Mon. Not. R. Astron. Soc. **000**, 1–?? (2003) Printed 27 October 2003 (MN LATEX style file v2.2)

Machine Learning and Image Analysis for Morphological Galaxy Classification

Jorge de la Calleja and Olac Fuentes
Computer Science Department, National Institute of Astrophysics, Optics and Electronics, Tonantzintla 72840, Puebla, México

27 October 2003

ABSTRACT
In this paper we present an experimental study of machine learning and image analysis for performing automated morphological galaxy classification. We have used a neural network, and a locally weighted regression method, and also we implemented homogeneous ensembles of classifiers. The ensemble of neural networks was created using the bagging ensemble method, and manipulation of input features was used to create the ensemble of locally weighed regression. The galaxies used were rotated, centered, and cropped, all in a fully automatic manner. In addition, we have used principal component analysis to reduce the dimensionality of the data, and to extract relevant information in the images. Preliminary experimental results using 10-fold cross-validation show that the homogeneous ensemble of locally weighted regression produces the best results with over 91% accuracy considering 3 galaxy types (E, S and Irr), and over 95% accuracy for 2 classes (E and S).

Key words: machine learning, principal component analysis, ensembles.

1 INTRODUCTION

The morphology of galaxies is generally an important issue in the large scale study of the Universe. Galaxy classification is the first step towards a greater understanding of the physics of galaxies. In recent years, with numerous digital sky surveys across a wide range of wavelengths, astronomy has become an immensely data-rich field. For example, the Sloan Digital Sky Survey will produce more than 50,000,000 images of galaxies (Ball 2002) in the near future. Classification of these images is usually done by visual inspection of photographic plates. However, this task is not easy, because it requires skill and experience, and is also time-consuming: catalogues containing human classification take years to complete and contain only tens of thousands of entries (Naim et al. 1995). Thus, our goal is to bring automation to the classification of galaxies using machine learning algorithms and image analysis. Various galaxy classification studies using machine learning techniques have been carried out in the last decade. For example, in 1992 Storrie-Lombardi et al. used a feedforward neural network to classify galaxies into 5 classes: E, SO, Sa+Sb, Sc+Sd and Irr. They used a network configuration of 13:13:5 (13 inputs, 13 neurons in the hidden layer and 5 outputs). They used 5217 galaxies, randomly divided into two groups: a training set of 1700 images and a test set of 3517 images. They reported a 64% classification accuracy.

Naim et al. (1995) trained a neural network with different architectures to classify 831 galaxies. Most of their experiments used 13 input parameters that were derived from a set of 24 parameters using principal component analysis. The output layer had one node with a range of possible values from -5 to 10, which represented the galaxy classes. The best result was obtained using a 13:5:1 architecture (13 inputs, 5 nodes in the hidden layer and 1 output), with an rms deviation of 1.8 T types.

Owens et al. (1996) used oblique decision trees to perform the classification. They repeated the experiments made by Storrie-Lombardi et. al (1992), using the same data and features. They reported an overall accuracy of 63% using 1700 images for training and 3517 for testing, and 64.6% using 5-fold cross validation.

In 2001 Bazell and Aha used ensembles of classifiers for the classification of 800 galaxies. They used a Naive Bayes classifier, a neural network, and a decision-tree induction algorithm with pruning (J48). They considered from 2 to 6 classes, and 14 features to perform the classification. Considering 3 classes, they reported that the ensemble of J48 produced the best result with 78.55% accuracy. The ensemble of neural networks produced the best result with 50.18% accuracy in the case of 5 classes.

Recently, some significant works have been presented using new approaches. An alternative to morphological classification into discrete types is to define a continuum of galaxy types in an n-dimensional parameter space. An example of this approach is the work done by Abraham et al. (2003), where the Gini coefficient is introduced as a new tool for characterizing the morphologies of galaxy images.

Madgwick (2003) investigated two statistical techniques to determine how accurately morphology can be estimated from the optical spectrum of a galaxy; he reported that the best results were obtained by artificial neural networks, which correlated galaxy morphology with spectra about 70% accuracy for early galaxies (E+SO), and 83% accuracy for late galaxies (S+Irr).

In 2002 Odewahn et al. presented an approach to galaxy classification based on the Fourier reconstruction of galaxy images. He used artificial neural networks to train classifiers that recognize morphological bars at the 80%-90% confidence level and can identify the Hubble type with a 1 σ scatter of 1.5 steps on the 16 steps stage axis of the revised Hubble system.

