Proceedings of the V Brazilian Conference on Neural Networks - V Congresso Brasileiro de Redes Neurais
pp. 223–228, April 2–5, 2001 - Rio de Janeiro - RJ - Brazil

Feature Extraction and Pattern Classification Based on Bayesian Decision Boundaries

Lee Luan Ling[1] ,Hugo Mauro Cavalcanti[2]
DECOM-FEEC-UNICAMP, State University of Campinas
CP 6101, Campinas, SP, Brazil
E-mails:[1]lee@decom.fee.unicamp.br, [2]hugo@decom.fee.unicamp.br

Abstract

The implementation of a pattern recognition system requires solutions to some basic problems: data acquisition, feature extraction and pattern classification. In this paper a novel and efficient approaches for feature extraction for pattern classification using neural networks is proposed. The method search for the minimum amount of features necessary for solving a given pattern classification problem based on the structure of an adequately trained MLP network. Experimentally we show that all informative discriminating features can be obtained from decision boundaries specified by the MLP network.

1. Introdution

Feature extraction is regarded as dimension reduction which finds sets of vectors best representing observations for a large variety of applications. Different applications require different approaches for feature reduction. Most popular methods include the least mean error approximation and the minimum misclassification probability approach. Improper feature extraction procedures may cause the loss of information and therefore compromises the system's performance. In fact, how to find good features and the smallest number of features best representing patterns without causing the degradation on discriminant power of a classification system had received considerable attention from researchers in the field of pattern recognition [1,2,3].

In general, the minimum misclassification probability has been an ultimate criterion for system or classifier design for pattern classification problems. The Bayesian decision theory provides a fundamental methodology for solving statistical classification problems when the probability distribution of the pattern is known. In the absence of this information, alternatives for sub-optimum classification are many. Frequently one assumes that the class patterns have some specific distribution functions in order to make the analysis easier. Others choose some easily implemented classic discriminant functions in order to avoid probability distribution estimation tasks.

Many recent developments in the field of neural networks have provided another potential alternative solutions for feature extraction and pattern classification problems[14]. Among them, neural networks are capable of performing many different tasks; to our concern, it learns data statistical properties and performs classification.

Using neural networks for feature extraction and data projection has been widely studied [4-6]. The relationship between the discriminant analysis and neural networks has been studied [7,8] and the nonlinear discriminant analysis has been adapted to neural networks [9]. Brill *et. al.* proposed a genetic algorithm for feature selection in the context of neural networks [10]. Recently, Lee *et. al.* proposed a decision boundary feature extraction procedure for neural networks [12]. The proposed procedure makes use of training samples to determine a decision boundary and some discriminant features are determined based on the found decision boundary. Two serious problems may arise using this feature extraction method. The first one is of time consuming when the size of training database is not moderate. In this case, instead of searching for a complete decision boundary, a small part of it, the so-called *effective boundary*, is implemented for the given classification problem as soon as a certain system performance in terms of classification rates can be achieved theoretically. Notice that the effective boundary usually presents a classification performance inferior to that provided by the real Bayesian decision boundary.

In this work a novel and efficient feature extraction method for pattern classification using neural networks is proposed. The method searches the minimum amount of features necessary to preserve the classification accuracy achieved by the original feature vectors based on the structure of an adequately trained MLP network. Theoretically we show that all informative discriminating features can be obtained from decision boundaries specified by the MLP network and confirmed this result experimentally

2. Features Extraction

Consider a two pattern classification problem where w_1 and w_2 represent the two classes. A feature extraction problem can be viewed as determining the representation of an observation vector *X* original in the vector space R^N by a vector *Y* in a m-dimensional subspace R^M such that

$$[P(w_1, X) - P(w_2, X)] \cdot [P(w_1, Y) - P(w_2, Y)] > 0$$

It is natural to expect that successful feature extraction operation can assure the same classification performance no matter in which vector space the test is carried out. Another way to interpret this feature reduction result in terms of classification is that only redundant data were excluded. Redundant information can be defined as data that does not contribute to the classification. More specifically, Lee and Landgrebe showed that a redundant feature vector, when introduced, can not alter the classification result, because in the feature space it is parallel to the decision boundary [11]. On the other hand, when a vector is informative, definitely it carries some information which helps to define the pattern class. A feature vector is said highly informative when it is orthogonal to the decision boundary.

