pmatrix} e_1^r \\ e_2^r \\ \vdots \\ e_n^r \end{pmatrix} \in B^n \tag{10}$$

where $B = \{0, 1\}$

Definition 4. *Let* $\dashv$ *be a new operation called IntToVector which takes* $r \in \{1, 2, ..., (2^f - 1)\}$ *and returns a column vector* $\mathbf{e}^r$ *with* r *value expressed in its binary form. From a register transfer level perspective (RTL) this can be expressed as* $bin(r) \rightarrow [\mathbf{e}^r]$. *For example: If* $r = 11$ *then* $\dashv \mathbf{e}^r$ *returns a column vector with* r *value in its binary form so the obtained vector is:*

$$\mathbf{e}^{11} = \begin{pmatrix} 1 \\ 0 \\ 1 \\ 1 \end{pmatrix}$$

where e_n^r is the Least Significant Bit (LSB)

Definition 5. *Let* $\|$ *be a new operation called MagVector which takes a column vector* $\mathbf{e}^r$ *of size* n *and returns an integer and positive value according to the following rule:*

$$\| \mathbf{e}^r = \sum_{j=1}^{n} \left(e_j^r \wedge 1 \right) \tag{11}$$

Where $\wedge$ is the logical AND operator.

Another relevant thing to mention is that the *Recalling Phase* is dramatically different from the previous models; it is carried out by the following rule:

$$y_i^{\mu} = \begin{cases} 1 \text{ if } \sum_{j=1}^{n} m_{ij} \cdot (x_j^{\mu} . e_j^{r}) = \bigvee_{h=1}^{m} \left[\sum_{j=1}^{n} m_{hj} \cdot (x_j^{\mu} . e_j^{r}) \right] \\ 0 \text{ otherwise} \end{cases} \tag{12}$$

where $\mu \in \{1, 2, ..., p\}$ and $r \in \{1, 2, ..., (2^f - 1)\}$

It is said that the *Recalling Phase* is dramatically different from the previous models because a masking vector $\mathbf{e}^r$ of size n masks each input vector x^{μ} of size n. This is where the masking technique comes into view. Using the previous definitions and the clear advantages that inherit from the *Hybrid Associative Classifier* model, it is possible to enunciate The HCM algorithm.

1. Let n be the dimension of each input pattern in the fundamental set, grouped in m different classes.
2. Each one of the input patterns belongs to a k class, $k \in \{1, 2, ..., m\}$, represented by a column vector which components will be assigned by $y_k^{\mu} = 1$, so $y_j^{\mu} = 0$ for $j = 1, 2..., k-1, k+1, ...m$.
3. Create a classifier using expression 4, 5 and 6.
4. Use the *IntToVector* operator to get the r-th masking vector as in expression 10
5. The recalling phase is carried out according to expression 12 so an r-th accuracy parameter is obtained
6. Store both parameters (the r-th accuracy parameter and the r-th masking vector) so feature selection can be evaluated in step 8
7. Compare the r-th accuracy parameter with the $(r-1)$-th accuracy parameter. The best accuracy value is stored thus accuracy improvements are achieved with each iteration
8. The same applies to the r-th masking vector. Feature selection can be evaluated using expression 11. So the smaller this number is, a better mask is obtained.
9. The new subset of features is obtained by a mask value represented by a column vector, where accuracy and feature selection are optimal

3 Experimental results

Throughout the experimental phase, two databases taken from the UCI Machine Learning Database Repository (http://www.ics.uci.edu/~mlearn) were included. The main characteristics of these data sets have been summarized in Table 1.

Contraceptive Method Choice and Australian Credit Approval databases were chosen because both data sets differ not only in features number, but in number of patterns and classes. The experiments have been carried out as follows: In the first (Learning) phase, the same number of input vectors for each class was randomly taken, which means that a balanced classifier is guaranteed. Particularly four vectors were taken from each class to get a total of twelve input vectors for the Contraceptive Method Choice and eight input vectors for

	CMC	Credit Approval
Number of Classes	3	2
Number of Patterns	1473	690
Original Set Size	9	14
Optimized Subset Size	**4**	**10**
Feature Optimization	**44%**	**71%**
Original Register Size	130 bits	360 bits
Optimized Register Size	**78 bits**	**120 bits**
Register Size Optimization	**40%**	**66%**

Table 1. Optimization Results of the data sets used

	1-NN	3-NN	C-Means	CHAT	HCM
CMC	37.24%	38.11%	31.84%	55.25%	**58.06%**
Credit Approval	58.33%	54.88%	50.81%	59.09%	**85.51%**

Table 2. Classification Rate of 5 different algorithms

the Australian Credit Approval. In the second (Recalling) phase the whole data set $\mathbf{x}^{\mu}$ was classified for each mask vector $\mathbf{e}^{r}$, in this way classifier accuracy for every mask was evaluated. The optimal mask was obtained by comparing the two main parameters (classifier accuracy and mask optimality). The same experimental procedure was carried out for each one of the two databases.