Godeyra and Lolling (2002) used two types of automatic galaxy classifiers, the first one uses geometric shape features as the basis for classification, and the second uses the direct pixel images of galaxies and neural networks to do the classification. The direct image based neural network classifier was able to learn 97% of the 171 training patterns presented to it. When the network was presented a test set of 37 independent patterns, it was able to classify 57% of the test cases.

In this paper we present a method for performing automated morphological galaxy classification using machine learning and image analysis. We used two machine learning algorithms: neural networks and locally weighted regression. We also implemented homogeneous ensembles of classifiers, bagging for the neural network and manipulation of input features for locally weighted regression. In the image analysis stage we standardized the galaxy images, i.e., we rotate, center and crop the images, all in a fully automatic manner. In addition, we used principal component analysis to reduce the data and find features that characterize them. Our results show that the homogeneous ensemble of locally weighted regression obtained the best accuracy, and it is better than the best results reported in the literature, considering 2 and 3 classes.

The paper is organized as follows: Section 2 describes the classification method that we developed, and the data used. Section 3 presents the machine learning algorithms and the ensemble methods. Section 4 presents experimental results and Section 5 presents some conclusions and directions for future research.

2 THE CLASSIFICATION METHOD

We developed a method for classifying galaxy images that is divided in three parts (see Fig 1): the Image Analysis Module (IAM), the Data Compression Module (DCM), and the Machine Learning Module (MLM). The method works as follows: it takes as input the galaxy images, then they are rotated, centered, and cropped in the AIM; next in the DCM we use principal component analysis to reduce the dimensionality of the data and to find a set of features (principal components). The projection of the images onto the principal components will be the input parameters to the MLM. At the end, we will have the classification of the galaxies, and the accuracy of each machine learning algorithm, as well as the accuracy of the ensembles. The next three subsections describe in detail each module.

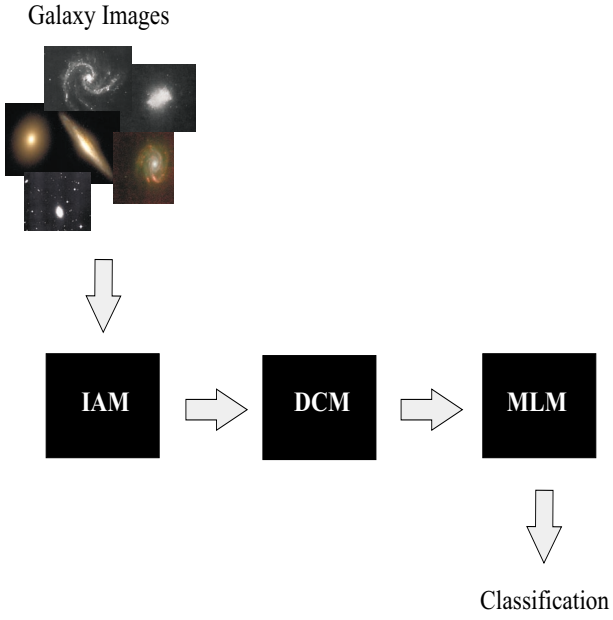


Figure 1. Parts of the classification method.

2.1 Image Analysis

The data set consisted of 310 images of galaxies. Most of them were taken from the NGC catalog on the web page of the Astronomical Society of the Pacific, and their classification was taken from the interactive NGC catalog online at www.seds.org. These images are of different sizes color format, and most of the time the galaxy contained in the image is not at the center. So, the aim of this module is to create invariant images to color, position, orientation and size.

First, we find the galaxy contained in the image applying a threshold, that is, from the original image I, we generate a binary image B, such that

$$B(i,j) = \begin{cases} 1 & if I(i,j) > threshold, \\ 0 & otherwise. \end{cases} \quad (1)$$

Then we obtain $\bar{i}$ and $\bar{j}$, the center row and column of the galaxy in the image, given by

$$\bar{i} = \frac{1}{m \times n} \sum_{i=1}^{m} \sum_{j=1}^{n} iB(i,j) \quad (2)$$

$$\bar{j} = \frac{1}{m \times n} \sum_{i=1}^{m} \sum_{j=1}^{n} jB(i,j) \quad (3)$$

where m and n are the number of rows and columns, respectively in the image. Then we obtain the covariance matrix of the points in the galaxy image

$$C = \sum_{i=1}^{m} \sum_{j=1}^{n} B(i,j)[i-\bar{i}, j-\bar{j}]^T [i-\bar{i}, j-\bar{j}] \quad (4)$$