In addition, recently Lee and Landgrebe has presented a boundary feature extraction algorithm for neural networks [12]. Having a MLP trained by the backpropagation algorithm, their feature extraction method uses the network output to define decision boundaries, that is, the decision boundaries are geometric regions in the input feature space where the network output *h(X)* is zero, i.e., $\{X \mid h(X) = 0\}$. The decision boundaries for a given pattern classification problem are therefore determined by monitoring the network response to the reference class training feature vectors selected from different classes.

3. Decision Boundaries

It was said that multilayer perceptrons (MLP) are capable of performing arbitrary functions, provided that a suitable learning algorithm and a sufficient number of hidden neurons are available [5,6]. It can be shown that a feedfoward MLP trained by a backpropagation algorithm can be viewed as an approximation to a Bayesian discriminant function, therefore minimizing the classification error probability[14].

It can be shown that a single perceptron defines a desired decision surface for a linearly separable classification problem by adjusting its weight [15]. More explicitly, the weight vector $w=(w_1, w_2, ..., w_n)$ and decision threshold *T* of a trained perceptron specify a hyperplane in the input feature space, where *w* is a normal vector to this hyperplane. In particular, when the training algorithm is for minimization of classification error rate, *w* and *T* jointly determine the optimum decision hyperplane for the classification problem.

It is a well-known fact that feedforward networks with a single hidden layer are capable of forming either convex decision regions or non-convex but connected regions in the input feature space [16]. Specifically the procedure that a MLP network performs to compose minimum error decision boundaries is to combine several small hypersurfaces, the subsets of the corresponding hyperplanes determined by input-to-node connection weight vectors. For this end, the nodes of the second hidden and those of the output layer impose more restrictions over regions separated by those hypersurfaces to form the desired decision boundary. Note that the number of hyperplanes used in the input feature space for the construction of the decision boundary is equal to the number of hidden units in the first hidden layer

4. Feature Extraction Algorithm

In this section, we introduce the procedure used to extract a subset of highly discriminant feature vectors based on the decision boundary. A subspace spanned by this subset of vectors offers the same classification capability as the original one. Although we adopt the same definitions and notations proposed by Lee *et. al.* [12], our approach is fundamentally different in two important concepts: (a) optimum decision boundaries in terms of minimum classification error rates; (b) optimum decision hyperplanes determined by the structure of Bayesian neural classifiers.

Let $N(X)$ be the unit normal vector to the decision boundary at a point *X* for a given classification problem. The decision boundary feature matrix Σ_B is defined as

$$\Sigma_B = \frac{1}{K} \int_S N(x) N^T(x) p(x) dx \qquad (1)$$

where $p(x)$ is a feature vector density function, $K = \int_S p(x)dx$, and *S* is the decision boundary. Using MLP classifiers, Eq. (1) can be simplified into $\Sigma_B = \sum M_i M_i^T$, where *M* is a column vector orthogonal to the hyperplane *S* defined by the corresponding input-to-node weight vector.

Next, we introduce the concept of discriminating capability of a feature vector necessary to understand our feature extraction algorithm for classification.

Definition 1: A feature vector is said non-discriminant if it is parallel to every hypersurface composing the decision boundary.

Definition 2: A feature vector is discriminant if it is not a non-discriminant feature vector.

Definition 3: A feature vector is said highly discriminant if it is perpendicular to every hypersurface composing the decision boundary.

Then the proposed feature extraction principle can be illustrated via the following lemmas.

Lemma 1: The Bayes decision boundary can be asymptotically approximated by the hyperplanes (or hypersurfaces) defined by the weighting vectors of MLP when the number of hidden neurons in the hidden layer is large enough.

Lemma 2: The smallest number of discriminant feature vectors necessary for achieving the minimum classification error rate is equal to the number of non-zero eigenvalues of the decision boundary feature matrix S_B.

Lemma 3: The eigenvectors of the decision boundary feature matrix S_B of a pattern classification problem corresponding to nonzero eigenvalues are the features vectors necessary to attain the same classification performance as in the original feature space for the pattern classification problem.