In order to estimate the HCM optimal mask efficiency, the developed method was tested by performing the following experiment several times. In the Learning Phase, the same number of input vectors for each class was randomly taken, conversely to the previous Recalling Phase, we conducted the classification process as a bi-class problem so a coherent comparison between HCM results and previous researchers' results can be done [16]. It is to be said that the results on each data set have been averaged over twenty experiments; all conducted using the same criterion. (summarized in Table 2.)

4 Results Analysis

As it is shown in Table 1 the original number of features for each database, CMC and Australian Credit Approval, is 9 and 14 respectively. After HCM algorithm performs a coherent feature selection, represented by a mask value, optimized subsets are obtained. Feature optimization for CMC database is close to 44% which means that only 56% is relevant information for classification purposes. While using near half of the initial features HCM algorithm improves classification rate over the other classification methodologies considered, consistent results are shown in Table 2.

Even more representative results of the HCM algorithm effectiveness appear with Australian Credit Approval database. Feature optimization for this database is close to 71% which means that only 29% of the initial set of features

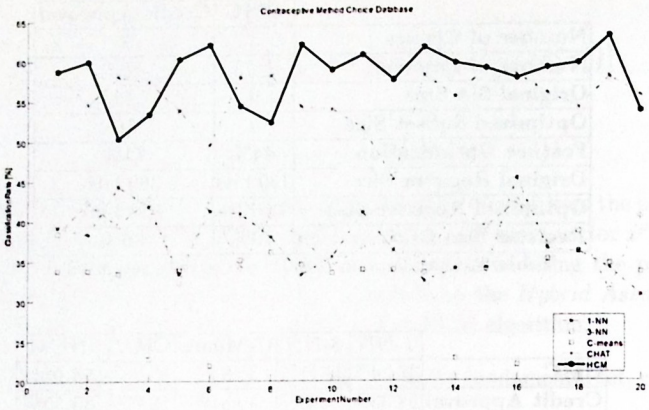

Fig. 1. Contraceptive Method Choice Database Classification Rate.

increases class separation. Whenever an optimal subset of features is obtained, astonishing classification rate will be achieved. Higher predictive accuracy can often be obtained when data dimensionality is reduced.

There are two remarkable facts to be taken into consideration. The former concerns about the number of features that was optimized for each database; 4 out of 9 features were coherently selected for the Contraceptive Method Choice database and 10 out of 14 features were coherently selected for the Australian Credit Approval database. The latter is that the HCM classification rate maintains higher predictive accuracy over the other classification methodologies considered.

5 Conclusions

In this paper an original method for optimized pattern classifier hardware implementation, whose basic operation rests on the HCM algorithm for optimality criterion evaluation, is presented. Experimental results have shown that this algorithm is an efficient way to get a mask value which represents the optimal subset of features. While maintaining the discriminatory information necessary for classifier accuracy improvement, these optimized subsets allow considerable register size reduction for hardware implementation purposes. A remarkable thing to mention is that feature optimization not only reduces data dimensionality, which represents an important reduction on hardware resources, it also means significant power savings on ASIC implementations due to register size reduction. After HCM algorithm performs a coherent feature selection, promising results have shown that accuracy has been improved in comparison to the other methodologies considered.

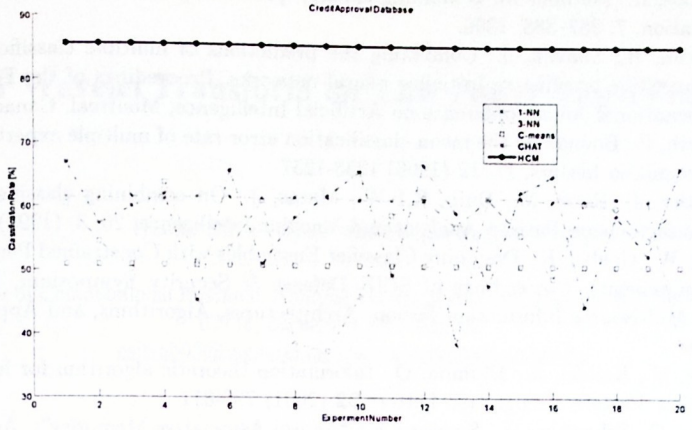

Fig. 2. Australian Credit Approval Database Classification Rate.