The galaxy's main axis is given by the first eigenvector (the eigenvector with the largest corresponding eigenvalue) of C, the covariance matrix. We then rotate the image so that the main axis is horizontal (see Fig 2). The angle is given by

$$\theta = arctan(PC1[1]/PC1[2]) \quad (5)$$

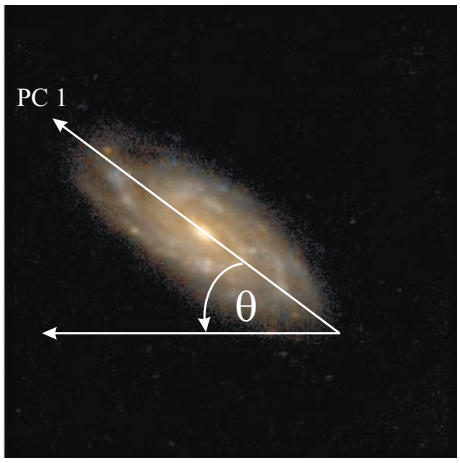


Figure 2. Rotation of the galaxy using the first principal component (PC1).

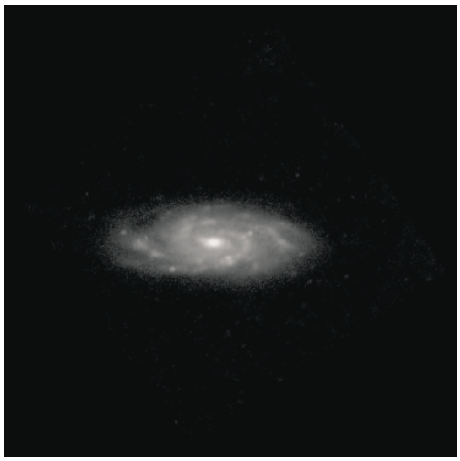


Figure 3. Rotated galaxy.

where PC1 is the first principal component. Then we use an image warping technique to rotate the image (see Fig 3).

After that we crop the image, eliminating the columns that contain only background (black) pixels. Finally, we stretch and standardize the images to a size of 128x128 pixels. Figure 2.1 shows examples of the image processing stage for an elliptical galaxy, a spiral galaxy, and an irregular galaxy.

2.2 Data Compression

A general idea for galaxy classification is to extract the relevant information in a galaxy image, encode it as efficiently as possible, and compare one galaxy encoding with a database of similarly encoded images. We have used principal component analysis (PCA) to find this relevant information. The basic idea in PCA is to find the components (the eigenvectors) of the covariance matrix of the set of galaxy images, so that they explain the maximum amount of variance possible by n linearly-transformed components. These eigenvectors can be thought of as a set of features which together characterize the variation among the objects (Turk and Pentland 1991), in this case the galaxy images. By using these eigenvectors to display a galaxy image, we can call it an *eigengalaxy*. In Figure 5 we show some of these images.

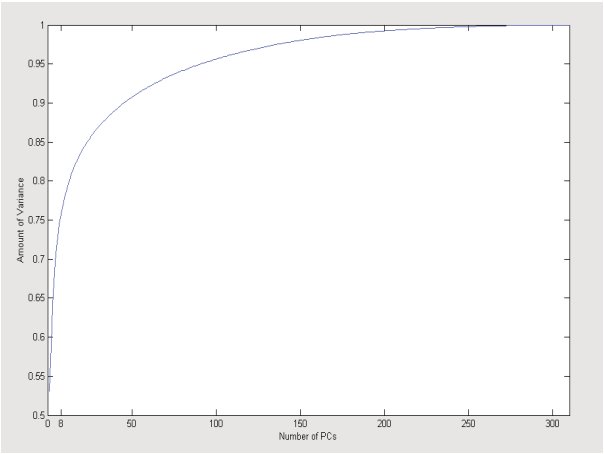


Figure 6. Amount of cumulative variance of the PCs.