Lemma 4: The smallest number of discriminant feature vectors for minimum error probability classification is given by the largest number of linearly independent w_i's.

Proposed Feature Extraction and Pattern Classification Algorithm:

1) Train a MLP with all reference feature vectors;
2) Find an orthonormal vector N_i for each hyperplane defined by the corresponding weighting vector;
3) Implement decision boundary feature matrix S_B;
4) Select those eigenvectors of S_B with significant eigenvalues to form the basis vectors for the new and reduced feature space;
5) Map all training feature vectors from the original feature space to the new feature space forming therefore a new reference database;
6) Train a MLP using the new reference database.

5. Experimental Studies

Two sets of experiments were carried out. Using synthetic gaussian data, the first set of experiments explicitly illustrate decision hypersurfaces (or hyperplanes) in the input feature space. The second set of experiments employs both synthetic gaussian data and real human signature data to elucidate our discriminant feature extraction algorithm. For sake of simplicity, without losing generality, we assume only uniform prior probabilities.

5.1. Example 1 (A Two-Class Problem)

On purpose, we elaborated a two-class problem with a common mean vector. A single hidden layer perceptrons with 10 neurons in the hidden layer is trained by 500 samples from each class. Precisely two classes have the following statistics, that is, mean vectors and covariance matrices:

$$M_1 = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, C_1 = \begin{bmatrix} 9 & 0 \\ 0 & 9 \end{bmatrix}, \; M_2 = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, C_2 = \begin{bmatrix} 0.5 & 0 \\ 0 & 0.5 \end{bmatrix}$$

Notice that two classes have a common mean vectors and the theoretical decision boundary is a cycle. Figure 1 shows how nine decision hyperplanes are located geometrically in the input feature space and help for the decision boundary formation.

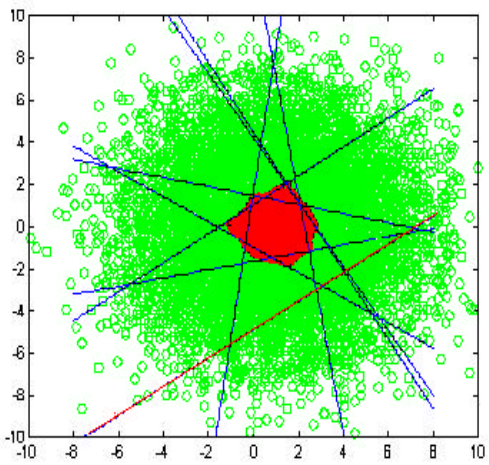

Figure 1. Decision hyperplanes in the input feature space

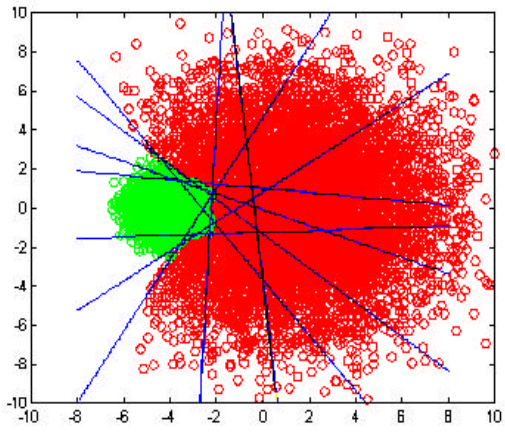

Figure 2. Decision hyperplanes in the input feature space

5.2. Example 2 (A Two-Class Problem)

Two gaussian pattern classes have the following statistics:

$$M_1 = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, C_1 = \begin{bmatrix} 9 & 0 \\ 0 & 9 \end{bmatrix}, M_2 = \begin{bmatrix} -4 \\ 0 \end{bmatrix}, \quad C_2 = \begin{bmatrix} 0.5 & 0 \\ 0 & 0.5 \end{bmatrix}$$

which have distinct mean vectors and covariance matrices. Figure 2 shows how 9 hyperplanes attempt to approximate to a parabola which is the optimum decision boundary.