It is worth pointing out that this method does not have to compute a new classifier at each step, which represents an important reduction on computational costs. The learning phase of the process becomes swifter given the reduced number of input vectors considered. One and only one classifier is used throughout the entire process, which implies simplification on methodology. Classifier accuracy is not related to the number of input vectors considered in the learning phase, conversely, classifier accuracy is directly related to the mask value that allows elimination of redundant or irrelevant information.

Another clear advantage of this method is that the optimal mask search algorithm is applied only to those patterns that were previously considered during the learning phase, which means that no additional patterns are required to increase classifier accuracy.

This paper represents the initial works on optimized pattern classifier hardware implementation using feature selection and data dimensionality reduction.

Acknowledgments The authors of the present paper would like to thank the following institutions for their support: National Polytechnic Institute, Mexico (CIC, CGPI, PIFI, COFAA) , CONACyT and SNI.

References

[1] Nadler, M., Smith, E.: Pattern Recognition Engineering. John Wiley and Sons Inc., 1993.

[2] Nanni, L.: Cluster-based pattern discrimination: A novel technique for feature selection. Pattern Recognition Letters **27** (2006) 682-687

[3] Jacobs, R.: Methods for combining experts' probability assessments. Neural Computation, 7, 867–888, 1996.
[4] Maclin, R., Shavlik, J.: Combining the predictions of multiple classifiers: Using competitive learning to initialize neural networks. Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence, Montreal, Canada, 1995.
[5] Smyth, P.: Bounds on the mean classification error rate of multiple expert. Pattern Recognition Letters, 17, 12 (1996) 1253-1257
[6] Kittler, J., Hatef, M., Duin, R.P.W., Matas, J.: On combining classifiers. IEEE Transactions on Pattern Analysis and Machine Intelligence, 20, 3, (1998) 226-239
[7] Yan, W., Goebel, K.: Designing Classifier Ensembles with Constrained Performance Requirements. Proceedings of SPIE Defense & Security Symposium, Multisensor Multisource Information Fusion: Architectures, Algorithms, and Applications, 2004.
[8] Last, M., Kandel, A., Maimon, O.: Information-theoretic algorithm for feature selection. Pattern Recognition Letters **22** (2001) 799-811
[9] Palm, G., Schwenker, F., Sommer, F.: "Neural Associative Memories"., Associative Processing and Processors. Los Alamitos: IEEE Computer Society (1997) 307-326
[10] Dong, M., Kothari, R.: Feature subset selection using a new definition of classifiability. Pattern Recognition Letters **24** (2003) 1215-1225
[11] Gasca, E., Sánchez, J.S., Alonso, R.: Eliminating redundancy and irrelevance using a new MLP-based feature selection method. Pattern Recognition Letters **39** (2006) 313-315
[12] Kohonen, T.: Correlation Matrix Memories. IEEE Transactions on Computers. **21(4)**. (1972) 353-359
[13] Steinbuch, K. Die Lernmatrix. Kybernetik 1,1, (1961) 36-45.
[14] Anderson, J.A., Rosenfeld, E.: Neurocomputing: Fundations of Research, Cambridge: MIT Press. (1990)
[15] Hassoun, M. H. Fundamentals of Artificial Neural Networks, Cambridge: MIT Press. (1995)
[16] Santiago-Montero, R.: Hybrid Associative Classifier Based on Lernmatrix and Linear Associator (In Spanish). M.S. Thesis. Center for Computing Research, México (2003)
[17] Steinbuch, K. & Frank, H.: Nichtdigitale Lernmatrizen als Perzeptoren, Kybernetik, 1, 3, (1961) 117-124.
[18] Almuallim, H., & Dietterich, T.G. (1991).: Learning with many irrelevant features. Ninth National Conference on Artificial Intelligence (1991) 547-552. MIT Press.
[19] Shapira, Y., Gath, I.: Feature selection for multiple binary classification problems. Pattern Recognition Letters **20** (1999) 823-832