The formulation of standard PCA is as follows. Consider a set of M objects $O_1, O_2, \ldots, O_M$, where the mean object of the set is defined by

$$X = \frac{1}{M}\sum_{n=1}^{M} O_n \tag{6}$$

Each object differs from the mean by the vector

$$\theta_i = O_i - X \tag{7}$$

Therefore, principal component analysis seeks a set of M orthogonal vectors v and their associated eigenvalues k which best describes the distribution of the data. The vectors v and scalars k are the eigenvectors and eigenvalues, respectively, of the covariance matrix

$$C = \sum_{i=1}^{M}\sum_{n=1}^{M} \theta_n \theta_n^T = AA^T \tag{8}$$

where the matrix $A = [\theta_1, \theta_2, \ldots, \theta_M]$.

The associated eigenvalues allow us to rank the eigenvectors according their usefulness in characterizing the variation among the objects.

Then, this module takes as input the set of processed images, and finds the principal components (PCs). The projection of the images onto the PCs are the input parameters to the machine learning algorithms, that is

$$Proj = PCs^T(I - M) \tag{9}$$

where I is the training set of images and M is the mean image.

We have used 8, 13 and 25 PCs because they represent about 70%, 75% and 85% of the information respectively in the data set. Figure 6 shows the amount of cumulative variance of the PCs, and Figure 7 shows the reconstruction of three galaxy images using 25 PCs.

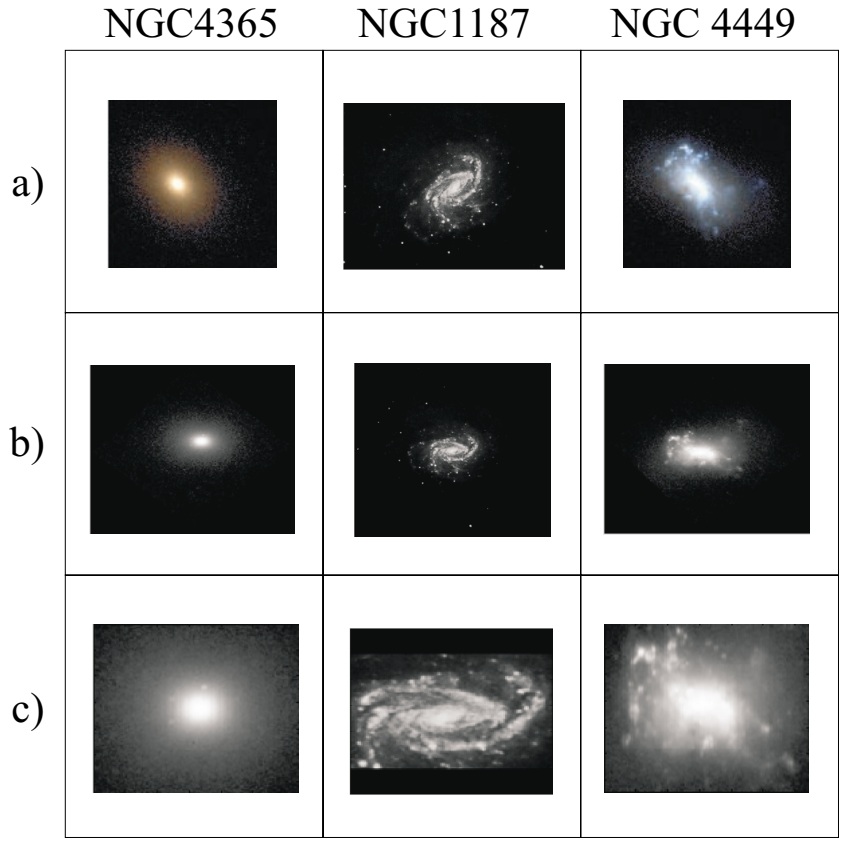


Figure 4. Examples of Galaxies, a) Original images, b) Rotated images, and c) Cropped images.

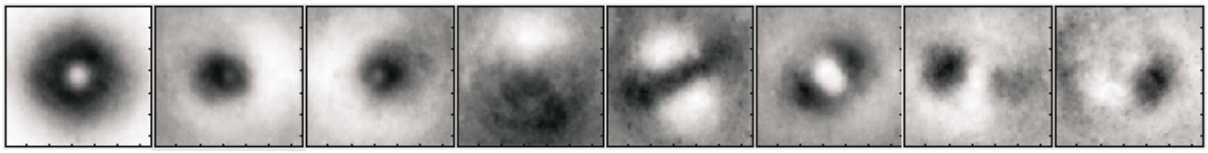


Figure 5. Eight eigengalaxies.