5.3. Example 3 (A Linear Separable Problem)

This is a linearly separable problem where Gaussian data are generated for the following statistics:

$$M_1 = \begin{bmatrix} -2 \\ 0 \end{bmatrix}, C_1 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, M_2 = \begin{bmatrix} 2 \\ 0 \end{bmatrix}, \quad C_2 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

Table 1 shows the eigenvalues of the decision boundary feature matrix (S_B) along with the proportion of eigenvalue (Pro.Ev.) and the accumulation of eigenvalues (Acc.Ev.). The eigenvalues are sorted in the decreasing order. The classification accuracies (Cl.Ac.) obtained using the corresponding eigenvectors are also shown along the normalized classification accuracy (N.Cl.Ac.) obtained by dividing the classification accuracies by the classification accuracy obtained using all features.

Table 1. Eigenvalues of Σ_B and some statistics

	Ev.	Pro.Ev. (%)	Acc. Ev. (%)	Cl.Ac. (%)	N.Cl.Ac. (%)
1	356.75	99,94	99.94	97.56	99,94
2	0.1992	0,06	100	97.62	100

5.4. Example 4 (A Multi-Modal Problem)

This is a two class muiti-modal problem with the following class statistics:

$$M_{11} = \begin{bmatrix} -8 \\ -8 \end{bmatrix}, C_{11} = \begin{bmatrix} 5 & 2 \\ 2 & 5 \end{bmatrix}, M_{12} = \begin{bmatrix} -8 \\ -8 \end{bmatrix}, C_{12} = \begin{bmatrix} 7 & 2 \\ 2 & 7 \end{bmatrix}$$

$$M_{21} = \begin{bmatrix} -7 \\ 5 \end{bmatrix}, C_{21} = \begin{bmatrix} 4 & 2 \\ 2 & 4 \end{bmatrix}, M_{22} = \begin{bmatrix} -3 \\ -12 \end{bmatrix}, C_{22} = \begin{bmatrix} 7 & 4 \\ 4 & 7 \end{bmatrix}$$

Table 2 shows the eigenvalues of the decision boundary feature matrix (S_B) along with Pro.Ev., Acc.Ev., Cl.Ac., and N.Cl.Ac.

Table 2. Eigenvalues of S_B and some statistics

	Ev.	Pro Ev. (%)	Acc. Ev. (%)	Cl.Ac. (%)	N.Cl.Ac. (%)
1	200.465	56.81	56.81	98.86	99,32
2	152.401	43.19	100	99.53	100

5.5. Example 5 (A Multi-Dimensional)

This is a two-class problem with the following statistics:

$$M_1 = \begin{bmatrix} -6 \\ -4 \\ -8 \end{bmatrix}, C_1 = \begin{bmatrix} 9 & 4 & 5 \\ 4 & 8 & 4 \\ 5 & 4 & 9 \end{bmatrix}, M_2 = \begin{bmatrix} 5 \\ 5 \\ 5 \end{bmatrix}, \quad C_2 = \begin{bmatrix} 8 & 2 & 6 \\ 2 & 9 & 2 \\ 6 & 2 & 10 \end{bmatrix},$$

Table 3 shows the eigenvalues of the decision boundary feature matrix (S_B) along with Pro.Ev., Acc.Ev., Cl.Ac., and N.Cl.Ac.

Table 3. Eigenvalues of S_B and some statistics

	Ev.	ProEv. (%)	Acc. Ev. (%)	Cl.Ac. (%)	N.Cl.Ac. (%)
1	200.8037	99.37	99.37	99.15	99,96
2	1.1008	0.43	99.80	99.19	99.99
3	0.5121	0.20	100	99.20	100

5.6. Example 6 (A Multi-Class and Multi-Dimensional Problem)

This is a three class problem with the following statistics:

$$M_1 = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}, C_1 = \begin{bmatrix} 4 & 0 & 0 \\ 0 & 4 & 0 \\ 0 & 0 & 9 \end{bmatrix},$$

$$M_{21} = \begin{bmatrix} 5 \\ 0 \\ 0 \end{bmatrix}, C_{21} = \begin{bmatrix} 2 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 9 \end{bmatrix}, M_{22} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}, C_{22} = \begin{bmatrix} 4 & 0 & 0 \\ 0 & 4 & 0 \\ 0 & 0 & 9 \end{bmatrix},$$

$$M_{31} = \begin{bmatrix} 0 \\ 5 \\ 0 \end{bmatrix}, C_{31} = \begin{bmatrix} 9 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 9 \end{bmatrix}, M_{32} = \begin{bmatrix} 0 \\ -5 \\ 0 \end{bmatrix}, C_{32} = \begin{bmatrix} 9 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 9 \end{bmatrix}$$