2.3 Machine Learning

In this module we have implemented two machine learning algorithms: a neural network and a locally weighted regression method. In addition we implemented homogeneous ensembles. Each algorithm and ensemble takes the projection of the images onto the PCs as input, and the classification and accuracy of each one will be the output of this module.

3 THE METHODS

In this section, we will give a description of the machine learning algorithms, and the ensemble methods.

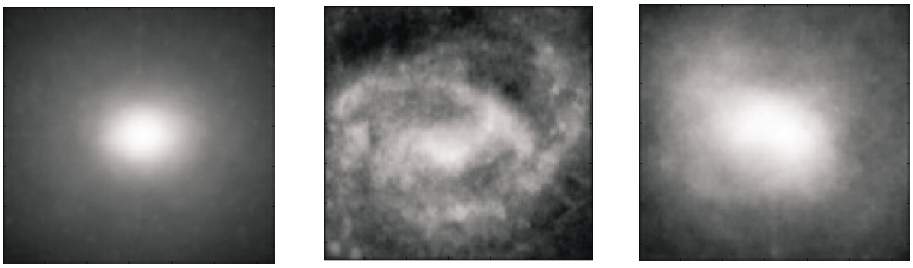
3.1 Neural Networks

We used a feedforward neural network with three layers trained with backpropagation, and the tan-sigmoid function as the transfer function. The input layer had different numbers of input nodes, depending on the number of PCs (8, 13 or 25). For the hidden layer one third of the input nodes. For the output layer 2, 3, 5 and 7 nodes according to the galaxy classification. The classes are: E and S; E+SO and S; E, S and Irr; E, SO, Sa+Sb, Sc+Sd and Irr; and E, SO, Sa, Sb, Sc, Sd and Irr.

3.2 Locally Weighted Regression

Locally weighted regression (LWR) belongs to the family of instance-based learning methods. In this algorithm, we simply store all the available training data, and when a new query instance is encountered, we find the training examples similar to the query, and we use them to classify the new query instance. LWR constructs an explicit approximation to the target function f over a local region surrounding a query point x_q (Mitchell 1997). In this work we use a linear model around the query point to approximate the target function.

Given a query point x_q, to predict its output parameters y_q, we assign to each example in the training set a weight



NGC4365 NGC1187 NGC4449

Figure 7. Reconstruction of galaxy images using 25PCs.

given by the inverse of the distance from the training point to the query point:

$$w_i = \frac{1}{|x_q - x_i|} \tag{10}$$

Let W, the weight matrix, be a diagonal matrix with entries $w_1, \ldots, w_n$. Let X be a matrix whose rows are the vectors $x_1, \ldots, x_n$, the input parameters of the examples in the training set, with the addition of a "1" in the last column. Let Y be a matrix whose rows are the vectors $y_1, , y_n$, the output parameters of the examples in the training set. Then the weighted training data are given by $Z = WX$ and the weighted target function is $V = WY$. Then we use the estimator for the target function defined as (Fuentes 2001)

$$y_q = x_q^T Z^* V \tag{11}$$

where Z^* is the pseudoinverse of Z.

Although LWR is normally applied to regression problems, it is easy to adapt it to perform classification tasks. For an n-class classification problem, we supply as output parameter for each example a vector with n-elements, where the ith element of the vector is 1 if the example belongs to class i and 0 otherwise. When we classify a test example, we assign it to the class with the highest corresponding value in the output vector.

3.3 Ensemble methods

An ensemble of classifiers consists of a set of classifiers whose individual decisions are combined in some way, normally by voting. Ensembles often yield better results than individual classifiers. We used *bagging* (Dietterich 1997) for the ensemble of neural networks and *manipulation of input features* (Dietterich 1997) to create the ensemble of locally weighted regression.

The idea in bagging is to generate randomly n subsets with the examples from the original training set, and to use each of this subsets for creating a classifier. Each subset is obtained by randomly sampling, with replacement, from the original training set, thus some of the examples will appear more than once in the subset, while others will not appear at all.

In manipulation of input features we randomly select two thirds of the features available to the learning algorithm to create each classifier.

In the homogenous ensemble, the same learning algorithm is implemented by each member of the ensemble, and they are forced to produce non-correlated results by using different training sets. We constructed three and seven classifiers and combined their results by unweighted voting, and if there is a tie, we assign the most common class.