Table 4 shows the eigenvalues of the decision boundary feature matrix (S_B) along with Pro.Ev., Acc.Ev., Cl.Ac., and N.Cl.Ac.

Table 4. Eigenvalues of S_B and some statistics

	Ev.	Pro Ev. (%)	Acc. Ev. (%)	Cl.Ac. (%)	N.Cl.Ac. (%)
1	602.159	69.08	69.08	60.60	71.02
2	251.493	28.5	97.93	84.95	99.56
3	18.0485	2.07	100	85.32	100

5.7. Example 7 (Handwritten Signature Recognition)

The database used in this experiments was originally derived from 550 human handwritten signatures contributed equally from 5 people, and a set of 32 feature values are extracted from each signature [17]. The 550 signature data are divided equally, half for neural classifier training and half for classifier testing. Since there are 5 classes, the implemented MLP classifier has 5 outputs each representing a class.

The implemented neural Baysian classifier has classification accuracy of 93.48% on the average. Table 5 lists the eigenvalues of the decision boundary feature matrix Σ_B along with Pro.Ev., Acc.Ev., Cl.Ac., and N.Cl.Ac. Interesting enough is the fact that a few number of the most discrimianant features (for instance, 5 to 10 features) are capable of achieving 98% of the maximum performance given by the Baysian classifier. This result suggests that many features out of 32 are discriminantly redundant. However, it should be noticed that many features (for instance, feature 11 to feature 21), although are relatively less discrimiant with respect to the high discriminant features, they are definitely essential to help to achieve the utmost performance.

Table 5. Eigenvalues of S_B and some statistics

	Ev.	Pro.Ev. (%)	Acc.Ev (%)	Cl.Ac. (%)	N.Cl.Ac. (%)
1	328.745	34.9327	34.9327	27.27	29.12
2	259.926	27.6199	62.5526	59.09	63.11
3	147.271	15.6492	78.2018	76.36	81.55
4	96.8837	10.2949	88.4967	86.54	92.42
5	76.3852	8.1167	96.6134	90.00	96.12
6	24.5991	2.6139	99.2273	90.36	96.50
7	3.4758	0.3693	99.5966	90.36	96.50
8	1.9930	0.2118	99.8084	90.90	97.08
9	0.5434	0.0577	99.8662	90.72	96.89
10	0.3159	0.0336	99.8997	91.45	97.67
11	0.1723	0.0183	99.9181	91.81	98.06
12	0.1171	0.0124	99.9305	92.18	98.45
13	0.0844	0.0090	99.9395	92.18	98.45
14	0.0735	0.0078	99.9473	92.36	98.46
15	0.0694	0.0074	99.9547	92.72	99.03
16	0.0606	0.0064	99.9611	92.54	98.84
17	0.0531	0.0056	99.9667	92.72	99.03
18	0.0485	0.0052	99.9719	92.72	99.03
19	0.0391	0.0042	99.9761	92.90	99.23
20	0.0334	0.0036	99.9796	92.90	99.23
21	0.0317	0.0034	99.9830	93.09	99.42
22	0.0279	0.0030	99.9859	93.09	99.42
23	0.0250	0.0027	99.9886	93.27	99.62
24	0.0220	0.0023	99.9909	93.27	99.62
25	0.0177	0.0019	99.9928	92.90	99.23
26	0.0164	0.0017	99.9946	93.27	99.62
27	0.0135	0.0014	99.9960	93.45	99.81
28	0.0115	0.0012	99.9972	93.27	99.62
29	0.0100	0.0011	99.9983	93.45	99.81
30	0.0066	0.0007	99.9990	93.63	100
31	0.0060	0.0006	99.9996	93.27	99.62
32	0.0036	0.0004	100	93.63	100

6. Conclusions

In this paper, we presented an alternative approach for feature extraction and pattern classification based on optimum decision boundaries. The optimum decision boundary, in the sense of minimizing the classification error probability, can be approximated by connected hypersurfaces in the input feature space, which are subset of hyperplanes defined by the structure and parameters of Bayes neural classifiers. Such an approximation is tight when the number of hidden neurons is sufficiently large. The use of neural classifiers allows us to solve some complex classification problems with arbitrary decision boundaries without assuming underlying probability distribution functions of the sample data, and still achieving the best classification performance.