4 EXPERIMENTAL RESULTS

We implemented locally weighted regression in Matlab, and used the feedforward network that is implemented in the Matlab Neural Network Toolbox. We used 10-fold cross-validation for all the experiments, that is, we randomly divided the original dataset into ten equally-sized subsets and performed ten experiments, using in each experiment one of the subsets for testing and the other nine for training.

The network had different configurations depending on the number of PCs as input, number of nodes in the hidden layer, and nodes in the output layer. Thus we had the following architectures: 8:3:3(8 nodes as input, 3 nodes in the hidden layer, and 3 nodes for the classes), 13:5:3, 25:9:3, 8:3:5, 13:5:5, 25:9:5, 8:3:7, 13:5:7 and 25:9:7. The network was trained for 200 epochs using the backpropagation algorithm, and with a learning rate of 0.0015. To build the linear model around the query point, in locally weighted regression, we used the 200, 250, 270 and 293 closest points for 2, 3, 5 and 7 classes respectively, and we weighted each point using their Euclidian distance to the query point.

We reduce the data set to 293 galaxies for doing the experiments with 5 and 7 classes, because seventeen spiral galaxies do not have the subclassification O, a, b, c or d.

Tables 1 and 2 show the accuracy for each of the individual classifiers, and for the homogeneous ensembles. This was obtained by averaging the results of 5 runs of 10-fold cross-validation for each method. The columns *Ind*, $E3$ and $E7$ mean Individual classifier, Ensemble of 3 classifiers and Ensemble of 7 classifiers, respectively.

As we can see in Table 1, the ensemble of 7 classifiers obtained the best results considering 3 and 7 classes, with 91.80% and 44.63% accuracy respectively, while the individual classifier obtained the best accuracy for 5 classes. In almost all cases the ensembles slightly improve the individual classifiers results, although this small improvement might not be enough to justify the increase in the running time.

The ensemble of 7 classifiers of neural networks obtained the best results for 3 and 5 classes with 90.58% and 50.29% accuracy respectively (see Table 2). Considering 7 classes, the ensemble of 3 classifiers obtained the best result with 44.23% accuracy.

Table 1. Accuracy for LWR. In bold the best results.

	Locally Weighted Regression								
	3 classes			5 classes			7 classes		
PCs	*Ind*	*E*3	*E*7	*Ind*	*E*3	*E*7	*Ind*	*E*3	*E*7
8	91.14	90.96	91.67	50.72	50.50	50.71	43.37	44.49	44.39
13	91.09	91.03	91.22	**52.29**	50.52	50.22	44.34	43.74	**44.63**
25	90.51	91.16	**91.80**	48.12	49.79	49.88	39.06	42.54	42.45

Table 2. Accuracy for ANNs. In bold the best results.

	Artificial Neural Networks								
	3 classes			5 classes			7 classes		
PCs	*Ind*	*E*3	*E*7	*Ind*	*E*3	*E*7	*Ind*	*E*3	*E*7
8	89.60	90.45	**90.58**	45.71	47.09	46.88	41.84	**44.23**	43.42
13	88.96	89.54	**90.58**	46.87	45.95	49.00	39.61	41.35	43.08
25	85.99	88.96	90.06	44.25	47.70	**50.29**	37.29	40.14	43.66

Table 3. Accuracy for LWR and ANNs considering 2 classes. In bold the best results.

	LWR		ANNs	
Classes	*Ind*	*E*7	*Ind*	*E*7
E - S	94.04	**95.11**	93.77	93.82
E+SO - S	89.29	**90.36**	86.08	87.08

The results for LWR and ANNs considering 2 classes are summarized in Table 3. We only consider elliptical and spiral galaxies because the number of irregular galaxies is significantly smaller in the data set. We use 13 PCs as input parameters and perform experiments using the individual classifier and the ensemble of seven classifiers. Thus the ANN had a configuration 13:5:2. As we can observe, LWR again obtained the best results with 95.11% accuracy for E-S, and 90.36% accuracy for E+SO-S.

4.1 Discussion

Our results show that locally weighted regression obtains better results than artificial neural networks, and also it is faster. However, the ensembles of neural networks improve their individual classification more than the ensembles of locally weighted regression. It happens because artificial neural networks is an unstable algorithm, and bagging works especially well in this kind of algorithms (Dietterich 1997).

Because locally weighted regression obtains almost the same results using individual classifiers or ensembles, we recommend to use an individual classifier with 13 principal components.

In tables 4, 5 and 6 we present the confusion matrix for the best classifier obtained for 3, 5 and 7 classes. We can observe that none of them classified irregular galaxies correctly. We suppose that it happens due to the small set of this kind of galaxies we have for training. However, we consider the method could classify them correctly by supplying to it a larger data set. That is why we did experiments considering only elliptical and spiral galaxies. Tables 7 and 8 show the confusion matrix for the best classifier for the case of 2 classes.

We can notice the percents in these tables are better than the indicated in tables 1 and 2, because they are obtained considering the best precision of the classifier for one run, instead of the average of 5 runs.

Table 4. Confusion matrix for best 3 class case, 92.58% accuracy.

	E	S	Irr
E	10	8	0
S	4	277	0
Irr	0	11	0

Table 5. Confusion matrix for best 5 class case, 56.33% accuracy.

	E	SO	Sa+Sb	Sc+Sd	Irr
E	10	1	4	3	0
SO	2	4	6	7	0
Sa+Sb	8	3	49	44	0
Sc+Sd	3	1	35	102	0
Irr	1	0	2	8	0

Table 6. Confusion matrix for best 7 class case, 48.50% accuracy.

	E	SO	Sa	Sb	Sc	Sd	Irr
E	9	1	1	6	1	0	0
SO	2	4	1	3	9	0	0
Sa	3	4	4	9	6	0	0
Sb	3	1	2	18	53	1	0
Sc	2	1	1	21	107	0	1
Sd	0	0	0	0	8	0	0
Irr	0	0	0	2	9	0	0

Table 7. Confusion matrix for best 2 class case (E and S), 95.66% accuracy.

	E	S
E	8	10
S	3	278

5 CONCLUSIONS

We have shown experimentally that the algorithms and the ensembles obtained very good results. In fact, our results show that the homogeneous ensemble of locally weighted regression obtained better accuracy than the ones reported in the literature, considering 2 and 3 classes. The use of standardized images helps to improve the classification accuracy. Principal component analysis is useful to reduce the data, and we have shown that a small number of principal components is enough for classifying the galaxies, in fact, those that represent about 75% of information in the data set suffice. Locally weighted regression is faster than artificial neural networks and obtained better results.

Future work includes:

- Repeating the experiments with a larger set of galaxies.
- Using other machine learning algorithms such as support vector machines.

ACKNOWLEDGMENTS

We would like to thank CONACyT for partially supporting this work under grants J31877-A and 166596.

Table 8. Confusion matrix for best 2 class case (E+S0 and S), 91.65% accuracy.

	E+SO	S
E+SO	17	20
S	5	257

REFERENCES

Abraham R.G. et al., 2003, ApJ, 588, 218
Ball Nicholas M., Master's thesis, Universisy of Sussex, 2002
Bazell D., Aha D.W., 2001, ApJ, 548, 219-223
Dietterich T.G., 1997, AI Mag., 18(4), 97-136
Fuentes O., 2001, In Proceedings of the IASTED International Conference of Artificial Intelligence and Soft Computing (ASC2001)
Goderya S.N., Lolling S.M., 2002, ASS, 279, 377
Madgwick D.S., 2003, MNRAS, 338, 197
Matlab Neural Network Toolbox Documentation, 2003, http://www.mathworks.com/access/helpdesk/help/toolbox/nnet/nnet.shtml
Mitchell T.M., 1997, Machine Learning, (New York: McGraw-Hill)
Naim A., Lahav O., Sodré, Jr., Storrie-Lombardi M.C., 1995, MNRAS, 275, 567
NGC Images on the Net of the ASP, 2003, http://www.apsky.org/ngc/ngc.html
Odewahan S.C. et al., 2002, ApJ, 568, 539
Owens E.A., Griffiths R.E., Ratnatunga K.U., 1996, MNRAS, 281, 153
Storrie-Lombardi M.C., Lahav O., Sodré L., Jr., Storrie-Lombardi L.J., 1992, MNRAS, 259, 8
The Interactive NGC Catalog Online of SEDS, 2003, http://www.seds.org/~spider/ngc/ngc.html
Turk M.A., Pentland A.P., 1991, In Proceedings of the IEEE Conf. on Computer Vision and Pattern Recognition