Discriminability of feature vectors can be easily evaluated by analyzing geometric positions of the hyperplanes. For this end, a decision boundary feature matrix was defined and implemented using the normal vectors to the hyperplanes. It was stated that the number of discriminant feature vectors is equal to the number of eigenvectors of non-zero eigenvalues of the decision boundary feature matrix.

Experiments show that the proposed feature extraction and pattern classification method is extremely fast and efficient with respect to the method proposed in [12].

References

[1] O. Ersoy and D. Hong, "Parallel self-organizing hierarchical neural networks," *IEEE Trans. Neural Network*, vol. 1, June 1990.

[2] F. Fukushima and N. Wake, "Handwritten alphanumeric character recognition by the neocognitron," *IEEE Trans Neural Networks*, vol. 2, pp. 355-365, May 1991.

[3] A. Moonpenn, J. Lambe, and A. P. Thakoor, "Electronic implementation of associative memory based on neural –network models," *IEEE Trans.Syst. Man, Cybern.*, vol. SMC-17, pp. 325-331, Mar./Apr. 1987.

[4] P. Baldi and K. Hornik, "Neural networks and principal component analysis: Learning from examples without local minima," *Neural Networks*, vol. 2, pp. 53-58, 1989.

[5] P. Foldiak, “Adaptive network for optimal linear feature extraction,” *IEEE Int. Joint Conf. Neural Networks*, vol. 2, pp. 355-365, May 1991.

[6] A. K. Jain and J. Mao, “Artificial neural networks for nonlinear projection of multivariate data,” in Proc. *IEEE Int. Joint Conf. Neural Networks*, Baltimore, MD, 1992, pp.335-340.

[7] P. Gallinari, S. Thiria, F. Fogelman-Souli, “On the relations between discriminant analysis and multilayer perceptrons,” *Neural Networks*, vol. 4, pp. 349-360, 1991.

[8] A. R. Webb and D. Lowe, “On the relations between discriminant analysis and multilayer perceptrons,” *Neural Networks*, vol. 3, pp. 367-375, 1990.

[9] J. Mao and A. K. Jain, “Discriminant analysis neural networks,” in Proc. *IEEE Int. Joint Conf. Neural Networks*, San Francisco, CA, 1993, pp.300-305.

[10] F. Z. Brill, D. E. Brown and W. N. Martin, “Fast generic selection of features for neural network classifier,” *IEEE Trans Neural Networks*, vol. 3, pp. 324-328, May 1992.

[11] C. Lee and D. ª Landgrebe, “Feature extraction based on Decision Boundaries,” *IEEE Trans Patt. Anal. Machine Intell*., vol. 15, pp. 388-400, April 1993.

[12] C. Lee and D. ª Landgrebe, “Decision boundary feature extraction for neural networks,” *IEEE Trans .Neural Networks.*, vol. 8, pp. 75-83, Jan. 1997.

[13] S. Haykin, Neural Networks – A Comprehensive Foundation, Prentice Hall, 1994.

[14] H. Cavalvanti, Feature extraction based via neural networks, M.S. Thesis, UNICAMP, 2000.

[15] M. L. Minsky and S. A. Papert, *Perceptrons*, Cambridge, MA:MIT Press, 1969.

[16] J. Makhoul, A. El-Jaroudi and R. Schwarts, “Formation of disconnected decision regions with a single hidden layer,” Proc. *1989 Intl. Joint Conf. Neural Networks*, Washington, DC, pp. I-455-460, July 1989.

[17] M. G. Lizarraga, An automatic system for human signature visualization and verification, M.S. Thesis, UNICAMP, 1996